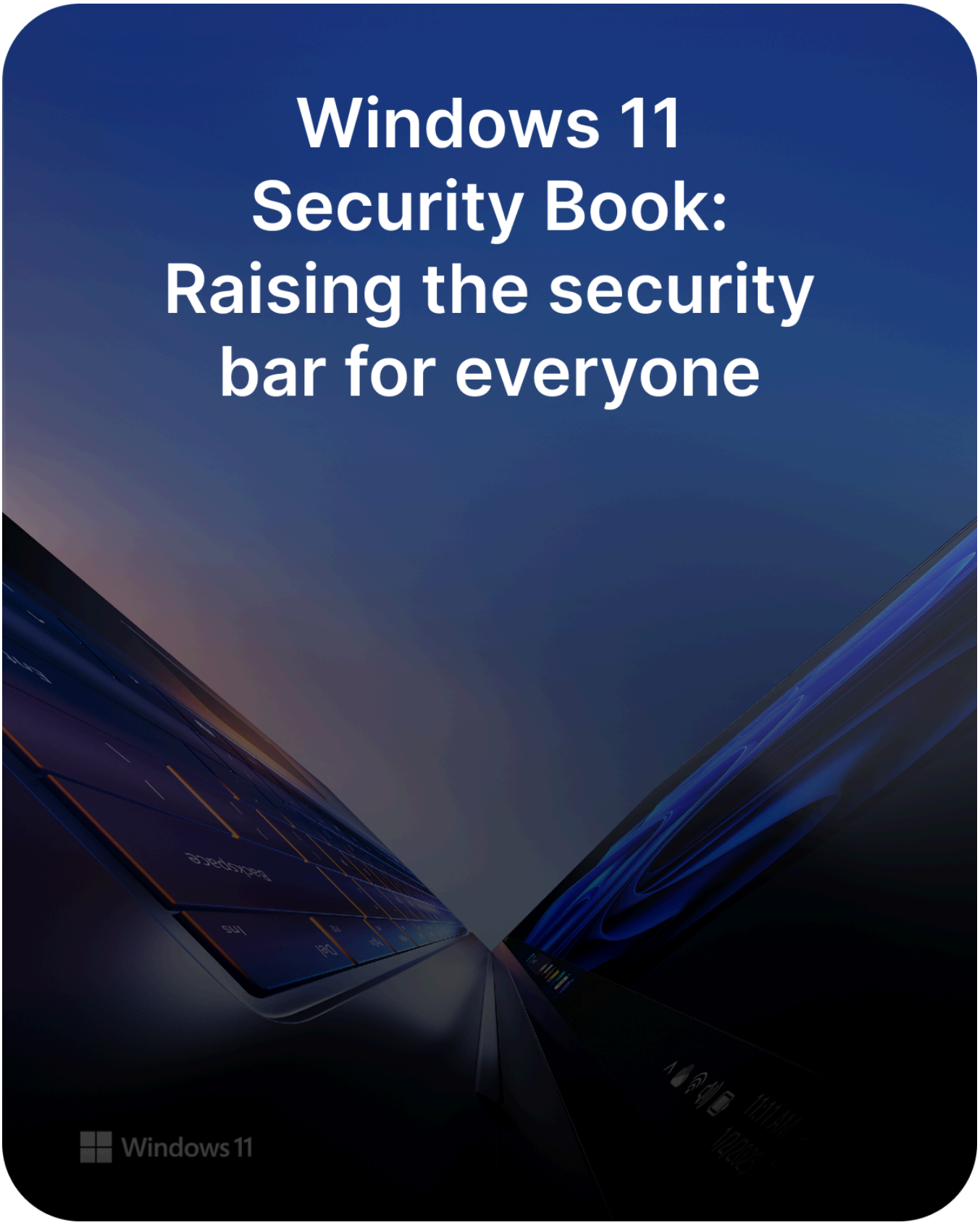
Tell us about your PDF experience.

Windows 11 Security Book (2025 edition)

Applies to: Windows 11

Summarize this article for me



Introduction

The Security Book not only provides valuable technical information, it also shows how we think about elevating security and resiliency for every organization.

Each Windows release brings new capabilities, stronger defenses, and relentless innovation—because the stakes have never been higher. Today, digital transformation and AI adoption are accelerating at breakneck speed. But so are the tactics of cybercriminals. Leading threats include identity attacks, ransomware, targeted phishing attempts, and business email compromise[1].

Attackers don't always break in—they sign in, targeting people, applications, and devices at the heart of every organization. That's why security isn't just a feature in Windows; it's our foundation and our future. Even if you don't have a squad of security specialists, Windows is designed to help you defend against the most common and damaging attacks—like phishing, malware, and credential theft—right out of the box.

Microsoft's commitment: security by design; security by default

Microsoft prioritizes security over all else. Announced in 2023, the Microsoft Secure Future Initiative (SFI) is dedicated to advancing cybersecurity in everything we create and provide. Products and services are secure by design and secure by default. For secure operations, we continuously apply what we learn from incidents to improve our security and privacy models, security architecture, and technical controls.

Through the SFI we have mobilized the equivalent of 35,000 full-time engineers committed to the highest security standards into how we design, build, test, and operate our products and services. We synthesize more than 100 trillion signals daily to understand digital threats and criminal cyberactivity[1]. And we thwarted US$4 billion in fraud attempts in one year with new policies, detection models, and investigation methods[2].

Working together with a shared focus is key to improving global security, from individuals and organizations to governments and industries. The world is moving toward a secure by design and secure by default approach, where technology producers are tasked with incorporating security during the initial design phase, and offering products that deliver protection right out of the box. As part of our commitment to making the world a safer place, we build security into every innovation, including Windows 11.

This secure by design approach spans the Windows edition range including Home, Pro, Enterprise, Enterprise IoT, and Education editions. Hardware and software work together to optimize security, and Copilot+ PCs offer the highest level of Windows protection. Copilot+ PCs are the fastest, most intelligent, and secure Windows devices ever. These groundbreaking PCs come with Secured-core PC protection and the latest safeguards like the Microsoft Pluton security processor and Windows Enhanced Sign-in Security enabled by default.

Better together protection for a modern security landscape

Windows 11 can strengthen security while reducing costs through simpler management and greater flexibility. Windows PCs are easily managed at scale with your modern device management solution like Microsoft Intune, with hundreds of configuration possibilities to meet unique business needs.

Seamless integration with solutions like Microsoft Defender, Entra ID, and Intune[3] opens to the door to multiple benefits, including passwordless authentication, granular management of security policies across endpoints, and ability to leverage new time-saving technologies that increase efficiency and protection.

For example, Windows backup for organizations strengthens identity-based recovery and protection with automated backup from PC to cloud, including user data and personalized settings. Windows Autopatch and hotpatch automate upgrades and updates for Windows PCs and Microsoft 365 apps—covering security and productivity features, drivers, and firmware—with hotpatch enabling faster security and compliance by applying critical updates without a restart[4]. Organizations can also take advantage of new Microsoft Security Copilot agents in Intune to save significant time investigating, identifying, and remediating threats.

With Windows at the center, organizations benefit from layers of security and intelligent management solutions. As a result, they can increase resilience and adopt the latest technologies with confidence, including advanced AI.

Security priorities and benefits

Windows 11 enables you to focus on your work, not your security settings. Hardware and software work together to shrink the attack surface, protect system integrity, and shield valuable data.

Out-of-the-box features such as credential safeguards, malware shields, and application protection led to a reported 62% drop in security incidents, including a 3x reduction in

firmware attacks[5].

> Businesses reported a 62% drop in security incidents, including a 3x reduction in firmware attacks[5].

Identity and data protection

Attackers increasingly target employees and their devices. Windows 11 provides proactive protection against credential theft, with Windows Hello and TPM 2.0 shielding identities, and features like passkeys and secure biometric sign-in virtually eliminating the risk of lost or stolen passwords. Enhanced phishing protection means businesses report 2.9x fewer instances of identity theft with hardware-backed protection.

Looking ahead, Windows 11 is already integrating post-quantum cryptography algorithms to safeguard sensitive information against emerging threats from quantum computing.

> A streamlined, chip-to-cloud security solution based on Windows 11 improved productivity for IT and security teams by a reported 25%[7].

Application safeguards

Robust application controls keep business data secure and employees productive. Windows App Control for Business ensures only trusted code runs, while Trusted Signing enables developers to effortlessly sign their applications, supporting authenticity and integrity. Integrated privacy controls and least-privilege principles help organizations and regulators trust that critical data is protected.

Device health and access control

Windows 11 and chip-to-cloud security provide the tools to attest that devices connecting to your network are trustworthy. Enforce security policies and conditional access with cloud based device management solutions like Intune and Entra ID, plus comprehensive security baselines. Security by default enables secure work anywhere and simplifies IT—businesses with Copilot+ PCs anticipate up to 30% drop in device management time due to improved reliability and support for proactive diagnostics[7].

Chip-to-cloud security

With Windows 11 devices, hardware and software work together to protect sensitive data from the core of your PC all the way to the cloud. Comprehensive protection helps keep your organization secure, no matter where people work. Microsoft cloud offerings include Windows 365 Cloud PCs, which are accessible from any device and provide Windows security with the power and scalability of the cloud.



Conclusion

We'll continue to innovate with security by design and security by default at the heart of every new Windows 11 PC and Windows 11 IoT device. This commitment ensures our products not only meet but exceed the security expectations of our customers—providing robust protection against modern cyber threats while maintaining ease of use and performance. With intelligent solutions based on Windows, you can innovate with confidence, equipped with enterprise-grade security by default, granular policy management, and device-to-cloud resilience.

Learn more

- Windows security features licensing and edition requirements

---

Last updated on 03/02/2026

Windows security features licensing and edition requirements

Article • 12/02/2024 • Applies to: ✅ Windows 11

This article lists the security features that are available in Windows.

Select one of the two tabs to learn about licensing requirements to use the security features, or to learn about the Windows edition requirements that support them:

Licensing requirements

Expand table

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Access Control (ACL/SACL)	Yes	Yes	Yes	Yes	Yes
Account Lockout Policy	Yes	Yes	Yes	Yes	Yes
Active Directory domain join, Microsoft Entra join, and Microsoft Entra hybrid join with single sign-on (SSO)	Yes	Yes	Yes	Yes	Yes
Always On VPN (device tunnel)	❌	Yes	Yes	Yes	Yes
App containers	Yes	Yes	Yes	Yes	Yes
AppLocker	❌	Yes	Yes	Yes	Yes
Assigned Access (kiosk mode)	Yes	Yes	Yes	Yes	Yes
Attack surface reduction (ASR)	Yes	Yes	Yes	Yes	Yes
Azure Code Signing	Yes	Yes	Yes	Yes	Yes

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
BitLocker enablement	Yes	Yes	Yes	Yes	Yes
BitLocker management	✗	Yes	Yes	Yes	Yes
Bluetooth pairing and connection protection	Yes	Yes	Yes	Yes	Yes
Common Criteria certifications	Yes	Yes	Yes	Yes	Yes
Controlled folder access	Yes	Yes	Yes	Yes	Yes
Credential Guard	✗	Yes	Yes	Yes	Yes
Device health attestation service	Yes	Yes	Yes	Yes	Yes
Direct Access	✗	Yes	Yes	Yes	Yes
Domain Name System (DNS) security	Yes	Yes	Yes	Yes	Yes
Email Encryption (S/MIME)	Yes	Yes	Yes	Yes	Yes
Encrypted hard drive	Yes	Yes	Yes	Yes	Yes
Enhanced phishing protection with SmartScreen	Yes	Yes	Yes	Yes	Yes
Exploit protection	Yes	Yes	Yes	Yes	Yes
Federal Information Processing Standard (FIPS) 140 validation	Yes	Yes	Yes	Yes	Yes
Federated sign-in	✗	✗	✗	Yes	Yes
FIDO2 security key	Yes	Yes	Yes	Yes	Yes
Hardware-enforced stack protection	Yes	Yes	Yes	Yes	Yes

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Hypervisor-protected Code Integrity (HVCI)	Yes	Yes	Yes	Yes	Yes
Kernel Direct Memory Access (DMA) protection	Yes	Yes	Yes	Yes	Yes
Local Security Authority (LSA) Protection	Yes	Yes	Yes	Yes	Yes
Measured boot	Yes	Yes	Yes	Yes	Yes
Microsoft Defender Antivirus	Yes	Yes	Yes	Yes	Yes
Microsoft Defender Application Guard (MDAG) configure via MDM	✗	Yes	Yes	Yes	Yes
Microsoft Defender Application Guard (MDAG) for Edge enterprise mode and enterprise management	✗	Yes	Yes	Yes	Yes
Microsoft Defender Application Guard (MDAG) for Edge standalone mode	Yes	Yes	Yes	Yes	Yes
Microsoft Defender Application Guard (MDAG) for Microsoft Office	✗	✗	✗	✗	✗
Microsoft Defender Application Guard (MDAG) public APIs	✗	Yes	Yes	Yes	Yes
Microsoft Defender for Endpoint	✗	✗	Yes	✗	Yes

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Microsoft Defender SmartScreen	Yes	Yes	Yes	Yes	Yes
Microsoft Pluton	Yes	Yes	Yes	Yes	Yes
Microsoft Security Development Lifecycle (SDL)	Yes	Yes	Yes	Yes	Yes
Microsoft vulnerable driver blocklist	Yes	Yes	Yes	Yes	Yes
Microsoft Windows Insider Preview bounty program	Yes	Yes	Yes	Yes	Yes
Modern device management through (MDM)	Yes	Yes	Yes	Yes	Yes
OneFuzz service	Yes	Yes	Yes	Yes	Yes
Opportunistic Wireless Encryption (OWE)	Yes	Yes	Yes	Yes	Yes
Passkeys	Yes	Yes	Yes	Yes	Yes
Personal Data Encryption (PDE)	❌	Yes	Yes	Yes	Yes
Privacy Resource Usage	Yes	Yes	Yes	Yes	Yes
Privacy Transparency and Controls	Yes	Yes	Yes	Yes	Yes
Remote Credential Guard	Yes	Yes	Yes	Yes	Yes
Remote wipe	Yes	Yes	Yes	Yes	Yes
Secure Boot and Trusted Boot	Yes	Yes	Yes	Yes	Yes
Secured-core configuration lock	Yes	Yes	Yes	Yes	Yes

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Secured-core PC firmware protection	Yes	Yes	Yes	Yes	Yes
Security baselines	Yes	Yes	Yes	Yes	Yes
Server Message Block (SMB) file service	Yes	Yes	Yes	Yes	Yes
Server Message Block Direct (SMB Direct)	Yes	Yes	Yes	Yes	Yes
Smart App Control	Yes	Yes	Yes	Yes	Yes
Smart Cards for Windows Service	Yes	Yes	Yes	Yes	Yes
Software Bill of Materials (SBOM)	Yes	Yes	Yes	Yes	Yes
Tamper protection settings for MDE	Yes	Yes	Yes	Yes	Yes
Transport Layer Security (TLS)	Yes	Yes	Yes	Yes	Yes
Trusted Platform Module (TPM)	Yes	Yes	Yes	Yes	Yes
Universal Print	✗	Yes	Yes	Yes	Yes
User Account Control (UAC)	Yes	Yes	Yes	Yes	Yes
Virtual private network (VPN)	Yes	Yes	Yes	Yes	Yes
Virtualization-based security (VBS)	Yes	Yes	Yes	Yes	Yes
Web sign-in	Yes	Yes	Yes	Yes	Yes
WiFi Security	Yes	Yes	Yes	Yes	Yes
Windows application software	Yes	Yes	Yes	Yes	Yes

Feature name	Windows Pro/Pro Education	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
development kit (SDK)					
Windows Autopatch	✗	Yes	Yes	✗	✗
Windows Autopilot	Yes	Yes	Yes	Yes	Yes
Windows Defender Application Control (WDAC)	Yes	Yes	Yes	Yes	Yes
Windows Defender System Guard	Yes	Yes	Yes	Yes	Yes
Windows Firewall	Yes	Yes	Yes	Yes	Yes
Windows Hello for Business	Yes	Yes	Yes	Yes	Yes
Windows Hello for Business Enhanced Security Sign-in (ESS)	Yes	Yes	Yes	Yes	Yes
Windows Hotpatch	✗	Yes	Yes	✗	✗
Windows LAPS	Yes	Yes	Yes	Yes	Yes
Windows passwordless experience	Yes	Yes	Yes	Yes	Yes
Windows presence sensing	Yes	Yes	Yes	Yes	Yes
Windows Sandbox	Yes	Yes	Yes	Yes	Yes
Windows security policy settings and auditing	Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows Commercial Licensing overview.

Feedback

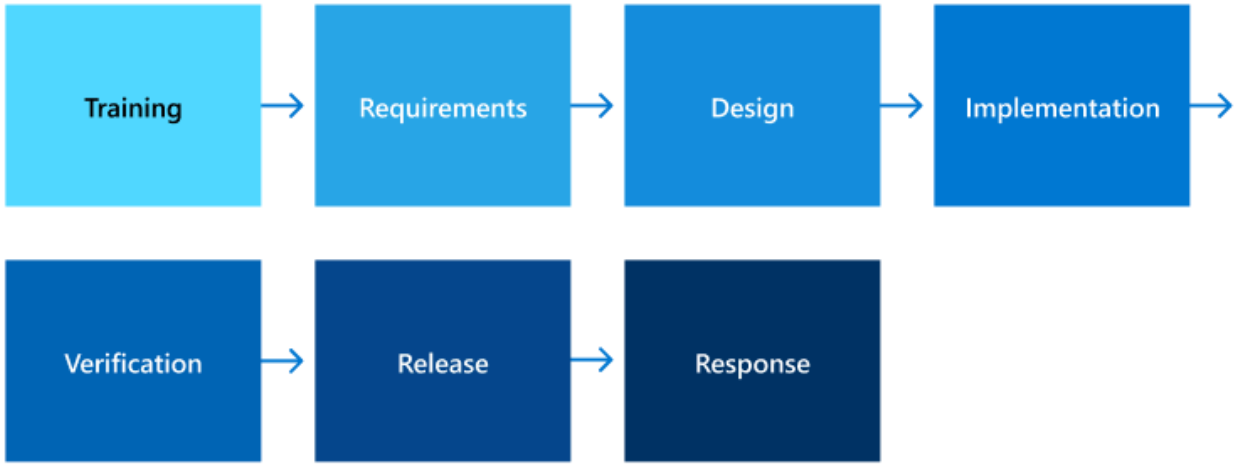
Was this page helpful? Yes No

Provide product feedback

Microsoft Security Development Lifecycle (SDL)

09/29/2025

Security and privacy should never be an afterthought when developing secure software. A formal process must be in place to ensure the development team considers security and privacy at all points of the product's lifecycle. Microsoft's Security Development Lifecycle (SDL) embeds comprehensive security requirements, technology-specific tooling, and mandatory processes into the development and operation of all software products. All development teams at Microsoft must adhere to the SDL processes and requirements. This adherence results in more secure software with fewer and less severe vulnerabilities at a reduced development cost.



Microsoft SDL consists of seven components, including five core phases and two supporting security activities. The five core phases are requirements, design, implementation, verification, and release. Each of these phases contains mandatory checks and approvals to ensure all security and privacy requirements and best practices are properly addressed. The two supporting security activities, training and response, are conducted before and after the core phases respectively to ensure they're properly implemented, and software remains secure after deployment.

Training

All Microsoft employees are required to complete general security and privacy awareness training as well as specific training related to their role. New employees receive initial training upon hire, and annual refresher training is required throughout their employment at Microsoft.

Developers and engineers must also participate in role-specific training to keep them informed on security basics and recent trends in secure development. All full-time employees, interns,

contingent staff, subcontractors, and third parties are also encouraged and provided with the opportunity to seek advanced security and privacy training.

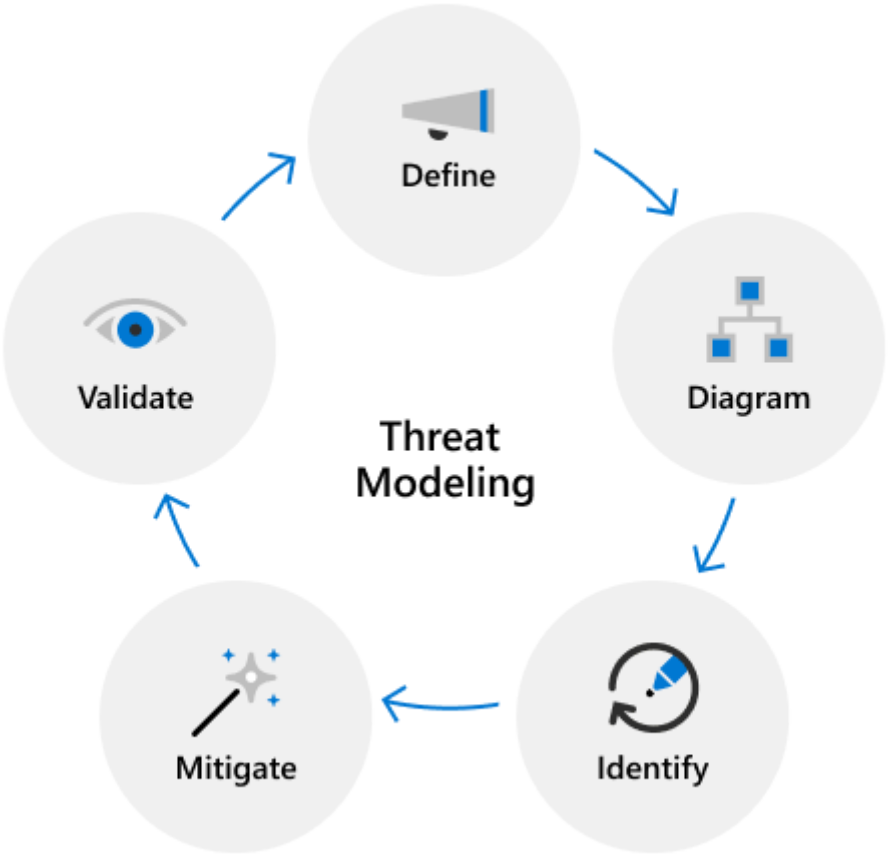
Requirements

Every product, service, and feature Microsoft develops starts with clearly defined security and privacy requirements. These requirements form the foundation of secure applications and inform their design. Development teams define these requirements based on factors such as the type of data the product handles, known threats, best practices, regulations and industry requirements, and lessons learned from previous incidents. Once defined, the development team clearly documents and tracks the requirements.

Software development is a continuous process, meaning that the associated security and privacy requirements change throughout the product's lifecycle to reflect changes in functionality and the threat landscape.

Design

Once you define the security, privacy, and functional requirements, you can start designing the software. As part of the design process, create threat models to help identify, categorize, and rate potential threats according to risk. Maintain and update threat models throughout the lifecycle of each product as you make changes to the software.



The threat modeling process begins by defining the different components of a product and how they interact with each other in key functional scenarios, such as authentication. Create Data Flow Diagrams (DFDs) to visually represent key data flow interactions, data types, ports, and protocols used. Use DFDs to identify and prioritize threats for mitigation that you add to the product's security requirements.

Service teams use Microsoft's Threat Modeling Tool to create threat models, which enable the team to:

- Communicate about the security design of their systems
- Analyze security designs for potential security issues by using a proven methodology
- Suggest and manage mitigation for security issues

Before releasing any product, review all threat models for accuracy and completeness, including mitigation for unacceptable risks.

Implementation

Implementation begins with developers writing code according to the plan they created in the previous two phases. Microsoft provides developers with a suite of secure development tools to effectively implement all the security, privacy, and function requirements of the software

they design. These tools include compilers, secure development environments, and built-in security checks.

Verification

Before releasing any written code, you need to complete several checks and approvals to verify that the code conforms to SDL, meets design requirements, and is free of coding errors. A manual review is conducted by a reviewer who isn't the engineer that developed the code. Separation of duties is an important control in this step to minimize the risk of code being written and released that leads to accidental or malicious harm.

You also need to run various automated checks that are built into the pipeline to analyze code during check-in and when builds are compiled. The security checks used at Microsoft fall into the following categories:

- **Static code analysis**: Analyzes source code for potential security flaws, including the presence of credentials in code.
- **Binary analysis**: Assesses vulnerabilities at the binary code level to confirm code is production ready.
- **Credential and secret scanner**: Identifies possible instances of credential and secret exposure in source code and configuration files.
- **Encryption scanning**: Validates encryption best practices in source code and code execution.
- **Fuzz testing**: Uses malformed and unexpected data to exercise APIs and parsers to check for vulnerabilities and validate error handling.
- **Configuration validation**: Analyzes the configuration of production systems against security standards and best practices.
- **Component Governance (CG)**: Open-source software detection and checking of version, vulnerability, and legal obligations.

If the manual reviewer or automated tools find any issues with the code, they notify the submitter. The submitter must make the necessary changes before submitting the code for review again.

Additionally, both internal and external providers regularly conduct penetration tests on Microsoft online services. Penetration tests provide another means for discovering security flaws that other methods didn't detect. For more information about penetration testing at Microsoft, see Attack simulation in Microsoft 365.

Release

After passing all required security tests and reviews, builds aren't immediately released to all customers. The release process systematically and gradually releases builds to larger and larger groups, referred to as rings, in what is called a safe deployment process (SDP). The SDP rings can generally be defined as:

- **Ring 0**: The development team responsible for the service or feature
- **Ring 1**: All Microsoft employees
- **Ring 2**: Users outside of Microsoft who have configured their organization or specific users to be on the targeted release channel
- **Ring 3**: Worldwide standard release in sub-phases

Builds stay in each of these rings for an appropriate number of days with high load periods, except for Ring 3 since the build is appropriately tested for stability in the earlier rings.

Response

After release, all Microsoft services are extensively logged and monitored. A centralized proprietary near-real-time monitoring system identifies potential security incidents. For more information about security monitoring and security incident management at Microsoft, see Security monitoring overview and Microsoft security incident management.

Windows FIPS 140 validation

Article • 11/13/2024

The Federal Information Processing Standard (FIPS) Publication 140 is a U.S. government standard that defines the minimum-security requirements for cryptographic modules in IT products. This topic introduces FIPS 140 validation for the Windows cryptographic modules. The Windows cryptographic modules are used across different Microsoft products, including Windows client operating systems, Windows Server operating systems, and Azure cloud services.

Microsoft maintains an active commitment to meeting the requirements of the FIPS 140 standard, having validated cryptographic modules against it since it was first established in 2001. Windows cryptographic modules are validated under the Cryptographic Module Validation Program (CMVP), a joint effort between the U.S. National Institute of Standards and Technology (NIST) and the Canadian Centre for Cyber Security (CCCS). The CMVP validates cryptographic modules against the Security Requirements for Cryptographic Modules (part of FIPS 140) and related FIPS cryptography standards. The NIST Information Technology Laboratory operates related programs that Microsoft also participates in: the Cryptographic Algorithm Validation Program (CAVP) certifies FIPS-approved cryptographic algorithms and the Entropy Validation program certifies entropy sources to the NIST SP 800-90B standard.

Windows client operating systems and cryptographic modules

The Windows client releases listed below include cryptographic modules that have completed FIPS 140 validation. Click on the release for details, including the CMVP certificate, Security Policy document, and algorithm scope for each module. When the CMVP certificate validation label includes the note *When operated in FIPS mode*, specific configuration and security rules outlined in the Security Policy must be followed.

Windows 11 releases

- Windows 11, version 21H2

Windows 10 releases

- Windows 10, version 21H1 (May 2021 Update)
- Windows 10, version 20H2 (October 2020 Update)

- Windows 10, version 2004 (May 2020 Update)
- Windows 10, version 1909 (November 2019 Update)
- Windows 10, version 1903 (May 2019 Update)
- Windows 10, version 1809 (October 2018 Update)
- Windows 10, version 1803 (April 2018 Update)
- Windows 10, version 1709 (Fall Creators Update)
- Windows 10, version 1703 (Creators Update)
- Windows 10, version 1607 (Anniversary Update)
- Windows 10, version 1511 (November Update)
- Windows 10, version 1507

Previous Windows releases

- Windows 8.1
- Windows 8
- Windows 7
- Windows Vista SP1
- Windows Vista
- Windows XP SP3
- Windows XP SP2
- Windows XP SP1
- Windows XP
- Windows 2000 SP3
- Windows 2000 SP2
- Windows 2000 SP1
- Windows 2000
- Windows 95 and Windows 98
- Windows NT 4.0

Related products

- Windows Embedded Compact 7 and Windows Embedded Compact 8
- Windows CE 6.0 and Windows Embedded Compact 7
- Outlook Cryptographic Provider

Windows Server operating systems and cryptographic modules

The Windows Server releases listed below include cryptographic modules that have completed FIPS 140 validation. Click on the release for details, including the CMVP

certificate, Security Policy document, and algorithm scope for each module. When the CMVP certificate validation label includes the note *When operated in FIPS mode*, specific configuration and security rules outlined in the Security Policy must be followed.

Windows Server 2022, 2019, and 2016 releases

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

Windows Server semi-annual releases

- Windows Server, version 20H2
- Windows Server, version 2004
- Windows Server, version 1909
- Windows Server, version 1903
- Windows Server, version 1809
- Windows Server, version 1803
- Windows Server, version 1709

Previous Windows Server releases

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2
- Windows Server 2008
- Windows Server 2003 SP2
- Windows Server 2003 SP1
- Windows Server 2003

Use Windows in a FIPS approved mode of operation

To use Windows and Windows Server in a FIPS 140 approved mode of operation, all of the specific configuration and security rules outlined in the module Security Policy documents must be followed. To view or download the Security Policy documents for a given product release, navigate to the listing of FIPS 140 validated modules for the release in the sections above and select the links to the Security Policy documents.

As part of the configuration rules outlined in the Security Policy documents, Windows and Windows Server may be configured to run in a FIPS 140 approved mode of operation, commonly referred to as "FIPS mode." In current versions of Windows, when you enable the FIPS mode setting, the Cryptographic Primitives Library (bcryptprimitives.dll) and Kernel Mode Cryptographic Primitives Library (CNG.sys) modules will run self-tests before Windows runs cryptographic operations. These self-tests meet FIPS 140 requirements and ensure that the modules are functioning properly. The Cryptographic Primitives Library and the Kernel Mode Cryptographic Primitives Library are the only modules that use the FIPS mode configuration setting. FIPS mode does not control which cryptographic algorithms are used. The FIPS mode setting is intended for use only by the Cryptographic Primitives Library (bcryptprimitives.dll) and Kernel Mode Cryptographic Primitives Library (CNG.sys) components in Windows.

Determine if a Windows service or application is FIPS 140 compliant

Microsoft validates the cryptographic modules used in Windows and other products, not individual Windows services or applications. Contact the vendor of the service or application for information on whether it calls a validated Windows cryptographic module (i.e., a module validated by the CMVP as meeting the FIPS 140 requirements and issued a certificate) in a FIPS compliant manner (i.e., by calling for FIPS 140 validated cryptography and configured according to a defined FIPS-approved mode of operation).

FIPS 140 and the Commercial National Security Algorithm Suite

The *Commercial National Security Algorithm* (CNSA) suite is a set of cryptographic algorithms promulgated by the National Security Agency as a replacement for NSA Suite B cryptographic algorithms. Many CNSA cryptographic algorithms are also approved under the FIPS 140 standard. To determine whether a CNSA algorithm was included in the scope of CAVP validated algorithms used in a Microsoft product, navigate to the listing of FIPS 140 validated modules for the product in the sections above and reference the algorithm scope listed for each validated module. Further algorithm details are available in each module Security Policy document.

FIPS 140 and Common Criteria certifications

FIPS 140 and *Common Criteria* are two complementary but different security standards. Whereas FIPS 140 validates cryptographic functionality, Common Criteria evaluates a

broader selection of security functions in IT products. Common Criteria evaluations may rely on FIPS 140 validations to provide assurance that basic cryptographic functionality is implemented properly. For information about Microsoft's Common Criteria certification program, see Common Criteria certifications.

Contact

Contact fips@microsoft.com with questions or to provide feedback on this topic.

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules for Windows 11

Article • 11/12/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows 11, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows 11, version 21H2

Build: 10.0.22000. Validated Edition: Windows 11

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Boot Manager	#4546	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Cryptographic Primitives Library	#4825	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES
Kernel Mode Cryptographic Primitives Library	#4766	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules for Windows 10

Article • 02/24/2025

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows 10, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows 10, version 21H1 (May 2021 Update)

Build: 10.0.19043. Validated Editions: Pro, Enterprise

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Cryptographic Primitives Library	#4825	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES
Kernel Mode Cryptographic Primitives Library	#4766	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES

Windows 10, version 20H2 (October 2020 Update)

Build: 10.0.19042. Validated Editions: Pro, Enterprise

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Cryptographic Primitives Library	#4825	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES
Kernel Mode Cryptographic Primitives Library	#4766	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES

Windows 10, version 2004 (May 2020 Update)

Build: 10.0.19041. Validated Editions: Home, Pro, Enterprise, Education

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
TCB Launcher *Applies only to Enterprise Edition.*	#4457	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Windows Resume	#4348	FIPS Approved: AES, HMAC, KBKDF, RSA, and SHS

Windows 10, version 1909 (November 2019 Update)

Build: 10.0.18363. Validated Editions: Home, Pro, Enterprise, Education

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
TCB Launcher *Applies only to Enterprise Edition.*	#4457	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows Resume	#4348	FIPS Approved: AES, HMAC, KBKDF, RSA, and SHS

Windows 10, version 1903 (May 2019 Update)

Build: 10.0.18362. Validated Editions: Home, Pro, Enterprise, Education

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows Resume	#4348	FIPS Approved: AES, HMAC, KBKDF, RSA, and SHS

Windows 10, version 1809 (October 2018 Update)

Build: 10.0.17763. Validated Editions: Home, Pro, Enterprise, Education

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3644	FIPS Approved: RSA and SHS
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3651	FIPS Approved: RSA and SHS
Virtual TPM	#3690	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#3615	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows 10, version 1803 (April 2018 Update)

Build: 10.0.17134. Validated Editions: Home, Pro, Enterprise, Education

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3195	FIPS Approved: AES, RSA, and SHS

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3096	FIPS Approved: AES, RSA, and SHS
Windows OS Loader	#3480	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows 10, version 1709 (Fall Creators Update)

Build: 10.0.16299. Validated Editions: Home, Pro, Enterprise, Education, S, Surface Hub, Mobile

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3195	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
		Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3096	FIPS Approved: AES, RSA, and SHS
Windows Resume	#3091	FIPS Approved: AES, RSA, and SHS
Windows OS Loader	#3194	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG

Windows 10, version 1703 (Creators Update)

Build: 10.0.15063. Validated Editions: Home, Pro, Enterprise, Education, S, Surface Hub, Mobile

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter *Applies only to Pro, Enterprise, Education, S, Mobile, and Surface Hub Editions.*	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity (ci.dll)	#3093	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	#3095	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library (cng.sys)	#3094	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity (skci.dll) *Applies only to Pro, Enterprise, Education, and S Editions.*	#3096	FIPS Approved: AES, RSA, and SHS

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Windows OS Loader	#3090	FIPS Approved: AES, RSA, and SHS
Windows Resume *Applies only to Home, Pro, Enterprise, Education, and S Editions.*	#3091	FIPS Approved: AES, RSA, and SHS

Windows 10, version 1607 (Anniversary Update)

Build: 10.0.14393.1770. Validated Editions: Windows 10 (Home/Consumer), Pro, Enterprise, Enterprise LTSB, Mobile

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Windows OS Loader (winload)	#3502	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG
BitLocker Windows Resume (winresume) *Applies only to Home, Pro, Enterprise, and Enterprise LTSB Editions.*	#3501	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3487	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS
Code Integrity (ci.dll)	#3510	FIPS Approved: AES, RSA, and SHS
Secure Kernel Code Integrity (skci.dll) *Applies only to Pro, Enterprise, and Enterprise LTSB Editions.*	#3513	FIPS Approved: RSA and SHS; Other Allowed: MD5

Build: 10.0.14393. Validated Editions: Windows 10 (Home/Consumer), Pro, Enterprise, Enterprise LTSB, Mobile

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter (dumpfve.sys)	#2934	FIPS Approved: AES

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Applies only to Pro, Enterprise, Enterprise LTSB, and Mobile Editions.		
BitLocker Windows OS Loader (winload)	#2932	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG
BitLocker Windows Resume (winresume) *Applies only to Home, Pro, Enterprise, and Enterprise LTSB Editions.*	#2933	FIPS Approved: AES, RSA, and SHS; Other Allowed: MD5
Boot Manager	#2931	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS; Other Allowed: MD5, Non-Compliant PBKDF, and VMK KDF
Code Integrity (ci.dll)	#2935	FIPS Approved: RSA and SHS
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	#2937	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5 and MD5
Kernel Mode Cryptographic Primitives Library (cng.sys)	#2936	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity (skci.dll) *Applies only to Pro, Enterprise, and Enterprise LTSB Editions.*	#2938	FIPS Approved: RSA and SHS; Other Allowed: MD5

Windows 10, version 1511 (November Update)

Build: 10.0.10586.1176. Validated Editions: Home, Pro, Enterprise, Enterprise LTSB, Mobile, Surface Hub

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Windows OS Loader (winload)	#3451	FIPS Approved: AES, RSA, and

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
		SHS
BitLocker Windows Resume (winresume) *Applies only to Home, Pro, and Enterprise Editions.*	#3464	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3447	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS
Code Integrity (ci.dll)	#3469	FIPS Approved: AES, RSA, and SHS

Build: 10.0.10586. Validated Editions: Home, Pro, Enterprise, Enterprise LTSB, Mobile, Surface Hub

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter (dumpfve.sys) *Applies only to Pro, Enterprise, Mobile, and Surface Hub Editions.*	#2703	FIPS Approved: AES
BitLocker Windows OS Loader (winload) *Applies only to Home, Pro, Enterprise, Mobile, and Surface Hub Editions.*	#2701	FIPS Approved: AES, RSA, and SHS; Other Allowed: MD5 and NDRNG
BitLocker Windows Resume (winresume) *Applies only to Home, Pro, and Enterprise Editions.*	#2702	FIPS Approved: AES, RSA, and SHS; Other Allowed: MD5
Boot Manager *Applies only to Home, Pro, Enterprise, Mobile, and Surface Hub Editions.*	#2700	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS; Other Allowed: MD5, Non-Compliant KDF, and Non-Compliant PBKDF
Code Integrity (ci.dll)	#2604	FIPS Approved: RSA and SHS; Other Allowed: Non-Compliant AES and MD5
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	#2606	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS,

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
		and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library (cng.sys)	#2605	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity (skci.dll) *Applies only to Enterprise and Enterprise LTSB Editions.*	#2607	FIPS Approved: RSA and SHS

Windows 10, version 1507

Build: 10.0.10240.17643. Validated Editions: Enterprise LTSB

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Windows OS Loader (winload)	#3427	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG
BitLocker Windows Resume (winresume)	#3426	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3415	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS
Code Integrity (ci.dll)	#3437	FIPS Approved: AES, RSA, and SHS

Build: 10.0.10240. Validated Editions: Home, Pro, Enterprise, Enterprise LTSB, Mobile, and Surface Hub

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter (dumpfve.sys)	#2603	FIPS Approved: AES

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Applies only to Pro, Enterprise, and Enterprise LTSB Editions.		
BitLocker Windows OS Loader (winload) *Applies only to Home, Pro, Enterprise, and Enterprise LTSB Editions.*	#2601	FIPS Approved: AES, RSA, and SHS; Other Allowed: MD5 and NDRNG
BitLocker Windows Resume (winresume) *Applies only to Home, Pro, Enterprise, and Enterprise LTSB Editions.*	#2602	FIPS Approved: AES, RSA, and SHS; Other Allowed: MD5
Boot Manager *Applies only to Home, Pro, Enterprise, and Enterprise LTSB Editions.*	#2600	FIPS Approved: AES, HMAC, KTS, PBKDF, RSA, and SHS; Other Allowed: MD5, Non-Compliant KDF, and Non-Compliant PBKDF
Code Integrity (ci.dll)	#2604	FIPS Approved: RSA and SHS; Other Allowed: Non-Compliant AES and MD5
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	#2606	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library (cng.sys)	#2605	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity (skci.dll) *Applies only to Enterprise and Enterprise LTSB Editions.*	#2607	FIPS Approved: RSA and SHS

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in previous Windows versions

Article • 02/01/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in versions of Windows prior to Windows 10, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows 8.1

Validated Editions: RT, Pro, Enterprise, Phone, Embedded

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Dump Filter (dumpfve.sys)	6.3.9600 6.3.9600.17031	#2354
BitLocker Windows OS Loader (winload)	6.3.9600 6.3.9600.17031	#2352
BitLocker Windows Resume (winresume) *Applies only to Pro, Enterprise, and Embedded 8.*	6.3.9600 6.3.9600.17031	#2353
Boot Manager	6.3.9600 6.3.9600.17031	#2351
Code Integrity (ci.dll)	6.3.9600 6.3.9600.17031	#2355
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	6.3.9600 6.3.9600.17031	#2357
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.3.9600 6.3.9600.17042	#2356

Windows 8

Validated Editions: RT, Home, Pro, Enterprise, Phone

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Dump Filter (DUMPFVE.SYS)	6.2.9200	#1899
BitLocker Windows OS Loader (WINLOAD)	6.2.9200	#1896
BitLocker Windows Resume (WINRESUME) *Applies only to Home and Pro*	6.2.9200	#1898
Boot Manager	6.2.9200	#1895
Code Integrity (CI.DLL)	6.2.9200	#1897
Enhanced Cryptographic Provider (RSAENH.DLL)	6.2.9200	#1894
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH.DLL)	6.2.9200	#1893
Cryptographic Primitives Library (BCRYPTPRIMITIVES.DLL)	6.2.9200	#1892
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.2.9200	#1891

Windows 7

Validated Editions: Windows 7, Windows 7 SP1

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Drive Encryption	6.1.7600.16385, 6.1.7600.16429, 6.1.7600.16757, 6.1.7600.20536, 6.1.7600.20873, 6.1.7600.20897, 6.1.7600.20916, 6.1.7601.17514, 6.1.7601.17556, 6.1.7601.21634, 6.1.7601.21655, and 6.1.7601.21675	1332
Boot Manager	6.1.7600.16385 and 6.1.7601.17514	1319
Code Integrity (CI.DLL)	6.1.7600.16385, 6.1.7600.17122, 6.1.7600.21320, 6.1.7601.17514, 6.1.7601.17950, and	1327

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
	6.1.7601.22108	
Cryptographic Primitives Library (BCRYPTPRIMITIVES.DLL)	6.1.7600.16385 and 6.1.7601.17514	1329
Enhanced Cryptographic Provider (RSAENH.DLL)	6.1.7600.16385 (no change in SP1)	1330
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH.DLL)	6.1.7600.16385 (no change in SP1)	1331
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.1.7600.16385, 6.1.7600.16915, 6.1.7600.21092, 6.1.7601.17514, 6.1.7601.17725, 6.1.7601.17919, 6.1.7601.21861, 6.1.7601.22076	1328
Winload OS Loader (winload.exe)	6.1.7600.16385, 6.1.7600.16757, 6.1.7600.20897, 6.1.7600.20916, 6.1.7601.17514, 6.1.7601.17556, 6.1.7601.21655, and 6.1.7601.21675	1326

Windows Vista SP1

Validated Edition: Ultimate Edition

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Boot Manager (bootmgr)	6.0.6001.18000 and 6.0.6002.18005	978
Cryptographic Primitives Library (bcrypt.dll)	6.0.6001.22202, 6.0.6002.18005, and 6.0.6002.22872	1001
Code Integrity (ci.dll)	6.0.6001.18000, 6.0.6001.18023, 6.0.6001.22120, and 6.0.6002.18005	980
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	6.0.6001.18000 and 6.0.6002.18005	1003

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	6.0.6001.22202 and 6.0.6002.18005	1002
Kernel Mode Security Support Provider Interface (ksecdd.sys)	6.0.6001.18709, 6.0.6001.18272, 6.0.6001.18796, 6.0.6001.22202, 6.0.6001.22450, 6.0.6001.22987, 6.0.6001.23069, 6.0.6002.18005, 6.0.6002.18051, 6.0.6002.18541, 6.0.6002.18643, 6.0.6002.22152, 6.0.6002.22742, and 6.0.6002.22869	1000
Winload OS Loader (winload.exe)	6.0.6001.18000, 6.0.6001.18027, 6.0.6001.18606, 6.0.6001.22125, 6.0.6001.22861, 6.0.6002.18005, 6.0.6002.18411 and 6.0.6002.22596	979

Windows Vista

Validated Edition: Ultimate Edition

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Drive Encryption	6.0.6000.16386	947
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	6.0.6000.16386	894
Enhanced Cryptographic Provider (RSAENH)	6.0.6000.16386	893
Kernel Mode Security Support Provider Interface (ksecdd.sys)	6.0.6000.16386, 6.0.6000.16870 and 6.0.6000.21067	891

Windows XP SP3

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	5.1.2600.5507	990

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	5.1.2600.5507	989
Kernel Mode Cryptographic Module (FIPS.SYS)	5.1.2600.5512	997

Windows XP SP2

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
DSS/Diffie-Hellman Enhanced Cryptographic Provider	5.1.2600.2133	240
Microsoft Enhanced Cryptographic Provider	5.1.2600.2161	238

Windows XP SP1

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Microsoft Enhanced Cryptographic Provider	5.1.2600.1029	238

Windows XP

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Kernel Mode Cryptographic Module	5.1.2600.0	241

Windows 2000 SP3

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base DSS Cryptographic Provider, Base Cryptographic Provider, DSS/Diffie-Hellman Enhanced Cryptographic Provider, and Enhanced Cryptographic Provider	Base DSS: 5.0.2195.3665 (SP3), Base: 5.0.2195.3839 (SP3), DSS/DH Enh: 5.0.2195.3665 (SP3), Enh: 5.0.2195.3839 (SP3)	103
Kernel Mode Cryptographic Module (FIPS.SYS)	5.0.2195.1569	106

Windows 2000 SP2

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base DSS Cryptographic Provider, Base Cryptographic Provider, DSS/Diffie-Hellman Enhanced Cryptographic Provider, and Enhanced Cryptographic Provider	Base DSS 5.0.2195.2228 (SP2), Base 5.0.2195.2228 (SP2), DSS/DH Enh 5.0.2195.2228 (SP2), Enh 5.0.2195.2228 (SP2)	103
Kernel Mode Cryptographic Module (FIPS.SYS)	5.0.2195.1569	106

Windows 2000 SP1

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base DSS Cryptographic Provider, Base Cryptographic Provider, DSS/Diffie-Hellman Enhanced Cryptographic Provider, and Enhanced Cryptographic Provider	Base DSS 5.0.2150.1391 (SP1), Base 5.0.2150.1391 (SP1), DSS/DH Enh: 5.0.2150.1391 (SP1), Enh 5.0.2150.1391 (SP1)	103

Windows 2000

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base DSS Cryptographic Provider, Base Cryptographic Provider, DSS/Diffie-Hellman Enhanced Cryptographic Provider, and Enhanced Cryptographic Provider	5.0.2150.1	76

Windows 95 and Windows 98

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base DSS Cryptographic Provider, Base Cryptographic Provider, DSS/Diffie-Hellman Enhanced Cryptographic Provider, and Enhanced Cryptographic Provider	5.0.1877.6 and 5.0.1877.7	75

Windows NT 4.0

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Base Cryptographic Provider	5.0.1877.6 and 5.0.1877.7	68

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in Windows Server 2022

Article • 11/13/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows Server 2022, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, see its linked Security Policy document or module certificate.

Windows Server 2022

Build: 10.0.20348. Validated Editions: Standard, Datacenter, and Datacenter: Azure.

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Cryptographic Primitives Library	#4825	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES
Kernel Mode Cryptographic Primitives Library	#4766	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in Windows Server 2019

Article • 04/05/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows Server 2019, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, see its linked Security Policy document or module certificate.

Windows Server 2019

Builds: 10.0.17763.10021 and 10.0.17763.10127. Validated Edition: Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4688	FIPS Approved: AES, RSA, and SHS
Boot Manager	#4484	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4602	FIPS Approved: RSA and SHS
Cryptographic Primitives Library	#4687	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: ECDH and NDRNG
Kernel Mode Cryptographic Primitives Library	#4670	AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: ECDH
Secure Kernel Code Integrity	#4640	FIPS Approved: RSA and SHS
Virtual TPM	#4686	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Windows OS Loader	#4545	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG

Build: 10.0.17763.107. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	Version	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	10.0.17763	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	10.0.17763	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	10.0.17763	#3644	FIPS Approved: RSA and SHS
Cryptographic Primitives Library	10.0.17763	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	10.0.17763	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	10.0.17763	#3651	FIPS Approved: RSA and SHS
Virtual TPM	10.0.17763	#3690	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	10.0.17763	#3615	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in Windows Server 2016

Article • 02/01/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows Server 2016, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows Server 2016

Build: 10.0.14393.1770. Validated Editions: Standard, Datacenter, Storage Server.

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Windows OS Loader (winload)	#3502	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG
BitLocker Windows Resume (winresume)	#3501	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3487	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS
Code Integrity (ci.dll)	#3510	FIPS Approved: AES, RSA, and SHS
Secure Kernel Code Integrity (skci.dll)	#3513	FIPS Approved: RSA and SHS; Other Allowed: MD5

Build: 10.0.14393. Validated Editions: Standard, Datacenter, Storage Server.

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter (dumpfve.sys)	#2934	FIPS Approved: AES
BitLocker Windows OS Loader (winload)	#2932	FIPS Approved: AES, RSA, and SHS; Other: NDRNG
BitLocker Windows Resume (winresume)	#2933	FIPS Approved: AES, RSA, and SHS; Other: MD5
Boot Manager	#2931	FIPS Approved: AES, HMAC, PBKDF, RSA, and SHS; Other: MD5, Non-Compliant PBKDF, and VMK KDF
Code Integrity (ci.dll)	#2935	FIPS Approved: RSA and SHS
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	#2937	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other: HMAC-MD5 and MD5.
Kernel Mode Cryptographic Primitives Library (cng.sys)	#2936	FIPS Approved: AES, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity (skci.dll)	#2938	FIPS Approved: RSA and SHS; Other: MD5

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in Windows Server semi-annual releases

Article • 11/13/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in Windows Server semi-annual releases, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows Server, version 20H2

Build: 10.0.19042. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Cryptographic Primitives Library	#4825	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES
Kernel Mode Cryptographic Primitives Library	#4766	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, ENT (P), HMAC, KAS, KAS-SSC, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES

Windows Server, version 2004

Build: 10.0.19041. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows Server, version 1909

Build: 10.0.18363. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows Server, version 1903

Build: 10.0.18362. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#4538	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3923	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#4511	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#4536	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Kernel Mode Cryptographic Primitives Library	#4515	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: NDRNG
Secure Kernel Code Integrity	#4512	FIPS Approved: AES, RSA, and SHS
Virtual TPM	#4537	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
Windows OS Loader	#4339	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows Server, version 1809

Build: 10.0.17763. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3644	FIPS Approved: RSA and SHS
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3651	FIPS Approved: RSA and SHS
Virtual TPM	#3690	FIPS Approved: AES, CKG, CVL, DRBG, ECDSA, HMAC, KAS, KBKDF, KTS, RSA, and SHS; Other Allowed: NDRNG
Windows OS Loader	#3615	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows Server, version 1803

Build: 10.0.17134. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3195	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3096	FIPS Approved: AES, RSA, and SHS
Windows OS Loader	#3480	FIPS Approved: AES, CKG, DRBG, RSA, and SHS; Other Allowed: NDRNG

Windows Server, version 1709

Build: 10.0.16299. Validated Editions: Standard Core, Datacenter Core

Expand table

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
BitLocker Dump Filter	#3092	FIPS Approved: AES, RSA, and SHS
Boot Manager	#3089	FIPS Approved: AES, CKG, HMAC, PBKDF, RSA, and SHS
Code Integrity	#3195	FIPS Approved: AES, RSA, and SHS
Cryptographic Primitives Library	#3197	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and

Cryptographic Module (linked to Security Policy document)	CMVP Certificate #	Validated Algorithms
		Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Kernel Mode Cryptographic Primitives Library	#3196	FIPS Approved: AES, CKG, CVL, DRBG, DSA, ECDSA, HMAC, KAS, KBKDF, KTS, PBKDF, RSA, SHS, and Triple-DES; Other Allowed: HMAC-MD5, MD5, and NDRNG
Secure Kernel Code Integrity	#3096	FIPS Approved: AES, RSA, and SHS
Windows OS Loader	#3194	FIPS Approved: AES, RSA, and SHS; Other Allowed: NDRNG

Feedback

Was this page helpful? Yes No

Provide product feedback

FIPS 140 validated modules in previous Windows Server versions

Article • 02/01/2024

The following tables list the completed FIPS 140 validations of cryptographic modules used in versions of Windows Server prior to Windows Server 2016, organized by major release of the operating system. The linked Security Policy document for each module provides details on the module capabilities and the policies the operator must follow to use the module in its FIPS approved mode of operation. For information on using the overall operating system in its FIPS approved mode, see Use Windows in a FIPS approved mode of operation. For details on the FIPS approved algorithms used by each module, including CAVP algorithm certificates, see the module's linked Security Policy document or CMVP module certificate.

Windows Server 2012 R2

Validated Editions: Server, Storage Server, StorSimple 8000 Series, Azure StorSimple Virtual Array Windows Server 2012 R2

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Dump Filter (dumpfve.sys) *Doesn't apply to Azure StorSimple Virtual Array Windows Server 2012 R2*	6.3.9600 6.3.9600.17031	2354
BitLocker Windows OS Loader (winload)	6.3.9600 6.3.9600.17031	2352
BitLocker Windows Resume (winresume) *Doesn't apply to Azure StorSimple Virtual Array Windows Server 2012 R2*	6.3.9600 6.3.9600.17031	2353
Boot Manager	6.3.9600 6.3.9600.17031	2351
Code Integrity (ci.dll)	6.3.9600 6.3.9600.17031	2355
Cryptographic Primitives Library (bcryptprimitives.dll and ncryptsslp.dll)	6.3.9600 6.3.9600.17031	2357
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.3.9600 6.3.9600.17042	2356

Windows Server 2012

Validated Editions: Server, Storage Server

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Dump Filter (DUMPFVE.SYS)	6.2.9200	1899
BitLocker Windows OS Loader (WINLOAD)	6.2.9200	1896
BitLocker Windows Resume (WINRESUME)	6.2.9200	1898
Boot Manager	6.2.9200	1895
Code Integrity (CI.DLL)	6.2.9200	1897
Enhanced Cryptographic Provider (RSAENH.DLL)	6.2.9200	1894
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH.DLL)	6.2.9200	1893
Cryptographic Primitives Library (BCRYPTPRIMITIVES.DLL)	6.2.9200	[1892]
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.2.9200	1891

Windows Server 2008 R2

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
BitLocker Drive Encryption	6.1.7600.16385, 6.1.7600.16429, 6.1.7600.16757, 6.1.7600.20536, 6.1.7600.20873, 6.1.7600.20897, 6.1.7600.20916, 6.1.7601.17514, 6.1.7601.17556, 6.1.7601.21634, 6.1.7601.21655 or 6.1.7601.21675	1339
Boot Manager (bootmgr)	6.1.7600.16385 or 6.1.7601.17514	1321
Cryptographic Primitives Library (bcryptprimitives.dll)	66.1.7600.16385 or 6.1.7601.17514	1336

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	6.1.7600.16385	1337
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	6.1.7600.16385	1338
Kernel Mode Cryptographic Primitives Library (cng.sys)	6.1.7600.16385, 6.1.7600.16915, 6.1.7600.21092, 6.1.7601.17514, 6.1.7601.17919, 6.1.7601.17725, 6.1.7601.21861 and 6.1.7601.22076	1335
Winload OS Loader (winload.exe)	6.1.7600.16385, 6.1.7600.16757, 6.1.7600.20897, 6.1.7600.20916, 6.1.7601.17514, 6.1.7601.17556, 6.1.7601.21655 and 6.1.7601.21675	1333

Windows Server 2008

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Boot Manager (bootmgr)	6.0.6001.18000, 6.0.6002.18005 and 6.0.6002.22497	1004
Code Integrity (ci.dll)	6.0.6001.18000 and 6.0.6002.18005	1006
Cryptographic Primitives Library (bcrypt.dll)	6.0.6001.22202, 6.0.6002.18005 and 6.0.6002.22872	1008
Enhanced Cryptographic Provider (RSAENH)	6.0.6001.22202 and 6.0.6002.18005	1010
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	6.0.6001.18000 and 6.0.6002.18005	1009
Kernel Mode Security Support Provider Interface (ksecdd.sys)	6.0.6001.18709, 6.0.6001.18272, 6.0.6001.18796, 6.0.6001.22202, 6.0.6001.22450, 6.0.6001.22987, 6.0.6001.23069, 6.0.6002.18005, 6.0.6002.18051, 6.0.6002.18541, 6.0.6002.18643, 6.0.6002.22152, 6.0.6002.22742 and 6.0.6002.22869	1007

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Winload OS Loader (winload.exe)	6.0.6001.18000, 6.0.6001.18606, 6.0.6001.22861, 6.0.6002.18005, 6.0.6002.18411, 6.0.6002.22497 and 6.0.6002.22596	1005

Windows Server 2003 SP2

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	5.2.3790.3959	868
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	5.2.3790.3959	875
Kernel Mode Cryptographic Module (FIPS.SYS)	5.2.3790.3959	869

Windows Server 2003 SP1

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	5.2.3790.1830 [Service Pack 1])	382
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	5.2.3790.1830 [Service Pack 1]	381
Kernel Mode Cryptographic Module (FIPS.SYS)	5.2.3790.1830 [SP1]	405

Windows Server 2003

Expand table

Cryptographic Module	Version (link to Security Policy)	CMVP Certificate #
Enhanced Cryptographic Provider (RSAENH)	5.2.3790.0	382
Enhanced DSS and Diffie-Hellman Cryptographic Provider (DSSENH)	5.2.3790.0	381
Kernel Mode Cryptographic Module (FIPS.SYS)	5.2.3790.0	405

Feedback

Was this page helpful? Yes No

Provide product feedback

Common Criteria certifications

Article • 02/01/2024

Microsoft is committed to optimizing the security of its products and services. As part of that commitment, Microsoft supports the *Common Criteria for Information Technology Security Evaluation* program, ensures that products incorporate the features and functions required by relevant Common Criteria *Protection Profiles*, and completes Common Criteria certifications of Microsoft Windows products. This topic lists the Windows products certified against the Common Criteria (current and archived), together with documentation from each certification.

Windows client operating systems

The Windows client releases listed below have been certified against one or more Protection Profiles, as listed on the Common Criteria Portal. Click on a release for its certification details, including links to certification documents. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* (where available) providing details on the evaluator's actions.

Windows 11 releases

- Windows 11, version 22H2
- Windows 11, version 21H2

Windows 10 releases

- Windows 10, version 22H2 (2022 Update)
- Windows 10, version 21H2 (November 2021 Update)
- Windows 10, version 21H1 (May 2021 Update)
- Windows 10, version 20H2 (October 2020 Update)
- Windows 10, version 2004 (May 2020 Update)
- Windows 10, version 1909 (November 2019 Update)
- Windows 10, version 1903 (May 2019 Update)
- Windows 10, version 1803 (April 2018 Update)
- Windows 10, version 1709 (Fall Creators Update)

- Windows 10, version 1703 (Creators Update)
- Windows 10, version 1607 (Anniversary Update)
- Windows 10, version 1511 (November 2015 Update)
- Windows 10, version 1507 (initial release)

Previous Windows releases

- Windows 8.1
- Windows 8
- Windows 7
- Windows Vista

Windows Server operating systems

The Windows Server releases listed below have been certified against one or more Protection Profiles, as listed on the Common Criteria Portal. Click on a release for its certification details, including links to certification documents. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* (where available) providing details on the evaluator's actions.

Windows Server 2022, 2019, and 2016 releases

- Windows Server 2022
- Windows Server 2019
- Windows Server 2016

Windows Server semi-annual releases

- Windows Server, version 20H2 (October 2020 Update)
- Windows Server, version 2004 (May 2020 Update)
- Windows Server, version 1909 (November 2019 Update)
- Windows Server, version 1903 (May 2019 Update)
- Windows Server, version 1809 (October 2018 Update)
- Windows Server, version 1803 (April 2018 Update)
- Windows Server, version 1709 (Fall Creators Update)

Previous Windows Server releases

- Windows Server 2012 R2
- Windows Server 2012
- Windows Server 2008 R2
- Windows Server 2008
- Windows Server 2003 Certificate Server
- Windows Rights Management Services

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows 11 Common Criteria certifications

The following tables list the completed Windows 11 Common Criteria certifications and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* providing details on the evaluator's actions.

Windows 11, versions 24H2 and 23H2

Expand table

Product details	Date	Scope	Documents
Version 24H2 - build: 10.0.26100.1, validated editions: Enterprise, Pro, Education and IoT Enterprise. Version 23H2 - build: 10.0.22631.2428, validated edition: Enterprise.	January 15, 2026	Certified against the Protection Profile for General Purpose Operating Systems (4.3), the PP-Module for VPN Client (2.4), the PP-Module for Wireless Local Area Network Client (1.0), the PP-Module for Bluetooth (1.0) and the Functional Package for Transport Layer Security (TLS) Version 2.0.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 11, version 22H2

Expand table

Product details	Date	Scope	Documents
Build: 10.0.22621.1. Validated editions: Enterprise, Pro, Education, IoT Enterprise.	January 17, 2024	Certified against the Protection Profile for General Purpose Operating Systems (4.2.1), the PP-Module for VPN Client (2.4), the PP-Module for Wireless Local Area Network Client (1.0) and the PP-Module for Bluetooth (1.0).	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 11, version 21H2

Expand table

Product details	Date	Scope	Documents
Build: 10.0.22000.1. Validated edition: Enterprise.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Last updated on 01/23/2026

Windows 10 Common Criteria certifications

Article • 02/01/2024

The following tables list the completed Windows 10 Common Criteria certifications and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* providing details on the evaluator's actions.

Windows 10, version 22H2 (2022 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19045.2006. Validated editions: Pro, Enterprise.	January 17, 2024	Certified against the Protection Profile for General Purpose Operating Systems, the PP-Module for VPN Client, the PP-Module for Wireless Local Area Network Client and the PP-Module for Bluetooth.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 21H2 (November 2021 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19044.1288. Validated editions: Pro, Enterprise.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 21H1 (May 2021 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19043.1052. Validated editions: Pro, Enterprise.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 20H2 (October 2020 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19042.1052. Validated editions: Pro, Enterprise.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 2004 (May 2020 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19041. Validated editions: Home, Pro, Enterprise.	December 31, 2021	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 1909 (November 2019 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.18363. Validated edition: Enterprise.	February 11, 2021	(Hyper-V certification.) Certified against the Protection Profile for Virtualization, including the Extended Package for Server Virtualization.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 10.0.18363. Validated editions: Home, Pro, Enterprise.	June 5, 2020	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 1903 (May 2019 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.18362. Validated editions: Home, Pro, Enterprise.	October 26, 2019	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 1809 (October 2018 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.17763. Validated editions: Home, Pro, Enterprise.	September 7, 2019	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for	Security Target; Administrative Guide; Assurance Activity

Product details	Date	Scope	Documents
		Wireless Local Area Network Clients.	Report ; Certification Report

Windows 10, version 1803 (April 2018 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.17134. Validated editions: Home, Pro, Enterprise.	February 6, 2019	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Windows 10, version 1709 (Fall Creators Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.16299. Validated editions: Home, Pro, Enterprise, S.	April 20, 2018	Certified against the Protection Profile for General Purpose Operating Systems.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Windows 10, version 1703 (Creators Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.15063. Validated editions: Home, Pro, Enterprise, S.	February 21, 2018	Certified against the Protection Profile for General Purpose Operating Systems.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Windows 10, version 1607 (Anniversary Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.14393. Validated editions: Pro, Enterprise, Mobile.	April 12, 2017	(Mobile certification.) Certified against the Protection Profile for Mobile Device Fundamentals.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Validated editions: Home, Pro, Enterprise.	February 6, 2017	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Validated editions: Home, Pro, Enterprise.	December 29, 2016	(VPN certification.) Certified against the Protection Profile for IPsec Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 1511 (November 2015 Update)

Expand table

Product details	Date	Scope	Documents
Validated editions: Pro, Enterprise.	June 23, 2016	(Mobile certification.) Certified against the Protection Profile for Mobile Device Fundamentals.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows 10, version 1507 (initial version released July 2015)

Expand table

Product details	Date	Scope	Documents
Validated edition: Enterprise.	November 20, 2017	(Hyper-V certification.) Certified against the Protection Profile for Server Virtualization.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Validated edition: Pro, Enterprise.	November 10, 2016	(VPN certification.) Certified against the Protection Profile	Security Target; Administrative Guide;

Product details	Date	Scope	Documents
		for IPsec Virtual Private Network Clients.	Assurance Activity Report; Certification Report
Validated editions: Pro, Enterprise.	May 12, 2016	(Mobile certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 10.0.10240. Validated editions: Home, Pro, Enterprise.	April 6, 2016	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Feedback

Was this page helpful? Yes No

Provide product feedback

Common Criteria certifications for previous Windows releases

Article • 12/23/2024

The following tables list the completed Common Criteria certifications for Windows releases before Windows 10 and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation.

Windows 8.1

Expand table

Product details	Date	Scope	Documents
Validated editions: Pro (on Microsoft Surface 3); Windows Phone 8.1 (GDR2 on Microsoft Lumia 635 and 830).	August 27, 2015	Certified against the Protection Profile for Mobile Device Fundamentals	Security Target; Administrative Guide; Certification Report
Validated editions: Pro (on Microsoft Surface Pro 3).	April 21, 2015	Certified against the Protection Profile for Mobile Device Fundamentals	Security Target; Administrative Guide; Certification Report
Validated editions: Pro (on Microsoft Surface Pro 2 and Dell Venue 8 Pro); Enterprise (on Lenovo X1 Carbon and HP Pro x2 410 G1); Windows Phone 8.1 (on Microsoft Lumia 520).	March 16, 2015	Certified against the Protection Profile for Mobile Device Fundamentals	Security Target; Administrative Guide; Certification Report

Windows 8

Expand table

Product details	Date	Scope	Documents
Validated editions: Pro, Enterprise.	January 9, 2015	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Certification Report
Validated editions: Windows 8, RT.	January 9, 2015	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Certification Report
Validated editions: Pro, Enterprise.	April 7, 2014	(Disk encryption certification.) Certified against the Protection Profile for Full Disk Encryption.	Security Target; Administrative Guide; Certification Report
Validated editions: Windows 8, Pro, Enterprise, RT.	January 31, 2014	(VPN certification.) Certified against the Protection Profile for IPsec Virtual Private Network Clients.	Security Target; Administrative Guide; Certification Report

Windows 7

Expand table

Product details	Date	Scope	Documents
Validated editions: Enterprise, Ultimate.	March 24, 2011	Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Certification Report

Windows Vista

Expand table

Product details	Date	Scope	Documents
Validated edition: Enterprise.	August 15, 2009	EAL 4. Controlled Access Protection Profile. CC Part 2: security functional requirements. CC Part 3: security assurance requirements.	Security Target; Certification Report
Validated edition: Enterprise.	September 17, 2008	EAL 1. CC Part 2: security functional requirements. CC Part 3: security assurance requirements.	Security Target; Certification Report

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Server 2025, 2022, 2019, and 2016 Common Criteria certifications

The following tables list the completed Common Criteria certifications for Windows Server 2025, 2022, 2019, and 2016 releases and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* providing details on the evaluator's actions.

Windows Server 2025

Expand table

Product details	Date	Scope	Documents
Build: 10.0.26100.1. Validated editions: Standard, Datacenter.	January 15, 2026	Certified against the Protection Profile for General Purpose Operating Systems (4.3), the PP-Module for VPN Client (2.4), the PP-Module for Wireless Local Area Network Client (1.0), the PP-Module for Bluetooth (1.0) and the Functional Package for Transport Layer Security (TLS) Version 2.0.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server 2022

Expand table

Product details	Date	Scope	Documents
Build: 10.0.20348.587. Validated editions: Standard, Datacenter.	January 17, 2024	Certified against the Protection Profile for General Purpose Operating Systems (4.2.1), the PP-Module for VPN Client (2.4), the PP-Module for Wireless Local Area Network Client (1.0) and the PP-Module for Bluetooth (1.0).	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 10.0.20348.1. Validated editions: Standard, Datacenter.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and	Security Target; Administrative Guide; Assurance Activity

Product details	Date	Scope	Documents
		the Module for Virtual Private Network Clients.	Report ; Certification Report

Windows Server 2019

Expand table

Product details	Date	Scope	Documents
Build: 10.0.17763. Validated editions: Standard, Datacenter.	February 11, 2021	(Hyper-V certification.) Certified against the Protection Profile for Virtualization, including the Extended Package for Server Virtualization.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report
Build: 10.0.17763. Validated editions: Standard, Datacenter.	September 7, 2019	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Windows Server 2016

Expand table

Product details	Date	Scope	Documents
Validated editions: Standard, Datacenter.	November 20, 2017	(Hyper-V certification.) Certified against the Protection Profile for Server Virtualization.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report
Build: 10.0.14393. Validated editions: Standard, Datacenter.	February 6, 2017	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report
Validated editions: Standard, Datacenter.	December 29, 2016	(VPN certification.) Certified against the Protection Profile for IPsec Virtual Private Network Clients.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Last updated on 01/23/2026

Windows Server semi-annual Common Criteria certifications

Article • 02/01/2024

The following tables list the completed Common Criteria certifications for Windows Server semi-annual releases and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* providing details on the evaluator's actions.

Windows Server, version 20H2 (October 2020 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19042.1052. Validated editions: Standard, Datacenter.	January 26, 2023	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 2004 (May 2020 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.19041. Validated editions: Standard, Datacenter.	December 31, 2021	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 1909 (November 2019 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.18363. Validated editions: Standard, Datacenter.	February 11, 2021	(Hyper-V certification.) Certified against the Protection Profile for Virtualization, including the Extended Package for Server Virtualization.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 10.0.18363. Validated editions: Standard, Datacenter.	June 5, 2020	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients and the Module for Virtual Private Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 1903 (May 2019 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.18362. Validated editions: Standard, Datacenter.	October 26, 2019	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 1809 (October 2018 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.17763. Validated editions: Standard, Datacenter.	February 11, 2021	(Hyper-V certification.) Certified against the Protection Profile for Virtualization, including the Extended Package for Server Virtualization.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 10.0.17763. Validated editions: Standard, Datacenter.	September 7, 2019	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 1803 (April 2018 Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.17134. Validated editions: Standard Core, Datacenter Core.	February 6, 2019	Certified against the Protection Profile for General Purpose Operating Systems, including the Extended Package for Wireless Local Area Network Clients.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server, version 1709 (Fall Creators Update)

Expand table

Product details	Date	Scope	Documents
Build: 10.0.16299. Validated editions: Standard Core, Datacenter Core.	April 20, 2018	Certified against the Protection Profile for General Purpose Operating Systems.	Security Target ; Administrative Guide ; Assurance Activity Report ; Certification Report

Feedback

Was this page helpful? Yes No

Provide product feedback

Common Criteria certifications for previous Windows Server releases

Article • 02/24/2025

The following tables list the completed Common Criteria certifications for Windows Server releases before Windows Server 2016 and provide links to certification documents, organized by major release of the operating system. The *Security Target* describes the product editions in scope, the security functionality in the product, and the assurance measures from the Protection Profile used as part of the evaluation. The *Administrative Guide* provides guidance on configuring the product to match the evaluated configuration. The *Certification Report* or *Validation Report* documents the results of the evaluation, with the *Assurance Activity Report* (when available) providing details on the evaluator's actions.

Windows Server 2012 R2

Expand table

Product details	Date	Scope	Documents
Validated editions: Standard, Datacenter.	November 20, 2017	(Hyper-V certification.) Certified against the Protection Profile for Server Virtualization.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report
Build: 6.3.9600. Validated editions: Standard, Datacenter.	April 6, 2016	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Assurance Activity Report; Certification Report

Windows Server 2012

Expand table

Product details	Date	Scope	Documents
Validated editions: Standard, Datacenter.	January 9, 2015	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Administrative Guide; Certification Report

Product details	Date	Scope	Documents
Validated editions: Standard, Datacenter.	April 7, 2014	(Disk encryption certification.) Certified against the Protection Profile for Full Disk Encryption.	Security Target; Administrative Guide; Certification Report
Validated editions: Standard, Datacenter.	January 31, 2014	(VPN certification.) Certified against the Protection Profile for IPsec Virtual Private Network Clients.	Security Target; Administrative Guide; Certification Report

Windows Server 2008 R2

Expand table

Product details	Date	Scope	Documents
Validated editions: Standard, Enterprise, Datacenter, Itanium.	March 24, 2011	(OS certification.) Certified against the Protection Profile for General Purpose Operating Systems.	Security Target; Certification Report
Server Core 2008 R2: Hyper-V Server Role	July 24, 2009	(Hyper-V certification.) Common Criteria for Information Technology Security Evaluation Version 3.1 Revision 3. It is CC Part 2 extended and Part 3 conformant, with a claimed Evaluation Assurance Level of EAL4, augmented by ALC_FLR.3.	Security Target; Administrative Guide; Certification Report

Windows Server 2008

Expand table

Product details	Date	Scope	Documents
Validated edition: Standard, Enterprise, Datacenter.	August 15, 2009	Controlled Access Protection Profile. CC Part 2: security functional requirements. CC Part 3: security assurance requirements, at EAL 4.	Security Target; Certification Report

Product details	Date	Scope	Documents
Microsoft Windows Server Core 2008: Hyper-V Server Role.	July 24, 2009	CC Part 2: security functional requirements. CC Part 3: security assurance requirements, at EAL 4.	Security Target; Administrative Guide ; Certification Report
Validated edition: Standard, Enterprise, Datacenter.	September 17, 2008	CC Part 2: security functional requirements. CC Part 3: security assurance requirements, at EAL 1.	Security Target; Certification Report

Windows Server 2003 Certificate Server

Expand table

Product details	Date	Scope	Documents
Microsoft Certificate Server, as part of Windows Server 2003 SP1, Enterprise Edition	April 1, 2007	CC Part 2: security functional requirements. CC Part 3: security assurance requirements at EAL 4, augmented with ALC_FLR.3 and AVA_VLA.4.	Security Target ; Certification Report

Windows Rights Management Services

Expand table

Product details	Date	Scope	Documents
Microsoft Windows Rights Management Services 1.0 with SP2	August 8, 2007	CC Part 2: security functional requirements. CC Part 3: security assurance requirements at EAL 4, augmented with ALC_FLR.3.	Security Target ; Certification Report

Feedback

Was this page helpful? Yes No

Provide product feedback

System Guard: How a hardware-based root of trust helps protect Windows

Applies to: ✅ Windows 11, ✅ Windows 10

To protect critical resources such as the Windows authentication stack, single sign-on tokens, the Windows Hello biometric stack, and the Virtual Trusted Platform Module, a system's firmware and hardware must be trustworthy.

System Guard reorganizes the existing Windows system integrity features under one roof and sets up the next set of investments in Windows security. It's designed to make these security guarantees:

- Protect and maintain the integrity of the system as it starts up
- Validate that system integrity has truly been maintained through local and remote attestation

Maintaining the integrity of the system as it starts

Static Root of Trust for Measurement (SRTM)

With Windows 7, one of the means attackers would use to persist and evade detection was to install what is often referred to as a bootkit or rootkit on the system. This malicious software would start before Windows started, or during the boot process itself, enabling it to start with the highest level of privilege.

With Windows 10 running on modern hardware, a hardware-based root of trust helps ensure that no unauthorized firmware or software (such as a bootkit) can start before the Windows bootloader. This hardware-based root of trust comes from the device's Secure Boot feature, which is part of the Unified Extensible Firmware Interface (UEFI). This technique of measuring the static early boot UEFI components is called the Static Root of Trust for Measurement (SRTM).

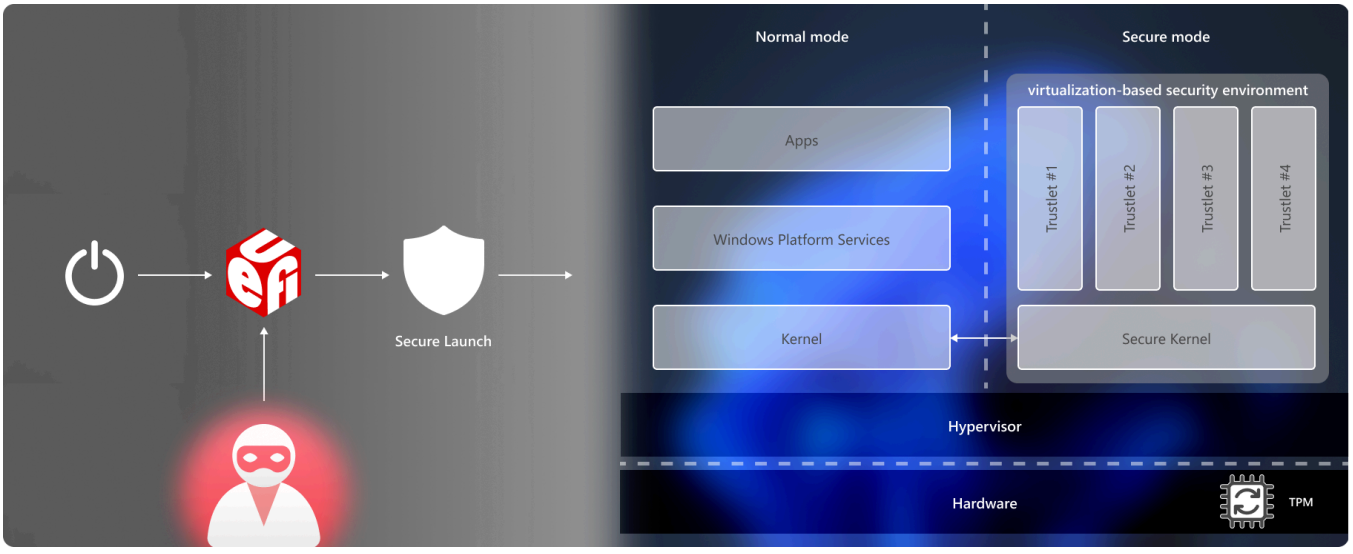
As there are thousands of PC vendors that produce many models with different UEFI BIOS versions, there becomes an incredibly large number of SRTM measurements upon bootup. Two techniques exist to establish trust here—either maintain a list of known 'bad' SRTM measurements (also known as a blocklist), or a list of known 'good' SRTM measurements (also known as an allowlist).

Each option has a drawback:

- A list of known 'bad' SRTM measurements allows a hacker to change just 1 bit in a component to create an entirely new SRTM hash that needs to be listed. This means that the SRTM flow is inherently brittle - a minor change can invalidate the entire chain of trust.
- A list of known 'good' SRTM measurements requires each new BIOS/PC combination measurement to be carefully added, which is slow. Also, a bug fix for UEFI code can take a long time to design, build, retest, validate, and redeploy.

Secure Launch—the Dynamic Root of Trust for Measurement (DRTM)

System Guard Secure Launch, first introduced in Windows 10 version 1809, aims to alleviate these issues by using a technology known as the Dynamic Root of Trust for Measurement (DRTM). DRTM lets the system freely boot into untrusted code initially, but shortly after launches the system into a trusted state by taking control of all CPUs and forcing them down a well-known and measured code path. This has the benefit of allowing untrusted early UEFI code to boot the system, but then being able to securely transition into a trusted and measured state.



Secure Launch simplifies management of SRTM measurements because the launch code is now unrelated to a specific hardware configuration. This means the number of valid code measurements is small, and future updates can be deployed more widely and quickly.

System Management Mode (SMM) protection

System Management Mode (SMM) is a special-purpose CPU mode in x86 microcontrollers that handles power management, hardware configuration, thermal monitoring, and anything else the manufacturer deems useful. Whenever one of these system operations is requested, a nonmaskable interrupt (SMI) is invoked at runtime, which executes SMM code installed by the

BIOS. SMM code executes in the highest privilege level and is invisible to the OS, which makes it an attractive target for malicious activity. Even if System Guard Secure Launch is used to late launch, SMM code can potentially access hypervisor memory and change the hypervisor.

To defend against this, two techniques are used:

- Paging protection to prevent inappropriate access to code and data
- SMM hardware supervision and attestation

Paging protection can be implemented to lock certain code tables to be read-only to prevent tampering. This prevents access to any memory that hasn't been assigned.

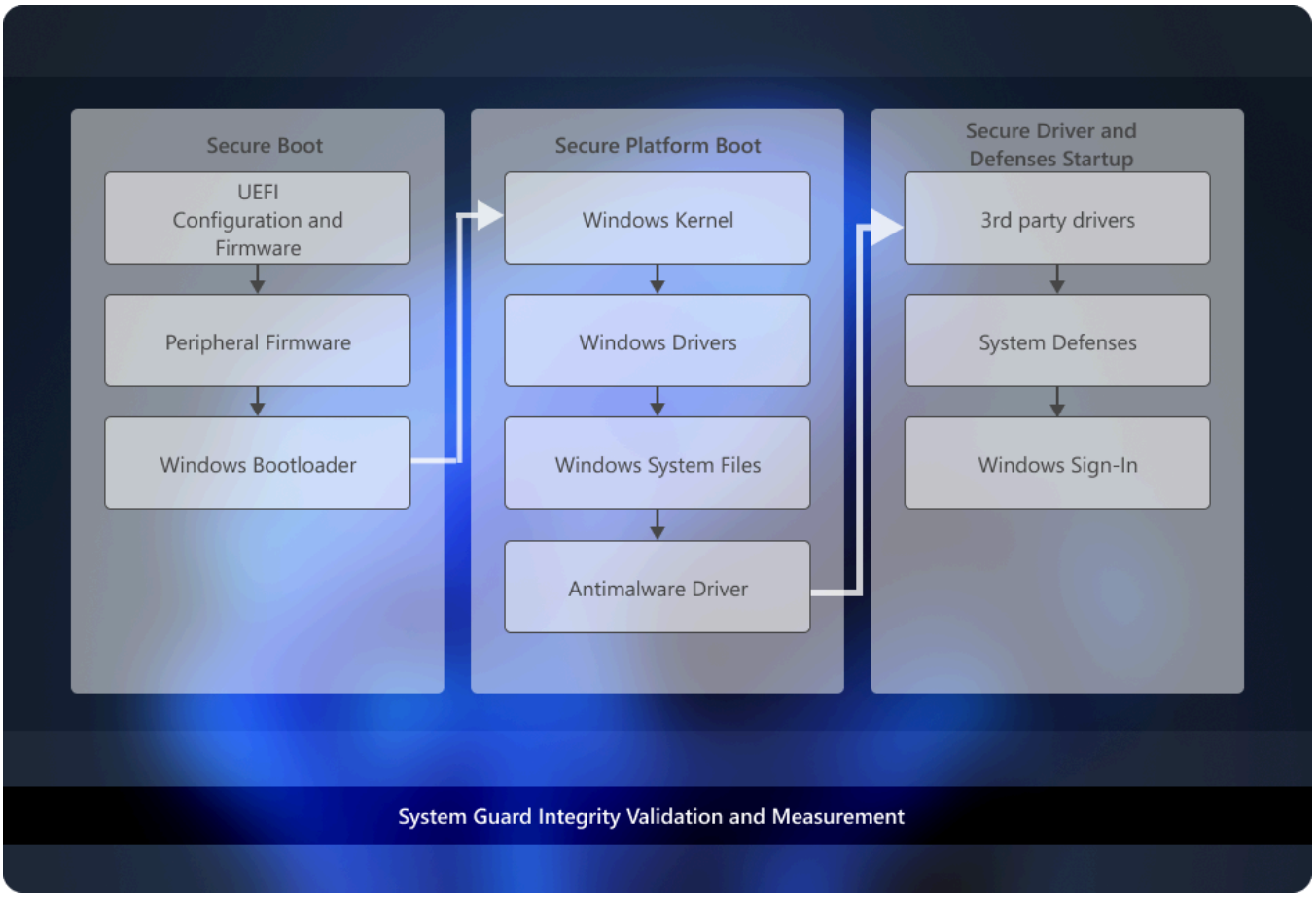
A hardware-enforced processor feature known as a supervisor SMI handler can monitor the SMM and make sure it doesn't access any part of the address space that it isn't supposed to.

SMM protection is built on top of the Secure Launch technology and requires it to function. In the future, Windows 10 will also measure this SMI Handler's behavior and attest that no OS-owned memory has been tampered with.

Validating platform integrity after Windows is running (run time)

While System Guard provides advanced protection that will help protect and maintain the integrity of the platform during boot and at run time, the reality is that we must apply an "assume breach" mentality to even our most sophisticated security technologies. We can trust that the technologies are successfully doing their jobs, but we also need the ability to verify that they were successful in achieving their goals. For platform integrity, we can't just trust the platform, which potentially could be compromised, to self-attest to its security state. So System Guard includes a series of technologies that enable remote analysis of the device's integrity.

As Windows boots, a series of integrity measurements are taken by System Guard using the device's Trusted Platform Module 2.0 (TPM 2.0). System Guard Secure Launch doesn't support earlier TPM versions, such as TPM 1.2. This process and data are hardware-isolated away from Windows to help ensure that the measurement data isn't subject to the type of tampering that could happen if the platform was compromised. From here, the measurements can be used to determine the integrity of the device's firmware, hardware configuration state, and Windows boot-related components, to name a few.



After the system boots, System Guard signs and seals these measurements using the TPM. Upon request, a management system like Intune or Microsoft Configuration Manager can acquire them for remote analysis. If System Guard indicates that the device lacks integrity, the management system can take a series of actions, such as denying the device access to resources.

Windows edition and licensing requirements

The following table lists the Windows editions that support System Guard:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

System Guard license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

System requirements for System Guard

This feature is available for the following processors:

- Intel® vPro™ processors starting with Intel® Coffeelake, Whiskeylake, or later silicon
- AMD® processors starting with Zen2 or later silicon
- Qualcomm® processors with SD850 or later chipsets

Requirements for Intel® vPro™ processors starting with Intel® Coffeelake, Whiskeylake, or later silicon

Expand table

Name	Description
64-bit CPU	A 64-bit computer with minimum four cores (logical processors) is required for hypervisor and Virtualization-based security (VBS). For more information about Hyper-V, see Hyper-V on Windows Server 2016 or Introduction to Hyper-V on Windows 10. For more information about hypervisor, see Hypervisor Specifications.
Trusted Platform Module (TPM) 2.0	Platforms must support a discrete TPM 2.0. Integrated/firmware TPMs aren't supported, except Intel chips that support Platform Trust Technology (PTT), which is a type of integrated hardware TPM that meets the TPM 2.0 spec.
Windows DMA Protection	Platforms must meet the Windows DMA Protection Specification (all external DMA ports must be off by default until the OS explicitly powers them).
SMM communication buffers	All SMM communication buffers must be implemented in EfiRuntimeServicesData, EfiRuntimeServicesCode, EfiACPIMemoryNVS, or EfiReservedMemoryType memory types.
SMM Page Tables	Must NOT contain any mappings to EfiConventionalMemory (for example no OS/VMM owned memory). Must NOT contain any mappings to code sections within EfiRuntimeServicesCode. Must NOT have execute and write permissions for the same page Must allow ONLY that TSEG pages can be marked executable and the memory map must report TSEG EfiReservedMemoryType. BIOS SMI handler must be implemented such that SMM page tables are locked on every SMM entry.

Name	Description
Modern/Connected Standby	Platforms must support Modern/Connected Standby.

TPM AUX Index	Platform must set up a AUX index with index, attributes, and policy that exactly corresponds to the AUX index specified in the TXT DG with a data size of exactly 104 bytes (for SHA256 AUX data). (NameAlg = SHA256)
	Platforms must set up a PS (Platform Supplier) index with:
	• Exactly the "TXT PS2" style Attributes on creation as follows:
	  ◦ AuthWrite
	  ◦ PolicyDelete
	  ◦ WriteLocked
	  ◦ WriteDefine
	  ◦ AuthRead
	  ◦ WriteDefine
	  ◦ NoDa
	  ◦ Written
	  ◦ PlatformCreate
	• A policy of exactly PolicyCommandCode(CC = TPM2_CC_UndefineSpaceSpecial) (SHA256 NameAlg and Policy)
	• Size of exactly 70 bytes
	• NameAlg = SHA256
	• Also, it must have been initialized and locked (TPMA_NV_WRITTEN = 1, TPMA_NV_WRITELOCKED = 1) at time of OS launch.
	PS index data DataRevocationCounters, SINITMinVersion, and PolicyControl must all be 0x00

AUX Policy	The required AUX policy must be as follows: • A = TPM2_PolicyLocality (Locality 3 & Locality 4) • B = TPM2_PolicyCommandCode (TPM_CC_NV_UndefineSpecial) • authPolicy = {A} OR {{A} AND {B}} • authPolicy digest = 0xef, 0x9a, 0x26, 0xfc, 0x22, 0xd1, 0xae, 0x8c, 0xec, 0xff, 0x59, 0xe9, 0x48, 0x1a, 0xc1, 0xec, 0x53, 0x3d, 0xbe, 0x22, 0x8b, 0xec, 0x6d, 0x17, 0x93, 0x0f, 0x4c, 0xb2, 0xcc, 0x5b, 0x97, 0x24

TPM NV Index	Platform firmware must set up a TPM NV index for use by the OS with:
	• Handle: 0x01C101C0
	• Attributes:
	  ◦ TPMA_NV_POLICYWRITE
	  ◦ TPMA_NV_PPREAD
	  ◦ TPMA_NV_OWNERREAD
	  ◦ TPMA_NV_AUTHREAD
	  ◦ TPMA_NV_POLICYREAD
	  ◦ TPMA_NV_NO_DA
	  ◦ TPMA_NV_PLATFORMCREATE
	  ◦ TPMA_NV_POLICY_DELETE
	• A policy of:

Name	Description
	○ A = TPM2_PolicyAuthorize(MSFT_DRTM_AUTH_BLOB_SigningKey) ○ B = TPM2_PolicyCommandCode(TPM_CC_NV_UndefineSpaceSpecial) ○ authPolicy = {A} OR {{A} AND {B}} ○ Digest value of 0xcb, 0x45, 0xc8, 0x1f, 0xf3, 0x4b, 0xcf, 0x0a, 0xfb, 0x9e, 0x1a, 0x80, 0x29, 0xfa, 0x23, 0x1c, 0x87, 0x27, 0x30, 0x3c, 0x09, 0x22, 0xdc, 0xce, 0x68, 0x4b, 0xe3, 0xdb, 0x81, 0x7c, 0x20, 0xe1
Platform firmware	Platform firmware must carry all code required to execute an Intel® Trusted Execution Technology secure launch: • Intel® SINIT ACM must be carried in the OEM BIOS • Platforms must ship with a production ACM signed by the correct production Intel® ACM signer for the platform
Platform firmware update	It's recommended to update System firmware via UpdateCapsule in Windows Update.

Requirements for AMD® processors starting with Zen2 or later silicon

Expand table

Name	Description
64-bit CPU	A 64-bit computer with minimum four cores (logical processors) is required for hypervisor and Virtualization-based security (VBS). For more information about Hyper-V, see Hyper-V on Windows Server 2016 or Introduction to Hyper-V on Windows 10. For more information about hypervisor, see Hypervisor Specifications.
Trusted Platform Module (TPM) 2.0	Platforms must support a discrete TPM 2.0 OR Microsoft Pluton TPM.
Windows DMA Protection	Platforms must meet the Windows DMA Protection Specification (all external DMA ports must be off by default until the OS explicitly powers them).
SMM communication buffers	All SMM communication buffers must be implemented in EfiRuntimeServicesData, EfiRuntimeServicesCode, EfiACPIMemoryNVS, or EfiReservedMemoryType memory types.
SMM Page Tables	Must NOT contain any mappings to EfiConventionalMemory (for example no OS/VMM owned memory). Must NOT contain any mappings to code sections within EfiRuntimeServicesCode. Must NOT have execute and write permissions for the same page BIOS SMI handler must be implemented such that SMM page tables are locked on every SMM entry.

Name	Description
Modern/Connected Standby	Platforms must support Modern/Connected Standby.

TPM NV Index	Platform firmware must set up a TPM NV index for use by the OS with:
	• Handle: 0x01C101C0
	• Attributes:
	  ◦ TPMA_NV_POLICYWRITE
	  ◦ TPMA_NV_PPREAD
	  ◦ TPMA_NV_OWNERREAD
	  ◦ TPMA_NV_AUTHREAD
	  ◦ TPMA_NV_POLICYREAD
	  ◦ TPMA_NV_NO_DA
	  ◦ TPMA_NV_PLATFORMCREATE
	  ◦ TPMA_NV_POLICY_DELETE
	• A policy of:
	  ◦ A = TPM2_PolicyAuthorize(MSFT_DRTM_AUTH_BLOB_SigningKey)
	  ◦ B = TPM2_PolicyCommandCode(TPM_CC_NV_UndefineSpaceSpecial)
	  ◦ authPolicy = {A} OR {{A} AND {B}}
	  ◦ Digest value of 0xcb, 0x45, 0xc8, 0x1f, 0xf3, 0x4b, 0xcf, 0x0a, 0xfb, 0x9e, 0x1a, 0x80, 0x29, 0xfa, 0x23, 0x1c, 0x87, 0x27, 0x30, 0x3c, 0x09, 0x22, 0xdc, 0xce, 0x68, 0x4b, 0xe3, 0xdb, 0x81, 0x7c, 0x20, 0xe1

Platform firmware	Platform firmware must carry all code required to execute Secure Launch: • AMD® Secure Launch platforms must ship with AMD® DRTM driver devnode exposed and the AMD® DRTM driver installed Platform must have AMD® Secure Processor Firmware Anti-Rollback protection enabled Platform must have AMD® Memory Guard enabled.
Platform firmware update	It's recommended to update System firmware via UpdateCapsule in Windows Update.

Requirements for Qualcomm® processors with SD850 or later chipsets

Expand table

Name	Description
Monitor Mode Communication	All Monitor Mode communication buffers must be implemented in either EfiRuntimeServicesData (recommended), data sections of EfiRuntimeServicesCode as described by the Memory Attributes Table, EfiACPIMemoryNVS, or EfiReservedMemoryType memory types

Name	Description
Monitor Mode Page Tables	All Monitor Mode page tables must: • NOT contain any mappings to EfiConventionalMemory (for example no OS/VMM owned memory) • They must NOT have execute and write permissions for the same page • Platforms must only allow Monitor Mode pages marked as executable • The memory map must report Monitor Mode as EfiReservedMemoryType • Platforms must provide mechanism to protect the Monitor Mode page tables from modification
Modern/Connected Standby	Platforms must support Modern/Connected Standby.
Platform firmware	Platform firmware must carry all code required to launch.
Platform firmware update	It's recommended to update System firmware via UpdateCapsule in Windows Update.

Last updated on 08/15/2025

Trusted Platform Module Technology Overview

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the Trusted Platform Module (TPM) and how Windows uses it for access control and authentication.

Feature description

The Trusted Platform Module (TPM) technology is designed to provide hardware-based, security-related functions. A TPM chip is a secure crypto-processor that is designed to carry out cryptographic operations. The chip includes multiple physical security mechanisms to make it tamper-resistant, and malicious software is unable to tamper with the security functions of the TPM. Some of the advantages of using TPM technology are:

- Generate, store, and limit the use of cryptographic keys.
- Use it for device authentication by using the TPM's unique RSA key, which is burned into the chip.
- Help ensure platform integrity by taking and storing security measurements of the boot process.

The most common TPM functions are used for system integrity measurements and for key creation and use. During the boot process of a system, the boot code that is loaded (including firmware and the operating system components) can be measured and recorded in the TPM. The integrity measurements can be used as evidence for how a system started and to make sure that a TPM-based key was used only when the correct software was used to boot the system.

TPM-based keys can be configured in various ways. One option is to make a TPM-based key unavailable outside the TPM. This is good to mitigate phishing attacks because it prevents the key from being copied and used without the TPM. TPM-based keys can also be configured to require an authorization value to use them. If too many incorrect authorization guesses occur, the TPM activates its dictionary attack logic and prevents further authorization value guesses.

Different versions of the TPM are defined in specifications by the Trusted Computing Group (TCG). For more information, see the TCG Web site.

Windows edition and licensing requirements

The following table lists the Windows editions that support Trusted Platform Module (TPM):

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Trusted Platform Module (TPM) license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Automatic initialization of the TPM with Windows

Starting with Windows 10 and Windows 11, the operating system automatically initializes and takes ownership of the TPM. This means that in most cases, we recommend that you avoid configuring the TPM through the TPM management console, **TPM.msc**. There are a few exceptions, mostly related to resetting or performing a clean installation on a PC. For more information, see Clear all the keys from the TPM.

> **Note**
>
> We're no longer actively developing the TPM management console beginning with Windows Server 2019 and Windows 10, version 1809.

In certain specific enterprise scenarios limited to Windows 10, versions 1507 and 1511, Group Policy might be used to back up the TPM owner authorization value in Active Directory. Because the TPM state persists across operating system installations, this TPM information is stored in a location in Active Directory that is separate from computer objects.

Practical applications

Certificates can be installed or created on computers that are using the TPM. After a computer is provisioned, the RSA private key for a certificate is bound to the TPM and can't be exported.

The TPM can also be used as a replacement for smart cards, which reduces the costs associated with creating and disbursing smart cards.

Automated provisioning in the TPM reduces the cost of TPM deployment in an enterprise. New APIs for TPM management can determine if TPM provisioning actions require physical presence of a service technician to approve TPM state change requests during the boot process.

Anti-malware software can use the boot measurements of the operating system start state to prove the integrity of a computer running Windows. These measurements include the launch of Hyper-V to test that datacenters using virtualization aren't running untrusted hypervisors. With BitLocker Network Unlock, IT administrators can push an update without concerns that a computer is waiting for PIN entry.

The TPM has several Group Policy settings that might be useful in certain enterprise scenarios. For more info, see TPM Group Policy Settings.

Device health attestation

Device health attestation enables enterprises to establish trust based on hardware and software components of a managed device. With device heath attestation, you can configure an MDM server to query a health attestation service that allows or denies a managed device access to a secure resource.

Some security issues that you can check on the devices include:

- Is Data Execution Prevention supported and enabled?
- Is BitLocker Drive Encryption supported and enabled?
- Is SecureBoot supported and enabled?

> **Note**
>
> Windows supports Device Health Attestation with TPM 2.0. TPM 2.0 requires UEFI firmware. A device with legacy BIOS and TPM 2.0 won't work as expected.

Supported versions for device health attestation

Expand table

TPM version	Windows 11	Windows 10	Windows Server 2022	Windows Server 2019	Windows Server 2016
TPM 1.2		>= ver 1607		Yes	>= ver 1607
TPM 2.0	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

Last updated on 08/15/2025

TPM fundamentals

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article provides a description of the Trusted Platform Module (TPM 1.2 and TPM 2.0) components, and explains how they're used to mitigate dictionary attacks.

A TPM is a microchip designed to provide basic security-related functions, primarily involving encryption keys. The TPM is installed on the motherboard of a computer, and it communicates with the rest of the system by using a hardware bus.

Devices that incorporate a TPM can create cryptographic keys and encrypt them, so that the keys can only be decrypted by the TPM. This process, often called "wrapping" or "binding" a key, can help protect the key from disclosure. Each TPM has a primary wrapping key, called the **storage root key**, which is stored within the TPM itself. The private portion of a storage root key, or **endorsement key**, that is created in a TPM is never exposed to any other component, software, process, or user.

You can specify whether encryption keys that the TPM creates can be migrated or not. If you specify that they can be migrated, the public and private portions of the key can be exposed to other components, software, processes, or users. If you specify that encryption keys can't be migrated, the private portion of the key is never exposed outside the TPM.

Devices that incorporate a TPM can also create a key wrapped and tied to certain platform measurements. This type of key can be unwrapped only when those platform measurements have the same values that they had when the key was created. This process is referred to as *sealing the key to the TPM*. Decrypting the key is called *unsealing*. The TPM can also seal and unseal data that is generated outside the TPM. With sealed key and software, such as BitLocker Drive Encryption, data can be locked until specific hardware or software conditions are met.

With a TPM, private portions of key pairs are kept separate from the memory that is controlled by the operating system. Keys can be sealed to the TPM, and certain assurances about the state of a system (assurances that define the trustworthiness of a system) can be made before the keys are unsealed and released for use. The TPM uses its own internal firmware and logic circuits to process instructions. Hence, it doesn't rely on the operating system and it isn't exposed to vulnerabilities that might exist in the operating system or application software.

- For information about which versions of Windows support which versions of the TPM, see Trusted Platform Module technology overview.
- For more information about which TPM services can be controlled centrally by using Group Policy settings, see TPM Group Policy Settings.

The features that are available in the versions are defined in specifications by the Trusted Computing Group (TCG). For more information, see the Trusted Platform Module page on the Trusted Computing Group website.

The following sections provide an overview of the technologies that support the TPM:

- Measured Boot with support for attestation
- TPM-based Virtual Smart Card
- TPM-based certificate storage
- TPM Cmdlets
- Physical presence interface
- TPM 1.2 states and initialization
- Endorsement keys
- TPM Key Attestation
- Anti-hammering

Measured Boot with support for attestation

The Measured Boot feature provides anti-malware software with a trusted (resistant to spoofing and tampering) log of all boot components. Anti-malware software can use the log to determine whether components that ran before it are trustworthy or infected with malware. It can also send the Measured Boot logs to a remote server for evaluation. The remote server can start remediation actions by interacting with software on the client or through out-of-band mechanisms, as appropriate.

TPM-based Virtual Smart Card

> ⚠ **Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

The Virtual Smart Card emulates the functionality of traditional smart cards. Virtual Smart Cards use the TPM chip, rather than using a separate physical smart card and reader. This greatly reduces the management and deployment cost of smart cards in an enterprise. To the end user, the Virtual Smart Card is always available on the device. If a user needs to use more than one device, a Virtual Smart Card must be issued to the user for each device. A computer that is shared among multiple users can host multiple Virtual Smart Cards, one for each user.

TPM-based certificate storage

The TPM protects certificates and RSA keys. The TPM key storage provider (KSP) provides easy and convenient use of the TPM as a way of strongly protecting private keys. The TPM KSP generates keys when an organization enrolls for certificates. The TPM also protects certificates that are imported from an outside source. TPM-based certificates are standard certificates. The certificate can never leave the TPM from which the keys are generated. The TPM can also be used for crypto-operations through Cryptography API: Next Generation (CNG).

TPM Cmdlets

You can manage the TPM using Windows PowerShell. For details, see TPM Cmdlets in Windows PowerShell.

Physical presence interface

For TPM 1.2, the TCG specifications for TPMs require physical presence (typically, pressing a key) for turning on the TPM, turning it off, or clearing it. These actions typically can't be automated with scripts or other automation tools unless the individual OEM supplies them.

TPM 1.2 states and initialization

TPM 1.2 has multiple possible states. Windows automatically initializes the TPM, which brings it to an enabled, activated, and owned state.

Endorsement keys

A trusted application can use TPM only if the TPM contains an endorsement key, which is an RSA key pair. The private half of the key pair is held inside the TPM and it's never revealed or accessible outside the TPM.

Key attestation

TPM key attestation allows a certification authority to verify that a private key is protected by a TPM and that the TPM is one that the certification authority trusts. Endorsement keys proven valid are used to bind the user identity to a device. The user certificate with a TPM-attested key provides higher security assurance backed up by nonexportability, anti-hammering, and isolation of keys provided by a TPM.

Anti-hammering

When a TPM processes a command, it does so in a protected environment. For example a dedicated micro controller on a discrete chip, or a special hardware-protected mode on the main CPU. A TPM is used to create a cryptographic key that isn't disclosed outside the TPM. It's used in the TPM after the correct authorization value is provided.

TPMs have anti-hammering protection that is designed to prevent brute force attacks, or more complex dictionary attacks, that attempt to determine authorization values for using a key. The basic approach is for the TPM to allow only a limited number of authorization failures before it prevents more attempts to use keys and locks. Providing a failure count for individual keys isn't technically practical, so TPMs have a global lockout when too many authorization failures occur.

Because many entities can use the TPM, a single authorization success can't reset the TPM's anti-hammering protection. This prevents an attacker from creating a key with a known authorization value and then using it to reset the TPM's protection. TPMs are designed to forget about authorization failures after a period of time so the TPM doesn't enter a lockout state unnecessarily. A TPM owner password can be used to reset the TPM's lockout logic.

TPM 2.0 anti-hammering

TPM 2.0 has well defined anti-hammering behavior. This is in contrast to TPM 1.2 for which the anti-hammering protection was implemented by the manufacturer and the logic varied widely throughout the industry.

For systems with TPM 2.0, the TPM is configured by Windows to lock after 32 authorization failures and to forget one authorization failure every 10 minutes. This means that a user could quickly attempt to use a key with the wrong authorization value 32 times. For each of the 32 attempts, the TPM records if the authorization value was correct or not. This inadvertently causes the TPM to enter a locked state after 32 failed attempts.

Attempts to use a key with an authorization value for the next 10 minutes wouldn't return success or failure. Instead, the response indicates that the TPM is locked. After 10 minutes, one authorization failure is forgotten and the number of authorization failures remembered by the TPM drops to 31. The TPM leaves the locked state and returns to normal operation. With the correct authorization value, keys could be used normally if no authorization failures occur during the next 10 minutes. If a period of 320 minutes elapses with no authorization failures, the TPM doesn't remember any authorization failures, and 32 failed attempts could occur again.

Windows doesn't require TPM 2.0 systems to forget about authorization failures when the system is fully powered off or when the system has hibernated. Windows requires that authorization failures are forgotten when the system is running normally, in a sleep mode, or in low power states other than off. If a Windows system with TPM 2.0 is locked, the TPM leaves lockout mode if the system is left on for 10 minutes.

The anti-hammering protection for TPM 2.0 can be fully reset immediately by sending a reset lockout command to the TPM, and providing the TPM owner password. By default, Windows automatically provisions TPM 2.0 and stores the TPM owner password for use by system administrators.

In some implementations, the TPM owner authorization value is stored centrally in Active Directory, and not on the local system. An administrator can execute `tpm.msc` and choose to reset the TPM lockout time. If the TPM owner password is stored locally, it's used to reset the lockout time. If the TPM owner password isn't available on the local system, the administrator must provide it. If an administrator attempts to reset the TPM lockout state with the wrong TPM owner password, the TPM doesn't allow another attempt to reset the lockout state for 24 hours.

TPM 2.0 allows some keys to be created without an authorization value associated with them. These keys can be used when the TPM is locked. For example, BitLocker with a default TPM-only configuration is able to use a key in the TPM to start Windows, even when the TPM is locked.

Rationale behind the defaults

Originally, BitLocker allowed from 4 to 20 characters for a PIN. Windows Hello has its own PIN for sign-in, which can be 4 to 127 characters. Both BitLocker and Windows Hello use the TPM to prevent PIN brute-force attacks.

Windows 10, version 1607 and earlier used Dictionary Attack Prevention parameters. The Dictionary Attack Prevention Parameters provide a way to balance security needs with usability. For example, when BitLocker is used with a TPM + PIN configuration, the number of PIN guesses is limited over time. A TPM 2.0 in this example could be configured to allow only 32 PIN guesses immediately, and then only one more guess every two hours. This totals a maximum of about 4,415 guesses per year. If the PIN is four digits, all 9999 possible PIN combinations could be attempted in a little over two years.

Starting in Windows 10, version 1703, the minimum length for the BitLocker PIN was increased to six characters, to better align with other Windows features that use TPM 2.0, including Windows Hello. Increasing the PIN length requires a greater number of guesses for an attacker. Therefore, the lockout duration between each guess was shortened to allow legitimate users to

retry a failed attempt sooner while maintaining a similar level of protection. In case the legacy parameters for lockout threshold and recovery time need to be used, make sure that GPO is enabled and configure the system to use legacy Dictionary Attack Prevention Parameters setting for TPM 2.0.

TPM-based smart cards

The Windows TPM-based smart card, which is a virtual smart card, can be configured to allow sign in to the system. In contrast with physical smart cards, the sign-in process uses a TPM-based key with an authorization value. The following list shows the advantages of virtual smart cards:

- Physical smart cards can enforce lockout for only the physical smart card PIN, and they can reset the lockout after the correct PIN is entered. With a virtual smart card, the TPM's anti-hammering protection isn't reset after a successful authentication. The allowed number of authorization failures before the TPM enters lockout includes many factors.
- Hardware manufacturers and software developers can use the security features of the TPM to meet their requirements.
- The intent of selecting 32 failures as the lock-out threshold is to avoid users to lock the TPM (even when learning to type new passwords or if they frequently lock and unlock their computers). If users lock the TPM, they must wait 10 minutes or use other credentials to sign in, such as a user name and password.

Last updated on 08/15/2025

How Windows uses the Trusted Platform Module

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Windows operating system places hardware-based security deeper inside many features, maximizing platform security while increasing usability. To achieve many of these security enhancements, Windows makes extensive use of the Trusted Platform Module (TPM). This article offers an overview of the TPM, describes how it works, and discusses the benefits that TPM brings to Windows and the cumulative security effect of running Windows on a device with a TPM.

TPM Overview

The TPM is a cryptographic module that enhances computer security and privacy. Protecting data through encryption and decryption, protecting authentication credentials, and proving which software is running on a system are basic functionalities associated with computer security. The TPM helps with all these scenarios and more.

Historically, TPMs have been discrete chips soldered to a computer's motherboard. Such implementations allow the computer's original equipment manufacturer (OEM) to evaluate and certify the TPM separate from the rest of the system. Although discrete TPM implementations are still common, they can be problematic for integrated devices that are small or have low power consumption. Some newer TPM implementations integrate TPM functionality into the same chipset as other platform components while still providing logical separation similar to discrete TPM chips.

TPMs are passive: they receive commands and return responses. To realize the full benefit of a TPM, the OEM must carefully integrate system hardware and firmware with the TPM to send it commands and react to its responses. TPMs were originally designed to provide security and privacy benefits to a platform's owner and users, but newer versions can provide security and privacy benefits to the system hardware itself. Before it can be used for advanced scenarios, a TPM must be provisioned. Windows automatically provisions a TPM, but if the operating system is reinstalled, the TPM may be required to be explicitly reprovisioned before it can use all the TPM's features.

The Trusted Computing Group (TCG) is the nonprofit organization that publishes and maintains the TPM specification. The TCG exists to develop, define, and promote vendor-neutral, global industry standards that support a hardware-based root of trust for interoperable trusted computing platforms. The TCG also publishes the TPM specification as the international

standard ISO/IEC 11889, using the Publicly Available Specification Submission Process that the Joint Technical Committee 1 defines between the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

OEMs implement the TPM as a component in a trusted computing platform, such as a PC, tablet, or phone. Trusted computing platforms use the TPM to support privacy and security scenarios that software alone can't achieve. For example, software alone can't reliably report whether malware is present during the system startup process. The close integration between TPM and platform increases the transparency of the startup process and supports evaluating device health by enabling reliable measuring and reporting of the software that starts the device. Implementation of a TPM as part of a trusted computing platform provides a hardware root of trust-that is, it behaves in a trusted way. For example, if a key stored in a TPM has properties that disallow exporting the key, that key *truly can't leave the TPM*.

The TCG designed the TPM as a low-cost, mass-market security solution that addresses the requirements of different customer segments. There are variations in the security properties of different TPM implementations just as there are variations in customer and regulatory requirements for different sectors. In public-sector procurement, for example, some governments clearly define security requirements for TPMs, whereas others don't.

Certification programs for TPMs-and technology in general-continue to evolve as the speed of innovation increases. Although having a TPM is clearly better than not having a TPM, Microsoft's best advice is to determine your organization's security needs and research any regulatory requirements associated with procurement for your industry. The result is a balance between scenarios used, assurance level, cost, convenience, and availability.

TPM in Windows

The security features of Windows combined with the benefits of a TPM offer practical security and privacy benefits. The following sections start with major TPM-related security features in Windows and go on to describe how key technologies use the TPM to enable or increase security.

Platform Crypto Provider

Windows includes a cryptography framework called Cryptographic API: Next Generation (CNG), the basic approach of which is to implement cryptographic algorithms in different ways but with a common application programming interface (API). Applications that use cryptography can use the common API without knowing the details of how an algorithm is implemented much less the algorithm itself.

Although CNG sounds like a mundane starting point, it illustrates some of the advantages that a TPM provides. Underneath the CNG interface, Windows or third parties supply a cryptographic provider (that is, an implementation of an algorithm) implemented as software libraries alone or in a combination of software and available system hardware or non-Microsoft hardware. If implemented through hardware, the cryptographic provider communicates with the hardware behind the software interface of CNG.

The Platform Crypto Provider, introduced in the Windows 8, exposes the following special TPM properties, which software-only CNG providers can't offer or can't offer as effectively:

- **Key protection**. The Platform Crypto Provider can create keys in the TPM with restrictions on their use. The operating system can load and use the keys in the TPM without copying the keys to system memory, where they're vulnerable to malware. The Platform Crypto Provider can also configure keys that a TPM protects so that they aren't removable. If a TPM creates a key, the key is unique and resides only in that TPM. If the TPM imports a key, the Platform Crypto Provider can use the key in that TPM, but that TPM isn't a source for making more copies of the key or enabling the use of copies elsewhere. In sharp contrast, software solutions that protect keys from copying are subject to reverse-engineering attacks, in which someone figures out how the solution stores keys or makes copies of keys while they are in memory during use.

- **Dictionary attack protection**. Keys that a TPM protects can require an authorization value such as a PIN. With dictionary attack protection, the TPM can prevent attacks that attempt a large number of guesses to determine the PIN. After too many guesses, the TPM simply returns an error saying no more guesses are allowed for a period of time. Software solutions might provide similar features, but they can't provide the same level of protection, especially if the system restarts, the system clock changes, or files on the hard disk that count failed guesses are rolled back. In addition, with dictionary attack protection, authorization values such as PINs can be shorter and easier to remember while still providing the same level of protection as more complex values when using software solutions.

These TPM features give Platform Crypto Provider distinct advantages over software-based solutions. A practical way to see these benefits in action is when using certificates on a Windows device. On platforms that include a TPM, Windows can use the Platform Crypto Provider to provide certificate storage. Certificate templates can specify that a TPM use the Platform Crypto Provider to protect the key associated with a certificate. In mixed environments, where some computers might not have a TPM, the certificate template could prefer the Platform Crypto Provider over the standard Windows software provider. If a certificate is configured as not able to be exported, the private key for the certificate is restricted and can't be exported from the TPM. If the certificate requires a PIN, the PIN gains the TPM's dictionary attack protection automatically.

Virtual Smart Card

> ⚠ **Warning**
>
> **Windows Hello for Business** and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

Smart cards are physical devices that typically store a single certificate and the corresponding private key. Users insert a smart card into a built-in or USB card reader and enter a PIN to unlock it. Windows can then access the card's certificate and use the private key for authentication or to unlock BitLocker protected data volumes. Smart cards are popular because they provide two-factor authentication that requires both something the user has (that is, the smart card) and something the user knows (such as the smart card PIN). However, smart cards can be expensive because they require purchase and deployment of both smart cards and smart card readers.

In Windows, the Virtual Smart Card feature allows the TPM to mimic a permanently inserted smart card. The TPM becomes *something the user has* but still requires a PIN. While physical smart cards limit the number of PIN attempts before locking the card and requiring a reset, a virtual smart card relies on the TPM's dictionary attack protection to prevent too many PIN guesses.

For TPM-based virtual smart cards, the TPM protects the use and storage of the certificate private key, so that it can't be copied when it is in use or stored and used elsewhere. Using a component that is part of the system rather than a separate physical smart card, can reduce total cost of ownership. The *lost card* or *card left at home* scenarios aren't applicable, and the benefits of smart card-based multifactor authentication is preserved. For users, virtual smart cards are simple to use, requiring only a PIN to unlock. Virtual smart cards support the same scenarios that physical smart cards support, including signing in to Windows or authenticating for resource access.

Windows Hello for Business

Windows Hello for Business provides authentication methods intended to replace passwords, which can be difficult to remember and easily compromised. In addition, username/password solutions for authentication often reuse the same credential combinations on multiple devices and services. If those credentials are compromised, they're compromised in multiple places. Windows Hello for Business combines the information provisioned on each device (that is, the

cryptographic key) with additional information to authenticate users. On a system that has a TPM, the TPM can protect the key. If a system doesn't have a TPM, software-based techniques protect the key. The additional information the user supplies can be a PIN value or, if the system has the necessary hardware, biometric information, such as fingerprint or facial recognition. To protect privacy, the biometric information is used only on the provisioned device to access the provisioned key: it isn't shared across devices.

The adoption of new authentication technology requires that identity providers and organizations deploy and use that technology. Windows Hello for Business lets users authenticate with their existing Microsoft account, an Active Directory account, a Microsoft Entra account, or even non-Microsoft Identity Provider Services or Relying Party Services that support Fast ID Online V2.0 authentication.

Identity providers have flexibility in how they provision credentials on client devices. For example, an organization might provision only those devices that have a TPM so that the organization knows that a TPM protects the credentials. The ability to distinguish a TPM from malware acting like a TPM requires the following TPM capabilities (see Figure 1):

- **Endorsement key**. The TPM manufacturer can create a special key in the TPM called an endorsement key. An endorsement key certificate, signed by the manufacturer, says that the endorsement key is present in a TPM that the manufacturer made. Solutions can use the certificate with the TPM containing the endorsement key to confirm a scenario really involves a TPM from a specific TPM manufacturer (instead of malware acting like a TPM).
- **Attestation identity key**. To protect privacy, most TPM scenarios don't directly use an actual endorsement key. Instead, they use attestation identity keys, and an identity certificate authority (CA) uses the endorsement key and its certificate to prove that one or more attestation identity keys actually exist in a real TPM. The identity CA issues attestation identity key certificates. More than one identity CA will generally see the same endorsement key certificate that can uniquely identify the TPM, but any number of attestation identity key certificates can be created to limit the information shared in other scenarios.

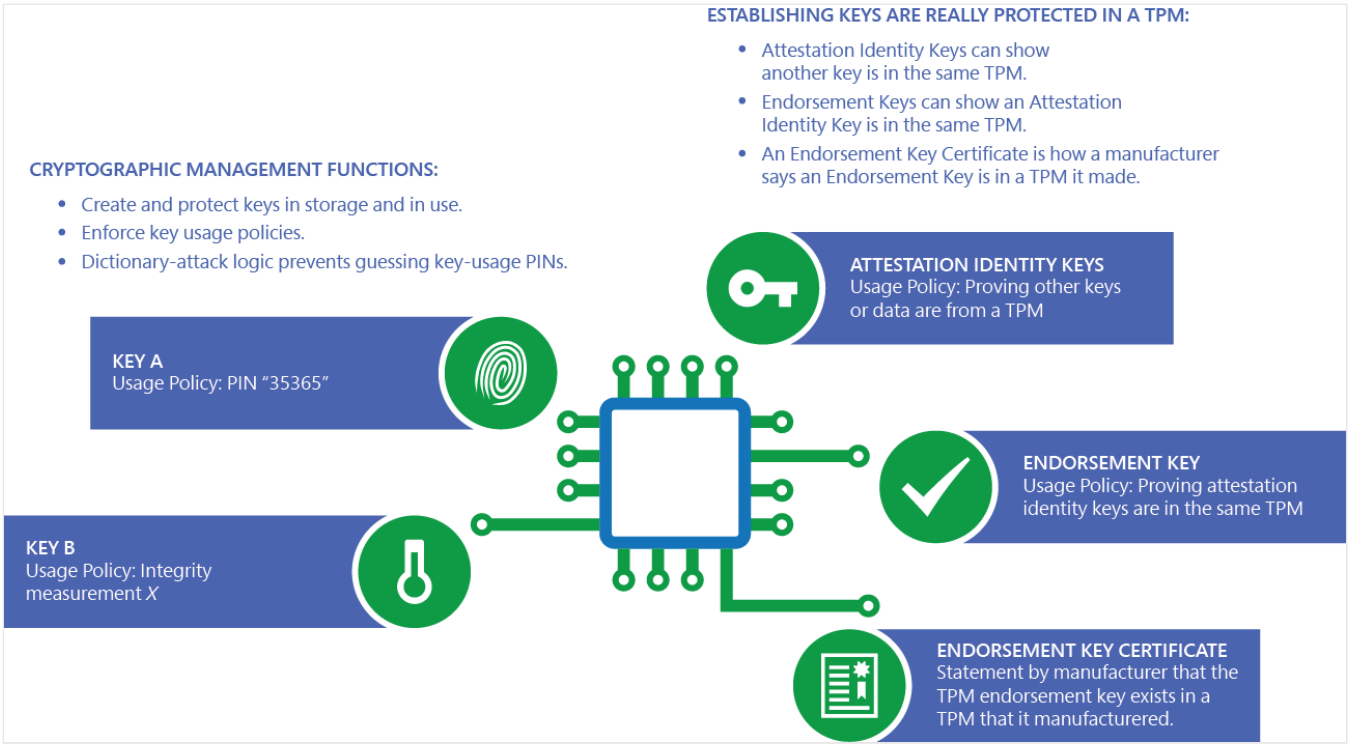


Figure 1: TPM Cryptographic Key Management

For Windows Hello for Business, Microsoft can fill the role of the identity CA. Microsoft services can issue an attestation identity key certificate for each device, user, and identify provider to ensure that privacy is protected and to help identity providers ensure that device TPM requirements are met before Windows Hello for Business credentials are provisioned.

BitLocker Drive Encryption

BitLocker provides full-volume encryption to protect data at rest. The most common device configuration splits the hard drive into several volumes. The operating system and user data reside on one volume that holds confidential information, and other volumes hold public information such as boot components, system information, and recovery tools. (These other volumes are used infrequently enough that they don't need to be visible to users.) Without more protections in place, if the volume containing the operating system and user data isn't encrypted, someone can boot another operating system and easily bypass the intended operating system's enforcement of file permissions to read any user data.

In the most common configuration, BitLocker encrypts the operating system volume so that if the computer or hard disk is lost or stolen when powered off, the data on the volume remains confidential. When the computer is turned on, starts normally, and proceeds to the Windows sign-in prompt, the only path forward is for the user to sign in with their credentials, allowing the operating system to enforce its normal file permissions. If something about the boot process changes, however-for example, a different operating system is booted from a USB device-the operating system volume and user data can't be read and aren't accessible. The TPM and system firmware collaborate to record measurements of how the system started,

including loaded software and configuration details such as whether boot occurred from the hard drive or a USB device. BitLocker relies on the TPM to allow the use of a key only when startup occurs in an expected way. The system firmware and TPM are carefully designed to work together to provide the following capabilities:

- **Hardware root of trust for measurement**. A TPM allows software to send it commands that record measurements of software or configuration information. This information can be calculated using a hash algorithm that essentially transforms a lot of data into a small, statistically unique hash value. The system firmware has a component called the Core Root of Trust for Measurement (CRTM) that is implicitly trusted. The CRTM unconditionally hashes the next software component and records the measurement value by sending a command to the TPM. Successive components, whether system firmware or operating system loaders, continue the process by measuring any software components they load before running them. Because each component's measurement is sent to the TPM before it runs, a component can't erase its measurement from the TPM. (However, measurements are erased when the system is restarted.) The result is that at each step of the system startup process, the TPM holds measurements of boot software and configuration information. Any changes in boot software or configuration yield different TPM measurements at that step and later steps. Because the system firmware unconditionally starts the measurement chain, it provides a hardware-based root of trust for the TPM measurements. At some point in the startup process, the value of recording all loaded software and configuration information diminishes and the chain of measurements stops. The TPM allows for the creation of keys that can be used only when the platform configuration registers that hold the measurements have specific values.

- **Key used only when boot measurements are accurate**. BitLocker creates a key in the TPM that can be used only when the boot measurements match an expected value. The expected value is calculated for the step in the startup process when Windows Boot Manager runs from the operating system volume on the system hard drive. Windows Boot Manager, which is stored unencrypted on the boot volume, needs to use the TPM key so that it can decrypt data read into memory from the operating system volume and startup can proceed using the encrypted operating system volume. If a different operating system is booted or the configuration is changed, the measurement values in the TPM will be different, the TPM won't let Windows Boot Manager use the key, and the startup process can't proceed normally because the data on the operating system can't be decrypted. If someone tries to boot the system with a different operating system or a different device, the software or configuration measurements in the TPM will be wrong and the TPM won't allow use of the key needed to decrypt the operating system volume. As a failsafe, if measurement values change unexpectedly, the user can always use the BitLocker recovery key to access volume data. Organizations can configure BitLocker to store the recovery key-in Active Directory Domain Services (AD DS).

Device hardware characteristics are important to BitLocker and its ability to protect data. One consideration is whether the device provides attack vectors when the system is at the sign-in screen. For example, if the Windows device has a port that allows direct memory access so that someone can plug in hardware and read memory, an attacker can read the operating system volume's decryption key from memory while at the Windows sign-in screen. To mitigate this risk, organizations can configure BitLocker so that the TPM key requires both the correct software measurements and an authorization value. The system startup process stops at Windows Boot Manager, and the user is prompted to enter the authorization value for the TPM key or insert a USB device with the value. This process stops BitLocker from automatically loading the key into memory where it might be vulnerable, but has a less desirable user experience.

Newer hardware and Windows work better together to disable direct memory access through ports and reduce attack vectors. The result is that organizations can deploy more systems without requiring users to enter additional authorization information during the startup process. The right hardware allows BitLocker to be used with the "TPM-only" configuration giving users a single sign-on experience without having to enter a PIN or USB key during boot.

Device Encryption

Device Encryption is the consumer version of BitLocker, and it uses the same underlying technology. How it works is if a customer logs on with a Microsoft account and the system meets Modern Standby hardware requirements, BitLocker Drive Encryption is enabled automatically in Windows. The recovery key is backed up in the Microsoft cloud and is accessible to the consumer through his or her Microsoft account. The Modern Standby hardware requirements inform Windows that the hardware is appropriate for deploying Device Encryption and allows use of the "TPM-only" configuration for a simple consumer experience. In addition, Modern Standby hardware is designed to reduce the likelihood that measurement values change and prompt the customer for the recovery key.

For software measurements, Device Encryption relies on measurements of the authority providing software components (based on code signing from manufacturers such as OEMs or Microsoft) instead of the precise hashes of the software components themselves. This permits servicing of components without changing the resulting measurement values. For configuration measurements, the values used are based on the boot security policy instead of the numerous other configuration settings recorded during startup. These values also change less frequently. The result is that Device Encryption is enabled on appropriate hardware in a user-friendly way while also protecting data.

Measured Boot

Windows 8 introduced Measured Boot as a way for the operating system to record the chain of measurements of software components and configuration information in the TPM through the initialization of the Windows operating system. In previous Windows versions, the measurement chain stopped at the Windows Boot Manager component itself, and the measurements in the TPM weren't helpful for understanding the starting state of Windows.

The Windows boot process happens in stages and often involves non-Microsoft drivers to communicate with vendor-specific hardware or implement anti-malware solutions. For software, Measured Boot records measurements of the Windows kernel, Early-Launch anti-malware drivers, and boot drivers in the TPM. For configuration settings, Measured Boot records security-relevant information such as signature data that anti-malware drivers use and configuration data about Windows security features (for example, whether BitLocker is on or off).

Measured Boot ensures that TPM measurements fully reflect the starting state of Windows software and configuration settings. If security settings and other protections are set up correctly, they can be trusted to maintain the security of the running operating system thereafter. Other scenarios can use the operating system's starting state to determine whether the running operating system should be trusted.

TPM measurements are designed to avoid recording any privacy-sensitive information as a measurement. As an additional privacy protection, Measured Boot stops the measurement chain at the initial starting state of Windows. Therefore, the set of measurements doesn't include details about which applications are in use or how Windows is being used. Measurement information can be shared with external entities to show that the device is enforcing adequate security policies and didn't start with malware.

The TPM provides the following way for scenarios to use the measurements recorded in the TPM during boot:

- **Remote Attestation**. Using an attestation identity key, the TPM can generate and cryptographically sign a statement (or *quote*) of the current measurements in the TPM. Windows can create unique attestation identity keys for various scenarios to prevent separate evaluators from collaborating to track the same device. Additional information in the quote is cryptographically scrambled to limit information sharing and better protect privacy. By sending the quote to a remote entity, a device can attest which software and configuration settings were used to boot the device and initialize the operating system. An attestation identity key certificate can provide further assurance that the quote is coming from a real TPM. Remote attestation is the process of recording measurements in the TPM, generating a quote, and sending the quote information to another system that evaluates the measurements to establish trust in a device. Figure 2 illustrates this process.

When new security features are added to Windows, Measured Boot adds security-relevant configuration information to the measurements recorded in the TPM. Measured Boot enables remote attestation scenarios that reflect the system firmware and the Windows initialization state.

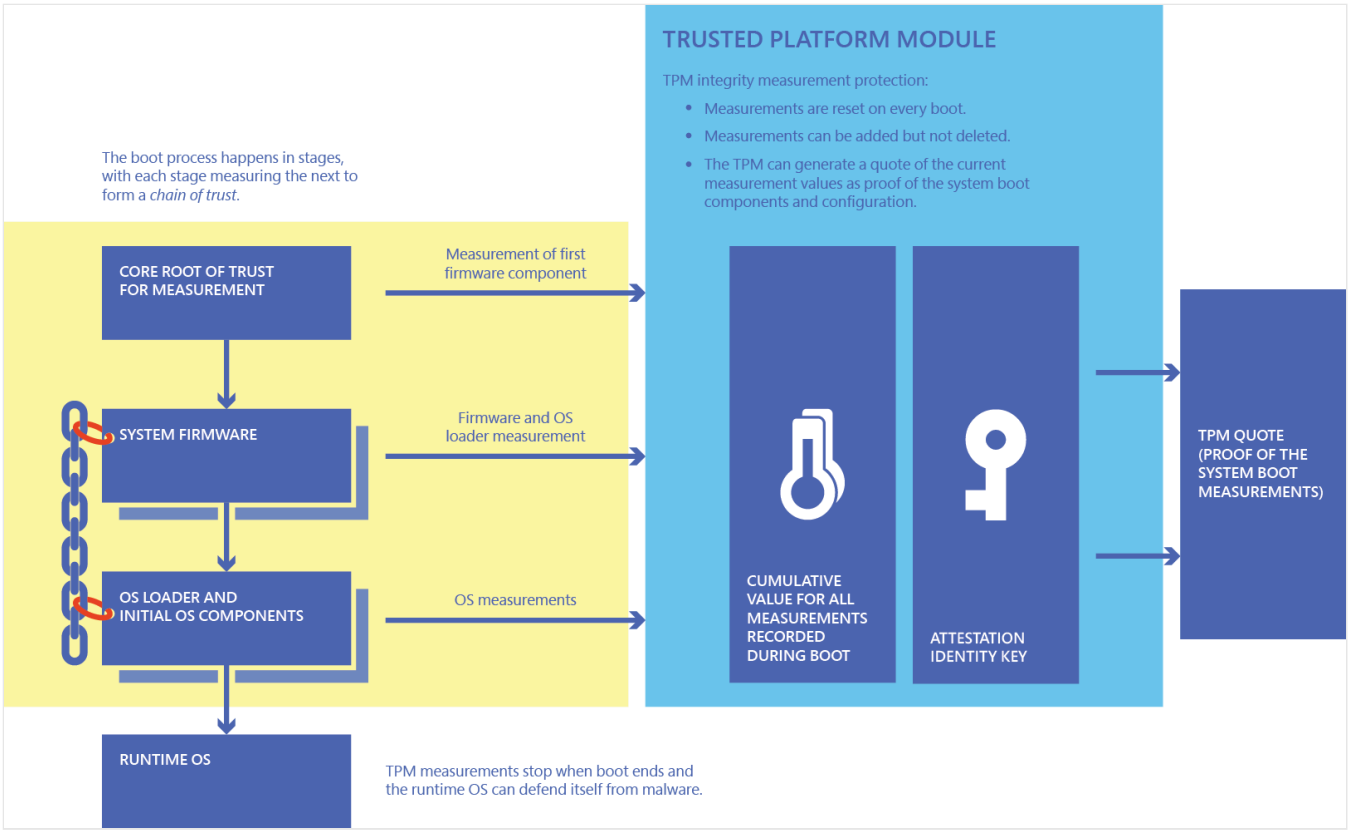


Figure 2: Process used to create evidence of boot software and configuration using a TPM

Health Attestation

Some Windows improvements help security solutions implement remote attestation scenarios. Microsoft provides a Health Attestation service, which can create attestation identity key certificates for TPMs from different manufacturers and parse measured boot information to extract simple security assertions, such as whether BitLocker is on or off. The simple security assertions can be used to evaluate device health.

Mobile device management (MDM) solutions can receive simple security assertions from the Microsoft Health Attestation service for a client without having to deal with the complexity of the quote or the detailed TPM measurements. MDM solutions can act on the security information by quarantining unhealthy devices or blocking access to cloud services such as Microsoft Office 365.

Credential Guard

Credential Guard is a new feature in Windows that helps protect Windows credentials in organizations that have deployed AD DS. Historically, a user's credentials (such as a sign-in password) were hashed to generate an authorization token. The user employed the token to access resources that they were permitted to use. One weakness of the token model is that malware that had access to the operating system kernel could look through the computer's memory and harvest all the access tokens currently in use. The attacker could then use harvested tokens to sign in to other machines and collect more credentials. This kind of attack is called a "pass the hash" attack, a malware technique that infects one machine to infect many machines across an organization.

Similar to the way Microsoft Hyper-V keeps virtual machines (VMs) separate from one another, Credential Guard uses virtualization to isolate the process that hashes credentials in a memory area that the operating system kernel can't access. This isolated memory area is initialized and protected during the boot process so that components in the larger operating system environment can't tamper with it. Credential Guard uses the TPM to protect its keys with TPM measurements, so they're accessible only during the boot process step when the separate region is initialized; they aren't available for the normal operating system kernel. The local security authority code in the Windows kernel interacts with the isolated memory area by passing in credentials and receiving single-use authorization tokens in return.

The resulting solution provides defense in depth, because even if malware runs in the operating system kernel, it can't access the secrets inside the isolated memory area that actually generates authorization tokens. The solution doesn't solve the problem of key loggers because the passwords such loggers capture actually pass through the normal Windows kernel, but when combined with other solutions, such as smart cards for authentication, Credential Guard greatly enhances the protection of credentials in Windows.

Conclusion

The TPM adds hardware-based security benefits to Windows. When installed on hardware that includes a TPM, Window delivers remarkably improved security benefits. The following table summarizes the key benefits of the TPM's major features.

Expand table

Feature	Benefits when used on a system with a TPM
Platform Crypto Provider	- If the machine is compromised, the private key associated with the certificate can't be copied off the device. - The TPM's dictionary attack mechanism protects PIN values to use a certificate.
Virtual Smart Card	Achieve security similar to that of physical smart cards without deploying physical smart cards or card readers.

Feature	Benefits when used on a system with a TPM
Windows Hello for Business	- Credentials provisioned on a device can't be copied elsewhere. - Confirm a device's TPM before credentials are provisioned.
BitLocker Drive Encryption	Multiple options are available for enterprises to protect data at rest while balancing security requirements with different device hardware.
Device Encryption	With a Microsoft account and the right hardware, consumers' devices seamlessly benefit from data-at-rest protection.
Measured Boot	A hardware root of trust contains boot measurements that help detect malware during remote attestation.
Health Attestation	MDM solutions can easily perform remote attestation and evaluate client health before granting access to resources or cloud services such as Office 365.
Credential Guard	Defense in depth increases so that even if malware has administrative rights on one machine, it's significantly more difficult to compromise additional machines in an organization.

Although some of the aforementioned features have more hardware requirements (for example, virtualization support), the TPM is a cornerstone of Windows security. Microsoft and other industry stakeholders continue to improve the global standards associated with TPM and find more applications that use it to provide tangible benefits to customers. Microsoft has included support for most TPM features in its version of Windows for the Internet of Things (IoT) called Windows IoT Core. IoT devices that might be deployed in insecure physical locations and connected to cloud services like Azure IoT Hub for management can use the TPM in innovative ways to address their emerging security requirements.

Last updated on 08/15/2025

Manage TPM commands

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article for the IT professional describes how to manage which Trusted Platform Module (TPM) commands are available to domain users and to local users.

After a computer user takes ownership of the TPM, the TPM owner can limit which TPM commands can be run by creating a list of blocked TPM commands. The list can be created and applied to all computers in a domain by using Group Policy, or a list can be created for individual computers by using the TPM MMC. Because some hardware vendors might provide more commands or the Trusted Computing Group might decide to add commands in the future, the TPM MMC also supports the ability to block new commands.

The following procedures describe how to manage the TPM command lists. You must be a member of the local Administrators group.

Block TPM commands by using the Local Group Policy Editor

1. Open the Local Group Policy Editor (`gpedit.msc`). If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then select **Yes**.

   > ⓘ **Note**
   >
   > Administrators with appropriate rights in a domain can configure a Group Policy Object (GPO) that can be applied through Active Directory Domain Services (AD DS).

2. In the console tree, under **Computer Configuration**, expand **Administrative Templates**, and then expand **System**.

3. Under **System**, select **Trusted Platform Module Services**.

4. In the details pane, double-click **Configure the list of blocked TPM commands**.

5. Select **Enabled**, and then select **Show**.

6. For each command that you want to block, select **Add**, enter the command number, and then select **OK**.

   > ⓘ **Note**

For a list of commands, see links in the TPM Specification.

7. After adding numbers for each command that you want to block, select **OK** twice.
8. Close the Local Group Policy Editor.

Block or allow TPM commands by using the TPM MMC

1. Open the TPM MMC (`tpm.msc`). If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then select **Yes**.
2. In the console tree, select **Command Management**. A list of TPM commands is displayed.
3. In the list, select a command that you want to block or allow.
4. Under **Actions**, select **Block Selected Command** or **Allow Selected Command** as needed. If **Allow Selected Command** is unavailable, that command is currently blocked by Group Policy.

Block new commands

1. Open the TPM MMC (`tpm.msc`). If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then select **Yes**.
2. In the console tree, select **Command Management**. A list of TPM commands is displayed.
3. In the **Action** pane, select **Block New Command**. The **Block New Command** dialog box is displayed.
4. In the **Command Number** text box, type the number of the new command that you want to block, and then select **OK**. The command number you entered is added to the blocked list.

Use the TPM cmdlets

You can manage the TPM using Windows PowerShell. For details, see TrustedPlatformModule PowerShell cmdlets.

Last updated on 08/15/2025

Manage TPM lockout

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article for the IT professional describes how to manage the lockout feature for the Trusted Platform Module (TPM) in Windows.

About TPM lockout

The TPM locks itself to prevent tampering or malicious attacks. TPM lockout often lasts for a variable amount of time or until the computer is turned off. While the TPM is in lockout mode, it generally returns an error message when it receives commands that require an authorization value. One exception is that the TPM always allows the owner at least one attempt to reset the TPM lockout when it is in lockout mode.

Windows takes ownership of the TPM ownership upon first boot. By default, Windows doesn't retain the TPM owner password.

In some cases, encryption keys are protected by a TPM by requiring a valid authorization value to access the key. A common example is configuring BitLocker Drive Encryption to use the TPM plus PIN key protector. In this scenario, the user must type the correct PIN during the boot process to access the volume encryption key protected by the TPM. To prevent malicious users or software from discovering authorization values, TPMs implement protection logic. The protection logic is designed to slow or stop responses from the TPM if it detects that an entity might be trying to guess authorization values.

TPM 2.0

TPM 2.0 devices have standardized lockout behavior which Windows configures. TPM 2.0 devices have a maximum count threshold and a healing time. Windows configures the maximum count to be 32 and the healing time to be 10 minutes. This configuration means that every continuous 10 minutes of powered on operation without an event causes the counter to decrease by 1.

If your TPM is in lockout mode or is responding slowly to commands, you can reset the lockout value by using the following procedures. Resetting the TPM lockout requires the TPM owner's authorization. This value is no longer retained by default starting with Windows 10 version 1607 and higher.

TPM 1.2

The industry standards from the Trusted Computing Group (TCG) specify that TPM manufacturers must implement some form of protection logic in TPM 1.2 and TPM 2.0 chips. TPM 1.2 devices implement different protection mechanisms and behavior. In general, the TPM chip takes exponentially longer to respond if incorrect authorization values are sent to the TPM. Some TPM chips might not store failed attempts over time. Other TPM chips might store every failed attempt indefinitely. Therefore, some users might experience increasingly longer delays when they mistype an authorization value that is sent to the TPM. These delays can prevent them from using the TPM for some time.

Reset the TPM lockout by using the TPM MMC

> ⓘ **Note**
>
> This procedure is only available if you have configured Windows to retain the TPM Owner Password. By default, this password isn't available in Windows 10 starting with version 1607 and higher.

The following procedure explains the steps to reset the TPM lockout by using the TPM MMC.

Reset the TPM lockout

1. Open the TPM MMC (tpm.msc).
2. In the **Action** pane, select **Reset TPM Lockout** to start the Reset TPM Lockout Wizard.
3. Choose one of the following methods to enter the TPM owner password:
    - If you saved your TPM owner password to a `.tpm` file, select **I have the owner password file**, and then type the path to the file, or select **Browse** to navigate to the file location.
    - If you want to manually enter your TPM owner password, select **I want to enter the owner password**, and then type the password in the text box provided.

> ⓘ **Note**
>
> If you enabled BitLocker and your TPM at the same time, and you printed your BitLocker recovery password when you turned on BitLocker, your TPM owner password may have printed with it.

Use Group Policy to manage TPM lockout settings

The TPM Group Policy settings in the following list are located at:

Computer Configuration > **Administrative Templates** > **System** > **Trusted Platform Module Services**

- Standard User Lockout Duration

  This policy setting allows you to manage the duration in minutes for counting standard user authorization failures for TPM commands that require authorization. An authorization failure occurs each time a user sends a command to the TPM and receives an error message that indicates an authorization failure occurred. Authorization failures that are older than the duration you set are ignored. If the number of TPM commands with an authorization failure within the lockout duration equals a threshold, the user is prevented from sending commands to the TPM that require authorization.

- Standard User Individual Lockout Threshold

  This policy setting allows you to manage the maximum number of authorization failures for the TPM for each user. This value is the maximum number of authorization failures that each user can have before the user isn't allowed to send commands to the TPM that require authorization. If the number of authorization failures equals the duration that is set for the policy setting, the user is prevented from sending commands to the TPM that require authorization.

- Standard User Total Lockout Threshold

  This policy setting allows you to manage the maximum number of authorization failures for the TPM for all standard users. If the total number of authorization failures for all users equals the duration that is set for the policy, all users are prevented from sending commands to the TPM that require authorization.

For information about mitigating dictionary attacks that use the lockout settings, see TPM fundamentals.

Use the TPM cmdlets

You can manage the TPM using Windows PowerShell. For details, see TPM Cmdlets in Windows PowerShell.

Last updated on 08/15/2025

Change the TPM owner password

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article for the IT professional describes how to change the password or PIN for the owner of the Trusted Platform Module (TPM) that is installed on your system.

About the TPM owner password

Starting with Windows 10, version 1607, Windows doesn't retain the TPM owner password when provisioning the TPM. The password is set to a random high entropy value and then discarded.

> **ⓘ Important**
>
> Although the TPM owner password isn't retained starting with Windows 10, version 1607, you can change a default registry key to retain it. However, we strongly recommend that you don't make this change. To retain the TPM owner password, under the registry key `HKLM\Software\Policies\Microsoft\TPM`, create a `REG_DWORD` value of `OSManagedAuthLevel` and set it to `4`.
>
> For Windows versions newer than Windows 10 1703, the default value for this key is 5. A value of 5 means:
>
> - **TPM 2.0**: Keep the lockout authorization.
> - **TPM 1.2**: Discard the Full TPM owner authorization and retain only the Delegated authorization.
>
> Unless the registry key value is changed from 5 to 4 before the TPM is provisioned, the owner password isn't saved.

Only one owner password exists for each TPM. The TPM owner password allows the ability to enable, disable, or clear the TPM without having physical access to the computer, for example, by using the command-line tools remotely. The TPM owner password also allows manipulation of the TPM dictionary attack logic. Windows takes ownership of the TPM as part of the provisioning process on each boot. Ownership can change when you share the password or clear your ownership of the TPM so someone else can initialize it.

Without the owner password, you can still perform all the preceding actions with a physical presence confirmation from UEFI.

Other TPM management options

Instead of changing your owner password, you can also use the following options to manage your TPM:

- **Clear the TPM** - If you want to invalidate all of the existing keys that have been created since you took ownership of the TPM, you can clear it. For important precautions for this process, and instructions for completing it, see Clear all the keys from the TPM.
- **Turn off the TPM** - With TPM 1.2 and Windows 10, versions 1507 and 1511, you can turn off the TPM. Turn off the TPM if you want to keep all existing keys and data intact and disable the services that are provided by the TPM. For more info, see Turn off the TPM.

Changing the TPM owner password

With Windows 10, version 1507 or 1511, if you have opted specifically to preserve the TPM owner password, you can use the saved password to change to a new password.

To change to a new TPM owner password, in `TPM.msc`, select **Change Owner Password**, and follow the instructions. It prompts to provide the owner password file or to type the password. Then you can create a new password, either automatically or manually, and save the password in a file or as a printout.

Use the TPM cmdlets

You can manage the TPM using Windows PowerShell. For details, see TPM Cmdlets in Windows PowerShell.

Last updated on 08/15/2025

TPM Group Policy settings

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the Trusted Platform Module (TPM) Services that can be controlled centrally by using Group Policy settings. The Group Policy settings for TPM services are located under **Computer Configuration** > **Administrative Templates** > **System** > **Trusted Platform Module Services**.

Configure the list of blocked TPM commands

This policy setting allows you to manage the Group Policy list of Trusted Platform Module (TPM) commands blocked by Windows.

If you enable this policy setting, Windows blocks the specified commands from being sent to the TPM on the computer. TPM commands are referenced by a command number. For example, command number `129` is `TPM_OwnerReadInternalPub`, and command number `170` is `TPM_FieldUpgrade`.

If you disable or don't configure this policy setting, only those TPM commands specified through the default or local lists may be blocked by Windows. The default list of blocked TPM commands is preconfigured by Windows. You can view the default list by running `tpm.msc`, navigating to the "Command Management" section, and making visible the "On Default Block List" column. The local list of blocked TPM commands is configured outside of Group Policy by running `tpm.msc` or through scripting against the Win32_Tpm interface.

Configure the system to clear the TPM if it isn't in a ready state

This policy setting configures the system to prompt the user to clear the TPM if the TPM is detected to be in any state other than Ready. This policy takes effect only if the system's TPM is in a state other than Ready, including if the TPM is "Ready, with reduced functionality". The prompt to clear the TPM will start occurring after the next reboot, upon user sign-in only if the logged in user is part of the Administrators group for the system. The prompt can be dismissed, but will reappear after every reboot and sign-in until the policy is disabled or until the TPM is in a Ready state.

Ignore the default list of blocked TPM commands

This policy setting allows you to enforce or ignore the computer's local list of blocked Trusted Platform Module (TPM) commands.

If you enable this policy setting, Windows ignores the computer's local list of blocked TPM commands and will only block those TPM commands specified by Group Policy or the default list.

The local list of blocked TPM commands is configured outside of Group Policy by running `tpm.msc` or through scripting against the `Win32_Tpm` interface. The default list of blocked TPM commands is preconfigured by Windows. See the related policy setting to configure the Group Policy list of blocked TPM commands.

If you disable or don't configure this policy setting, Windows blocks the TPM commands found in the local list, in addition to commands in the Group Policy and default lists of blocked TPM commands.

Ignore the local list of blocked TPM commands

This policy setting configures how much of the TPM owner authorization information is stored in the registry of the local computer. Depending on the amount of TPM owner authorization information stored locally, the operating system and TPM-based applications can perform certain TPM actions, which require TPM owner authorization without requiring the user to enter the TPM owner password.

You can choose to have the operating system store either the full TPM owner authorization value, the TPM administrative delegation blob plus the TPM user delegation blob, or none.

If you enable this policy setting, Windows stores the TPM owner authorization in the registry of the local computer according to the operating system managed TPM authentication setting you choose.

Choose the operating system managed TPM authentication setting of "Full" to store the full TPM owner authorization, the TPM administrative delegation blob and the TPM user delegation blob in the local registry. This setting allows use of the TPM without requiring remote or external storage of the TPM owner authorization value. This setting is appropriate for scenarios, which don't depend on preventing reset of the TPM anti-hammering logic or changing the TPM owner authorization value. Some TPM-based applications may require this setting be changed before features, which depend on the TPM anti-hammering logic can be used.

Choose the operating system managed TPM authentication setting of "Delegated" to store only the TPM administrative delegation blob and the TPM user delegation blob in the local

registry. This setting is appropriate for use with TPM-based applications that depend on the TPM anti-hammering logic.

Choose the operating system managed TPM authentication setting of "None" for compatibility with previous operating systems and applications or for use with scenarios that require TPM owner authorization not be stored locally. Using this setting might cause issues with some TPM-based applications.

> **Note**
>
> If the operating system managed TPM authentication setting is changed from "Full" to "Delegated", the full TPM owner authorization value is regenerated and any copies of the original TPM owner authorization value are invalidated.

Configure the level of TPM owner authorization information available to the operating system

> **Important**
>
> Beginning with Windows 10 version 1703, the default value is 5. This value is implemented during provisioning so that another Windows component can either delete it or take ownership of it, depending on the system configuration. For TPM 2.0, a value of 5 means keep the lockout authorization. For TPM 1.2, it means discard the Full TPM owner authorization and retain only the Delegated authorization.

This policy setting configured which TPM authorization values are stored in the registry of the local computer. Certain authorization values are required in order to allow Windows to perform certain actions.

Expand table

TPM 1.2 value	TPM 2.0 value	Purpose	Kept at level 0?	Kept at level 2?	Kept at level 4?
OwnerAuthAdmin	StorageOwnerAuth	Create SRK	No	Yes	Yes
OwnerAuthEndorsement	EndorsementAuth	Create or use EK (1.2 only: Create AIK)	No	Yes	Yes
OwnerAuthFull	LockoutAuth	Reset/change Dictionary Attack	No	No	Yes

TPM 1.2 value	TPM 2.0 value	Purpose	Kept at level 0?	Kept at level 2?	Kept at level 4?
		Protection			

There are three TPM owner authentication settings that are managed by the Windows operating system. You can choose a value of **Full**, **Delegate**, or **None**.

- **Full**: This setting stores the full TPM owner authorization, the TPM administrative delegation blob, and the TPM user delegation blob in the local registry. With this setting, you can use the TPM without requiring remote or external storage of the TPM owner authorization value. This setting is appropriate for scenarios that don't require you to reset the TPM anti-hammering logic or change the TPM owner authorization value. Some TPM-based applications may require that this setting is changed before features that depend on the TPM anti-hammering logic can be used. Full owner authorization in TPM 1.2 is similar to lockout authorization in TPM 2.0. Owner authorization has a different meaning for TPM 2.0.
- **Delegated**: This setting stores only the TPM administrative delegation blob and the TPM user delegation blob in the local registry. This setting is appropriate for use with TPM-based applications that depend on the TPM antihammering logic. This is the default setting in Windows prior to version 1703.
- **None**: This setting provides compatibility with previous operating systems and applications. You can also use it for scenarios when TPM owner authorization can't be stored locally. Using this setting might cause issues with some TPM-based applications.

> **Note**
>
> If the operating system managed TPM authentication setting is changed from **Full** to **Delegated**, the full TPM owner authorization value will be regenerated, and any copies of the previously set TPM owner authorization value will be invalid.

Registry information

Registry key: `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\TPM` DWORD: `OSManagedAuthLevel`

The following table shows the TPM owner authorization values in the registry.

Expand table

Value Data	Setting
0	None
2	Delegated
4	Full

If you enable this policy setting, the Windows operating system stores the TPM owner authorization in the registry of the local computer according to the TPM authentication setting you choose.

On Windows 10 prior to version 1607, if you disable or don't configure this policy setting, and the **Turn on TPM backup to Active Directory Domain Services** policy setting is also disabled or not configured, the default setting is to store the full TPM authorization value in the local registry. If this policy is disabled or not configured, and the **Turn on TPM backup to Active Directory Domain Services** policy setting is enabled, only the administrative delegation and the user delegation blobs are stored in the local registry.

Standard User Lockout Duration

This policy setting allows you to manage the duration in minutes for counting standard user authorization failures for Trusted Platform Module (TPM) commands requiring authorization. An authorization failure occurs each time a standard user sends a command to the TPM and receives an error response that indicates an authorization failure occurred. Authorization failures that are older than the duration you set are ignored. If the number of TPM commands with an authorization failure within the lockout duration equals a threshold, a standard user is prevented from sending commands that require authorization to the TPM.

The TPM is designed to protect itself against password guessing attacks by entering a hardware lockout mode when it receives too many commands with an incorrect authorization value. When the TPM enters a lockout mode, it's global for all users (including administrators) and for Windows features such as BitLocker Drive Encryption.

This setting helps administrators prevent the TPM hardware from entering a lockout mode by slowing the speed at which standard users can send commands that require authorization to the TPM.

For each standard user, two thresholds apply. Exceeding either threshold prevents the user from sending a command that requires authorization to the TPM. Use the following policy settings to set the lockout duration:

- Standard User Individual Lockout Threshold: This value is the maximum number of authorization failures that each standard user can have before the user isn't allowed to send commands that require authorization to the TPM.
- Standard User Total Lockout Threshold: This value is the maximum total number of authorization failures that all standard users can have before all standard users aren't allowed to send commands that require authorization to the TPM.

An administrator with the TPM owner password can fully reset the TPM's hardware lockout logic by using the Windows Defender Security Center. Each time an administrator resets the TPM's hardware lockout logic, all prior standard user TPM authorization failures are ignored. This allows standard users to immediately use the TPM normally.

If you don't configure this policy setting, a default value of 480 minutes (8 hours) is used.

Standard User Individual Lockout Threshold

This policy setting allows you to manage the maximum number of authorization failures for each standard user for the Trusted Platform Module (TPM). This value is the maximum number of authorization failures that each standard user can have before the user isn't allowed to send commands that require authorization to the TPM. If the number of authorization failures for the user within the duration that is set for the **Standard User Lockout Duration** policy setting equals this value, the standard user is prevented from sending commands that require authorization to the Trusted Platform Module (TPM).

This setting helps administrators prevent the TPM hardware from entering a lockout mode by slowing the speed at which standard users can send commands that require authorization to the TPM.

An authorization failure occurs each time a standard user sends a command to the TPM and receives an error response indicating an authorization failure occurred. Authorization failures older than the duration are ignored.

An administrator with the TPM owner password can fully reset the TPM's hardware lockout logic by using the Windows Defender Security Center. Each time an administrator resets the TPM's hardware lockout logic, all prior standard user TPM authorization failures are ignored. This allows standard users to immediately use the TPM normally.

If you don't configure this policy setting, a default value of 4 is used. A value of zero means that the operating system won't allow standard users to send commands to the TPM, which might cause an authorization failure.

Standard User Total Lockout Threshold

This policy setting allows you to manage the maximum number of authorization failures for all standard users for the Trusted Platform Module (TPM). If the total number of authorization failures for all standard users within the duration that is set for the **Standard User Lockout Duration** policy equals this value, all standard users are prevented from sending commands that require authorization to the Trusted Platform Module (TPM).

This setting helps administrators prevent the TPM hardware from entering a lockout mode because it slows the speed standard users can send commands requiring authorization to the TPM.

An authorization failure occurs each time a standard user sends a command to the TPM and receives an error response indicating an authorization failure occurred. Authorization failures older than the duration are ignored.

An administrator with the TPM owner password can fully reset the TPM's hardware lockout logic by using the Windows Defender Security Center. Each time an administrator resets the TPM's hardware lockout logic, all prior standard user TPM authorization failures are ignored. This allows standard users to immediately use the TPM normally.

If you don't configure this policy setting, a default value of 9 is used. A value of zero means that the operating system won't allow standard users to send commands to the TPM, which might cause an authorization failure.

Configure the system to use legacy Dictionary Attack Prevention Parameters setting for TPM 2.0

Introduced in Windows 10, version 1703, this policy setting configures the TPM to use the Dictionary Attack Prevention Parameters (lockout threshold and recovery time) to the values that were used for Windows 10 Version 1607 and below.

> ⓘ **Important**
>
> Setting this policy takes effect only if:
>
> - The TPM was originally prepared using a version of Windows after Windows 10 Version 1607
> - The system has a TPM 2.0.

> ⓘ **Note**

Enabling this policy will only take effect after the TPM maintenance task runs (which typically happens after a system restart). Once this policy has been enabled on a system and has taken effect (after a system restart), disabling it will have no impact and the system's TPM will remain configured using the legacy Dictionary Attack Prevention parameters, regardless of the value of this group policy. The only ways for the disabled setting of this policy to take effect on a system where it was once enabled are to either:

- Disable it from group policy
- Clear the TPM on the system

TPM Group Policy settings in Windows Security

You can change what users see about TPM in **Windows Security**. The Group Policy settings for the TPM area in **Windows Security** are located under **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Security** > **Device security**.

Disable the Clear TPM button

If you don't want users to be able to select the **Clear TPM** button in **Windows Security**, you can disable it with this Group Policy setting. Select **Enabled** to make the **Clear TPM** button unavailable for use.

Hide the TPM Firmware Update recommendation

If you don't want users to see the recommendation to update TPM firmware, you can disable it with this setting. Select **Enabled** to prevent users from seeing a recommendation to update their TPM firmware when a vulnerable firmware is detected.

Related articles

- TPM Cmdlets in Windows PowerShell
- BitLocker planning guide

Last updated on 08/15/2025

Back up the TPM recovery information to AD DS

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

In Windows 11, you can back up a device's Trusted Platform Module (TPM) information to Active Directory Domain Services (AD DS), enabling remote management of the TPM.

For more information, see Back up the TPM Recovery Information to AD DS.

Last updated on 08/15/2025

Troubleshoot the TPM

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article provides information how to troubleshoot the Trusted Platform Module (TPM):

- Troubleshoot TPM initialization
- Clear all the keys from the TPM
- Turn on or turn off the TPM

For information about the TPM cmdlets, see TPM Cmdlets in Windows PowerShell.

About TPM initialization and ownership

Windows automatically initializes and takes ownership of the TPM. There's no need for you to initialize the TPM and create an owner password.

TPM initialization

If you find that Windows isn't able to initialize the TPM automatically, review the following information:

- You can try clearing the TPM to the factory default values, allowing Windows to reinitialize it. For important precautions for this process, and instructions for completing it, see Clear all the keys from the TPM.
- If the TPM is a TPM 2.0 and isn't detected by Windows, verify that your computer hardware contains a Unified Extensible Firmware Interface (UEFI) that is Trusted Computing Group-compliant. Also, ensure that in the UEFI settings, the TPM hasn't been disabled or hidden from the operating system.
- If you have TPM 1.2 with Windows 11, the TPM might be turned off, and need to be turned back on, as described in Turn on the TPM. When it's turned back on, Windows reinitializes it.
- If you're attempting to set up BitLocker with the TPM, check which TPM driver is installed on the computer. We recommend always using one of the TPM drivers that is provided by Microsoft and is protected with BitLocker. If a non-Microsoft TPM driver is installed, it may prevent the default TPM driver from loading and cause BitLocker to report that a TPM isn't present on the computer. If you have a non-Microsoft driver installed, remove it, and then allow the operating system to initialize the TPM.

Network connection issues for domain-joined Windows 11 devices

If you have Windows 11, the initialization of the TPM can't complete when your computer has network connection issues and both of the following conditions exist:

- An administrator has configured your computer to require that TPM recovery information be saved in Active Directory Domain Services (AD DS). This requirement can be configured through group policy.
- A domain controller can't be reached. This scenario may occur on a device that is currently disconnected from the internal network, separated from the domain by a firewall, or experiencing a network component failure (such as an unplugged cable or a faulty network adapter).

If these issues occur, an error message appears, and you can't complete the initialization process. To avoid the issue, allow Windows to initialize the TPM while you're connected to the corporate network, and you can contact a domain controller.

Systems with multiple TPMs

Some systems may have multiple TPMs and the active TPM may be toggled in UEFI. Windows doesn't support this configuration. If you switch TPMs, Windows might not properly detect or interact with the new TPM. If you plan to switch TPMs, you should toggle to the new TPM, clear it, and reinstall Windows. For more information, see Clear all the keys from the TPM.

For example, toggling TPMs cause BitLocker to enter recovery mode. We strongly recommend that, on systems with two TPMs, one TPM is selected for use and the selection isn't changed.

Clear all the keys from the TPM

You can use the Windows Defender Security Center app to clear the TPM as a troubleshooting step, or as a final preparation before a clean installation of a new operating system. Preparing for a clean installation in this way helps ensure that the new operating system can fully deploy any TPM-based functionality that it includes, such as attestation. However, even if the TPM isn't cleared before a new operating system is installed, most TPM functionality will probably work correctly.

Clearing the TPM resets it to an unowned state. After you clear the TPM, the Windows operating system will automatically reinitialize it and take ownership again.

> ⚠ **Warning**

Clearing the TPM can result in data loss. For more information, see the next section, "Precautions to take before clearing the TPM."

Precautions to take before clearing the TPM

Clearing the TPM can result in data loss. To protect against such loss, review the following precautions:

- Clearing the TPM causes you to lose all created keys associated with the TPM, and data protected by those keys, such as a virtual smart card or a sign-in PIN. Make sure that you have a backup and recovery method for any data that is protected or encrypted by the TPM.
- Don't clear the TPM on a device you don't own, such as a work or school PC, without being instructed to do so by your IT administrator.
- If you want to temporarily suspend TPM operations on Windows 11, you can turn off the TPM. For more information, see Turn off the TPM.
- Always use functionality in the operating system (such as TPM.msc) to clear the TPM. Don't clear the TPM directly from UEFI.
- Because your TPM security hardware is a physical part of your computer, before clearing the TPM, you might want to read the manuals or instructions that came with your computer, or search the manufacturer's website.

Membership in the local Administrators group, or equivalent, is the minimum required to complete this procedure.

To clear the TPM

1. Open the Windows Defender Security Center app.
2. Select **Device security**.
3. Select **Security processor details**.
4. Select **Security processor troubleshooting**.
5. Select **Clear TPM**.

    - You'll be prompted to restart the computer. During the restart, you might be prompted to press a button to confirm that you wish to clear the TPM.
    - After the device restarts, your TPM will be automatically prepared for use by Windows.

Turn on or turn off the TPM

Normally, the TPM is turned on as part of the TPM initialization process. You don't normally need to turn the TPM on or off. However, if necessary you can do so by using the TPM MMC.

Turn on the TPM

If you want to use the TPM after you've turned it off, you can use the following procedure to turn on the TPM.

1. Open the TPM MMC (tpm.msc).
2. In the **Action** pane, select **Turn TPM On** to display the **Turn on the TPM Security Hardware** page. Read the instructions on this page.
3. Select **Shutdown** (or **Restart**), and then follow the UEFI screen prompts.

After the device restarts, but before you sign in to Windows, you'll be prompted to accept the reconfiguration of the TPM. The acceptance ensures that the user has physical access to the computer and that malicious software isn't attempting to make changes to the TPM.

Turn off the TPM

If you want to stop using the services that are provided by the TPM, you can use the TPM MMC to turn off the TPM.

1. Open the TPM MMC (`tpm.msc`).
2. In the **Action** pane, select **Turn TPM Off** to display the **Turn off the TPM security hardware** page.
3. In the **Turn off the TPM security hardware** dialog box, select a method to enter your owner password and turning off the TPM:

    - If you saved your TPM owner password on a removable storage device, insert it, and then select **I have the owner password file**. In the **Select backup file with the TPM owner password** dialog box, select **Browse** to locate the `.tpm` file that is saved on your removable storage device, select **Open**, and then select **Turn TPM Off**.
    - If you don't have the removable storage device with your saved TPM owner password, select **I want to enter the password**. In the **Type your TPM owner password** dialog box, type your password (including hyphens), and then select **Turn TPM Off**.
    - If you didn't save your TPM owner password or no longer know it, select **I do not have the TPM owner password**, and follow the instructions that are provided in the dialog box and subsequent UEFI screens to turn off the TPM without entering the password.

Use the TPM cmdlets

You can manage the TPM using Windows PowerShell. For details, see TPM Cmdlets in Windows PowerShell.

Last updated on 08/15/2025

PCR banks on TPM 2.0 devices

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

For steps on how to switch PCR banks on TPM 2.0 devices on your PC, you should contact your OEM or UEFI vendor. This article provides background about what happens when you switch PCR banks on TPM 2.0 devices.

A Platform Configuration Register (PCR) is a memory location in the TPM that has some unique properties. The size of the value that can be stored in a PCR is determined by the size of a digest generated by an associated hashing algorithm. A SHA-1 PCR can store 20 bytes - the size of a SHA-1 digest. Multiple PCRs associated with the same hashing algorithm are referred to as a *PCR bank*.

To store a new value in a PCR, the existing value is extended with a new value as follows:
`PCR[N] = HASHalg( PCR[N] || ArgumentOfExtend)`

The existing value is concatenated with the argument of the TPM Extend operation. The resulting concatenation is then used as input to the associated hashing algorithm, which computes a digest of the input. The computed digest becomes the new value of the PCR.

The TCG PC Client Platform TPM Profile Specification defines the inclusion of at least one PCR bank with 24 registers. The only way to reset the first 16 PCRs is to reset the TPM itself. This restriction helps to ensure that the value of those PCRs can only be modified via the TPM Extend operation.

Some TPM PCRs are used as checksums of log events. The log events are extended in the TPM as the events occur. Later, an auditor can validate the logs by computing the expected PCR values from the log and comparing them to the PCR values of the TPM. Since the first 16 TPM PCRs can't be modified arbitrarily, a match between an expected PCR value in that range and the actual TPM PCR value provides assurance of an unmodified log.

How does Windows use PCRs?

To bind the use of a TPM based key to a certain state of the device, the key can be sealed to an expected set of PCR values. For instance, PCRs 0 through 7 have a well-defined value after the boot process, when the OS is loaded. When the hardware, firmware, or boot loader of the machine changes, the change can be detected in the PCR values. Windows uses this capability to make certain cryptographic keys only available at certain times during the boot process. For instance, the BitLocker key can be used at a certain point in the boot, but not before or after.

It's important to note that this binding to PCR values also includes the hashing algorithm used for the PCR. For instance, a key can be bound to a specific value of the `SHA-1 PCR[12]`, if using the SHA-256 PCR bank, even with the same system configuration. Otherwise, the PCR values won't match.

What happens when PCR banks are switched?

When the PCR banks are switched, the algorithm used to compute the hashed values stored in the PCRs during extend operations is changed. Each hash algorithm returns a different cryptographic signature for the same inputs.

As a result, if the currently used PCR bank is switched, all keys that are bound to the previous PCR values no longer work. For example, if you had a key bound to the SHA-1 value of PCR[12] and later changed the PCR bank to SHA-256, the banks wouldn't match, and you would be unable to use that key. The BitLocker key is secured using the PCR banks and Windows can't unseal it if the PCR banks are switched while BitLocker is enabled.

What can I do to switch PCRs when BitLocker is already active?

Before switching PCR banks, you should suspend or disable BitLocker or have the recovery key ready. For steps on how to switch PCR banks on your PC, contact your OEM or UEFI vendor.

How can I identify which PCR bank is being used?

You can configure a TPM to have multiple PCR banks active. When BIOS performs measurements, it does so into all active PCR banks, depending on its capability to make these measurements. BIOS may choose to deactivate PCR banks that it doesn't support or *cap* PCR banks that it doesn't support by extending a separator. The following registry value identifies which PCR banks are active:

- Registry key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\IntegrityServices`
- DWORD: `TPMActivePCRBanks`
- Defines which PCR banks are currently active. This value should be interpreted as a bitmap for which the bits are defined in the TCG Algorithm Registry Table 21 of Revision 1.27.

Windows checks which PCR banks are active and supported by the BIOS. Windows also checks if the measured boot log supports measurements for all active PCR banks. Windows prefers the

use of the SHA-256 bank for measurements and falls back to SHA1 PCR bank if one of the preconditions isn't met.

You can identify which PCR bank is currently used by Windows by looking at the registry:

- Registry key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\IntegrityServices`
- DWORD: `TPMDigestAlgID`
- Algorithm ID of the PCR bank that Windows is currently using. This value represents an algorithm identifier as defined in the TCG Algorithm Registry Table 3 of Revision 1.27.

Windows only uses one PCR bank to continue boot measurements. All other active PCR banks are extended with a separator to indicate that they aren't used by Windows and measurements that appear to be from Windows shouldn't be trusted.

Last updated on 08/15/2025

TPM recommendations

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article provides recommendations for Trusted Platform Module (TPM) technology for Windows.

For a basic feature description of TPM, see the Trusted Platform Module Technology Overview.

TPM design and implementation

Traditionally, TPMs are discrete chips soldered to a computer's motherboard. Such implementations allow the computer's original equipment manufacturer (OEM) to evaluate and certify the TPM separate from the rest of the system. Discrete TPM implementations are common. However, they can be problematic for integrated devices that are small or have low power consumption. Some newer TPM implementations integrate TPM functionality into the same chipset as other platform components while still providing logical separation similar to discrete TPM chips.

TPMs are passive: they receive commands and return responses. To realize the full benefit of a TPM, the OEM must carefully integrate system hardware and firmware with the TPM to send it commands and react to its responses. TPMs were originally designed to provide security and privacy benefits to a platform's owner and users, but newer versions can provide security and privacy benefits to the system hardware itself. Before it can be used for advanced scenarios, however, a TPM must be provisioned. Windows automatically provisions a TPM, but if the user is planning to reinstall the operating system, they may need to clear the TPM before reinstalling so that Windows can take full advantage of the TPM.

The Trusted Computing Group (TCG) is the nonprofit organization that publishes and maintains the TPM specification. The TCG exists to develop, define, and promote vendor-neutral, global industry standards. These standards support a hardware-based root of trust for interoperable trusted computing platforms. The TCG also publishes the TPM specification as the international standard ISO/IEC 11889, using the Publicly Available Specification Submission Process that the Joint Technical Committee 1 defines between the International Organization for Standardization (ISO) and the International Electrotechnical Commission (IEC).

OEMs implement the TPM as a component in a trusted computing platform, such as a PC, tablet, or phone. Trusted computing platforms use the TPM to support privacy and security scenarios that software alone can't achieve. For example, software alone can't reliably report whether malware is present during the system startup process. The close integration between TPM and platform increases the transparency of the startup process and supports evaluating

device health by enabling reliable measuring and reporting of the software that starts the device. Implementation of a TPM as part of a trusted computing platform provides a hardware root of trust-that is, it behaves in a trusted way. For example, if a key stored in a TPM has properties that disallow exporting the key, that key truly can't leave the TPM.

The TCG designed the TPM as a low-cost, mass-market security solution that addresses the requirements of different customer segments. There are variations in the security properties of different TPM implementations just as there are variations in customer and regulatory requirements for different sectors. In public-sector procurement, for example, some governments clearly define security requirements for TPMs whereas others don't.

TPM 1.2 vs. 2.0 comparison

From an industry standard, Microsoft has been an industry leader in moving and standardizing on TPM 2.0, which has many key realized benefits across algorithms, crypto, hierarchy, root keys, authorization, and NV RAM.

Why TPM 2.0?

TPM 2.0 products and systems have important security advantages over TPM 1.2, including:

- The TPM 1.2 spec only allows for the use of RSA and the SHA-1 hashing algorithm.
- For security reasons, some entities are moving away from SHA-1. Notably, NIST requires many federal agencies to move to SHA-256 as of 2014, and technology leaders, including Microsoft and Google have removed support for SHA-1 based signing or certificates in 2017.
- TPM 2.0 **enables greater crypto agility** by being more flexible with respect to cryptographic algorithms.
  - TPM 2.0 supports newer algorithms, which can improve drive signing and key generation performance. For the full list of supported algorithms, see the TCG Algorithm Registry. Some TPMs don't support all algorithms.
  - For the list of algorithms that Windows supports in the platform cryptographic storage provider, see CNG Cryptographic Algorithm Providers.
  - TPM 2.0 achieved ISO standardization (ISO/IEC 11889:2015).
  - Use of TPM 2.0 may help eliminate the need for OEMs to make exception to standard configurations for certain countries and regions.
- TPM 2.0 offers a more **consistent experience** across different implementations.
  - TPM 1.2 implementations vary in policy settings. This may result in support issues as lockout policies vary.
  - TPM 2.0 lockout policy is configured by Windows, ensuring a consistent dictionary attack protection guarantee.

- While TPM 1.2 parts are discrete silicon components, which are typically soldered on the motherboard, TPM 2.0 is available as a **discrete (dTPM)** silicon component in a single semiconductor package, an **integrated** component incorporated in one or more semiconductor packages - alongside other logic units in the same package(s), and as a **firmware (fTPM)** based component running in a trusted execution environment (TEE) on a general purpose SoC.

> ⓘ **Note**
>
> TPM 2.0 is not supported in Legacy and CSM Modes of the BIOS. Devices with TPM 2.0 must have their BIOS mode configured as Native UEFI only. The Legacy and Compatibility Support Module (CSM) options must be disabled. For added security Enable the Secure Boot feature.
>
> Installed Operating System on hardware in legacy mode will stop the OS from booting when the BIOS mode is changed to UEFI. Use the tool MBR2GPT before changing the BIOS mode which will prepare the OS and the disk to support UEFI.

Discrete, Integrated, or Firmware TPM?

There are three implementation options for TPMs:

- Discrete TPM chip as a separate component in its own semiconductor package.
- Integrated TPM solution, using dedicated hardware integrated into one or more semiconductor packages alongside, but logically separate from, other components.
- Firmware TPM solution, running the TPM in firmware in a Trusted Execution mode of a general purpose computation unit.

Windows uses any compatible TPM in the same way. Microsoft doesn't take a position on which way a TPM should be implemented and there's a wide ecosystem of available TPM solutions, which should suit all needs.

Is there any importance for TPM for consumers?

For end consumers, TPM is behind the scenes but is still relevant. TPM is used for Windows Hello, Windows Hello for Business and in the future, will be a component of many other key security features in Windows. TPM secures the PIN, helps encrypt passwords, and builds on our overall Windows experience story for security as a critical pillar. Using Windows on a system with a TPM enables a deeper and broader level of security coverage.

TPM 2.0 Compliance for Windows

Windows for desktop editions (Home, Pro, Enterprise, and Education)

- Since July 28, 2016, all new device models, lines, or series (or if you're updating the hardware configuration of an existing model, line, or series with a major update, such as CPU, graphic cards) must implement and enable by default TPM 2.0 (details in section 3.7 of the Minimum hardware requirements page). The requirement to enable TPM 2.0 only applies to the manufacturing of new devices. For TPM recommendations for specific Windows features, see TPM and Windows Features.

IoT Core

- TPM is optional on IoT Core.

Windows Server 2016

- TPM is optional for Windows Server SKUs unless the SKU meets the other qualification (AQ) criteria for the Host Guardian Services scenario in which case TPM 2.0 is required.

TPM and Windows Features

The following table defines which Windows features require TPM support.

Expand table

Windows Features	TPM Required	Supports TPM 1.2	Supports TPM 2.0	Details
Measured Boot	Yes	Yes	Yes	Measured Boot requires TPM 1.2 or 2.0 and UEFI Secure Boot. TPM 2.0 is recommended since it supports newer cryptographic algorithms. TPM 1.2 only supports the SHA-1 algorithm, which is being deprecated.
BitLocker	No	Yes	Yes	TPM 1.2 or 2.0 are supported but TPM 2.0 is recommended. Device Encryption requires Modern Standby including TPM 2.0 support
Device Encryption	Yes	N/A	Yes	Device Encryption requires Modern Standby/Connected Standby certification,

Windows Features	TPM Required	Supports TPM 1.2	Supports TPM 2.0	Details
				which requires TPM 2.0.
App Control for Business	No	Yes	Yes	
System Guard (DRTM)	Yes	No	Yes	TPM 2.0 and UEFI firmware is required.
Credential Guard	No	Yes	Yes	Windows 10, version 1507 (End of Life as of May 2017) only supported TPM 2.0 for Credential Guard. Beginning with Windows 10, version 1511, TPM 1.2 and 2.0 are supported. Paired with System Guard, TPM 2.0 provides enhanced security for Credential Guard. Windows 11 requires TPM 2.0 by default to facilitate easier enablement of this enhanced security for customers.
Device Health Attestation	Yes	Yes	Yes	TPM 2.0 is recommended since it supports newer cryptographic algorithms. TPM 1.2 only supports the SHA-1 algorithm, which is being deprecated.
Windows Hello/Windows Hello for Business	No	Yes	Yes	Microsoft Entra join supports both versions of TPM, but requires TPM with keyed-hash message authentication code (HMAC) and Endorsement Key (EK) certificate for key attestation support. TPM 2.0 is recommended over TPM 1.2 for better performance and security. Windows Hello as a FIDO platform authenticator takes advantage of TPM 2.0 for key storage.
UEFI Secure Boot	No	Yes	Yes	
TPM Platform Crypto Provider Key Storage Provider	Yes	Yes	Yes	
Virtual Smart Card	Yes	Yes	Yes	
Certificate storage	No	Yes	Yes	TPM is only required when the certificate is stored in the TPM.
Autopilot	No	N/A	Yes	If you intend to deploy a scenario, which requires TPM (such as white glove and self-

Windows Features	TPM Required	Supports TPM 1.2	Supports TPM 2.0	Details
				deploying mode), then TPM 2.0 and UEFI firmware are required.
SecureBIO	Yes	No	Yes	TPM 2.0 and UEFI firmware is required.

OEM Status on TPM 2.0 system availability and certified parts

Government customers and enterprise customers in regulated industries may have acquisition standards that require use of common certified TPM parts. As a result, OEMs, who provide the devices, may be required to use only certified TPM components on their commercial class systems. For more information, contact your OEM or hardware vendor.

Last updated on 08/15/2025

Microsoft Pluton security processor

Applies to: ✅ Windows 11

Microsoft Pluton security processor is a chip-to-cloud security technology built with Zero Trust principles at the core. Microsoft Pluton provides hardware-based root of trust, secure identity, secure attestation, and cryptographic services. Pluton technology combines a secure subsystem integrated into the system-on-chip (SoC) with Microsoft-authored software that runs on it.

Microsoft Pluton is currently available on devices with the following chipsets running on Windows 11:

1. **AMD**: Ryzen™ 6000, 7000, 8000, 9000 and Ryzen™ AI Series processors
2. **Intel®:** Core™ Series Processors - Ultra 200V Series, Ultra Series 3 and Series 3 processors
3. **Qualcomm**: Snapdragon® 8cx Gen 3 and Snapdragon® X Series processors

What is Microsoft Pluton?

Designed by Microsoft and built by silicon partners, Microsoft Pluton is a secure crypto-processor built into the CPU for security at the core to ensure code integrity and the latest protection with updates delivered by Microsoft through Windows Update. Pluton protects credentials, identities, personal data, and encryption keys. Information is significantly harder to be removed even if an attacker installs malware or has complete physical possession of the PC.

Microsoft Pluton is designed to provide the functionality of the Trusted Platform Module (TPM) and deliver other security functionality beyond what is possible with the TPM 2.0 specification, and allows for other Pluton firmware and OS features to be delivered over time via Windows Update. For more information, see Microsoft Pluton as TPM.

Pluton is built on proven technology used in Xbox and Azure Sphere, and provides hardened integrated security capabilities to Windows 11 devices in collaboration with leading silicon partners. For more information, see Meet the Microsoft Pluton processor – The security chip designed for the future of Windows PCs.

How can Pluton help customers?

Pluton is built with the goal of providing customers with better end-to-end security experiences. It does so by doing three things:

1. **Zero-trust security and reliability**: Customer security scenarios often span devices and cloud services. Windows PCs and services like Microsoft Entra and Intune need to work harmoniously together to provide frictionless security. Pluton is designed, built, and

maintained in close collaboration with teams across Microsoft to ensure that customers get both high security and reliability.
2. **Innovation**: Pluton platform and the functionality it provides is informed by customer feedback and Microsoft's threat intelligence. As an example, Pluton platforms in 2024 AMD and Intel systems will start to use a Rust-based firmware foundation given the importance of memory safety.
3. **Continuous improvement**: Pluton platform supports loading new firmware delivered through operating system updates. This functionality is supported alongside the typical mechanism of UEFI capsule updates that update the Pluton firmware that is resident on the system's SPI flash and loaded during early system boot. The additional support for dynamically loading valid new Pluton firmware through operating system updates facilitates continuous improvements both for bug fixes and new features.

A practical example: zero-trust security with device-based conditional access policies

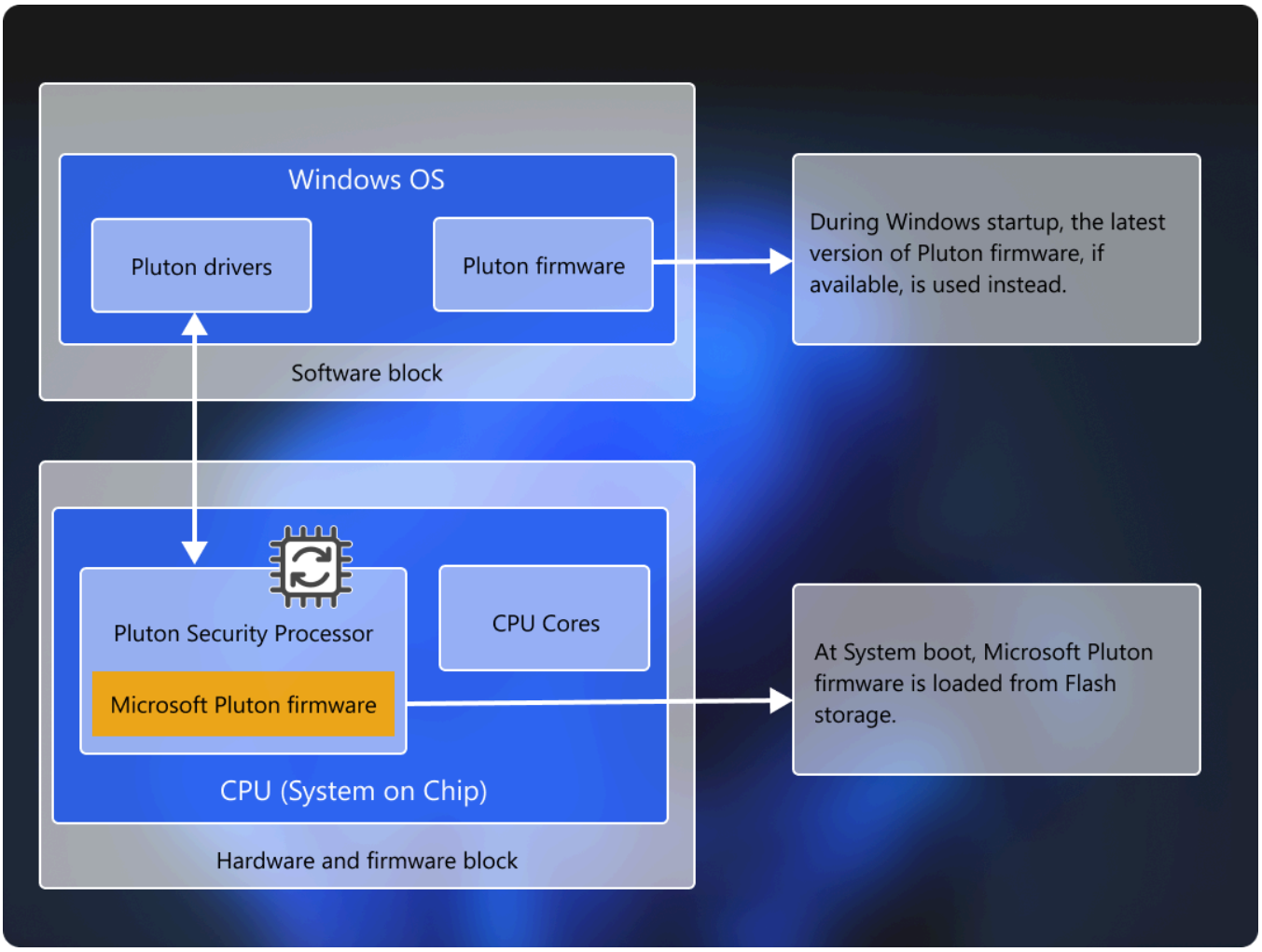
An increasingly important zero-trust workflow is conditional access – gating access to resources like Sharepoint documents based on verifying whether requests are coming from a valid, healthy source. Microsoft Intune, for example, supports different workflows for conditional access including device-based conditional access which allows organizations to set policies that ensure that managed devices are healthy and compliant before granting access to the organization's apps and services.

To ensure that Intune gets an accurate picture about the device's health as part of enforcing these policies, ideally it has tamper-resistant logs on the state of the relevant security capabilities. This is where hardware security is critical as any malicious software running on the device could attempt to provide false signals to the service. One of the core benefits of a hardware security technology like the TPM, is that it has a tamper-resistant log of the state of the system. Services can cryptographically validate that logs and the associated system state reported by the TPM truly come from the TPM.

For the end-to-end scenario to be truly successful at scale, the hardware-based security isn't enough. Since access to enterprise assets is being gated based on security settings that are being reported by the TPM logs, it's critical that these logs are available reliably. Zero-trust security essentially requires high reliability.

With Pluton, when it's configured as the TPM for the system, customers using conditional access get the benefits of Pluton's security architecture and implementation with the reliability that comes from the tight integration and collaboration between Pluton and other Microsoft components and services.

Microsoft Pluton security architecture overview



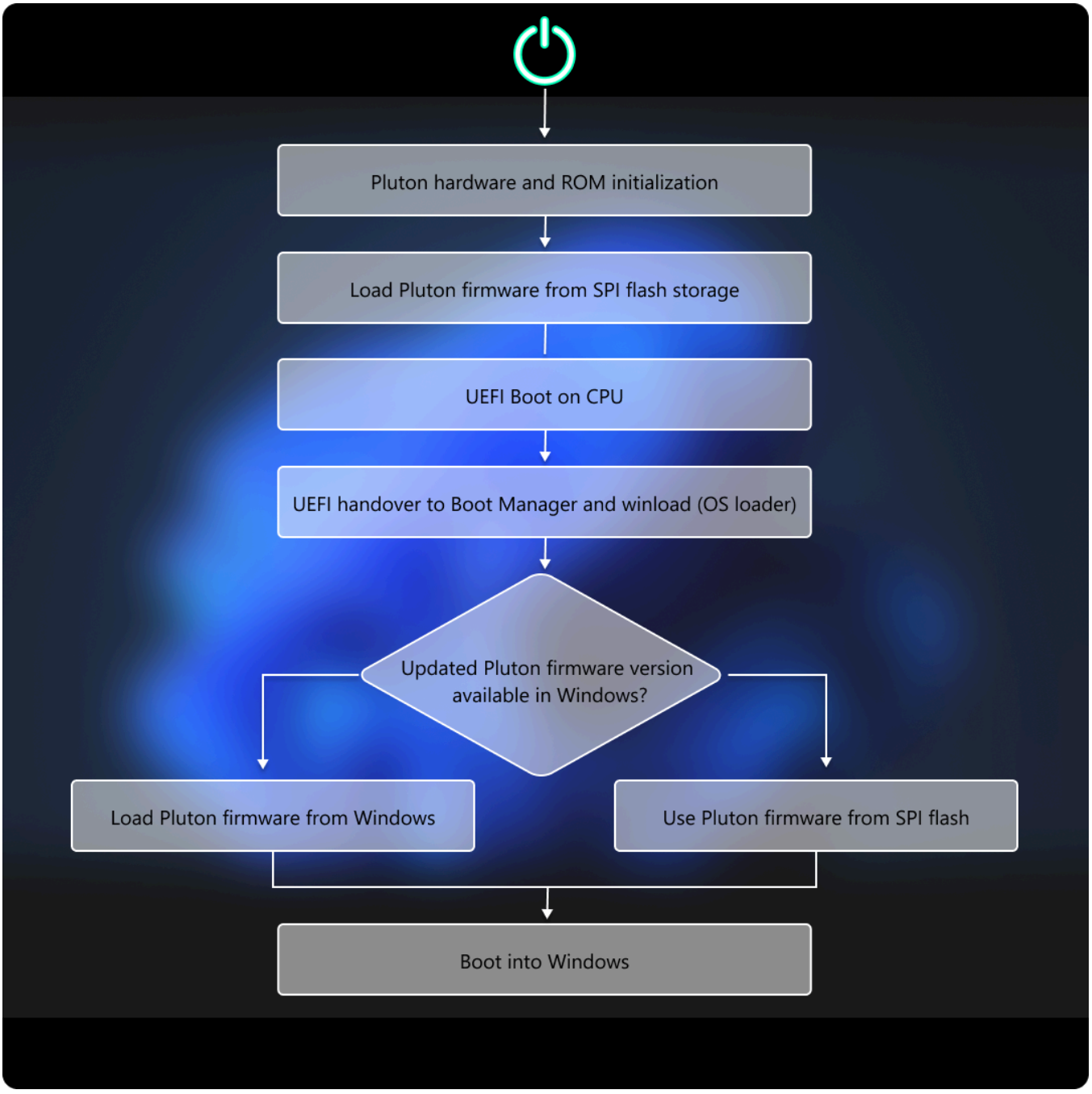
Pluton Security subsystem consists of the following layers:

Expand table

	Description
Hardware	Pluton Security Processor is a secure element tightly integrated into the SoC subsystem. It provides a trusted execution environment while delivering cryptographic services required for protecting sensitive resources and critical items like keys, data, etc.
Firmware	Microsoft authorized firmware provides required secure features and functionality, and exposes interfaces that operating system software and applications can use to interact with Pluton. The firmware is stored in the flash storage available on the motherboard. When the system boots, the firmware is loaded as a part of Pluton Hardware initialization. During Windows startup, a copy of this firmware (or the latest firmware obtained from Windows Update, if available) is loaded in the operating system. For more information, see Firmware load flow
Software	Operating system drivers and applications available to an end user to allow seamless usage of the hardware capabilities provided by the Pluton security subsystem.

Firmware load flow

When the system boots, Pluton hardware initialization is performed by loading the Pluton firmware from the Serial Peripheral Interface (SPI) flash storage available on the motherboard. During Windows startup however, the latest version of the Pluton firmware is used by the operating system. If newer firmware isn't available, Windows uses the firmware that was loaded during the hardware initialization. This diagram illustrates this process:



Windows edition and licensing requirements

The following table lists the Windows editions that support Microsoft Pluton:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Microsoft Pluton license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Related articles

Microsoft Pluton as TPM

Last updated on 01/30/2026

Microsoft Pluton as Trusted Platform Module

Applies to: ✅ Windows 11

Microsoft Pluton is designed to provide the functionality of the Trusted Platform Module (TPM) thereby establishing the silicon root of trust. Microsoft Pluton supports the TPM 2.0 industry standard allowing customers to immediately benefit from the enhanced security in Windows features that rely on TPM including BitLocker, Windows Hello, and System Guard.

As with other TPMs, credentials, encryption keys, and other sensitive information can't be easily extracted from Pluton even if an attacker installs malware or has complete physical possession of the device. Storing sensitive data like encryption keys securely within the Pluton processor, which is isolated from the rest of the system, helps ensure that emerging attack techniques such as speculative execution can't access key material.

Pluton also solves the major security challenge of keeping its own root-of-trust firmware up to date across the entire PC ecosystem, by delivering firmware updates from Windows Update. Today customers receive updates to their security firmware from various sources, which can make it difficult for them to apply these updates.

To learn more about the TPM related scenarios that benefit from Pluton, see TPM and Windows Features.

Microsoft Pluton as a security processor alongside discrete TPM

Microsoft Pluton can be used as a TPM, or with a TPM. Although Pluton builds security directly into the CPU, device manufacturers might choose to use discrete TPM as the default TPM, while having Pluton available to the system as a security processor for use cases beyond the TPM.

Pluton is integrated within the SoC subsystem, and provides a flexible, updatable platform for running firmware that implements end-to-end security functionality authored, maintained, and updated by Microsoft.

Related articles

Microsoft Pluton security processor

Last updated on 04/15/2025

Virtualization-based Security (VBS)

Article • 02/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

Virtualization-based security, or VBS, uses hardware virtualization and the Windows hypervisor to create an isolated virtual environment that becomes the root of trust of the OS that assumes the kernel can be compromised. Windows uses this isolated environment to host a number of security solutions, providing them with greatly increased protection from vulnerabilities in the operating system, and preventing the use of malicious exploits which attempt to defeat protections. VBS enforces restrictions to protect vital system and operating system resources, or to protect security assets such as authenticated user credentials.

One such example security solution is memory integrity, which protects and hardens Windows by running kernel mode code integrity within the isolated virtual environment of VBS. Kernel mode code integrity is the Windows process that checks all kernel mode drivers and binaries before they're started, and prevents unsigned or untrusted drivers or system files from being loaded into system memory. Memory integrity also restricts kernel memory allocations that could be used to compromise the system, ensuring that kernel memory pages are only made executable after passing code integrity checks inside the secure runtime environment, and executable pages themselves are never writable. That way, even if there are vulnerabilities like a buffer overflow that allow malware to attempt to modify memory, executable code pages cannot be modified, and modified memory cannot be made executable.

> **Note**
>
> Memory integrity is sometimes referred to as *hypervisor-protected code integrity (HVCI)* or *hypervisor enforced code integrity*, and was originally released as part of *Device Guard*. Device Guard is no longer used except to locate memory integrity and VBS settings in Group Policy or the Windows registry.

VBS requires the following components be present and properly configured.

Expand table

Hardware requirement	Details
64-bit CPU	Virtualization-based security (VBS) requires the Windows hypervisor, which is only supported on 64-bit IA processors with virtualization extensions, including Intel VT-X and AMD-v.

Hardware requirement	Details
Second Level Address Translation (SLAT)	VBS also requires that the processor's virtualization support includes Second Level Address Translation (SLAT), either Intel VT-X2 with Extended Page Tables (EPT), or AMD-v with Rapid Virtualization Indexing (RVI).
IOMMUs or SMMUs (Intel VT-D, AMD-Vi, Arm64 SMMUs)	All I/O devices capable of DMA must be behind an IOMMU or SMMU. An IOMMU can be used to enhance system resiliency against memory attacks.
Trusted Platform Module (TPM) 2.0	For more information, see Trusted Platform Module (TPM) 2.0.
Firmware support for SMM protection	System firmware must adhere to the recommendations for hardening SMM code described in the Windows SMM Security Mitigations Table (WMST) specification. The WSMT specification contains details of an ACPI table that was created for use with Windows operating systems that support VBS features. Firmware must implement the protections described in the WSMT specification, and set the corresponding protection flags as described in the specification to report compliance with these requirements to the operating system.
Unified Extensible Firmware Interface (UEFI) Memory Reporting	UEFI firmware must adhere to the following memory map reporting format and memory allocation guidelines in order for firmware to ensure compatibility with VBS. • **UEFI v2.6 Memory Attributes Table (MAT)** - To ensure compatibility with VBS, firmware must cleanly separate EFI runtime memory ranges for code and data, and report this to the operating system. Proper segregation and reporting of EFI runtime memory ranges allows VBS to apply the necessary page protections to EFI runtime services code pages within the VBS secure region. Conveying this information to the OS is accomplished using the EFI_MEMORY_ATTRIBUTES_TABLE. To implement the UEFI MAT, follow these guidelines: 1. The entire EFI runtime must be described by this table. 2. All appropriate attributes for EfiRuntimeServicesData and EfiRuntimeServicesCode pages must be marked. 3. These ranges must be aligned on page boundaries (4KB), and can not overlap. • **EFI Page Protections** -All entries must include attributes EFI_MEMORY_RO, EFI_MEMORY_XP, or both. All UEFI memory that is marked executable must be read only. Memory marked writable must not be executable. Entries may not be left with neither of the attributes set, indicating memory that is both executable and writable.

Hardware requirement	Details
Secure Memory Overwrite Request (MOR) revision 2	Secure MOR v2 is enhanced to protect the MOR lock setting using a UEFI secure variable. This helps guard against advanced memory attacks. For details, see Secure MOR implementation.
Memory integrity-compatible drivers	Ensure all system drivers have been tested and verified to be compatible with memory integrity. The Windows Driver Kit and Driver Verifier contain tests for driver compatibility with memory integrity. There are three steps to verify driver compatibility: 1. Use Driver Verifier with the **Code Integrity compatibility** checks enabled. 2. Run the Hypervisor Code Integrity Readiness Test in the Windows HLK. 3. Test the driver on a system with VBS and memory integrity enabled. This step is imperative to validate the driver's behavior with memory integrity, as static code analysis tools simply aren't capable of detecting all memory integrity violations possible at runtime.
Secure Boot	Secure Boot must be enabled on devices leveraging VBS. For more information, see Secure Boot

VBS works on VMs that have either nested virtualization support or **Guest VSM** enabled. The latter is enabled by default for Generation 2 VMs on Hyper-V. This also includes all Generation 2 VMs on Microsoft Azure, and Generation 1 VMs that have nested virtualization enabled. A list of supported Azure VM series is detailed in the table below.

Expand table

VM Series Name	Nested Virtualization	VM Generation
Av2	Yes	1 (certain internal sizes support generation 2)
B	No	1 and 2
Dsv2/Dv2/Dv3/Ev3	Yes	1
Dsv3/Ddsv3	Yes	1 and 2
Dsv4/Ddsv4	Yes	1 and 2
Esv3/Edsv3	Yes	1 and 2
Esv4/Edsv4	Yes	1 and 2

VM Series Name	Nested Virtualization	VM Generation
Ev4/Edv4	Yes	Ev4 - 1 only Edv4 -1&2
Dv4/Ddv4	Yes	1 and 2
Dv5/Ddv5/Dsv5/Ddsv5	Yes	1 and 2
Ev5/Edv5/Esv5/Edsv5	Yes	1 and 2
Dasv5/Dadsv5/Easv5/ Eadsv5	Yes	1 and 2
Ebsv5/Edbsv5	Yes	1 and 2
Fsv2	Yes	1 and 2
Fx	Yes	2
Lsv2	Yes	1 and 2

Related topics

For more info about Hyper-V, see Hyper-V on Windows Server 2016 or Introduction to Hyper-V on Windows 10. For more info about hypervisor, see Hypervisor Specifications.

Enable virtualization-based protection of code integrity

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> ⚠ **Warning**
>
> Some applications and hardware device drivers may be incompatible with memory integrity. This incompatibility can cause devices or software to malfunction and in rare cases may result in a boot failure (blue screen). Such issues may occur after memory integrity has been turned on or during the enablement process itself. If compatibility issues occur, see Troubleshooting for remediation steps.

Memory integrity is a Virtualization-based security (VBS) feature available in Windows. Memory integrity and VBS improve the threat model of Windows and provide stronger protections against malware trying to exploit the Windows kernel. VBS uses the Windows hypervisor to create an isolated virtual environment that becomes the root of trust of the OS that assumes the kernel can be compromised. Memory integrity is a critical component that protects and hardens Windows by running kernel mode code integrity within the isolated virtual environment of VBS. Memory integrity also restricts kernel memory allocations that could be used to compromise the system.

> ⓘ **Note**
>
> - Memory integrity is sometimes referred to as *hypervisor-protected code integrity (HVCI)* or *hypervisor enforced code integrity*, and was originally released as part of *Device Guard*. Device Guard is no longer used except to locate memory integrity and VBS settings in Group Policy or the Windows registry.
> - Memory integrity works better with Intel Kabylake and higher processors with *Mode-Based Execution Control*, and AMD Zen 2 and higher processors with *Guest Mode Execute Trap* capabilities. Older processors rely on an emulation of these features, called *Restricted User Mode*, and will have a bigger impact on performance. When nested virtualization is enabled, memory integrity works better when the VM is version >= 9.3.

- Azure VMs do not support memory integrity where **Secure Boot with DMA** is selected. If this is selected, VBS will show as enabled but not running. For this reason, please make sure to choose **Secure Boot** only using one of the methods below.

Memory integrity features

- Protects modification of the Control Flow Guard (CFG) bitmap for kernel mode drivers.
- Protects the kernel mode code integrity process that ensures that other trusted kernel processes have a valid certificate.

How to turn on memory integrity

To enable memory integrity on Windows devices with supporting hardware throughout an enterprise, use any of these options:

Windows Security

Enable memory integrity using Windows Security

Memory integrity can be turned on in **Windows Security** settings and found at **Windows Security** > **Device security** > **Core isolation details** > **Memory integrity**. For more information, see Device protection in Windows Security.

Beginning with Windows 11 22H2, **Windows Security** shows a warning if memory integrity is turned off. The warning indicator also appears on the Windows Security icon in the Windows Taskbar and in the Windows Notification Center. The user can dismiss the warning from within **Windows Security**.

Validate enabled VBS and memory integrity features

Use Win32_DeviceGuard WMI class

Windows 10, Windows 11, and Windows Server 2016 and higher have a WMI class for VBS-related properties and features: **Win32_DeviceGuard**. This class can be queried from an elevated Windows PowerShell session by using the following command:

PowerShell

```
Get-CimInstance -ClassName Win32_DeviceGuard -Namespace
root\Microsoft\Windows\DeviceGuard
```

> **Note**
>
> Mode Based Execution Control property will only be listed as available starting with Windows 10 version 1803 and Windows 11 version 21H2. This value is reported for both Intel's *Mode-Based Execution Control* and AMD's *Guest Mode Execute Trap* capabilities.

The output of this command provides details of the available hardware-based security features and those features that are currently enabled.

- **InstanceIdentifier**: A string that is unique to a particular device and set by WMI.
- **Version**: This field lists the version of this WMI class. The only valid value now is **1.0**.
- **AvailableSecurityProperties**: This field helps to enumerate and report state on the relevant security properties for VBS and memory integrity.

Expand table

Value	Description
0	If present, no relevant properties exist on the device.
1	If present, hypervisor support is available.
2	If present, Secure Boot is available.
3	If present, DMA protection is available.
4	If present, Secure Memory Overwrite is available.
5	If present, NX protections are available.
6	If present, SMM mitigations are available.
7	If present, MBEC/GMET is available.
8	If present, APIC virtualization is available.

- **CodeIntegrityPolicyEnforcementStatus**: This field indicates the code integrity policy enforcement status.

Expand table

Value	Description
0	Off
1	Audit.
2	Enforced.

- **RequiredSecurityProperties**: This field describes the required security properties to enable VBS.

Expand table

Value	Description
0	Nothing is required.
1	If present, hypervisor support is needed.
2	If present, Secure Boot is needed.
3	If present, DMA protection is needed.
4	If present, Secure Memory Overwrite is needed.
5	If present, NX protections are needed.
6	If present, SMM mitigations are needed.
7	If present, MBEC/GMET is needed.

- **SecurityServicesConfigured**: This field indicates whether Credential Guard or memory integrity is configured.

Expand table

Value	Description
0	No services are configured.
1	If present, Credential Guard is configured.
2	If present, memory integrity is configured.
3	If present, System Guard Secure Launch is configured.
4	If present, SMM Firmware Measurement is configured.
5	If present, Kernel-mode Hardware-enforced Stack Protection is configured.

Value	Description
6	If present, Kernel-mode Hardware-enforced Stack Protection is configured in Audit mode.
7	If present, Hypervisor-Enforced Paging Translation is configured.

- **SecurityServicesRunning**: This field indicates whether Credential Guard or memory integrity is running.

Expand table

Value	Description
0	No services running.
1	If present, Credential Guard is running.
2	If present, memory integrity is running.
3	If present, System Guard Secure Launch is running.
4	If present, SMM Firmware Measurement is running.
5	If present, Kernel-mode Hardware-enforced Stack Protection is running.
6	If present, Kernel-mode Hardware-enforced Stack Protection is running in Audit mode.
7	If present, Hypervisor-Enforced Paging Translation is running.

- **SmmIsolationLevel**: This field indicates the SMM isolation level.
- **UsermodeCodeIntegrityPolicyEnforcementStatus**: This field indicates the user mode code integrity policy enforcement status.

Expand table

Value	Description
0	Off
1	Audit.
2	Enforced.

- **VirtualizationBasedSecurityStatus**: This field indicates whether VBS is enabled and running.

Expand table

Value	Description
0	VBS isn't enabled.
1	VBS is enabled but not running.
2	VBS is enabled and running.

- **VirtualMachineIsolation**: This field indicates whether virtual machine isolation is enabled.
- **VirtualMachineIsolationProperties**: This field indicates the set of virtual machine isolation properties that are available.

Expand table

Value	Description
1	AMD SEV-SNP
2	Virtualization-based Security
3	Intel TDX

Use msinfo32.exe

Another method to determine the available and enabled VBS features is to run msinfo32.exe from an elevated PowerShell session. When you run this program, the VBS features are displayed at the bottom of the **System Summary** section.



Troubleshooting

- If a device driver fails to load or crashes at runtime, you might be able to update the driver using **Device Manager**.
- If you experience a critical error during boot or your system is unstable after turning on memory integrity, you can recover using the Windows Recovery Environment (Windows RE).

    1. First, disable any policies that are used to enable VBS and memory integrity, for example Group Policy.

2. Then, boot to Windows RE on the affected computer, see Windows RE Technical Reference.

3. After logging in to Windows RE, set the memory integrity registry key to off:

   Windows Command Prompt

   ```
   reg add
   "HKLM\SYSTEM\CurrentControlSet\Control\DeviceGuard\Scenarios\HypervisorEnf
   orcedCodeIntegrity" /v "Enabled" /t REG_DWORD /d 0 /f
   ```

4. Finally, restart your device.

> ⓘ **Note**
>
> If you turned on memory integrity with UEFI lock, you will need to disable Secure Boot to complete the Windows RE recovery steps.

Memory integrity deployment in virtual machines

Memory integrity can protect a Hyper-V virtual machine, just as it would a physical machine. The steps to enable memory integrity are the same from within the virtual machine.

Memory integrity protects against malware running in the guest virtual machine. It doesn't provide extra protection from the host administrator. From the host, you can disable memory integrity for a virtual machine:

PowerShell

```
Set-VMSecurity -VMName <VMName> -VirtualizationBasedSecurityOptOut $true
```

Requirements for running memory integrity in Hyper-V virtual machines

- The Hyper-V host must run at least Windows Server 2016 or Windows 10 version 1607.
- The Hyper-V virtual machine must be Generation 2, and running at least Windows Server 2016 or Windows 10.
- Memory integrity and nested virtualization can be enabled at the same time. To enable the Hyper-V role on the virtual machine, you must first install the Hyper-V role in a Windows nested virtualization environment.
- Virtual Fibre Channel adapters aren't compatible with memory integrity. Before attaching a virtual Fibre Channel Adapter to a virtual machine, you must first opt out of

Virtualization-based security using `Set-VMSecurity`.

- The AllowFullSCSICommandSet option for pass-through disks isn't compatible with memory integrity. Before configuring a pass-through disk with AllowFullSCSICommandSet, you must first opt out of Virtualization-based security using `Set-VMSecurity`.

Last updated on 08/15/2025

Memory integrity and VBS enablement

Memory integrity is a virtualization-based security (VBS) feature available in Windows 10, Windows 11, and Windows Server 2016 or higher. Memory integrity and VBS improve the threat model of Windows and provide stronger protections against malware trying to exploit the Windows kernel. VBS uses the Windows hypervisor to create an isolated virtual environment that becomes the root of trust of the OS that assumes the kernel can be compromised. Memory integrity is a critical component that protects and hardens Windows by running kernel mode code integrity within the isolated virtual environment of VBS. Memory integrity also restricts kernel memory allocations that could be used to compromise the system, ensuring that kernel memory pages are only made executable after passing code integrity checks inside the secure runtime environment, and executable pages themselves are never writable.

> **Note**
>
> Memory integrity is sometimes referred to as *hypervisor-protected code integrity (HVCI)* or *hypervisor enforced code integrity*, and was originally released as part of *Device Guard*. Device Guard is no longer used except to locate memory integrity and VBS settings in Group Policy or the Windows registry.

See Virtualization Based Security System Resource Protections for more details on these protections.

Default enablement

Memory integrity is turned on by default on clean installs of Windows 11, and previously only on clean installs of Windows 10 in S mode, on compatible hardware as described in this article. It's also turned on by default on all Secured-core PCs. On other systems that don't meet the memory integrity auto-enablement requirements, customers can opt in using any of the methods described in how to enable memory integrity. IT Pros and end users always have the final control of whether memory integrity is enabled.

Hardware features for automatic enablement

Memory integrity is turned on by default when a PC meets the following minimum hardware features:

Expand table

Component	Detail
Processor	• Intel 8th generation or later starting with Windows 11, version 22H2 (11th generation Core processors and newer only for Windows 11, version 21H2) • AMD Zen 2 architecture and newer • Qualcomm Snapdragon 8180 and newer
RAM	Minimum 8GB (Only applicable for x64 processors)

Component	Detail
Storage	SSD with a minimum size of 64GB
Drivers	Memory integrity-compatible drivers must be installed. See Driver compatibility with memory integrity for more information about drivers.
BIOS	Virtualization must be enabled

If you're building an image that won't automatically enable memory integrity, you can still configure your image so that it's turned on by default.

> **Note**
>
> Auto-enablement pertains only to clean installs, not upgrades of existing devices.

> **Note**
>
> Intel 11th generation Core desktop processors are not included in current default enablement logic. However, they are a recommended platform for memory integrity and it can be enabled by the OEM.

Memory integrity and VBS controls

This section enumerates how device manufacturers and end users can interact with memory integrity and VBS. To learn about how to control memory integrity state as an administrator, see How to turn on memory integrity.

Turn on memory integrity

Although Windows will turn memory integrity on by default for most systems, there are several reasons that may prevent that from happening. As an OEM, you can ensure that memory integrity is turned on for your devices by configuring registry keys in the OS image.

Recommended configuration

Set the following two registry keys in your image to ensure memory integrity is turned on.

Expand table

Registry key	Value
HKLM\System\CurrentControlSet\Control\DeviceGuard\Scenarios\HypervisorEnforcedCodeIntegrity	Enabled=1
HKLM\System\CurrentControlSet\Control\DeviceGuard\Scenarios\HypervisorEnforcedCodeIntegrity	WasEnabledBy=1

Registry key	Value
HKLM\System\CurrentControlSet\Control\DeviceGuard\Scenarios\HypervisorEnforcedCodeIntegrity	EnabledBootId=<Current BootId>

The BootId is a counter that increments on each successful boot and can be found at the registry key: HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\Memory Management\PrefetchParameters\BootId The *WasEnabledBy* and *EnabledBootId* registry keys control a setting that safeguards against having an unbootable device. When set, the device will automatically turn off memory integrity if the system crashes during boot, potentially caused by memory integrity blocking an incompatible boot-critical driver. This auto-disablement feature is only available while BootId is less than EnabledBootId + 3. In some versions of Windows, the auto-disable functionality is designed to revert if the boot failures continue even after memory integrity was turned off, indicating that memory integrity was not the root cause of the failures.

> ⓘ **Note**
>
> For high security systems, *WasEnabledBy* and *EnabledBootId* should NOT be set.

Troubleshooting

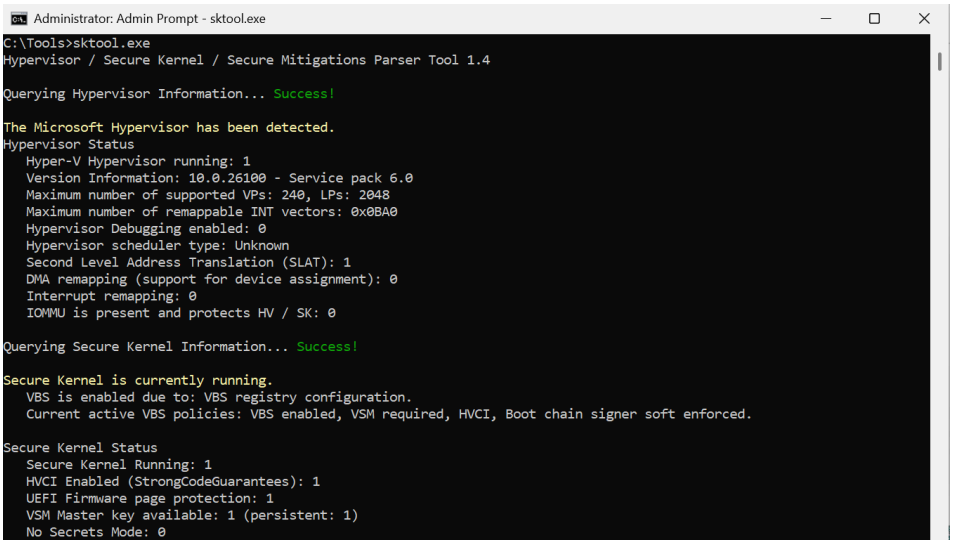
Using SkTool to check the Hypervisor and VBS state

The latest Windows SDK include SkTool, a useful utility for troubleshooting any kind of issue related to Virtualization-Based Security (VBS) and the Hypervisor, or simply for inspecting their current state. The SkTool.exe binary is located in the **bin** folder of the Windows SDK. The typical path is:

```
"C:\Program Files (x86)\Windows Kits\10\bin\10.0.26100.0\<Arch>
```

where corresponds to your system architecture (e.g., x64, arm64).

When run without any command-line switches, SkTool displays the current state of the Hypervisor and VBS, providing also the motivation why VBS has been started or **not** started:



The tool supports also other different command line switches, the most important being the following:

- */status* - Display Hypervisor and Secure Kernel information (default command line switch).
- */lkey* - Display the VSM master key (LKey) provisioning status.
- */mitigations* - Display NT and Secure Kernel mitigations information.

In particular, the "*/lkey*" switch allows you to troubleshoot problems with Enhanced Sign-In Security (ESS), such as Windows Hello, in situations where it refuses to accept the current user's PIN or the Face identification.

For example, here is what the tool reports when the VSM master key fails to unseal because something in the overall system's security posture (measured by the TPM) has changed (in this case, Secure Boot has been disabled):

```
C:\Tools>sktool.exe /lkey
Hypervisor / Secure Kernel / Secure Mitigations Parser Tool 1.4

VBS Master Key Provisioning information
Found UEFI Master Key provisioning information.
   The system has tried to unseal the VBS master key with Status 0xC0000225.
   A New VSM Master key has been generated (status 0x0) and stored in a UEFI variable (in
plaintext).
```

Identifying memory integrity state

The following volatile regkey reflects the state of memory integrity:

Expand table

Registry key	Value
HKLM\System\CurrentControlSet\Control\CI\State	HVCIEnabled

Other ways of checking memory integrity state are to look at MsInfo32 under **Virtualization-based Security Services Running** or refer to the Core isolation settings page in the Windows Security app to see the value of Memory integrity. There is also a WMI interface for checking using management tools, see Validate enabled VBS and memory integrity features.

Debugging Driver Issues

Check the Code Integrity logs to see if any drivers were blocked from loading as a result of memory integrity. These are in Event Viewer under the following path:

`Applications and Service Logs\Microsoft\Windows\CodeIntegrity\Operational`

Generally, memory integrity compatibility events have EventID=3087

Check results of memory integrity default enablement

To see details on the results of memory integrity default enablement, check the *setupact.log* and search for `HVCI`. You should see one of the following result logs, as well as the succeeding/failing checks leading to the enablement decision:

Memory integrity enabled: `SYSPRP HVCI: Enabling HVCI`

Memory integrity not enabled: `SYSPRP HVCI: OS does not meet HVCI auto-enablement requirements. Exiting now.`

If the device was opted out of memory integrity enablement via the regkey method detailed earlier, then this will be the only log from memory integrity's sysprep. If the device had a compatibility issue, it should be identified in the preceding logs with the error message:

`SYSPRP HVCI: Compatibility did not pass. VBS_COMPAT_ISSUES 0xXXXXXXXX`

The following is an enumeration of the potential VBS or memory integrity Compat Issues. Each issue is represented by a single index in a bit array, and the error message outputs the hex value resulting from each error bit being present.

Expand table

Bit Index	Compat Issue	Hex Value	Architecture
0	Unsupported architecture (eg. x86)	0x00000001	
1	SLAT required	0x00000002	x64
2	Secure Boot capability required	0x00000004	x64
3	IOMMU required	0x00000008	x64
4	MBEC/GMET Required	0x00000010	x64

Bit Index	Compat Issue	Hex Value	Architecture
5	UEFI Required	0x00000020	x64
6	UEFI WX Memory Attributes Table required	0x00000040	x64
7	ACPI WSMT table required	0x00000080	x64
8	UEFI MOR Lock required	0x00000100	x64
9	Deprecated		
10	Hardware virtualization required	0x00000400	x64
11	Secure Launch required	0x00000800	ARM64
12	Deprecated		
13	Device failing to meet 64GB minimum required volume size	0x00002000	x64, ARM64
14	System drive SSD required	0x00004000	x64, ARM64
15	Intel CET Required (Only applicable on W11 21H2)	0x00008000	x64
16	ARM SoC is not compatible with VBS	0x00010000	ARM64
17	8GB RAM required	0x00020000	x64

An example of an error code and error identification: `VBS_COMPAT_ISSUES 0x000000C0`

0x000000C0 -> 0x00000080 AND 0x00000040 -> UEFI WX Memory Attributes Table required, ACPI WSMT table required

---

Last updated on 10/28/2025

Windows 11 Secured-core PCs

Article • 05/06/2024

Microsoft works closely with OEM partners to help ensure that all certified Windows systems deliver a secure operating environment. Windows integrates closely with the hardware to deliver protections that take advantage of available hardware capabilities:

- Baseline Windows security – recommended baseline for all individual systems that provides foundational system integrity protections. Leverages TPM 2.0 for a hardware root of trust, secure boot and BitLocker drive encryption.
- Virtualization-based security enabled – leverages virtualization capabilities from hardware and the hypervisor to provide additional protection for critical subsystems and data.
- Secured-core – recommended for the most sensitive systems and industries like financial, healthcare, and government agencies. Builds on the previous layers and leverages advanced processor capabilities to provide protection from firmware attacks.

Secured-core PCs

Microsoft is working closely with OEM partners and silicon vendors to build Secured-core PCs that features deeply integrated hardware, firmware and software to ensure enhanced security for devices, identities and data.

Secured-core PCs provide protections that are useful against sophisticated attacks and can provide increased assurance when handling mission-critical data in some of the most data-sensitive industries, such as healthcare workers that handle medical records and other personally identifiable information (PII), commercial roles that handle high business impact and highly sensitive data, such as a financial controller with earnings data.

For general purpose laptops, tablets, 2-in-1's, mobile workstations, and desktops, Microsoft recommends using Security baselines for optimal configuration. For more info, see Windows security baselines.

Baseline Windows security is supported by Secure Boot, Bitlocker device encryption, Microsoft Defender, Windows Hello and a TPM 2.0 chip to provide a hardware root of trust for the OS platform. These features are designed to secure general purpose modern devices. If you are a decision maker purchasing new devices, your devices should meet the baseline Windows security requirements.

What makes a Secured-core PC

Expand table

Benefit	Feature	Hardware/Firmware requirement	Baseline Windows Security	Secured-core PCs
Create a hardware backed root of trust				
Secure Boot	Secure Boot is enabled in the BIOS by default.	✅	✅	✅
Secure Boot	3rd party UEFI CA not trusted by default, with BIOS option for enabling trust			✅
Trusted Platform Module 2.0 (TPM)	Meet the latest Microsoft requirements for the Trusted Computing Group (TCG) specification	✅	✅	✅
Direct Memory Access (DMA) Protection	The device supports Memory Access Protection (Kernel DMA Protection)	✅	✅	✅
Defend against firmware level attacks (either of the 2 approaches specified can be used)	System Guard Secure Launch (D-RTM) with System Management Mode (SMM) isolation	Enabled on device (via Secure Launch)	✅	✅
S-RTM and Standalone MM with MM supervisor (the approach	Supported on devices that have the FASR firmware		✅	

Benefit	Feature	Hardware/Firmware requirement	Baseline Windows Security	Secured-core PCs
implemented on FASR devices)				
Protect the OS from execution of unverified code	Hypervisor Code Integrity (HVCI)	Enabled on device	✅	✅
Provide advanced identity verification and protection	Windows Hello with Enhanced Sign-in Security (ESS)	A device with *Windows Hello with ESS* is enabled if it has the ESS hardware built-in components for face or fingerprint authentication, and the necessary support in BIOS	✅*	✅
Protect critical data if a device is lost, stolen or confiscated	BitLocker encryption	BitLocker can leverage the TPM 2.0 to encrypt and protect data	✅	

*Only possible on devices that have built-in Windows Hello biometric sign-in face or fingerprint sensors.

Secured-core PC configuration lock

Applies to: ✅ Windows 11

In an enterprise organization, IT administrators enforce policies on their corporate devices to keep the devices in a compliant state and protect the OS by preventing users from changing configurations and creating config drift. Config drift occurs when users with local admin rights change settings and put the device out of sync with security policies. Devices in a noncompliant state can be vulnerable until the next sync and configuration reset with the MDM. Windows 11 with config lock enables IT administrators to prevent config drift and keep the OS configuration in the desired state. With config lock, the OS monitors the registry keys that configure each feature and when it detects a drift, reverts to the IT-desired state in seconds.

Secured-core configuration lock (config lock) is a new secured-core PC (SCPC) feature that prevents configuration drift from secured-core PC features caused by unintentional misconfiguration. In short, it ensures a device intended to be a secured-core PC remains a secured-core PC.

To summarize, config lock:

- Enables IT to "lock" secured-core PC features when managed through MDM
- Detects drift remediates within seconds
- Doesn't prevent malicious attacks

Windows edition and licensing requirements

The following table lists the Windows editions that support Secured-core configuration lock:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Secured-core configuration lock license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Configuration Flow

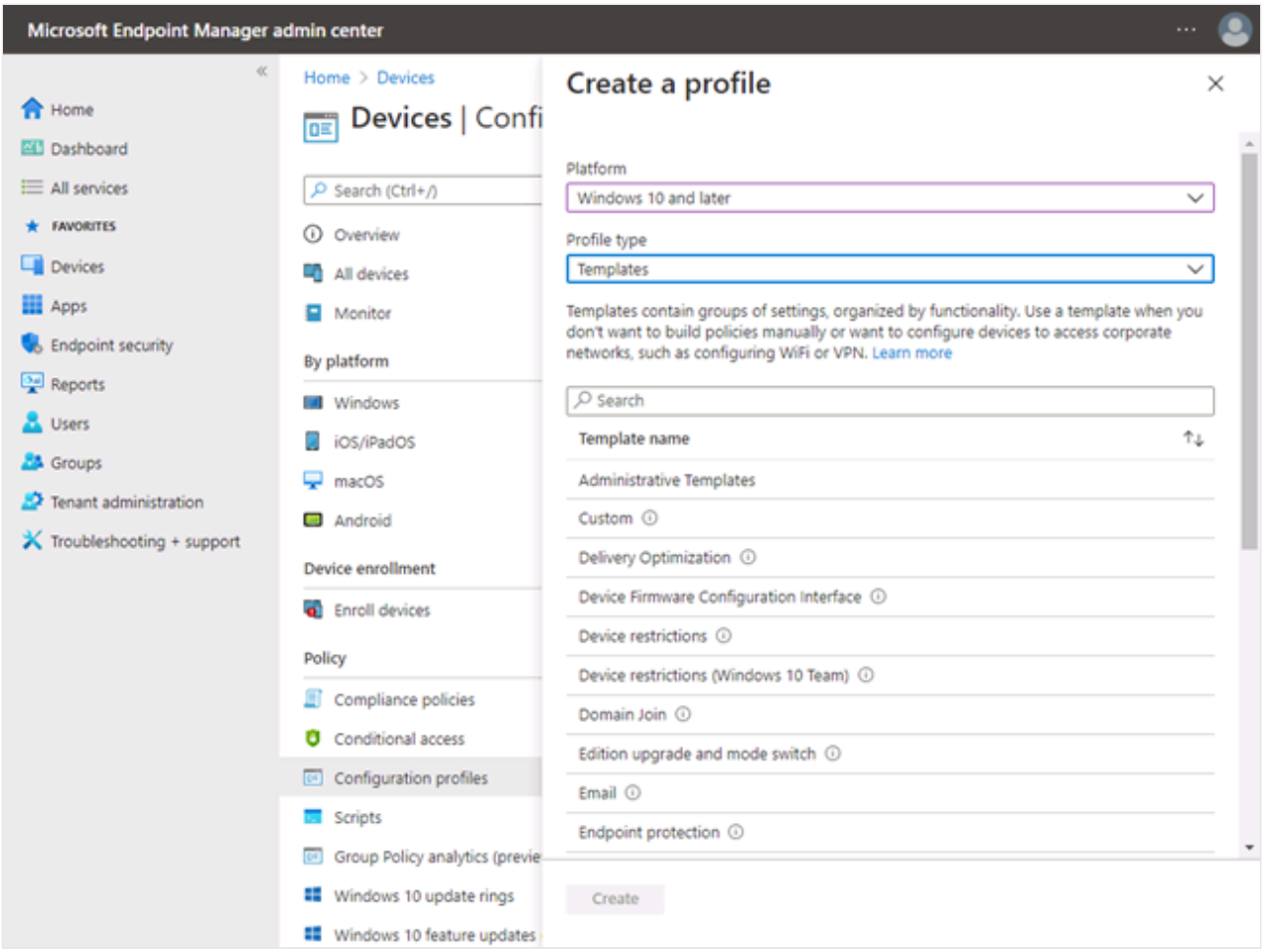
After a secured-core PCs reaches the desktop, config lock will prevent configuration drift by detecting if the device is a secured-core PC or not. When the device isn't a secured-core PC, the lock doesn't apply. If the device is a secured-core PC, config lock locks the policies listed under List of locked policies.

Enabling config lock using Microsoft Intune

Config lock isn't enabled by default, or turned on by the OS during boot. Rather, you need to turn it on.

The steps to turn on config lock using Microsoft Intune are as follows:

1. Ensure that the device to turn on config lock is enrolled in Microsoft Intune.
2. In the Intune admin center, select **Devices** > **Configuration Profiles** > **Create a profile**.
3. Select the following and press **Create**:
    - **Platform**: `Windows 10 and later`
    - **Profile type**: `Templates`
    - **Template name**: Custom

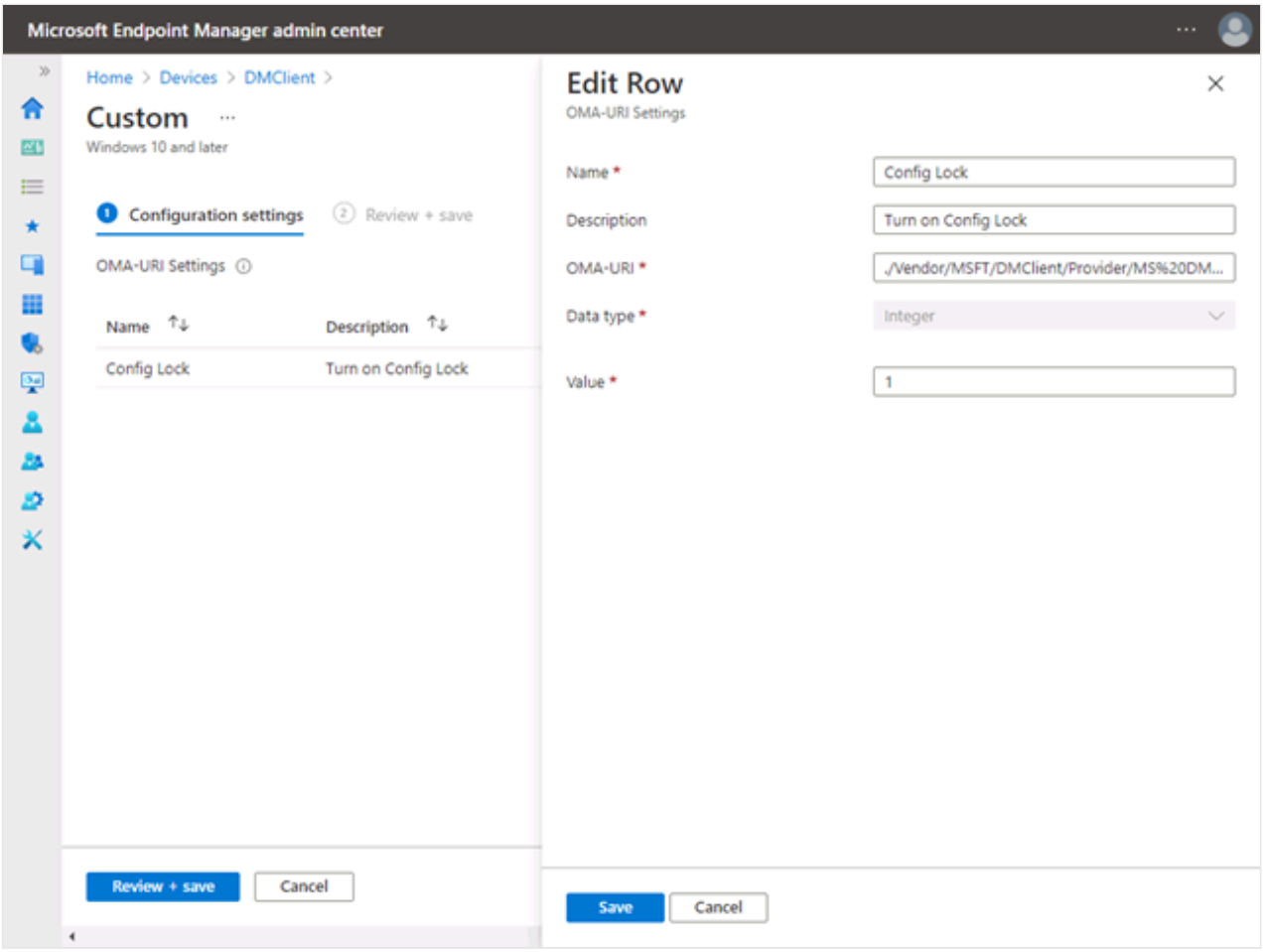


4. Name your profile.

5. When you reach the Configuration Settings step, select "Add" and add the following information:

   - **OMA-URI**: `./Vendor/MSFT/DMClient/Provider/MS%20DM%20Server/ConfigLock/Lock`
   - **Data type**: `Integer`
   - **Value**: `1`

   To turn off config lock, change the value to 0.

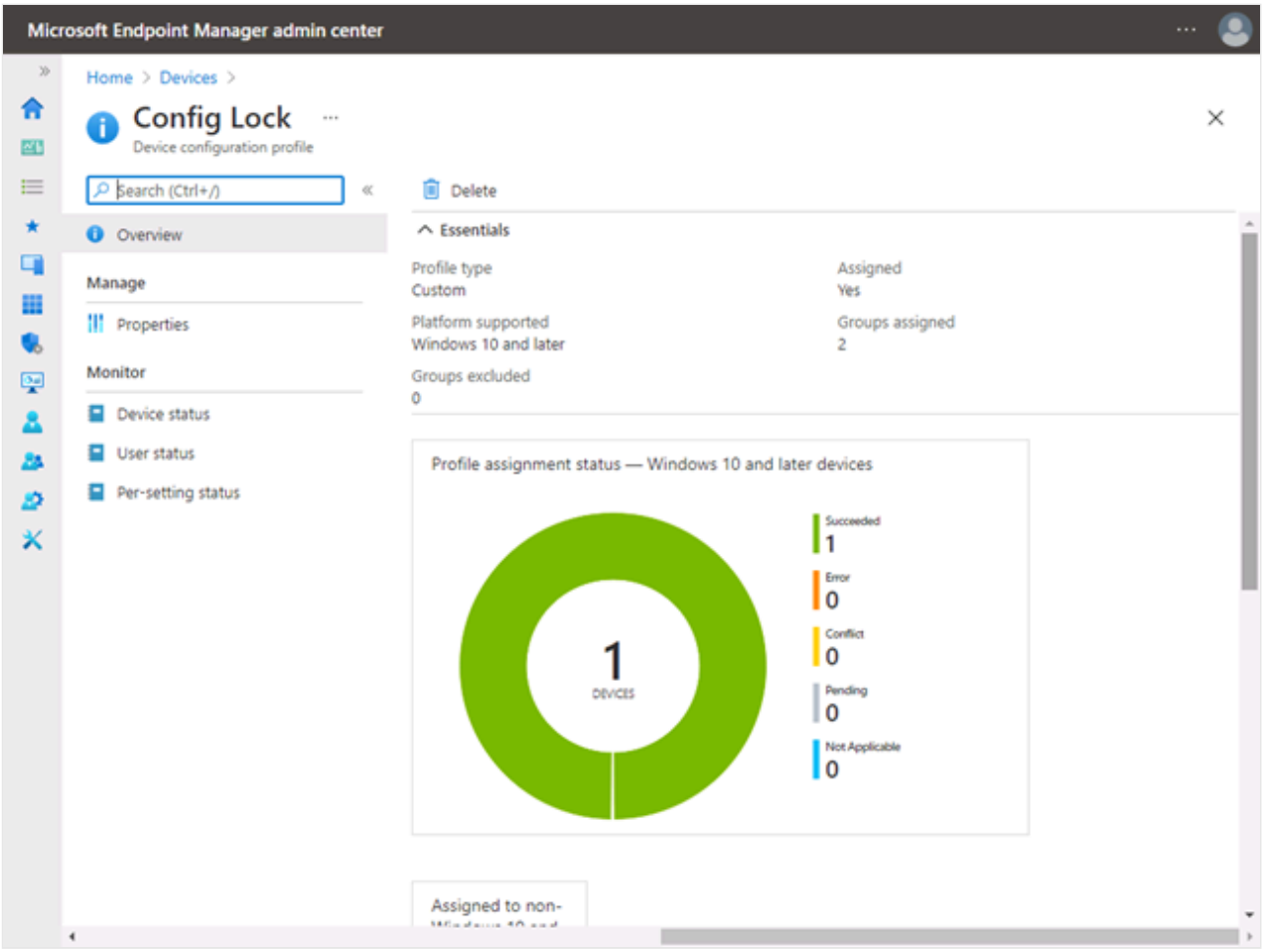


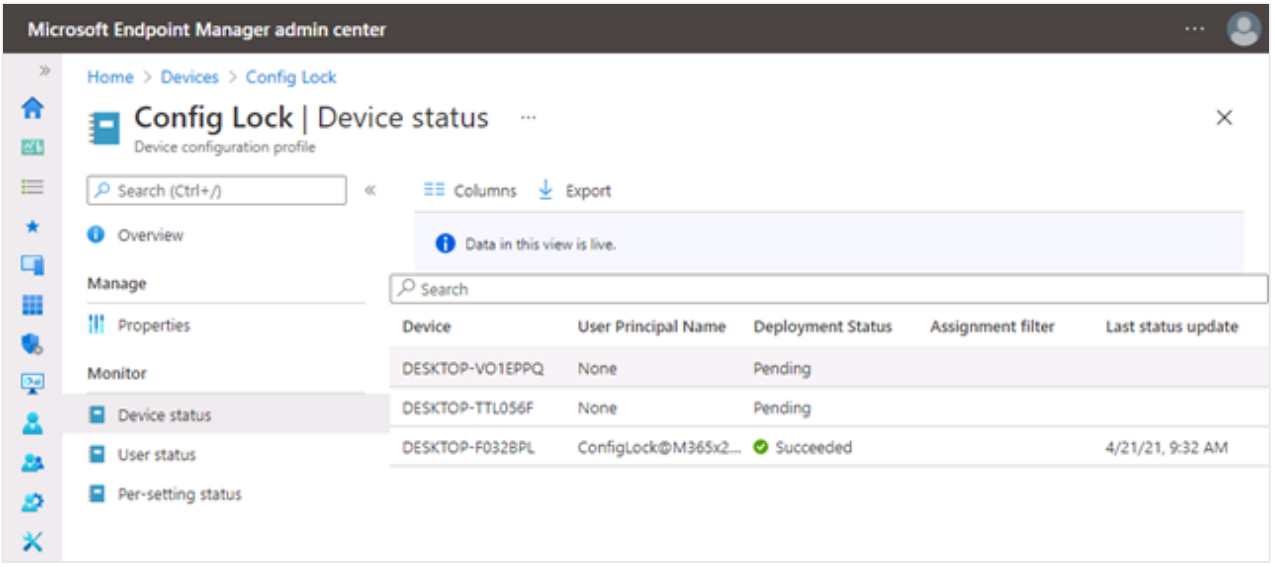
6. Select the devices to turn on config lock. If you're using a test tenant, you can select "+ Add all devices".

7. You don't need to set any applicability rules for test purposes.

8. Review the Configuration and select "Create" if everything is correct.

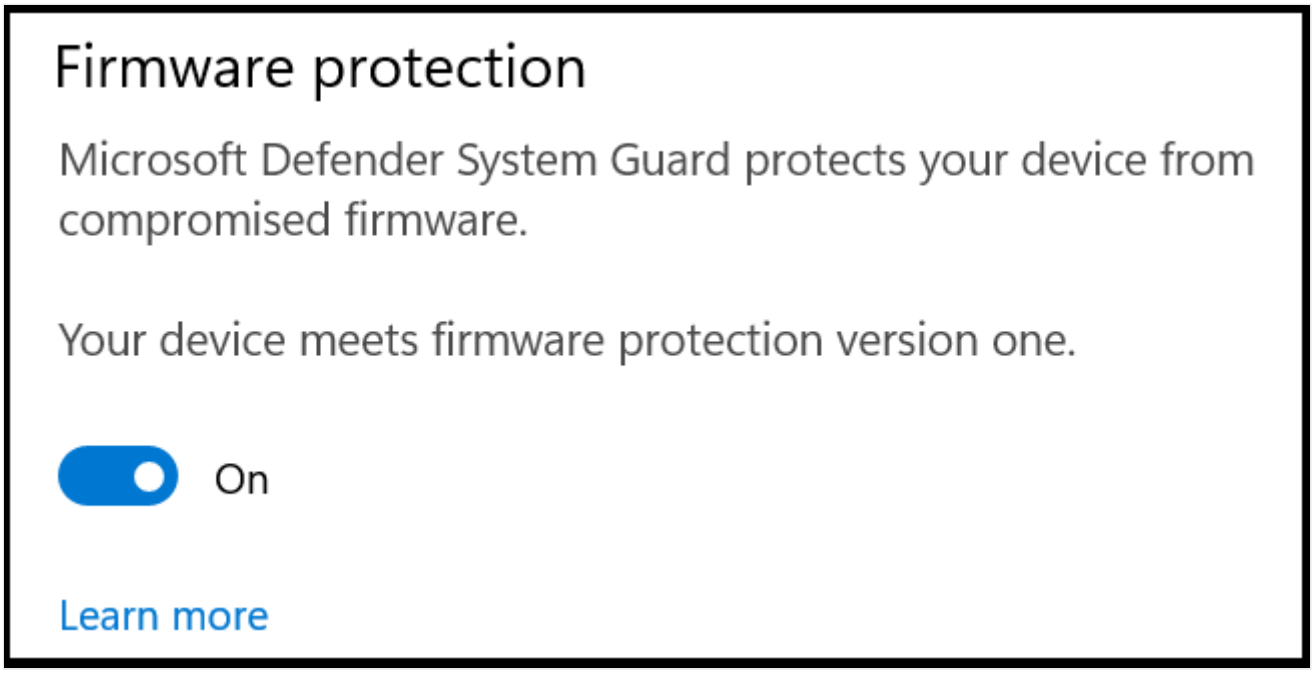
9. After the device syncs with the Microsoft Intune server, you can confirm if the config lock was successfully enabled.





Configuring secured-core PC features

Config lock is designed to ensure that a secured-core PC isn't unintentionally misconfigured. You keep the ability to enable or disable SCPC features, for example, firmware protection. You can make these changes with group policies or MDM services like Microsoft Intune.



FAQ

- Can I disable config lock? Yes. You can use MDM to turn off config lock completely or put it in temporary unlock mode for helpdesk activities.

List of locked policies

Expand table

CSPs
BitLocker
PassportForWork
WindowsDefenderApplicationGuard
ApplicationControl

Expand table

MDM policies	Supported by Group Policy
DataProtection/AllowDirectMemoryAccess	No
DataProtection/LegacySelectiveWipeID	No
DeviceGuard/ConfigureSystemGuardLaunch	Yes

MDM policies	Supported by Group Policy
DeviceGuard/EnableVirtualizationBasedSecurity	Yes
DeviceGuard/LsaCfgFlags	Yes
DeviceGuard/RequirePlatformSecurityFeatures	Yes
DeviceInstallation/AllowInstallationOfMatchingDeviceIDs	Yes
DeviceInstallation/AllowInstallationOfMatchingDeviceInstanceIDs	Yes
DeviceInstallation/AllowInstallationOfMatchingDeviceSetupClasses	Yes
DeviceInstallation/PreventDeviceMetadataFromNetwork	Yes
DeviceInstallation/PreventInstallationOfDevicesNotDescribedByOtherPolicySettings	Yes
DeviceInstallation/PreventInstallationOfMatchingDeviceIDs	Yes
DeviceInstallation/PreventInstallationOfMatchingDeviceInstanceIDs	Yes
DeviceInstallation/PreventInstallationOfMatchingDeviceSetupClasses	Yes
DmaGuard/DeviceEnumerationPolicy	Yes
WindowsDefenderSecurityCenter/CompanyName	Yes
WindowsDefenderSecurityCenter/DisableAccountProtectionUI	Yes
WindowsDefenderSecurityCenter/DisableAppBrowserUI	Yes
WindowsDefenderSecurityCenter/DisableClearTpmButton	Yes
WindowsDefenderSecurityCenter/DisableDeviceSecurityUI	Yes
WindowsDefenderSecurityCenter/DisableEnhancedNotifications	Yes
WindowsDefenderSecurityCenter/DisableFamilyUI	Yes
WindowsDefenderSecurityCenter/DisableHealthUI	Yes
WindowsDefenderSecurityCenter/DisableNetworkUI	Yes
WindowsDefenderSecurityCenter/DisableNotifications	Yes
WindowsDefenderSecurityCenter/DisableTpmFirmwareUpdateWarning	Yes
WindowsDefenderSecurityCenter/DisableVirusUI	Yes
WindowsDefenderSecurityCenter/DisallowExploitProtectionOverride	Yes
WindowsDefenderSecurityCenter/Email	Yes

MDM policies	Supported by Group Policy
WindowsDefenderSecurityCenter/EnableCustomizedToasts	Yes
WindowsDefenderSecurityCenter/EnableInAppCustomization	Yes
WindowsDefenderSecurityCenter/HideRansomwareDataRecovery	Yes
WindowsDefenderSecurityCenter/HideSecureBoot	Yes
WindowsDefenderSecurityCenter/HideTPMTroubleshooting	Yes
WindowsDefenderSecurityCenter/HideWindowsSecurityNotificationAreaControl	Yes
WindowsDefenderSecurityCenter/Phone	Yes
WindowsDefenderSecurityCenter/URL	Yes
SmartScreen/EnableAppInstallControl	Yes
SmartScreen/EnableSmartScreenInShell	Yes
SmartScreen/PreventOverrideForFilesInShell	Yes

Last updated on 08/04/2025

Kernel DMA Protection

Applies to: ✅ Windows 11, ✅ Windows 10

Kernel Direct Memory Access (DMA) Protection is a Windows security feature that protects against external peripherals from gaining unauthorized access to memory.

PCIe hot plug devices such as Thunderbolt, USB4, and CFexpress allow users to attach classes of external peripherals, including graphics cards, to their devices with the plug-and-play ease of USB. These devices are DMA-capable, and can access system memory and perform read and write operations without the need for the system processor's involvement. This capability is the reason behind the exceptional performance of PCI devices, but it also makes them susceptible to *drive-by DMA attacks*.

Drive-by DMA attacks are attacks that occur while the owner of the system isn't present and usually take just a few minutes, with simple-to-moderate attacking tools (affordable, off-the-shelf hardware and software), that don't require the disassembly of the device. For example, attackers can plug in a USB-like device while the device owner is on a break, and walk away with all the secrets on the machine, or inject a malware that allows them to have full control over the device remotely while bypassing the lock screen.

> ⓘ **Note**
>
> Kernel DMA Protection feature doesn't protect against DMA attacks via 1394/FireWire, PCMCIA, CardBus, or ExpressCard.

How Windows protects against DMA drive-by attacks

Windows uses the system *Input/Output Memory Management Unit (IOMMU)* to block external peripherals from starting and performing DMA, unless the drivers for these peripherals support memory isolation (such as DMA-remapping). Peripherals with DMA Remapping compatible drivers are automatically enumerated, started, and allowed to perform DMA to their assigned memory regions.

By default, peripherals with DMA Remapping incompatible drivers are blocked from starting and performing DMA until an authorized user signs into the system or unlocks the screen. IT administrators can modify the default behavior applied to devices with DMA Remapping incompatible drivers using MDM or group policies.

User experience

When Kernel DMA Protection is enabled:

- Peripherals with DMA Remapping-compatible device drivers are automatically enumerated and started
- Peripherals with DMA Remapping-incompatible drivers are blocked from starting if the peripheral was plugged in before an authorized user logs in, or while the screen is locked. Once the system is unlocked, the peripheral driver is started by the OS, and the peripheral continues to function normally until the system is rebooted, or the peripheral is unplugged. The peripheral will continue to function normally if the user locks the screen or signs out of the system.

Windows edition and licensing requirements

The following table lists the Windows editions that support Kernel Direct Memory Access (DMA) protection:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Kernel Direct Memory Access (DMA) protection license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

System compatibility

Kernel DMA Protection requires UEFI firmware support, and Virtualization-based Security (VBS) isn't required.

Kernel DMA Protection isn't compatible with other BitLocker DMA attacks countermeasures. It's recommended to disable the BitLocker DMA attacks countermeasures if the system supports Kernel DMA Protection. Kernel DMA Protection provides higher security bar for the system over the BitLocker DMA attack countermeasures, while maintaining usability of external peripherals.

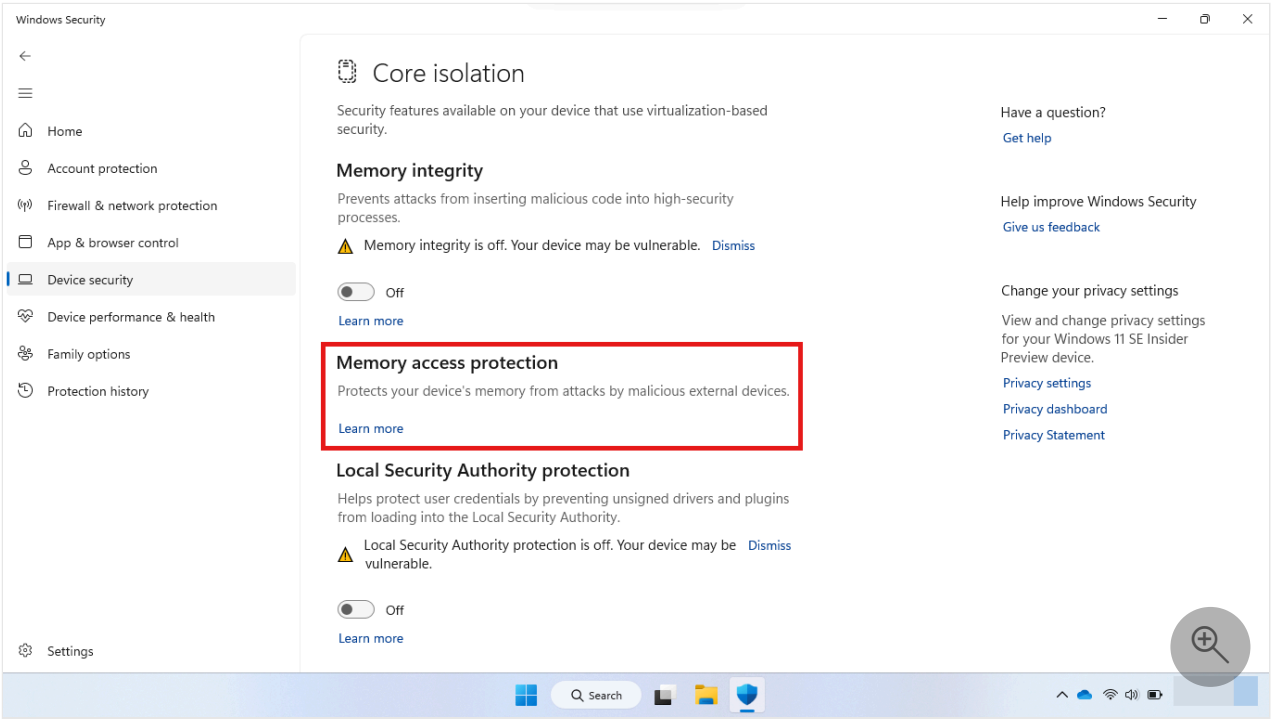
> **Note**
>
> DMA remapping support for graphics devices was added in Windows 11 with the WDDM 3.0 driver model; Windows 10 doesn't support this feature.

Check if Kernel DMA Protection is enabled

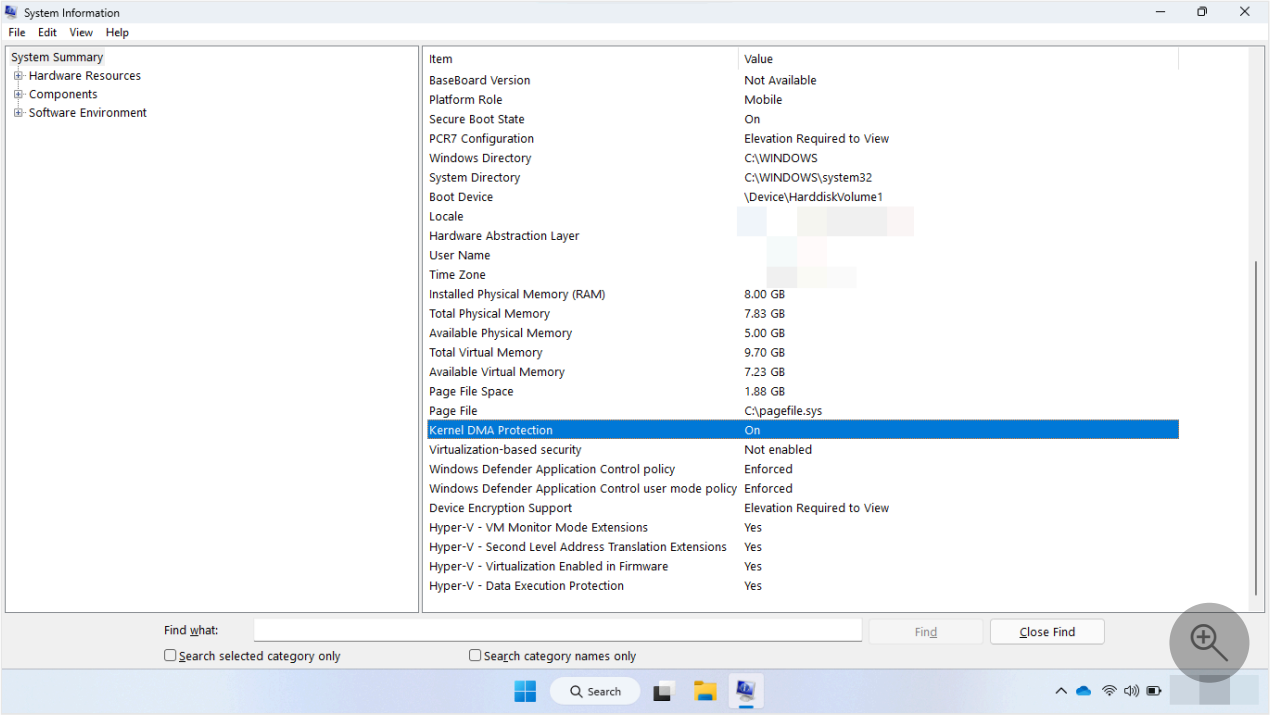
Systems that support Kernel DMA Protection enable the feature automatically, with no user or IT admin configuration required.

You can use the Windows Security settings to check if Kernel DMA Protection is enabled:

1. Open **Windows Security**.
2. Select **Device security** > **Core isolation details** > **Memory access protection**



Alternatively, you can use the System Information desktop app (`msinfo32.exe`). If the system supports Kernel DMA Protection, the **Kernel DMA Protection** value is set to **ON**.



If the current state of **Kernel DMA Protection** is **OFF** and **Hyper-V - Virtualization Enabled in Firmware** is **NO**:

- Reboot into UEFI settings
- Turn on Intel Virtualization Technology
- Turn on Intel Virtualization Technology for I/O (VT-d)
- Reboot system into Windows

> ⓘ **Note**
>
> If the **Hyper-V** Windows feature is enabled, all the Hyper-V-related features will be hidden, and **A hypervisor has been detected. Features required for Hyper-V will not be displayed** entity will be shown at the bottom of the list. It means that **Hyper-V - Virtualization Enabled in Firmware** is set to **YES**.
>
> Enabling Hyper-V virtualization in Firmware (IOMMU) is required to enable **Kernel DMA Protection**, even when the firmware has the flag of *ACPI Kernel DMA Protection Indicators* described in Kernel DMA Protection (Memory Access Protection) for OEMs.

If the state of **Kernel DMA Protection** remains Off, then the system doesn't support Kernel DMA Protection.

For systems that don't support Kernel DMA Protection, refer to the BitLocker countermeasures or Thunderbolt 3 and Security on Microsoft Windows Operating system for other means of DMA protection.

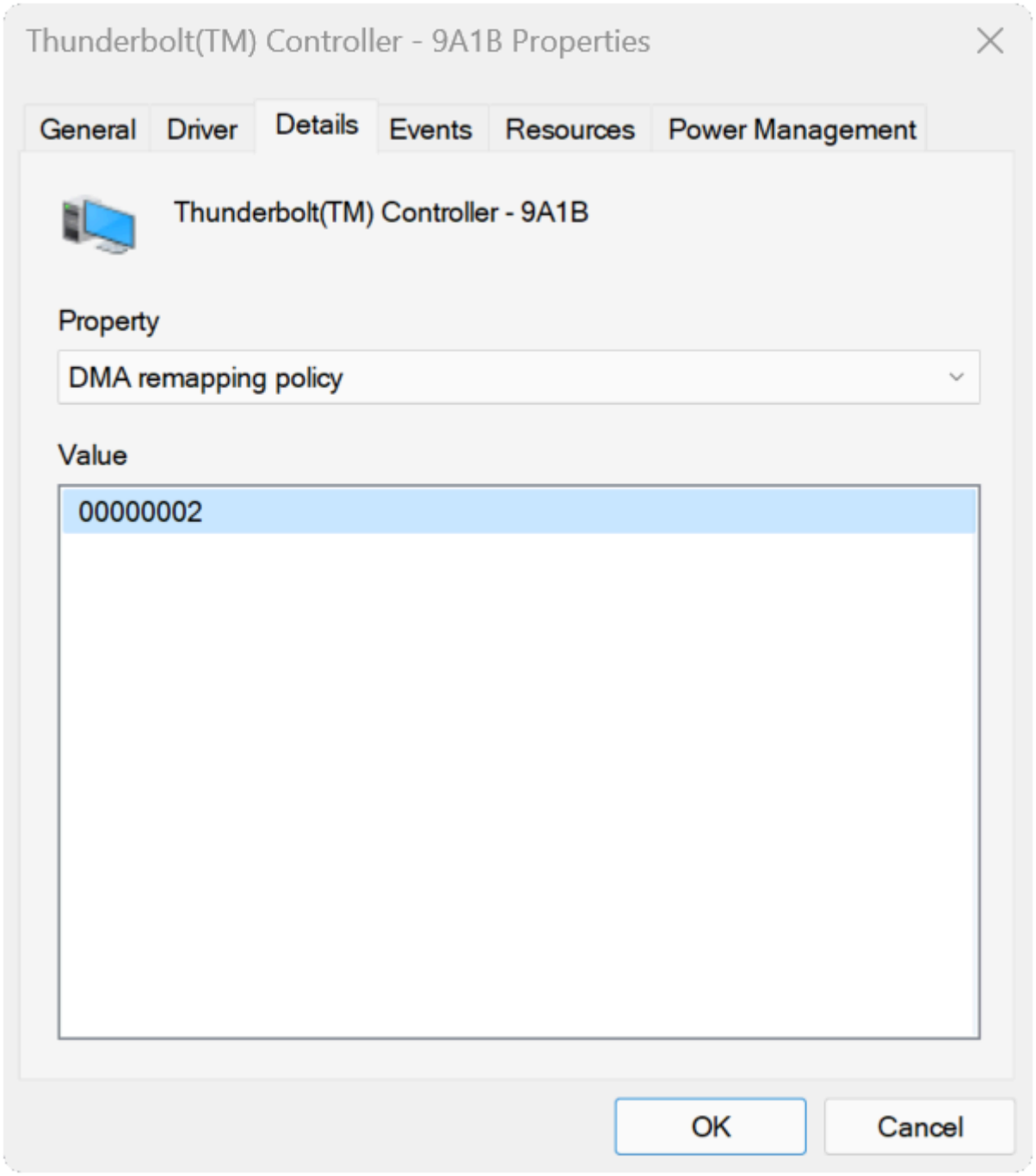
Frequently asked questions

Does Kernel DMA Protection prevent drive-by DMA attacks during Boot?

No, Kernel DMA Protection only protects against drive-by DMA attacks after the OS is loaded. It's the responsibility of the system firmware/BIOS to protect against attacks via the Thunderbolt 3 ports during boot.

How can I check if a certain driver supports DMA-remapping?

Not all devices and drivers support DMA-remapping. To check if a specific driver is opted into DMA-remapping, check the values corresponding to the DMA Remapping Policy property in the Details tab of a device in Device Manager*. A value of **0** or **1** means that the device driver doesn't support DMA-remapping. A value of **2** means that the device driver supports DMA-remapping. If the property isn't available, then the device driver doesn't support DMA-remapping. Check the driver instance for the device you're testing. Some drivers may have varying values depending on the location of the device (internal vs. external).



When the drivers for PCI or Thunderbolt 3 peripherals don't support DMA-remapping?

Use the Windows-provided drivers for the peripherals, when available. If there are no class drivers provided by Windows for your peripherals, contact your peripheral vendor/driver vendor to update the driver to support DMA Remapping.

My system's Kernel DMA Protection is off. Can DMA-remapping for a specific device be turned on?

Yes. DMA remapping for a specific device can be turned on independent from Kernel DMA Protection. For example, if the driver opts in and VT-d (Virtualization Technology for Directed

I/O) is turned on, then DMA remapping is enabled for the devices driver even if Kernel DMA Protection is turned off.

Kernel DMA Protection is a policy that allows or blocks devices to perform DMA, based on their remapping state and capabilities.

Do Microsoft drivers support DMA-remapping?

The Microsoft inbox drivers for USB XHCI (3.x) Controllers, Storage AHCI/SATA Controllers, and Storage NVMe Controllers support DMA Remapping.

Do drivers for non-PCI devices need to be compatible with DMA-remapping?

No. Devices for non-PCI peripherals, such as USB devices, don't perform DMA, thus no need for the driver to be compatible with DMA Remapping.

How can an enterprise enable the External device enumeration policy?

The External device enumeration policy controls whether to enumerate external peripherals that aren't compatible with DMA-remapping. Peripherals that are compatible with DMA-remapping are always enumerated. Peripherals that aren't, can be blocked, allowed, or allowed only after the user signs in (default).

The policy can be enabled by using:

- Group Policy: **Administrative Templates\System\Kernel DMA Protection\Enumeration policy for external devices incompatible with Kernel DMA Protection**
- Mobile Device Management (MDM): DmaGuard policies

Last updated on 08/15/2025

System Guard Secure Launch and SMM protection

Applies to: ✅ Windows 11, ✅ Windows 10

This article explains how to configure System Guard Secure Launch and System Management Mode (SMM) protection to improve the startup security of Windows 10 and Windows 11 devices. The information below is presented from a client perspective.

> ⓘ **Note**
>
> System Guard Secure Launch feature requires a supported processor. For more information, see System requirements for System Guard.

How to enable System Guard Secure Launch

You can enable System Guard Secure Launch by using any of these options:

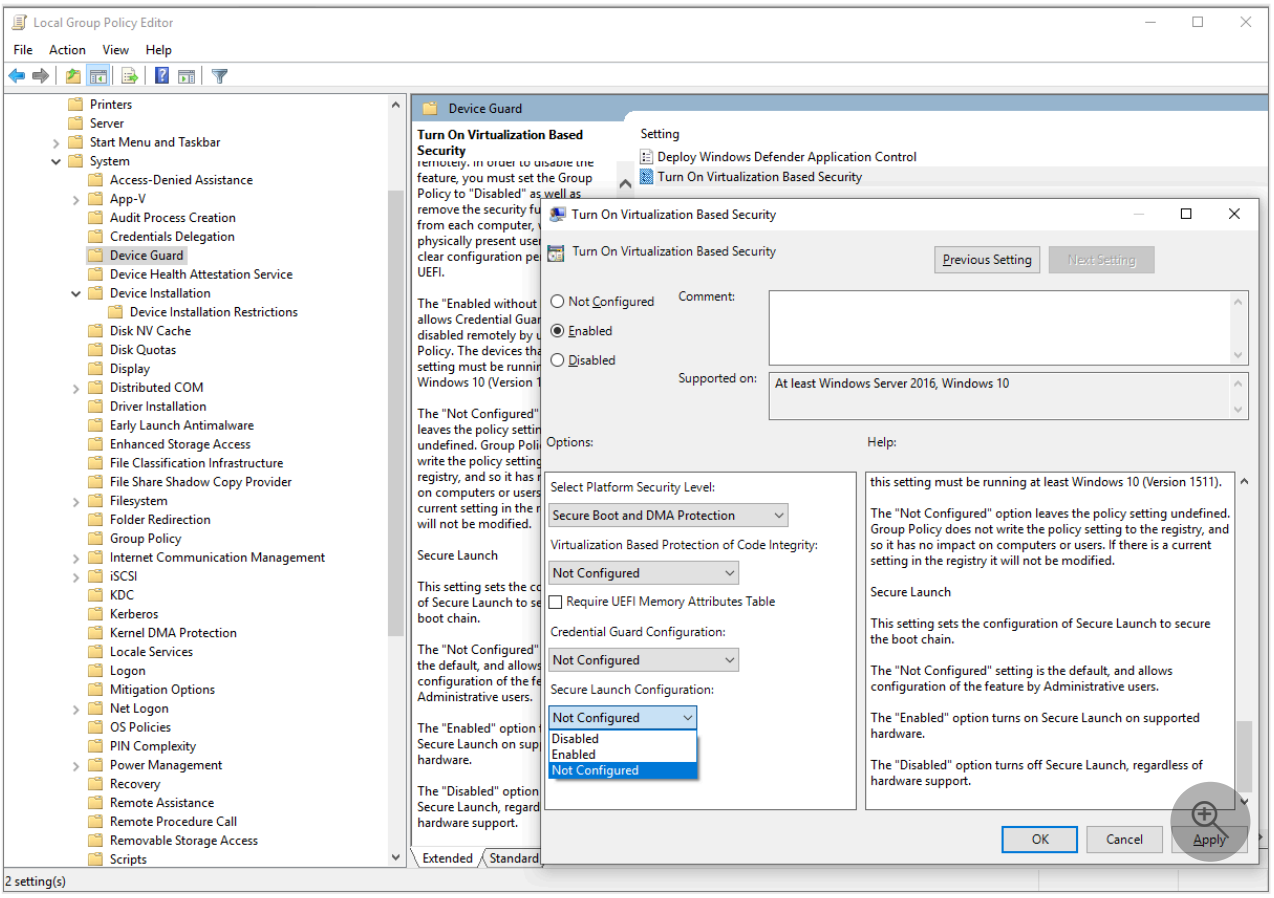
- Mobile Device Management (MDM)
- Group Policy
- Windows Security settings
- Registry

Mobile Device Management

System Guard Secure Launch can be configured for Mobile Device Management (MDM) by using DeviceGuard policies in the Policy CSP, DeviceGuard/ConfigureSystemGuardLaunch.

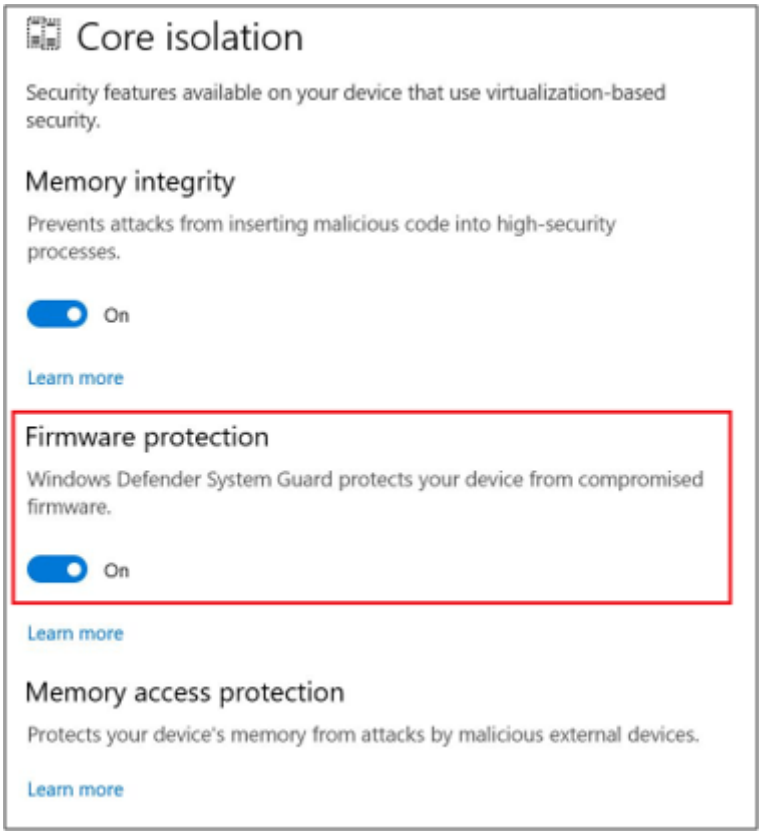
Group Policy

1. Select **Start** > type and then select **Edit group policy**.
2. Select **Computer Configuration** > **Administrative Templates** > **System** > **Device Guard** > **Turn On Virtualization Based Security** > **Secure Launch Configuration**.



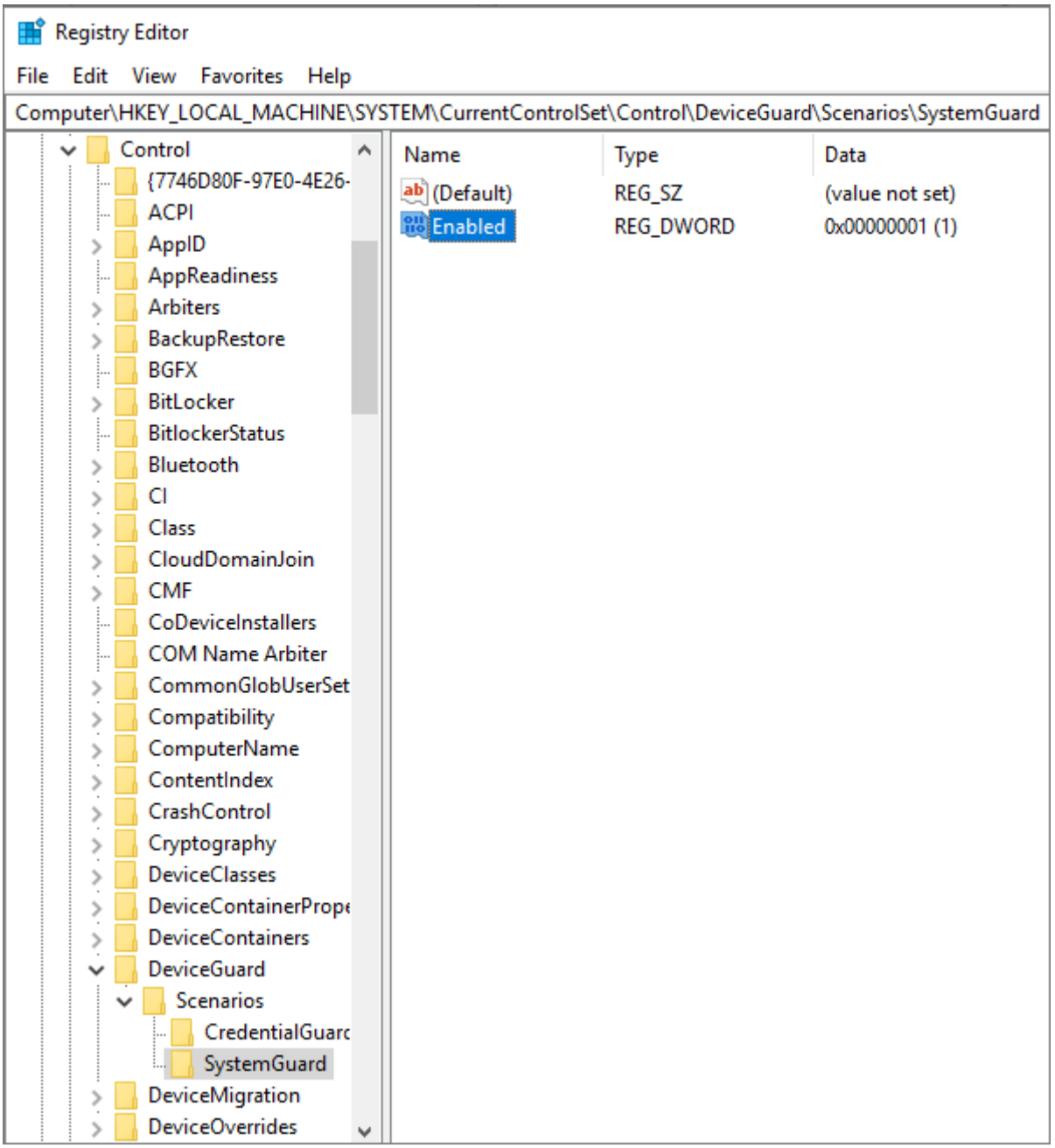
Windows Security

Select **Start** > **Settings** > **Update & Security** > **Windows Security** > **Open Windows Security** > **Device security** > **Core isolation** > **Firmware protection**.



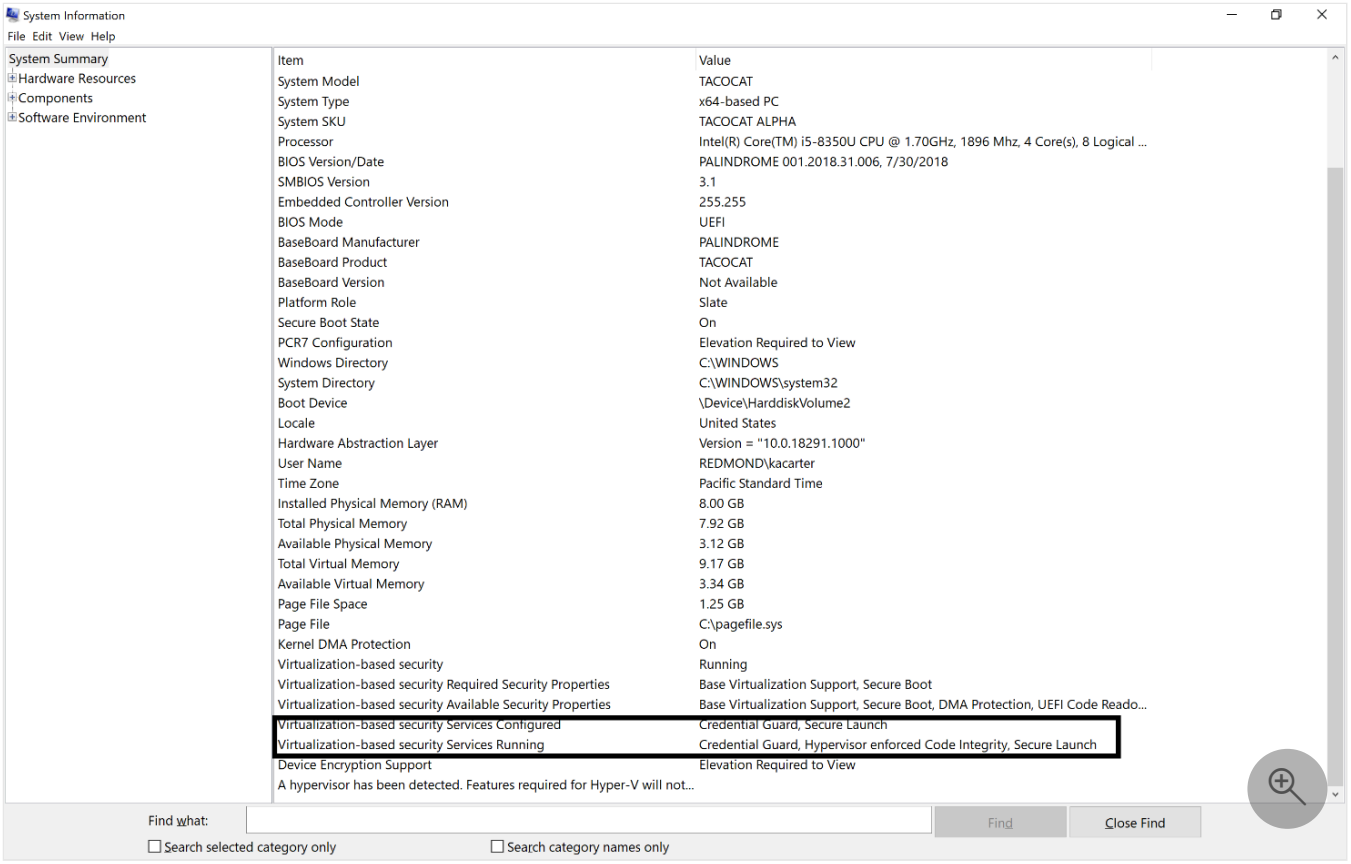
Registry

1. Open Registry editor.
2. Select **HKEY_LOCAL_MACHINE** > **SYSTEM** > **CurrentControlSet** > **Control** > **DeviceGuard** > **Scenarios**.
3. Right-click **Scenarios** > **New** > **Key** and name the new key **SystemGuard**.
4. Right-click **SystemGuard** > **New** > **DWORD (32-bit) Value** and name the new DWORD **Enabled**.
5. Double-click **Enabled**, change the value to **1**, and click **OK**.



How to verify System Guard Secure Launch is configured and running

To verify that Secure Launch is running, use System Information (MSInfo32). Select **Start**, search for **System Information**, and look under **Virtualization-based Security Services Running** and **Virtualization-based Security Services Configured**.



> **Note**
>
> To enable System Guard Secure launch, the platform must meet all the baseline requirements for System Guard, Device Guard, Credential Guard, and Virtualization Based Security.

> **Note**
>
> For more information around AMD processors, see Microsoft Security Blog: Force firmware code to be measured and attested by Secure Launch on Windows 10.

Last updated on 08/15/2025

Sysmon Overview

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

Built-in System Monitor (Sysmon) is an optional Windows feature on Windows 11 and Windows Server 2025 that when enabled, remains resident across system reboots to monitor and log system activity to the Windows event log. It provides detailed information about process creation, network connection, and file creation time change events. By collecting its events through Windows Event Collection or a SIEM agent, you can better understand what the system is doing. These events help you detect malicious or unusual behavior and reveal how intruders or malware move and operate within your environment.

Sysmon doesn't analyze the events it generates, nor does it attempt to conceal its presence from attackers.

Sysmon capabilities

Sysmon includes the following capabilities:

- Logs process creation with full command line for both current and parent processes.
- Records the hash of process image files using SHA1 (the default), MD5, SHA256 or IMPHASH.
- Multiple hashes can be used at the same time.
- Includes a process GUID in process create events to allow for correlation of events even when Windows reuses process IDs.
- Includes a session GUID in each event to allow correlation of events on same logon session.
- Logs loading of drivers or DLLs with their signatures and hashes.
- Logs when disks or volumes are opened with raw read access.
- Optionally logs network connections, including each connection's source process, IP addresses, port numbers, hostnames and port names.
- Detects changes in file creation time to understand when a file was actually created. Modification of file create timestamps is a technique commonly used by malware to cover its tracks.

- Automatically reload configuration if changed in the registry.
- Rule filtering to include or exclude certain events dynamically.
- Generates events from early in the boot process to capture activity made by even sophisticated kernel-mode malware.

These capabilities give security teams better context than default audit logs alone and improve detection of suspicious behaviors such as living-off-the-land execution, lateral movement, and malicious activity.

Usage

Some common usage examples featuring simple command-line options to install and uninstall Sysmon, as well as to check and modify its configuration:

Install:

PowerShell

```
sysmon -i [<configfile>]
```

Update configuration:

PowerShell

```
sysmon -c [<configfile>]
```

Install event manifest:

PowerShell

```
sysmon -m
```

Print schema:

PowerShell

```
sysmon -s
```

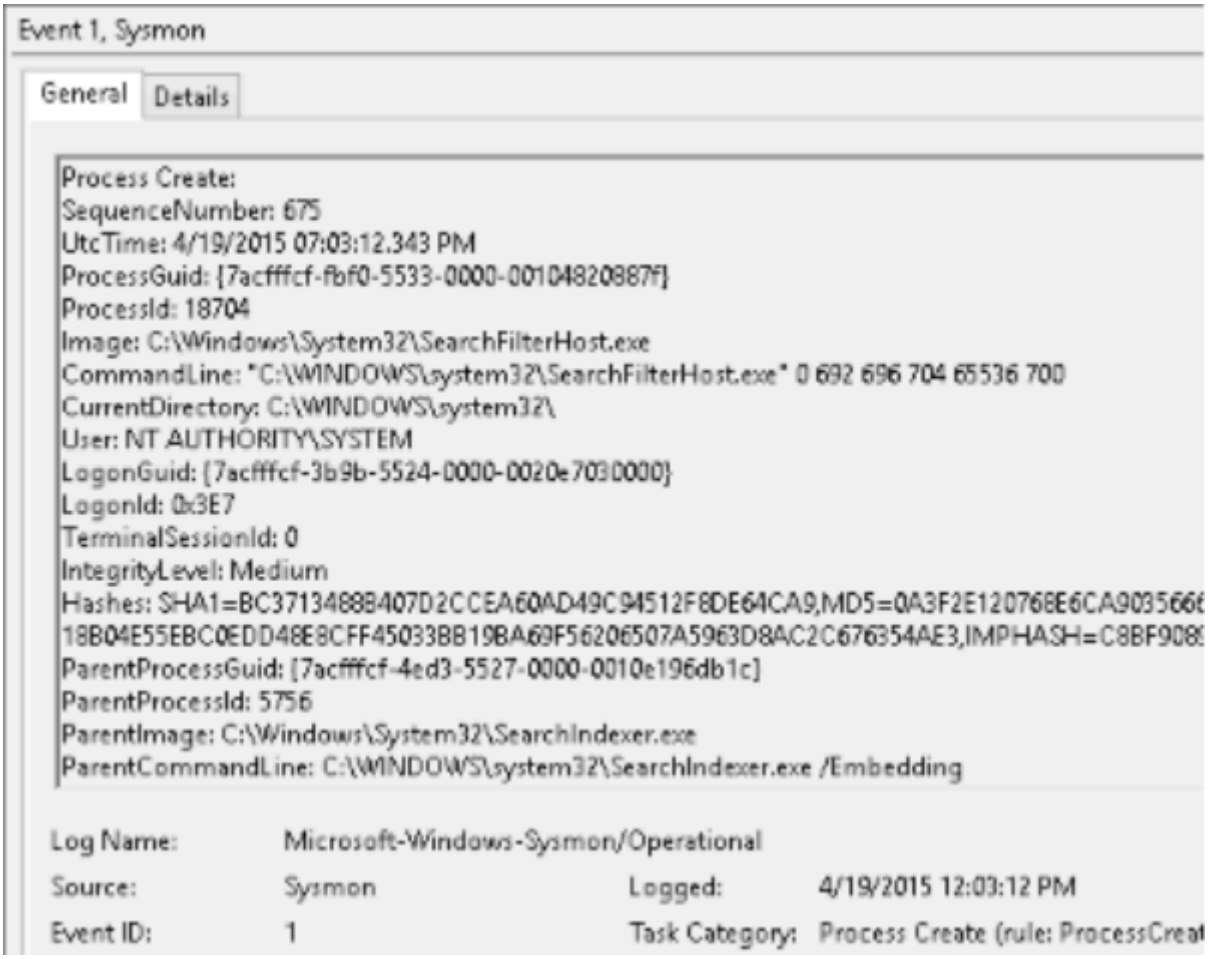
Uninstall:

PowerShell

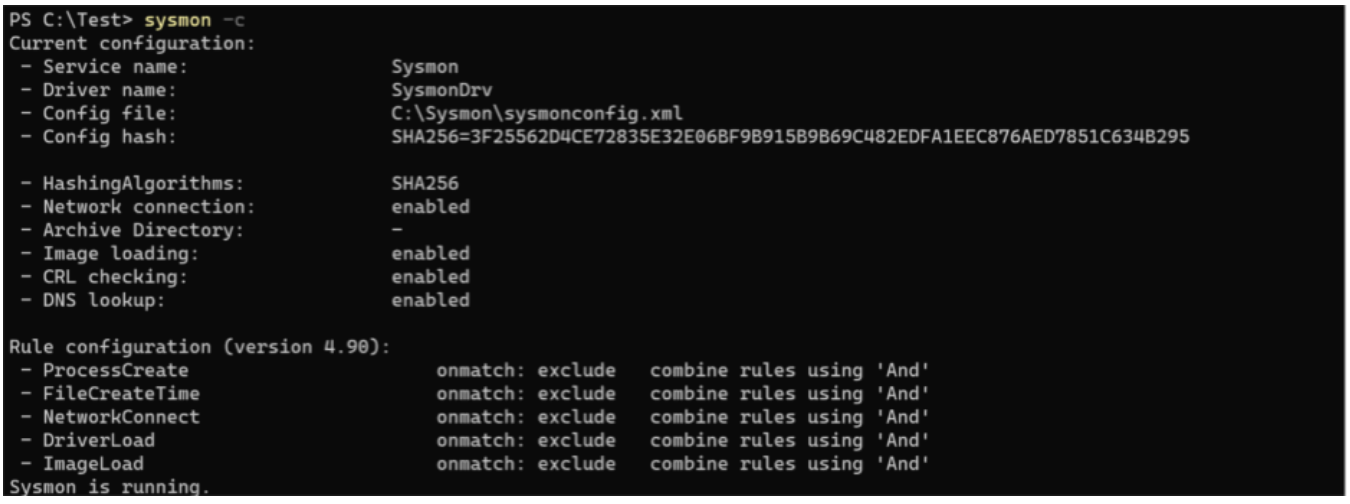
```
sysmon -u [force]
```

Screenshots

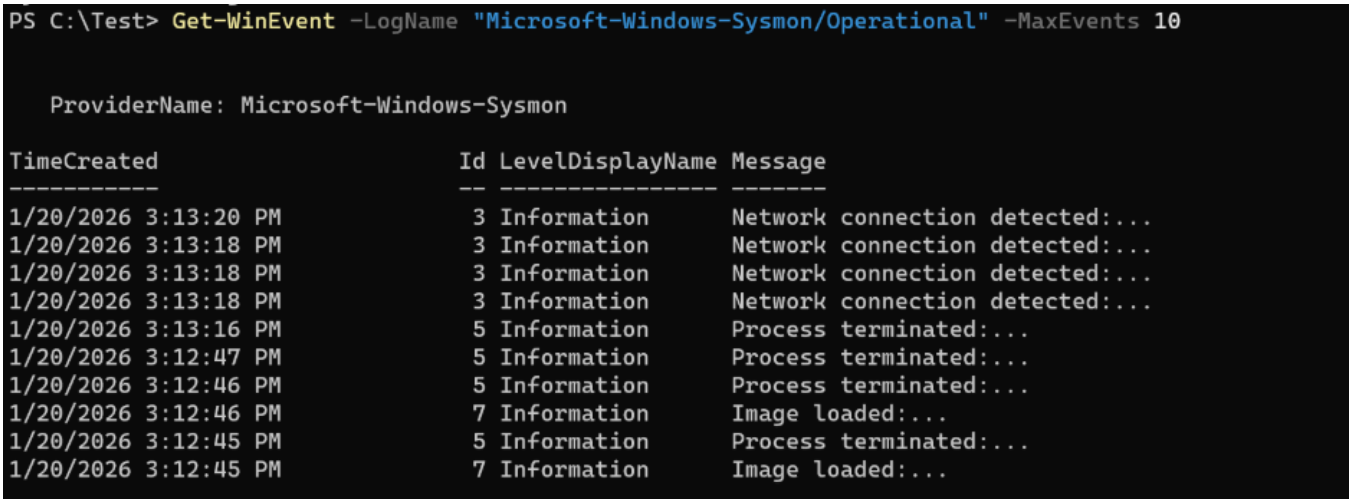
The following screenshot shows typical Sysmon telemetry in the built-in Windows Event Viewer:



The following screenshot showcases Sysmon outputs when confirming the telemetry configuration:



The following screenshot showcases Sysmon actively storing telemetry events:



```
PS C:\Test> Get-WinEvent -LogName "Microsoft-Windows-Sysmon/Operational" -MaxEvents 10


   ProviderName: Microsoft-Windows-Sysmon

TimeCreated                     Id LevelDisplayName Message
-----------                     -- ---------------- -------
1/20/2026 3:13:20 PM             3 Information      Network connection detected:...
1/20/2026 3:13:18 PM             3 Information      Network connection detected:...
1/20/2026 3:13:18 PM             3 Information      Network connection detected:...
1/20/2026 3:13:18 PM             3 Information      Network connection detected:...
1/20/2026 3:13:16 PM             5 Information      Process terminated:...
1/20/2026 3:12:47 PM             5 Information      Process terminated:...
1/20/2026 3:12:46 PM             5 Information      Process terminated:...
1/20/2026 3:12:46 PM             7 Information      Image loaded:...
1/20/2026 3:12:45 PM             5 Information      Process terminated:...
1/20/2026 3:12:45 PM             7 Information      Image loaded:...
```

Localized events

Built-in Sysmon events written to the Windows event log are localized, providing a consistent experience with other Windows system events and aligning with the language configured on the device. While the rendered event message (the display text shown in tools such as Event Viewer) is localized, the underlying XML event data is not localized and remains consistent across all languages. This XML representation can be used for reliable event collection, aggregation, and analysis across environments.

This behavior differs from standalone Sysmon and is important to consider when integrating built-in Sysmon with log collection pipelines, SIEM solutions, or custom event processing logic.

> ⓘ **Note**
>
> If you currently collect or process Sysmon events using non-XML representations (such as rendered message text), you may need to update event processing scripts to account for localization on non-English systems.

Servicing and updates

Enhancements and non-security improvements to built-in Sysmon are delivered through the standard Windows quality update process, with changes first available in optional Windows preview updates and then broadly deployed in the next regularly scheduled Windows update. For more information, see Windows quality updates overview.

During these feature updates:

- Built-in Sysmon binaries are updated whether or not Sysmon is currently enabled.

- If built-in Sysmon is enabled, the transition to the updated binaries is seamless, existing configuration is preserved, and no system restart is required.
- If built-in Sysmon isn't enabled, the update replaces the binaries but Sysmon remains disabled.

Feature updates to built-in Sysmon don't impact standalone Sysmon installations on the device. Coexistence between built-in Sysmon and standalone Sysmon isn't supported.

Testing and validation

Organizations should follow standard Windows update testing practices when adopting new built-in Sysmon features:

- Evaluate optional preview updates in test or pilot environments.
- Review Sysmon event output and behavior after updates.
- Adjust Sysmon configuration as needed prior to broad deployment.

This approach allows teams to validate changes while maintaining alignment with their existing Windows update management processes.

Quality updates

If a critical security issue affecting built-in Sysmon is identified, the fix is delivered as part of the Monthly Security Update Release (Update Tuesday / B release).

Security updates are deployed broadly and don't require opt-in preview updates.

Updates apply regardless of whether built-in Sysmon is enabled, ensuring that Sysmon components remain protected.

Coexistence with Sysmon & Other Security Products

Built-in Sysmon is designed to operate as a native Windows capability and doesn't support coexistence with standalone Sysmon.

- Standalone Sysmon and built-in Sysmon can't be enabled on the same device at the same time.
- Standalone Sysmon must be uninstalled before enabling built-in Sysmon.

- Windows updates that deliver or update built-in Sysmon don't affect standalone Sysmon installations. If standalone Sysmon is installed, it remains unchanged and built-in Sysmon remains disabled.

Sysmon is compatible with other security products such as:

- Windows Event Forwarding and Windows Event Collection, which aggregate Sysmon logs centrally for analysis.
- Microsoft Defender for Endpoint and other EDR platforms that consume Sysmon events for enhanced detection logic.
- SIEM solutions to correlate Sysmon telemetry with other log sources.

Sysmon’s filtering and configuration capabilities allow administrators to tailor event capture so that data volumes remain manageable, minimizing overlap with other agents while maximizing signal quality. It doesn't conflict with antivirus or other monitoring tools, as it provides raw telemetry rather than active prevention.

Last updated on 02/24/2026

Enable and configure Sysmon in Windows

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

This article describes how to enable and configure Sysmon to collect detailed security telemetry on Windows systems. It covers installation options, service behavior, and configuration requirements needed to begin capturing high-fidelity event data for security monitoring and forensics.

Sysmon is disabled by default. You must explicitly install and configure it to produce events. Effective deployment requires selecting an appropriate configuration, understanding include and exclude filtering behavior, and ensuring the service is protected and reliably logs data for downstream analysis.

What Sysmon does and doesn't do

- Sysmon records system activity and writes events to the Windows Event Log.
- Sysmon doesn't analyze events or generate alerts.
- Sysmon doesn't block or prevent activity.

To analyze Sysmon events, forward them to Windows Event Collection or a Security Information and Event Management(SIEM) solution e.g Splunk

Prerequisites

Before enabling built-in Sysmon, ensure that:

- A device running a supported version Windows 11 or later.
- You have administrator privileges or equivalent on the device.
- Standalone Sysmon isn't installed. Built-in Sysmon doesn't support coexistence with standalone Sysmon.

Check for existing installs

Built-in Sysmon is designed to operate as a native Windows capability and doesn't support coexistence with standalone Sysmon.

To verify Sysmon isn't already running, follow the steps.

1. Open a PowerShell window as an administrator.
2. Check for the Sysmon service using the following command.

   PowerShell

   ```
   Get-Service sysmon*
   ```

3. Review the output, you shouldn't return any results. If you do, make sure to uninstall the standalone Sysmon before continuing.

Enable built-in Sysmon

Built-in Sysmon is disabled by default and must be explicitly enabled. To enable Sysmon, follow the steps.

1. To install the Sysmon optional feature, run the following command.

   PowerShell

   ```
   Enable-WindowsOptionalFeature -Online -FeatureName Sysmon
   ```

2. To install Sysmon using the default configuration, run the following command.

   PowerShell

   ```
   sysmon –i
   ```

Provide a Sysmon configuration

Sysmon uses a configuration file to determine which events are logged and which events are filtered out.

The configuration file is written in XML and defines rules for logging specific Sysmon events. Some example event types are; Process Create, Network Connect, and File Create. For more information about how configuration files work, see Understanding Sysmon configuration files.

1. Create or obtain a Sysmon configuration file. The configuration typically includes:

   a. Enabled event types

    b. Include and exclude filtering rules

    c. Hash algorithms and metadata options

2. Examples configs and guides available:

    a. Sysmon-config – Default high-quality Sysmon event configuration maintained by SwiftOnSecurity:

    https://github.com/SwiftOnSecurity/sysmon-config

    b. Sysmon-modular – Modular Sysmon configuration with MITRE ATT&CK coverage maintained by Olaf Hartong

    https://github.com/olafhartong/sysmon-modular

    c. SysmonCommunityGuide – Guide, technical details, and examples for Sysmon maintained by Carlos Perez

    https://github.com/trustedsec/SysmonCommunityGuide

    d. Sysmon configuration deep dive

The configuration files typically include:

- Enabled event types
- Include and exclude filtering rules
- Hash algorithms and metadata options

You can find examples configuration files and guides available in the following resources:

- Sysmon-config – Default high-quality Sysmon event configuration maintained by SwiftOnSecurity:

  https://github.com/SwiftOnSecurity/sysmon-config

- Sysmon-modular – Modular Sysmon configuration with MITRE ATT&CK coverage maintained by Olaf Hartong

  https://github.com/olafhartong/sysmon-modular

- SysmonCommunityGuide by Carlos Perez – A guide, technical details, and examples for Sysmon
- Sysinternals: System Monitor deep dive (demo)

1. Place the configuration file in any directory of your choosing, e.g:

   `C:\Sysmon\sysmonconfig.xml`

2. Open a PowerShell window as an administrator

3. Change directory to the directory where you placed your configuration file using the `cd` command. For example, `cd c:\Sysmon`

4. To apply your configuration file perform one of the following:
   a. Configure Sysmon with the configuration file using the following command. Make sure you replace `C:\Sysmon\sysmonconfig.xml` with the path to your configuration file.

   PowerShell

   ```
   sysmon – c C:\Sysmon\sysmonconfig.xml
   ```

   Or

   a. If you've not installed Sysmon yet you can install Sysmon with the configuration file using the following command. Make sure you replace `C:\Sysmon\sysmonconfig.xml` with the path to your configuration file.

   PowerShell

   ```
   sysmon -i C:\Sysmon\sysmonconfig.xml
   ```

   Once applied, the configuration takes effect immediately. No restart is required.

> **ⓘ Important**
>
> A restrictive or unoptimized configuration may generate high event volume. Review and test configurations before broad deployment.

Example configuration

The following example shows a minimal Sysmon configuration that enables common event types:

XML

```
<Sysmon schemaversion="4.90">
```

```
  <HashAlgorithms>SHA256</HashAlgorithms>

  <EventFiltering>
    <ProcessCreate onmatch="exclude" />
    <NetworkConnect onmatch="exclude" />
    <FileCreate onmatch="exclude" />
  </EventFiltering>
</Sysmon>
```

This example enables basic visibility and can be expanded with include and exclude rules to reduce noise.

> **Tip**
>
> `Process create on match -exclude` with no sub parameters includes all process creation events

Verify that Sysmon is enabled and configured

After enabling and configuring built-in Sysmon, verify that it is functioning correctly using the following steps.

1. Select the **Start** button, type **Event viewer**, and open **Event viewer** from the best match list.
2. In Event Viewer, go to **Applications and Services Logs** > **Microsoft** > **Windows** > **Sysmon** > **Operational**.
3. Confirm that Sysmon events are present, such as Process Create, Network Connect, File Create. If events appear, built-in Sysmon is enabled and using the active configuration.

Updating the Sysmon configuration

To update the Sysmon configuration, follow these steps.

1. Modify the Sysmon configuration file using the guidance in Understanding sysmon configuration files.
2. Apply the updated configuration using the following command. Remember to replace `C:\Sysmon\sysmonconfig.xml` with the path to your configuration file.

PowerShell

```
sysmon -c C:\Sysmon\sysmonconfig.xml
```

Configuration updates are applied dynamically and don't require a system restart.

Next steps

- Tune Sysmon filters to balance visibility and event volume.
- Forward Sysmon events to a centralized log collection or SIEM.
- View this how to on reading/forwarding and tuning Sysmon post config.

Last updated on 02/24/2026

Read and tune Sysmon events

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

This article describes how to review Sysmon events after configuration, interpret common event patterns, and tune filtering rules to balance visibility and event volume.

Prerequisites

To make sure you are set up to utilise sysmon, ensure the following is true:

- Sysmon is enabled and configured on the device. For more information about installing Sysmon, see Enable and configure Sysmon.
- Sysmon events are visible locally in Event Viewer or forwarded to a centralized log platform.
- You understand which Sysmon event types are enabled in your configuration.

Reading Sysmon events locally

Sysmon writes events to the Windows Event Log. To view Sysmon logs, follow the steps:

1. Select the **Start** button, type **Event viewer**, and open **Event viewer** from the best match list.
2. In Event Viewer, go to **Applications and Services Logs** > **Microsoft** > **Windows** > **Sysmon** > **Operational**.
3. Select an event and view its details.

Each event includes structured fields such as:

- Event ID and name (for example, Process Create)
- Process and parent process information
- Command line
- File paths
- Network addresses and ports

- Hashes and identifiers

Interpreting common Sysmon event types

Understanding the intent of each event type helps distinguish expected behavior from suspicious activity. The following lists the common Sysmon event types.

Expand table

Type	Indication	Look for
Process create	Indicates a process had started	Unusual parent-child relationships
		Unexpected command-line arguments
		Script engines or administrative tools running in non-administrative contexts
Network create	Indicates an outbound or inbound network connection initiated by a process.	Rare destination IP addresses or ports
		Network activity from processes that typically do not communicate externally
		Repeated connections to the same destination
File create	Indicates that a file was created or overwritten	Executables written to user-writable locations
		Files created shortly after suspicious process execution
Image create	Indicates a dynamic link library (DLL) or executable image was loaded	Unsigned or unexpected modules
		Rare DLL's loaded into commonly abused processes
Registry events	Indicate registry key or value changes	Modifications to autorun or persistence-related keys
		Modifications to autorun or persistence-related keys
		Changes occurring shortly after process creation events Sysmon configurations should be tuned over time to reduce noise while preserving useful signal

Tune Sysmon configuration

Sysmon configurations should be tuned over time to reduce noise while preserving useful signal.

Identify high-volume events

Start by determining which Sysmon event types generate the most data in your environment. Use your log aggregation platform to summarize event counts by Event ID over a representative time period (for example, one to two weeks). Summarizing as such quickly reveals whether events such as ProcessCreate, NetworkConnect, ImageLoad, or FileCreate are contributing disproportionately to overall telemetry volume. High-volume events are not inherently problematic, but they are the primary candidates for tuning.

Review event counts by event type

Once high-volume event types are identified, analyze trends over time. Look for patterns such as spikes during system startup, user logon, software updates, or development workflows. Break down counts by host role (for example, developer workstation vs. domain controller) to understand whether certain systems naturally produce more telemetry. This helps avoid over-tuning based on activity that is expected only on specific device classes.

Identify processes or paths that generate excessive events

Within a high-volume event type, group and sort by relevant fields such as Image, CommandLine, TargetFilename, DestinationPort, or RegistryKey. Doing so typically reveals a few processes, file paths, or services responsible for most events. For example, a security agent may repeatedly access processes, or a development tool may generate frequent file creation events. Focus on identifying consistent, repeatable sources of noise rather than rare outliers.

Determine whether the activity is expected

Evaluate whether the identified behavior is legitimate and required for normal operations. Consult system owners or application teams if necessary. If the activity is expected and low-risk, consider adding narrowly scoped exclusion rules targeting the specific process, path, or condition responsible. Avoid disabling entire event categories or using overly broad string matches, as doing so can remove important investigative context. After applying changes, compare telemetry volumes before and after tuning to validate the impact and confirm that meaningful signals remain visible.

Apply include and exclude rules

Use filtering rules to focus on high-value activity.

Examples:

- Exclude known benign processes from Network Connect events.
- Include only specific file paths for File Create events.
- Exclude common system DLLs from Image Load events.

> **Tip**
>
> Start with broader logging and gradually add exclusions rather than aggressively filtering early.

Last updated on 02/24/2026

Understanding Sysmon configuration files

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

Sysmon configuration files define what telemetry Sysmon records and what it ignores. Because Sysmon captures detailed, low-level system activity, configuration design directly determines the usefulness, performance, and investigative value of the resulting data.

This article explains the structure of Sysmon configuration files, how filtering rules are evaluated, and how to reason about configuration design when deploying Sysmon as a built-in Windows feature.

Configuration files

Configuration files define the behavior of Sysmon by specifying global settings and event filtering rules. They make it easier to deploy a preset configuration and to filter captured events.

In built-in Sysmon, configuration is applied through Windows configuration mechanisms and evaluated at runtime. Configuration updates take effect immediately and persist across reboots.

Basic configuration structure

A simple Sysmon configuration XML file looks like this:

XML

```xml
<Sysmon schemaversion="4.82">
  <!-- Capture all hashes -->
  <HashAlgorithms>*</HashAlgorithms>
  <EventFiltering>
    <!-- Log all drivers except if the signature -->
    <!-- contains Microsoft or Windows -->
    <DriverLoad onmatch="exclude">
      <Signature condition="contains">microsoft</Signature>
      <Signature condition="contains">windows</Signature>
    </DriverLoad>
        <!-- Do not log process termination -->
       <ProcessTerminate onmatch="include" />
       <!-- Log network connection if the destination port equals 443 -->
       <!-- or 80, and process isn't InternetExplorer -->
    <NetworkConnect onmatch="include">
      <DestinationPort>443</DestinationPort>
```

```
      <DestinationPort>80</DestinationPort>
    </NetworkConnect>      <NetworkConnect onmatch="exclude">
       <Image condition="end with">iexplore.exe</Image>
    </NetworkConnect>
  </EventFiltering>
</Sysmon>
```

This example demonstrates:

- Global configuration entries
- Event-specific filtering
- Combined include and exclude rules to balance visibility and noise

Schema version

The configuration file contains a schemaversion attribute on the `<Sysmon>` tag. This version is independent of the Sysmon binary version and allows Sysmon to correctly parse older configuration files.

Schema versioning:

- Preserves compatibility across Sysmon updates
- Allows new capabilities to be added safely
- Ensures older configurations continue to function

Configuration entries

Configuration entries define global Sysmon behavior. They influence how events are enriched or handled rather than which event types are generated.

Expand table

Entry	Value	Default	Description
ArchiveDirectory	String	Sysmon	Directory at the root of each volume where deleted files are archived when ile-delete archiving is enabled. The directory is protected by a system ACL.
CheckRevocation	Boolean	True	Controls whether digital signature revocation checks are performed when validating signed binaries.

Entry	Value	Default	Description
CopyOnDeletePE	Boolean	False	Preserves deleted executable (PE) files by copying them to the archive directory.
CopyOnDeleteSIDs	Strings	None	Comma-separated list of account SIDs for which deleted files are preserved.
CopyOnDeleteExtensions	Strings	None	Process names for which deleted files are preserved.
DnsLookup	Boolean	True	Enables reverse DNS lookup for network events to resolve IP addresses to host names.
DriverName	String	Auto-generated	Specifies the name used for the Sysmon driver and service images.
HashAlgorithms	Strings	None	Hash algorithms applied to files. Supported values include MD5, SHA1, SHA256, IMPHASH, or * (all).

Configuration entries control event enrichment and handling, not event selection.

Configuration schema and event definition

Sysmon configuration is based on a well-defined schema that describes:

- Supported event types
- Available fields for filtering
- Field data types

This schema defines what configuration files can express and ensures consistent parsing across Sysmon versions.

Some configuration entries enable functionality (such as hashing or enrichment), but events must still be explicitly enabled and filtered under `<EventFiltering>`.

The following example shows the schema definition for the RawAccessRead event type:

XML

```
<event name="SYSMON_RAWACCESS_READ" value="9" level="Informational"
template="RawAccessRead detected" rulename="RawAccessRead" version="2">
  <data name="UtcTime" inType="win:UnicodeString" outType="xs:string"/>
  <data name="ProcessGuid" inType="win:GUID"/>
  <data name="ProcessId" inType="win:UInt32" outType="win:PID"/>
  <data name="Image" inType="win:UnicodeString" outType="xs:string"/>
```

```
  <data name="Device" inType="win:UnicodeString" outType="xs:string"/>
</event>
```

This illustrates that:

- Each event exposes a fixed set of fields
- Only schema-defined fields can be used in filtering rules
- Field types constrain valid filtering conditions

Event filtering

Event filtering controls which Sysmon events are logged. Each event type has its own filter tag under `<EventFiltering>`.

Expand table

ID Tag	Event	Description
1	ProcessCreate	Process create
2	FileCreateTime	File creation time
3	NetworkConnect	Network connection
4	n/a	Sysmon service state change (not filterable)
5	ProcessTerminate	Process terminated
6	DriverLoad	Driver loaded
7	ImageLoad	Image loaded
8	CreateRemoteThread	Remote thread creation
9	RawAccessRead	Raw disk access
10	ProcessAccess	Process accessed
11	FileCreate	File created
12–14	RegistryEvent	Registry modifications
15	FileCreateStreamHash	File stream created
16	n/a	Sysmon configuration change (not filterable)
17–18	PipeEvent	Named pipe activity

ID Tag	Event	Description
19–21	WmiEvent	WMI activity
22	DNSQuery	DNS query
23	FileDelete	File delete (archived)
24	ClipboardChange	Clipboard content change
25	ProcessTampering	Process image change
26	FileDeleteDetected	File delete (logged)
27–29	FileBlock / FileExecutable	Executable file events

Include and exclude rules

Each event filter specifies an onmatch attribute:

- include – only matching events are logged
- exclude – all events are logged except matching ones

Exclude rules always take precedence over include rules.

Rule logic and conditions

Rules reference event fields and apply conditions to them.

- Same-field rules behave as OR
- Different-field rules behave as AND

Expand table

Condition	Description
is	Exact match
contains	Substring match
begin with / end with	Prefix or suffix
image	Path-aware image matching
contains any / contains all	Multi-value matching

Condition	Description
excludes*	Negative matching
less than / more than	Lexicographical comparison

Named rules

Rules can be named to provide audit context:

XML

```xml
<NetworkConnect onmatch="exclude">
  <Image name="network iexplore" condition="contains">iexplore.exe</Image>
</NetworkConnect>
```

Named rules improve investigation clarity and maintainability.

Rule groups

Rule groups allow explicit control over logical rule combination:

XML

```xml
<EventFiltering>
  <RuleGroup name="group 1" groupRelation="and">
    <ProcessCreate onmatch="include">
      <Image condition="contains">timeout.exe</Image>
      <CommandLine condition="contains">100</CommandLine>
    </ProcessCreate>
  </RuleGroup>
  <RuleGroup groupRelation="or">
    <ProcessTerminate onmatch="include">
      <Image condition="contains">timeout.exe</Image>
      <Image condition="contains">ping.exe</Image>
    </ProcessTerminate>
  </RuleGroup>
  <ImageLoad onmatch="include"/>
</EventFiltering>
```

Rule groups allow:

- Precise behavioral matching
- Different logic per event type
- Reduced rule duplication

Configuration change visibility

Whenever the Sysmon configuration is updated, Sysmon generates a configuration change event, providing auditability and change tracking.

Design principles

Effective Sysmon configurations:

- Start broad and refine over time
- Exclude known benign activity
- Preserve investigative context
- Avoid over-specific rules

Once telemetry is excluded, it cannot be recovered retroactively.

What configuration files do not do

Configuration files:

- Do not detect threats
- Do not generate alerts
- Do not classify behavior
- Do not replace SIEM or EDR logic

They define what evidence exists, not how it is interpreted.

Summary

Sysmon configuration files are the foundation of effective Sysmon deployments. Understanding schema structure, filtering semantics, rule logic, and real-world patterns is essential to building configurations that scale and retain long-term investigative value.

Last updated on 02/24/2026

Understanding Sysmon events

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

Sysmon events provide detailed, low-level telemetry about system activity on Windows devices. Each event represents a specific class of behavior—such as process execution, network communication, file modification, or configuration change—that can be used to understand how software operates on a system over time.

This article explains what Sysmon events represent, how they relate to one another, and how to reason about them conceptually when investigating or hunting for malicious activity.

Where Sysmon events are recorded

On modern versions of Windows, Sysmon events are written to:

```
Applications and Services Logs -> Microsoft -> Windows -> Sysmon -> Operational
```

Event timestamps are recorded in UTC.

Because Sysmon is built into Windows, events are always written using the standard Windows Event Log infrastructure and can be collected using Windows Event Collection, Security Information and Event Management(SIEM) agents, or cloud-based log ingestion pipelines.

What a Sysmon event represents

A Sysmon event is a structured record of an observed system action. It describes:

- *What* happened (event type)
- *When* it happened (timestamp)
- *Which process* was involved
- *What context* surrounded the action (command line, file paths, network details)

Sysmon events are:

- Observational, not interpretive
- Deterministic, not probabilistic
- Composable, meaning they gain meaning when correlated

No single event indicates malicious activity by itself.

Events as behavioral building blocks

Sysmon events are most useful when viewed as behavioral signals that form sequences.

For example:

- Process Create — something started
- DNS Query— it resolved infrastructure
- Network Connect — it communicated externally
- File Create — it wrote artifacts
- Registry Event — it persisted

Understanding Sysmon means thinking in chains and timelines, not isolated events.

Core event categories

Sysmon events can be grouped conceptually by the type of system behavior they describe.

Process lifecycle and execution

These events describe how processes start, run, and interact.

Process Create (Event ID 1)

Records detailed information about a newly created process, including:

- Full command line
- Parent process
- Cryptographic hashes
- A globally unique **Process GUID** for correlation

This event is foundational for:

- Execution chain analysis

- Command-line inspection
- Threat hunting

Process Terminated (Event ID 5)

Records when a process exits. Useful for:

- Completing timelines
- Measuring execution duration
- Identifying short-lived processes

Process Access (Event ID 10)

Records when one process opens another process. This is commonly associated with:

- Credential theft
- Memory inspection
- Injection techniques

Because this event can be noisy, it's typically used with targeted filtering.

CreateRemoteThread (Event ID 8)

Indicates that a process created a thread in another process—a classic code injection technique.

This event is low-volume and high-signal.

Process Tampering (Event ID 25)

Generated when Sysmon detects process image manipulation techniques such as:

- Process hollowing
- Herpaderping

This event indicates attempts to disguise or replace process contents and is strongly associated with advanced malware.

Network and name resolution activity

These events describe how processes interact with the network.

Network Connect (Event ID 3)

Records TCP and UDP connections initiated by processes, including:

- Source and destination addresses
- Ports
- Protocols
- Associated Process GUID

This event is disabled by default due to volume and should be enabled selectively.

DNS Query (Event ID 22)

Records DNS queries issued by a process, regardless of whether the query succeeds.

DNS events are especially useful for:

- Identifying external infrastructure
- Detecting beaconing behavior
- Spotting malware such as cryptocurrency miners

DNS events often provide early indicators even when network traffic is encrypted.

File system activity

These events describe file creation, modification, and deletion.

File Create (Event ID 11)

Records when a file is created or overwritten.

Common uses:

- Detecting dropped payloads
- Monitoring startup and persistence locations
- Investigating malware staging

File Creation Time Change (Event ID 2)

Records when a process explicitly modifies a file's creation timestamp.

Attackers may use this to obscure the true origin of malicious files, though legitimate software also performs this action.

File Delete (Event ID 23 / 26)

Records file deletion activity.

- Event ID 23 also archives the deleted file
- Event ID 26 records deletion without archiving

These events are useful for ransomware and cleanup analysis.

File Create Stream Hash (Event ID 15)

Records named alternate data streams, including Zone.Identifier streams.

This event helps detect files originating from the internet and browser download activity.

Executable File Detection and Blocking (Events 27–29)

These events record when executable files are detected or blocked during creation.

They provide insight into executable staging behavior and defensive enforcement.

Module and driver loading

These events describe how code is introduced into memory.

Image Load (Event ID 7)

Records when a DLL or executable image is loaded into a process.

This event is powerful for:

- DLL injection detection
- Side-loading analysis

Because of its high volume, it should be configured carefully.

Driver Load (Event ID 6)

Records when a kernel driver is loaded, including signature and hash information.

Driver loading is rare and high-impact, making this event valuable for detecting kernel-level threats.

Configuration, persistence, and IPC

These events describe system configuration changes and interprocess communication.

Registry Events (Event IDs 12–14)

Record registry key and value creation, modification, deletion, and renaming.

they're essential for detecting:

- Persistence mechanisms
- Configuration tampering

- Malware setup behavior

Sysmon uses abbreviated root key names (for example, HKLM, HKCU) to normalize data.

Named Pipe Events (Event IDs 17–18)

Record creation and connection to named pipes.

Named pipes are frequently used for:

- Interprocess communication
- Malware coordination between components

WMI Events (Event IDs 19–21)

Record WMI filter, consumer, and binding registration.

WMI is a common persistence and execution mechanism for fileless malware.

Sysmon service and configuration events

Service State Change (Event ID 4)

Records when the Sysmon service starts or stops.

Unexpected stops can indicate tampering attempts.

Configuration Change (Event ID 16)

Records when the Sysmon configuration is updated.

This event provides auditability and change tracking for telemetry selection.

Error handling

Error (Event ID 255)

Indicates internal Sysmon errors, such as:

- Resource exhaustion
- Failed operations

- Internal bugs

These events should be monitored to ensure telemetry reliability.

Events as evidence, not alerts

A core principle of Sysmon is neutrality.

- Events don't indicate malicious intent
- Rare doesn't mean malicious
- Common doesn't mean safe

Meaning emerges through:

- Correlation
- Context
- Behavioral patterns over time

Summary

Sysmon events provide the raw behavioral telemetry needed to understand how systems operate and how attacks unfold. Each event captures a specific class of activity, and their true value comes from correlating them into timelines and behavioral narratives.

Well-configured and well-understood Sysmon events enable:

- Threat hunting
- Incident response
- Forensic analysis
- Long-term behavioral baselining

Last updated on 02/24/2026

Secure the Windows boot process

Applies to: ✅ Windows 11, ✅ Windows 10

Windows has many features to help protect you from malware, and it does an amazingly good job. Except for apps that businesses develop and use internally, all Microsoft Store apps must meet a series of requirements to be certified and included in the Microsoft Store. This certification process examines several criteria, including security, and is an effective means of preventing malware from entering the Microsoft Store. Even if a malicious app does get through, Windows includes a series of security features that can mitigate the effect. For instance, Microsoft Store apps are sandboxed and lack the privileges necessary to access user data or change system settings.

Windows has multiple levels of protection for desktop apps and data, too. Windows Defender Antivirus uses cloud-powered real-time detection to identify and quarantine apps that are known to be malicious. Windows Defender SmartScreen warns users before allowing them to run an untrustworthy app, even if it's recognized as malware. Before an app can change system settings, the user would have to grant the app administrative privileges by using User Account Control.

Those components are just some of the ways that Windows protects you from malware. However, those security features protect you only after Windows starts. Modern malware, and bootkits specifically, are capable of starting before Windows, completely bypassing OS security, and remaining hidden.

Running Windows 10 or Windows 11 on a PC with Unified Extensible Firmware Interface (UEFI) support ensures that Trusted Boot safeguards your PC against malware right from the moment you power it on. This protection continues until your anti-malware software takes over. If, by any chance, malware manages to infect your PC, it won't be able to stay hidden. Trusted Boot can verify the system's integrity to your infrastructure in a manner that malware can't mask. Even for PCs without UEFI, Windows offers enhanced startup security compared to earlier Windows versions.

To begin, let's take a closer look at rootkits and their functioning. Following that, we'll illustrate how Windows can ensure your protection.

The threat: rootkits

Rootkits are a sophisticated and dangerous type of malware. They run in kernel mode, using the same privileges as the OS. Because rootkits have the same rights as the OS and start before it, they can completely hide themselves and other applications. Often, rootkits are part of an

entire suite of malware that can bypass local logins, record passwords and keystrokes, transfer private files, and capture cryptographic data.

Different types of rootkits load during different phases of the startup process:

- **Firmware rootkits.** These kits overwrite the firmware of the PC's basic input/output system or other hardware so the rootkit can start before Windows.
- **Bootkits.** These kits replace the OS's bootloader (the small piece of software that starts the OS) so that the PC loads the bootkit before the OS.
- **Kernel rootkits.** These kits replace a portion of the OS kernel so the rootkit can start automatically when the OS loads.
- **Driver rootkits.** These kits pretend to be one of the trusted drivers that Windows uses to communicate with the PC hardware.

The countermeasures

Windows supports four features to help prevent rootkits and bootkits from loading during the startup process:

- **Secure Boot.** PCs with UEFI firmware and a Trusted Platform Module (TPM) can be configured to load only trusted OS bootloaders.
- **Trusted Boot.** Windows checks the integrity of every component of the startup process before loading it.
- **Early Launch Anti-Malware (ELAM).** ELAM tests all drivers before they load and prevents unapproved drivers from loading.
- **Measured Boot.** The PC's firmware logs the boot process, and Windows can send it to a trusted server that can objectively assess the PC's health.

Figure 1 shows the Windows startup process.

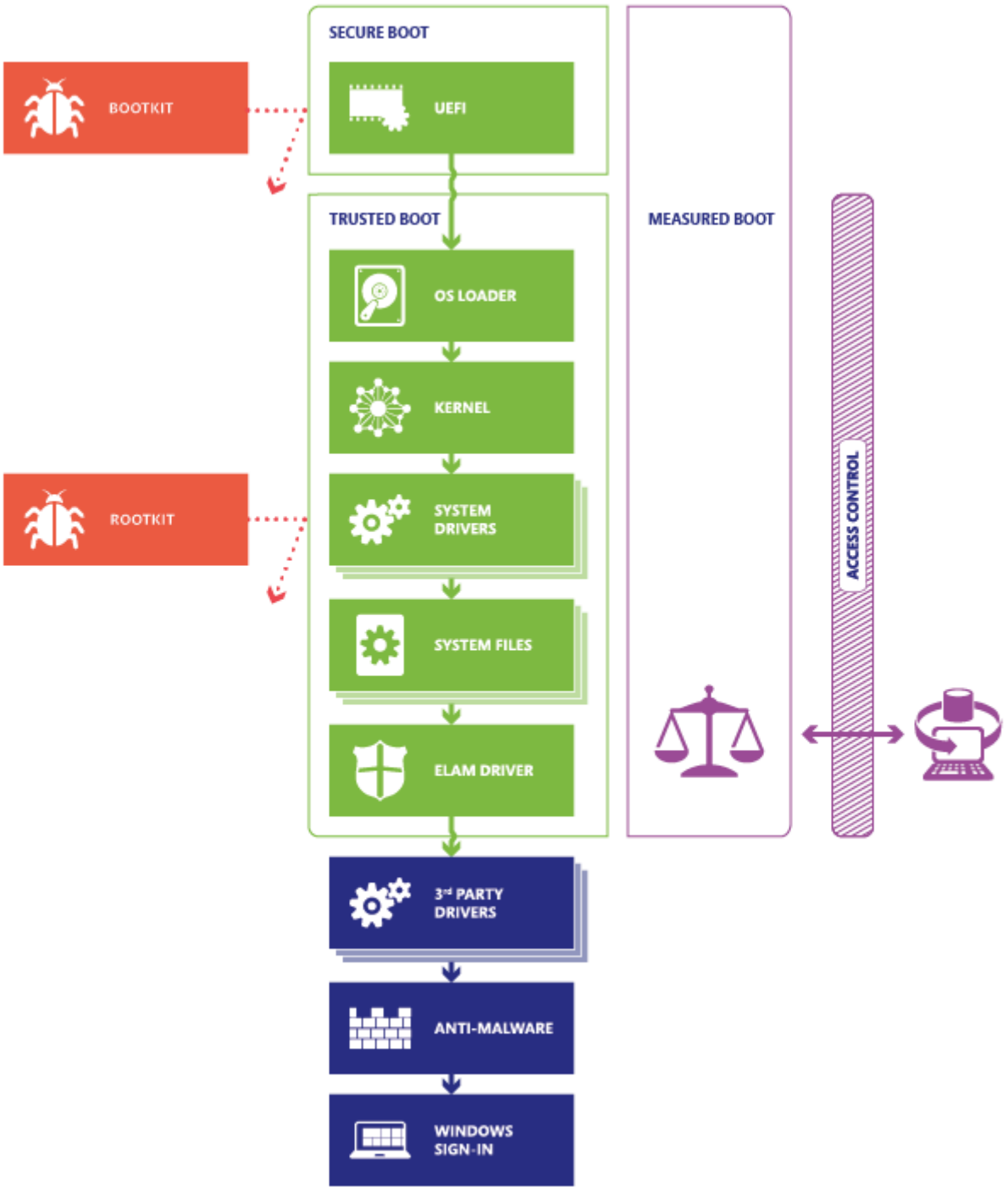


Figure 1. Secure Boot, Trusted Boot, and Measured Boot block malware at every stage:

Secure Boot and Measured Boot are only possible on PCs with UEFI 2.3.1 and a TPM chip. Fortunately, all Windows 10 and Windows 11 PCs that meet Windows Hardware Compatibility Program requirements have these components, and many PCs designed for earlier versions of Windows have them as well.

The sections that follow describe Secure Boot, Trusted Boot, ELAM, and Measured Boot.

Secure Boot

When a PC starts, it first finds the OS bootloader. PCs without Secure Boot run whatever bootloader is on the PC's hard drive. There's no way for the PC to tell whether it's a trusted OS or a rootkit.

When a PC equipped with UEFI starts, the PC first verifies that the firmware is digitally signed, reducing the risk of firmware rootkits. If Secure Boot is enabled, the firmware examines the bootloader's digital signature to verify that it hasn't been modified. If the bootloader is intact, the firmware starts the bootloader only if one of the following conditions is true:

- **The bootloader was signed using a trusted certificate.** For PCs certified for Windows, the Microsoft certificate is trusted.
- **The user has manually approved the bootloader's digital signature.** This action allows the user to load non-Microsoft operating systems.

All x86-based Certified For Windows PCs must meet several requirements related to Secure Boot:

- They must have Secure Boot enabled by default.
- They must trust Microsoft's certificate (and thus any bootloader Microsoft has signed).
- They must allow the user to configure Secure Boot to trust other bootloaders.
- They must allow the user to completely disable Secure Boot.

These requirements help protect you from rootkits while allowing you to run any OS you want. You have three options for running non-Microsoft operating systems:

- **Use an OS with a certified bootloader.** Because all Certified For Windows PCs must trust Microsoft's certificate, Microsoft offers a service to analyze and sign any non-Microsoft bootloader so that it will be trusted by all Certified For Windows PCs. In fact, an open source bootloader capable of loading Linux is already available. To begin the process of obtaining a certificate, go to https://partner.microsoft.com/dashboard.
- **Configure UEFI to trust your custom bootloader.** All Certified For Windows PCs allow you to trust a non-certified bootloader by adding a signature to the UEFI database, allowing you to run any OS, including homemade operating systems.
- **Turn off Secure Boot.** All *Certified For Windows* PCs allow you to turn off Secure Boot so that you can run any software. This action doesn't help protect you from bootkits, however.

To prevent malware from abusing these options, the user must manually configure the UEFI firmware to trust a non-certified bootloader or to turn off Secure Boot. Software can't change the Secure Boot settings.

The default state of Secure Boot has a wide circle of trust, which can result in customers trusting boot components they may not need. Since the Microsoft 3rd Party UEFI CA certificate signs the bootloaders for all Linux distributions, trusting the Microsoft 3rd Party UEFI CA

signature in the UEFI database increase s the attack surface of systems. A customer who intended to only trust and boot a single Linux distribution will trust all distributions - more than their desired configuration. A vulnerability in any of the bootloaders exposes the system and places the customer at risk of exploit for a bootloader they never intended to use, as seen in recent vulnerabilities, for example with the GRUB bootloader or firmware-level rootkit affecting boot components. Secured-core PCs require Secure Boot to be enabled and configured to distrust the Microsoft 3rd Party UEFI CA signature, by default, to provide customers with the most secure configuration of their PCs possible.

To trust and boot operating systems, like Linux, and components signed by the UEFI signature, Secured-core PCs can be configured in the BIOS menu to add the signature in the UEFI database by following these steps:

1. Open the firmware menu, either:
    - Boot the PC, and press the manufacturer's key to open the menus. Common keys used: Esc, Delete, F1, F2, F10, F11, or F12. On tablets, common buttons are Volume up or Volume down. During startup, there's often a screen that mentions the key. If there's not one, or if the screen goes by too fast to see it, check your manufacturer's site.
    - Or, if Windows is already installed, from either the Sign on screen or the Start menu, select Power ( ) > hold Shift while selecting Restart. Select Troubleshoot > Advanced options > UEFI Firmware settings.
2. From the firmware menu, navigate to Security > Secure Boot and select the option to trust the "3rd Party CA".
3. Save changes and exit.

Microsoft continues to collaborate with Linux and IHV ecosystem partners to design least privileged features to help you stay secure and opt-in trust for only the publishers and components you trust.

Like most mobile devices, Arm-based devices, such as the Microsoft Surface RT device, are designed to run only Windows 8.1. Therefore, Secure Boot can't be turned off, and you can't load a different OS. Fortunately, there's a large market of ARM processor devices designed to run other operating systems.

Trusted Boot

Trusted Boot takes over where Secure Boot ends. The bootloader verifies the digital signature of the Windows kernel before loading it. The Windows kernel, in turn, verifies every other component of the Windows startup process, including the boot drivers, startup files, and ELAM.

If a file has been modified, the bootloader detects the problem and refuses to load the corrupted component. Often, Windows can automatically repair the corrupted component, restoring the integrity of Windows and allowing the PC to start normally.

Early Launch anti-malware

Because Secure Boot has protected the bootloader and Trusted Boot has protected the Windows kernel, the next opportunity for malware to start is by infecting a non-Microsoft boot driver. Traditional anti-malware apps don't start until after the boot drivers have been loaded, giving a rootkit disguised as a driver the opportunity to work.

Early Launch anti-malware (ELAM) can load a Microsoft or non-Microsoft anti-malware driver before all non-Microsoft boot drivers and applications, thus continuing the chain of trust established by Secure Boot and Trusted Boot. Because the OS hasn't started yet, and because Windows needs to boot as quickly as possible, ELAM has a simple task: examine every boot driver and determine whether it is on the list of trusted drivers. If it's not trusted, Windows doesn't load it.

An ELAM driver isn't a full-featured anti-malware solution; that loads later in the boot process. Windows Defender (included with Windows) supports ELAM, as does several non-Microsoft anti-malware apps.

Measured Boot

If a PC in your organization does become infected with a rootkit, you need to know about it. Enterprise anti-malware apps can report malware infections to the IT department, but that doesn't work with rootkits that hide their presence. In other words, you can't trust the client to tell you whether it's healthy.

As a result, PCs infected with rootkits appear to be healthy, even with anti-malware running. Infected PCs continue to connect to the enterprise network, giving the rootkit access to vast amounts of confidential data and potentially allowing the rootkit to spread across the internal network.

Measured Boot works with the TPM and non-Microsoft software in Windows. It allows a trusted server on the network to verify the integrity of the Windows startup process. Measured Boot uses the following process:

1. The PC's UEFI firmware stores in the TPM a hash of the firmware, bootloader, boot drivers, and everything that is loaded before the anti-malware app.
2. At the end of the startup process, Windows starts the non-Microsoft remote attestation client. The trusted attestation server sends the client a unique key.

3. The TPM uses the unique key to digitally sign the log recorded by the UEFI.
4. The client sends the log to the server, possibly with other security information.

Depending on the implementation and configuration, the server can now determine whether the client is healthy. It can grant the client access to either a limited quarantine network or to the full network.

Figure 2 illustrates the Measured Boot and remote attestation process.

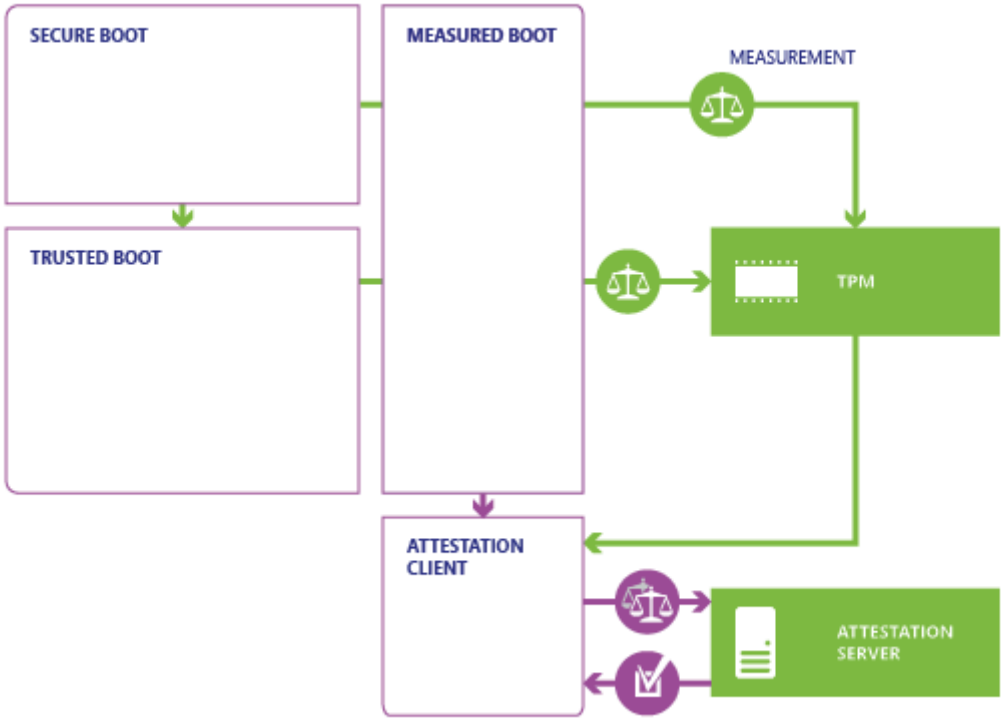


Figure 2. Measured Boot proves the PC's health to a remote server:

Windows includes the application programming interfaces to support Measured Boot. However, to take advantage of it, you need non-Microsoft tools to implement a remote attestation client and trusted attestation server. For example, see the following tools from Microsoft Research:

- TPM Platform Crypto-Provider Toolkit
- TSS.MSR

Measured Boot uses the power of UEFI, TPM, and Windows to give you a way to confidently assess the trustworthiness of a client PC across the network.

Summary

Secure Boot, Trusted Boot, and Measured Boot create an architecture that is fundamentally resistant to bootkits and rootkits. In Windows, these features have the potential to eliminate kernel-level malware from your network. With Windows, you can trust the integrity of your OS.

Last updated on 08/18/2025

Secure Boot and Trusted Boot

Applies to: ✅ Windows 11

This article describes Secure Boot and Trusted Boot, security measures built into Windows 11.

Secure Boot and Trusted Boot help prevent malware and corrupted components from loading when a Windows 11 device is starting. Secure Boot starts with initial boot-up protection, and then Trusted Boot picks up the process. Together, Secure Boot and Trusted Boot help to ensure your Windows 11 system boots up safely and securely.

Secure Boot

The first step in protecting the operating system is to ensure that it boots securely after the initial hardware and firmware boot sequences safely finish their early boot sequences. Secure Boot makes a safe and trusted path from the Unified Extensible Firmware Interface (UEFI) through the Windows kernel's Trusted Boot sequence. Malware attacks on the Windows boot sequence are blocked by the signature-enforcement handshakes throughout the boot sequence between the UEFI, bootloader, kernel, and application environments.

As the PC begins the boot process, it first verifies that the firmware is digitally signed, reducing the risk of firmware rootkits. Secure Boot then checks all code that runs before the operating system, and checks the OS bootloader's digital signature to ensure that it's trusted by the Secure Boot policy and hasn't been tampered with.

Trusted Boot

Trusted Boot picks up the process that started with Secure Boot. The Windows bootloader verifies the digital signature of the Windows kernel before loading it. The Windows kernel, in turn, verifies every other component of the Windows startup process, including boot drivers, startup files, and your anti-malware product's early-launch anti-malware (ELAM) driver. If any of these files were tampered, the bootloader detects the problem and refuses to load the corrupted component. Tampering or malware attacks on the Windows boot sequence are blocked by the signature-enforcement handshakes between the UEFI, bootloader, kernel, and application environments.

Often, Windows can automatically repair the corrupted component, restoring the integrity of Windows and allowing the Windows 11 device to start normally.

Windows edition and licensing requirements

The following table lists the Windows editions that support Secure Boot and Trusted Boot:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Secure Boot and Trusted Boot license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

See also

Secure the Windows boot process

Last updated on 08/18/2025

Measured Boot

Article11/17/2021

Platforms

Clients - Windows 8 **Servers** - Windows Server 2012

Description

As antimalware (AM) software has become better and better at detecting runtime malware, attackers are also becoming better at creating rootkits that can hide from detection. Detecting malware that starts early in the boot cycle is a challenge that most AM vendors address diligently. Typically, they create system hacks that are not supported by the host operating system and can actually result in placing the computer in an unstable state. Up to this point, Windows has not provided a good way for AM to detect and resolve these early boot threats. Windows 8 introduces a new feature called Measured Boot, which measures each component, from firmware up through the boot start drivers, stores those measurements in the Trusted Platform Module (TPM) on the machine, and then makes available a log that can be tested remotely to verify the boot state of the client.

Manifestation

The Measured Boot feature provides AM software with a trusted (resistant to spoofing and tampering) log of all boot components that started before AM software. AM software can use the log to determine whether components that ran before it are trustworthy or if they are infected with malware. The AM software on the local machine can send the log to a remote sever for evaluation. The remote server may initiate remediation actions either by interacting with software on the client or through out-of-band mechanisms, as appropriate.

Mitigation

In enterprise scenarios, the system administrator has control of how Measured Boot info is used. In end-user scenarios for example, online banking), the consumer must opt in to use Measured Boot for the specific service.

Solution

A white paper is being prepared that details the APIs and function calls that must be made for various Measured Boot scenarios.

Control the health of Windows devices

Applies to: ✅ Windows 11, ✅ Windows 10

This article details an end-to-end solution that helps you protect high-value assets by enforcing, controlling, and reporting the health of Windows devices.

Introduction

For Bring Your Own Device (BYOD) scenarios, users bring commercially available devices to access both work-related resources and their personal data. Users want to use the device of their choice to access the organization's applications, data, and resources not only from the internal network but also from anywhere. This phenomenon is also known as the consumerization of IT.

Users want to have the best productivity experience when accessing corporate applications and working on organization data from their devices. That means they don't tolerate being prompted to enter their work credentials each time they access an application or a file server. From a security perspective, it also means that users manipulate corporate credentials and corporate data on unmanaged devices.

With the increased use of BYOD, there will be more unmanaged and potentially unhealthy systems accessing corporate services, internal resources, and cloud apps.

Even managed devices can be compromised and become harmful. Organizations need to detect when security has been breached and react as early as possible in order to protect high-value assets.

As Microsoft moves forward, security investments are increasingly focused on security preventive defenses and also on detection and response capabilities.

Windows is an important component of an end-to-end security solution that focuses not only on the implementation of security preventive defenses, but adds device health attestation capabilities to the overall security strategy.

Description of a robust end-to-end security solution

Today's computing threat landscape is increasing at a speed never encountered before. The sophistication of criminal attacks is growing, and there's no doubt that malware now targets both consumers and professionals in all industries.

During recent years, one particular category of threat has become prevalent: advanced persistent threats (APTs). The term APT is commonly used to describe any attack that seems to target individual organizations on an ongoing basis. In fact, this type of attack typically involves determined adversaries who may use any methods or techniques necessary.

With the BYOD phenomena, a poorly maintained device represents a target of choice. For an attacker, it's an easy way to breach the security network perimeter, gain access to, and then steal high-value assets.

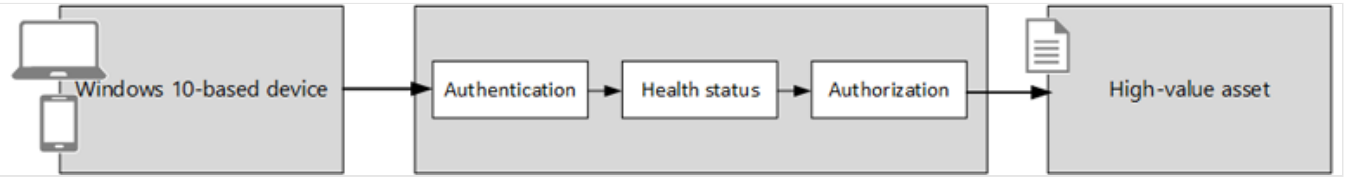
The attackers target individuals, not specifically because of who they are, but because of who they work for. An infected device brings malware into an organization, even if the organization has hardened the perimeter of networks or has invested in its defensive posture. A defensive strategy isn't sufficient against these threats.

A different approach

Rather than the traditional focus on the prevention of compromise, an effective security strategy assumes that determined adversaries will successfully breach any defenses. It means that it's necessary to shift focus away from preventative security controls to detection of, and response to, security issues. The implementation of the risk management strategy, therefore, balances investment in prevention, detection, and response.

Because mobile devices are increasingly being used to access corporate information, some way to evaluate device security or health is required. This section describes how to provision device health assessment in such a way that high-value assets can be protected from unhealthy devices.

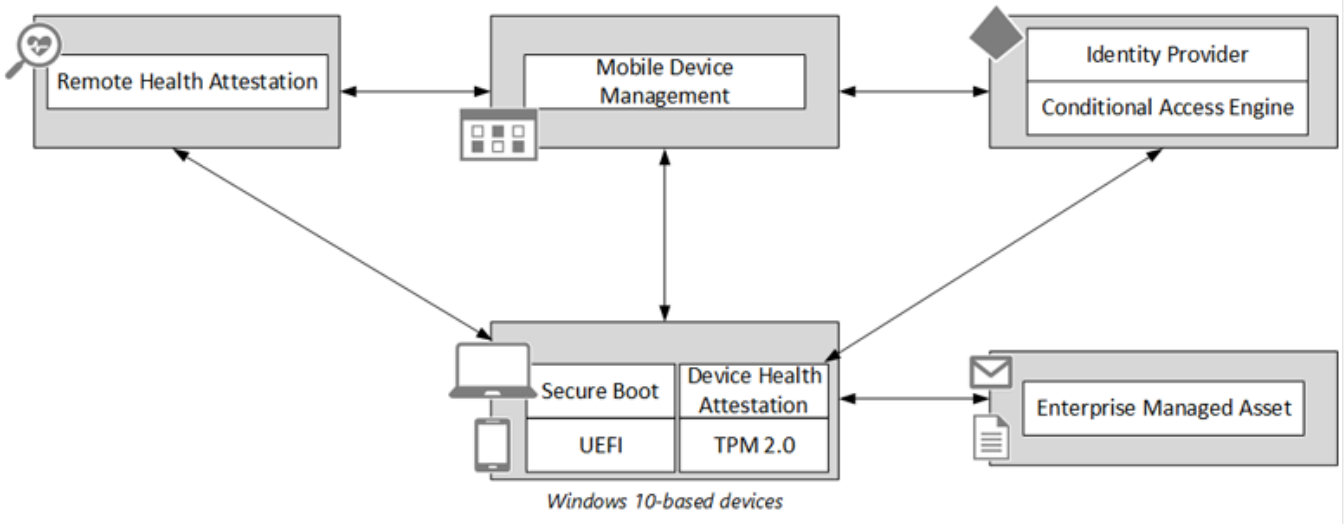
Devices that are used to access corporate resources must be trusted. An efficient end-to-end security approach is able to evaluate device health and use the current security state when granting access to a high-value asset.



A robust design needs to establish the user's identity, strengthen the authentication method if needed, and learn behavior like the network location the user regularly connects from. Also, a modern approach must be able to release sensitive content only if user devices are determined to be healthy and secure.

The following figure shows a solution built to assess device health from the cloud. The device authenticates the user through a connection to an identity provider in the cloud. If the

managed asset contains highly confidential information, the conditional access engine of the identity provider may elect to verify the security compliance of the mobile device before access is granted. The user's device is able to prove its health status that can be sent at any time or when mobile device management (MDM) requests it.



Windows devices can be protected from low-level rootkits and bootkits by using low-level hardware technologies such as Unified Extensible Firmware Interface (UEFI) Secure Boot.

Secure Boot is a firmware validation process that helps prevent rootkit attacks; it's part of the UEFI specification. The intent of UEFI is to define a standard way for the operating system to communicate with modern hardware, which can perform faster and with more efficient input/output (I/O) functions than older, software interrupt-driven BIOS systems.

A device health attestation module can communicate measured boot data that is protected by a Trusted Platform Module (TPM) to a remote service. After the device successfully boots, boot process measurement data is sent to a trusted cloud service (Health Attestation Service) using a more secure and tamper-resistant communication channel.

Remote health attestation service performs a series of checks on the measurements. It validates security related data points, including boot state (Secure Boot, Debug Mode, and so on), and the state of components that manage security (BitLocker, Device Guard, and so on). It then conveys the health state of the device by sending a health encrypted blob back to the device.

An MDM solution typically applies configuration policies and deploys software to devices. MDM defines the security baseline and knows the level of compliance of the device with regular checks to see what software is installed and what configuration is enforced, and determining the health status of the device.

An MDM solution asks the device to send device health information and forward the health encrypted blob to the remote health attestation service. The remote health attestation service verifies device health data, checks that MDM is communicating to the same device, and then issues a device health report back to the MDM solution.

An MDM solution evaluates the health assertions and, depending on the health rules belonging to the organization, can decide if the device is healthy. If the device is healthy and compliant, MDM passes that information to the identity provider so the organization's access control policy can be invoked to grant access.

Access to content is then authorized to the appropriate level of trust for whatever the health status and other conditional elements indicate.

Depending on the requirements and the sensitivity of the managed asset, device health status can be combined with user identity information when processing an access request. Access to content is then authorized to the appropriate level of trust. The Conditional Access engine may be structured to allow more verification as needed by the sensitivity of the managed asset. For example, if access to high-value data is requested, further security authentication may need to be established by querying the user to answer a phone call before access is granted.

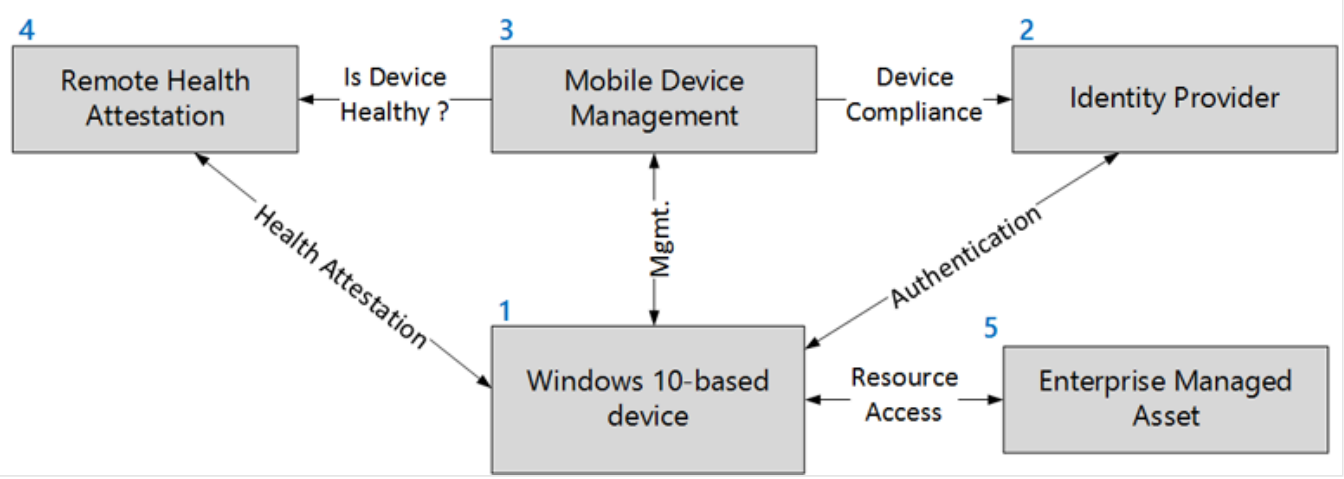
Microsoft's security investments in Windows

In Windows, there are three pillars of investments:

- **Secure identities.** Microsoft is part of the FIDO alliance that aims to provide an interoperable method of secure authentication by moving away from the use of passwords for authentication, both on the local system and for services like on-premises resources and cloud resources.
- **Information protection.** Microsoft is making investments to allow organizations to have better control over who has access to important data and what they can do with that data. With Windows, organizations can take advantage of policies that specify which applications are considered to be corporate applications and can be trusted to access secure data.
- **Threat resistance.** Microsoft is helping organizations to better secure enterprise assets against the threats of malware and attacks by using security defenses relying on hardware.

Protect, control, and report on the security status of Windows-based devices

This section is an overview that describes different parts of the end-to-end security solution that helps protect high-value assets and information from attackers and malware.



Expand table

Number	Part of the solution	Description
1	Windows-based device	The first time a Windows-based device is powered on, the out-of-box experience (OOBE) screen is displayed. During setup, the device can be automatically registered into Microsoft Entra ID and enrolled in MDM. A Windows-based device with TPM can report health status at any time by using the Health Attestation Service available with all supported editions of Windows.
2	Identity provider	Microsoft Entra ID contains users, registered devices, and registered application of organization's tenant. A device always belongs to a user and a user can have multiple devices. A device is represented as an object with different attributes like the compliance status of the device. A trusted MDM can update the compliance status. Microsoft Entra ID is more than a repository. Microsoft Entra ID is able to authenticate users and devices and can also authorize access to managed resources. Microsoft Entra ID has a conditional access control engine that uses the identity of the user, the location of the device and also the compliance status of the device when making a trusted access decision.
3	Mobile device management	Windows has MDM support that enables the device to be managed out-of-box without deploying any agent. MDM can be Microsoft Intune or any third-party MDM solution that is compatible with Windows.
4	Remote health attestation	The Health Attestation Service is a trusted cloud service operated by Microsoft that performs a series of health checks and reports to MDM what Windows security features are enabled on the device. Security verification includes boot state (WinPE, Safe Mode, Debug/test modes) and components that manage security and integrity of runtime operations (BitLocker, Device Guard).
5	Enterprise managed asset	Enterprise managed asset is the resource to protect. For example, the asset can be Office 365, other cloud apps, on-premises

Number	Part of the solution	Description
		web resources published by Microsoft Entra ID, or even VPN access.

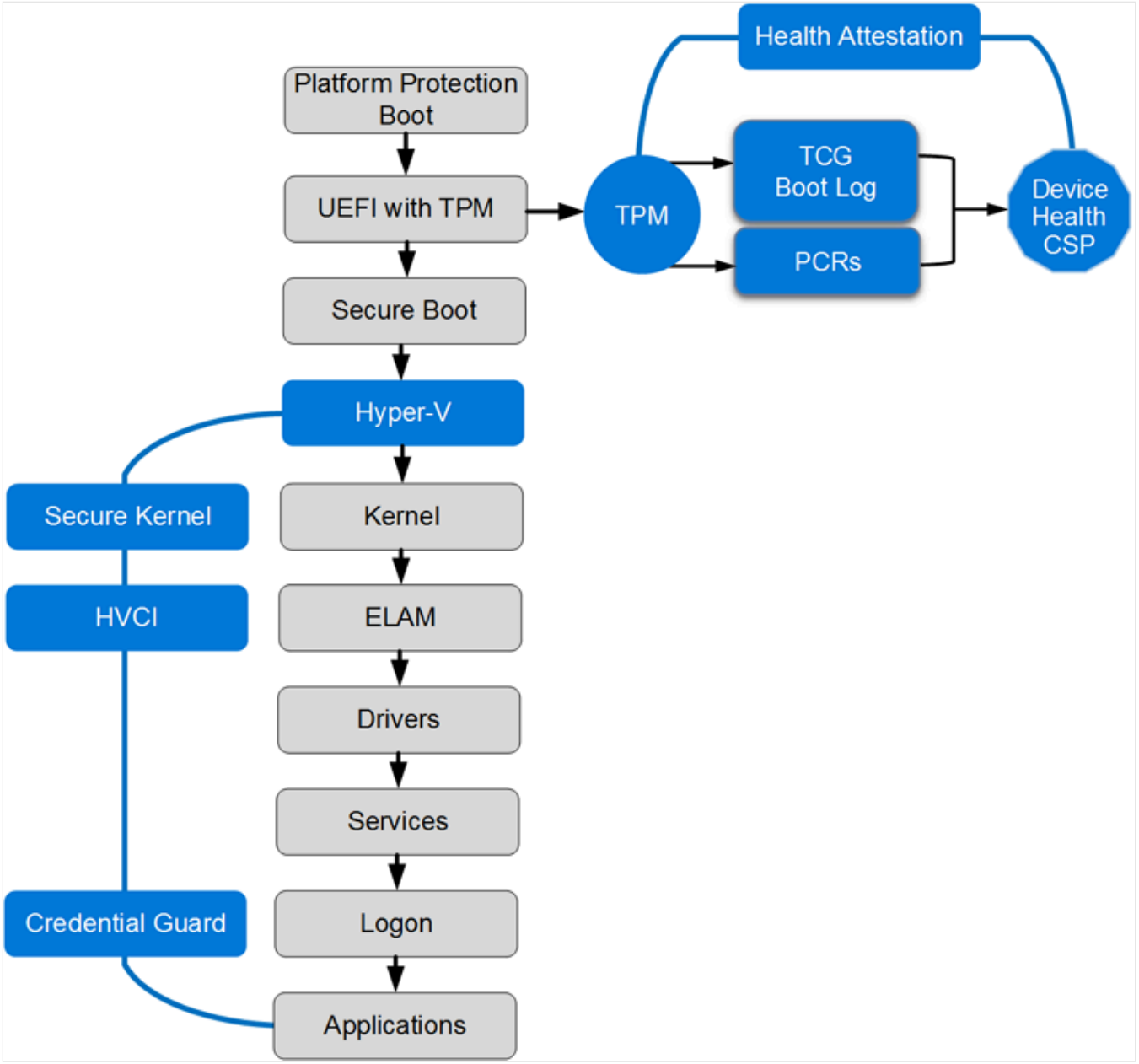
The combination of Windows-based devices, identity provider, MDM, and remote health attestation creates a robust end-to-end-solution that provides validation of health and compliance of devices that access high-value assets.

Protect devices and enterprise credentials against threats

This section describes what Windows offers in terms of security defenses and what control can be measured and reported to.

Windows hardware-based security defenses

The most aggressive forms of malware try to insert themselves into the boot process as early as possible so that they can take control of the operating system early and prevent protection mechanisms and anti-malware software from working. This type of malicious code is often called a rootkit or bootkit. The best way to avoid having to deal with low-level malware is to secure the boot process so that the device is protected from the very start. Windows supports multiple layers of boot protection. Some of these features are available only if specific types of hardware are installed. For more information, see the Hardware requirements section.



Windows supports features to help prevent sophisticated low-level malware like rootkits and bootkits from loading during the startup process:

- **Trusted Platform Module.** A Trusted Platform Module (TPM) is a hardware component that provides unique security features.

  Windows uses security characteristics of a TPM for measuring boot integrity sequence (and based on that, unlocking automatically BitLocker protected drives), for protecting credentials or for health attestation.

  A TPM implements controls that meet the specification described by the Trusted Computing Group (TCG). At the time of this writing, there are two versions of TPM specification produced by TCG that aren't compatible with each other:
  - The first TPM specification, version 1.2, was published in February 2005 by the TCG and standardized under ISO / IEC 11889 standard.

    - The latest TPM specification, referred to as TPM 2.0, was released in April 2014 and has been approved by the ISO/IEC Joint Technical Committee (JTC) as ISO/IEC 11889:2015.

  Windows uses the TPM for cryptographic calculations as part of health attestation and to protect the keys for BitLocker, Windows Hello, virtual smart cards, and other public key certificates. For more information, see TPM requirements in Windows.

  Windows recognizes versions 1.2 and 2.0 TPM specifications produced by the TCG. For the most recent and modern security features, Windows supports only TPM 2.0.

  TPM 2.0 provides a major revision to the capabilities over TPM 1.2:

    - Update crypto strength to meet modern security needs
      - Support for SHA-256 for PCRs
      - Support for HMAC command

    - Cryptographic algorithms flexibility to support government needs
      - TPM 1.2 is severely restricted in terms of what algorithms it can support
      - TPM 2.0 can support arbitrary algorithms with minor updates to the TCG specification documents

    - Consistency across implementations
      - The TPM 1.2 specification allows vendors wide latitude when choosing implementation details
      - TPM 2.0 standardizes much of this behavior

- **Secure Boot.** Devices with UEFI firmware can be configured to load only trusted operating system bootloaders. Secure Boot doesn't require a TPM.

  The most basic protection is the Secure Boot feature, which is a standard part of the UEFI 2.2+ architecture. On a PC with conventional BIOS, anyone who can take control of the boot process can boot by using an alternative OS loader, and potentially gain access to system resources. When Secure Boot is enabled, you can boot using only an OS loader that's signed using a certificate stored in the UEFI Secure Boot DB. Naturally, the Microsoft certificate used to digitally sign the Windows OS loaders are in that store, which allows UEFI to validate the certificate as part of its security policy. Secure Boot must be enabled by default on all computers that are certified for Windows under the Windows Hardware Compatibility Program.

  Secure Boot is a UEFI firmware-based feature, which allows for the signing and verification of critical boot files and drivers at boot time. Secure Boot checks signature values of the Windows Boot Manager, BCD store, Windows OS loader file, and other boot critical DLLs at boot time before the system is allowed to fully boot into a usable operating system by using policies that are defined by the OEM at build time. Secure

Boot prevents many types of boot-based rootkit, malware, and other security-related attacks against the Windows platform. Secure Boot protects the operating system boot process whether booting from local hard disk, USB, PXE, or DVD, or into full Windows or Windows Recovery Environment (RE). Secure Boot protects the boot environment of a Windows installation by verifying the signatures of the critical boot components to confirm malicious activity didn't compromise them. Secure Boot protection ends after the Windows kernel file (ntoskrnl.exe) has been loaded.

> ⓘ **Note**
>
> Secure Boot protects the platform until the Windows kernel is loaded. Then protections like ELAM take over.

- **Secure Boot configuration policy.** Extends Secure Boot functionality to critical Windows configuration.

  Examples of protected configuration information include protecting Disable Execute bit (NX option) or ensuring that the test signing policy (code integrity) can't be enabled. This protective action ensures that the binaries and configuration of the computer can be trusted after the boot process has completed. Secure Boot configuration policy does this protective action with UEFI policy. These signatures for these policies are signed in the same way that operating system binaries are signed for use with Secure Boot.

  The Secure Boot configuration policy must be signed by a private key that corresponds to one of the public keys stored in the Key Exchange Key (KEK) list. The Microsoft Certificate Authority (CA) will be present in the KEK list of all Windows certified Secure Boot systems. By default, a policy signed by the Microsoft KEK shall be work on all Secure Boot systems. BootMgr must verify the signature against the KEK list before applying a signed policy. With Windows, the default Secure Boot configuration policy is embedded in bootmgr.

  The bootloader verifies the digital signature of the Windows kernel before loading it. The Windows kernel, in turn, verifies every other component of the Windows startup process, including the boot drivers, startup files, and the ELAM component. This step is important and protects the rest of the boot process by verifying that all Windows boot components have integrity and can be trusted.

- **Early Launch Antimalware (ELAM).** ELAM tests all drivers before they load and prevents unapproved drivers from loading.

  Traditional anti-malware apps don't start until after the boot drivers have been loaded, which gives a rootkit that is disguised as a driver the opportunity to work. ELAM is a Windows mechanism introduced in a previous version of Windows that allows anti-

malware software to run early in the boot sequence. Thus, the anti-malware component is the first third-party component to run and control the initialization of other boot drivers until the Windows operating system is operational. When the system is started with a complete runtime environment (network access, storage, and so on), then a full-featured anti-malware is loaded.

ELAM can load a Microsoft or non-Microsoft anti-malware driver before all non-Microsoft boot drivers and applications, thus continuing the chain of trust established by Secure Boot and Trusted Boot. Because the operating system hasn't started yet, and because Windows needs to boot as quickly as possible, ELAM has a simple task: Examine every boot driver and determine whether it is on the list of trusted drivers. If it's not trusted, Windows won't load it.

> **Note**
>
> Windows Defender, Microsoft's antimalware included by default in Windows, supports ELAM; it can be replaced with a third-party antimalware compatible solution. The name of the Windows Defender ELAM driver is WdBoot.sys. Windows Defender uses its ELAM driver to roll back any malicious changes made to the Windows Defender driver at the next reboot. This prevents kernel mode malware making lasting changes to Windows Defender's mini-filter driver before shutdown or reboot.

The ELAM signed driver is loaded before any other third-party drivers or applications, which allows the anti-malware software to detect and block any attempts to tamper with the boot process by trying to load unsigned or untrusted code.

The ELAM driver is a small driver with a small policy database that has a narrow scope, focused on drivers that are loaded early at system launch. The policy database is stored in a registry hive that is also measured to the TPM, to record the operational parameters of the ELAM driver. An ELAM driver must be signed by Microsoft and the associated certificate must contain the complementary EKU (1.3.6.1.4.1.311.61.4.1).

- **Virtualization-based security (Hyper-V + Secure Kernel).** Virtualization-based security is a new enforced security boundary that allows you to protect critical parts of Windows.

  Virtualization-based security isolates sensitive code like Kernel Mode Code Integrity or sensitive corporate domain credentials from the rest of the Windows operating system. For more information, see Virtualization-based security section.

- **Hypervisor-protected Code Integrity (HVCI).** Hypervisor-protected Code Integrity is a feature of Device Guard that ensures only drivers, executables, and DLLs that comply with

the Device Guard Code Integrity policy are allowed to run.

When enabled and configured, Windows can start the Hyper-V Virtualization-based security services. HVCI helps protect the system core (kernel), privileged drivers, and system defenses, like anti-malware solutions, by preventing malware from running early in the boot process, or after startup.

HVCI uses Virtualization-based security to isolate Code Integrity, the only way kernel memory can become executable is through a Code Integrity verification. This dependency on verification means that kernel memory pages can never be Writable and Executable (W+X) and executable code can't be directly modified.

> ⓘ **Note**
>
> Device Guard devices that run Kernel Mode Code Integrity with virtualization-based security must have compatible drivers. For additional information, please read the Driver compatibility with Device Guard in Windows blog post.

The Device Guard Code Integrity feature lets organizations control what code is trusted to run into the Windows kernel and what applications are approved to run in user mode. It's configurable by using a policy. Device Guard Code Integrity policy is a binary file that Microsoft recommends you sign. The signing of the Code Integrity policy aids in the protection against a malicious user with Administrator privileges trying to modify or remove the current Code Integrity policy.

- **Credential Guard.** Credential Guard protects corporate credentials with hardware-based credential isolation.

  In Windows, Credential Guard aims to protect domain corporate credentials from theft and reuse by malware. With Credential Guard, Windows implemented an architectural change that fundamentally prevents the current forms of the pass-the-hash (PtH) attack.

  This attack-free state is accomplished by using Hyper-V and the new Virtualization-based security feature to create a protected container where trusted code and secrets are isolated from the Windows kernel. This accomplishment means that even if the Windows kernel is compromised, an attacker has no way to read and extract the data required to initiate a PtH attack. Credential Guard prevents this unauthorized access because the memory where secrets are stored is no longer accessible from the regular OS, even in kernel mode - the hypervisor controls who can access the memory.

- **Health attestation.** The device's firmware logs the boot process, and Windows can send it to a trusted server that can check and assess the device's health.

Windows takes measurements of the UEFI firmware and each of the Windows and anti-malware components are made as they load during the boot process. Additionally, they're taken and measured sequentially, not all at once. When these measurements are complete, their values are digitally signed and stored securely in the TPM and can't be changed unless the system is reset.

For more information, see Secured Boot and Measured Boot: Hardening Early Boot Components Against Malware.

During each subsequent boot, the same components are measured, which allows comparison of the measurements against an expected baseline. For more security, the values measured by the TPM can be signed and transmitted to a remote server, which can then perform the comparison. This process, called *remote device health attestation*, allows the server to verify health status of the Windows device.

Although Secure Boot is a proactive form of protection, health attestation is a reactive form of boot protection. Health attestation ships disabled in Windows and is enabled by an anti-malware or an MDM vendor. Unlike Secure Boot, health attestation won't stop the boot process and enter remediation when a measurement doesn't work. But with conditional access control, health attestation helps to prevent access to high-value assets.

Virtualization-based security

Virtualization-based security provides a new trust boundary for Windows and uses Hyper-V hypervisor technology to enhance platform security. Virtualization-based security provides a secure execution environment to run specific Windows trusted code (trustlet) and to protect sensitive data.

Virtualization-based security helps to protect against a compromised kernel or a malicious user with Administrator privileges. Virtualization-based security isn't trying to protect against a physical attacker.

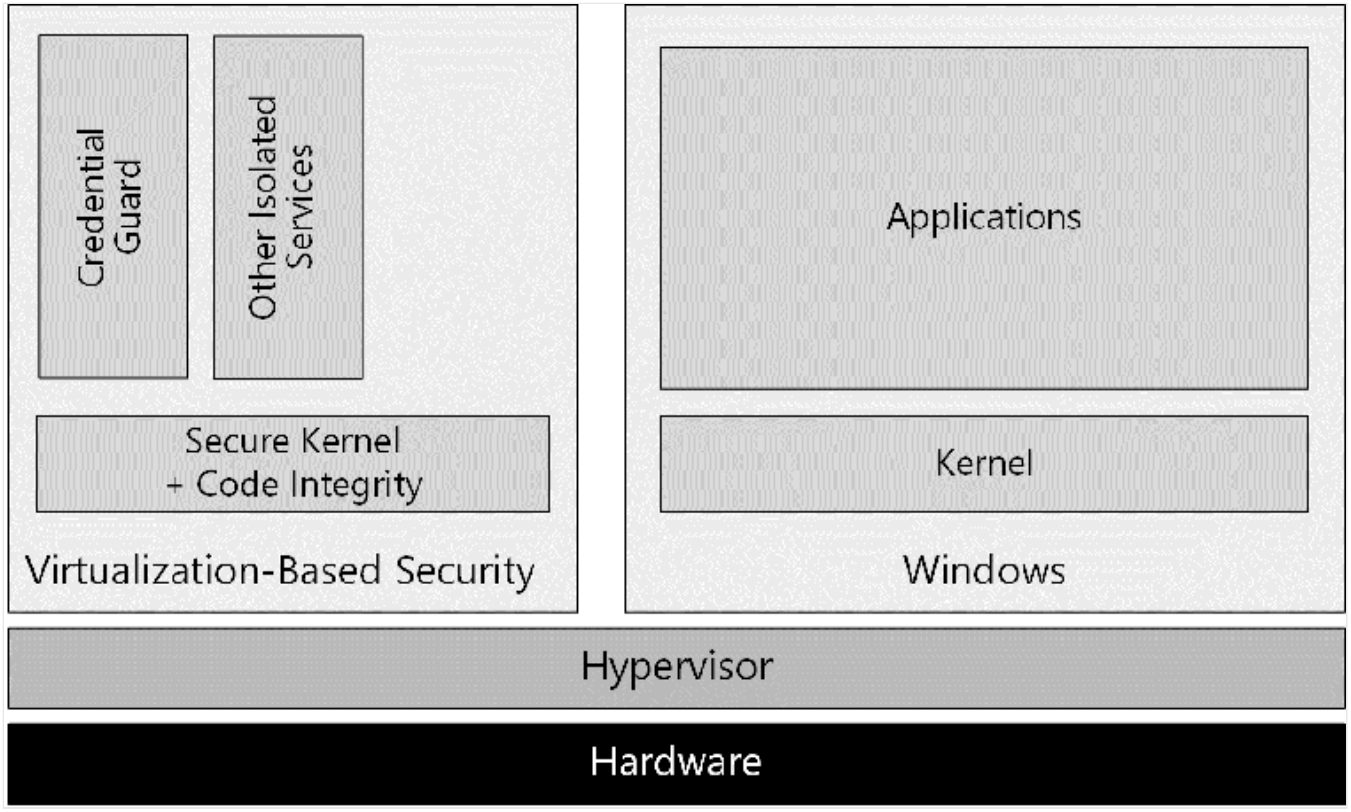
The following Windows services are protected with Virtualization-based security:

- **Credential Guard** (LSA Credential Isolation): prevents pass-the-hash attacks and enterprise credential theft that happens by reading and dumping the content of lsass memory
- **Device Guard** (Hyper-V Code Integrity): Device Guard uses the new Virtualization-based security in Windows to isolate the Code Integrity service from the Windows kernel itself, which lets the service use signatures defined by your enterprise-controlled policy to help determine what is trustworthy. In effect, the Code Integrity service runs alongside the kernel in a Windows hypervisor-protected container.

- **Other isolated services**: for example, on Windows Server 2016, there's the vTPM feature that allows you to have encrypted virtual machines (VMs) on servers.

> ⓘ **Note**
>
> Virtualization-based security is only available with Enterprise edition. Virtualization-based security requires devices with UEFI (2.3.1 or higher) with Secure Boot enabled, x64 processor with Virtualization Extensions and SLAT enabled. IOMMU, TPM 2.0. and support for Secure Memory overwritten are optional, but recommended.

The schema below is a high-level view of Windows with Virtualization-based security.



Credential Guard

In Windows, when Credential Guard is enabled, Local Security Authority Subsystem Service (lsass.exe) runs a sensitive code in an Isolated user mode to help protect data from malware that may be running in the normal user mode. This code execution helps ensure that protected data isn't stolen and reused on remote machines, which mitigates many PtH-style attacks.

Credential Guard helps protect credentials by encrypting them with either a per-boot or persistent key:

- **The per-boot key** is used for any in-memory credentials that don't require persistence. An example of such a credential would be a ticket-granting ticket (TGT) session key. This key

is negotiated with a Key Distribution Center (KDC) every time authentication occurs and is protected with a per-boot key.
- **The persistent key**, or some derivative, is used to help protect items that are stored and reloaded after a reboot. Such protection is intended for long-term storage, and must be protected with a consistent key. Credential Guard is activated by a registry key and then enabled by using a UEFI variable. This activation is done to protect against remote modifications of the configuration. The use of a UEFI variable implies that physical access is required to change the configuration. When lsass.exe detects that credential isolation is enabled, it then spawns LsaIso.exe as an isolated process, which ensures that it runs within isolated user mode. The startup of LsaIso.exe is performed before initialization of a security support provider, which ensures that the secure mode support routines are ready before any authentication begins.

Device Guard

Device Guard is a feature of Windows Enterprise that allows organizations to lock down a device to help protect it from running untrusted software. In this configuration, the only applications allowed to run are those applications that are trusted by the organization.

The trust decision to execute code is performed by using Hyper-V Code Integrity, which runs in Virtualization-based security, a Hyper-V protected container that runs alongside regular Windows.

Hyper-V Code Integrity is a feature that validates the integrity of a driver or system file each time it's loaded into memory. Code integrity detects whether an unsigned driver or system file is being loaded into the kernel, or whether a system file has been modified by malicious software that is being run by a user account with Administrator privileges. On x64-based versions of Windows, kernel-mode drivers must be digitally signed.

> ⓘ **Note**
>
> Independently of activation of Device Guard Policy, Windows drivers must be signed by Microsoft, and more specifically, by the WHQL (Windows Hardware Quality Labs) portal. Additionally, starting in October 2015, the WHQL portal will only accept driver submissions, including both kernel and user mode driver submissions, that have a valid Extended Validation ("EV") Code Signing Certificate.

With Device Guard, organizations are now able to define their own Code Integrity policy for use on x64 systems running Windows Enterprise. Organizations have the ability to configure the policy that determines what is trusted to run. These include drivers and system files, and

traditional desktop applications and scripts. The system is then locked down to only run applications that the organization trusts.

Device Guard is a built-in feature of Windows Enterprise that prevents the execution of unwanted code and applications. Device Guard can be configured using two rule actions - allow and deny:

- **Allow** limits execution of applications to an allowed list of code or trusted publisher and blocks everything else.
- **Deny** completes the allow trusted publisher approach by blocking the execution of a specific application.

At the time of this writing, and according to Microsoft's latest research, more than 90 percent of malware is unsigned completely. So implementing a basic Device Guard policy can simply and effectively help block malware. In fact, Device Guard has the potential to go further, and can also help block signed malware.

Device Guard needs to be planned and configured to be truly effective. It isn't just a protection that is enabled or disabled. Device Guard is a combination of hardware security features and software security features that, when configured together, can lock down a computer to help ensure the most secure and resistant system possible.

There are three different parts that make up the Device Guard solution in Windows:

- The first part is a base **set of hardware security features** introduced with the previous version of Windows. TPM for hardware cryptographic operations and UEFI with modern firmware, along with Secure Boot, allows you to control what the device is running when the systems start.
- After the hardware security feature, there's the code integrity engine. In Windows, **Code Integrity is now fully configurable** and now resides in Isolated user mode, a part of the memory that is protected by Virtualization-based security.
- The last part of Device Guard is **manageability**. Code Integrity configuration is exposed through specific Group Policy Objects, PowerShell cmdlets, and MDM configuration service providers (CSPs).

For more information on how to deploy Device Guard in an enterprise, see the Device Guard deployment guide.

Device Guard scenarios

As previously described, Device Guard is a powerful way to lock down systems. Device Guard isn't intended to be used broadly and it may not always be applicable, but there are some high-interest scenarios.

Device Guard is useful and applicable on fixed workloads systems like cash registers, kiosk machines, Secure Admin Workstations (SAWs), or well managed desktops. Device Guard is highly relevant on systems that have a well-defined software that are expected to run and don't change too frequently. It could also help protect Information Workers (IWs) beyond just SAWs, as long as what they need to run is known and the set of applications isn't going to change on a daily basis.

SAWs are computers that are built to help significantly reduce the risk of compromise from malware, phishing attacks, bogus websites, and PtH attacks, among other security risks. Although SAWs can't be considered a "silver bullet" security solution to these attacks, these types of clients are helpful as part of a layered, defense-in-depth approach to security.

To protect high-value assets, SAWs are used to make secure connections to those assets.

Similarly, on corporate fully managed workstations, where applications are installed by using a distribution tool like Microsoft Configuration Manager, Intune, or any third-party device management, then Device Guard is applicable. In that type of scenario, the organization has a good idea of the software that an average user is running.

It could be challenging to use Device Guard on corporate, lightly managed workstations where the user is typically allowed to install software on their own. When an organization offers great flexibility, it's difficult to run Device Guard in enforcement mode. Nevertheless, Device Guard can be run in Audit mode, and in that case, the event log contains a record of any binaries that violated the Device Guard policy. When Device Guard is used in Audit mode, organizations can get rich data about drivers and applications that users install and run.

Before you can benefit from the protection included in Device Guard, Code Integrity policy must be created by using tools provided by Microsoft, but the policy can be deployed with common management tools, like Group Policy. The Code Integrity policy is a binary-encoded XML document that includes configuration settings for both the User and Kernel-modes of Windows, along with restrictions on Windows script hosts. Device Guard Code Integrity policy restricts what code can run on a device.

> ⓘ **Note**
>
> Device Guard policy can be signed in Windows, which adds additional protection against administrative users changing or removing this policy.

Signed Device Guard policy offers stronger protection against a malicious local administrator trying to defeat Device Guard.

When the policy is signed, the GUID of the policy is stored in a UEFI pre-OS secure variable that offers tampering protection. The only way to update the Device Guard policy later is to provide

a new version of the policy signed by the same signer or from a signer specified as part of the Device Guard policy into the UpdateSigner section.

The importance of signing applications

On computers with Device Guard, Microsoft proposes to move from a world where unsigned apps can be run without restriction to a world where only signed and trusted code is allowed to run on Windows.

With Windows, organizations make line-of-business (LOB) apps available to members of the organization through the Microsoft Store infrastructure. More specifically, LOB apps are available in a private store within the public Microsoft Store. Microsoft Store signs and distributes Universal Windows apps and Classic Windows apps. All apps downloaded from the Microsoft Store are signed.

In organizations today, many LOB applications are unsigned. Code signing is frequently viewed as a tough problem to solve for various reasons, like the lack of code signing expertise. Even if code signing is a best practice, many internal applications aren't signed.

Windows includes tools that allow IT pros to take applications that have been already packaged and run them through a process to create more signatures that can be distributed along with existing applications.

Why are anti-malware and device management solutions still necessary?

Although allowlist mechanisms are efficient at ensuring that only trusted applications can be run, they can't prevent the compromise of a trusted (but vulnerable) application by malicious content designed to exploit a known vulnerability. Device Guard doesn't protect against user mode malicious code run by exploiting vulnerabilities.

Vulnerabilities are weaknesses in software that could allow an attacker to compromise the integrity, availability, or confidentiality of the device. Some of the worst vulnerabilities allow attackers to exploit the compromised device by causing it to run malicious code without the user's knowledge.

It's common to see attackers distributing specially crafted content in an attempt to exploit known vulnerabilities in user mode software like web browsers (and their plug-ins), Java virtual machines, PDF readers, or document editors. As of today, 90 percent of discovered vulnerabilities affect user mode applications compared to the operating system and kernel mode drivers that host them.

To combat these threats, patching is the single most effective control, with anti-malware software forming complementary layers of defense.

Most application software has no facility for updating itself, so even if the software vendor publishes an update that fixes the vulnerability, the user may not know that the update is available or how to obtain it, and therefore remains vulnerable to attack. Organizations still need to manage devices and to patch vulnerabilities.

MDM solutions are becoming prevalent as a light-weight device management technology. Windows extends the management capabilities that have become available for MDMs. One key feature Microsoft has added to Windows is the ability for MDMs to acquire a strong statement of device health from managed and registered devices.

Device health attestation

Device health attestation uses the TPM to provide cryptographically strong and verifiable measurements of the chain of software used to boot the device.

For Windows-based devices, Microsoft introduces a new public API that will allow MDM software to access a remote attestation service called Windows Health Attestation Service. A health attestation result, in addition with other elements, can be used to allow or deny access to networks, apps, or services, based on whether devices prove to be healthy.

For more information on device health attestation, see the Detect an unhealthy Windows-based device section.

Windows edition and licensing requirements

The following table lists the Windows editions that support Device health attestation service:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Device health attestation service license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Hardware requirements

The following table details the hardware requirements for both Virtualization-based security services and the health attestation feature. For more information, see Minimum hardware requirements.

Expand table

Hardware	Motivation
UEFI 2.3.1 or later firmware with Secure Boot enabled	Required to support UEFI Secure Boot. UEFI Secure Boot ensures that the device boots only authorized code. Additionally, Boot Integrity (Platform Secure Boot) must be supported following the requirements in Hardware Compatibility Specification for Systems for Windows under the subsection: "System.Fundamentals.Firmware.CS.UEFISecureBoot.ConnectedStandby"
Virtualization extensions, such as Intel VT-x, AMD-V, and SLAT must be enabled	Required to support Virtualization-based security. **Note:** Device Guard can be enabled without using Virtualization-based security.
X64 processor	Required to support Virtualization-based security that uses Windows Hypervisor. Hyper-V is supported only on x64 processor (and not on x86). Direct Memory Access (DMA) protection can be enabled to provide extra memory protection but requires processors to include DMA protection technologies.
IOMMU, such as Intel VT-d, AMD-Vi	Support for the IOMMU in Windows enhances system resiliency against DMA attacks.
Trusted Platform Module (TPM)	Required to support health attestation and necessary for other key protections for Virtualization-based security. TPM 2.0 is supported. Support for TPM 1.2 was added beginning in Windows 10, version 1607 (RS1)

This section presented information about several closely related controls in Windows. The multi-layer defenses and in-depth approach help to eradicate low-level malware during boot sequence. Virtualization-based security is a fundamental operating system architecture change that adds a new security boundary. Device Guard and Credential Guard respectively help to

block untrusted code and protect corporate domain credentials from theft and reuse. This section also briefly discussed the importance of managing devices and patching vulnerabilities. All these technologies can be used to harden and lock down devices while limiting the risk of attackers compromising them.

Detect an unhealthy Windows-based device

As of today, many organizations only consider devices to be compliant with company policy after they've passed various checks that show, for example, that the operating system is in the correct state, properly configured, and has security protection enabled. Unfortunately, with today's systems, this form of reporting isn't entirely reliable because malware can spoof a software statement about system health. A rootkit, or a similar low-level exploit, can report a false healthy state to traditional compliance tools.

The biggest challenge with rootkits is that they can be undetectable to the client. Because they start before anti-malware, and they have system-level privileges, they can completely disguise themselves while continuing to access system resources. As a result, traditional computers infected with rootkits appear to be healthy, even with anti-malware running.

As previously discussed, the health attestation feature of Windows uses the TPM hardware component to securely record a measurement of every boot-related component, including firmware, Windows kernel, and even early boot drivers. Because health attestation uses the hardware-based security capabilities of TPM, the log of all boot measured components remains out of the reach of any malware.

After the devices attest a trusted boot state, they can prove that they aren't running low-level malware that could spoof later compliance checks. TPM-based health attestation provides a reliable anchor of trust for assets that contain high-value data.

What is the concept of device health?

To understand the concept of device health, it's important to know traditional measures that IT pros have taken to prevent the breach of malware. Malware control technologies are highly focused on the prevention of installation and distribution.

However, the use of traditional malware prevention technologies like anti-malware or patching solutions brings a new set of issues for IT pros: the ability to monitor and control the compliance of devices accessing organization's resources.

The definition of device compliance will vary based on an organization's installed anti-malware, device configuration settings, patch management baseline, and other security requirements. But health of the device is part of the overall device compliance policy.

The health of the device isn't binary and depends on the organization's security implementation. The Health Attestation Service provides information back to the MDM on which security features are enabled during the boot of the device by using trustworthy hardware TPM.

But health attestation only provides information, which is why an MDM solution is needed to take and enforce a decision.

Remote device health attestation

In Windows, health attestation refers to a feature where Measured Boot data generated during the boot process is sent to a remote device health attestation service operated by Microsoft.

This approach is the most secure one available for Windows-based devices to detect when security defenses are down. During the boot process, the TCG log and PCRs' values are sent to a remote Microsoft cloud service. Logs are then checked by the Health Attestation Service to determine what changes have occurred on the device.

A relying party like an MDM can inspect the report generated by the remote health attestation service.

> **Note**
>
> To use the health attestation feature of Windows, the device must be equipped with a discrete or firmware TPM. There is no restriction on any particular edition of Windows.

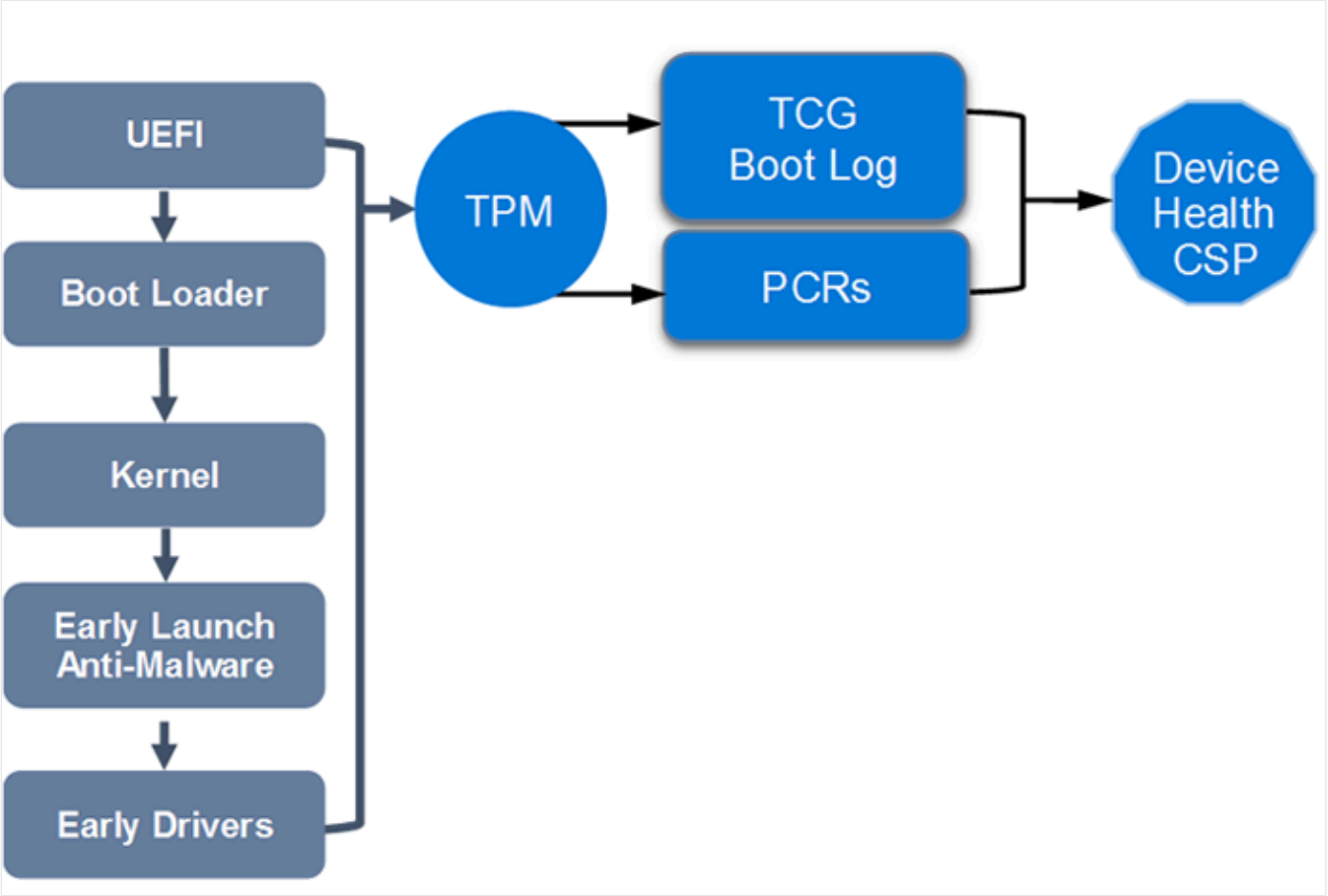
Windows supports health attestation scenarios by allowing applications access to the underlying health attestation configuration service provider (CSP) so that applications can request a health attestation token. The measurement of the boot sequence can be checked at any time locally by an anti-malware or an MDM agent.

Remote device health attestation combined with an MDM provides a hardware-rooted method for reporting the current security status and detecting any changes, without having to trust the software running on the system.

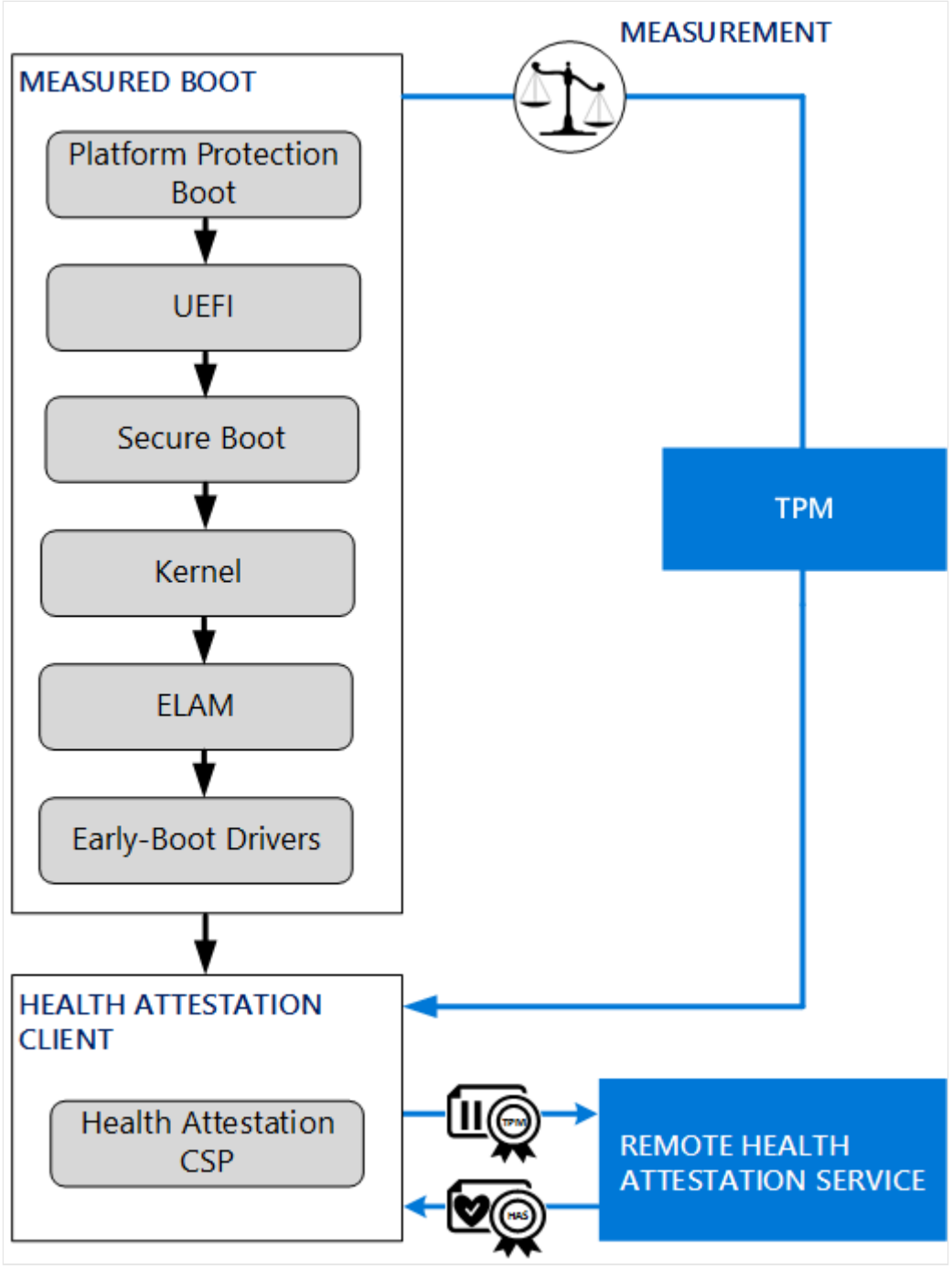
In the case where malicious code is running on the device, the use of a remote server is required. If a rootkit is present on the device, the anti-malware is no longer reliable, and its behavior can be hijacked by a malicious code running early in the startup sequence. This reason is what makes it important to use Secure Boot and Device Guard, to control which code is loaded during the boot sequence.

The anti-malware software can search to determine whether the boot sequence contains any signs of malware, such as a rootkit. It can also send the TCG log and the PCRs to a remote health attestation server to provide a separation between the measurement component and the verification component.

Health attestation logs the measurements in various TPM Platform Configuration Registers (PCRs) and TCG logs during the boot process.



When you start a device equipped with TPM, a measurement of different components is performed. These components include firmware, UEFI drivers, CPU microcode, and also all the Windows drivers whose type is Boot Start. The raw measurements are stored in the TPM PCR registers while the details of all events (executable path, authority certification, and so on) are available in the TCG log.



The health attestation process works as follows:

1. Hardware boot components are measured.
2. Operating system boot components are measured.
3. If Device Guard is enabled, current Device Guard policy is measured.
4. Windows kernel is measured.
5. Antivirus software is started as the first kernel mode driver.
6. Boot start drivers are measured.
7. MDM server through the MDM agent issues a health check command by using the Health Attestation CSP.
8. Boot measurements are validated by the Health Attestation Service

> ⓘ **Note**

By default, the last 100 system boot logs and all associated resume logs are archived in the %SystemRoot%\logs\measuredboot folder. The number of retained logs may be set with the registry **REG_DWORD** value **PlatformLogRetention** under the **HKLM\SYSTEM\CurrentControlSet\Services\TPM** key. A value of **0** will turn off log archival and a value of **0xffffffff** will keep all logs.

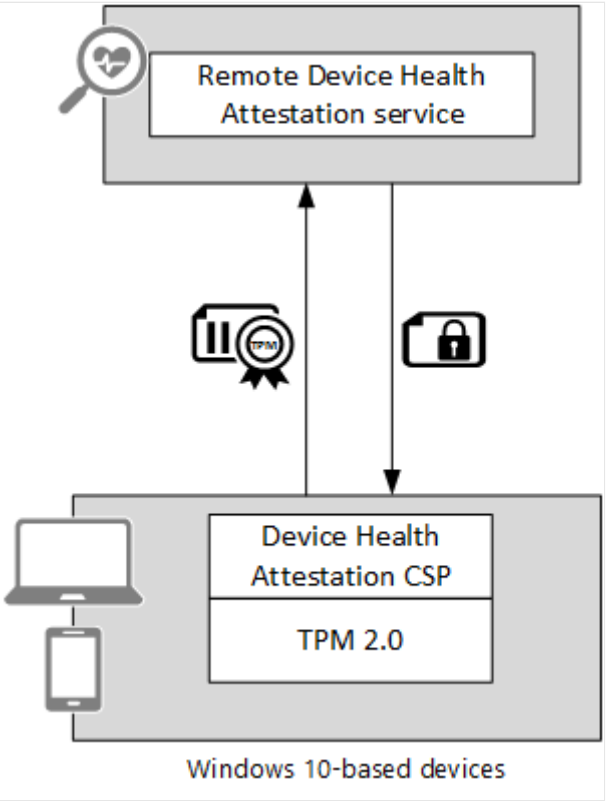
The following process describes how health boot measurements are sent to the health attestation service:

1. The client (a Windows-based device with TPM) initiates the request with the remote device health attestation service. Because the health attestation server is expected to be a Microsoft cloud service, the URI is already pre-provisioned in the client.

2. The client then sends the TCG log, the AIK signed data (PCR values, boot counter) and the AIK certificate information.

3. The remote device heath attestation service then:
   a. Verifies that the AIK certificate is issued by a known and trusted CA and the certificate is valid and not revoked.
   b. Verifies that the signature on the PCR quotes is correct and consistent with the TCG log value.
   c. Parses the properties in the TCG log.
   d. Issues the device health token that contains the health information, the AIK information, and the boot counter information. The health token also contains valid issuance time. The device health token is encrypted and signed, that means that the information is protected and only accessible to issuing health attestation service.

4. The client stores the health encrypted blob in its local store. The device health token contains device health status, a device ID (the Windows AIK), and the boot counter.



Device health attestation components

The device health attestation solution involves different components that are TPM, Health Attestation CSP, and the Windows Health Attestation Service. Those components are described in this section.

Trusted Platform Module

This section describes how PCRs (that contain system configuration data), endorsement key (EK) (that act as an identity card for TPM), SRK (that protect keys) and AIKs (that can report platform state) are used for health attestation reporting.

In a simplified manner, the TPM is a passive component with limited resources. It can calculate random numbers, RSA keys, decrypt short data, store hashes taken when booting the device.

A TPM incorporates in a single component:

- An RSA 2048-bit key generator
- A random number generator
- Nonvolatile memory for storing EK, SRK, and AIK keys
- A cryptographic engine to encrypt, decrypt, and sign
- Volatile memory for storing the PCRs and RSA keys

Endorsement key

The TPM has an embedded unique cryptographic key called the endorsement key. The TPM endorsement key is a pair of asymmetric keys (RSA size 2048 bits).

The endorsement key public key is used for sending securely sensitive parameters, such as when taking possession of the TPM that contains the defining hash of the owner password. The EK private key is used when creating secondary keys like AIKs.

The endorsement key acts as an identity card for the TPM. For more information, see Understand the TPM endorsement key.

The endorsement key is often accompanied by one or two digital certificates:

- One certificate is produced by the TPM manufacturer and is called the **endorsement certificate**. The endorsement certificate is used to prove the authenticity of the TPM (for example, that it's a real TPM manufactured by a specific chip maker) to local processes, applications, or cloud services. The endorsement certificate is created during manufacturing or the first time the TPM is initialized by communicating with an online service.
- The other certificate is produced by the platform builder and is called the **platform certificate** to indicate that a specific TPM is integrated with a certain device. For certain devices that use firmware-based TPM produced by Intel or Qualcomm, the endorsement certificate is created when the TPM is initialized during the OOBE of Windows.

> **Note**
>
> Secure Boot protects the platform until the Windows kernel is loaded. Then protections like Trusted Boot, Hyper-V Code Integrity and ELAM take over. A device that uses Intel TPM or Qualcomm TPM gets a signed certificate online from the manufacturer that has created the chip and then stores the signed certificate in TPM storage. For the operation to succeed, if you are filtering Internet access from your client devices, you must authorize the following URLs:

- For Intel firmware TPM: `https://ekop.intel.com/ekcertservice`
- For Qualcomm firmware TPM: `https://ekcert.spserv.microsoft.com/`

Attestation Identity Keys

Because the endorsement certificate is unique for each device and doesn't change, the usage of it may present privacy concerns because it's theoretically possible to track a specific device. To avoid this privacy problem, Windows issues a derived attestation anchor based on the endorsement certificate. This intermediate key, which can be attested to an endorsement key,

is the Attestation Identity Key (AIK) and the corresponding certificate is called the AIK certificate. This AIK certificate is issued by a Microsoft cloud service.

> **Note**
>
> Before the device can report its health using the TPM attestation functions, an AIK certificate must be provisioned in conjunction with a third-party service like the Microsoft Cloud CA service. After it is provisioned, the AIK private key can be used to report platform configuration. Windows creates a signature over the platform log state (and a monotonic counter value) at each boot by using the AIK.

The AIK is an asymmetric (public/private) key pair that is used as a substitute for the EK as an identity for the TPM for privacy purposes. The private portion of an AIK is never revealed or used outside the TPM and can only be used inside the TPM for a limited set of operations. Furthermore, it can only be used for signing, and only for limited, TPM-defined operations.

Windows creates AIKs protected by the TPM, if available, that are 2048-bit RSA signing keys. Microsoft is hosting a cloud service called Microsoft Cloud CA to establish cryptographically that it's communicating with a real TPM and that the TPM possesses the presented AIK. After the Microsoft Cloud CA service has established these facts, it will issue an AIK certificate to the Windows-based device.

Many existing devices that will upgrade to Windows 10 won't have a TPM, or the TPM won't contain an endorsement certificate. **To accommodate those devices, Windows 10 allows the issuance of AIK certificates without the presence of an endorsement certificate**. Such AIK certificates aren't issued by Microsoft Cloud CA. These certificates aren't as trustworthy as an endorsement certificate that is burned into the device during manufacturing, but it will provide compatibility for advanced scenarios like Windows Hello for Business without TPM.

In the issued AIK certificate, a special OID is added to attest that endorsement certificate was used during the attestation process. This information can be used by a relying party to decide whether to reject devices that are attested using AIK certificates without an endorsement certificate or accept them. Another scenario can be to not allow access to high-value assets from devices that are attested by an AIK certificate that isn't backed by an endorsement certificate.

Storage root key

The storage root key (SRK) is also an asymmetric key pair (RSA with a minimum of 2048-bits length). The SRK has a major role and is used to protect TPM keys, so that these keys can't be used without the TPM. The SRK key is created when the ownership of the TPM is taken.

Platform Configuration Registers

The TPM contains a set of registers that are designed to provide a cryptographic representation of the software and state of the system that booted. These registers are called Platform Configuration Registers (PCRs).

The measurement of the boot sequence is based on the PCR and TCG log. To establish a static root of trust, when the device is starting, the device must be able to measure the firmware code before execution. In this case, the Core Root of Trust for Measurement (CRTM) is executed from the boot, calculates the hash of the firmware, then stores it by expanding the register PCR[0] and transfers execution to the firmware.

PCRs are set to zero when the platform is booted, and it's the job of the firmware that boots the platform to measure components in the boot chain and to record the measurements in the PCRs. Typically, boot components take the hash of the next component that is to be run and record the measurements in the PCRs. The initial component that starts the measurement chain is implicitly trusted. This component is the CRTM. Platform manufacturers are required to have a secure update process for the CRTM or not permit updates to it. The PCRs record a cumulative hash of the components that have been measured.

The value of a PCR on its own is hard to interpret (it's just a hash value), but platforms typically keep a log with details of what has been measured, and the PCRs merely ensure that the log hasn't been tampered with. The logs are referred as a TCG log. Each time a register PCR is extended, an entry is added to the TCG log. Thus, throughout the boot process, a trace of the executable code and configuration data is created in the TCG log.

TPM provisioning

For the TPM of a Windows-based device to be usable, it must first be provisioned. The process of provisioning differs based on TPM versions, but, when successful, it results in the TPM being usable and the owner authorization data (ownerAuth) for the TPM being stored locally on the registry.

When the TPM is provisioned, Windows will first attempt to determine the EK and locally stored **ownerAuth** values by looking in the registry at the following location: **HKLM\SYSTEM\CurrentControlSet\Services\TPM\WMI\Endorsement**

During the provisioning process, the device may need to be restarted.

The **Get-TpmEndorsementKeyInfo PowerShell** cmdlet can be used with administrative privilege to get information about the endorsement key and certificates of the TPM.

If the TPM ownership isn't known but the EK exists, the client library will provision the TPM and will store the resulting **ownerAuth** value into the registry if the policy allows it will store the SRK public portion at the following location: **HKLM\SYSTEM\CurrentControlSet\Services\TPM\WMI\Admin\SRKPub**

As part of the provisioning process, Windows will create an AIK with the TPM. When this operation is performed, the resulting AIK public portion is stored in the registry at the following location: **HKLM\SYSTEM\CurrentControlSet\Services\TPM\WMI\WindowsAIKPub**

> ⓘ **Note**
>
> For provisioning AIK certificates and filtering Internet access, you must authorize the following wildcard URL: `https://\*.microsoftaik.azure.net`

Windows Health Attestation CSP

Windows contains a configuration service provider (CSP) specialized for interacting with the health attestation feature. A CSP is a component that plugs into the Windows MDM client and provides a published protocol for how MDM servers can configure settings and manage Windows-based devices. The management protocol is represented as a tree structure that can be specified as URIs with functions to perform on the URIs such as "get", "set", "delete", and so on.

The following list is that of the functions performed by the Windows Health Attestation CSP:

- Collects data that is used to verify a device's health status
- Forwards the data to the Health Attestation Service
- Provisions the Health Attestation Certificate that it receives from the Health Attestation Service
- Upon request, forwards the Health Attestation Certificate (received from the Health Attestation Service) and related runtime information to the MDM server for verification

During a health attestation session, the Health Attestation CSP forwards the TCG logs and PCRs' values that are measured during the boot, by using a secure communication channel to the Health Attestation Service.

When an MDM server validates that a device has attested to the Health Attestation Service, it will be given a set of statements and claims about how that device booted, with the assurance that the device didn't reboot between the time that it attested its health and the time that the MDM server validated it.

Windows Health Attestation Service

The role of Windows Health Attestation Service is essentially to evaluate a set of health data (TCG log and PCR values), make a series of detections (based on available health data) and generate encrypted health blob or produce report to MDM servers.

> ⓘ **Note**
>
> Both device and MDM servers must have access to **has.spserv.microsoft.com** using the TCP protocol on port 443 (HTTPS).

Checking that a TPM attestation and the associated log are valid takes several steps:

1. First, the server must check that the reports are signed by **trustworthy AIKs**. This verification might be done by checking that the public part of the AIK is listed in a database of assets, or perhaps that a certificate has been checked.
2. After the key has been checked, the signed attestation (a quote structure) should be checked to see whether it's a **valid signature over PCR values**.
3. Next the logs should be checked to ensure that they match the PCR values reported.
4. Finally, the logs themselves should be examined by an MDM solution to see whether they represent **known or valid security configurations**. For example, a simple check might be to see whether the measured early OS components are known to be good, that the ELAM driver is as expected, and that the ELAM driver policy file is up to date. If all of these checks succeed, an attestation statement can be issued that later can be used to determine whether or not the client should be granted access to a resource.

The Health Attestation Service provides the following information to an MDM solution about the health of the device:

- Secure Boot enablement
- Boot and kernel debug enablement
- BitLocker enablement
- VSM enabled
- Signed or unsigned Device Guard Code Integrity policy measurement
- ELAM loaded
- Safe Mode boot, DEP enablement, test signing enablement
- Device TPM has been provisioned with a trusted endorsement certificate

For completeness of the measurements, see Health Attestation CSP.

The following list shows some key items that can be reported back to MDM for Windows-based devices:

- PCR0 measurement
- Secure Boot Enabled
- Secure Boot db matches Expected
- Secure Boot dbx is up to date
- Secure Boot policy GUID matches Expected
- BitLocker enabled
- Virtualization-based security enabled
- ELAM was loaded
- Code Integrity version is up to date
- Code Integrity policy hash matches expected

Use MDM and the Health Attestation Service

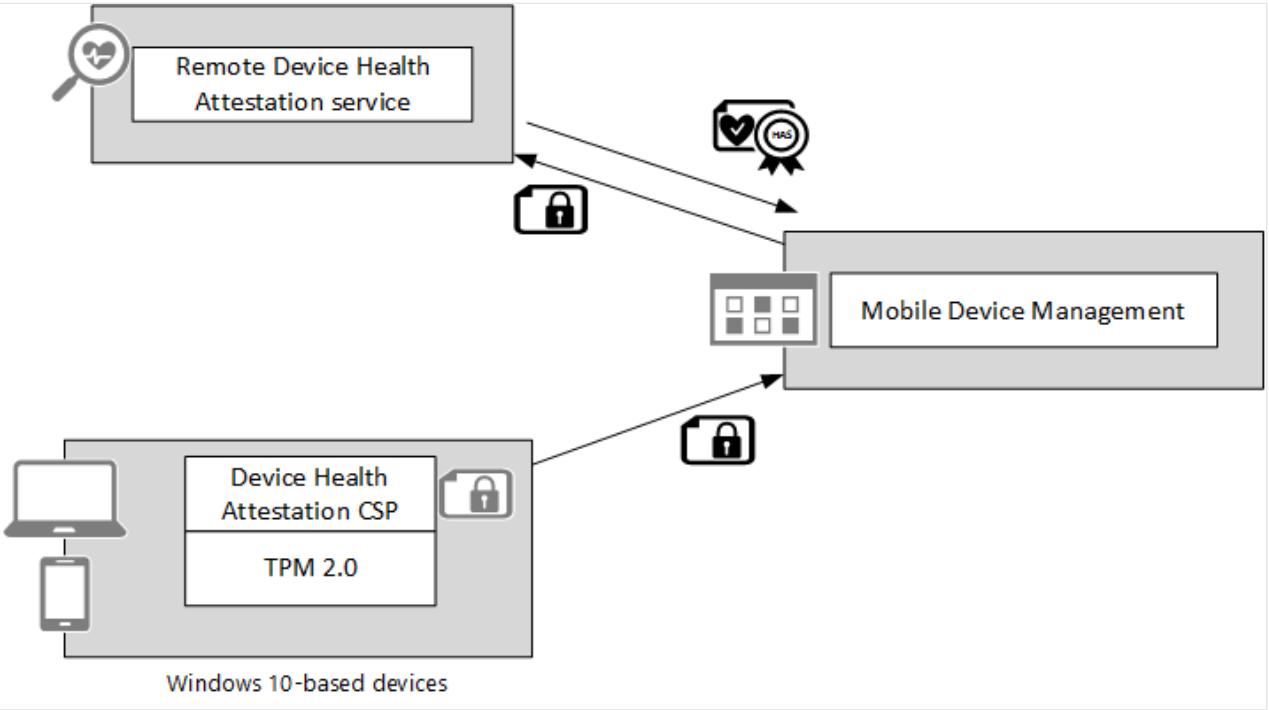
To make device health relevant, the MDM solution evaluates the device health report and is configured to the organization's device health requirements.

A solution that uses MDM and the Health Attestation Service consists of three main parts:

1. A device with health attestation enabled. This enablement will be done as a part of enrollment with an MDM provider (health attestation will be disabled by default).

2. After this service is enabled, and every boot thereafter, the device will send health measurements to the Health Attestation Service hosted by Microsoft, and it will receive a health attestation blob in return.

3. At any point after this cycle, an MDM server can request the health attestation blob from the device and ask Health Attestation Service to decrypt the content and validate that it's been attested.



Interaction between a Windows-based device, the Health Attestation Service, and MDM can be performed as follows:

1. The client initiates a session with the MDM server. The URI for the MDM server would be part of the client app that initiates the request. The MDM server at this time could request the health attestation data by using the appropriate CSP URI.

2. The MDM server specifies a nonce along with the request.

3. The client then sends the AIK quoted nonce + the boot counter and the health blob information. This health blob is encrypted with a Health Attestation Service public key that only the Health Attestation Service can decrypt.

4. The MDM server:
   a. Verifies that the nonce is as expected.
   b. Passes the quoted data, the nonce and the encrypted health blob to the Health Attestation Service server.

5. The Health Attestation Service:
   a. Decrypts the health blob.
   b. Verifies that the boot counter in the quote is correct using the AIK in the health blob and matches the value in the health blob.
   c. Verifies that the nonce matches in the quote and the one that is passed from MDM.
   d. Because the boot counter and the nonce are quoted with the AIK from the health blob, it also proves that the device is the same one as the one for which the health blob has been generated.
   e. Sends data back to the MDM server including health parameters, freshness, and so on.

> **Note**
>
> The MDM server (relying party) never performs the quote or boot counter validation itself. It gets the quoted data and the health blob (which is encrypted) and sends the data to the Health Attestation Service for validation. This way, the AIK is never visible to the MDM, which thereby addresses privacy concerns.

Setting the requirements for device compliance is the first step to ensure that registered devices that don't meet health and compliance requirements are detected, tracked, and have actions enforced by the MDM solution.

Devices that attempt to connect to resources must have their health evaluated so that unhealthy and noncompliant devices can be detected and reported. To be fully efficient, an end-to-end security solution must impose a consequence for unhealthy devices like refusing access to high-value assets. That consequence for an unhealthy device is the purpose of conditional access control, which is detailed in the next section.

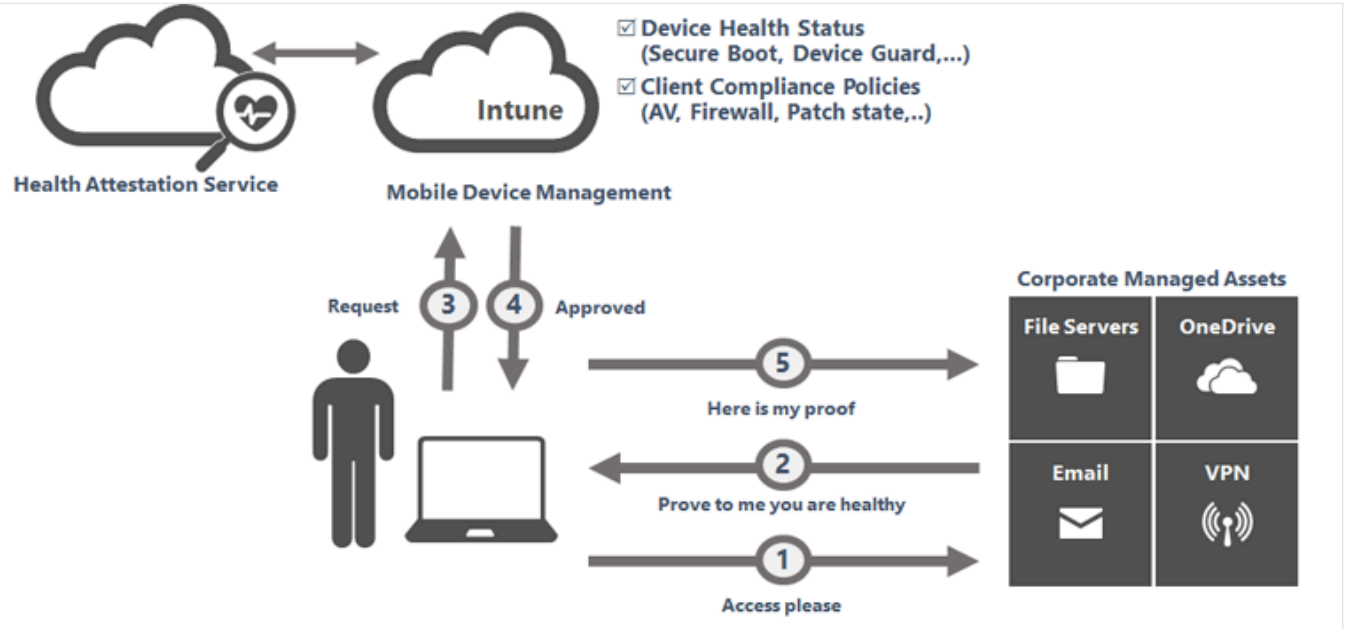
Control the security of a Windows-based device before access is granted

Today's access control technology, in most cases, focuses on ensuring that the right people get access to the right resources. If users can authenticate, they get access to resources using a device that the organization's IT staff and systems know little about. Perhaps there's some check such as ensuring that a device is encrypted before giving access to email, but what if the device is infected with malware?

The remote device health attestation process uses measured boot data to verify the health status of the device. The health of the device is then available for an MDM solution like Intune.

> **Note**
>
> For the latest information on Intune and Windows features support, see **What's new in Microsoft Intune**.

The figure below shows how the Health Attestation Service is expected to work with Microsoft's cloud-based Intune MDM service.



An MDM solution can then use health state statements and take them to the next level by coupling with client policies that will enable conditional access to be granted based on the device's ability to prove that it's malware free, its anti-malware system is functional and up to date, the firewall is running, and the devices patch state is compliant.

Finally, resources can be protected by denying access to endpoints that are unable to prove they're healthy. This feature is much needed for BYOD devices that need to access organizational resources.

Built-in support of MDM in Windows

Windows has an MDM client that ships as part of the operating system. This MDM client enables MDM servers to manage Windows-based devices without requiring a separate agent.

Non-Microsoft MDM server support

Non-Microsoft MDM servers can manage Windows by using the MDM protocol. The built-in management client is able to communicate with a compatible server that supports the OMA-DM protocol to perform enterprise management tasks. For more information, see Microsoft Entra integration with MDM.

> **Note**
>
> MDM servers do not need to create or download a client to manage Windows. For more information, see Mobile device management.

The third-party MDM server will have the same consistent first-party user experience for enrollment, which also provides simplicity for Windows users.

Management of Windows Defender by third-party MDM

This management infrastructure makes it possible for IT pros to use MDM-capable products like Intune, to manage health attestation, Device Guard, or Windows Defender on Windows-based devices, including BYODs that aren't domain joined. IT pros will be able to manage and configure all of the actions and settings they're familiar with customizing by using Intune with Intune Endpoint Protection on down-level operating systems. Admins that currently only manage domain joined devices through Group Policy will find it easy to transition to managing Windows-based devices by using MDM because many of the settings and actions are shared across both mechanisms.

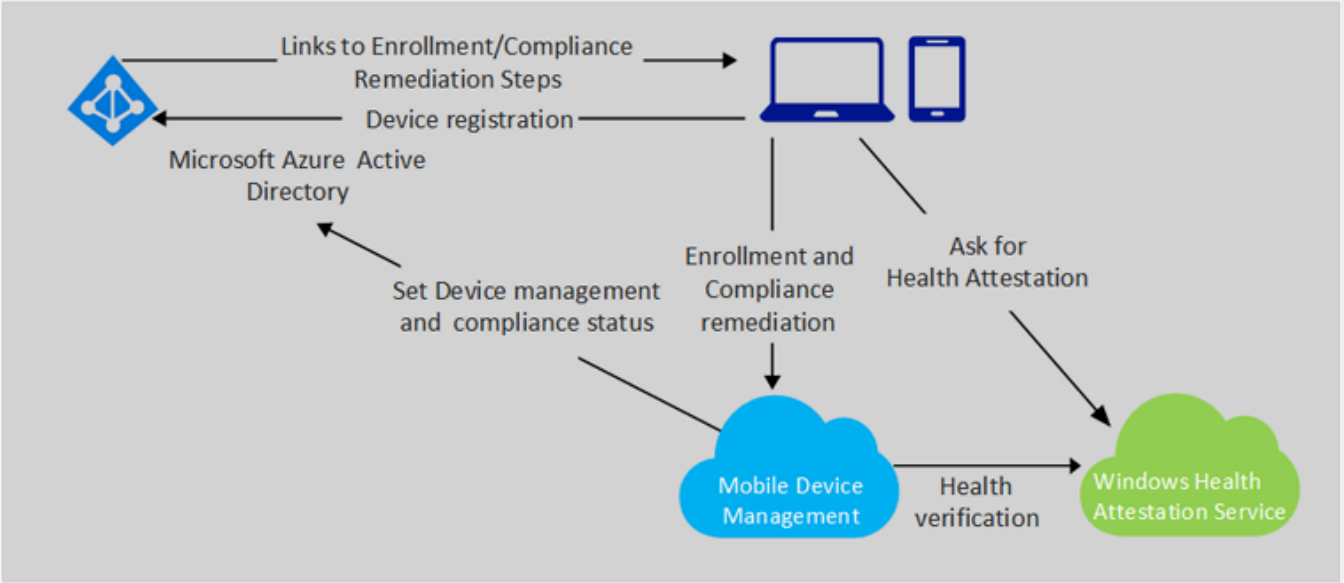
For more information on how to manage Windows security and system settings with an MDM solution, see Custom URI settings for Windows devices.

Conditional access control

On most platforms, the Microsoft Entra device registration happens automatically during enrollment. The device states are written by the MDM solution into Microsoft Entra ID, and then read by Office 365 (or by any authorized Windows app that interacts with Microsoft Entra ID) the next time the client tries to access an Office 365 compatible workload.

If the device isn't registered, the user will get a message with instructions on how to register (also known as enrolling). If the device isn't compliant, the user will get a different message that redirects them to the MDM web portal where they can get more information on the compliance problem and how to resolve it.

Microsoft Entra ID authenticates the user and the device, **MDM** manages the compliance and conditional access policies, and the **Health Attestation Service** reports about the health of the device in an attested way.



Who does what?
Mobile Device Management: Evaluate policy compliance for device
Health Attestation Service: performs a series of health checks and reports to the MDM solution
Azure AD: Authenticate user and device and issue security token

Office 365 conditional access control

Microsoft Entra ID enforces conditional access policies to secure access to Office 365 services. A tenant admin can create a conditional access policy that blocks a user on a non-compliant device from accessing an Office 365 service. The user must conform to the company's device policies before access can be granted to the service. Alternately, the admin can also create a policy that requires users to just enroll their devices to gain access to an Office 365 service. Policies may be applied to all users of an organization, or limited to a few target groups and enhanced over time to include more target groups.

When a user requests access to an Office 365 service from a supported device platform, Microsoft Entra authenticates the user and device from which the user launches the request; and grants access to the service only when the user conforms to the policy set for the service. Users that don't have their device enrolled are given remediation instructions on how to enroll and become compliant to access corporate Office 365 services.

When a user enrolls, the device is registered with Microsoft Entra ID, and enrolled with a compatible MDM solution like Intune.

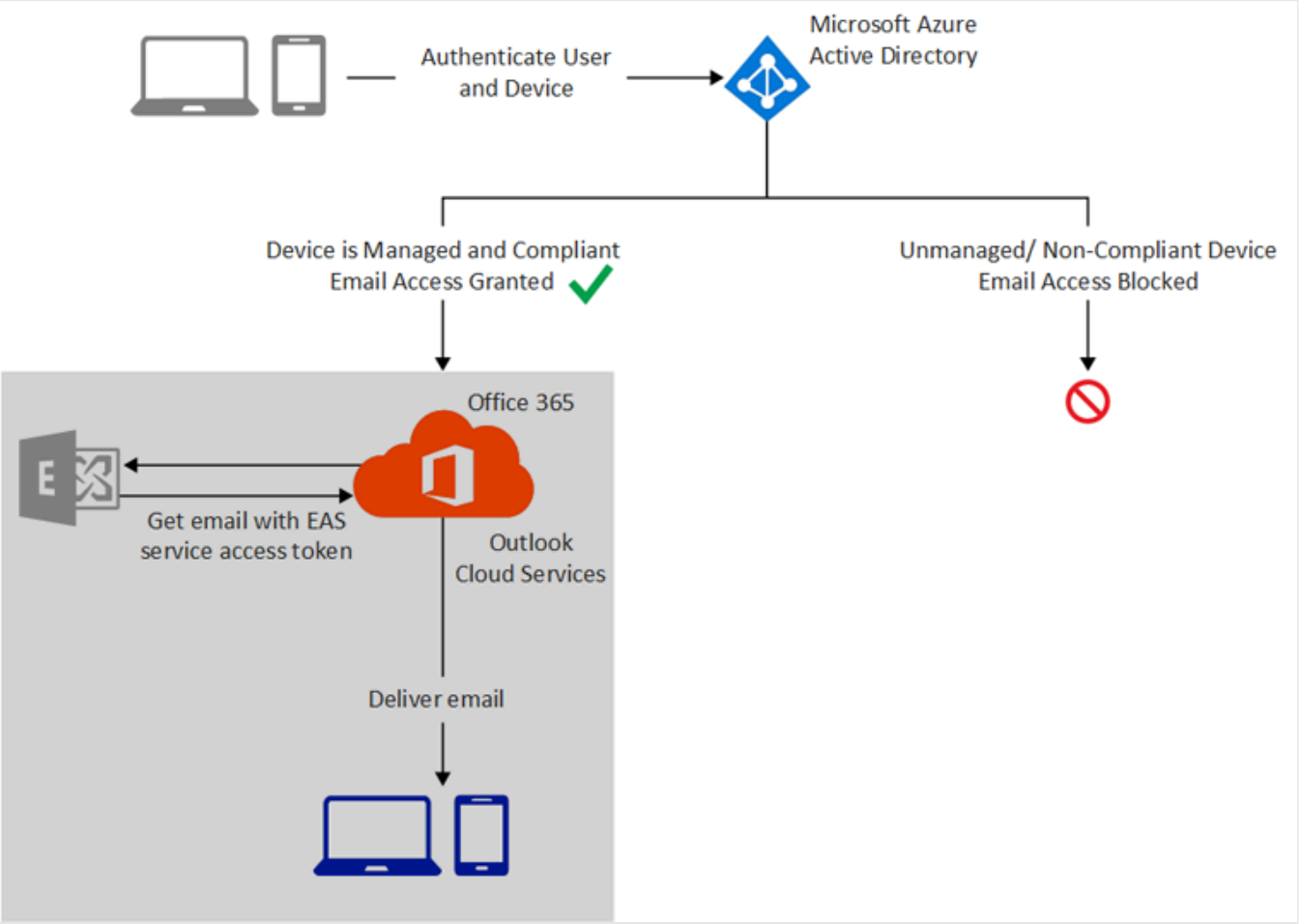
> ⓘ **Note**
>
> Microsoft is working with third-party MDM ISVs to support automated MDM enrollment and policy based access checks. Steps to turn on auto-MDM enrollment with Microsoft

Entra ID and Intune are explained in the [Windows, Microsoft Entra ID And Microsoft Intune: Automatic MDM Enrollment Powered By The Cloud!] blog post.

When a user enrolls a device successfully, the device becomes trusted. Microsoft Entra ID provides single-sign-on to access company applications and enforces conditional access policy to grant access to a service not only the first time the user requests access, but every time the user requests to renew access.

The user will be denied access to services when sign-in credentials are changed, a device is lost/stolen, or the compliance policy isn't met at the time of request for renewal.

Depending on the type of email application that employees use to access Exchange online, the path to establish secured access to email can be slightly different. However, the key components: Microsoft Entra ID, Office 365/Exchange Online, and Intune, are the same. The IT experience and end-user experience also are similar.



Clients that attempt to access Office 365 will be evaluated for the following properties:

- Is the device managed by an MDM?
- Is the device registered with Microsoft Entra ID?
- Is the device compliant?

To get to a compliant state, the Windows-based device needs to:

- Enroll with an MDM solution.
- Register with Microsoft Entra ID.
- Be compliant with the device policies set by the MDM solution.

> **Note**
>
> At the present time, conditional access policies are selectively enforced on users on iOS and Android devices. For more information, see the Microsoft Entra ID, Microsoft Intune and Windows - Using the cloud to modernize enterprise mobility! blog post.

Cloud and on-premises apps conditional access control

Conditional access control is a powerful policy evaluation engine built into Microsoft Entra ID. It gives IT pros an easy way to create access rules beyond Office 365 that evaluate the context of a user's sign in to make real-time decisions about which applications they should be allowed to access.

IT pros can configure conditional access control policies for cloud SaaS applications secured by Microsoft Entra ID and even on-premises applications. Access rules in Microsoft Entra ID use the conditional access engine to check device health and compliance state reported by a compatible MDM solution like Intune in order to determine whether to allow access.

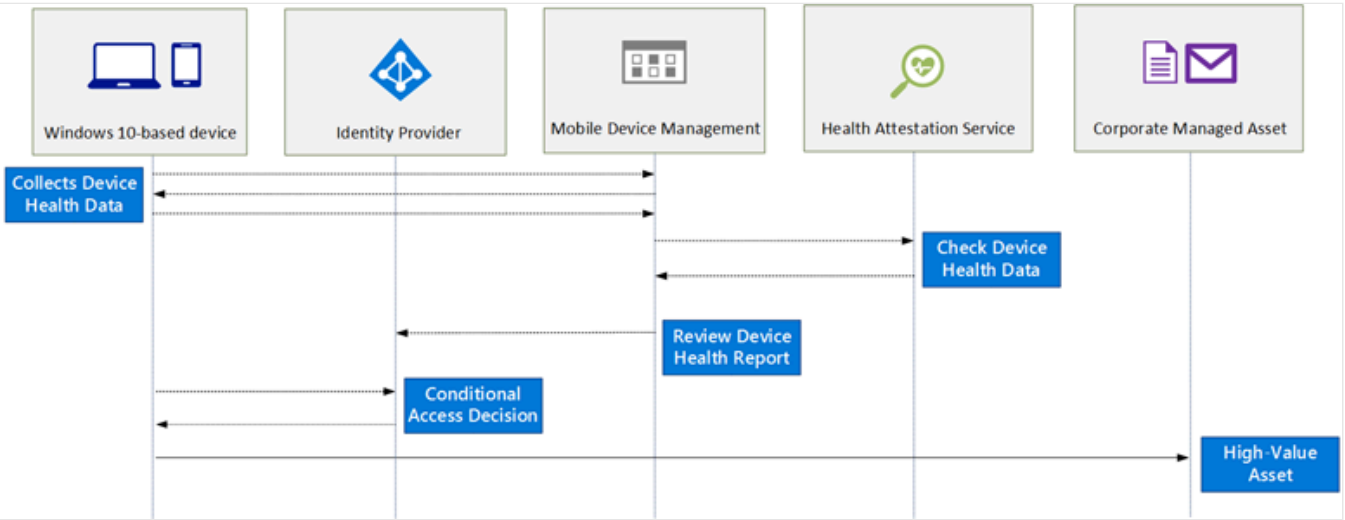
For more information about conditional access, see Azure Conditional Access Preview for SaaS Apps.

> **Note**
>
> Conditional access control is a Microsoft Entra ID P1 or P2 feature that's also available with EMS. If you don't have a Microsoft Entra ID P1 or P2 subscription, you can get a trial from the Microsoft Azure site.

For on-premises applications there are two options to enable conditional access control based on a device's compliance state:

- For on-premises applications that are published through the Microsoft Entra application proxy, you can configure conditional access control policies as you would for cloud applications. For more information, see Using Microsoft Entra application proxy to publish on-premises apps for remote users.
- Additionally, Microsoft Entra Connect will sync device compliance information from Microsoft Entra ID to on-premises AD. ADFS on Windows Server 2016 will support conditional access control based on a device's compliance state. IT pros will configure

conditional access control policies in ADFS that use the device's compliance state reported by a compatible MDM solution to secure on-premises applications.



The following process describes how Microsoft Entra Conditional Access works:

1. User has already enrolled with MDM through Workplace Access/Azure AD join, which registers device with Microsoft Entra ID.
2. When the device boots or resumes from hibernate, a task "Tpm-HASCertRetr" is triggered to request in background a health attestation blob. Device sends TPM boot measurements to the Health Attestation Service.
3. Health Attestation Service validates device state and issues an encrypted blob to the device based on the health state with details on failed checks (if any).
4. User logs on and the MDM agent contacts the Intune/MDM server.
5. MDM server pushes down new policies if available and queries health blob state and other inventory state.
6. Device sends a health attestation blob previously acquired and also the value of the other state inventory requested by the Intune/MDM server.
7. Intune/MDM server sends the health attestation blob to Health Attestation Service to be validated.
8. Health Attestation Service validates that the device that sent the health attestation blob is healthy, and returns this result to Intune/MDM server.
9. Intune/MDM server evaluates compliance based on the compliance and the queried inventory/health attestation state from device.
10. Intune/MDM server updates compliance state against device object in Microsoft Entra ID.
11. User opens app, attempts to access a corporate managed asset.
12. Access gated by compliance claim in Microsoft Entra ID.
13. If the device is compliant and the user is authorized, an access token is generated.
14. User can access the corporate managed asset.

For more information about Microsoft Entra join, see Microsoft Entra ID & Windows: Better Together for Work or School, a white paper.

Conditional access control is a topic that many organizations and IT pros may not know and they should. The different attributes that describe a user, a device, compliance, and context of access are powerful when used with a conditional access engine. Conditional access control is an essential step that helps organizations secure their environment.

Takeaways and summary

The following list contains high-level key takeaways to improve the security posture of any organization. However, the few takeaways presented in this section shouldn't be interpreted as an exhaustive list of security best practices.

- **Understand that no solution is 100 percent secure**

  If determined adversaries with malicious intent gain physical access to the device, they could eventually break through its security layers and control it.

- **Use health attestation with an MDM solution**

  Devices that attempt to connect to high-value assets must have their health evaluated so that unhealthy and noncompliant devices can be detected, reported, and eventually blocked.

- **Use Credential Guard**

  Credential Guard is a feature that greatly helps protect corporate domain credentials from pass-the-hash attacks.

- **Use Device Guard**

  Device Guard is a real advance in security and an effective way to help protect against malware. The Device Guard feature in Windows blocks untrusted apps (apps not authorized by your organization).

- **Sign Device Guard policy**

  Signed Device Guard policy helps protect against a user with administrator privileges trying to defeat the current policy. When a policy is signed, the only way to modify Device Guard later is to provide a new version of the policy signed by the same signer or from a signer specify as part of the Device Guard policy.

- **Use virtualization-based security**

  When you have Kernel Mode Code Integrity protected by Virtualization-based security, the code integrity rules are still enforced even if a vulnerability allows unauthorized kernel

mode memory access. Keep in mind that Device Guard devices that run Kernel Code Integrity with Virtualization-based security must have compatible drivers.

- **Start to deploy Device Guard with Audit mode**

  Deploy Device Guard policy to targeted computers and devices in Audit mode. Monitor the Code Integrity event log that indicates a program or a driver would have been blocked if Device Guard was configured in Enforcement mode. Adjust Device Guard rules until a high level of confidence has been reached. After the testing phase has been completed, Device Guard policy can be switched to Enforcement mode.

- **Build an isolated reference machine when deploying Device Guard**

  Because the corporate network can contain malware, you should start to configure a reference environment that is isolated from your main corporate network. After that, you can create a code integrity policy that includes the trusted applications you want to run on your protected devices.

- **Use AppLocker when it makes sense**

  Although AppLocker isn't considered a new Device Guard feature, it complements Device Guard functionality for some scenarios like being able to deny a specific Universal Windows application for a specific user or a group of users.

- **Lock down firmware and configuration**

  After Windows is installed, lock down firmware boot options access. This lockdown prevents a user with physical access from modifying UEFI settings, disabling Secure Boot, or booting other operating systems. Also, in order to protect against an administrator trying to disable Device Guard, add a rule in the current Device Guard policy that will deny and block execution of the **C:\Windows\System32\SecConfig.efi** tool.

Health attestation is a key feature of Windows that includes client and cloud components to control access to high-value assets based on a user and their device's identity and compliance with corporate governance policy. Organizations can choose to detect and report unhealthy devices, or to configure health enforcement rules based on their needs. Health attestation provides an end-to-end security model and integration points, which vendors and software developers can use to build and integrate a customized solution.

Related articles

- Protect derived domain credentials with Credential Guard
- Device Guard deployment guide
- Trusted Platform Module technology overview

Last updated on 08/18/2025

Cryptography and Certificate Management

Applies to: ✅ Windows 11, ✅ Windows 10

Cryptography

Cryptography uses code to convert data so that only a specific recipient can read it by using a key. Cryptography enforces privacy to prevent anyone except the intended recipient from reading data, integrity to ensure data is free of tampering, and authentication that verifies identity to ensure that communication is secure. The cryptography stack in Windows extends from the chip to the cloud enabling Windows, applications, and services protect system and user secrets.

Cryptography in Windows is Federal Information Processing Standards (FIPS) 140 certified. FIPS 140 certification ensures that US government approved algorithms are being used (RSA for signing, ECDH with NIST curves for key agreement, AES for symmetric encryption, and SHA2 for hashing), tests module integrity to prove that no tampering occurred and proves the randomness for entropy sources.

Windows cryptographic modules provide low-level primitives such as:

- Random number generators (RNG)
- Symmetric and asymmetric encryption (support for AES 128/256 and RSA 512 to 16384, in 64-bit increments and ECDSA over NIST-standard prime curves P-256, P-384, P-521)
- Hashing (support for SHA-256, SHA-384, SHA-512, and SHA-3*)
- Signing and verification (padding support for OAEP, PSS, PKCS1)
- Key agreement and key derivation (support for ECDH over NIST-standard prime curves P-256, P-384, P-521, and HKDF)

These modules are natively exposed on Windows through the Crypto API (CAPI) and the Cryptography Next Generation API (CNG) which is powered by Microsoft's open-source cryptographic library SymCrypt. Application developers can use these APIs to perform low-level cryptographic operations (BCrypt), key storage operations (NCrypt), protect static data (DPAPI), and securely share secrets (DPAPI-NG).

*With this release we added support for the SHA-3 family of hash functions and SHA-3 derived functions (SHAKE, cSHAKE, and KMAC). These are the latest standardized hash functions by the National Institute of Standards and Technology (NIST) and can be leveraged through the Windows CNG library. Below is a list of the supported SHA-3 functions:

Supported SHA-3 hash functions: SHA3-256, SHA3-384, SHA3-512 (SHA3-224 is not supported) Supported SHA-3 HMAC algorithms: HMAC-SHA3-256, HMAC-SHA3-384, HMAC-

SHA3-512 Supported SHA-3 derived algorithms: extendable-output functions (XOF) (SHAKE128, SHAKE256), customizable XOFs (cSHAKE128, cSHAKE256), and KMAC (KMAC128, KMAC256, KMACXOF128, KMACXOF256).

Certificate management

Windows offers several APIs to operate and manage certificates. Certificates are crucial to public key infrastructure (PKI) as they provide the means for safeguarding and authenticating information. Certificates are electronic documents used to claim ownership of a public key. Public keys are used to prove server and client identity, validate code integrity, and used in secure emails. Windows offers users the ability to autoenroll and renew certificates in Active Directory with Group Policy to reduce the risk of potential outages due to certificate expiration or misconfiguration. Windows validates certificates through an automatic update mechanism that downloads certificate trust lists (CTL) daily. Trusted root certificates are used by applications as a reference for trustworthy PKI hierarchies and digital certificates. The list of trusted and untrusted certificates are stored in the CTL and can be updated by administrators. In the case of certificate revocation, a certificate is added as an untrusted certificate in the CTL causing it to be revoked globally across user devices immediately.

Windows also offers enterprise certificate pinning to help reduce man-in-the-middle attacks by enabling users to protect their internal domain names from chaining to unwanted certificates. A web application's server authentication certificate chain is checked to ensure it matches a restricted set of certificates. Any web application triggering a name mismatch starts event logging and prevents user access from Microsoft Edge.

Last updated on 08/18/2025

Assigned Access overview

03/10/2025Applies to: ✅ Windows 11, ✅ Windows 10

Assigned Access is a Windows feature that you can use to configure a device as a kiosk or with a restricted user experience.

When you configure a **kiosk experience**, a single Universal Windows Platform (UWP) application or Microsoft Edge is executed in full screen, above the lock screen. Users can only use that application. If the kiosk app is closed, it automatically restarts. Practical examples include:

- Public browsing
- Interactive digital signage

When you configure a **restricted user experience**, users can only execute a defined list of applications, with a tailored Start menu and Taskbar. Different policy settings and AppLocker rules are enforced, creating a locked down experience. The users can access a familiar Windows desktop, while limiting their access, reducing distractions, and potential for inadvertent uses. Ideal for shared devices, you can create different configurations for different users. Practical examples include:

- Frontline worker devices
- Student devices
- Lab devices

> **Note**
>
> When you configure a restricted user experience, different policy settings are applied to the device. Some policy settings apply to standard users only, and some to administrator accounts too. For more information, see Assigned Access policy settings.

System requirements

Here are the requirements for Assigned Access:

- To use a kiosk experience, User account control (UAC) must be enabled
- To use a kiosk experience, you must sign in from the console. The kiosk experience isn't supported over a remote desktop connection

Windows edition requirements

The following list contains the Windows editions that support Assigned Access:

✅ Pro
✅ Enterprise / Enterprise LTSC
✅ Education
✅ IoT Enterprise / IoT Enterprise LTSC

Next steps

Learn how to configure Assigned Access:

- Configure a single-app kiosk experience with Assigned Access
- Configure a restricted user experience (multi-app kiosk) with Assigned Access

🚀 Quickstarts

If you want to quickly test Assigned Access, check out the following quickstarts:

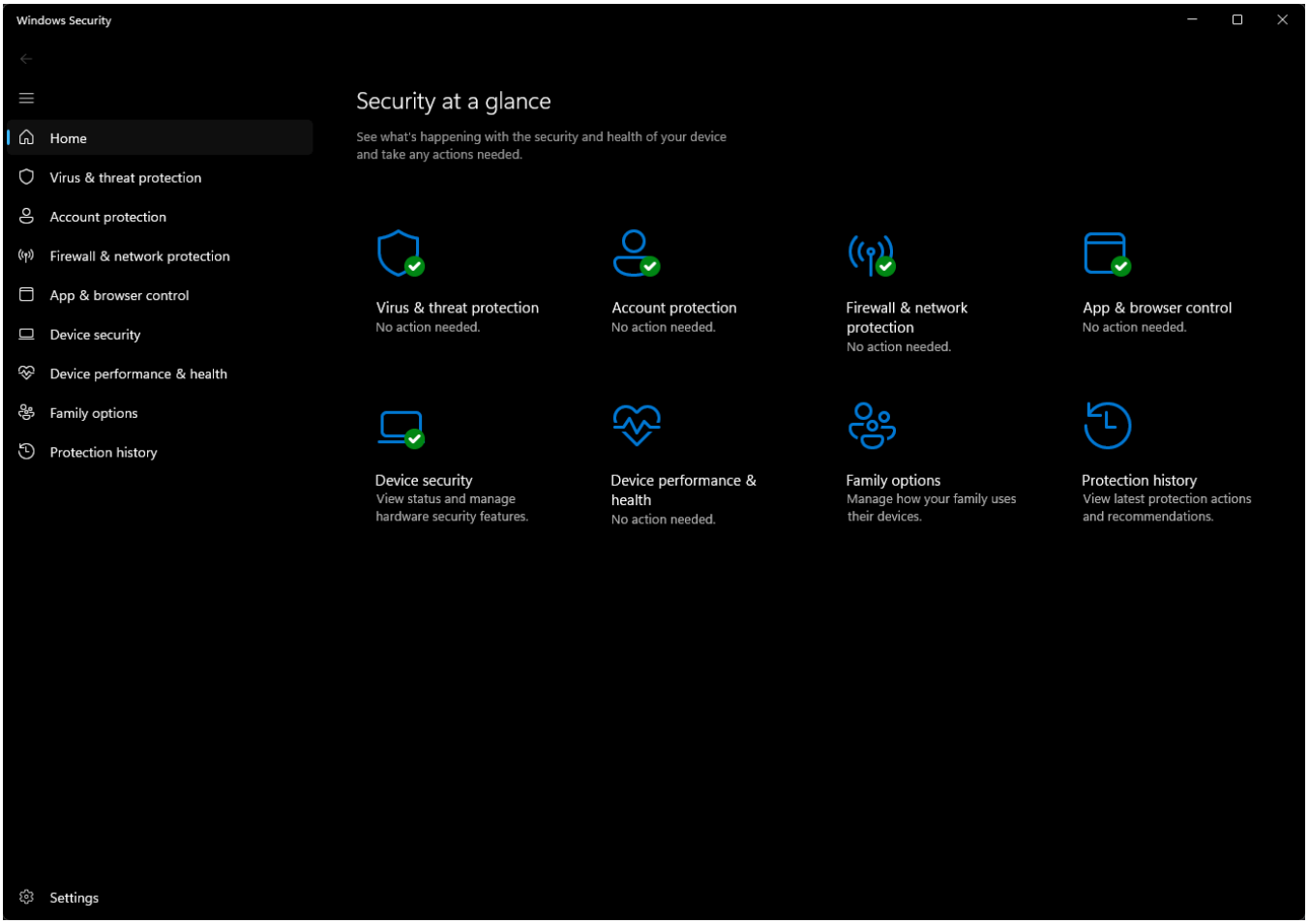
- Quickstart: configure a single-app kiosk with Assigned Access
- Quickstart: configure a restricted user experience with Assigned Access

Windows Security

Applies to: ✅ Windows 11, ✅ Windows 10

This article describes **Windows Security** settings, and provides information on configuring certain features, including:

- Showing and customizing contact information
- Hiding notifications



> **Note**
>
> **Windows Security** is a client interface on Windows 10, version 1703 and later. It isn't the Microsoft Defender Security Center web portal console that is used to review and manage Microsoft Defender for Endpoint.

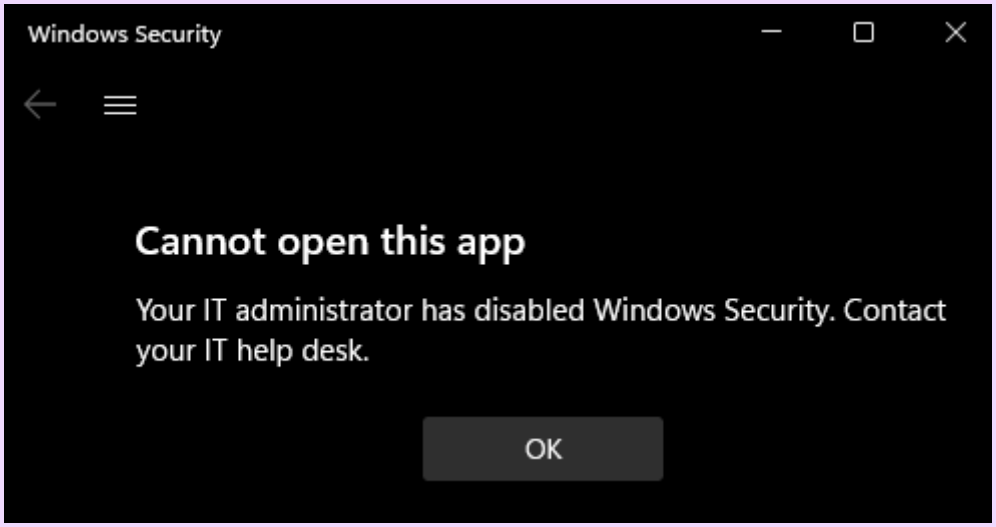
You can't uninstall **Windows Security**, but you can do one of the following actions:

- Disable the interface on Windows Server 2016.
- Hide all of the sections on client computers.
- Disable Microsoft Defender Antivirus, if needed. For more information, see Enable and configure Microsoft Defender Antivirus always-on protection in group policy.

For more information about each section, options for configuring the sections, and how to hide each of them, see the following articles:

- Virus & threat protection, which has information and access to antivirus ransomware protection settings and notifications, including Controlled folder access, and sign-in to Microsoft OneDrive.
- Account protection, which has information and access to sign-in and account protection settings.
- Firewall & network protection, which has information and access to firewall settings, including Windows Firewall.
- App & browser control, covering Windows Defender SmartScreen settings and Exploit protection mitigations.
- Device security, which provides access to built-in device security settings.
- Device performance & health, which has information about drivers, storage space, and general Windows Update issues.
- Family options, which include access to parental controls along with tips and information for keeping kids safe online.

> **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Open Windows Security

- Select the icon in the notification area on the taskbar.



- Search the Start menu for **Windows Security**.



- Open an area from Windows **Settings** > **Privacy & security** > **Windows Security**.



> ⓘ **Note**
>
> Settings configured with management tools, such as group policy, Microsoft Intune, or Microsoft Configuration Manager, take precedence over the settings in the Windows Security.

How Windows Security works with Windows security features

> ⓘ **Important**
>
> **Microsoft Defender Antivirus** and **Windows Security** use similarly named services for specific purposes.
>
> The **Windows Security** uses the Windows Security Service (*SecurityHealthService* or *Windows Security Health Service*), which in turn utilizes the Windows Security Center Service (*wscsvc*). This service makes sure that **Windows Security** provides the most up-to-date information about the protection status on the endpoint. This information includes protection offered by third-party antivirus products, Windows Firewall, third-party firewalls, and other security protection.
>
> These services don't affect the state of Microsoft Defender Antivirus. Disabling or modifying these services doesn't disable Microsoft Defender Antivirus. It leads to a lowered protection state on the endpoint, even if you're using a third-party antivirus product.
>
> Microsoft Defender Antivirus is disabled automatically when a third-party antivirus product is installed and kept up to date.

Windows Security operates as a separate app or process from each of the individual features, and displays notifications through the Action Center.

It acts as a collector or single place to see the status and perform some configuration for each of the features.

If you disable any of the individual features, it prevents that feature from reporting its status in **Windows Security**. For example, if you disable a feature through group policy or other management tools, such as Microsoft Configuration Manager, **Windows Security** itself still runs and shows status for the other security features.

> ⓘ **Important**
>
> If you individually disable any of the services, it doesn't disable the other services or **Windows Security** itself.

For example, using a third-party antivirus disables Microsoft Defender Antivirus. However, **Windows Security** still runs, shows its icon in the taskbar, and displays information about the

other features, such as Windows Defender SmartScreen and Windows Firewall.

> **⚠ Warning**
>
> Disabling the Windows Security Center Service doesn't disable Microsoft Defender Antivirus or **Windows Firewall**. If you disable the Windows Security Center Service, or configure its associated group policy settings to prevent it from starting or running, **Windows Security** might display stale or inaccurate information about any antivirus or firewall products installed on the device.
>
> It may also prevent Microsoft Defender Antivirus from enabling itself if you have an old or outdated third-party antivirus, or if you uninstall any third-party antivirus products you might have previously installed. This will significantly lower the protection of your device and could lead to malware infection.

Last updated on 04/15/2025

Virus and threat protection

Applies to: ✅ Windows 11, ✅ Windows 10

The **Virus & threat protection** section contains information and settings for antivirus protection from Microsoft Defender Antivirus and non-Microsoft antivirus products. These settings include Controlled folder access settings to prevent unknown apps from changing files in protected folders, plus Microsoft OneDrive configuration to help you recover from a ransomware attack. This area also notifies users and provides recovery instructions if there's a ransomware attack.

IT administrators and IT pros can get more configuration information from these articles:

- Microsoft Defender Antivirus in Windows Security
- Microsoft Defender Antivirus documentation library
- Protect important folders with Controlled folder access
- Defend yourself from cybercrime with new Office 365 capabilities
- Microsoft Defender for Office 365
- Ransomware detection and recovering your files

You can hide the **Virus & threat protection** section or the **Ransomware protection** area from users of the machine. This option can be useful if you don't want users in your organization to see or have access to user-configured options for these features.

Hide the Virus & threat protection section

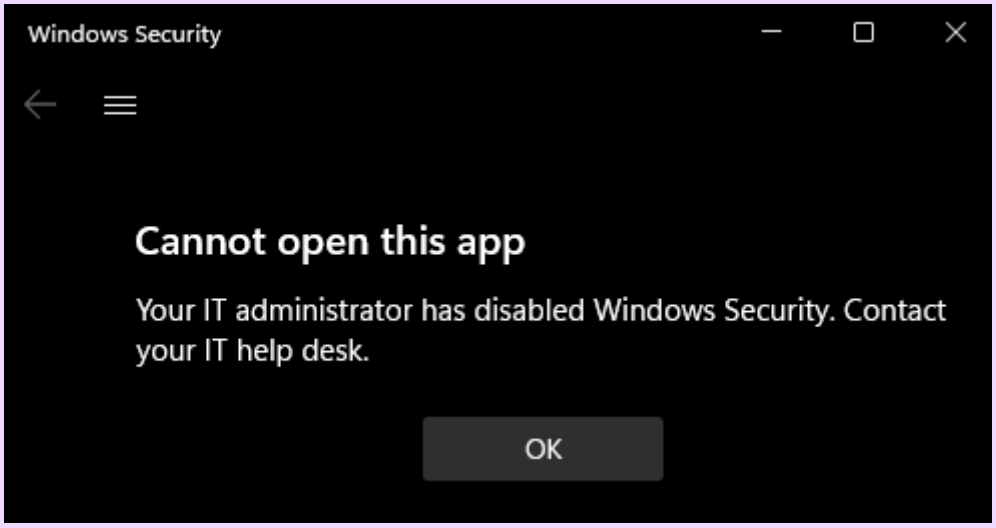
You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Virus and threat protection**.

4. Open the **Hide the Virus and threat protection area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Hide the Ransomware protection area

You can choose to hide the **Ransomware protection** area by using Group Policy. When hidden, this area doesn't appear on the **Virus & threat protection** section of **Windows Security**.

> ⓘ **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Virus and threat protection**.
4. Open the **Hide the Ransomware data recovery area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

Last updated on 04/15/2025

Account protection

Applies to: ✅ Windows 11, ✅ Windows 10

The **Account protection** section contains information and settings for account protection and sign-in. You can get more information about these capabilities from the following list:

- Microsoft Account
- Windows Hello for Business
- Lock your Windows 10 PC automatically when you step away from it

You can also choose to hide the section from users of the device, if you don't want your users to access or view user-configured options for these features.

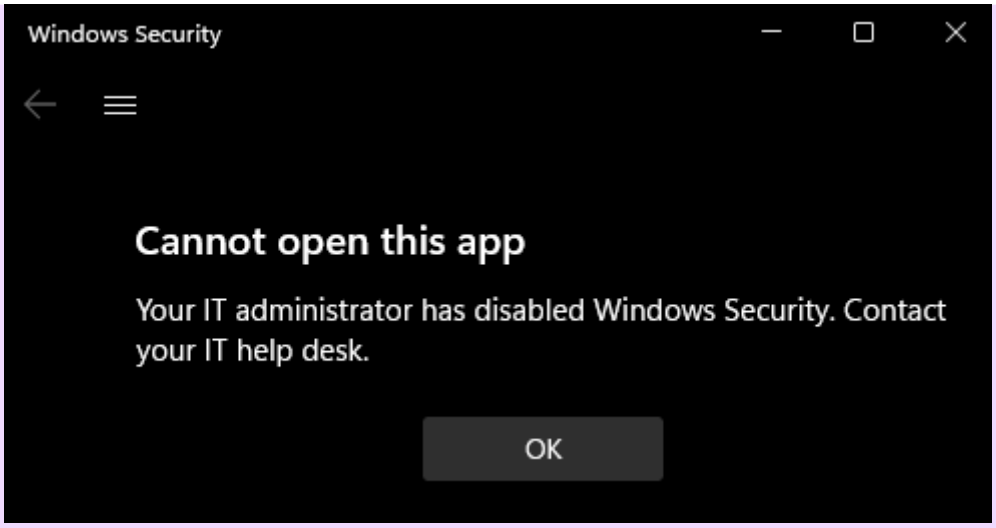
Hide the Account protection section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> ⓘ **Important**
>
> You must have Windows 10, version 1803 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In the **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Account protection**.
4. Open the **Hide the Account protection area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Last updated on 04/15/2025

Firewall and network protection

Applies to: ✅ Windows 11, ✅ Windows 10

The **Firewall & network protection** section contains information about the firewalls and network connections used by the machine, including the status of Windows Firewall and any other non-Microsoft firewalls. IT administrators and IT pros can get configuration guidance from the Windows Firewall with Advanced Security documentation library.

This section can be hidden from users of the machine. This information is useful if you don't want users in your organization to have access to user-configured options for the features shown in the section.

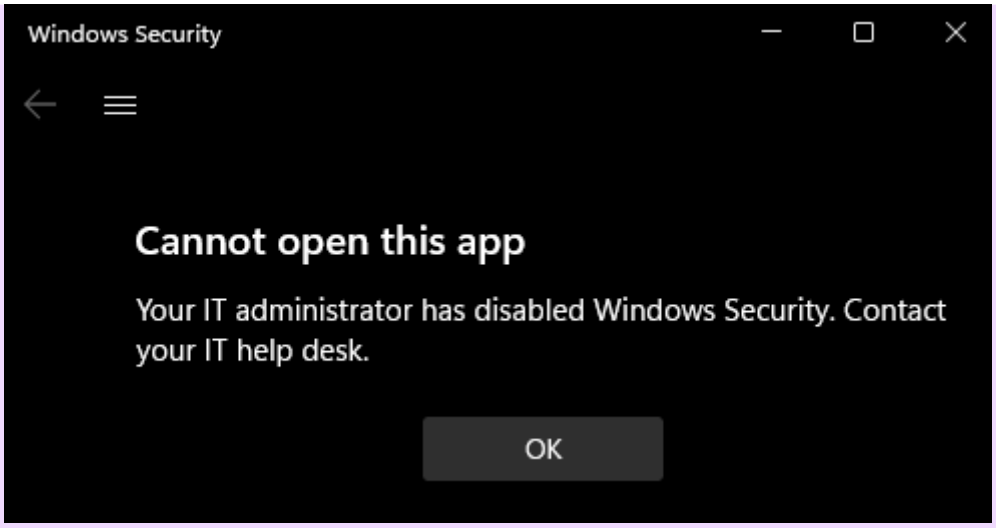
Hide the Firewall & network protection section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> ⓘ **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Firewall and network protection**.
4. Open the **Hide the Firewall and network protection area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Last updated on 04/15/2025

App and browser control

Applies to: ✅ Windows 11, ✅ Windows 10

The **App and browser control** section contains information and settings for Windows Defender SmartScreen. IT administrators and IT pros can get configuration guidance from the Windows Defender SmartScreen documentation library.

In Windows 10, version 1709 and later, the section also provides configuration options for Exploit protection. You can prevent users from modifying these specific options with Group Policy. IT administrators can get more information at Exploit protection.

You can also choose to hide the section from users of the machine. This option can be useful if you don't want users in your organization to have access to user-configured options for the features shown in the section.

Prevent users from making changes to the Exploit protection area in the App & browser control section

You can prevent users from modifying settings in the Exploit protection area. The settings are either grayed out or don't appear if you enable this setting. Users still have access to other settings in the App & browser control section, such as those settings for Windows Defender SmartScreen, unless those options are separately.

> ⓘ **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In the **Group Policy Management Editor**, go to **Computer configuration**, select **Policies** and then **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **App and browser protection**.
4. Select the **Prevent users from modifying settings** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

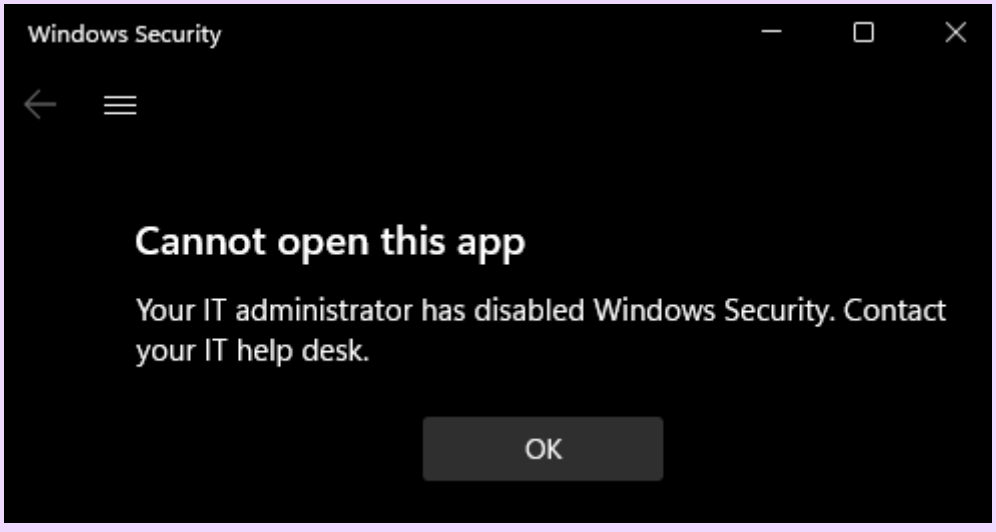
Hide the App & browser control section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.
2. In the **Group Policy Management Editor** go to **Computer configuration**, select **Policies** and then **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **App and browser protection**.
4. Open the **Hide the App and browser protection area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Last updated on 04/15/2025

Device security

Applies to: ✅ Windows 11, ✅ Windows 10

The **Device security** section contains information and settings for built-in device security.

You can choose to hide the section from users of the machine. This option can be useful if you don't want users in your organization to have access to user-configured options for the features shown in the section.

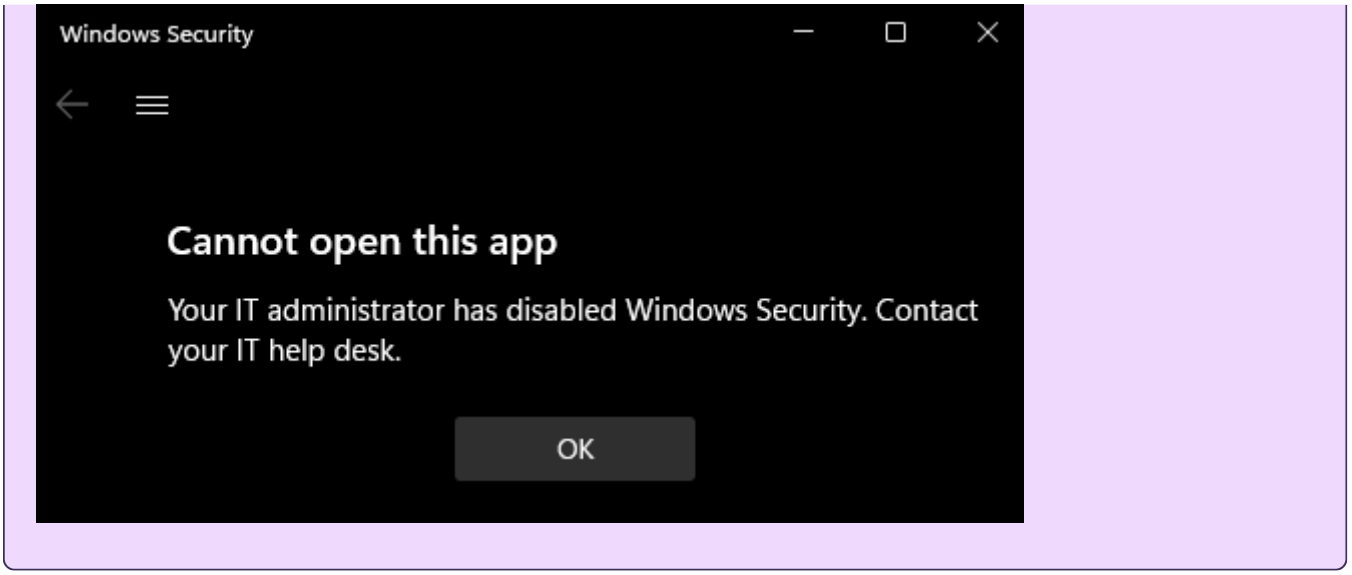
Hide the Device security section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> ⓘ **Important**
>
> You must have Windows 10, version 1803 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and then select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Device security**.
4. Open the **Hide the Device security area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Disable the Clear TPM button

If you don't want users to be able to select the **Clear TPM** button in **Windows Security**, you can disable it.

> **Important**
>
> You must have Windows 10, version 1809 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management computer, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and then select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Device security**.
4. Open the **Disable the Clear TPM button** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

Hide the TPM Firmware Update recommendation

If you don't want users to see the recommendation to update TPM firmware, you can disable it.

1. On your Group Policy management computer, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and then select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Device security**.

4. Open the **Hide the TPM Firmware Update recommendation** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

Last updated on 04/15/2025

Device performance and health

Applies to: ✅ Windows 11, ✅ Windows 10

The **Device performance & health** section contains information about hardware, devices, and drivers related to the machine.

This section can be hidden from users of the machine. This option can be useful if you don't want users in your organization to have access to user-configured options for the features shown in the section.

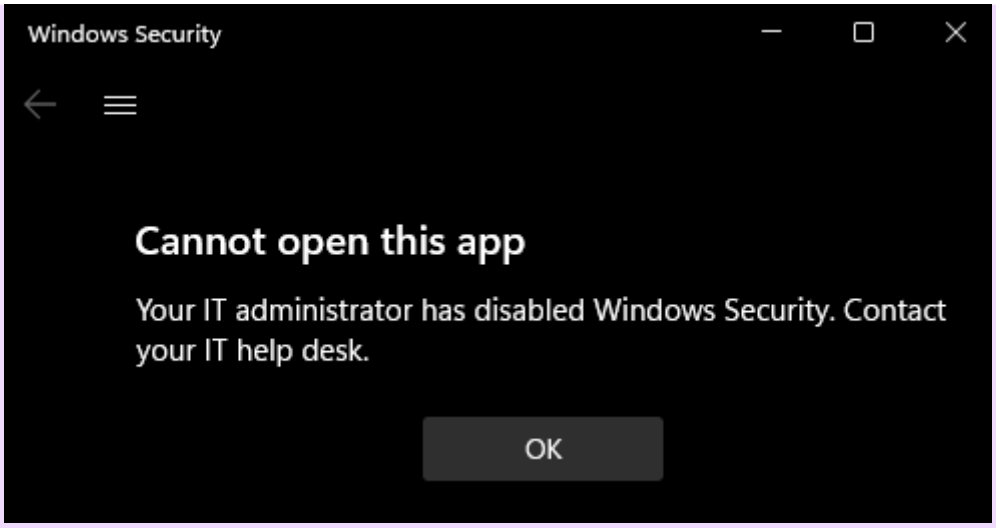
Hide the Device performance & health section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> ⓘ **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Device performance and health**.
4. Open the **Hide the Device performance and health area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:



Last updated on 04/15/2025

Family options

Applies to: ✅ Windows 11, ✅ Windows 10

The **Family options** section contains links to settings and further information for parents of a Windows PC. It isn't intended for enterprise or business environments.

Home users can learn more at the Help protection your family online in Windows Security article at support.microsoft.com

This section can be hidden from users of the machine. This option can be useful if you don't want users in your organization to see or have access to this section.

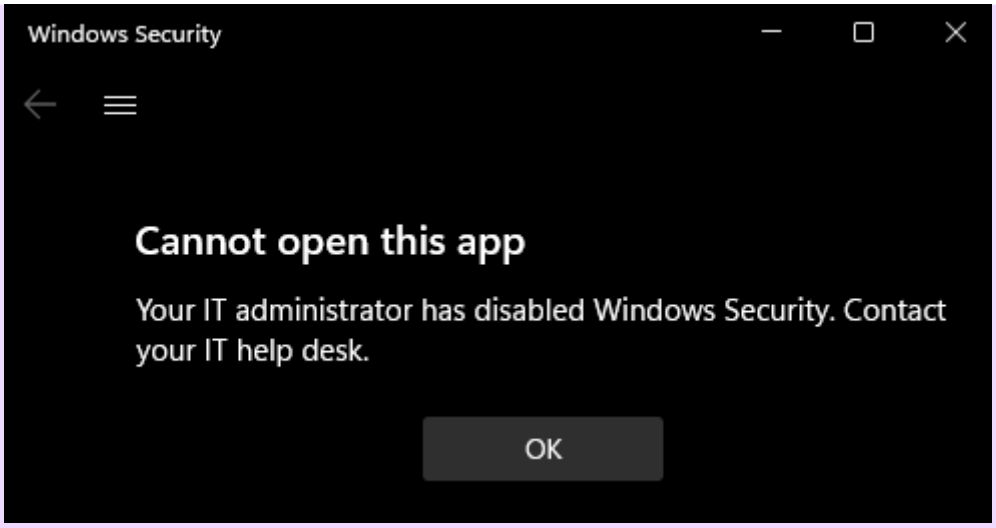
Hide the Family options section

You can choose to hide the entire section by using Group Policy. When hidden, this section doesn't appear on the home page of **Windows Security**, and its icon isn't shown on the navigation bar on the side.

> ⓘ **Important**
>
> You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Family options**.
4. Open the **Hide the Family options area** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> ⓘ **Note**
>
> If you hide all sections, then **Windows Security** shows a restricted interface, as in the following screenshot:

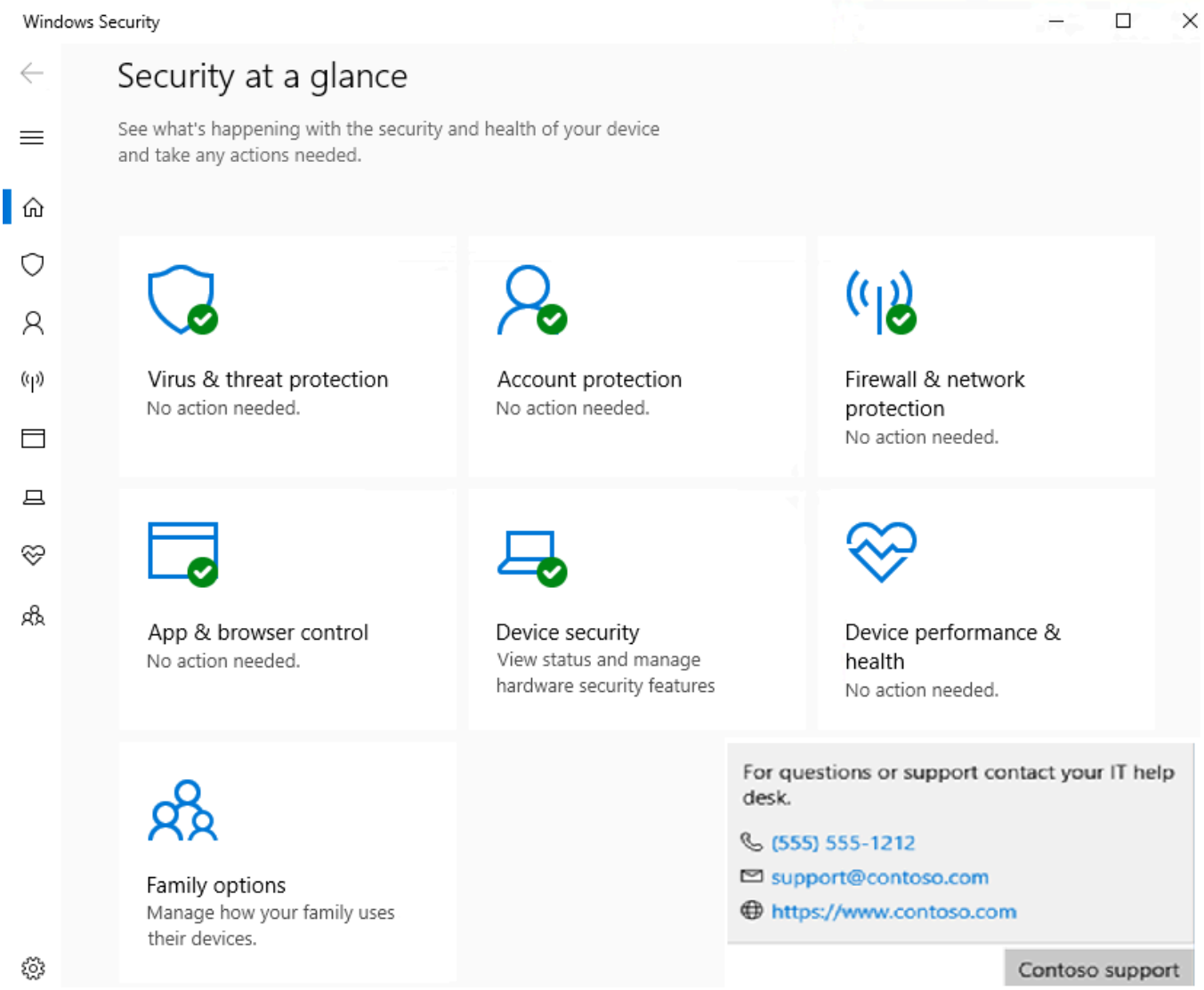


Last updated on 04/15/2025

Customize the Windows Security settings for your organization

Applies to: ✅ Windows 11, ✅ Windows 10

You can add information about your organization in a contact card in **Windows Security**. You can include a link to a support site, a phone number for a help desk, and an email address for email-based support.



This information is shown in some enterprise-specific notifications (including notifications for the Block at first sight feature, and potentially unwanted applications).

Users can select the displayed information to initiate a support request:

- Select **Call** or the phone number to open Skype to start a call to the displayed number.
- Select **Email** or the email address to create a new email in the machine's default email app addressed to the displayed email.
- Select **Help portal** or the website URL to open the machine's default web browser and go to the displayed address.

Requirements

You must have Windows 10, version 1709 or later. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

Use Group Policy to enable and customize contact information

There are two stages to using the contact card and customized notifications. First, you have to enable the contact card or custom notifications (or both), and then you must specify at least a name for your organization and one piece of contact information.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.

2. In the **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.

3. Expand the tree to **Windows components** > **Windows Security** > **Enterprise Customization**.

4. Enable the contact card and the customized notifications by configuring two separate Group Policy settings. They both use the same source of information (explained in Steps 5 and 6). You can enable both, or select one or the other:
   a. To enable the contact card, open the **Configure customized contact information** setting and set it to **Enabled**. Select **OK**.
   b. To enable the customized notifications, open the **Configure customized notifications** setting and set it to **Enabled**. Select **OK**.

5. After you enable the contact card or the customized notifications (or both), you must configure the **Specify contact company name** to **Enabled**. Enter your company or organization's name in the field in the **Options** section. Select **OK**.

6. To ensure the custom notifications or contact card appear, you must also configure at least one of the following settings. Open the setting, select **Enabled**, and then add the contact information in the field under **Options**:
   a. **Specify contact email address or Email ID**
   b. **Specify contact phone number or Skype ID**
   c. **Specify contact website**

> ⓘ **Note**

If you enable **Configure customized notifications** and **Specify contact website** policies, the contact website must begin with `http:` or `https:` (for example, `https://contoso.com/help`) to allow the user to interact with the notification and navigate to the specified URL.

7. Select **OK** after you configure each setting to save your changes.

To enable the customized notifications and add the contact information in Intune, see these articles:

- Manage device security with endpoint security policies in Microsoft Intune.
- Settings for the Windows Security experience profile in Microsoft Intune.

> **Important**
>
> You must specify the contact company name and at least one contact method - email, phone number, or website URL. If you don't specify the contact name and a contact method, the contact card isn't visible, and notifications aren't customized.

Last updated on 04/15/2025

Hide Windows Security notifications

Applies to: ✅ Windows 11, ✅ Windows 10

Windows Security is used by many Windows security features to provide notifications about the health and security of the machine. These include notifications about firewalls, antivirus products, Windows Defender SmartScreen, and others.

In some cases, it might not be appropriate to show these notifications, for example, if you want to hide regular status updates, or if you want to hide all notifications to the users in your organization.

There are two levels to hiding notifications:

1. Hide noncritical notifications, such as regular updates about the number of scans Microsoft Defender Antivirus ran in the past week
2. Hide all notifications

If you set **Hide all notifications** to **Enabled**, changing the **Hide non-critical notifications** setting has no effect.

Use Group Policy to hide noncritical notifications

You can hide notifications that describe regular events related to the health and security of the machine. These notifications are the ones that don't require an action from the machine's user. It can be useful to hide these notifications if you find they're too numerous or you have other status reporting on a larger scale (such as Windows Update for Business reports or Microsoft Configuration Manager reporting).

> ⓘ **Important**
>
> You must have Windows 10, version 1903 or higher. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. Download the latest Administrative Templates (.admx) for Windows 10, v2004.
2. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object (GPO) you want to configure and select **Edit**.
3. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
4. Expand the tree to **Windows components** > **Windows Security** > **Notifications**. For Windows 10 version 1803 and below, the path would be **Windows components** > **Windows Defender Security Center** > **Notifications**

5. Open the **Hide non-critical notifications** setting and set it to **Enabled**. Select **OK**.
6. Deploy the updated GPO as you normally do.

Use Group Policy to hide all notifications

You can hide all notifications that are sourced from **Windows Security**. This option might be useful if you don't want users of the machines from inadvertently modifying settings, running antivirus scans, or otherwise performing security-related actions without your input.

> **Important**
>
> You must have Windows 10, version 1903 or higher. The ADMX/ADML template files for earlier versions of Windows don't include these Group Policy settings.

1. On your Group Policy management machine, open the Group Policy Management Console. Right-click the Group Policy Object you want to configure and select **Edit**.
2. In **Group Policy Management Editor**, go to **Computer configuration** and select **Administrative templates**.
3. Expand the tree to **Windows components** > **Windows Security** > **Notifications**.
4. Open the **Hide all notifications** setting and set it to **Enabled**. Select **OK**.
5. Deploy the updated GPO as you normally do.

> **Note**
>
> You can use the following registry key and DWORD value to **Hide all notifications**.

```text
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender Security Center\Notifications]
  "DisableNotifications"=dword:00000001
```

You can use the following registry key and DWORD value to **Hide not-critical notifications**.

```text
[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows Defender Security Center\Notifications]
  "DisableEnhancedNotifications"=dword:00000001
```

Notifications

Expand table

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
Network isolation	Your IT administrator has caused Windows Defender to disconnect your device. Contact IT help desk.	SENSE_ISOLATION	Yes	Firewall and network protection notification
Network isolation customized	*Company name* has caused Windows Defender to disconnect your device. Contact IT help desk *phone number*, *email address*, *url*.	SENSE_ISOLATION_CUSTOM (body)	Yes	Firewall and network protection notification
Restricted access	Your IT administrator has caused Windows Defender to limit actions on this device. Some apps may not function as expected. Contact IT help desk.	SENSE_PROCESS_RESTRICTION	Yes	Firewall and network protection notification
Restricted access customized	*Company* has caused Windows Defender to limit actions on this device. Some apps may not function as expected. Contact IT help desk.	SENSE_PROCESS_RESTRICTION_CUSTOM (body)	Yes	Firewall and network protection notification

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
HVCI, driver compat check fails (upon trying to enable)	There may be an incompatibility on your device.	HVCI_ENABLE_FAILURE	Yes	Firewall and network protection notification
HVCI, reboot needed to enable	The recent change to your protection settings requires a restart of your device.	HVCI_ENABLE_SUCCESS	Yes	Firewall and network protection notification
Item skipped in scan, due to exclusion setting, or network scanning disabled by admin	The Microsoft Defender Antivirus scan skipped an item due to exclusion or network scanning settings.	ITEM_SKIPPED	Yes	Virus & threat protection notification
Remediation failure	Microsoft Defender Antivirus couldn't completely resolve potential threats.	CLEAN_FAILED	Yes	Virus & threat protection notification
Follow-up action (restart & scan)	Microsoft Defender Antivirus found *threat* in *file name*. Restart and scan your device. Restart and scan	MANUALSTEPS_REQUIRED	Yes	Virus & threat protection notification
Follow-up action (restart)	Microsoft Defender Antivirus found *threat* in *file*. Restart your device.	WDAV_REBOOT	Yes	Virus & threat protection notification

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
Follow-up action (Full scan)	Microsoft Defender Antivirus found *threat* in *file*. Run a full scan of your device.	FULLSCAN_REQUIRED	Yes	Virus & threat protection notification
Sample submission prompt	Review files that Windows Defender will send to Microsoft. Sending this information can improve how Microsoft Defender Antivirus helps protect your device.	SAMPLE_SUBMISSION_REQUIRED	Yes	Virus & threat protection notification
OS support ending warning	Support for your version of Windows is ending. When this support ends, Microsoft Defender Antivirus won't be supported, and your device might be at risk.	SUPPORT_ENDING	Yes	Virus & threat protection notification
OS support ended, device at risk	Support for your version of Windows has ended. Microsoft Defender Antivirus is no longer supported, and your device might be at risk.	SUPPORT_ENDED *and* SUPPORT_ENDED_NO_DEFENDER	Yes	Virus & threat protection notification
Summary notification, items found	Microsoft Defender Antivirus	RECAP_FOUND_THREATS_SCANNED	No	Virus & threat

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
	successfully took action on *n* threats since your last summary. Your device was scanned *n* times.			protection notification
Summary notification, items found, no scan count	Microsoft Defender Antivirus successfully took action on *n* threats since your last summary.	RECAP_FOUND_THREATS	No	Virus & threat protection notification
Summary notification, **no** items found, scans performed	Microsoft Defender Antivirus didn't find any threats since your last summary. Your device was scanned *n* times.	RECAP_NO THREATS_SCANNED	No	Virus & threat protection notification
Summary notification, **no** items found, no scans	Microsoft Defender Antivirus didn't find any threats since your last summary.	RECAP_NO_THREATS	No	Virus & threat protection notification
Scan finished, manual, threats found	Microsoft Defender Antivirus scanned your device at *timestamp* on *date*, and took action against threats.	RECENT_SCAN_FOUND_THREATS	No	Virus & threat protection notification
Scan finished, manual, **no** threats found	Microsoft Defender Antivirus scanned your device at	RECENT_SCAN_NO_THREATS	No	Virus & threat protection notification

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
	timestamp on *date*. No threats were found.			
Threat found	Microsoft Defender Antivirus found threats. Get details.	CRITICAL	No	Virus & threat protection notification
LPS on notification	Microsoft Defender Antivirus is periodically scanning your device. You're also using another antivirus program for active protection.	PERIODIC_SCANNING_ON	No	Virus & threat protection notification
Long running BaFS	Your IT administrator requires a security scan of this item. The scan could take up to *n* seconds.	BAFS	No	Firewall and network protection notification
Long running BaFS customized	*Company* requires a security scan of this item. The scan could take up to *n* seconds.	BAFS_DETECTED_CUSTOM (body)	No	Firewall and network protection notification
Sense detection	This application was removed because it was blocked by your IT security settings	WDAV_SENSE_DETECTED	No	Firewall and network protection notification
Sense detection customized	This application was removed because it was blocked by your	WDAV_SENSE_DETECTED_CUSTOM (body)	No	Firewall and network protection notification

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
	IT security settings			
Ransomware specific detection	Microsoft Defender Antivirus has detected threats, which may include ransomware.	WDAV_RANSOMWARE_DETECTED	No	Virus & threat protection notification
ASR (HIPS) block	Your IT administrator caused Windows Defender Security Center to block this action. Contact your IT help desk.	HIPS_ASR_BLOCKED	No	Firewall and network protection notification
ASR (HIPS) block customized	*Company* caused Windows Defender Security Center to block this action. Contact your IT help desk.	HIPS_ASR_BLOCKED_CUSTOM (body)	No	Firewall and network protection notification
CFA (FolderGuard) block	Controlled folder access blocked *process* from making changes to the folder *path*	FOLDERGUARD_BLOCKED	No	Firewall and network protection notification
Network protect (HIPS) network block customized	*Company* caused Windows Defender Security Center to block this network connection. Contact your IT help desk.	HIPS_NETWORK_BLOCKED_CUSTOM (body)	No	Firewall and network protection notification

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
Network protection (HIPS) network block	Your IT administrator caused Windows Defender Security Center to block this network connection. Contact your IT help desk.	HIPS_NETWORK_BLOCKED	No	Firewall and network protection notification
PUA detection, not blocked	Your settings cause the detection of any app that might perform unwanted actions on your computer.	PUA_DETECTED	No	Firewall and network protection notification
PUA notification	Your IT settings caused Microsoft Defender Antivirus to block an app that may potentially perform unwanted actions on your device.	PUA_BLOCKED	No	Firewall and network protection notification
PUA notification, customized	*Company* caused Microsoft Defender Antivirus to block an app that may potentially perform unwanted actions on your device.	PUA_BLOCKED_CUSTOM (body)	No	Firewall and network protection notification
Network isolation ended			No	Firewall and network

Purpose	Notification text	Toast Identifier	Critical?	Notification Toggle
				protection notification
Network isolation ended, customized			No	Firewall and network protection notification
Restricted access ended			No	Firewall and network protection notification
Restricted access ended, customized			No	Firewall and network protection notification
Dynamic lock on, but bluetooth off			No	Account protection notification
Dynamic lock on, bluetooth on, but device unpaired			No	Account protection notification
Dynamic lock on, bluetooth on, but unable to detect device			No	Account protection notification
NoPa or federated no hello			No	Account protection notification
NoPa or federated hello broken			No	Account protection notification

Last updated on 04/15/2025

BitLocker overview

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Note**
>
> This article is intended for IT professionals and device administrators. If you're an end user or consumer looking for help with BitLocker, see **BitLocker overview**.

BitLocker is a Windows security feature that provides encryption for entire volumes, addressing the threats of data theft or exposure from lost, stolen, or inappropriately decommissioned devices.

Data on a lost or stolen device is vulnerable to unauthorized access, either by running a software-attack tool against it, or by transferring the device's hard drive to a different device. BitLocker helps mitigate unauthorized data access by enhancing file and system protections, rendering data inaccessible when BitLocker-protected devices are decommissioned or recycled.

BitLocker and TPM

BitLocker provides maximum protection when used with a Trusted Platform Module (TPM), which is a common hardware component installed on Windows devices. The TPM works with BitLocker to ensure that a device hasn't been tampered with while the system is offline.

In *addition* to the TPM, BitLocker can lock the normal startup process until the user supplies a personal identification number (PIN) or inserts a removable device that contains a *startup key*. These security measures provide multifactor authentication and assurance that the device can't start or resume from hibernation until the correct PIN or startup key is presented.

On devices that don't have a TPM, BitLocker can still be used to encrypt the operating system drive. This implementation requires the user to either:

- Use a startup key, which is a file stored on a removable drive that is used to start the device, or when resuming from hibernation.
- Use a password. This option isn't secure since it's subject to brute force attacks as there isn't a password lockout logic. As such, the password option is discouraged and disabled by default.

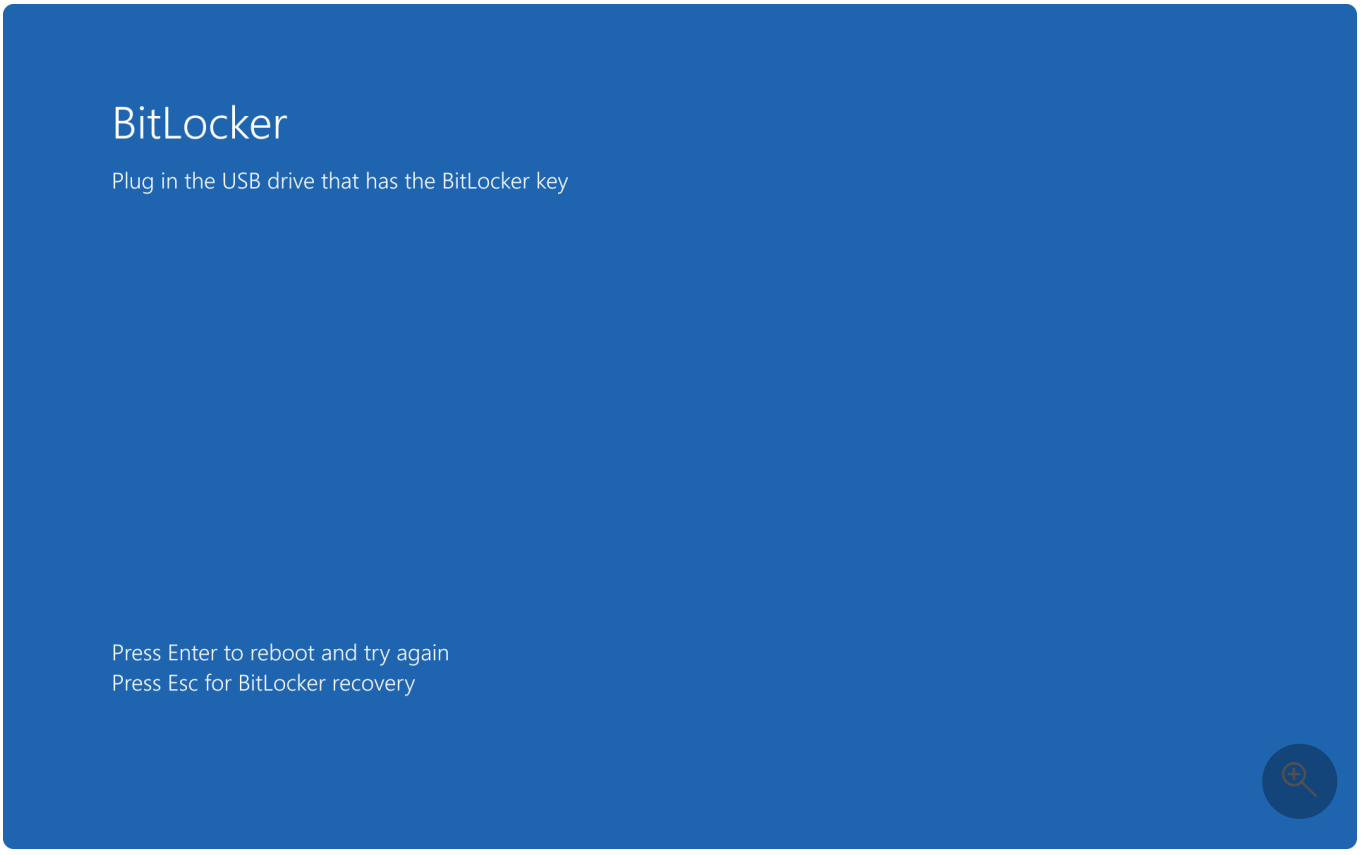
Both options don't provide the preboot system integrity verification offered by BitLocker with a TPM.

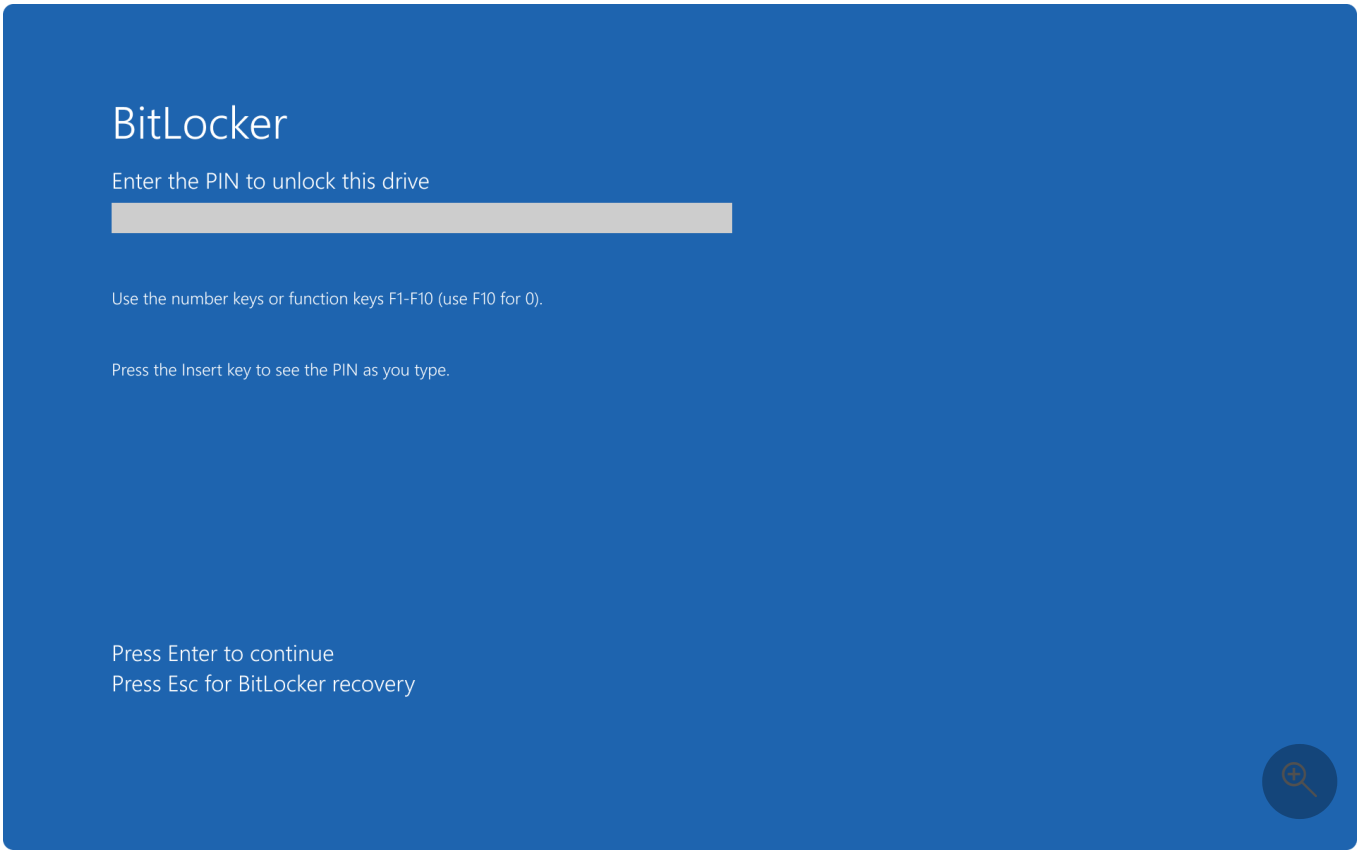
The following screenshots show the BitLocker preboot screen when a startup key, PIN, or password is required to start the device:

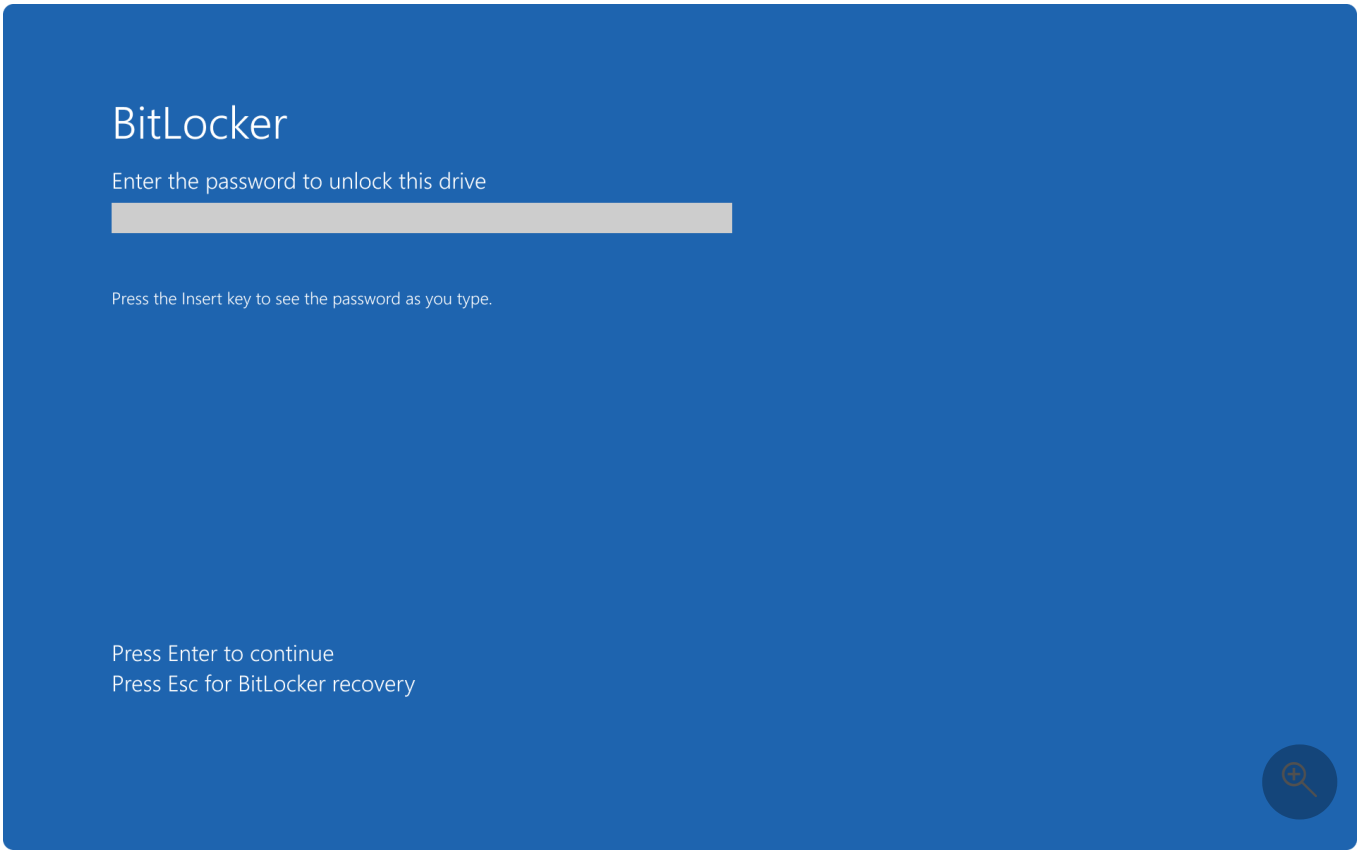
BitLocker preboot screen with startup key:

BitLocker preboot screen with PIN:

BitLocker preboot screen with password:







System requirements

BitLocker has the following requirements:

- For BitLocker to use the system integrity check provided by a TPM, the device must have TPM 1.2 or later versions. If a device doesn't have a TPM, saving a startup key on a removable drive is mandatory when enabling BitLocker.
- A device with a TPM must also have a *Trusted Computing Group* (TCG)-compliant BIOS or UEFI firmware. The BIOS or UEFI firmware establishes a chain of trust for the preboot startup, and it must include support for *TCG-specified Static Root of Trust Measurement*. A computer without a TPM doesn't require TCG-compliant firmware.
- The system BIOS or UEFI firmware (for TPM and non-TPM devices) must support the USB mass storage device class, and reading files on a USB drive in the preboot environment.

> **Note**
>
> TPM 2.0 is not supported in *Legacy* and *Compatibility Support Module (CSM)* modes of the BIOS. Devices with TPM 2.0 must have their BIOS mode configured as native UEFI only. The Legacy and CSM options must be disabled. For added security, enable the *secure boot* feature.
>
> Installed operating system on hardware in Legacy mode stops the OS from booting when the BIOS mode is changed to UEFI. Use the tool **mbr2gpt.exe** before changing the BIOS

mode, which prepares the OS and the disk to support UEFI.

- The hard disk must be partitioned with at least two drives:
  - The *operating system drive* (or boot drive) contains the OS and its support files. It must be formatted with the NTFS file system.
  - The *system drive* contains files required to boot, decrypt, and load the operating system. BitLocker isn't enabled on this drive. For BitLocker to work, the system drive:
    - must not be encrypted.
    - must differ from the operating system drive.
    - must be formatted with the FAT32 file system on computers that use UEFI-based firmware, or with the NTFS file system on computers that use BIOS firmware.
    - it's recommended that to be approximately 350 MB in size. After BitLocker is turned on, it should have approximately 250 MB of free space.

    > **Important**
    >
    > When installed on a new device, Windows automatically creates the partitions that are required for BitLocker.
    >
    > If the drive was prepared as a single contiguous space, BitLocker requires a new volume to hold the boot files. `BdeHdCfg.exe` can create the volume. For more information about using the tool, see Bdehdcfg in the Command-Line Reference.

> **Note**
>
> When installing the BitLocker optional component on a server, the *Enhanced Storage* feature must be installed. The feature is used to support hardware encrypted drives.

Windows edition and licensing requirements

The following table lists the Windows editions that support BitLocker enablement:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

BitLocker enablement license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

> **Note**
>
> Licensing requirements for BitLocker enablement are different from the licensing requirements for BitLocker *management*. To learn more, review the how-to guide: **configure BitLocker**.

Device encryption

Device encryption is a Windows feature that provides a simple way for some devices to enable BitLocker encryption automatically. Device encryption is available on all Windows versions, and it requires a device to meet either Modern Standby or HSTI security requirements. Device encryption can't have externally accessible ports that allow DMA access. Device encryption encrypts only the OS drive and fixed drives, it doesn't encrypt external/USB drives.

> **Important**
>
> Starting in Windows 11, version 24H2, the prerequisites of DMA and HSTI/Modern Standby are removed. As a result, more devices are eligible for automatic and manual device encryption. For more information, see **BitLocker drive encryption in Windows 11 for OEMs**.

Unlike a standard BitLocker implementation, device encryption is enabled automatically so that the device is always protected. When a clean installation of Windows is completed and the out-of-box experience is finished, the device is prepared for first use. As part of this preparation, device encryption is initialized on the OS drive and fixed data drives on the computer with a clear key that is the equivalent of standard BitLocker suspended state. In this state, the drive is shown with a warning icon in Windows Explorer. The yellow warning icon is removed after the TPM protector is created and the recovery key is backed up.

- If the device is Microsoft Entra joined or Active Directory domain joined, the clear key is removed once the recovery key is successfully backed up to Microsoft Entra ID or Active Directory Domain Services (AD DS). The following policy settings must be enabled for the recovery key to be backed up: Choose how BitLocker-protected operating system drives can be recovered.
  - For Microsoft Entra joined devices: the recovery password is created automatically when the user authenticates to Microsoft Entra ID, then the recovery key is backed up to Microsoft Entra ID, the TPM protector is created, and the clear key is removed.

    - For AD DS joined devices: the recovery password is created automatically when the computer joins the domain. The recovery key is then backed up to AD DS, the TPM protector is created, and the clear key is removed.
- If the device isn't Microsoft Entra joined or Active Directory domain joined, a Microsoft account with administrative privileges on the device is required. When the administrator uses a Microsoft account to sign in, the clear key is removed, a recovery key is uploaded to the online Microsoft account, and a TPM protector is created. Should a device require the recovery key, the user is guided to use an alternate device and navigate to a recovery key access URL to retrieve the recovery key by using their Microsoft account credentials.
- If a device uses only local accounts, then it remains unprotected even though the data is encrypted.

> **Important**
>
> Device encryption uses the `XTS-AES 128-bit` encryption method, by default. In case you configure a policy setting to use a different encryption method, you can use the Enrollment Status Page to avoid the device to begin encryption with the default method. BitLocker has a logic that doesn't start encrypting until the end of OOBE, after the Enrollment Status Page device configuration phase is complete. This logic gives a device enough time to receive the BitLocker policy settings before starting encryption.
>
> If a different encryption method and/or cipher strength is needed but the device is already encrypted, it must first be decrypted before the new encryption method and/or cipher strength can be applied. After the device is decrypted, you can apply different BitLocker settings.

If a device doesn't initially qualify for device encryption, but then a change is made that causes the device to qualify (for example, by turning on *Secure Boot*), device encryption enables BitLocker automatically as soon as it detects it.

You can check whether a device meets requirements for device encryption in the System Information app (`msinfo32.exe`). If the device meets the requirements, System Information shows a line that reads:

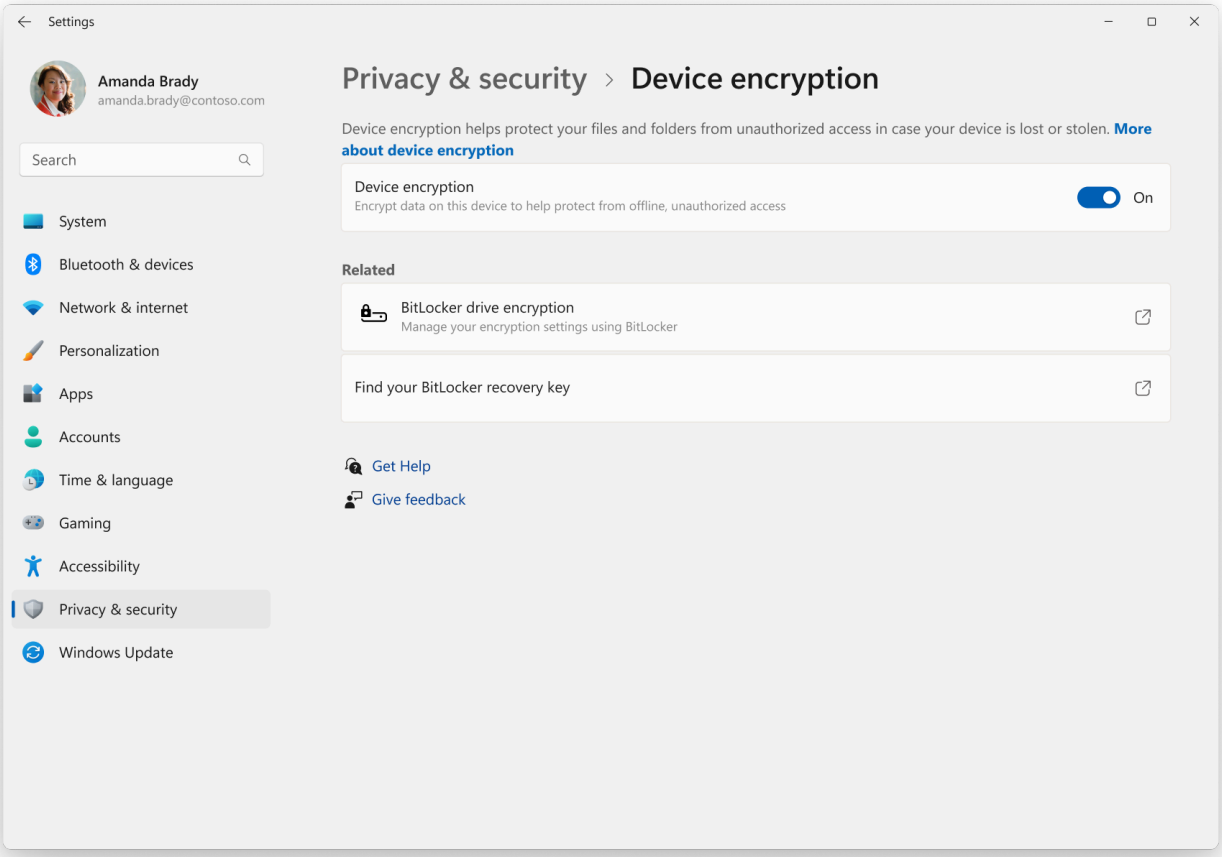
Expand table

Item	Value
Device Encryption Support	Meets prerequisites

> **Note**
>
> If Windows detects user activity during encryption, automatic device encryption might be delayed or temporarily halted; this is particularly true if the device is operating on battery to conserve power.

Difference between BitLocker and device encryption

- Device encryption turns on BitLocker automatically on device encryption-qualifying devices, with the recovery key automatically backed up to Microsoft Entra ID, AD DS, or the user's Microsoft account.
- Device encryption adds a device encryption setting in the Settings app, which can be used to turn device encryption on or off.
  - The Settings UI doesn't show device encryption enabled until encryption is complete.



> **Note**
>
> If device encryption is turned off, it will no longer automatically enable itself in the future. The user must enable it manually in Settings

Disable device encryption

It's recommended to keep device encryption on for any systems that support it. However, you can prevent the automatic device encryption process by changing the following registry setting:

Expand table

Path	Name	Type	Value
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\BitLocker`	`PreventDeviceEncryption`	REG_DWORD	0x1

For more information about device encryption, see BitLocker device encryption hardware requirements.

Next steps

Learn about technologies and features to protect against attacks on the BitLocker encryption key:

BitLocker countermeasures >

Last updated on 07/29/2025

BitLocker countermeasures

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows uses hardware solutions and security features that protect BitLocker encryption keys against attacks. These technologies include *Trusted Platform Module (TPM)*, *Secure Boot*, and *Measured Boot*.

Protection before startup

Before Windows starts, security features implemented as part of the device hardware and firmware must be relied on, including TPM and secure boot:

- a *TPM* is a chip designed to provide basic security-related functions, primarily involving encryption keys. BitLocker binds encryption keys with the TPM to ensure that the device hasn't been tampered with while the system is offline. For more information about TPM, see Trusted Platform Module
- *Unified Extensible Firmware Interface (UEFI)* is a programmable boot environment that initializes devices and starts the operating system's bootloader. The UEFI specification defines a firmware execution authentication process called Secure Boot
- *Secure Boot* blocks untrusted firmware and bootloaders (signed or unsigned) from being able to start on the system. By default, BitLocker provides integrity protection for Secure Boot by utilizing the TPM PCR[7] measurement. An unauthorized EFI firmware, EFI boot application, or bootloader can't run and acquire the BitLocker key

BitLocker and reset attacks

To defend against malicious reset attacks, BitLocker uses the *TCG Reset Attack Mitigation*, also known as *MOR bit* (Memory Overwrite Request), before extracting keys into memory.

Security policies

Preboot authentication and DMA policies provide extra protection for BitLocker.

Preboot authentication

Preboot authentication with BitLocker can require the use of either user input, such as a PIN, a startup key, or both to authenticate prior to making the contents of the system drive accessible.

BitLocker accesses and stores the encryption keys in memory only after preboot authentication is completed. If Windows can't access the encryption keys, the device can't read or edit the files on the system drive. The only option for bypassing preboot authentication is entering the *recovery key*.

Preboot authentication is designed to prevent the encryption keys from being loaded to system memory without the trusted user supplying another authentication factor. This feature helps mitigate DMA and memory remanence attacks.

On devices with a compatible TPM, operating system drives that are BitLocker-protected can be unlocked in four ways:

- **TPM-only**: this option doesn't require any interaction with the user to unlock and provide access to the drive. If the TPM validation succeeds, the user sign-in experience is the same as a standard sign-in. If the TPM is missing or changed, or if BitLocker detects changes to the BIOS or UEFI configuration, critical operating system startup files, or the boot configuration, BitLocker enters recovery mode. The user must then enter a recovery password to regain access to the data. This option is more convenient for sign-in but less secure than the other options, which require an additional authentication factor
- **TPM with startup key**: in addition to the protection that the TPM-only provides, part of the encryption key is stored on a USB flash drive, referred to as a *startup key*. Data on the encrypted volume can't be accessed without the startup key
- **TPM with PIN**: in addition to the protection that the TPM provides, BitLocker requires that the user enters a PIN. Data on the encrypted volume can't be accessed without entering the PIN. TPMs also have anti-hammering protection that is designed to prevent brute force attacks that attempt to determine the PIN
- **TPM with startup key and PIN**: in addition to the protection that the TPM provides, part of the encryption key is stored on a USB flash drive, and a PIN is required to authenticate the user to the TPM. This configuration provides multifactor authentication so that if the USB key is lost or stolen, it can't be used for access to the drive, because the PIN is also required

Preboot authentication with a PIN can mitigate an attack vector for devices that use a bootable eDrive because an exposed eDrive bus can allow an attacker to capture the BitLocker encryption key during startup. Preboot authentication with a PIN can also mitigate DMA port attacks during the window of time between when BitLocker unlocks the drive and Windows boots to the point that Windows can set any port-related policies that have been configured.

On the other hand, Preboot authentication-prompts can be inconvenient to users. In addition, users who forget their PIN or lose their startup key are denied access to their data until they can contact their organization's support team to obtain a recovery key. Preboot authentication

can also make it more difficult to update unattended or remotely administered devices because a PIN must be entered when a device reboots or resumes from hibernation.

To address these issues, BitLocker Network Unlock can be deployed. Network Unlock allows systems that meet the hardware requirements and have BitLocker enabled with TPM+PIN to boot into Windows without user intervention. It requires direct ethernet connectivity to a Windows Deployment Services (WDS) server.

To learn more, see the policy setting Require additional authentication at startup.

Protect DMA ports

It's important to protect DMA ports, as external peripherals might gain unauthorized access to memory. Depending on the device capabilities, there are different options to protect DMA ports. To learn more, see the policy setting Disable new DMA devices when this computer is locked.

Attack countermeasures

This section covers countermeasures for specific types of attacks.

Bootkits and rootkits

A physically present attacker might attempt to install a bootkit or rootkit-like piece of software into the boot chain in an attempt to steal the BitLocker keys. The TPM should observe this installation via PCR measurements, and the BitLocker key isn't released.

> **Note**
>
> BitLocker protects against this attack by default.

A BIOS password is recommended for defense-in-depth in case a BIOS exposes settings that might weaken the BitLocker security promise. Intel Boot Guard and AMD Hardware Verified Boot support stronger implementations of Secure Boot that provide additional resilience against malware and physical attacks. Intel Boot Guard and AMD Hardware Verified Boot are part of platform boot verification standards for a highly secure Windows device.

Brute force attacks against a PIN

Require TPM + PIN for anti-hammering protection.

DMA attacks

See Protect DMA ports earlier in this article.

Paging file, crash dump, and Hyberfil.sys attacks

These files are secured on an encrypted volume by default when BitLocker is enabled on OS drives. It also blocks automatic or manual attempts to move the paging file.

Memory remanence

Enable secure boot and mandatorily use a password to change BIOS settings. For scenarios requiring protection against these advanced attacks, configure a `TPM+PIN` protector, disable *standby* power management, and shut down or hibernate the device before it leaves the control of an authorized user.

The Windows default power settings cause devices to enter *sleep mode* when idle. When a device transitions to sleep, running programs and documents are persisted in memory. When a device resumes from sleep, users aren't required to reauthenticate with a PIN or USB startup key to access encrypted data. This scenario might lead to conditions where data security is compromised.

When a device *hibernates*, the drive is locked. When the device resumes from hibernation, the drive is unlocked, which means that users must provide a PIN or a startup key if using multifactor authentication with BitLocker.

Therefore, organizations that use BitLocker might want to use Hibernate instead of Sleep for improved security.

> **Note**
>
> This setting doesn't have an impact on TPM-only mode, because it provides a transparent user experience at startup and when resuming from the Hibernate states.

Tricking BitLocker to pass the key to a rogue operating system

An attacker might modify the boot manager configuration database (BCD), which is stored on a nonencrypted partition and add an entry point to a rogue operating system on a different partition. During the boot process, BitLocker code makes sure that the operating system that the encryption key obtained from the TPM is given to, is cryptographically verified to be the

intended recipient. Because this strong cryptographic verification already exists, we don't recommend storing a hash of a disk partition table in PCR 5.

An attacker might also replace the entire operating system disk while preserving the platform hardware and firmware, and could then extract a protected BitLocker key blob from the metadata of the victim OS partition. The attacker could then attempt to unseal that BitLocker key blob by calling the TPM API from an operating system under their control. This can't succeed because when Windows seals the BitLocker key to the TPM, it does it with a PCR 11 value of 0. To successfully unseal the blob, PCR 11 in the TPM must have a value of 0. However, when the boot manager passes the control to any boot loader (legitimate or rogue), it always changes PCR 11 to a value of 1. Since the PCR 11 value is guaranteed to be different after exiting the boot manager, the attacker can't unlock the BitLocker key.

Attacker countermeasures

The following sections cover mitigations for different types of attackers.

Attacker without much skill or with limited physical access

Physical access might be limited in a form factor that doesn't expose buses and memory. For example, there are no external DMA-capable ports, no exposed screws to open the chassis, and memory is soldered to the mainboard.

This attacker of opportunity doesn't use destructive methods or sophisticated forensics hardware/software.

Mitigation:

- Preboot authentication set to TPM only (the default)

Attacker with skill and lengthy physical access

Targeted attack with plenty of time; the attacker opens the case, solder, and uses sophisticated hardware or software.

Mitigation:

- Preboot authentication set to TPM with a PIN protector (with a sophisticated alphanumeric PIN [enhanced pin] to help the TPM anti-hammering mitigation).

  -And-

- Disable Standby power management and shut down or hibernate the device before it leaves the control of an authorized user. This configuration can be set using the following policy settings:
  - **Computer Configuration** > **Policies** > **Administrative Templates** > **Windows Components** > **File Explorer** > **Show hibernate in the power options menu**
  - **Computer Configuration** > **Policies** > **Administrative Templates** > **Power Management** > **Sleep Settings** >
    - **Allow standby states (S1-S3) when sleeping (plugged in)**
    - **Allow standby states (S1-S3) when sleeping (on battery)**

> **Important**
>
> These settings are **not configured** by default.

For some systems, bypassing TPM-only might require opening the case and require soldering, but can be done for a reasonable cost. Bypassing a TPM with a PIN protector would cost more, and require brute forcing the PIN. With a sophisticated enhanced PIN, it could be nearly impossible. To learn more about the policy setting, see Allow enhanced PINs for startup.

For secure administrative workstations, it's recommended to:

- use a TPM with PIN protector
- disable standby power management
- shut down or hibernate the device before it leaves the control of an authorized user

Next steps

Learn how to plan for a BitLocker deployment in your organization:

BitLocker planning guide >

Last updated on 07/29/2025

BitLocker planning guide

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

A BitLocker deployment strategy includes defining the appropriate policies and configuration requirements based on your organization's security requirements. This article helps collecting the information to assist with a BitLocker deployment.

Audit the environment

To plan a BitLocker deployment, understand the current environment. Perform an informal audit to define the current policies, procedures, and hardware environment. Review the existing disk encryption software and the organization's security policies. If the organization isn't using disk encryption software, then these policies might not exist. If disk encryption software is in use, then the policies might need to change to use certain BitLocker features.

To help document the organization's current disk encryption security policies, answer the following questions:

Expand table

☑	Question
☐	*Are there policies to determine which devices must use BitLocker and which don't?*
☐	*What policies exist to control recovery password and recovery key storage?*
☐	*What are the policies for validating the identity of users who need to perform BitLocker recovery?*
☐	*What policies exist to control who in the organization has access to recovery data?*
☐	*What policies exist to control the decommission or retirement of devices?*
☐	*What encryption algorithm strength is in place?*

Encryption keys and authentication

A trusted platform module (TPM) is a hardware component installed in many Windows devices by the manufacturers. It works with BitLocker to help protect user data and to make sure a device hasn't been tampered with while the system was offline.

BitLocker can lock the normal startup process until the user supplies a personal identification number (PIN), or inserts a removable USB device that contains a startup key. These extra

security measures provide multifactor authentication. They also make sure that the computer doesn't start or resume from hibernation until the correct PIN or startup key is presented.

On devices that don't have a TPM, BitLocker can still be used to encrypt the Windows operating system volume. However, this implementation doesn't provide the pre-startup system integrity verification offered by BitLocker working with a TPM.

An effective implementation of information protection, like most security controls, considers usability and security. Users typically prefer a simple security experience. In fact, the more transparent a security solution becomes, the more likely users are to conform to it.

It's crucial that organizations protect information on their devices regardless of the state of the device or the intent of users. This protection shouldn't be cumbersome to users. One undesirable and previously commonplace situation is when the user is prompted for input during preboot, and then again during Windows sign-in. Challenging users for input more than once should be avoided.

The TPM is able to securely protect the BitLocker encryption key while it is at rest, and it can securely unlock the operating system drive. When the key is in use, and thus in memory, a combination of hardware and Windows capabilities can secure the key and prevent unauthorized access through cold-boot attacks. Although other countermeasures like PIN-based unlock are available, they aren't as user-friendly; depending on the devices' configuration they may not offer more security when it comes to key protection. For more information, see BitLocker countermeasures.

BitLocker key protectors

To protect the BitLocker encryption key, BitLocker can use different types of *protectors*. When enabling BitLocker, each protector receives a copy of the *Volume Master Key*, which is then encrypted by using its own mechanism.

Expand table

Key protector	Description
Auto-unlock	Used to automatically unlock volumes that don't host an operating system. BitLocker uses encrypted information stored in the registry and volume metadata to unlock any data volumes that use automatic unlocking.
Password and **Password for OS drive**	To unlock a drive, the user must supply a password. When used for OS drives, the user is prompted for a password in the preboot screen. This method doesn't offer any lockout logic, therefore it doesn't protect against brute force attacks.

Key protector	Description
Startup key	An encryption key that can be stored on removable media, with a file name format of `<protector_id>.bek`. The user is prompted for the USB flash drive that has the recovery key and/or startup key, and then reboot the device.
Smart card certificate	Used to unlock volumes that don't host an operating system. To unlock a drive, the user must use a smart card.
TPM	A hardware device used to help establish a secure root-of-trust, validating early boot components. The TPM protector can only be used with the OS drive.
TPM + PIN	A user-entered numeric or alphanumeric key protector that can only be used with OS volumes and in addition to the TPM.The TPM validates early boot components. The user must enter the correct PIN before the start-up process can continue, and before the drive can be unlocked. The TPM enters lockout if the incorrect PIN is entered repeatedly, to protect the PIN from brute force attacks. The number of repeated attempts that trigger a lockout is variable.
TPM + Startup key	The TPM successfully validates early boot components. The user must insert a USB drive containing the startup key before the OS can boot.
TPM + Startup key + PIN	The TPM successfully validates early boot components. The user must enter the correct PIN and insert a USB drive containing the startup key before the OS can boot.
Recovery password	A 48-digit number used to unlock a volume when it is in *recovery mode*. Numbers can often be typed on a regular keyboard. If the numbers on the normal keyboard aren't responding, the function keys (F1-F10) can be used to input the numbers.
TPM + Network Key	The TPM successfully validates early boot components, and a valid encrypted network key has been provided from a WDS server. This authentication method provides automatic unlock of OS volumes while maintaining multifactor authentication. This key protector can only be used with OS volumes.
Recovery key	An encryption key stored on removable media that can be used for recovering data encrypted on a BitLocker volume. The file name has a format of `<protector_id>.bek`.
Data Recovery Agent	Data recovery agents (DRAs) are accounts that are able to decrypt BitLocker-protected drives by using their certificates. Recovery of a BitLocker-protected drive can be accomplished by a data recovery agent that is configured with the proper certificate.
Active Directory user or group	A protector that is based on an Active Directory user or group security identified (SID). Data drives are automatically unlocked when such users attempt to access them.

Support for devices without TPM

Determine whether computers that don't have a TPM 1.2 or higher versions in the environment will be supported. If you decide to support devices without TPM, a user must use a USB startup key or a password to boot the system. The startup key requires extra support processes similar to multifactor authentication.

What areas of the organization need a baseline level of data protection?

The TPM-only authentication method provides the most transparent user experience for organizations that need a baseline level of data protection to meet security policies. It has the lowest total cost of ownership. TPM-only might also be more appropriate for devices that are unattended or that must reboot unattended.

However, TPM-only authentication method doesn't offer a high level of data protection. This authentication method protects against attacks that modify early boot components. But, the level of protection can be affected by potential weaknesses in hardware or in the early boot components. BitLocker's multifactor authentication methods significantly increase the overall level of data protection.

> **Tip**
>
> An advantage of TPM-only authentication is that a device can boot Windows without any user interaction. In case of lost or stolen device, there may be an advantage of this configuration: if the device is connected to the Internet, it can be remotely wiped with a device management solution like Microsoft Intune.

What areas of the organization need a more secure level of data protection?

If there are devices with highly sensitive data, then deploy BitLocker with multifactor authentication on those systems. Requiring the user to input a PIN significantly increases the level of protection for the system. BitLocker Network Unlock can also be used to allow these devices to automatically unlock when connected to a trusted wired network that can provide the Network Unlock key.

What multifactor authentication method does the organization prefer?

The protection differences provided by multifactor authentication methods can't be easily quantified. Consider each authentication method's impact on Helpdesk support, user education, user productivity, and any automated systems management processes.

Manage passwords and PINs

When BitLocker is enabled on a system drive and the device has a TPM, users can be required to enter a PIN before BitLocker unlocks the drive. Such a PIN requirement can prevent an attacker who has physical access to a device from even getting to the Windows sign-in, which makes it almost impossible for the attacker to access or modify user data and system files.

Requiring a PIN at startup is a useful security feature because it acts as a second authentication factor. However, this configuration comes with some costs, especially if you require to change the PIN regularly.

In addition, Modern Standby devices don't require a PIN for startup: they're designed to start infrequently and have other mitigations in place that further reduce the attack surface of the system.

For more information about how startup security works and the countermeasures that Windows provides, see Preboot authentication.

TPM hardware configurations

In the deployment plan, identify what TPM-based hardware platforms are supported. Document the hardware models from an OEM(s) used by the organization so that their configurations can be tested and supported. TPM hardware requires special consideration during all aspects of planning and deployment.

TPM 1.2 states and initialization

For TPM 1.2, there are multiple possible states. Windows automatically initializes the TPM, which brings it to an enabled, activated, and owned state. This state is the state that BitLocker requires before it can use the TPM.

Endorsement keys

For a TPM to be usable by BitLocker, it must contain an endorsement key, which is an RSA key pair. The private half of the key pair is held inside the TPM and is never revealed or accessible outside the TPM. If the TPM doesn't have an endorsement key, BitLocker forces the TPM to generate one automatically as part of BitLocker setup.

An endorsement key can be created at various points in the TPM's lifecycle, but needs to be created only once for the lifetime of the TPM. If an endorsement key doesn't exist for the TPM, it must be created before you can take TPM ownership.

For more information about the TPM and the TCG, see the Trusted Computing Group: Trusted Platform Module (TPM) Specifications.

Non-TPM hardware configurations

Devices without a TPM can still be protected with drive encryption using a startup key.

Use the following questions to identify issues that might affect the deployment in a non-TPM configuration:

- Is there a budget for USB flash drives for each of these devices?
- Do existing non-TPM devices support USB drives at boot time?

Test the individual hardware platforms with the BitLocker system check option while enabling BitLocker. The system check makes sure that BitLocker can read the recovery information from a USB device and encryption keys correctly before it encrypts the volume.

Disk configuration considerations

To function correctly, BitLocker requires a specific disk configuration. BitLocker requires two partitions that meet the following requirements:

- The operating system partition contains the operating system and its support files; it must be formatted with the NTFS file system
- The system partition (or boot partition) includes the files needed to load Windows after the BIOS or UEFI firmware has prepared the system hardware. BitLocker isn't enabled on this partition. For BitLocker to work, the system partition must not be encrypted, and must be on a different partition than the operating system. On UEFI platforms, the system partition must be formatted with the FAT 32-file system. On BIOS platforms, the system partition must be formatted with the NTFS file system. It should be at least 350 MB in size.

Windows setup automatically configures the disk drives of computers to support BitLocker encryption.

Windows Recovery Environment (Windows RE) is an extensible recovery platform that is based on Windows Pre-installation Environment (Windows PE). When the computer fails to start, Windows automatically transitions into this environment, and the Startup Repair tool in Windows RE automates the diagnosis and repair of an unbootable Windows installation. Windows RE also contains the drivers and tools that are needed to unlock a volume protected by BitLocker by providing a recovery key or recovery password. To use Windows RE with BitLocker, the Windows RE boot image must be on a volume that isn't protected by BitLocker.

Windows RE can also be used from boot media other than the local hard disk. If Windows RE isn't installed on the local hard disk of BitLocker-enabled computers, then different methods can be used to boot Windows RE. For example, Windows Deployment Services (WDS) or USB flash drive can be used for recovery.

BitLocker provisioning

Administrators can enable BitLocker before to operating system deployment from the *Windows Pre-installation Environment* (WinPE). This step is done with a randomly generated clear key protector applied to the formatted volume. It encrypts the volume before running the Windows setup process. If the encryption uses the **Used Disk Space Only** option, then this step takes only a few seconds, and can be incorporated into existing deployment processes. Preprovisioning requires a TPM.

To check the BitLocker status of a particular volume, administrators can look at the drive status in the BitLocker Control Panel applet or Windows Explorer. The **Waiting For Activation** status means that the drive was preprovisioned for BitLocker, and there's only a clear protector used to encrypt the volume. In this case, the volume isn't protected, and needs to have a secure key added to the volume before the drive is considered fully protected. Administrators can use the Control Panel options, PowerShell cmdlets, the `manage-bde.exe` tool, or WMI APIs to add an appropriate key protector. The volume status then is updated.

When using the Control Panel options, administrators can choose to **Turn on BitLocker** and follow the steps in the wizard to add a protector, such as a PIN for an operating system volume (or a password if no TPM exists), or a password or smart card protector to a data volume. Then the drive security window is presented before changing the volume status.

Used Disk Space Only encryption

The BitLocker Setup wizard provides administrators the ability to choose the *Used Disk Space Only* or *Full* encryption method when enabling BitLocker for a volume. Administrators can use BitLocker policy settings to enforce either Used Disk Space Only or Full disk encryption.

Launching the BitLocker Setup wizard prompts for the authentication method to be used (password and smart card are available for data volumes). Once the method is chosen and the recovery key is saved, the wizard asks to choose the drive encryption type. Select **Used Disk Space Only** or **Full** drive encryption.

With Used Disk Space Only, just the portion of the drive that contains data are encrypted. Unused space remains unencrypted. This behavior causes the encryption process to be faster, especially for new devices and data drives. When BitLocker is enabled with this method, as data

is added to the drive, the portion of the drive used is encrypted. So, there's never unencrypted data stored on the drive.

With Full drive encryption, the entire drive is encrypted, whether data is stored on it or not. This option is useful for drives that have been repurposed, and might contain data remnants from their previous use.

> **⊗ Caution**
>
> Exercise caution when encrypting only used space on an existing volume on which confidential data might have already been stored in an unencrypted state. When using used space encryption, sectors where previously unencrypted data are stored can be recovered through disk-recovery tools until they're overwritten by new encrypted data. In contrast, encrypting only used space on a brand-new volume can significantly decrease deployment time without the security risk because all new data will be encrypted as it's written to the disk.

Encrypted hard drive support

Encrypted hard drives provide onboard cryptographic capabilities to encrypt data on drives. This feature improves both drive and system performance by offloading cryptographic calculations from the device's processor to the drive itself. Data is rapidly encrypted by the drive by using dedicated, purpose-built hardware. If planning to use whole-drive encryption with Windows, Microsoft recommends researching hard drive manufacturers and models to determine whether any of their encrypted hard drives meet the security and budget requirements.

For more information about encrypted hard drives, see Encrypted hard drives.

Microsoft Entra ID and Active Directory Domain Services considerations

BitLocker integrates with Microsoft Entra ID and Active Directory Domain Services (AD DS) to provide centralized key management. By default, no recovery information is backed up to Microsoft Entra ID or AD DS. Administrators can configure policy setting for each drive type to enable backup of BitLocker recovery information.

The following recovery data is saved for each computer object:

- *Recovery password*: a 48-digit recovery password used to recover a BitLocker-protected volume. Users must enter this password to unlock a volume when BitLocker enters recovery mode
- *Key package*: with the key package and the recovery password, portions of a BitLocker-protected volume can be decrypted if the disk is severely damaged. Each key package works only with the volume it was created on, which is identified by the corresponding volume ID

FIPS support for recovery password protector

Devices configured to operate in FIPS mode can create FIPS-compliant recovery password protectors, which use the *FIPS-140 NIST SP800-132* algorithm.

> **Note**
>
> The United States Federal Information Processing Standard (FIPS) defines security and interoperability requirements for computer systems that are used by the U.S. Federal Government. The FIPS-140 standard defines approved cryptographic algorithms. The FIPS-140 standard also sets forth requirements for key generation and for key management. The National Institute of Standards and Technology (NIST) uses the Cryptographic Module Validation Program (CMVP) to determine whether a particular implementation of a cryptographic algorithm is compliant with the FIPS-140 standard. An implementation of a cryptographic algorithm is considered FIPS-140-compliant only if it has been submitted for and has passed NIST validation. An algorithm that has not been submitted cannot be considered FIPS-compliant even if the implementation produces identical data as a validated implementation of the same algorithm.

- FIPS-compliant recovery password protectors can be exported and stored in AD DS
- The BitLocker policy settings for recovery passwords work the same for all Windows versions that support BitLocker, whether in FIPS mode or not

Network Unlock

Some organizations have location-specific data security requirements, especially in environments with high-value data. The network environment might provide crucial data protection and enforce mandatory authentication. Therefore, policy states that those devices shouldn't leave the building or be disconnected from the corporate network. Safeguards like physical security locks and geofencing might help enforce this policy as reactive controls. Beyond these safeguards, a proactive security control that grants data access only when the device is connected to the corporate network is necessary.

Network Unlock enables BitLocker-protected devices to start automatically when connected to a wired corporate network on which Windows Deployment Services runs. Anytime the device isn't connected to the corporate network, a user must enter a PIN to unlock the drive (if PIN-based unlock is enabled). Network Unlock requires the following infrastructure:

- Client devices that have Unified Extensible Firmware Interface (UEFI) firmware version 2.3.1 or later, which supports Dynamic Host Configuration Protocol (DHCP)
- A Windows Server running the Windows deployment services (WDS) role
- A DHCP server

For more information about how to configure Network unlock feature, see Network Unlock.

BitLocker recovery

Organizations should carefully plan a BitLocker recovery strategy as part of the overall BitLocker implementation plan. There are different options when implementing a BitLocker recovery model, which are described in BitLocker recovery overview.

Monitor BitLocker

Organizations can use Microsoft Intune or Configuration Manager to monitor device encryption across multiple devices. For more information, see Monitor device encryption with Intune and View BitLocker reports in Configuration Manager.

Next steps

Learn how to plan a BitLocker recovery strategy for your organization:

BitLocker recovery overview >

Learn about the available options to configure BitLocker and how to configure them via Configuration Service Providers (CSP) or group policy (GPO):

Configure BitLocker >

Last updated on 07/29/2025

Configure BitLocker

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

To configure BitLocker, you can use one of the following options:

- Configuration Service Provider (CSP): this option is commonly used for devices managed by a Mobile Device Management (MDM) solution, like Microsoft Intune. The BitLocker CSP is used to configure BitLocker, and to report the status of different BitLocker functions to the MDM solution. With Microsoft Intune, you can use the BitLocker status in compliance policies, combining them with Conditional Access. Conditional Access can prevent or grant access to services like Exchange Online and SharePoint Online, based on the status of BitLocker. To learn more about the Intune options to configure and monitor BitLocker, check the following articles:
  - Manage BitLocker policy for Windows devices with Intune
  - Monitor device encryption with Intune
  - Use compliance policies to set rules for devices you manage with Intune
- Group policy (GPO): this option can be used for devices that are joined to an Active Directory domain and aren't managed by a device management solution. Group policy can also be used for devices that aren't joined to an Active Directory domain, using the local group policy editor
- Microsoft Configuration Manager: this option can be used for devices that are managed by Microsoft Configuration Manager using the BitLocker management agent. To learn more about options to configure BitLocker via Microsoft Configuration Manager, see Deploy BitLocker management

> **Note**
>
> Windows Server doesn't support the configuration of BitLocker using CSP or Microsoft Configuration Manager. Use GPO instead.

While many of the BitLocker policy settings can be configured using both CSP and GPO, there are some settings that are only available using one of the options. To learn about the policy settings available for both CSP and GPO, review the section BitLocker policy settings.

Windows edition and licensing requirements

The following table lists the Windows editions that support BitLocker management:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

BitLocker management license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
No	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

BitLocker policy settings

This section describes the policy settings to configure BitLocker via configuration service provider (CSP) and group policy (GPO).

> **Important**
>
> Most of the BitLocker policy settings are enforced when BitLocker is initially turned on for a drive. Encryption isn't restarted if settings change.

Policy settings list

The list of settings is sorted alphabetically and organized in four categories:

- **Common settings**: settings applicable to all BitLocker-protected drives
- **Operating system drive**: settings applicable to the drive where Windows is installed
- **Fixed data drives**: settings applicable to any local drives, except the operating system drive
- **Removable data drives**: settings applicable to any removable drives

Select one of the tabs to see the list of available settings:

Common settings

The following table lists the BitLocker policies applicable to all drive types, indicating if they're applicable via configuration service provider (CSP) and/or group policy (GPO). Select the policy name for more details.

Expand table

Policy name	CSP	GPO
Allow standard user encryption	✅	❌
Choose default folder for recovery password	❌	✅
Choose drive encryption method and cipher strength	✅	✅
Configure recovery password rotation	✅	❌
Disable new DMA devices when this computer is locked	❌	✅
Prevent memory overwrite on restart	❌	✅
Provide the unique identifiers for your organization	✅	✅
Require device encryption	✅	❌
Validate smart card certificate usage rule compliance	❌	✅

Allow standard user encryption

With this policy you can enforce the *Require device encryption* policy for scenarios where the policy is applied while the current logged-on user doesn't have administrative rights.

> **Important**
>
> The **Allow warning for other disk encryption** policy must be disabled to allow standard user encryption.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/BitLocker/`AllowStandardUserEncryption
GPO	Not available

Choose default folder for recovery password

Specify the default path that is displayed when the *BitLocker Drive Encryption setup wizard* prompts the user to enter the location of a folder in which to save the recovery password. You can specify either a fully qualified path or include the target computer's environment variables in the path:

- If the path isn't valid, the BitLocker setup wizard displays the computer's top-level folder view
- If you disable or don't configure this policy setting, the BitLocker setup wizard displays the computer's top-level folder view when the user chooses the option to save the recovery password in a folder

> **Note**
>
> This policy setting does not prevent the user from saving the recovery password in another folder.

Expand table

	Path
CSP	Not available
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption**

Choose drive encryption method and cipher strength

With this policy, you can configure an encryption algorithm and key cipher strength for fixed data drives, operating system drives, and removable data drives individually.

Recommended settings: `XTS-AES` algorithm for all drives. The choice of key size, 128 bit or 256 bit depends on the performance of the device. For more performant hard drives and CPU, choose 256-bit key, for less performant ones use 128.

> **Important**
>
> Key size might be required by regulators or industry.

If you disable or don't configure this policy setting, BitLocker uses the default encryption method of `XTS-AES 128-bit`.

> **Note**
>
> This policy doesn't apply to encrypted drives. Encrypted drives utilize their own algorithm, which is set by the drive during partitioning.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/BitLocker/`EncryptionMethodByDriveType
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption**

Configure recovery password rotation

With this policy you can configure a numeric recovery password rotation upon use for OS and fixed drives on Microsoft Entra joined and Microsoft Entra hybrid joined devices.

Possible values are:

- `0`: numeric recovery password rotation is turned off
- `1`: numeric recovery password rotation upon use is *on* for Microsoft Entra joined devices. This is also the default value
- `2`: numeric recovery password rotation upon use is *on* for both Microsoft Entra joined devices and Microsoft Entra hybrid joined devices

> **Note**
>
> The policy is effective only when Microsoft Entra ID or Active Directory backup for recovery password is configured to *required*
>
> - For OS drive: enable *Do not enable BitLocker until recovery information is stored to AD DS for operating system drives*
> - For fixed drives: enable "*Do not enable BitLocker until recovery information is stored to AD DS for fixed data drives*

Expand table

	Path
CSP	`./Device/Vendor/MSFT/BitLocker/`ConfigureRecoveryPasswordRotation
GPO	Not available

Disable new DMA devices when this computer is locked

When enabled, this policy setting blocks direct memory access (DMA) for all hot pluggable PCI ports until a user signs into Windows.

Once a user signs in, Windows enumerates the PCI devices connected to the host Thunderbolt PCI ports. Every time the user locks the device, DMA is blocked on hot plug Thunderbolt PCI ports with no children devices, until the user signs in again.

Devices that were already enumerated when the device was unlocked will continue to function until unplugged, or the system is rebooted or hibernated.

This policy setting is only enforced when BitLocker or device encryption is enabled.

> **ⓘ Important**
>
> This policy is not compatible with *Kernel DMA Protection*. It's recommended to disable this policy if the system supports Kernel DMA Protection, as Kernel DMA Protection provides higher security for the system. For more information about Kernel DMA Protection, see Kernel DMA Protection.

Expand table

	Path
CSP	Not available
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption**

Prevent memory overwrite on restart

This policy setting is used to control whether the computer's memory is overwritten when the device restarts. BitLocker secrets include key material used to encrypt data.

- If you enable this policy setting, memory isn't overwritten when the computer restarts. Preventing memory overwrite may improve restart performance but

increases the risk of exposing BitLocker secrets.
- If you disable or do not configure this policy setting, BitLocker secrets are removed from memory when the computer restarts.

> **Note**
>
> This policy setting applies only when BitLocker protection is enabled.

Expand table

	Path
CSP	Not available
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption**

Provide the unique identifiers for your organization

This policy setting allows you to associate unique organizational identifiers to a drive that is encrypted with BitLocker. The identifiers are stored as the *identification field* and *allowed identification field*:

- The identification field allows you to associate a unique organizational identifier to BitLocker-protected drives. This identifier is automatically added to new BitLocker-protected drives and can be updated on existing BitLocker-protected drives using the *BitLocker Drive Encryption: Configuration Tool* (`manage-bde.exe`)
- The allowed identification field is used in combination with the *Deny write access to removable drives not protected by BitLocker* policy setting to help control the use of removable drives in your organization. It's a comma separated list of identification fields from your organization or other external organizations. You can configure the identification fields on existing drives by using `manage-bde.exe`.

If you enable this policy setting, you can configure the identification field on the BitLocker-protected drive and any allowed identification field used by your organization. When a BitLocker-protected drive is mounted on another BitLocker-enabled device, the identification field and allowed identification field are used to determine whether the drive is from a different organization.

If you disable or don't configure this policy setting, the identification field is not required.

> **Important**

Identification fields are required for management of certificate-based data recovery agents on BitLocker-protected drives. BitLocker only manages and updates certificate-based data recovery agents when the identification field is present on a drive and is identical to the value configured on the device. The identification field can be any value of 260 characters or fewer.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/BitLocker/`IdentificationField
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption**

Require device encryption

This policy setting determines whether BitLocker is required:

- If enabled, encryption is triggered on all drives silently or non-silently based on Allow warning for other disk encryption policy
- If disabled, BitLocker isn't turned off for the system drive, but it stops prompting the user to turn BitLocker on.

> ⓘ Note
>
> Typically, BitLocker follows the Choose drive encryption method and cipher strength policy configuration. However, this policy setting will be ignored for self-encrypting fixed drives and self-encrypting OS drives.

Encryptable fixed data volumes are treated similarly to OS volumes, but they must meet other criteria to be encryptable:

- It must not be a dynamic volume
- It must not be a recovery partition
- It must not be a hidden volume
- It must not be a system partition
- It must not be backed by virtual storage
- It must not have a reference in the BCD store

> ⓘ Note

Only full disk encryption is supported when using this policy for silent encryption. For non-silent encryption, encryption type will depend on the ***Enforce drive encryption type on operating system drives*** and ***Enforce drive encryption type on fixed data drives*** policies configured on the device.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/BitLocker/`RequireDeviceEncryption
GPO	Not available

Validate smart card certificate usage rule compliance

This policy setting is used to determine which certificate to use with BitLocker by associating an object identifier (OID) from a smart card certificate to a BitLocker-protected drive. The object identifier is specified in the enhanced key usage (EKU) of a certificate.

BitLocker can identify which certificates may be used to authenticate a user certificate to a BitLocker-protected drive by matching the object identifier in the certificate with the object identifier that is defined by this policy setting. Default OID is `1.3.6.1.4.1.311.67.1.1`.

If you enable this policy setting, the object identifier specified in the **Object identifier** field must match the object identifier in the smart card certificate. If you disable or don't configure this policy setting, the default OID is used.

> **Note**
>
> BitLocker doesn't require that a certificate have an EKU attribute; however, if one is configured for the certificate, it must be set to an object identifier that matches the object identifier configured for BitLocker.

Expand table

	Path
CSP	Not available

	Path
GPO	**Computer Configuration > Administrative Templates > Windows Components > BitLocker Drive Encryption**

BitLocker and policy settings compliance

If a device isn't compliant with the configured policy settings, BitLocker might not be turned on, or BitLocker configuration might be modified until the device is in a compliant state. When a drive becomes out of compliance with policy settings, only changes to the BitLocker configuration that will bring it into compliance are allowed. Such scenario could occur, for example, if a previously encrypted drive becomes noncompliant by a policy setting change.

If multiple changes are necessary to bring the drive into compliance, BitLocker protection might need to be suspended, the necessary changes made, and then protection resumed. Such situation could occur, for example, if a removable drive is initially configured for unlock with a password, and then policy settings are changed to require smart cards. In this scenario, BitLocker protection needs to be suspended, delete the password unlock method, and add the smart card method. After this process is complete, BitLocker is compliant with the policy setting, and BitLocker protection on the drive can be resumed.

In other scenarios, to bring the drive into compliance with a change in policy settings, BitLocker might need to be disabled and the drive decrypted followed by re-enabling BitLocker and then re-encrypting the drive. An example of this scenario is when the BitLocker encryption method or cipher strength is changed.

To learn more how to manage BitLocker, review the BitLocker operations guide.

Configure and manage servers

Servers are often deployed, configured, and managed using PowerShell. The recommendation is to use group policy settings to configure BitLocker on servers, and to manage BitLocker using PowerShell.

BitLocker is an optional component in Windows Server. Follow the directions in Install BitLocker on Windows Server to add the BitLocker optional component.

The Minimal Server Interface is a prerequisite for some of the BitLocker administration tools. On a Server Core installation, the necessary GUI components must be added first. The steps to add shell components to Server Core are described in Using Features on Demand with Updated Systems and Patched Images and How to update local source media to add roles and

features. If a server is installed manually, then choosing Server with Desktop Experience is the easiest path because it avoids performing the steps to add a GUI to Server Core.

Lights-out data centers can take advantage of the enhanced security of a second factor while avoiding the need for user intervention during reboots by optionally using a combination of BitLocker (TPM+PIN) and BitLocker Network Unlock. BitLocker Network Unlock brings together the best of hardware protection, location dependence, and automatic unlock, while in the trusted location. For the configuration steps, see Network Unlock.

Next steps

Review the BitLocker operations guide to learn how to use different tools to manage and operate BitLocker.

BitLocker operations guide >

Last updated on 07/29/2025

BitLocker operations guide

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

There are different tools and options to manage and operate BitLocker:

- the BitLocker PowerShell module
- the BitLocker drive encryption tools
- Control Panel

The BitLocker drive encryption tools and BitLocker PowerShell module can be used to perform any tasks that can be accomplished through the BitLocker Control Panel. They're appropriate to use for automated deployments and other scripting scenarios.
The BitLocker Control Panel applet allows users to perform basic tasks such as turning on BitLocker on a drive and specifying unlock methods and authentication methods. The BitLocker Control Panel applet is appropriate to use for basic BitLocker tasks.

This article describes the BitLocker management tools and how to use them, providing practical examples.

BitLocker PowerShell module

The BitLocker PowerShell module enables administrators to integrate BitLocker options into existing scripts with ease. For a list of cmdlets included in module, their description and syntax, check the BitLocker PowerShell reference article.

BitLocker drive encryption tools

The BitLocker drive encryption tools include the two command-line tools:

- *Configuration Tool* (`manage-bde.exe`) can be used for scripting BitLocker operations, offering options that aren't present in the BitLocker Control Panel applet. For a complete list of the `manage-bde.exe` options, see the Manage-bde reference
- *Repair Tool* (`repair-bde.exe`) is useful for disaster recovery scenarios, in which a BitLocker protected drive can't be unlocked normally or using the recovery console

BitLocker Control Panel applet

Encrypting volumes with the BitLocker Control Panel (select **Start**, enter `BitLocker`, select **Manage BitLocker**) is how many users will use BitLocker. The name of the BitLocker Control

Panel applet is *BitLocker Drive Encryption*. The applet supports encrypting operating system, fixed data, and removable data volumes. The BitLocker Control Panel organizes available drives in the appropriate category based on how the device reports itself to Windows. Only formatted volumes with assigned drive letters appear properly in the BitLocker Control Panel applet.

Important: The Shell Hardware Detection Service (ShellHWDetection) is required to manage BitLocker via Control Panel or via Explorer.

Use BitLocker within Windows Explorer

Windows Explorer allows users to launch the **BitLocker Drive Encryption Wizard** by right-clicking a volume and selecting **Turn On BitLocker**. This option is available on client computers by default. On servers, the BitLocker feature and the Desktop-Experience feature must first be installed for this option to be available. After selecting **Turn on BitLocker**, the wizard works exactly as it does when launched using the BitLocker Control Panel.

Check the BitLocker status

To check the BitLocker status of a particular volume, administrators can look at the status of the drive in the BitLocker Control Panel applet, Windows Explorer, `manage-bde.exe` command-line tool, or Windows PowerShell cmdlets. Each option offers different levels of detail and ease of use.

Follow the instructions below verify the status of BitLocker, selecting the tool of your choice.

PowerShell

To determine the current state of a volume you can use the `Get-BitLockerVolume` cmdlet, which provides information on the volume type, protectors, protection status, and other details. For example:

PowerShell

```
PS C:\> Get-BitLockerVolume C: | fl

ComputerName          : DESKTOP
MountPoint            : C:
EncryptionMethod      : XtsAes128
AutoUnlockEnabled     :
AutoUnlockKeyStored   : False
MetadataVersion       : 2
VolumeStatus          : FullyEncrypted
ProtectionStatus      : On
LockStatus            : Unlocked
```

```
EncryptionPercentage : 100
WipePercentage       : 0
VolumeType           : OperatingSystem
CapacityGB           : 1000
KeyProtector         : {Tpm, RecoveryPassword}
```

Enable BitLocker

OS drive with TPM protector

The following example shows how to enable BitLocker on an operating system drive using only the TPM protector and no recovery key:

PowerShell

PowerShell

```
Enable-BitLocker C: -TpmProtector
```

OS drive with TPM protector and startup key

The following example shows how to enable BitLocker on an operating system drive using the TPM and *startup key* protectors.

Assuming the OS drive letter is `C:` and the USB flash drive is drive letter `E:`, here's the command:

PowerShell

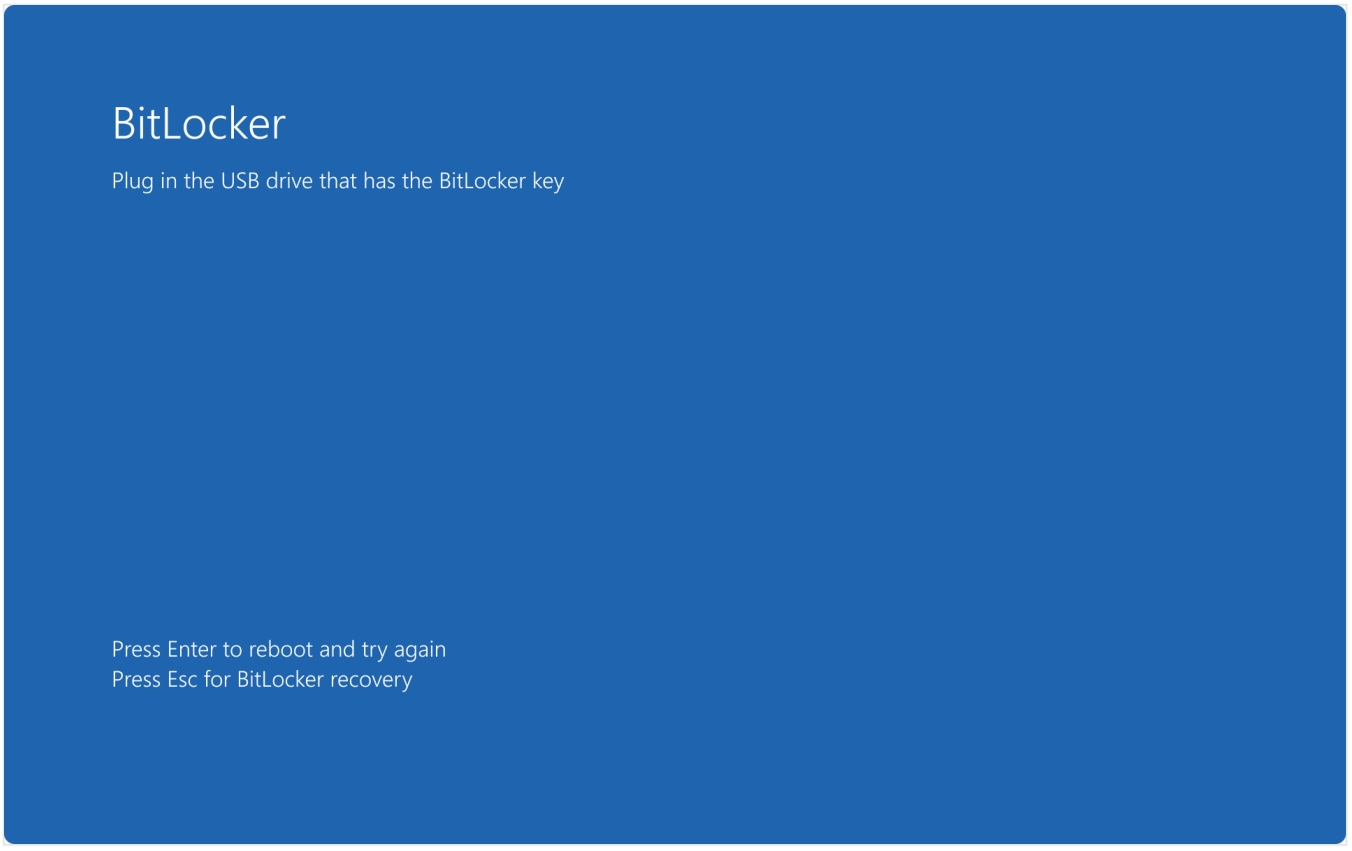
PowerShell

If you choose to skip the BitLocker hardware test, encryption starts immediately without the need for a reboot.

PowerShell

```
Enable-BitLocker C: -StartupKeyProtector -StartupKeyPath E: -SkipHardwareTest
```

After reboot, the BitLocker preboot screen displays and the USB startup key must be inserted before the operating system can be started:



Data volumes

Data volumes use a similar process for encryption as operating system volumes, but they don't require protectors for the operation to complete.

PowerShell

Add the desired protectors prior to encrypting the volume. The following example adds a password protector to the `E:` volume using the variable `$pw` as the password. The `$pw` variable is held as a SecureString value to store the user-defined password:

PowerShell

```
$pw = Read-Host -AsSecureString
<user inputs password>
Add-BitLockerKeyProtector E: -PasswordProtector -Password $pw
```

> **Note**
>
> The BitLocker cmdlet requires the key protector GUID enclosed in quotation marks to execute. Ensure the entire GUID, with braces, is included in the command.

Example: Use PowerShell to enable BitLocker with a TPM protector

PowerShell

```
Enable-BitLocker D: -EncryptionMethod XtsAes256 -UsedSpaceOnly -TpmProtector
```

Example: Use PowerShell to enable BitLocker with a TPM+PIN protector, in this case with a PIN set to *123456*:

PowerShell

```
$SecureString = ConvertTo-SecureString "123456" -AsPlainText -Force
Enable-BitLocker C: -EncryptionMethod XtsAes256 -UsedSpaceOnly -Pin
$SecureString -TPMandPinProtector
```

Manage BitLocker protectors

The management of BitLocker protectors consists in adding, removing, and backing up protectors.

Managed BitLocker protectors by using the following instructions, selecting the option that best suits your needs.

List protectors

The list of protectors available for a volume (`C:` in the example) can be listed by running the following command:

PowerShell

PowerShell

```
(Get-BitLockerVolume -mountpoint C).KeyProtector
```

Add protectors

Add a recovery password protector

PowerShell

PowerShell

```
Add-BitLockerKeyProtector -MountPoint C -RecoveryPasswordProtector
```

Add a password protector

A common protector for a *data volume* is the *password protector*. In the next example, a password protector is added to a volume.

PowerShell

PowerShell

```
Add-BitLockerKeyProtector -MountPoint D -PasswordProtector
```

Add an Active Directory protector

The Active Directory protector is a SID-based protector that can be added to both operating system and data volumes, although it doesn't unlock operating system volumes in the preboot environment. The protector requires the SID for the domain account or group to link with the protector. BitLocker can protect a cluster-aware disk by adding an SID-based protector for the Cluster Name Object (CNO) that lets the disk properly failover and unlock to any member computer of the cluster.

> **Important**
>
> The SID-based protector requires the use of an additional protector such as TPM, PIN, recovery key, etc. when used on operating system volumes.

> **Note**
>
> This option is not available for Microsoft Entra joined devices.

In this example, a domain SID-based protector is added to a previously encrypted volume. The user knows the SID for the user account or group they wish to add and uses the following command:

PowerShell

PowerShell

```
Add-BitLockerKeyProtector C: -ADAccountOrGroupProtector -ADAccountOrGroup "
<SID>"
```

To add the protector to a volume, either the domain SID or the group name preceded by the domain and a backslash are needed. In the following example, the **CONTOSO\Administrator** account is added as a protector to the data volume G.

PowerShell

```
Enable-BitLocker G: -AdAccountOrGroupProtector -AdAccountOrGroup
CONTOSO\Administrator
```

To use the SID for the account or group, the first step is to determine the SID associated with the security principal. To get the specific SID for a user account in Windows PowerShell, use the following command:

PowerShell

```
Get-ADUser -filter {samaccountname -eq "administrator"}
```

> **Note**
>
> Use of this command requires the RSAT-AD-PowerShell feature.

> **Tip**
>
> Information about the locally logged on user and group membership can be found using: `whoami.exe /all`.

Remove protectors

PowerShell

To remove existing protectors on a volume, use the `Remove-BitLockerKeyProtector` cmdlet. A GUID associated with the protector to be removed must be provided.

The following commands return the list of key protectors and GUIDS:

PowerShell

```
$vol = Get-BitLockerVolume C
$keyprotectors = $vol.KeyProtector
$keyprotectors
```

By using this information, the key protector for a specific volume can be removed using the command:

PowerShell

```
Remove-BitLockerKeyProtector <volume> -KeyProtectorID "{GUID}"
```

> **Note**
>
> The BitLocker cmdlet requires the key protector GUID enclosed in quotation marks to execute. Ensure the entire GUID, with braces, is included in the command.

> **Note**
>
> You must have at least one unlock method for any BitLocker-encrypted drives.

Suspend and resume

Some configuration changes may require to suspend BitLocker and then resume it after the change is applied.

Suspend and resume BitLocker by using the following instructions, selecting the option that best suits your needs.

Suspend BitLocker

PowerShell

PowerShell

```
Suspend-BitLocker -MountPoint D
```

Resume BitLocker

> ⓘ Note
>
> Resuming protection only works on devices that have accepted the Windows EULA.

PowerShell

PowerShell

```
Resume-BitLocker -MountPoint D
```

Reset and backup a recovery password

It's recommended to invalidate a recovery password after its use. In this example the recovery password protector is removed from the OS drive, a new protector added, and backed up to Microsoft Entra ID or Active Directory.

PowerShell

Remove all recovery passwords from the OS volume:

PowerShell

```
(Get-BitLockerVolume -MountPoint $env:SystemDrive).KeyProtector | `
  where-object {$_.KeyProtectorType -eq 'RecoveryPassword'} | `
  Remove-BitLockerKeyProtector -MountPoint $env:SystemDrive
```

Add a BitLocker recovery password protector for the OS volume:

PowerShell

```
Add-BitLockerKeyProtector -MountPoint $env:SystemDrive -
RecoveryPasswordProtector
```

Obtain the ID of the new recovery password:

PowerShell

```
(Get-BitLockerVolume -mountpoint $env:SystemDrive).KeyProtector | where-object
{$_.KeyProtectorType -eq 'RecoveryPassword'} | ft
```

```
KeyProtectorId,RecoveryPassword
```

> **Note**
>
> This next steps are not required if the policy setting Choose how BitLocker-protected operating system drives can be recovered is configured to **Require BitLocker backup to AD DS**.

Copy the ID of the recovery password from the output.

Using the GUID from the previous step, replace the `{ID}` in the following command and use the following command to back up the recovery password to Microsoft Entra ID:

PowerShell

```
BackuptoAAD-BitLockerKeyProtector -MountPoint $env:SystemDrive -KeyProtectorId "{ID}"
```

Or use the following command to back up the recovery password to Active Directory:

PowerShell

```
Backup-BitLockerKeyProtector -MountPoint $env:SystemDrive -KeyProtectorId "{ID}"
```

> **Note**
>
> The braces `{}` must be included in the ID string.

Disable BitLocker

Disabling BitLocker decrypts and removes any associated protectors from the volumes. Decryption should occur when protection is no longer required, and not as a troubleshooting step.

Disable BitLocker by using the following instructions, selecting the option that best suits your needs.

PowerShell

Windows PowerShell offers the ability to decrypt multiple drives in one pass. In the following example, the user has three encrypted volumes, which they wish to decrypt.

Using the Disable-BitLocker command, they can remove all protectors and encryption at the same time without the need for more commands. An example of this command is:

PowerShell

```
Disable-BitLocker
```

To avoid specifying each mount point individually, use the `-MountPoint` parameter in an array to sequence the same command into one line, without requiring additional user input. Example:

PowerShell

```
Disable-BitLocker -MountPoint C,D
```

Unlock a drive

If you connect a drive as a secondary drive to a device, and you have your BitLocker recovery key, you can unlock a BitLocker-enabled drive by using the following instructions.

In the next example, the `D` drive is the one to unlock. Select the option that best suits your needs.

PowerShell

PowerShell

```
Unlock-BitLocker -MountPoint D -RecoveryPassword xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx-xxxxxx
```

For more information, see Unlock-BitLocker

Last updated on 07/29/2025

BitLocker recovery overview

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

BitLocker recovery is the process by which access to a BitLocker-protected drive can be restored if the drive doesn't unlock using its default unlock mechanism.

This article describes scenarios that trigger BitLocker recovery, how to configure devices to save recovery information, and the options to restore access to a locked drive.

BitLocker recovery scenarios

The following list provides examples of common events that cause a device to enter BitLocker recovery mode when starting Windows:

- Entering the wrong PIN too many times
- Turning off the support for reading the USB device in the preboot environment from the BIOS or UEFI firmware if using USB-based keys instead of a TPM
- Having the CD or DVD drive before the hard drive in the BIOS boot order (common with virtual machines)
- upgrading Windows from a CD/DVD or a mounted ISO
- Docking or undocking a portable computer
- Changes to the NTFS partition table on the disk
- Changes to the boot manager
- Using PXE boot
- Turning off, disabling, deactivating, or clearing the TPM
- TPM self-test failure
- Upgrading the motherboard to a new one with a new TPM
- Upgrading critical early startup components, such as a BIOS or UEFI firmware upgrade
- Hiding the TPM from the operating system
- Modifying the Platform Configuration Registers (PCRs) used by the TPM validation profile
- Moving a BitLocker-protected drive into a new computer

- On devices with TPM 1.2, changing the BIOS or firmware boot device order
- Exceeding the maximum allowed number of failed sign-in attempts

> **Note**
>
> To take advantage of this functionality, you must configure the policy setting **Interactive logon: Machine account lockout threshold** located in **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**. Alternatively, use the Exchange ActiveSync **MaxFailedPasswordAttempts** policy setting, or the DeviceLock Configuration Service Provider (CSP).

As part of the BitLocker recovery process, it's recommended to determine what caused a device to enter in recovery mode. Root cause analysis might help to prevent the problem from occurring again in the future. For instance, if you determine that an attacker modified a device by obtaining physical access, you can implement new security policies for tracking who has physical presence.

For planned scenarios, such as a known hardware or firmware upgrades, initiating recovery can be avoided by temporarily suspending BitLocker protection. Suspending BitLocker leaves the drive fully encrypted, and the administrator can quickly resume BitLocker protection after the planned task is completed. Using *suspend* and *resume* also reseals the encryption key without requiring the entry of the recovery key.

> **Note**
>
> If suspended, BitLocker automatically resumes protection when the device is rebooted, unless a reboot count is specified using PowerShell or the `manage-bde.exe` command line tool. For more information about suspending BitLocker, review the BitLocker operations guide.

> **Tip**
>
> Recovery is described within the context of unplanned or undesired behavior. However, recovery can also be caused as an intended production scenario, for example in order to manage access control. When devices are redeployed to other departments or employees in the organization, BitLocker can be forced into recovery before the device is delivered to a new user.

Windows RE and BitLocker recovery

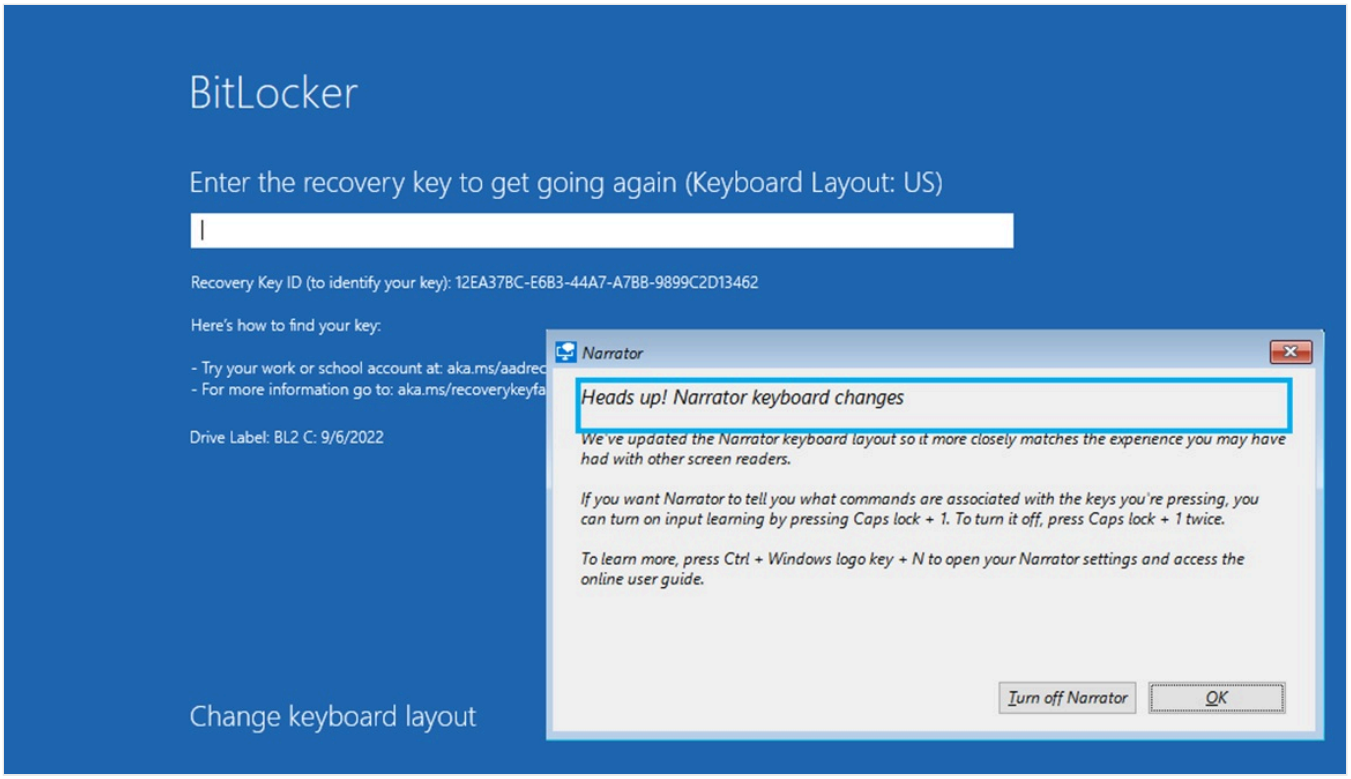
Windows Recovery Environment (Windows RE) can be used to recover access to a drive protected by BitLocker. If a device is unable to boot after two failures, *Startup Repair* starts automatically.

When Startup Repair is launched automatically due to boot failures, it only executes operating system and driver file repairs, provided that the boot logs or any available crash dump point to a specific corrupted file. On devices that support specific TPM measurements for PCR[7], the TPM validates that Windows RE is a trusted operating environment and unlocks any BitLocker-protected drives if Windows RE has not been modified. If the Windows RE environment has been modified, for example the TPM is disabled, the drives stay locked until the BitLocker recovery key is provided. If Startup Repair can't run automatically, and instead Windows RE is manually started from a repair disk, then the BitLocker recovery key must be provided to unlock the BitLocker-protected drives.

Windows RE will also ask for your BitLocker recovery key when you start a *Remove everything* reset from Windows RE on a device that uses the **TPM + PIN** or **Password for OS drive** protector. If you start BitLocker recovery on a keyboardless device with TPM-only protection, Windows RE, not the boot manager, will ask for the BitLocker recovery key. After you enter the key, you can access Windows RE troubleshooting tools or start Windows normally.

The BitLocker recovery screen that's shown by Windows RE has the accessibility tools like narrator and on-screen keyboard to help you enter your BitLocker recovery key:

- To activate the narrator during BitLocker recovery in Windows RE, press `WIN` + `CTRL` + `Enter`
- To activate the on-screen keyboard, tap on a text input control

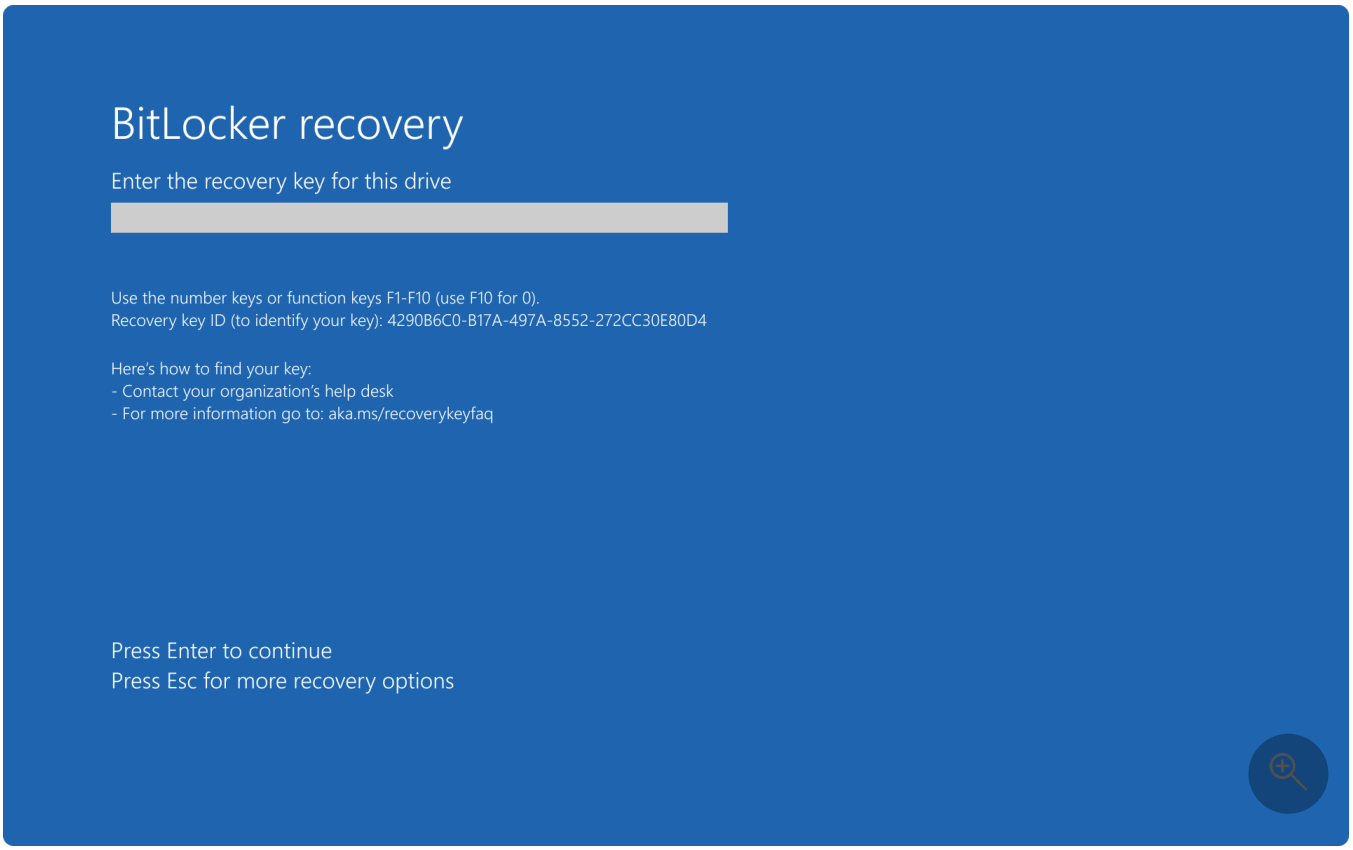


If the BitLocker recovery key is requested by the Windows boot manager, those tools might not be available.

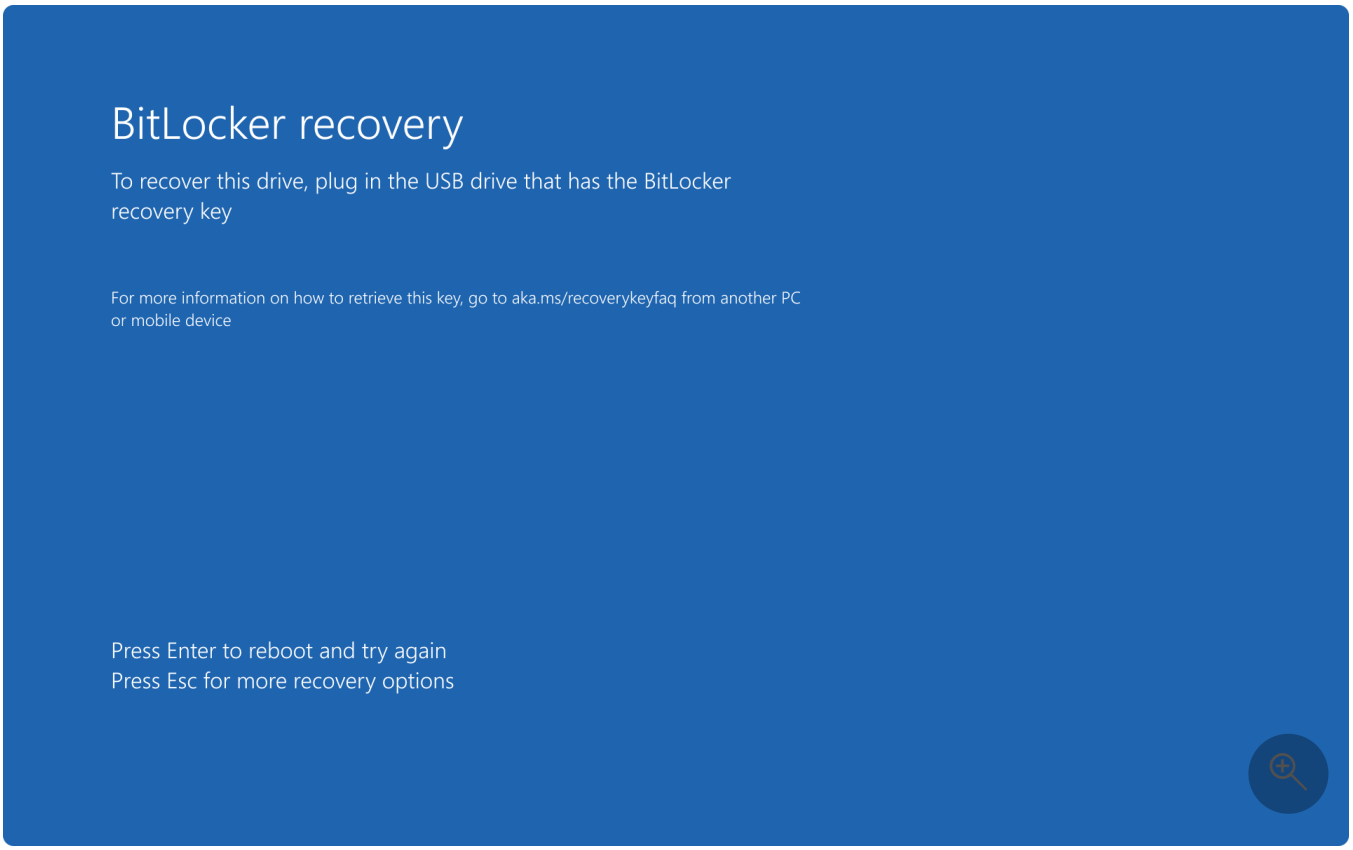
BitLocker recovery options

In a recovery scenario, the following options to restore access to the drive might be available, depending on the policy settings applied to the devices:

- **Recovery password**: a 48-digit number used to unlock a volume when it is in recovery mode. The recovery password might be saved as a text file, printed or stored in Microsoft Entra ID or Active Directory. The user can supply a recovery password, if available



- **Recovery key**: an encryption key stored on removable media that can be used for recovering data encrypted on a BitLocker volume. The file name has a format of `<protector_id>.bek`. For the OS drive, the recovery key can be used to gain access to the device if BitLocker detects a condition that prevents it from unlocking the drive when the device is starting up. A recovery key can also be used to gain access to fixed data drives and removable drives that are encrypted with BitLocker, if for some reason the password is forgotten or the device can't access the drive



- **Key package**: decryption key that can be used with the BitLocker Repair tool to reconstruct critical parts of a drive and salvage recoverable data. With the key package and either the *recovery password* or *recovery key*, portions of a corrupted BitLocker-protected drive can be decrypted. Each key package works only for a drive that has the corresponding drive identifier. A key package isn't generated automatically, and can be saved on a file or in Active Directory Domain Services. A key package can't be stored in Microsoft Entra ID
- **Data Recovery Agent certificate**: a Data Recovery Agent (DRA) is a type of certificate that is associated with an Active Directory security principal and that can be used to access any BitLocker encrypted drives configured with the matching public key. DRAs can use their credentials to unlock the drive. If the drive is an OS drive, the drive must be mounted as a data drive on another device for the DRA to unlock it

> **Tip**
>
> Both the *Recovery password* and *Recovery key* can be supplied by users in the Control Panel applet (for data and removable drives), or in the preboot recovery screen. It's recommended to configure policy settings to customize the preboot recovery screen, for example by adding a custom message, URL, and help desk contact information. For more information, review the article **BitLocker preboot recovery screen**.

When planning the BitLocker recovery process, first consult the organization's current best practices for recovering sensitive information. For example:

Expand table

Question
How does the organization handle lost or forgotten passwords?
How does the organization perform smart card PIN resets?
Are users allowed to save or retrieve recovery information for the devices that they own?
How much do you want users to be involved in the BitLocker configuration process? Do you want users to interact with the process, be silent, or both?
Where do you want to store the BitLocker recovery keys?
Do you want to enable recovery password rotation?

Answering the questions helps to determine the best BitLocker recovery process for the organization, and to configure BitLocker policy settings accordingly. For example, if the organization has a process for resetting passwords, a similar process can be used for BitLocker recovery. If users aren't allowed to save or retrieve recovery information, the organization can use a data recovery agents (DRAs), or automatically back up recovery information.

The following policy settings define the recovery methods that can be used to restore access to a BitLocker-protected drive:

- Choose how BitLocker-protected operating system drives can be recovered
- Choose how BitLocker-protected fixed drives can be recovered
- Choose how BitLocker-protected removable drives can be recovered

> **Tip**
>
> In each of these policies, select **Save BitLocker recovery information to Active Directory Domain Services** and then choose which BitLocker recovery information to store in AD DS. Use the option **Do not enable BitLocker until recovery information is stored in AD DS** to prevent users from enabling BitLocker unless the backup of BitLocker recovery information for the drive to Microsoft Entra ID or AD DS succeeds.

BitLocker recovery password

To recover BitLocker, a user can use a recovery password, if available. The BitLocker recovery password is unique to the device it was created on, and can be saved in different ways. Depending on the configured policy settings, the recovery password can be:

- Saved in Microsoft Entra ID, for Microsoft Entra joined
- Saved in AD DS, for devices that are joined to Active Directory
- Saved on text file
- Printed

Having access to the recovery password allows the holder to unlock a BitLocker-protected volume and access all of its data. Therefore, it's important for your organization to establish procedures to control access to recovery passwords and ensure that they're stored securely, separate from the devices they protect.

> ⓘ **Note**
>
> There's an option for storing the BitLocker recovery key in a user's Microsoft account. The option is available for devices that aren't members of a domain and that the user is using a Microsoft account. Storing the recovery password in a Microsoft account is the default recommended recovery key storage method for devices that aren't Microsoft Entra joined or Active Directory joined.

Backup of the recovery password should be configured before BitLocker is enabled, but can also be done after encryption, as described in the BitLocker operations guide.
The preferred backup methodology in an organization is to automatically store BitLocker recovery information in a central location. Depending on the organization's requirements, the recovery information can be stored in Microsoft Entra ID, AD DS, or file shares.

The recommendation is to use the following BitLocker backup methods:

- For Microsoft Entra joined devices, store the recovery key in Microsoft Entra ID
- For Active Directory joined devices, store the recovery key in AD DS

> ⓘ **Note**
>
> There's no automatic way to store the recovery key for removable storage devices in Microsoft Entra ID or AD DS. However, you can use PowerShell or the `manage.bde.exe` command to do so. For more information and examples, review the **BitLocker operations guide**.

Data Recovery Agents

DRAs can be used to recover OS drives, fixed data drives, and removable data drives. However, when used to recover OS drives, the operating system drive must be mounted on another device as a *data drive* for the DRA to be able to unlock the drive. Data recovery agents are added to the drive when it's encrypted, and can be updated after encryption occurs.

The benefit of using a DRA over password or key recovery is that the DRA acts as a *master key* for BitLocker. With a DRA you can recover any volume protected by the policy, without having to find a specific password or key for each individual volume.

To configure DRAs for devices that are joined to an Active Directory domain, the following steps are required:

1. Obtain a DRA certificate. The following key usage and enhanced key usage attributes are inspected by BitLocker before using the certificate.
   a. If a key usage attribute is present, it must be either:

      - `CERT_DATA_ENCIPHERMENT_KEY_USAGE`
      - `CERT_KEY_AGREEMENT_KEY_USAGE`
      - `CERT_KEY_ENCIPHERMENT_KEY_USAGE`

   b. If an enhanced key usage (EKU) attribute is present, it must be either:

      - As specified in the policy setting, or the default `1.3.6.1.4.1.311.67.1.1`
      - Any EKU object identifier supported by your certification authority (CA)

2. Add the DRA via group policy using the path: **Computer configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Public Key Policies** > **BitLocker Drive Encryption**
3. Configure the Provide the unique identifiers for your organization policy setting to associate a unique identifier to a new drive that is enabled with BitLocker. An identification field is a string that is used to uniquely identify a business unit or organization. Identification fields are required for management of data recovery agents on BitLocker-protected drives. BitLocker only manages and updates DRAs when an identification field is present on a drive, and is identical to the value configured on the device
4. Configure the following policy settings to allow recovery using a DRA for each drive type:

   - Choose how BitLocker-protected operating system drives can be recovered
   - Choose how BitLocker-protected fixed drives can be recovered
   - Choose how BitLocker-protected removable drives can be recovered

BitLocker recovery information stored in Microsoft Entra ID

The BitLocker recovery information for Microsoft Entra joined devices can be stored in Microsoft Entra ID. The advantage of storing the BitLocker recovery passwords in Microsoft Entra ID, is that users can easily retrieve the passwords for the devices assigned to them from the web, without involving the help desk.

Access to recovery passwords can also be delegated to the help desk, to facilitate support scenarios.

The BitLocker recovery password information stored in Microsoft Entra ID is a `bitlockerRecoveryKey` resource type. The resource can be retrieved from the Microsoft Entra admin center, the Microsoft Intune admin center (for devices enrolled in Microsoft Intune), using PowerShell, or using Microsoft Graph. For more information, see bitlockerRecoveryKey resource type.

BitLocker recovery information stored in AD DS

The BitLocker recovery information for a device joined to an Active Directory domain can be stored in AD DS. The information is stored in a child object of the computer object itself. Each BitLocker recovery object includes the recovery password and other recovery information. More than one BitLocker recovery object can exist under each computer object, because there can be more than one recovery password associated with a BitLocker-enabled volume.

The name of the BitLocker recovery object incorporates a globally unique identifier (GUID) and date and time information, for a fixed length of 63 characters. The syntax is `<Object Creation Date and Time><Recovery GUID>`.

> **Note**
>
> Active Directory maintains history of all recovery passwords for a computer object. Old recovery keys are not removed automatically from AD DS, unless the computer object is deleted.

The common name (cn) for the BitLocker recovery object is `ms-FVE-RecoveryInformation`. Each `ms-FVE-RecoveryInformation` object has the following attributes:

Expand table

Attribute Name	Description
`ms-FVE-RecoveryPassword`	The 48-digit recovery password used to recover a BitLocker-encrypted disk volume.
`ms-FVE-RecoveryGuid`	GUID associated with a BitLocker recovery password. In BitLocker's recovery mode, the GUID is displayed to the user, so that the correct recovery password can be located to unlock the volume. The GUID is also included in the name of the recovery object.
`ms-FVE-VolumeGuid`	GUID associated with a BitLocker-supported disk volume. While the password (stored in `ms-FVE-RecoveryGuid`) is unique for each recovery password, the volume identifier is unique for each BitLocker-encrypted volume.
`ms-FVE-KeyPackage`	Volume's BitLocker encryption key secured by the corresponding recovery password. With this key package and the recovery password (stored in `ms-FVE-RecoveryPassword`), portions of a BitLocker-protected volume can be decrypted if the disk is corrupted. Each key package works only for a volume that has the corresponding volume identifier (stored in `ms-FVE-VolumeGuid`). The BitLocker Repair Tool can be used to make use of the key package.

To learn more about the BitLocker attributes stored in AD DS, review the following articles:

- ms-FVE-KeyPackage attribute
- ms-FVE-RecoveryPassword attribute

The BitLocker key package isn't saved by default. To save the package along with the recovery password in AD DS, the **Backup recovery password and key package** policy setting must be selected in the policy that controls the recovery method. The key package can also be exported from a working volume.

If recovery information isn't backed up to AD DS, or if you want to save a key package in an alternative location, use the following command to generate a key package for a volume:

Windows Command Prompt

```
manage-bde.exe -KeyPackage C: -id <id> -path <path>
```

A file with a file name format of `BitLocker Key Package {<id>}.KPG` is created in the specified path.

> ⓘ Note

To export a new key package from an unlocked, BitLocker-protected volume, local administrator access to the working volume is required before any damage occurrs to the volume.

Next steps

Learn how to obtain BitLocker recovery information for Microsoft Entra joined, Microsoft Entra hybrid joined, and Active Directory joined devices, and how to restore access to a locked drive:

BitLocker recovery process >

Last updated on 07/29/2025

BitLocker recovery process

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

If a device or drive fails to unlock using the configured BitLocker mechanism, users may be able to self-recover it. If self-recovery isn't an option, or the user is unsure how to proceed, the helpdesk should have procedures in place to retrieve recovery information quickly and securely.

This article outlines the process of obtaining BitLocker recovery information for Microsoft Entra joined, Microsoft Entra hybrid joined, and Active Directory joined devices. It's assumed that the reader is already familiar with configuring devices to automatically back up BitLocker recovery information, and the available BitLocker recovery options. For more information, see the BitLocker recovery overview article.

> **Note**
>
> This article is intended for IT professionals and device administrators. If you're an end user or consumer looking for help with BitLocker, see Find your BitLocker recovery key.

Self-recovery

The BitLocker recovery password and recovery key for an operating system drive or a fixed data drive can be saved to one or more USB devices, printed, saved to Microsoft Entra ID or AD DS.

> **Tip**
>
> Saving BitLocker recovery keys to Microsoft Entra ID or AD DS is a recommended approach. That way, a BitLocker administrator or helpdesk can assist users in attaining their keys.

If self-recovery includes using a password or recovery key stored on a USB flash drive, the users must be warned not to store the USB flash drive in the same place as the device, especially during travel. For example, if both the device and the recovery items are in the same bag, it would be easy for an unauthorized user to access the device. Another policy to consider is having users contact the helpdesk before or after performing self-recovery so that the root cause can be identified.

A recovery key can't be stored in any of the following locations:

- The drive being encrypted
- The root directory of a nonremovable drive
- An encrypted volume

> ⚠ **Warning**
>
> A recovery key is sensitive information that allows users to unlock an encrypted drive and perform administrative tasks on the drive. For enhanced security, it's recommended to enable self-service in trusted environments only, or rely on helpdesk recovery.

Self-recovery with recovery password

If you have access to the recovery key, enter the 48-digits in the preboot recovery screen.

- If you are having issues entering the recovery password in the preboot recovery screen, or you can no longer boot your device, you can connect the drive to another device as a secondary drive. For more information about the unlock process, see Unlock a drive
- If unlocking with recovery password doesn't work you can use the BitLocker Repair tool to regain access to your drive

Self-recovery in Microsoft Entra ID

If BitLocker recovery keys are stored in Microsoft Entra ID, users can access them using the following URL: `https://myaccount.microsoft.com`. From the **Devices** tab, users can select a Windows device that they own, and select the option **View BitLocker Keys**.

> ⓘ **Note**
>
> By default, users can retrieve their BitLocker recovery keys from Microsoft Entra ID. This behavior can be modified with the option **Restrict users from recovering the BitLocker key(s) for their owned devices**. For more information, see Restrict member users' default permissions.

Self-recovery with USB flash drive

If users saved the recovery password on a USB drive, they can plug the drive into a locked device and follow the instructions. If the key was saved as a text file on the flash drive, users must use a different device to read the text file.

Helpdesk recovery

If a user doesn't have a self-service recovery option, the helpdesk should be able to assist the user with one of the following options:

- If the device is Microsoft Entra joined or Microsoft Entra hybrid joined, BitLocker recovery information can be retrieved from Microsoft Entra ID
- If the device is domain joined, recovery information can be retrieved from Active Directory
- If the device is configured to use a DRA, the encrypted drive can be mounted on another device as a *data drive* for the DRA to be able to unlock the drive

> **⚠ Warning**
>
> The backup of the BitLocker recovery password to Microsoft Entra ID or AD DS may not happen automatically. Devices should be configured with policy settings to enable automatic backup, as described the BitLocker recovery overview article.

The following list can be used as a template for creating a recovery process for recovery password retrieval by the helpdesk.

Expand table

	Recovery process step	Details
☐	Verify the user's identity	The person who is asking for the recovery password should be verified as the authorized user of that device. It should also be verified whether the device for which the user provided the name belongs to the user.
☐	Record the device name	The name of the user's device can be used to locate the recovery password in Microsoft Entra ID or AD DS.
☐	Record the recovery key ID	The recovery key ID can be used to locate the recovery password in Microsoft Entra ID or AD DS. The recovery key ID is displayed in the preboot recovery screen.
☐	Locate the recovery password	Locate the BitLocker recovery password using the device name or the recovery key ID from Microsoft Entra ID or AD DS.
☐	Root cause analysis	Before giving the user the recovery password, information should be gatherer to determine why the recovery is needed. The information can be used to perform root cause analysis.

☑	Recovery process step	Details
☐	Provide the user the recovery password	Since the 48-digit recovery password is long and contains a combination of digits, the user might mishear or mistype the password. The boot-time recovery console uses built-in checksum numbers to detect input errors in each 6-digit block of the 48-digit recovery password, and offers the user the opportunity to correct such errors.
☐	Rotate the recovery password	If automatic password rotation is configured, Microsoft Entra joined and Microsoft Entra hybrid joined devices generate a new recovery password and store it in Microsoft Entra ID. An administrator can also trigger password rotation on-demand, using Microsoft Intune or Microsoft Configuration Manager.

Helpdesk recovery in Microsoft Entra ID

There are a few Microsoft Entra ID roles that allow a delegated administrator to read BitLocker recovery passwords from the devices in the tenant. While it's common for organizations to use the existing Microsoft Entra ID *Cloud Device Administrator* or *Helpdesk Administrator* built-in roles, you can also create a custom role, delegating access to BitLocker keys using the `microsoft.directory/bitlockerKeys/key/read` permission. Roles can be delegated to access BitLocker recovery passwords for devices in specific Administrative Units.

> **Note**
>
> When devices that utilize Windows Autopilot are reused to join to Entra, **and there is a new device owner**, that new device owner must contact an administrator to acquire the BitLocker recovery key for that device. Custom role or administrative unit scoped administrators will continue to have access to BitLocker recovery keys for those devices that have undergone device ownership changes, unless the new device owner belongs to a custom role or adminstrative unit scope. In such an instance, the user will need to contact other scoped administrator for the recovery keys. For more information, see the article Find the primary user of an Intune device.

The Microsoft Entra admin center allows administrators to retrieve BitLocker recovery passwords. To learn more about the process, see View or copy BitLocker keys. Another option to access BitLocker recovery passwords is to use the Microsoft Graph API, which might be useful for integrated or scripted solutions. For more information about this option, see Get bitlockerRecoveryKey.

In the following example, we use Microsoft Graph PowerShell cmdlet Get-MgInformationProtectionBitlockerRecoveryKey to build a PowerShell function that retrieves

recovery passwords from Microsoft Entra ID:

PowerShell

```
function Get-EntraBitLockerKeys{
    [CmdletBinding()]
    param (
        [Parameter(Mandatory = $true, HelpMessage = "Device name to retrieve the
BitLocker keys from Microsoft Entra ID")]
        [string]$DeviceName
    )
    $DeviceID = (Get-MGDevice -filter "displayName eq '$DeviceName'").DeviceId
    if ($DeviceID){
      $KeyIds = (Get-MgInformationProtectionBitlockerRecoveryKey -Filter "deviceId
eq '$DeviceId'").Id
      if ($keyIds) {
        Write-Host -ForegroundColor Yellow "Device name: $devicename"
        foreach ($keyId in $keyIds) {
          $recoveryKey = (Get-MgInformationProtectionBitlockerRecoveryKey -
BitlockerRecoveryKeyId $keyId -Select "key").key
          Write-Host -ForegroundColor White " Key id: $keyid"
          Write-Host -ForegroundColor Cyan " BitLocker recovery key: $recoveryKey"
        }
        } else {
        Write-Host -ForegroundColor Red "No BitLocker recovery keys found for device
$DeviceName"
      }
    } else {
        Write-Host -ForegroundColor Red "Device $DeviceName not found"
    }
}

Install-Module Microsoft.Graph.Identity.SignIns -Scope CurrentUser -Force
Import-Module Microsoft.Graph.Identity.SignIns
Connect-MgGraph -Scopes 'BitlockerKey.Read.All' -NoWelcome
```

After the function is loaded, it can be used to retrieve BitLocker recovery passwords for a specific device. Example:

PowerShell

```
PS C:\> Get-EntraBitLockerKeys -DeviceName DESKTOP-53O32QI
Device name: DESKTOP-53O32QI
 Key id: 4290b6c0-b17a-497a-8552-272cc30e80d4
 BitLocker recovery key: 496298-461032-321464-595518-463221-173943-033616-139579
 Key id: 045219ec-a53b-41ae-b310-08ec883aaedd
 BitLocker recovery key: 158422-038236-492536-574783-256300-205084-114356-069773
```

> ⓘ **Note**

For devices that are managed by Microsoft Intune, BitLocker recovery passwords can be retrieved from the device properties in the Microsoft Intune admin center. For more information, see View details for recovery keys.

Helpdesk recovery in Active Directory Domain Services

To export a recovery password from AD DS, you must have *read access* to objects stored in AD DS. By default, only *Domain Administrators* have access to BitLocker recovery information, but access can be delegated to specific security principals.

To facilitate the retrieval of BitLocker recovery passwords from AD DS, you can use the *BitLocker Recovery Password Viewer* tool. The tool is included with the *Remote Server Administration Tools (RSAT)*, and it's an extension for the *Active Directory Users and Computers Microsoft Management Console (MMC)* snap-in.

With BitLocker Recovery Password Viewer you can:

- Check the Active Directory computer object's properties to retrieve the associated BitLocker recovery passwords
- Search Active Directory for BitLocker recovery password across all the domains in the Active Directory forest

The following procedures describe the most common tasks performed by using the BitLocker Recovery Password Viewer.

View the recovery passwords for a computer object

1. Open **Active Directory Users and Computers** MMC snap-in, and select the container or OU in which the computer objects is located
2. Right-click the computer object and select **Properties**
3. In the **Properties** dialog box, select the **BitLocker Recovery** tab to view the BitLocker recovery passwords that are associated with the computer

Locate a recovery password by using a password ID

1. In **Active Directory Users and Computers**, right-click the domain container and select **Find BitLocker Recovery Password**
2. In the **Find BitLocker Recovery Password** dialog box, type the first eight characters of the recovery password in the **Password ID (first 8 characters)** box, and select **Search**

Data Recovery Agents

If devices are configured with a DRA, the Helpdesk can use the DRA to unlock the drive. Once the BitLocker drive is attached to a device that has the private key of the DRA certificate, the drive can be unlocked by using the `manage-bde.exe` command.

For example, to list the DRA configured for a BitLocker-protected drive, use the following command:

Windows Command Prompt

```
C:\>manage-bde.exe -protectors -get D:

Volume D: [Local Disk]
All Key Protectors

    Data Recovery Agent (Certificate Based):
      ID: {3A8F7DEA-878F-4663-B149-EE2EC9ADE40B}
      Certificate Thumbprint:
        f46563b1d4791d5bd827f32265341ff9068b0c42
```

If the private key of the certificate with a thumbprint of `f46563b1d4791d5bd827f32265341ff9068b0c42` is available in the local certificate store, an administrator can use the following command to unlock the drive with the DRA protector:

Windows Command Prompt

```
manage-bde -unlock D: -Certificate -ct f46563b1d4791d5bd827f32265341ff9068b0c42
```

Post-recovery tasks

When a volume is unlocked using a recovery password:

- an event is written to the Event Log
- the platform validation measurements are reset in the TPM to match the current configuration
- the encryption key is released and is ready for on-the-fly encryption/decryption when data is written/read to and from the volume

After the volume is unlocked, BitLocker behaves the same way, regardless of how the access was granted.

> **Note**
>
> If you move an OS volume with a TPM protector to a different device and unlock it using a recovery protector, BitLocker will bind to the new TPM. Returning the volume to the

original device will prompt for the recovery protector due to the TPM mismatch. Once unlocked using recovery protector again, the volume will re-bind to the original device.

If a device experiences multiple recovery password events, an administrator should perform post-recovery analysis to determine the root cause of the recovery. Then, refresh the BitLocker platform validation to prevent entering a recovery password each time that the device starts up.

Determine the root cause of the recovery

If a user needed to recover the drive, it's important to determine the root cause that initiated the recovery as soon as possible. Properly analyzing the state of the computer and detecting tampering might reveal threats that have broader implications for enterprise security.

While an administrator can remotely investigate the cause of recovery in some cases, the user might need to bring the device that contains the recovered drive on site to analyze the root cause further. Here are some questions that can be used to help determine the root cause of the recovery:

Expand table

	Question
☐	*Which BitLocker protection mode is configured (TPM, TPM + PIN, TPM + startup key, startup key only)?*
☐	*If TPM mode is configured, was recovery caused by a boot file change?*
☐	*Which PCR profile is in use on the device?*
☐	*Did the user merely forget the PIN or lose the startup key?*
☐	*If recovery was caused by a boot file change, is the boot file change due to an intended user action (for example, BIOS upgrade), or a malicious software?*
☐	*When was the user last able to start the device successfully, and what might have happened to the device since then?*
☐	*Might the user have encountered malicious software or left the device unattended since the last successful startup?*

To help answer these questions, you can use the `manage-bde.exe -status` command to view the current configuration and protection mode. Scan the event log to find events that help indicate why recovery was initiated (for example, if a boot file change occurred).

Resolve the root cause

After you identify the cause of the recovery, BitLocker protection can be reset to avoid recovery on every startup.

The details of the reset can vary according to the root cause of the recovery. If root cause can't be determined, or if a malicious software or a rootkit infects the device, the helpdesk should apply best-practice virus policies to react appropriately.

> ⓘ **Note**
>
> BitLocker validation profile reset can be performed by suspending and resuming BitLocker.

Root cause

Steps

Unknown PIN

If a user has forgotten the PIN, the PIN must be reset while signed on to the computer in order to prevent BitLocker from initiating recovery each time the computer is restarted.

To prevent continued recovery due to an unknown PIN:

1. Unlock the device using the recovery password
2. From the BitLocker Control Panel applet, expand the drive and then select **Change PIN**
3. In the BitLocker Drive Encryption dialog, select **Reset a forgotten PIN**. If the signed in account isn't an administrator account, you must provide administrative credentials
4. In the PIN reset dialog, provide and confirm the new PIN to be used and then select **Finish**
5. The new PIN can be used the next time the drive needs to be unlocked

Lost startup key

If the USB flash drive that contains the startup key is lost, you can unlock the drive using the recovery key. A new startup can then be created using PowerShell, the Command Prompt, or the BitLocker Control Panel applet.

For examples how to add BitLocker protectors, review the BitLocker operations guide

Changes to boot files

This error occurs if the firmware is updated. BitLocker should be suspended before making changes to the firmware. Protection should then be resumed after the firmware update is complete. Suspending BitLocker prevents the device from going into recovery mode. However,

if changes happen when BitLocker protection is on, the recovery password can be used to unlock the drive and the platform validation profile is updated so that recovery doesn't occur the next time.

For examples how to suspend and resume BitLocker protectors, review the BitLocker operations guide

Rotate passwords

Administrators can configure a policy setting to enable automatic recovery password rotation for Microsoft Entra joined and Microsoft Entra hybrid joined devices.
When automatic recovery password rotation is enabled, devices automatically rotate the recovery password after the password is used to unlock the drive. This behavior helps to prevent the same recovery password from being used multiple times, which can be a security risk.

For more information, see configure recovery password rotation.

Another option is to initiate the rotation of recovery passwords for individual devices remotely using Microsoft Intune or Microsoft Configuration Manager.

To learn more how to rotate BitLocker recovery passwords using Microsoft Intune or Microsoft Configuration Manager, see:

- Microsoft Intune documentation
- Microsoft Configuration Manager documentation

BitLocker Repair tool

If the recovery methods discussed earlier in this document don't unlock the volume, the *BitLocker Repair tool* (`repair-bde.exe`) can be used to decrypt the volume at the block level. The tool uses the *BitLocker key package* to help recover encrypted data from severely damaged drives.

The recovered data can then be used to salvage encrypted data, even if the correct recovery password fails to unlock the damaged volume. It's recommended to still save the recovery password, as a key package can't be used without the corresponding recovery password.

Use the Repair tool in the following conditions:

- The drive is encrypted using BitLocker
- Windows doesn't start, or the BitLocker recovery screen doesn't start
- There isn't a backup copy of the data that is contained on the encrypted drive

> **Note**
>
> Damage to the drive may not be related to BitLocker. Therefore, it's recommended to try other tools to help diagnose and resolve the problem with the drive before using the BitLocker Repair Tool. The Windows Recovery Environment (Windows RE) provides more options to repair Windows.

The following limitations exist for Repair-bde:

- it can't repair a drive that failed *during* the encryption or decryption process
- it assumes that if the drive has any encryption, then the drive is fully encrypted

For a complete list of the `repair-bde.exe` options, see the Repair-bde reference.

> **Note**
>
> To export a key package from AD DS, you must have *read* access to the BitLocker recovery passwords and key packages that are stored in AD DS. By default, only Domain Admins have access to BitLocker recovery information, but access can be delegated to others.

Last updated on 07/29/2025

BitLocker preboot recovery screen

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

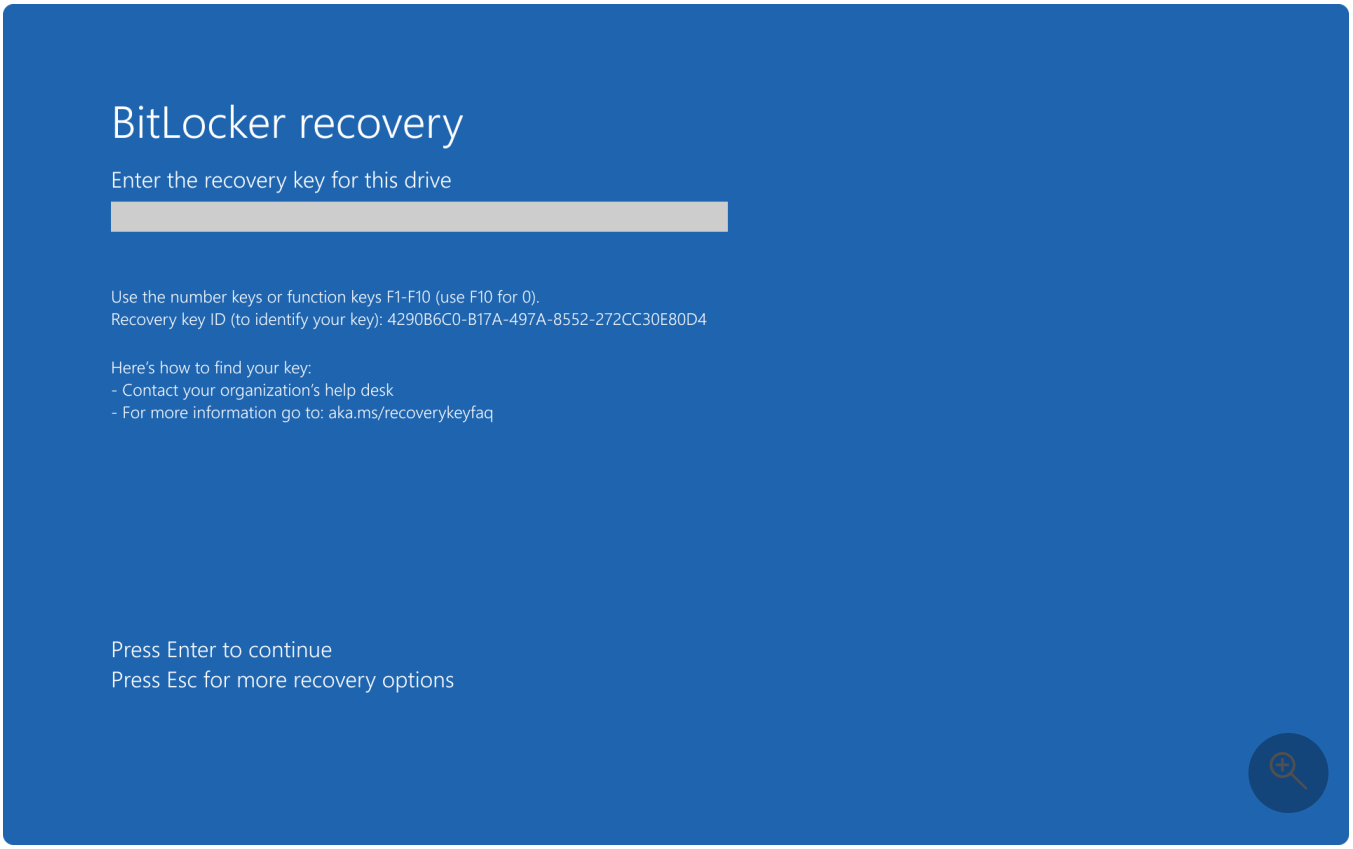
During BitLocker recovery, the *preboot recovery screen* is a critical touchpoint for users, offering a custom recovery message tailored to the organization's needs, a direct recovery URL for additional support, and strategic hints to assist users in locating their recovery key.

This article delves into the various elements displayed on the preboot recovery screen, detailing how policy settings and the status of recovery keys influence the information presented. Whether it's a personalized message or practical guidance, the preboot recovery screen is designed to streamline the recovery process for users.

> **Note**
>
> This article is intended for IT professionals and device administrators. If you're an end user or consumer looking for help with BitLocker, see Find your BitLocker recovery key.

Default preboot recovery screen

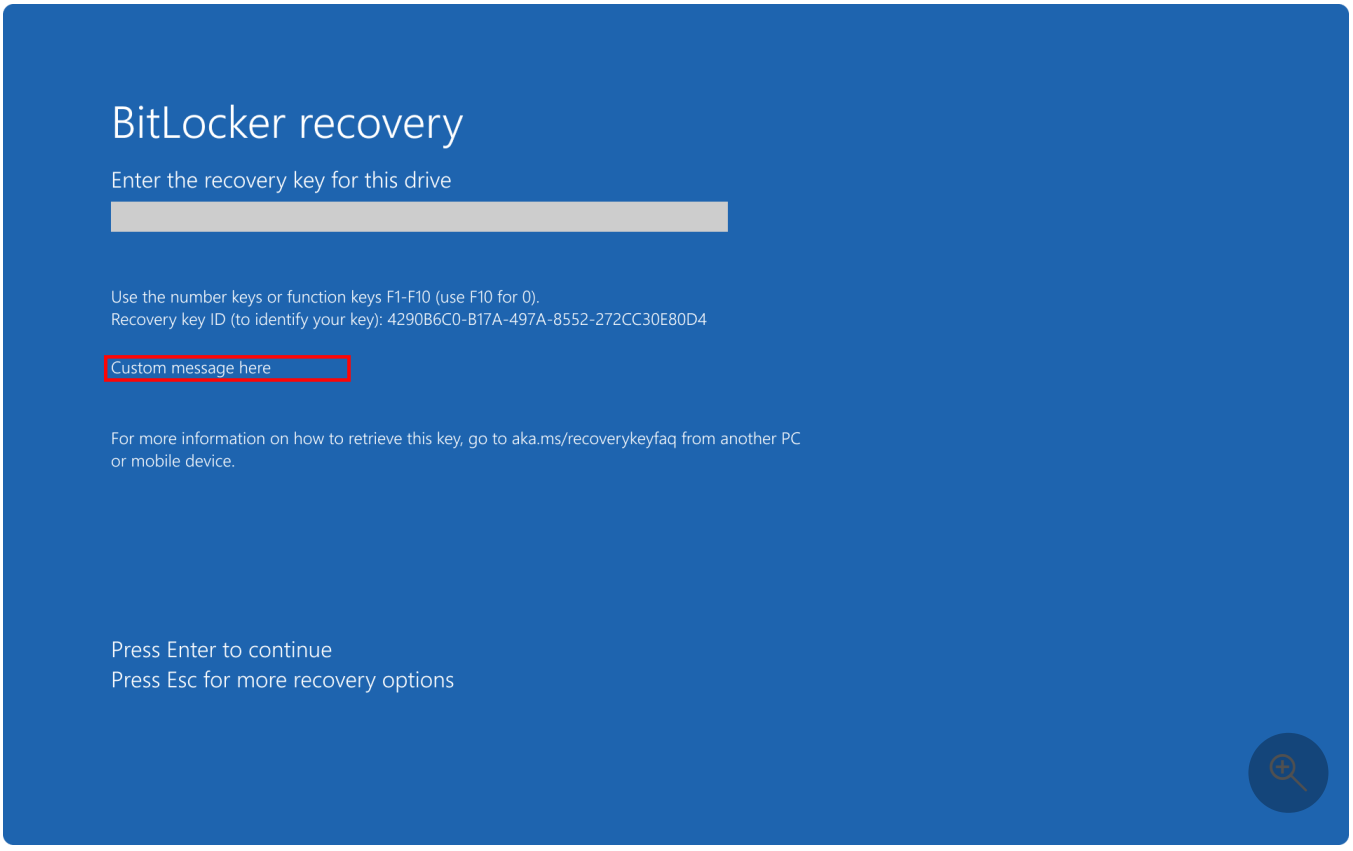
By default, the BitLocker recovery screen displays a generic message and the url **https://aka.ms/recoverykeyfaq**.



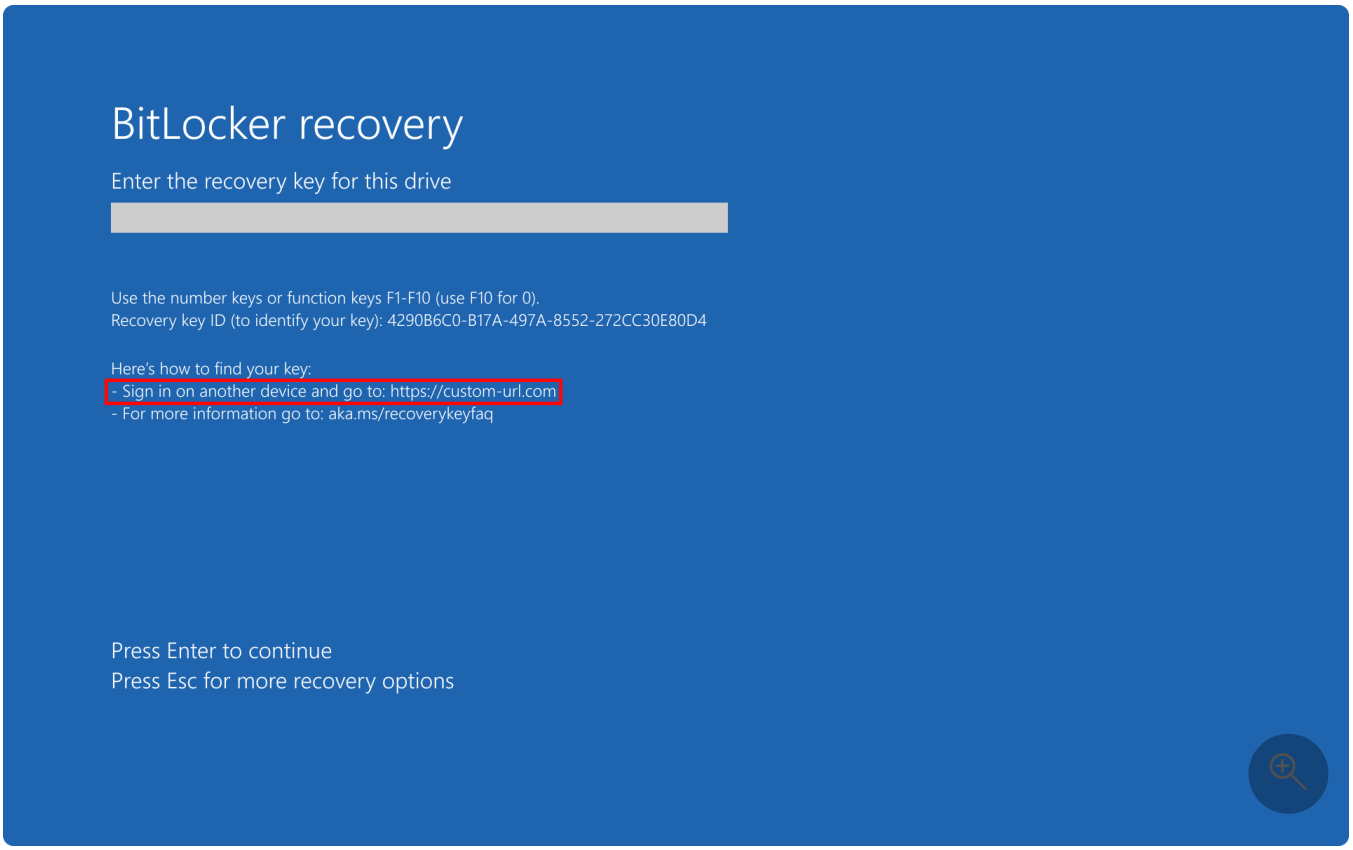
Custom recovery message

With BitLocker policy settings, you can configure a custom recovery message and URL on the BitLocker preboot recovery screen. The custom recovery message and URL can include the address of the BitLocker self-service recovery portal, the IT internal website, or a phone number for support.

BitLocker policy settings configured with a custom recovery message.



BitLocker policy settings configured with a custom recovery URL.



For more information how to configure a custom recovery message with policy settings, see Configure preboot recovery message and URL.

Recovery key hints

BitLocker metadata includes information about when and where a BitLocker recovery key was saved. This information isn't exposed through the UI or any public API. It's used solely by the BitLocker recovery screen in the form of hints to help a user locate a volume's recovery key. Hints are displayed on the recovery screen and refer to the location where the key was saved. The hints apply to both the boot manager recovery screen and the WinRE unlock screen.

There are rules governing which hint is shown during the recovery (in the order of processing):

1. Always display custom recovery message, if configured via policy settings
2. Always display generic hint: **For more information, go to https://aka.ms/recoverykeyfaq**
3. If multiple recovery keys exist on the volume, prioritize the last-created (and successfully backed up) recovery key
4. Prioritize keys with successful backup over keys that have never been backed up
5. Prioritize backup hints in the following order for remote backup locations:

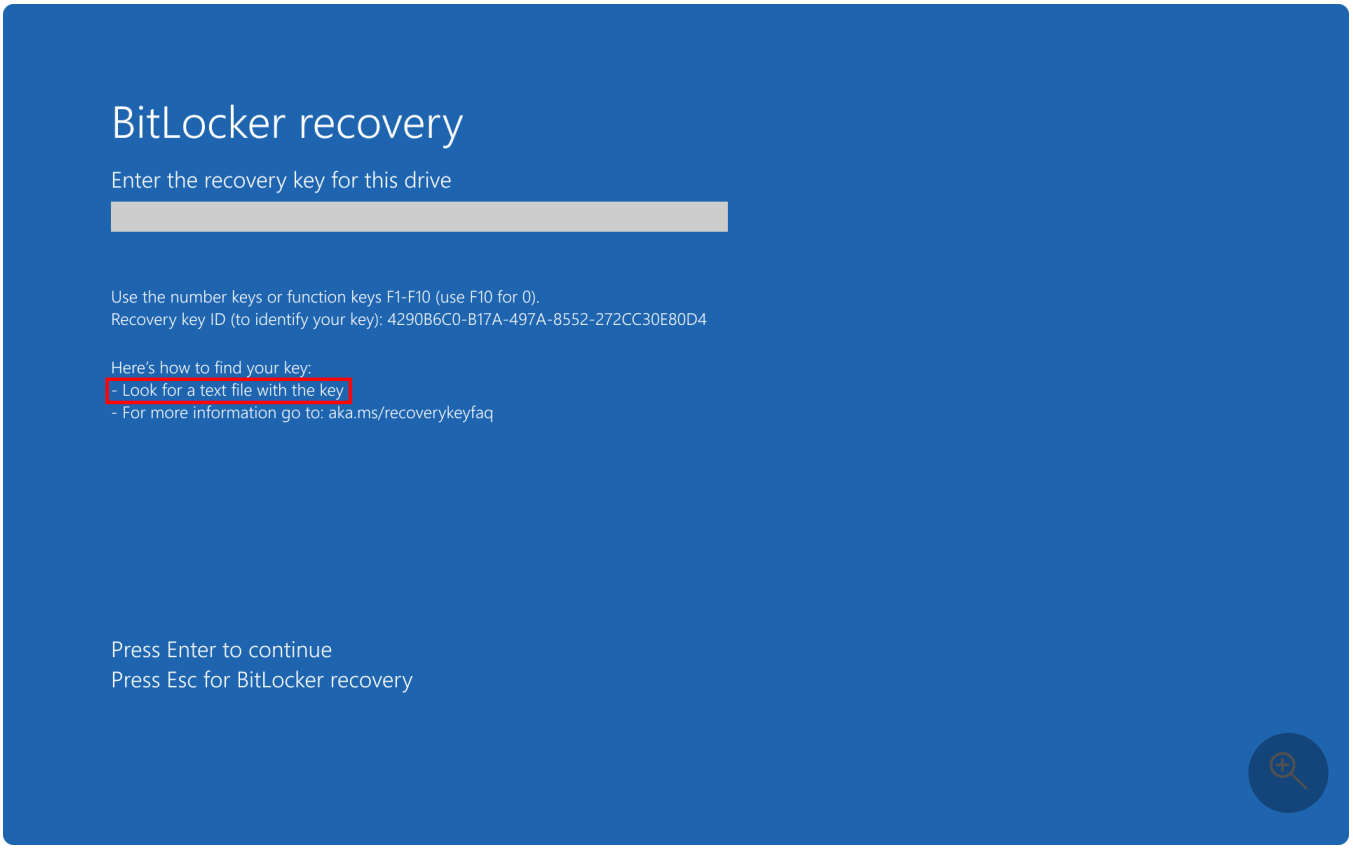
    - Microsoft account
    - Microsoft Entra ID
    - Active Directory

6. If a key has been printed and saved to file, display a combined hint **Look for a printout or a text file with the key**, instead of two separate hints
7. If multiple backups of the same type (remove vs. local) were done for the same recovery key, prioritize backup info with latest backup date
8. There's no specific hint for keys saved to an on-premises Active Directory. In this case, a custom message (if configured) or a generic message, **Contact your organization's help desk**, is displayed
9. If two recovery keys are present and only one backed up, the system asks for the backed up key, even if the other key is newer

Example: single recovery password saved to file and single backup

In this scenario, the recovery password is saved to a file

> **Important**
>
> It's not recommend to print recovery keys or saving them to a file. Instead, use Microsoft account, Microsoft Entra ID or Active Directory backup.

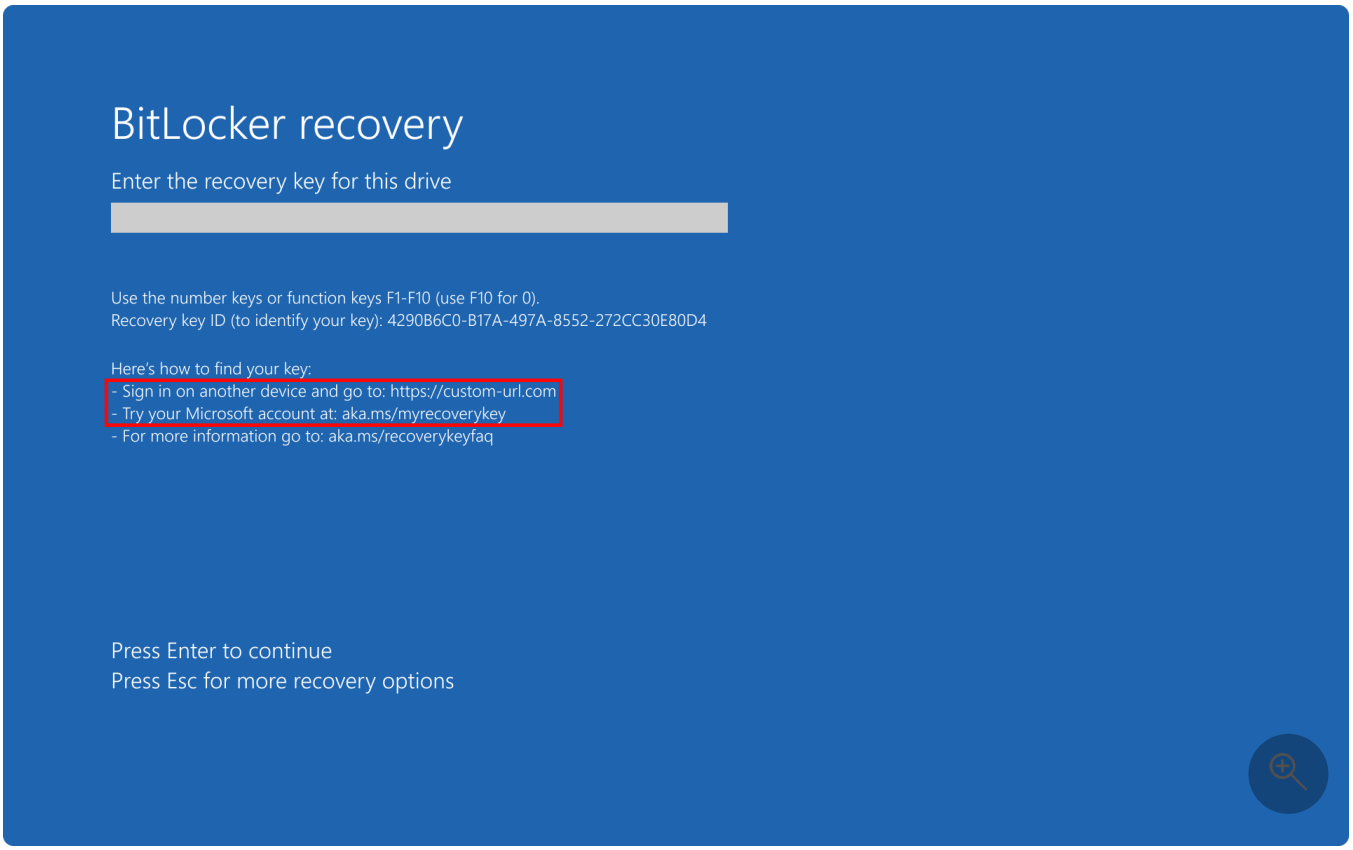


Example: single recovery password for Microsoft account and single backup

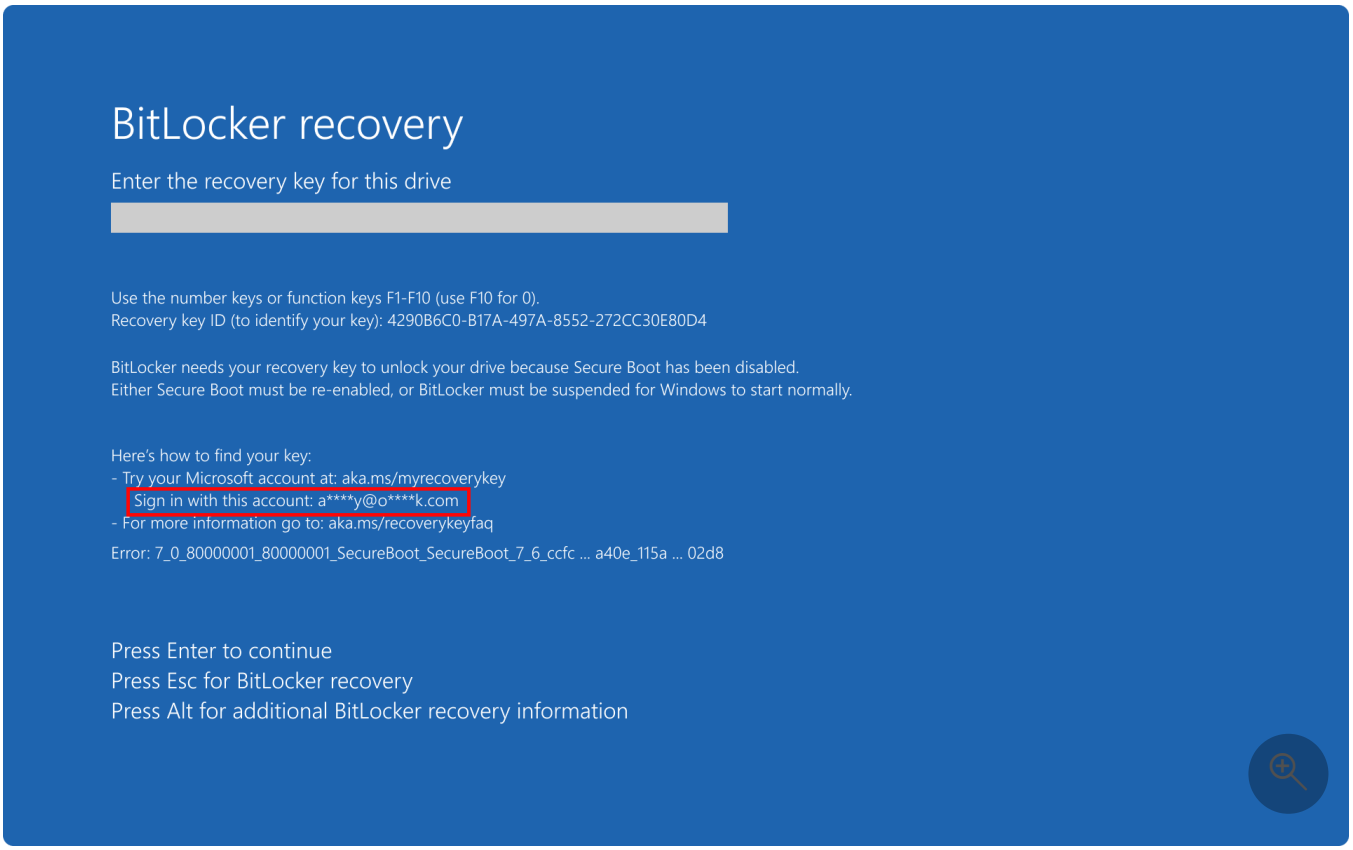
In this scenario, a custom URL is configured. The recovery password is:

- saved to Microsoft account
- not printed
- not saved to a file

Result: the hints for the custom URL and the Microsoft account (**https://aka.ms/myrecoverykey**) are displayed.



Starting in Windows 11, version 24H2, the BitLocker preboot recovery screen includes the Microsoft account (MSA) hint, if the recovery password is saved to an MSA. This hint helps the user to understand which MSA account was used to store recovery key information.

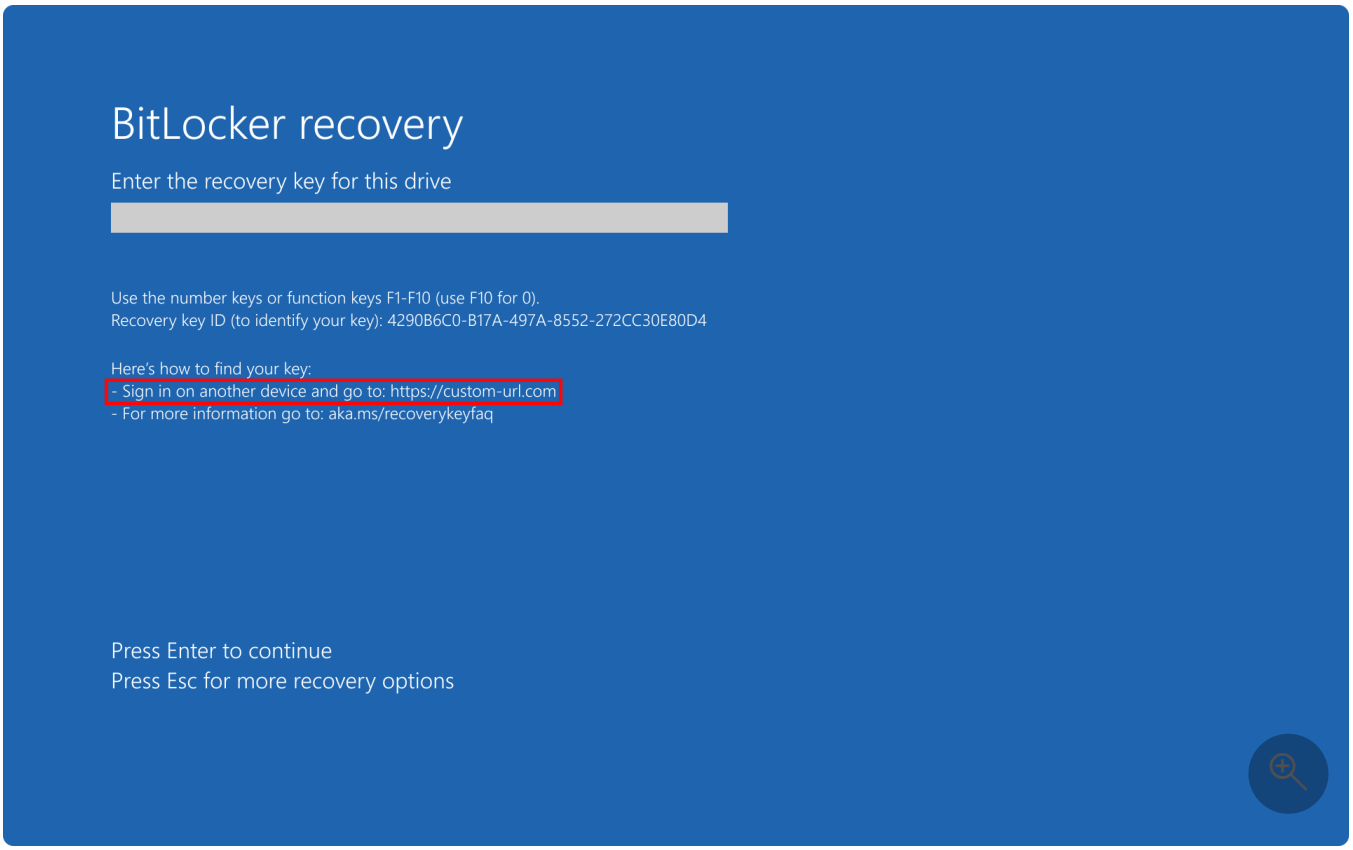


Example: single recovery password in AD DS and single backup

In this scenario, a custom URL is configured. The recovery password is:

- saved to Active Directory
- not printed
- not saved to a file

Result: only the custom URL is displayed.

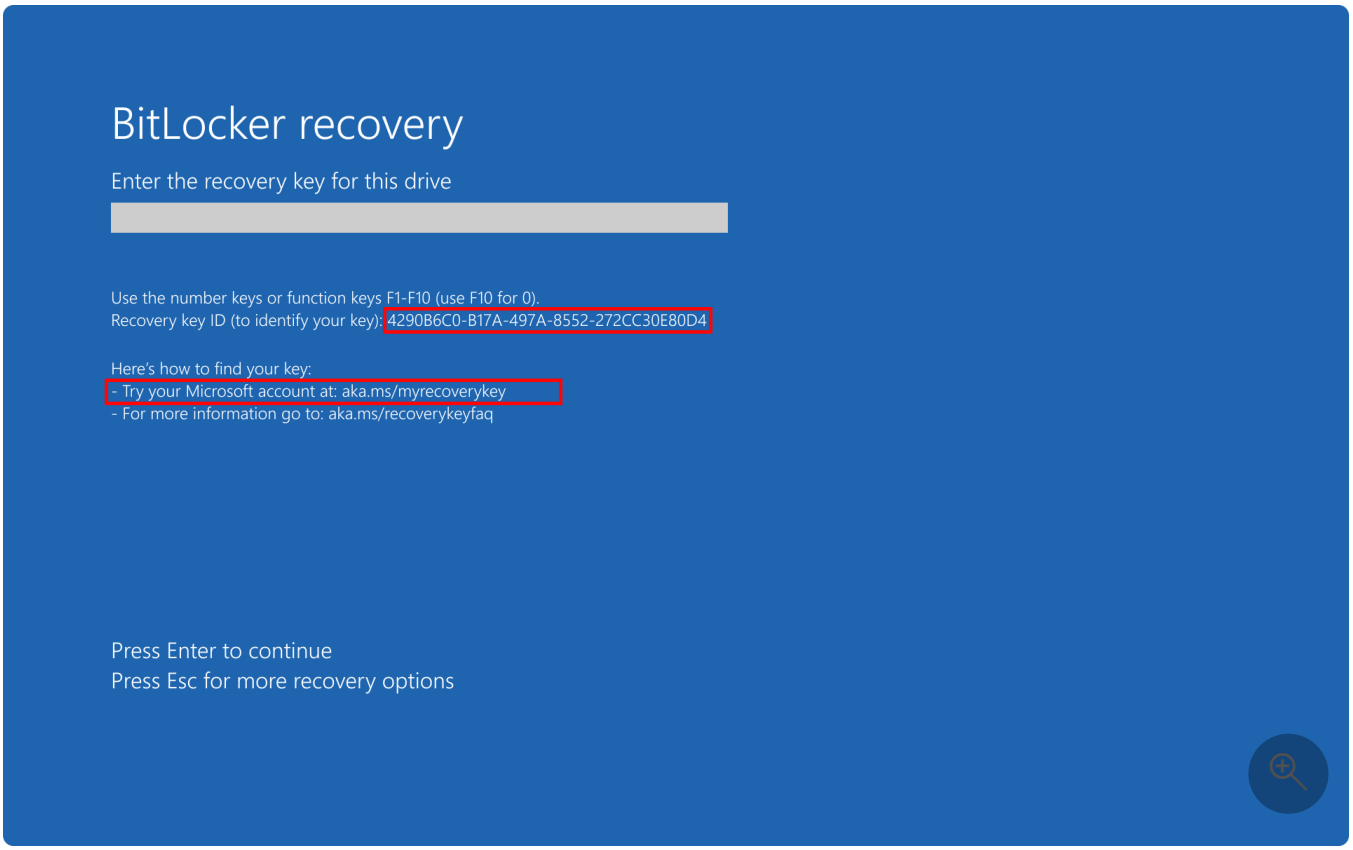


Example: single recovery password with multiple backups

In this scenario, the recovery password is:

- saved to Microsoft account
- saved to Microsoft Entra ID
- printed
- saved to file

Result: only the Microsoft account hint (**https://aka.ms/myrecoverykey**) is displayed.



Example: multiple recovery passwords with sinlge backup

In this scenario, there are two recovery passwords.

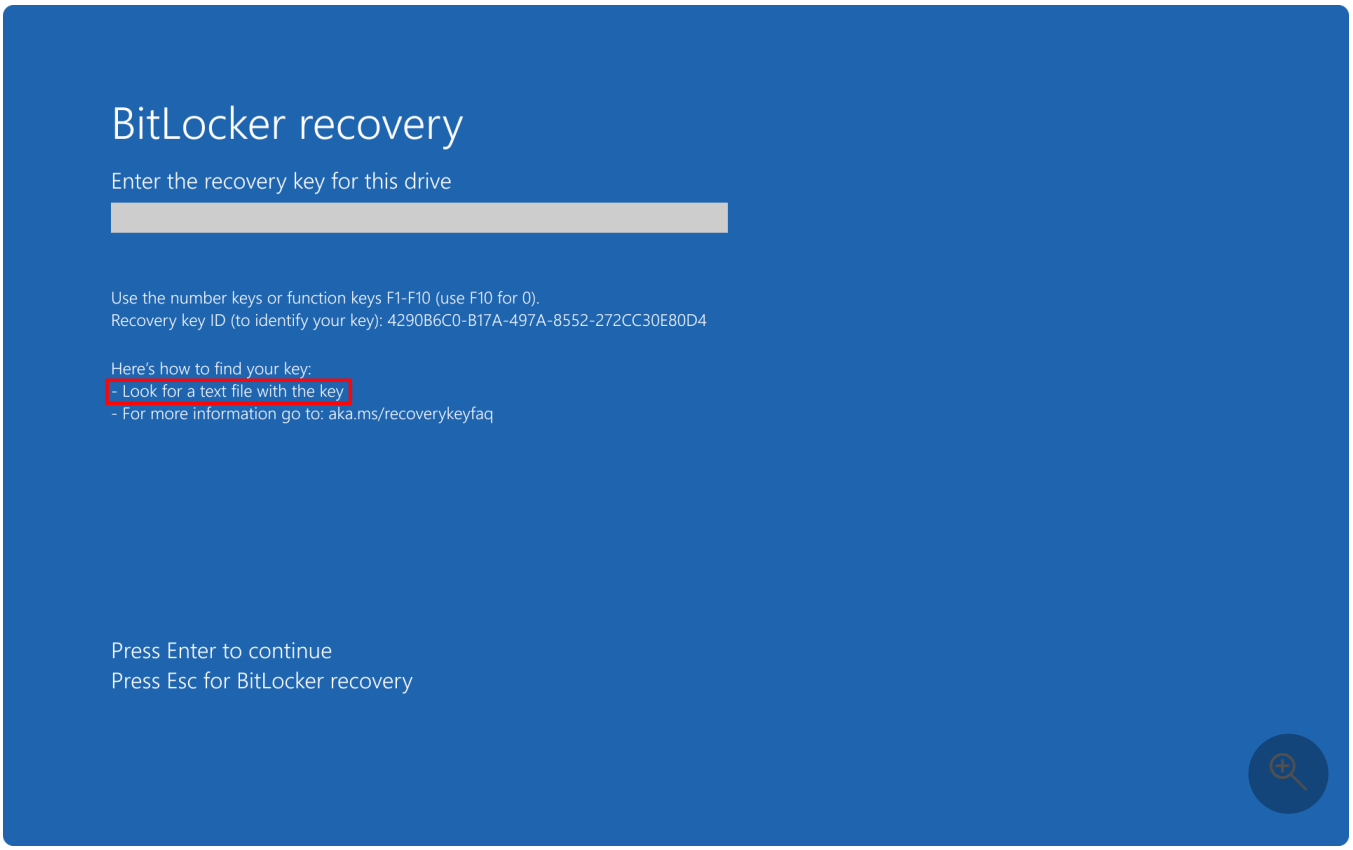
The recovery password #1 is:

- saved to file
- creation time: **1PM**
- key ID: **4290B6C0-B17A-497A-8552-272CC30E80D4**

The recovery password #2 is:

- not backed up
- creation time: **3PM**
- key ID: **045219EC-A53B-41AE-B310-08EC883AAEDD**

Result: only the hint for the successfully backed up key is displayed, even if it isn't the most recent key.



Example: multiple recovery passwords with multiple backups

In this scenario, there are two recovery passwords.

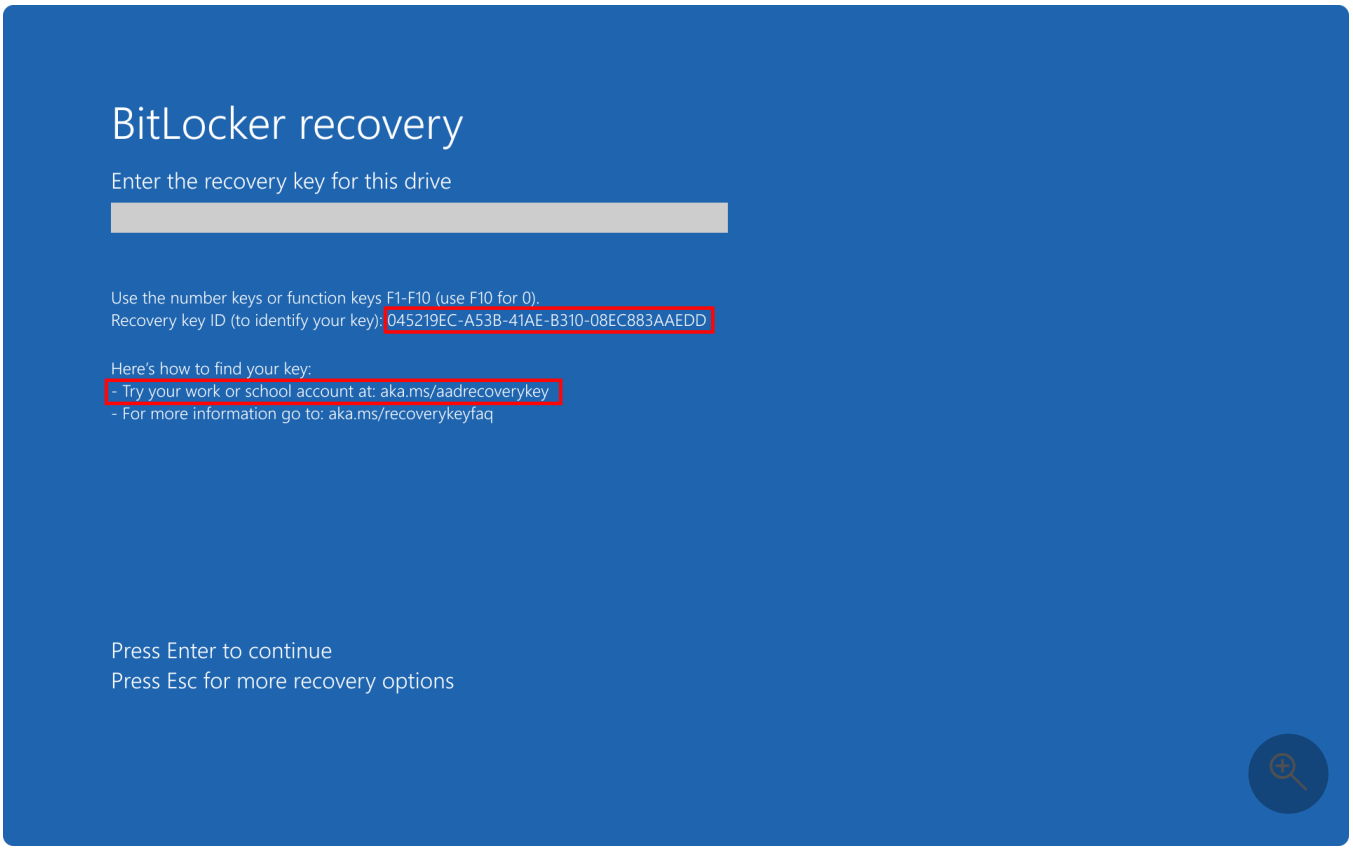
The recovery password #1 is:

- Saved to Microsoft account
- Saved to Microsoft Entra ID
- creation time: **1PM**
- key ID: **4290B6C0-B17A-497A-8552-272CC30E80D4**

The recovery password #2 is:

- Saved to Microsoft Entra ID
- creation time: **3PM**
- key ID: **045219EC-A53B-41AE-B310-08EC883AAEDD**

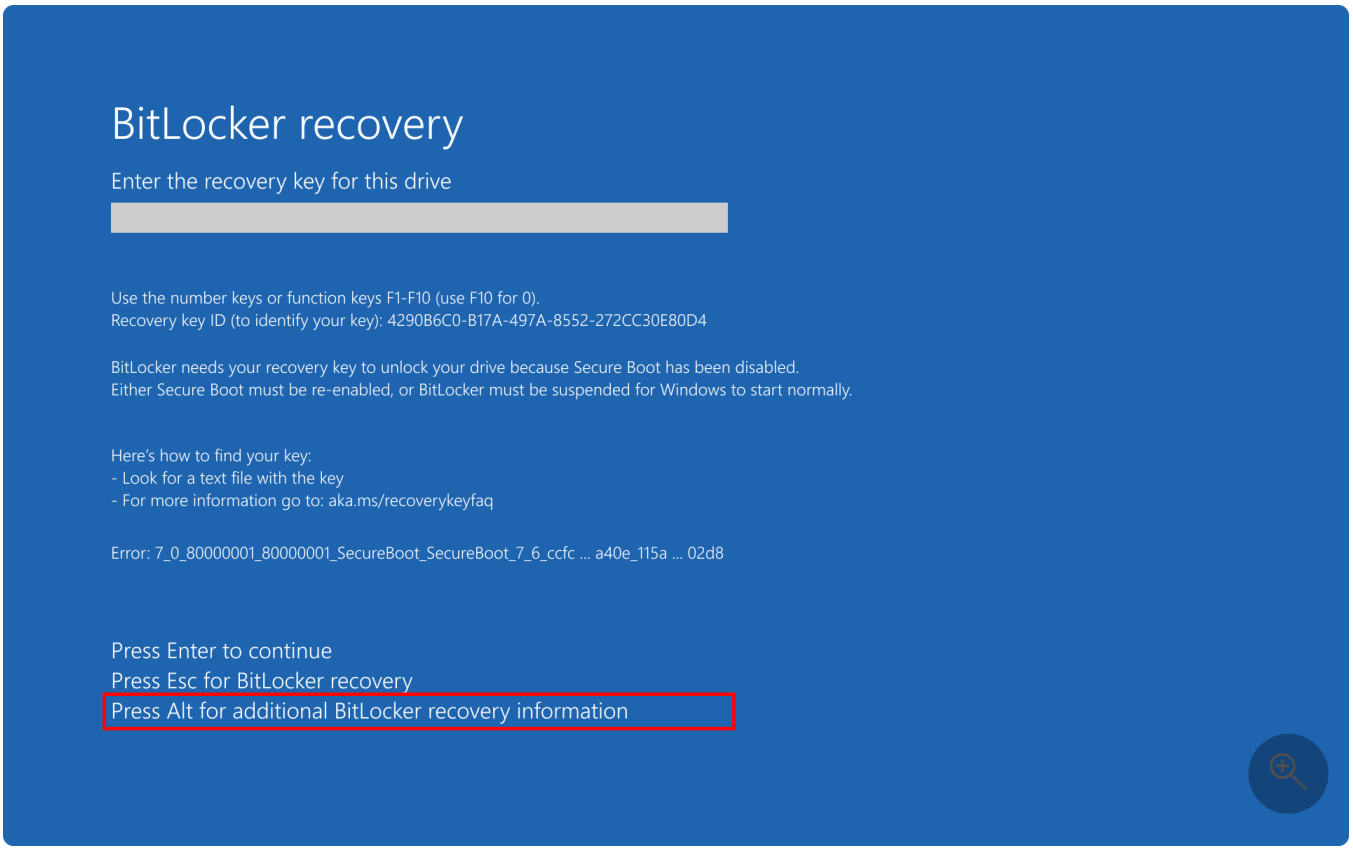
Result: the Microsoft Entra ID hint (**https://aka.ms/aadrecoverykey**), which is the most recent key saved, is displayed.



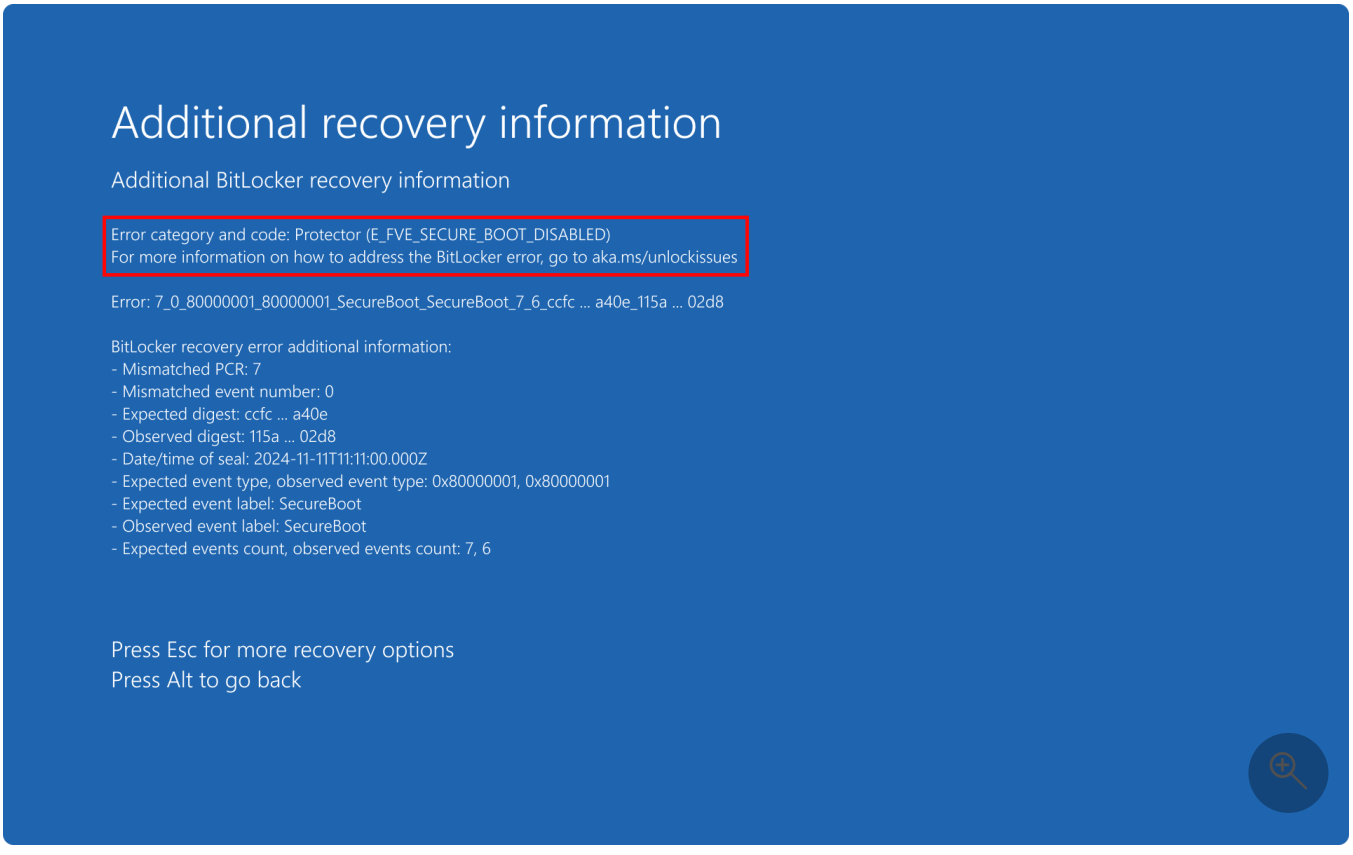
Additional recovery information screen

Starting in Windows 11, version 24H2, the BitLocker preboot recovery screen enhances the recovery error information. The recovery screen provides more detailed information about the nature of the recovery error, empowering users to better understand and address the issue.

Users have the option to review additional information about the recovery error by pressing the `Alt` key.



The **Additional recovery information** screen contains an *error category* and a *code*, which you can use to retrieve more details from the next section of this article.



The next sections describe the codes for each BitLocker error category. Within each section there's a table with the error message displayed on the recovery screen, and the cause of the

error. Some tables include possible resolution.

The error categories are:

- Initiated by user
- Code integrity
- Device lockout
- Boot configuration
- TPM
- Protector
- Unknown

Initiated by user

Expand table

Error code	Error cause	Resolution
`E_FVE_USER_REQUESTED_RECOVERY`	The user explicitly entered recovery mode from a screen with the option to `ESC` to recovery mode.	
`E_FVE_BOOT_DEBUG_ENABLED`	Boot debugging mode is enabled.	Remove the boot debugging option from the boot configuration database.

Code integrity

Driver signature enforcement is used to ensure code integrity of the operating system.

Expand table

Error code	Error cause
`E_FVE_CI_DISABLED`	Driver signature enforcement is disabled.

Device lockout

Device lockout threshold functionality allows an administrator to configure Windows sign in with BitLocker protection. After the configured number of failed Windows sign in attempts, the device reboots and can only be recovered by providing a BitLocker recovery method.

To take advantage of this functionality, you must configure the policy setting **Interactive logon: Machine account lockout threshold** located in **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**. Alternatively, use the Exchange ActiveSync **MaxFailedPasswordAttempts** policy setting, or the DeviceLock Configuration Service Provider (CSP).

Expand table

Error code	Error cause	Resolution
`E_FVE_DEVICE_LOCKEDOUT`	Device lockout triggered due to too many incorrect sign in attempts.	A BitLocker recovery method is required to return to the sign in screen.
`E_FVE_DEVICE_LOCKOUT_MISMATCH`	The device lockout counter is out of sync.	A BitLocker recovery method is required to return to the sign in screen.

Boot configuration

The *Boot Configuration Database (BCD)* contains critical information for the Windows boot environment.

Expand table

Error code	Error cause	Resolution

`E_FVE_BAD_CODE_ID`
 	
`E_FVE_BAD_CODE_OPTION`
	BitLocker entered recovery mode because a boot application changed. BitLocker tracks the data inside the BCD and BitLocker recovery can occur when this data changes without warning.
	
	Refer to the recovery screen to find the boot application that changed.
		To remediate this issue, restore the BCD configuration. A BitLocker recovery method is required to unlock the device if the BCD configuration can't be restored before booting.

For more information, see Boot Configuration Data settings and BitLocker.

TPM

The Trusted Platform Module (TPM) is cryptographic hardware or firmware used to secure a device. BitLocker creates a *TPM protector* to manage protection of the encryption keys used to

encrypt your data.

At boot, BitLocker attempts to communicate with the TPM to unlock the device and access your data.

Expand table

Error code	Error cause
`E_FVE_TPM_DISABLED`	A TPM is present but is disabled for use before or during boot.
`E_FVE_TPM_INVALIDATED`	A TPM is present but invalidated.
`E_FVE_BAD_SRK`	The TPM's internal Storage Root Key is corrupted.
`E_FVE_TPM_NOT_DETECTED`	The booting system doesn't have or doesn't detect a TPM.
`E_MATCHING_PCRS_TPM_FAILURE`	The TPM unexpectedly failed when unsealing the encryption key.
`E_FVE_TPM_FAILURE`	Catch-all for other TPM errors.

For more information, see Trusted Platform Module Technology Overview and BitLocker and TPM.

Protector

TPM protectors

The TPM contains multiple Platform Configuration Registers (PCRs) that can be used in the validation profile of the BitLocker TPM protector. The PCRs are used to validate the integrity of the boot process, that is, that the boot configuration and boot flow hasn't been tampered with.

BitLocker recovery can be the result of unexpected changes in the PCRs used in the TPM protector validation profile. Changes to PCRs not used in the TPM protector profile don't influence BitLocker.

Expand table

Error code	Error cause	Resolution
`E_FVE_PCR_MISMATCH`	The device's configuration changed. Possible causes include: - A bootable media is inserted. Removing it and restarting your device might fix this problem	A recovery method is required to unlock the device.

Error code	Error cause	Resolution
	- A firmware update was applied without updating the TPM protector	

For more examples, see BitLocker recovery scenarios.

Special cases for PCR 7

If the TPM protector uses PCR 7 in the validation profile, BitLocker expects PCR 7 to measure a specific set of events for Secure Boot. These measurements are defined in the UEFI spec. For more information, see Static Root of Trust Measurements

Expand table

Error code	Error cause	Resolution
`E_FVE_SECUREBOOT_DISABLED`	Secure Boot has been disabled. To access the encryption key and unlock your device, BitLocker expects Secure Boot to be on.	Re-enabling Secure Boot and rebooting the system might fix the recovery issue. Otherwise, a recovery method is required to access the device.
`E_FVE_SECUREBOOT_CHANGED`	The Secure Boot configuration unexpectedly changed. The boot configuration measured in PCR 7 changed. This may be either because of: - An additional measurement currently present that wasn't present when BitLocker updated the TPM protector - A missing measurement that was present when BitLocker last updated the TPM protector but now isn't present - An expected event has a different measurement	A recovery method is required to unlock the device.

Unknown

Expand table

Error code	Error cause	Resolution
`E_FVE_RECOVERY_ERROR_UNKNOWN`	BitLocker entered recovery mode because of an unknown error.	A recovery method is required to unlock the device.

Last updated on 07/29/2025

Install BitLocker on Windows Server

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

For all Windows Server editions, BitLocker isn't installed by default, but it can be installed using Server Manager or Windows PowerShell cmdlets. This article explains how to install BitLocker on Windows Server.

> ⓘ Note
>
> To install BitLocker you must have administrator privileges.

Install BitLocker with Server Manager

1. Open Server Manager by selecting the icon or running `servermanager.exe`
2. Select **Manage** from the **Server Manager Navigation** bar and select **Add Roles and Features**
3. Select **Next** at the **Before you begin** pane (if shown)
4. Under **Installation type**, select **Role-based or feature-based installation** and select **Next**
5. Under **Server Selection**, select the **Select a server from the server pool** pane and confirm the server on which you want to install the BitLocker feature and **Next**
6. Under **Server Roles** select **Next**
7. Under **Features**, select the box next to **BitLocker Drive Encryption**. The wizard shows the extra management features available for BitLocker. If you don't need the extra management features, deselect **Include management tools**

   > ⓘ Note
   >
   > The **Enhanced Storage** feature is a required feature for enabling BitLocker. This feature enables support for encrypted hard drives on capable systems.
8. Select **Next** and under **Confirmation** select **Install**

The BitLocker feature requires a restart to complete its installation. Selecting the **Restart the destination server automatically if required** option in the **Confirmation** pane forces a restart of the server after installation is complete

Install BitLocker with Windows PowerShell

Windows PowerShell offers administrators another option for BitLocker feature installation. Windows PowerShell installs features using the `servermanager` or `dism.exe` module. However, the `servermanager` and `dism.exe` modules don't always share feature name parity. Because of this mismatch of feature name parity, it's advisable to confirm the feature or role name prior to installation.

> **Note**
>
> The server must be restarted to complete the installation of BitLocker.

Use the servermanager module to install BitLocker

The `servermanager` Windows PowerShell module can use either the `Install-WindowsFeature` or `Add-WindowsFeature` to install the BitLocker feature. The `Add-WindowsFeature` cmdlet is a stub to the `Install-WindowsFeature`. This example uses the `Install-WindowsFeature` cmdlet. The feature name for BitLocker in the `servermanager` module is `BitLocker`.

By default, installation of features in Windows PowerShell doesn't include optional subfeatures or management tools as part of the installation process. What is installed as part of the installation process can be seen using the `-WhatIf` option in Windows PowerShell.

PowerShell

```
Install-WindowsFeature BitLocker -WhatIf
```

The results of this command show that only the BitLocker Drive Encryption feature is installed using this command.

To see what would be installed with the BitLocker feature, including all available management tools and subfeatures, use the following command:

PowerShell

```
Install-WindowsFeature BitLocker -IncludeAllSubFeature -IncludeManagementTools -
WhatIf | fl
```

The command to complete a full installation of the BitLocker feature with all available subfeatures and then to reboot the server at completion is:

PowerShell

```
Install-WindowsFeature BitLocker -IncludeAllSubFeature -IncludeManagementTools -
```

```
Restart
```

> **Important**
>
> Installing the BitLocker feature using Windows PowerShell does not install the Enhanced Storage feature. Administrators wishing to support Encrypted Hard Drives in their environment will need to install the Enhanced Storage feature separately.

Use the dism module to install BitLocker

The `dism.exe` Windows PowerShell module uses the `Enable-WindowsOptionalFeature` cmdlet to install features. The BitLocker feature name for BitLocker is `BitLocker`. The `dism.exe` module doesn't support wildcards when searching for feature names. To list feature names for the `dism.exe` module, use the `Get-WindowsOptionalFeatures` cmdlet. The following command lists all of the optional features in an online (running) operating system.

PowerShell

```
Get-WindowsOptionalFeature -Online | ft
```

From this output, there are three BitLocker-related optional feature names: **BitLocker**, **BitLocker-Utilities** and **BitLocker-NetworkUnlock**. To install the BitLocker feature, the **BitLocker** and **BitLocker-Utilities** features are the only required items.

To install BitLocker using the `dism.exe` module, use the following command:

PowerShell

```
Enable-WindowsOptionalFeature -Online -FeatureName BitLocker -All
```

The command doesn't include installation of the management tools for BitLocker, but you can do a complete installation of BitLocker and all available management tools with the following command:

PowerShell

```
Enable-WindowsOptionalFeature -Online -FeatureName BitLocker, BitLocker-Utilities -
All
```

> **Note**

When using `Enable-WindowsOptionalFeature`, the administrator is prompted to reboot the server, as the cmdlet doesn't have support for forcing a reboot.

After the server reboots, you can use BitLocker.

Last updated on 07/29/2025

Network Unlock

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Network Unlock is a BitLocker *key protector* for operating system volumes. Network Unlock enables easier management for BitLocker-enabled desktops and servers in a domain environment by providing automatic unlock of operating system volumes at system reboot when connected to a wired corporate network. Network Unlock requires the client hardware to have a DHCP driver implemented in its UEFI firmware. Without Network Unlock, operating system volumes protected by `TPM+PIN` protectors require a PIN to be entered when a device reboots or resumes from hibernation (for example, by Wake on LAN). Requiring a PIN after a reboot can make it difficult for enterprises to roll out software patches to unattended desktops and remotely administered servers.

Network Unlock allows BitLocker-enabled systems that have a `TPM+PIN` and that meet the hardware requirements to boot into Windows without user intervention. Network Unlock works in a similar fashion to the `TPM+StartupKey` at boot. Rather than needing to read the StartupKey from USB media, however, the Network Unlock feature needs the key to be composed from a key stored in the TPM and an encrypted network key that is sent to the server, decrypted and returned to the client in a secure session.

System requirements

Network Unlock must meet mandatory hardware and software requirements before the feature can automatically unlock domain-joined systems. These requirements include:

- Any supported operating system with UEFI DHCP drivers that can serve as Network Unlock clients
- Network Unlock clients with a TPM chip and at least one TPM protector
- A server running the Windows Deployment Services (WDS) role on any supported server operating system
- BitLocker Network Unlock optional feature installed on any supported server operating system
- A DHCP server, separate from the WDS server
- Properly configured public/private key pairing
- Network Unlock group policy settings configured
- Network stack enabled in the UEFI firmware of client devices

> ⓘ **Important**

To support DHCP within UEFI, the UEFI-based system should be in native mode and shouldn't have a compatibility support module (CSM) enabled.

For Network Unlock to work reliably, the first network adapter on the device, usually the onboard adapter, must be configured to support DHCP. This first network adapter must be used for Network Unlock. This configuration is especially worth noting when the device has multiple adapters, and some adapters are configured without DHCP, such as for use with a lights-out management protocol. This configuration is necessary because Network Unlock stops enumerating adapters when it reaches one with a DHCP port failure for any reason. Thus, if the first enumerated adapter doesn't support DHCP, isn't plugged into the network, or fails to report availability of the DHCP port for any reason, then Network Unlock fails.

The Network Unlock server component is installed on supported versions of Windows Server as a Windows feature that uses Server Manager or Windows PowerShell cmdlets. The feature name is `BitLocker Network Unlock` in Server Manager and `BitLocker-NetworkUnlock` in PowerShell.

Network Unlock requires Windows Deployment Services (WDS) in the environment where the feature will be utilized. Configuration of the WDS installation isn't required. However, the WDS service must be running on the server.

The network key is stored on the system drive along with an AES 256 session key and encrypted with the 2048-bit RSA public key of the Unlock server certificate. The network key is decrypted with the help of a provider on a supported version of Windows Server running WDS, and returned encrypted with its corresponding session key.

Network Unlock sequence

The unlock sequence starts on the client side when the Windows boot manager detects the existence of Network Unlock protector. It uses the DHCP driver in UEFI to obtain an IP address for IPv4 and then broadcasts a vendor-specific DHCP request that contains the network key and a session key for the reply, all encrypted by the server's Network Unlock certificate. The Network Unlock provider on the supported WDS server recognizes the vendor-specific request, decrypts it with the RSA private key, and returns the network key encrypted with the session key via its own vendor-specific DHCP reply.

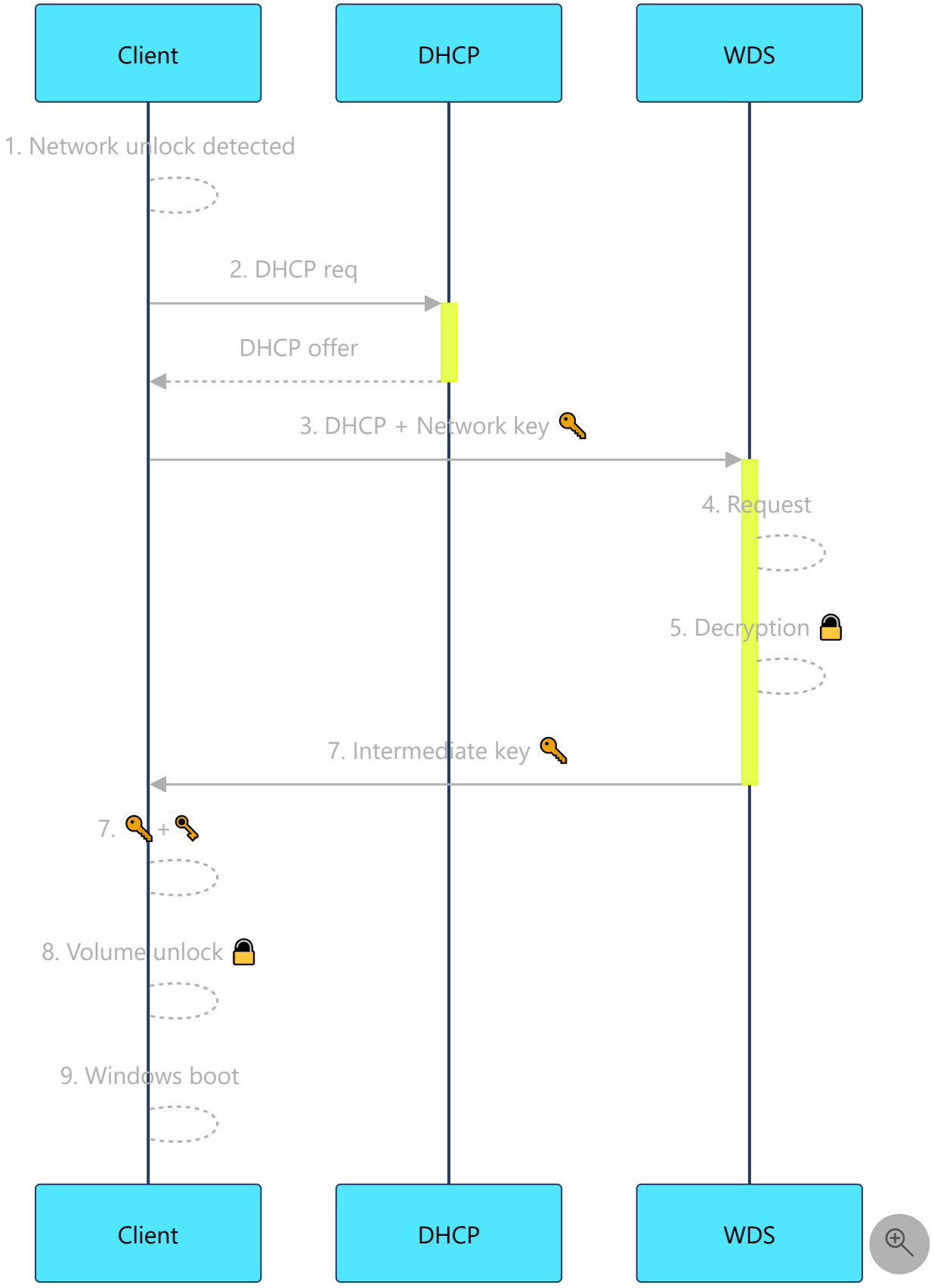
On the server side, the WDS server role has an optional plugin component, like a PXE provider, which is what handles the incoming Network Unlock requests. The provider can also be configured with subnet restrictions, which would require that the IP address provided by the client in the Network Unlock request belong to a permitted subnet to release the network key to the client. In instances where the Network Unlock provider is unavailable, BitLocker fails over

to the next available protector to unlock the drive. In a typical configuration, the standard `TPM+PIN` unlock screen is presented to unlock the drive.

The server side configuration to enable Network Unlock also requires provisioning a 2048-bit RSA public/private key pair in the form of an X.509 certificate, and distributing the public key certificate to the clients. This certificate is the *public key* that encrypts the intermediate network key (which is one of the two secrets required to unlock the drive; the other secret is stored in the TPM), and it must be managed and deployed via Group Policy.

The Network Unlock process follows these phases:

1. The Windows boot manager detects a Network Unlock protector in the BitLocker configuration
2. The client computer uses its DHCP driver in the UEFI to get a valid IPv4 IP address
3. The client computer broadcasts a vendor-specific DHCP request that contains a network key (a 256-bit intermediate key) and an AES-256 session key for the reply. The network key is encrypted by using the 2048-bit RSA Public Key of the Network Unlock certificate from the WDS server
4. The Network Unlock provider on the WDS server recognizes the vendor-specific request
5. The provider decrypts the request by using the WDS server's BitLocker Network Unlock certificate RSA private key
6. The WDS provider returns the network key encrypted with the session key by using its own vendor-specific DHCP reply to the client computer. This key is an intermediate key
7. The returned intermediate key is combined with another local 256-bit intermediate key. This key can be decrypted only by the TPM
8. This combined key is used to create an AES-256 key that unlocks the volume
9. Windows continues the boot sequence



Configure Network Unlock

The following steps allow an administrator to configure Network Unlock in an Active Directory domain.

Install the WDS server role

The BitLocker Network Unlock feature installs the WDS role if it isn't already installed. WDS can be installed separately, before BitLocker Network Unlock is installed, by using **Server Manager** or **PowerShell**. To install the role using Server Manager, select the **Windows Deployment Services** role in **Server Manager**.

To install the role by using PowerShell, use the following command:

```
PowerShell
Install-WindowsFeature WDS-Deployment
```

The WDS server must be configured so that it can communicate with DHCP (and optionally AD DS) and the client computer. The WDS server can be configured using the WDS management tool, `wdsmgmt.msc`, which starts the Windows Deployment Services Configuration wizard.

Confirm the WDS service is running

To confirm that the WDS service is running, use the Services Management Console or PowerShell. To confirm that the service is running in Services Management Console, open the console using `services.msc` and check the status of the **Windows Deployment Services** service.

To confirm that the service is running using PowerShell, use the following command:

```
PowerShell
Get-Service WDSServer
```

Install the Network Unlock feature

To install the Network Unlock feature, use Server Manager or PowerShell. To install the feature using Server Manager, select the **BitLocker Network Unlock** feature in the Server Manager console.

To install the feature by using PowerShell, use the following command:

```
PowerShell
Install-WindowsFeature BitLocker-NetworkUnlock
```

Create the certificate template for Network Unlock

A properly configured Active Directory Certification Authority can use this certificate template to create and issue Network Unlock certificates.

1. Open the Certificates Template snap-in (`certtmpl.msc`)
2. Locate the User template, right-click the template name and select **Duplicate Template**
3. On the **Compatibility** tab, change the **Certification Authority** and **Certificate recipient** fields to Windows Server 2016 and Windows 10, respectively. Ensure that the **Show resulting changes** dialog box is selected
4. Select the **General** tab of the template. The **Template display name** and **Template name** should identify that the template will be used for Network Unlock. Clear the check box for the **Publish certificate in Active Directory** option
5. Select the **Request Handling** tab. Select **Encryption** from the **Purpose** drop-down menu. Ensure that the **Allow private key to be exported** option is selected
6. Select the **Cryptography** tab. Set the **Minimum key size** to 2048. Any Microsoft cryptographic provider that supports RSA can be used for this template, but for simplicity and forward compatibility, it is recommended to use **Microsoft Software Key Storage Provider**
7. Select the **Requests must use one of the following providers** option and clear all options except for the cryptography provider selected, such as **Microsoft Software Key Storage Provider**
8. Select the **Subject Name** tab. Select **Supply in the request**. Select **OK** if the certificate templates pop-up dialog appears
9. Select the **Issuance Requirements** tab. Select both **CA certificate manager approval** and **Valid existing certificate** options
10. Select the **Extensions** tab. Select **Application Policies** and choose **Edit...**
11. In the **Edit Application Policies Extension** options dialog box, select **Client Authentication**, **Encrypting File System**, **and Secure Email** and choose **Remove**
12. On the **Edit Application Policies Extension** dialog box, select **Add**
13. On the **Add Application Policy** dialog box, select **New**. In the **New Application Policy** dialog box, enter the following information in the space provided and then select **OK** to create the BitLocker Network Unlock application policy:

- *Name:* **BitLocker Network Unlock**
- *Object Identifier:* **1.3.6.1.4.1.311.67.1.1**

14. Select the newly created **BitLocker Network Unlock** application policy and select **OK**

15. With the **Extensions** tab still open, select the **Edit Key Usage Extension** dialog. Select the **Allow key exchange only with key encryption (key encipherment)** option. Select the **Make this extension critical** option

16. Select the **Security** tab. Confirm that the **Domain Admins** group has been granted **Enroll** permission

17. Select **OK** to complete configuration of the template

To add the Network Unlock template to the certificate authority, open the certificate authority snap-in (`certsrv.msc`). Right-click **Certificate Templates**, and then choose **New, Certificate Template to issue**. Select the previously created BitLocker Network Unlock certificate.

After the Network Unlock template is added to the certificate authority, this certificate can be used to configure BitLocker Network Unlock.

Create the Network Unlock certificate

Network Unlock can use imported certificates from an existing public key infrastructure (PKI). Or it can use a self-signed certificate.

To enroll a certificate from an existing certificate authority:

1. On the WDS server, open Certificate Manager by using `certmgr.msc`
2. Under **Certificates - Current User**, right-click **Personal**
3. Select **All Tasks** > **Request New Certificate**
4. When the Certificate Enrollment wizard opens, select **Next**
5. Select **Active Directory Enrollment Policy**
6. Choose the certificate template that was created for Network Unlock on the domain controller. In case the message "More information is required to enroll for this certificate. Click here to configure settings." is shown, click on it. On the new window, in **Subject** tab, under **Alternative names**, select **DNS** and set the FQDN of the WDS server. Save the changes by clicking **OK** and then select **Enroll**
7. When prompted for more information, select **Subject Name** and provide a friendly name value. The friendly name should include information for the domain or organizational unit for the certificate For example: *BitLocker Network Unlock Certificate for Contoso domain*
8. Create the certificate. Ensure the certificate appears in the **Personal** folder
9. Export the public key certificate for Network Unlock:

a. Create a `.cer` file by right-clicking the previously created certificate, selecting **All Tasks**, and then selecting **Export**
b. Select **No, do not export the private key**
c. Select **DER encoded binary X.509** and complete exporting the certificate to a file
d. Give the file a name such as **BitLocker-NetworkUnlock.cer**

10. Export the public key with a private key for Network Unlock
    a. Create a `.pfx` file by right-clicking the previously created certificate, selecting **All Tasks**, and then selecting **Export**
    b. Select **Yes, export the private key**
    c. Complete the steps to create the `.pfx` file

To create a self-signed certificate, either use the `New-SelfSignedCertificate` cmdlet in Windows PowerShell or use `certreq.exe`. For example:

PowerShell

PowerShell

```
New-SelfSignedCertificate -CertStoreLocation Cert:\LocalMachine\My -Subject "CN=BitLocker Network Unlock certificate" -Provider "Microsoft Software Key Storage Provider" -KeyUsage KeyEncipherment -KeyUsageProperty Decrypt,Sign -KeyLength 2048 -HashAlgorithm sha512 -TextExtension @("1.3.6.1.4.1.311.21.10={text}OID=1.3.6.1.4.1.311.67.1.1","2.5.29.37={text}1.3.6.1.4.1.311.67.1.1")
```

certreq.exe

1. Create a text file with an `.inf` extension, for example:

   Windows Command Prompt

   ```
   notepad.exe BitLocker-NetworkUnlock.inf
   ```

2. Add the following contents to the previously created file:

   ini

   ```
   [NewRequest]
   Subject="CN=BitLocker Network Unlock certificate"
   ProviderType=0
   MachineKeySet=True
   Exportable=true
   RequestType=Cert
   KeyUsage="CERT_KEY_ENCIPHERMENT_KEY_USAGE"
   KeyUsageProperty="NCRYPT_ALLOW_DECRYPT_FLAG | NCRYPT_ALLOW_SIGNING_FLAG"
   KeyLength=2048
   ```

```
SMIME=FALSE
HashAlgorithm=sha512
[Extensions]
1.3.6.1.4.1.311.21.10 = "{text}"
_continue_ = "OID=1.3.6.1.4.1.311.67.1.1"
2.5.29.37 = "{text}"
_continue_ = "1.3.6.1.4.1.311.67.1.1"
```

3. Open an elevated Command Prompt and use the `certreq.exe` tool to create a new certificate. Use the following command, specifying the full path to the file that was created previously along with the file name:

   Windows Command Prompt

   ```
   certreq.exe -new BitLocker-NetworkUnlock.inf BitLocker-NetworkUnlock.cer
   ```

4. Verify that certificate was properly created by the previous command by confirming that the `.cer` file exists

5. Launch the **Certificates - Local Computer** console by running `certlm.msc`

6. Create a `.pfx` file by following the below steps the **Certificates - Local Computer** console:
   a. Navigate to **Certificates - Local Computer** > **Personal** > **Certificates**
   b. Right-click the previously imported certificate, select **All Tasks**, and then select **Export**
   c. Follow through the wizard to create the `.pfx` file

Deploy the private key and certificate to the WDS server

After creating the certificate and key, deploy them to the infrastructure to properly unlock systems. To deploy the certificates:

1. On the WDS server, launch the **Certificates - Local Computer** console by running `certlm.msc`
2. Right-click **BitLocker Drive Encryption Network Unlock** item under **Certificates (Local Computer)**, select **All Tasks**, and then select **Import**
3. In the **File to Import** dialog, choose the `.pfx` file created previously
4. Enter the password used to create the `.pfx` and complete the wizard

Configure group policy settings for Network Unlock

With certificate and key deployed to the WDS server for Network Unlock, the final step is to use group policy settings to deploy the public key certificate to the desired computers that will

use the Network Unlock key to unlock. Group policy settings for BitLocker can be found under **Computer Configuration** > **Administrative Templates** > **Windows Components** > **BitLocker Drive Encryption** using the Local Group Policy Editor or the Microsoft Management Console.

The following steps describe how to enable the group policy setting that is a requirement for configuring Network Unlock.

1. Open Group Policy Management Console (`gpmc.msc`)
2. Enable the policy **Require additional authentication at startup**, and then select **Require startup PIN with TPM** or **Allow startup PIN with TPM**
3. Turn on BitLocker with TPM+PIN protectors on all domain-joined computers

The following steps describe how to deploy the required group policy setting:

1. Copy the `.cer` file that was created for Network Unlock to the domain controller
2. On the domain controller, open Group Policy Management Console (`gpmc.msc`)
3. Create a new Group Policy Object or modify an existing object to enable the **Allow Network Unlock at startup** setting
4. Deploy the public certificate to clients:

   a. Within group policy management console, navigate to the following location:

   Computer Configuration > **Policies** > **Windows Settings** > **Security Settings** > **Public Key Policies** > **BitLocker Drive Encryption Network Unlock Certificate**.

   b. Right-click the folder and select **Add Network Unlock Certificate**

   c. Follow the wizard steps and import the `.cer` file that was copied earlier

   > ⓘ **Note**
   >
   > Only one Network Unlock certificate can be available at a time. If a new certificate is needed, delete the current certificate before deploying a new one. The Network Unlock certificate is located under the `HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\SystemCertificates\FVE_NKP` registry key on the client computer.

5. Reboot the clients after the Group Policy is deployed

   > ⓘ **Note**

The **Network (Certificate Based)** protector will be added only after a reboot, with the policy enabled and a valid certificate present in the FVE_NKP store.

Subnet policy configuration files on the WDS server (optional)

By default, all clients with the correct Network Unlock certificate and valid Network Unlock protectors that have wired access to a Network Unlock-enabled WDS server via DHCP are unlocked by the server. A subnet policy configuration file on the WDS server can be created to limit which are the subnet(s) the Network Unlock clients can use to unlock.

The configuration file called `bde-network-unlock.ini`, must be located in the same directory as the Network Unlock provider DLL (`%windir%\System32\Nkpprov.dll`) and it applies to both IPv6 and IPv4 DHCP implementations. If the subnet configuration policy becomes corrupted, the provider fails and stops responding to requests.

The subnet policy configuration file must use a `[SUBNETS]` section to identify the specific subnets. The named subnets may then be used to specify restrictions in certificate subsections. Subnets are defined as simple name-value pairs, in the common INI format, where each subnet has its own line, with the name on the left of the equal-sign, and the subnet identified on the right of the equal-sign as a Classless Inter-Domain Routing (CIDR) address or range. The key word `ENABLED` is disallowed for subnet names.

ini

```ini
[SUBNETS]
SUBNET1=10.185.250.0/24 ; a comment about this subrange could be here, after the
semicolon
SUBNET2=10.185.252.200/28
SUBNET3= 2001:4898:a:2::/64 ; an IPv6 subnet
SUBNET4=2001:4898:a:3::/64; in production, the admin would likely give more useful
names, like BUILDING9-EXCEPT-RECEP.
```

Following the `[SUBNETS]` section, there can be sections for each Network Unlock certificate, identified by the certificate thumbprint formatted without any spaces, which define the subnets clients that can be unlocked from that certificate.

> **Note**
>
> When specifying the certificate thumbprint, do not include any spaces. If spaces are included in the thumbprint, the subnet configuration fails because the thumbprint will not be recognized as valid.

Subnet restrictions are defined within each certificate section by denoting the allowed list of permitted subnets. If any subnets are listed in a certificate section, then only those subnets are permitted for that certificate. If no subnet is listed in a certificate section, then all subnets are permitted for that certificate. If a certificate doesn't have a section in the subnet policy configuration file, then no subnet restrictions are applied for unlocking with that certificate. For restrictions to apply to every certificate, there must be a certificate section for every Network Unlock certificate on the server, and an explicit allowed list set for each certificate section.

Subnet lists are created by putting the name of a subnet from the `[SUBNETS]` section on its own line below the certificate section header. Then, the server will only unlock clients with this certificate on the subnet(s) specified as in the list. For troubleshooting, a subnet can be quickly excluded without deleting it from the section by commenting it out with a prepended semi-colon.

ini

```ini
[2158a767e1c14e88e27a4c0aee111d2de2eafe60]
;Comments could be added here to indicate when the cert was issued, which Group Policy should get it, and so on.
;This list shows this cert is allowed to unlock clients only on the SUBNET1 and SUBNET3 subnets. In this example, SUBNET2 is commented out.
SUBNET1
;SUBNET2
SUBNET3
```

To disallow the use of a certificate altogether, add a `DISABLED` line to its subnet list.

Turn off Network Unlock

To turn off the unlock server, the PXE provider can be unregistered from the WDS server or uninstalled altogether. However, to stop clients from creating Network Unlock protectors, the **Allow Network Unlock at startup** group policy setting should be disabled. When this policy setting is updated to **disabled** on client computers, any Network Unlock key protector on the computer is deleted. Alternatively, the BitLocker Network Unlock certificate policy can be deleted on the domain controller to accomplish the same task for an entire domain.

> ⓘ **Note**
>
> Removing the FVE_NKP certificate store that contains the Network Unlock certificate and key on the WDS server will also effectively disable the server's ability to respond to unlock requests for that certificate. However, this is seen as an error condition and is not a supported or recommended method for turning off the Network Unlock server.

Update Network Unlock certificates

To update the certificates used by Network Unlock, administrators need to import or generate the new certificate for the server, and then update the Network Unlock certificate group policy setting on the domain controller.

> **Note**
>
> Servers that don't receive the group policy setting require a PIN when they boot. In such cases, find out why the servers don't receive the GPO to update the certificate.

Troubleshoot Network Unlock

Troubleshooting Network Unlock issues begins by verifying the environment. Many times, a small configuration issue can be the root cause of the failure. Items to verify include:

- Verify that the client hardware is UEFI-based and is on firmware version 2.3.1 and that the UEFI firmware is in native mode without a Compatibility Support Module (CSM) for BIOS mode enabled. Verification can be done by checking that the firmware doesn't have an option enabled such as "Legacy mode" or "Compatibility mode" or that the firmware doesn't appear to be in a BIOS-like mode
- If client hardware is a Secure Core device, you may need to disable Secure Core functionality
- All required roles and services are installed and started
- Public and private certificates have been published and are in the proper certificate containers. The presence of the Network Unlock certificate can be verified in the Microsoft Management Console (MMC.exe) on the WDS server with the certificate snap-ins for the local computer enabled. The client certificate can be verified by checking the registry key `HKEY_LOCAL_MACHINE\Software\Policies\Microsoft\SystemCertificates\FVE_NKP` on the client computer
- Group policy for Network Unlock is enabled and linked to the appropriate domains
- Verify whether group policy is reaching the clients properly. Verification of group policy can be done using the `GPRESULT.exe` or `RSOP.msc` utilities
- Verify whether the clients were rebooted after applying the policy

- Verify whether the **Network (Certificate Based)** protector is listed on the client. Verification of the protector can be done using either manage-bde or Windows PowerShell cmdlets. For example, the following command will list the key protectors currently configured on the C: drive of the local computer:

  PowerShell

  ```
  manage-bde.exe -protectors -get C:
  ```

  > **Note**
  >
  > Use the output of `manage-bde.exe` along with the WDS debug log to determine whether the proper certificate thumbprint is being used for Network Unlock.

Gather the following files to troubleshoot BitLocker Network Unlock.

- The Windows event logs. Specifically, get the BitLocker event logs and the `Microsoft-Windows-Deployment-Services-Diagnostics-Debug` log

  Debug logging is turned off by default for the WDS server role. To retrieve WDS debug logs, the WDS debug logs first need to be enabled. Use either of the following two methods to turn on WDS debug logging.

  - Start an elevated Command Prompt, and then run the following command:

    Windows Command Prompt

    ```
    wevtutil.exe sl Microsoft-Windows-Deployment-Services-Diagnostics/Debug /e:true
    ```

  - Open **Event Viewer** on the WDS server:

    1. In the left pane, navigate to **Applications and Services Logs** > **Microsoft** > **Windows** > **Deployment-Services-Diagnostics** > **Debug**
    2. In the right pane, select **Enable Log**

- The DHCP subnet configuration file (if one exists)
- The output of the BitLocker status on the volume. Gather this output into a text file by using `manage-bde.exe -status`. Or in Windows PowerShell, use `Get-BitLockerVolume`
- The Network Monitor capture on the server that hosts the WDS role, filtered by client IP address

---

Last updated on 07/29/2025

Protect cluster shared volumes and storage area networks with BitLocker

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the procedure to protect cluster shared volumes (CSVs) and storage area networks (SANs) with BitLocker.

BitLocker protects both physical disk resources and cluster shared volumes version 2.0 (CSV2.0). BitLocker on clustered volumes provides an extra layer of protection that can be used to protect sensitive, highly available data. The administrators use this extra layer of protection to increase the security to resources. Only certain user accounts provided access to unlock the BitLocker volume.

Configure BitLocker on cluster shared volumes

Volumes within a cluster are managed with the help of BitLocker based on how the cluster service *sees* the volume to be protected. The volume can be a physical disk resource such as a logical unit number (LUN) on a SAN, or network attached storage (NAS).

> **ⓘ Important**
>
> SANs used with BitLocker must have obtained Windows Hardware Certification. For more information, check Windows Hardware Lab Kit.

The volumes that are designated for a cluster must do the following tasks:

- turn on BitLocker: only after this task is done, the volumes can be added to the storage pool
- must put the resource into maintenance mode before BitLocker operations are completed.

Windows PowerShell or the `manage-bde.exe` command-line tool is the preferred method to manage BitLocker on CSV2.0 volumes. This method is recommended over the BitLocker Control Panel item because CSV2.0 volumes are mount points. Mount points are an NTFS object that is used to provide an entry point to other volumes. Mount points don't require the use of a drive letter. Volumes that lack drive letters don't appear in the BitLocker Control Panel item. Additionally, the new Active Directory-based protector option required for cluster disk resource or CSV2.0 resources isn't available in the Control Panel item.

> ⓘ Note
>
> Mount points can be used to support remote mount points on SMB-based network shares. This type of share is not supported for BitLocker encryption.

If there's a thinly provisioned storage, such as a dynamic virtual hard disk (VHD), BitLocker runs in **Used Disk Space Only** encryption mode. The `manage-bde.exe -WipeFreeSpace` command can't be used to transition the volume to full-volume encryption on thinly provisioned storage volumes. The usage of `manage-bde.exe -WipeFreeSpace` command is blocked to avoid expanding thinly provisioned volumes to occupy the entire backing store while wiping the unoccupied (free) space.

Active Directory-based protector

An Active Directory Domain Services (AD DS) protector can also be used for protecting clustered volumes held within the AD DS infrastructure. The **ADAccountOrGroup** protector is a domain security identifier (SID)-based protector that can be bound to a user account, machine account, or group. When an unlock request is made for a protected volume, the following events take place:

- BitLocker service interrupts the request and uses the BitLocker protect/unprotect APIs to unlock or deny the request.
- BitLocker will unlock protected volumes without user intervention by attempting protectors in the following order:
    1. Clear key
    2. Driver-based auto-unlock key
    3. **ADAccountOrGroup** protector
        a. Service context protector
        b. User protector
    4. Registry-based auto-unlock key

> ⓘ Note
>
> A Windows Server 2012 or later domain controller is required for this feature to work properly.

Turning on BitLocker before adding disks to a cluster using Windows PowerShell

BitLocker encryption is available for disks before these disks are added to a cluster storage pool.

> **Note**
>
> The advantage of The BitLocker encryption can even be made available for disks after they are added to a cluster storage pool. The advantage of encrypting volumes prior to adding them to a cluster is that the disk resource need not be suspended to complete the operation. To turn on BitLocker for a disk before adding it to a cluster:

1. Install the BitLocker Drive Encryption feature if it isn't already installed.
2. Ensure the disk is an NTFS-formatted one and has a drive letter assigned to it.
3. Identify the name of the cluster with Windows PowerShell.

   PowerShell

   ```
   Get-Cluster
   ```

4. Enable BitLocker on a volume with an **ADAccountOrGroup** protector, using the cluster name. For example, use a command such as:

   PowerShell

   ```
   Enable-BitLocker E: -ADAccountOrGroupProtector -ADAccountOrGroup CLUSTER$
   ```

   > **Warning**
   >
   > An **ADAccountOrGroup** protector must be configured using the cluster CNO for a BitLocker enabled volume to either be shared in a Cluster Shared Volume or to fail over properly in a traditional failover cluster.

5. Repeat the preceding steps for each disk in the cluster.
6. Add the volume(s) to the cluster.

Turning on BitLocker for a clustered disk using Windows PowerShell

When the cluster service owns a disk resource already, the disk resource needs to be set into maintenance mode before BitLocker can be enabled. To turn on the BitLocker for a clustered disk using Windows PowerShell, perform the following steps:

1. Install the BitLocker drive encryption feature if it isn't already installed.
2. Check the status of the cluster disk using Windows PowerShell.

   ```PowerShell
   Get-ClusterResource "Cluster Disk 1"
   ```

3. Put the physical disk resource into maintenance mode using Windows PowerShell.

   ```PowerShell
   Get-ClusterResource "Cluster Disk 1" | Suspend-ClusterResource
   ```

4. Identify the name of the cluster with Windows PowerShell.

   ```PowerShell
   Get-Cluster
   ```

5. Enable BitLocker a volume with an **ADAccountOrGroup** protector, using the cluster name. For example, use a command such as:

   ```PowerShell
   Enable-BitLocker E: -ADAccountOrGroupProtector -ADAccountOrGroup CLUSTER$
   ```

   > ⚠ **Warning**
   >
   > An **ADAccountOrGroup** protector must be configured using the cluster CNO for a BitLocker-enabled volume to either be shared in a cluster-shared Volume or to fail over properly in a traditional failover cluster.

6. Use **Resume-ClusterResource** to take back the physical disk resource out of maintenance mode:

PowerShell

```
Get-ClusterResource "Cluster Disk 1" | Resume-ClusterResource
```

7. Repeat the preceding steps for each disk in the cluster.

Adding BitLocker-encrypted volumes to a cluster using `manage-bde.exe`

`Manage-bde.exe` can also be used to enable BitLocker on clustered volumes. The steps needed to add a physical disk resource or CSV2.0 volume to an existing cluster are:

1. Verify that the BitLocker drive encryption feature is installed on the computer.
2. Ensure new storage is formatted as NTFS.
3. Encrypt the volume, add a recovery key and add the cluster administrator as a protector key using `manage-bde.exe` in a Command Prompt window. For example:

   Windows Command Prompt

   ```
   manage-bde.exe -on -used <drive letter> -RP -sid domain\CNO$ -sync
   ```

   a. BitLocker will check to see if the disk is already part of a cluster. If it is, administrators will encounter a hard block. Otherwise, the encryption continues.
   b. Using the -sync parameter is optional. However, using the -sync parameter has the advantage of ensuring the command waits until the encryption for the volume is completed. The volume is then released for use in the cluster storage pool.

4. Open the Failover Cluster Manager snap-in or cluster PowerShell cmdlets to enable the disk to be clustered.

   - Once the disk is clustered, it's enabled for CSV.

5. During the resource online operation, cluster checks whether the disk is BitLocker encrypted.
   a. If the volume isn't BitLocker enabled, traditional cluster online operations occur.
   b. If the volume is BitLocker enabled, BitLocker checks if the volume is **locked**. If the volume is **locked**, BitLocker impersonates the CNO and unlocks the volume using the CNO protector. If these actions by BitLocker fail, an event is logged. The logged event will state that the volume couldn't be unlocked and the online operation has failed.
6. Once the disk is online in the storage pool, it can be added to a CSV by right-clicking the disk resource, and choosing "**Add to cluster shared volumes**".

CSVs include both encrypted and unencrypted volumes. To check the status of a particular volume for BitLocker encryption run the `manage-bde.exe -status` command as an administrator with a path to the volume. The path must be one that is inside the CSV namespace. For example:

Windows Command Prompt

```
manage-bde.exe -status "C:\ClusterStorage\volume1"
```

Physical disk resources

Unlike CSV2.0 volumes, physical disk resources can only be accessed by one cluster node at a time. This condition means that operations such as encrypting, decrypting, locking, or unlocking volumes require a context to perform. For example, a physical disk resource can't unlock or decrypt if it isn't administering the cluster node that owns the disk resource because the disk resource isn't available.

Restrictions on BitLocker actions with cluster volumes

The following table contains information about both physical disk resources (that is, traditional failover cluster volumes) and cluster shared volumes (CSV) and the actions that are allowed by BitLocker in each situation.

Expand table

Action	On owner node of failover volume	On Metadata Server (MDS) of CSV	On (Data Server) DS of CSV	Maintenance Mode
`Manage-bde.exe -on`	Blocked	Blocked	Blocked	Allowed
`Manage-bde.exe -off`	Blocked	Blocked	Blocked	Allowed
`Manage-bde.exe Pause/Resume`	Blocked	Blocked**	Blocked	Allowed
`Manage-bde.exe -lock`	Blocked	Blocked	Blocked	Allowed
`Manage-bde.exe -wipe`	Blocked	Blocked	Blocked	Allowed

Action	On owner node of failover volume	On Metadata Server (MDS) of CSV	On (Data Server) DS of CSV	Maintenance Mode
Unlock	Automatic via cluster service	Automatic via cluster service	Automatic via cluster service	Allowed
`Manage-bde.exe -protector -add`	Allowed	Allowed	Blocked	Allowed
`Manage-bde.exe -protector -delete`	Allowed	Allowed	Blocked	Allowed
`Manage-bde.exe -autounlock`	Allowed (not recommended)	Allowed (not recommended)	Blocked	Allowed (not recommended)
`Manage-bde.exe -upgrade`	Allowed	Allowed	Blocked	Allowed
Shrink	Allowed	Allowed	Blocked	Allowed
Extend	Allowed	Allowed	Blocked	Allowed

> ⓘ Note
>
> Although the `manage-bde.exe -pause` command is blocked in clusters, the cluster service automatically resumes a paused encryption or decryption from the MDS node.

In the case where a physical disk resource experiences a failover event during conversion, the new owning node detects that the conversion isn't complete and completes the conversion process.

Other considerations when using BitLocker on CSV2.0

Some other considerations to take into account for BitLocker on clustered storage include:

- BitLocker volumes have to be initialized and begin encryption before they're available to add to a CSV2.0 volume
- If an administrator needs to decrypt a CSV volume, remove the volume from the cluster or put it into disk maintenance mode. The CSV can be added back to the cluster while waiting for decryption to complete
- If an administrator needs to start encrypting a CSV volume, remove the volume from the cluster or put it into maintenance mode
- If conversion is paused with encryption in progress and the CSV volume is offline from the cluster, the cluster thread (health check) automatically resumes conversion when the

volume is online to the cluster

- If conversion is paused with encryption in progress and a physical disk resource volume is offline from the cluster, the BitLocker driver automatically resumes conversion when the volume is online to the cluster
- If conversion is paused with encryption in progress, while the CSV volume is in maintenance mode, the cluster thread (health check) automatically resumes conversion when moving the volume back from maintenance
- If conversion is paused with encryption in progress, while the disk resource volume is in maintenance mode, the BitLocker driver automatically resumes conversion when the volume is moved back from maintenance mode

---

Last updated on 07/29/2025

Boot Configuration Data settings and BitLocker

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the Boot Configuration Data (BCD) settings that are used by BitLocker.

During the boot process, BitLocker verifies that the security sensitive BCD settings haven't changed since BitLocker was last enabled, resumed, or recovered.

If it's believed that there's a risk in excluding a particular BCD setting from the validation profile, you can include that BCD setting in the BCD validation coverage to suit the preferences for validation.
If the default BCD setting persistently triggers a recovery for benign changes, you can exclude that BCD setting from the validation coverage.

> ⓘ **Important**
>
> Devices with UEFI firmware can use secure boot to provide enhanced boot security. When BitLocker is able to use secure boot for platform and BCD integrity validation, as defined by the Allow Secure Boot for integrity validation policy setting, the Use enhanced Boot Configuration Data validation profile policy is ignored.

One of the benefits of using secure boot is that it can correct BCD settings during boot without triggering recovery events. Secure boot enforces the same BCD settings as BitLocker. Secure boot BCD enforcement isn't configurable from within the operating system.

Customize BCD validation settings

To modify the BCD settings that are validated by BitLocker, the administrator will add or exclude BCD settings from the platform validation profile by enabling and configuring the **Use enhanced Boot Configuration Data validation profile** policy setting.

For the purposes of BitLocker validation, BCD settings are associated with a specific set of Microsoft boot applications. These BCD settings can also be applied to the other Microsoft boot applications that aren't part of the set to which the BCD settings are already applicable for. This setting can be done by attaching any of the following prefixes to the BCD settings that are being entered in the group policy settings dialog:

- winload

- winresume
- memtest
- all of the above

All BCD settings are specified by combining the prefix value with either a hexadecimal (hex) value or a *friendly name*.

The BCD setting hex value is reported when BitLocker enters recovery mode and is stored in the event log (event ID 523). The hex value uniquely identifies the BCD setting that caused the recovery event.

You can quickly obtain the friendly name for the BCD settings on a computer by using the command `bcdedit.exe /enum all`.

Not all BCD settings have friendly names. For those settings without a friendly name, the hex value is the only way to configure an exclusion policy.

When specifying BCD values in the **Use enhanced Boot Configuration Data validation profile** policy setting, use the following syntax:

- Prefix the setting with the boot application prefix
- Append a colon `:`
- Append either the hex value or the friendly name
- If entering more than one BCD setting, each BCD setting will need to be entered on a new line

For example, either "`winload:hypervisordebugport`" or "`winload:0x250000f4`" yields the same value.

A setting that applies to all boot applications may be applied only to an individual application. However, the reverse isn't true. For example, one can specify either "`all:locale`" or "`winresume:locale`", but as the BCD setting "`win-pe`" doesn't apply to all boot applications, "`winload:winpe`" is valid, but "`all:winpe`" isn't valid. The setting that controls boot debugging ("`bootdebug`" or 0x16000010) will always be validated and will have no effect if it's included in the provided fields.

> **Note**
>
> Take care when configuring BCD entries in the policy setting. The Local Group Policy Editor doesn't validate the correctness of the BCD entry. BitLocker fails to be enabled if the policy setting specified is invalid.

Default BCD validation profile

The following table contains the default BCD validation profile used by BitLocker:

Expand table

Hex Value	Prefix	Friendly Name
0x11000001	all	device
0x12000002	all	path
0x12000030	all	loadoptions
0x16000010	all	bootdebug
0x16000040	all	advancedoptions
0x16000041	all	optionsedit
0x16000048	all	nointegritychecks
0x16000049	all	testsigning
0x16000060	all	isolatedcontext
0x1600007b	all	forcefipscrypto
0x22000002	winload	systemroot
0x22000011	winload	kernel
0x22000012	winload	hal
0x22000053	winload	evstore
0x25000020	winload	nx
0x25000052	winload	restrictapiccluster
0x26000022	winload	winpe
0x26000025	winload	lastknowngood
0x26000081	winload	safebootalternateshell
0x260000a0	winload	debug
0x260000f2	winload	hypervisordebug
0x26000116	winload	hypervisorusevapic

Hex Value	Prefix	Friendly Name
0x21000001	winresume	filedevice
0x22000002	winresume	filepath
0x26000006	winresume	debugoptionenabled

Full list of friendly names for ignored BCD settings

The following list is a full list of BCD settings with friendly names, which are ignored by default. These settings aren't part of the default BitLocker validation profile, but can be added if you see a need to validate any of these settings before allowing a BitLocker-protected operating system drive to be unlocked.

> **Note**
>
> Additional BCD settings exist that have hex values but do not have friendly names. These settings are not included in this list.

Expand table

Hex Value	Prefix	Friendly Name
0x12000004	all	description
0x12000005	all	locale
0x12000016	all	targetname
0x12000019	all	busparams
0x1200001d	all	key
0x1200004a	all	fontpath
0x14000006	all	inherit
0x14000008	all	recoverysequence
0x15000007	all	truncatememory
0x1500000c	all	firstmegabytepolicy
0x1500000d	all	relocatephysical
0x1500000e	all	avoidlowmemory

Hex Value	Prefix	Friendly Name
0x15000011	all	debugtype
0x15000012	all	debugaddress
0x15000013	all	debugport
0x15000014	all	baudrate
0x15000015	all	channel
0x15000018	all	debugstart
0x1500001a	all	hostip
0x1500001b	all	port
0x15000022	all	emsport
0x15000023	all	emsbaudrate
0x15000042	all	keyringaddress
0x15000047	all	configaccesspolicy
0x1500004b	all	integrityservices
0x1500004c	all	volumebandid
0x15000051	all	initialconsoleinput
0x15000052	all	graphicsresolution
0x15000065	all	displaymessage
0x15000066	all	displaymessageoverride
0x15000081	all	logcontrol
0x16000009	all	recoveryenabled
0x1600000b	all	badmemoryaccess
0x1600000f	all	traditionalkseg
0x16000017	all	noumex
0x1600001c	all	dhcp
0x1600001e	all	vm
0x16000020	all	bootems

Hex Value	Prefix	Friendly Name
0x16000046	all	graphicsmodedisabled
0x16000050	all	extendedinput
0x16000053	all	restartonfailure
0x16000054	all	highestmode
0x1600006c	all	bootuxdisabled
0x16000072	all	nokeyboard
0x16000074	all	bootshutdowndisabled
0x1700000a	all	badmemorylist
0x17000077	all	allowedinmemorysettings
0x22000040	all	fverecoveryurl
0x22000041	all	fverecoverymessage
0x31000003	all	ramdisksdidevice
0x32000004	all	ramdisksdipath
0x35000001	all	ramdiskimageoffset
0x35000002	all	ramdisktftpclientport
0x35000005	all	ramdiskimagelength
0x35000007	all	ramdisktftpblocksize
0x35000008	all	ramdisktftpwindowsize
0x36000006	all	exportascd
0x36000009	all	ramdiskmcenabled
0x3600000a	all	ramdiskmctftpfallback
0x3600000b	all	ramdisktftpvarwindow
0x21000001	winload	osdevice
0x22000013	winload	dbgtransport
0x220000f9	winload	hypervisorbusparams
0x22000110	winload	hypervisorusekey

Hex Value	Prefix	Friendly Name
0x23000003	winload	resumeobject
0x25000021	winload	pae
0x25000031	winload	removememory
0x25000032	winload	increaseuserva
0x25000033	winload	perfmem
0x25000050	winload	clustermodeaddressing
0x25000055	winload	x2apicpolicy
0x25000061	winload	numproc
0x25000063	winload	configflags
0x25000066	winload	groupsize
0x25000071	winload	msi
0x25000072	winload	pciexpress
0x25000080	winload	safeboot
0x250000a6	winload	tscsyncpolicy
0x250000c1	winload	driverloadfailurepolicy
0x250000c2	winload	bootmenupolicy
0x250000e0	winload	bootstatuspolicy
0x250000f0	winload	hypervisorlaunchtype
0x250000f3	winload	hypervisordebugtype
0x250000f4	winload	hypervisordebugport
0x250000f5	winload	hypervisorbaudrate
0x250000f6	winload	hypervisorchannel
0x250000f7	winload	bootux
0x250000fa	winload	hypervisornumproc
0x250000fb	winload	hypervisorrootprocpernode
0x250000fd	winload	hypervisorhostip

Hex Value	Prefix	Friendly Name
0x250000fe	winload	hypervisorhostport
0x25000100	winload	tpmbootentropy
0x25000113	winload	hypervisorrootproc
0x25000115	winload	hypervisoriommupolicy
0x25000120	winload	xsavepolicy
0x25000121	winload	xsaveaddfeature0
0x25000122	winload	xsaveaddfeature1
0x25000123	winload	xsaveaddfeature2
0x25000124	winload	xsaveaddfeature3
0x25000125	winload	xsaveaddfeature4
0x25000126	winload	xsaveaddfeature5
0x25000127	winload	xsaveaddfeature6
0x25000128	winload	xsaveaddfeature7
0x25000129	winload	xsaveremovefeature
0x2500012a	winload	xsaveprocessorsmask
0x2500012b	winload	xsavedisable
0x25000130	winload	claimedtpmcounter
0x26000004	winload	stampdisks
0x26000010	winload	detecthal
0x26000024	winload	nocrashautoreboot
0x26000030	winload	nolowmem
0x26000040	winload	vga
0x26000041	winload	quietboot
0x26000042	winload	novesa
0x26000043	winload	novga
0x26000051	winload	usephysicaldestination

Hex Value	Prefix	Friendly Name
0x26000054	winload	uselegacyapicmode
0x26000060	winload	onecpu
0x26000062	winload	maxproc
0x26000064	winload	maxgroup
0x26000065	winload	groupaware
0x26000070	winload	usefirmwarepcisettings
0x26000090	winload	bootlog
0x26000091	winload	sos
0x260000a1	winload	halbreakpoint
0x260000a2	winload	useplatformclock
0x260000a3	winload	forcelegacyplatform
0x260000a4	winload	useplatformtick
0x260000a5	winload	disabledynamictick
0x260000b0	winload	ems
0x260000c3	winload	onetimeadvancedoptions
0x260000c4	winload	onetimeoptionsedit
0x260000e1	winload	disableelamdrivers
0x260000f8	winload	hypervisordisableslat
0x260000fc	winload	hypervisoruselargevtlb
0x26000114	winload	hypervisordhcp
0x21000005	winresume	associatedosdevice
0x25000007	winresume	bootux
0x25000008	winresume	bootmenupolicy
0x26000003	winresume	customsettings
0x26000004	winresume	pae
0x25000001	memtest	passcount

Hex Value	Prefix	Friendly Name
0x25000002	memtest	testmix
0x25000005	memtest	stridefailcount
0x25000006	memtest	invcfailcount
0x25000007	memtest	matsfailcount
0x25000008	memtest	randfailcount
0x25000009	memtest	chckrfailcount
0x26000003	memtest	cacheenable
0x26000004	memtest	failuresenabled

Last updated on 07/29/2025

BitLocker FAQ

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Learn more about BitLocker by reviewing the frequently asked questions.

Overview and requirements

Does BitLocker support multifactor authentication?

Yes, BitLocker supports multifactor authentication for operating system drives. If BitLocker is enabled on a computer that has a TPM version 1.2 or later, additional forms of authentication can be used with the TPM protection.

Why are two partitions required?

Two partitions are required to run BitLocker because pre-startup authentication and system integrity verification must occur on a separate partition from the encrypted operating system drive. This configuration helps protect the operating system and the information in the encrypted drive.

How can I tell if a computer has a TPM?

The TPM status can be checked in **Windows Defender Security Center** > **Device Security** > **Security processor details**.

Can I use BitLocker on an operating system drive without a TPM?

Yes, BitLocker can be enabled on an operating system drive without a TPM, if the BIOS or UEFI firmware has the ability to read from a USB flash drive in the boot environment. BitLocker won't unlock the protected drive until BitLocker's own volume master key is first released by either the computer's TPM or by a USB flash drive containing the BitLocker startup key for that computer. However, computers without TPMs won't be able to use the system integrity verification that BitLocker can also provide. To help determine whether a computer can read from a USB device during the boot process, use the BitLocker system check as part of the BitLocker setup process. This system check performs tests to confirm that the computer can properly read from the USB devices at the appropriate time and that the computer meets other BitLocker requirements.

How do I obtain BIOS support for the TPM on my computer?

Contact the computer manufacturer to request a Trusted Computing Group (TCG)-compliant BIOS or UEFI boot firmware that meets the following requirements:

- It's compliant with the TCG standards for a client computer.
- It has a secure update mechanism to help prevent a malicious BIOS or boot firmware from being installed on the computer.

What user rights are required to use BitLocker?

To turn on, turn off, or change configurations of BitLocker on operating system and fixed data drives, membership to the local *Administrators* group is required. Standard users can turn on, turn off, or change configurations of BitLocker on removable data drives.

What is the recommended boot order for computers that are going to be BitLocker-protected?

The computer's startup options should be configured to have the hard disk drive first in the boot order, before any other drives such as CD/DVD drives or USB drives. If the hard disk isn't first and the computer typically boots from the hard disk, then a boot order change may be detected or assumed when removable media is found during boot. The boot order typically affects the system measurement that is verified by BitLocker and a change in boot order will cause a prompt for the BitLocker recovery key. For the same reason, if a laptop is used with a docking station, ensure that the hard disk drive is first in the boot order both when the laptop is docked and undocked.

BitLocker and Windows upgrade

Can I upgrade Windows versions with BitLocker enabled?

Yes.

What is the difference between suspending and decrypting BitLocker?

Decrypt completely removes BitLocker protection and fully decrypts the drive.

Suspend keeps the data encrypted but encrypts the BitLocker volume master key with a clear key. The clear key is a cryptographic key stored unencrypted and unprotected on the disk drive. By storing this key unencrypted, the *Suspend* option allows for changes or upgrades to the computer without the time and cost of decrypting and re-encrypting the entire drive. After the changes are made and BitLocker is again enabled, BitLocker will reseal the encryption key to the new values of the measured components that changed as a part of the upgrade, the volume master key is changed, the protectors are updated to match and the clear key is erased.

Do I have to suspend BitLocker protection to download and install system updates and upgrades?

No user action is required for BitLocker in order to apply updates from Microsoft, including Windows quality updates and feature updates. Users need to suspend BitLocker for Non-Microsoft software updates, such as:

- Some TPM firmware updates if these updates clear the TPM outside of the Windows API. Not every TPM firmware update will clear the TPM. Users don't have to suspend BitLocker if the TPM firmware update uses Windows API to clear the TPM because in this case, BitLocker will be automatically suspended. It's recommended that users test their TPM firmware updates if they don't want to suspend BitLocker protection.
- Non-Microsoft application updates that modify the UEFI\BIOS configuration.
- Manual or non-Microsoft updates to secure boot databases (only if BitLocker uses Secure Boot for integrity validation).
- Updates to UEFI\BIOS firmware, installation of additional UEFI drivers, or UEFI applications without using the Windows update mechanism (only if BitLocker doesn't use Secure Boot for integrity validation during updates).
- BitLocker can be checked if it uses Secure Boot for integrity validation with the command line `manage-bde.exe -protectors -get C:`. If Secure Boot for integrity validation is being used, it reports **Uses Secure Boot for integrity validation**.

> **Note**
>
> If BitLocker is suspended, you can resume BitLocker protection after the upgrade or update is installed. Upon resuming protection, BitLocker will reseal the encryption key to the new values of the measured components that changed as a part of the upgrade or update. If these types of upgrades or updates are applied without suspending BitLocker, the computer will enter recovery mode when restarting and will require a recovery key or password to access the computer.

BitLocker behavior on Entra ID-joined devices:

- When BitLocker is suspended automatically or without the `-rebootcount` parameter, Windows attempts to resume protection at the next reboot. A network connection is required to back up the recovery password to Microsoft Entra ID. If a policy setting requires BitLocker key backup, Windows waits up to 60 seconds for a network connection. If the recovery key can't be backed up, protection won't resume.
- If BitLocker is suspended as part of a built-in upgrade or PC reset flow, Windows attempts to export the recovery key and resumes protection regardless of whether the export succeeds.

Deployment and administration

Can BitLocker deployment be automated in an enterprise environment?

Yes, the deployment and configuration BitLocker can be automated using either Windows PowerShell or with the `manage-bde.exe` command. For more information about common BitLocker management commands, check the BitLocker operations guide.

Is there a noticeable performance impact when BitLocker is enabled on a computer?

Typically, there's a small performance overhead, often in single-digit percentages, which is relative to the throughput of the storage operations on which it needs to operate.

How long will initial encryption take when BitLocker is turned on?

Although BitLocker encryption occurs in the background while a user continues to work with the system remaining usable, encryption times vary depending on the type of drive that is being encrypted, the size of the drive, and the speed of the drive. If encrypting large drives, encryption may want to be scheduled during times when the drive isn't being used.

When BitLocker is enabled, BitLocker can also be set to encrypt the entire drive or just the used space on the drive. On a new hard drive, encrypting just the used spaced can be considerably faster than encrypting the entire drive. When this encryption option is selected, BitLocker automatically encrypts data as it is saved, ensuring that no data is stored unencrypted.

What happens if the computer is turned off during encryption or decryption?

If the computer is turned off or goes into hibernation, the BitLocker encryption and decryption process will resume where it stopped the next time Windows starts. BitLocker resuming encryption or decryption is true even if the power is suddenly unavailable.

Does BitLocker encrypt and decrypt the entire drive all at once when reading and writing data?

No, BitLocker doesn't encrypt and decrypt the entire drive when reading and writing data. The encrypted sectors in the BitLocker-protected drive are decrypted only as they're requested from system read operations. Blocks that are written to the drive are encrypted before the system writes them to the physical disk. No unencrypted data is ever stored on a BitLocker-protected drive.

How can I prevent users from storing data on an unencrypted drive?

Policy settings can be configured to require that data drives be BitLocker-protected before a BitLocker-protected computer can write data to them. For more info, see BitLocker policy settings. When these policy settings are enabled, the BitLocker-protected operating system will mount any data drives that aren't protected by BitLocker as read-only.

What is Used Disk Space Only encryption?

BitLocker lets users choose to encrypt just their data. Although it's not the most secure way to encrypt a drive, this option can reduce encryption time by more than 99 percent, depending on how much data that needs to be encrypted. For more information, see Used Disk Space Only encryption.

What system changes would cause the integrity check on the OS drive to fail?

The following types of system changes can cause an integrity check failure and prevent the TPM from releasing the BitLocker key to decrypt the protected operating system drive:

- Moving the BitLocker-protected drive into a new computer.
- Installing a new motherboard with a new TPM.

- Turning off, disabling, or clearing the TPM.
- Changing any boot configuration settings.
- Changing the BIOS, UEFI firmware, master boot record, boot sector, boot manager, option ROM, or other early boot components or boot configuration data.

What causes BitLocker to start into recovery mode when attempting to start the operating system drive?

Because BitLocker is designed to protect computers from numerous attacks, there are numerous reasons why BitLocker could start in recovery mode. For example:

- Changing the BIOS boot order to boot another drive in advance of the hard drive.
- Adding or removing hardware, such as inserting a new card in the computer.
- Removing, inserting, or completely depleting the charge on a smart battery on a portable computer.

In BitLocker, recovery consists of decrypting a copy of the volume master key using either a recovery key stored on a USB flash drive or a cryptographic key derived from a recovery password. The TPM isn't involved in any recovery scenarios, so recovery is still possible if the TPM fails boot component validation, malfunctions, or is removed.

What can prevent BitLocker from binding to PCR 7?

BitLocker can be prevented from binding to PCR 7 if a non-Windows OS booted prior to Windows, or if Secure Boot isn't available to the device, either because it's disabled or the hardware doesn't support it.

Can I swap hard disks on the same computer if BitLocker is enabled on the operating system drive?

Yes, multiple hard disks can be swapped on the same computer if BitLocker is enabled, but only if the hard disks were BitLocker-protected on the same computer. The BitLocker keys are unique to the TPM and the operating system drive. If a backup operating system or data drive needs to be prepared in case of a disk failure, make sure that they were matched with the correct TPM. Different hard drives can also be configured for different operating systems and then enable BitLocker on each one with different authentication methods (such as one with TPM-only and one with TPM+PIN) without any conflicts.

Can I access my BitLocker-protected drive if I insert the hard disk into a different computer?

Yes, if the drive is a data drive, it can be unlocked from the **BitLocker Drive Encryption** Control Panel item by using a password or smart card. If the data drive was configured for automatic unlock only, it will need to be unlocked by using the recovery key. The encrypted hard disk can be unlocked by a data recovery agent (if one was configured) or it can be unlocked by using the recovery key.

Why isn't the "Turn BitLocker on" option available when I right-click a drive?

Some drives can't be encrypted with BitLocker. Reasons a drive can't be encrypted include insufficient disk size, an incompatible file system, if the drive is a dynamic disk, or a drive is designated as the system partition. By default, the system drive (or system partition) is hidden from display. However, if it isn't created as a hidden drive when the operating system was installed due to a custom installation process, that drive might be displayed but can't be encrypted.

What type of disk configurations are supported by BitLocker?

Any number of internal, fixed data drives can be protected with BitLocker. On some versions ATA and SATA-based, direct-attached storage devices are also supported.

Key Management

How can I authenticate or unlock my removable data drive?

Removable data drives can be unlocked using a password or a smart card. A SID protector can also be configured to unlock a drive by using user domain credentials. After encryption has started, the drive can also be automatically unlocked on a specific computer for a specific user account. System administrators can configure which options are available for users including password complexity and minimum length requirements. To unlock by using a SID protector, use `manage-bde.exe`:

Windows Command Prompt

```
Manage-bde.exe -protectors -add e: -sid domain\username
```

What is the difference between a TPM owner password, recovery password, recovery key, PIN, enhanced PIN, and startup key?

There are multiple keys that can be generated and used by BitLocker. Some keys are required and some are optional protectors you can choose to use depending on the level of security you require.

TPM owner password

Prior to enabling BitLocker on a computer with a TPM version 1.2, you must initialize the TPM. The initialization process generates a TPM owner password, which is a password set on the TPM. You must be able to supply the TPM owner password to change the state of the TPM, such as when enabling or disabling the TPM or resetting the TPM lockout.

Recovery password and recovery key

When you set up BitLocker, you must choose how access to BitLocker-protected drives can be recovered in the event that the specified unlock method cannot be used (such as if the TPM cannot validate the boot components, the personal identification number (PIN) is forgotten, or the password is forgotten). In these situations, you must be able to supply either the recovery key or the recovery password to unlock the encrypted data on the drive. When you supply the recovery information, you can use either of the following formats:

- A recovery password consisting of 48 digits divided into eight groups. During recovery, you need to type this password into the BitLocker recovery console by using the function keys on your keyboard.
- A key file on a USB flash drive that is read directly by the BitLocker recovery console. During recovery, you need to insert this USB device.

PIN and enhanced PIN

For a higher level of security with the TPM, you can configure BitLocker with a personal identification number (PIN). The PIN is a user-created value that must be entered each time the computer starts or resumes from hibernation. The PIN can consist of 4 to 20 digits as specified by the *Configure minimum PIN length for startup* policy setting and is stored internally as a 256-bit hash of the entered Unicode characters. This value is never displayed to the user. The PIN is used to provide another factor of authentication in conjunction with TPM authentication. For an even higher level of security with the TPM, you can configure BitLocker to use enhanced PINs. Enhanced PINs are PINs that use the full keyboard character set in addition to the

numeric set to allow for more possible PIN combinations and are between 4 and 20 characters in length. To use enhanced PINs, you must enable the *Allow enhanced PINs for startup* policy setting before adding the PIN to the drive. By enabling this policy, all PINs created can utilize full keyboard characters.

Startup key

Configuring a startup key is another method to enable a higher level of security with the TPM. The startup key is a key stored on a USB flash drive, and the USB flash drive must be inserted every time the computer starts. The startup key is used to provide another factor of authentication in conjunction with TPM authentication. To use a USB flash drive as a startup key, the USB flash drive must be formatted by using the NTFS, FAT, or FAT32 file system.

> **ⓘ Important**
>
> You must have a startup key to use BitLocker on a non-TPM computer.

How can the recovery password and recovery key be stored?

The recovery password and recovery key for an operating system drive or a fixed data drive can be saved to a folder, saved to one or more USB devices, saved to a Microsoft Account, or printed.

For removable data drives, the recovery password and recovery key can be saved to a folder, saved to a Microsoft Account, or printed. By default, a recovery key for a removable drive can't be stored on a removable drive.

A domain administrator can also configure policy settings to automatically generate recovery passwords and store them in Active Directory Domain Services (AD DS) or Microsoft Entra ID for any BitLocker-protected drive.

Is it possible to add an additional method of authentication without decrypting the drive if I only have the TPM authentication method enabled?

The `Manage-bde.exe` command-line tool can be used to replace TPM-only authentication mode with a multifactor authentication mode. For example, if BitLocker is enabled with TPM authentication only and PIN authentication needs to be added, use the following commands

from an elevated Command Prompt, replacing *4-20 digit numeric PIN* with the desired numeric PIN:

Windows Command Prompt

```
manage-bde.exe -protectors -delete %systemdrive% -type tpm

manage-bde.exe -protectors -add %systemdrive% -tpmandpin <4-20 digit numeric PIN>
```

When should an additional method of authentication be considered?

New hardware that meets Windows Hardware Compatibility Program requirements make a PIN less critical as a mitigation, and having a TPM-only protector is likely sufficient when combined with policies like device lockout. For example, Surface Pro and Surface Book don't have external DMA ports to attack. For older hardware, where a PIN may be needed, it's recommended to enable enhanced PINs that allow non-numeric characters such as letters and punctuation marks, and to set the PIN length based on the risk tolerance and the hardware anti-hammering capabilities available to the TPMs on the computers.

If I lose my recovery information, will the BitLocker-protected data be unrecoverable?

BitLocker is designed to make the encrypted drive unrecoverable without the required authentication. When in recovery mode, the user needs the recovery password or recovery key to unlock the encrypted drive.

> **Important**
>
> Store the recovery information in Microsoft Entra ID, AD DS, Microsoft Account, or another safe location.

Can the USB flash drive that is used as the startup key also be used to store the recovery key?

While using a USB flash drive as both the startup key and for storage of the recovery key is technically possible, it isn't a best practice to use one USB flash drive to store both keys. If the USB flash drive that contains the startup key is lost or stolen, the recovery key will also be lost. In addition, inserting this key would cause the computer to automatically boot from the

recovery key even if TPM-measured files have changed, which circumvents the TPM's system integrity check.

Can I save the startup key on multiple USB flash drives?

Yes, computer's startup key can be saved on multiple USB flash drives. Right-clicking a BitLocker-protected drive and selecting **Manage BitLocker** will provide the options to save the recovery keys on additional USB flash drives as needed.

Can I save multiple (different) startup keys on the same USB flash drive?

Yes, BitLocker startup keys for different computers can be saved on the same USB flash drive.

Can I generate multiple (different) startup keys for the same computer?

Generating different startup keys for the same computer can be done through scripting. However, for computers that have a TPM, creating different startup keys prevents BitLocker from using the TPM's system integrity check.

Can I generate multiple PIN combinations?

Generating multiple PIN combinations can't be done.

What encryption keys are used in BitLocker? How do they work together?

Raw data is encrypted with the full volume encryption key, which is then encrypted with the volume master key. The volume master key is in turn encrypted by one of several possible methods depending on the authentication (that is, key protectors or TPM) and recovery scenarios.

Where are the encryption keys stored?

The full volume encryption key is encrypted by the volume master key and stored in the encrypted drive. The volume master key is encrypted by the appropriate key protector and stored in the encrypted drive. If BitLocker has been suspended, the clear key that is used to

encrypt the volume master key is also stored in the encrypted drive, along with the encrypted volume master key.

This storage process ensures that the volume master key is never stored unencrypted and is protected unless BitLocker is disabled. The keys are also saved to two additional locations on the drive for redundancy. The keys can be read and processed by the boot manager.

Why do I have to use the function keys to enter the PIN or the 48-character recovery password?

The F1 through F10 keys are universally mapped scan codes available in the pre-boot environment on all computers and in all languages. The numeric keys 0 through 9 aren't usable in the pre-boot environment on all keyboards.

When using an enhanced PIN, users should run the optional system check during the BitLocker setup process to ensure that the PIN can be entered correctly in the pre-boot environment.

How does BitLocker help prevent an attacker from discovering the PIN that unlocks my operating system drive?

It's possible that a personal identification number (PIN) can be discovered by an attacker performing a brute force attack. A brute force attack occurs when an attacker uses an automated tool to try different PIN combinations until the correct one is discovered. For BitLocker-protected computers, this type of attack, also known as a dictionary attack, requires that the attacker has physical access to the computer.

The TPM has the built-in ability to detect and react to these types of attacks. Because different manufacturers' TPMs may support different PIN and attack mitigations, contact the TPM's manufacturer to determine how the computer's TPM mitigates PIN brute force attacks. After the TPM's manufacturer is determined, contact the manufacturer to gather the TPM's vendor-specific information. Most manufacturers use the PIN authentication failure count to exponentially increase lockout time to the PIN interface. However, each manufacturer has different policies regarding when and how the failure counter is decreased or reset.

How can I determine the manufacturer of my TPM?

The TPM manufacturer can be determined in **Windows Defender Security Center** > **Device Security** > **Security processor details**.

How can I evaluate a TPM's dictionary attack mitigation mechanism?

The following questions can assist when asking a TPM manufacturer about the design of a dictionary attack mitigation mechanism:

- How many failed authorization attempts can occur before lockout?
- What is the algorithm for determining the duration of a lockout based on the number of failed attempts and any other relevant parameters?
- What actions can cause the failure count and lockout duration to be decreased or reset?

Can PIN length and complexity be managed with policy settings?

The minimum personal identification number (PIN) length can be configured by using the **Configure minimum PIN length for startup** Group Policy setting and allow the use of alphanumeric PINs by enabling the **Allow enhanced PINs for startup** policy setting. PIN complexity can't be required via policy settings.

For more info, see BitLocker policy settings.

How are the PIN and TPM used to derive the volume master key?

BitLocker hashes the user-specified personal identification number (PIN) by using SHA-256, and the first 160 bits of the hash are used as authorization data sent to the TPM to seal the volume master key. The volume master key is now protected by both the TPM and the PIN. To unseal the volume master key, you are required to enter the PIN each time the computer restarts or resumes from hibernation.

BitLocker To Go

What is BitLocker To Go?

BitLocker To Go is BitLocker Drive Encryption on removable data drives. This feature includes the encryption of:

- USB flash drives
- SD cards
- External hard disk drives

- Other drives that are formatted by using the NTFS, FAT16, FAT32, or exFAT file system

Drive partitioning must meet the BitLocker Drive Encryption Partitioning Requirements.

As with BitLocker, drives that are encrypted by BitLocker To Go can be opened by using a password or smart card on another computer. In Control Panel, use **BitLocker Drive Encryption**.

BitLocker and Active Directory Domain Services (AD DS)

What type of information is stored in AD DS?

Expand table

Stored information	Description
BitLocker recovery password	The recovery password allows unlocking of and access to the drive after a recovery incident. Domain administrators can view the BitLocker recovery password by using the BitLocker Recovery Password Viewer. For more information about this tool, see BitLocker: Use BitLocker Recovery Password Viewer.
BitLocker key package	The key package helps to repair damage to the hard disk that would otherwise prevent standard recovery. Using the key package for recovery requires the BitLocker Repair Tool, `Repair-bde`.

What if BitLocker is enabled on a computer before the computer joins the domain?

If BitLocker is enabled on a drive before policy settings are applied to enforce a backup, the recovery information won't be automatically backed up to AD DS when the computer joins the domain or when the policy settings are subsequently applied. However, the policy settings **Choose how BitLocker-protected operating system drives can be recovered**, **Choose how BitLocker-protected fixed drives can be recovered**, and **Choose how BitLocker-protected removable drives can be recovered** can be chosen to require the computer to be connected to a domain before BitLocker can be enabled to help ensure that recovery information for BitLocker-protected drives in the organization is backed up to AD DS.

For more information how to back up the recovery password to AD DS or Microsoft Entra ID, review the BitLocker operations guide.

> **Important**
>
> Joining a computer to the domain should be the first step for new computers within an organization. After computers are joined to a domain, storing the BitLocker recovery key to AD DS is automatic (when enabled with policy settings).

Is there an event log entry recorded on the client computer to indicate the success or failure of the Microsoft Entra ID or Active Directory backup?

Yes, an event log entry that indicates the success or failure of a backup is recorded on the client computer. However, even if an event log entry says "Success," the information could have been subsequently removed from AD DS, or BitLocker could have been reconfigured in such a way that the Active Directory information can no longer unlock the drive (such as by removing the recovery password key protector). In addition, it's also possible that the log entry could be spoofed.

Ultimately, determining whether a legitimate backup exists in AD DS requires querying AD DS with domain administrator credentials by using the BitLocker password viewer tool.

If I change the BitLocker recovery password on my computer and store the new password in AD DS, will AD DS overwrite the old password?

No. By design, BitLocker recovery password entries don't get deleted from AD DS. Therefore, multiple passwords might be seen for each drive. To identify the latest password, check the date on the object.

What happens if the backup initially fails? Will BitLocker retry it?

If the backup initially fails, such as when a domain controller is unreachable at the time when the BitLocker setup wizard is run, BitLocker doesn't try again to back up the recovery information to AD DS.

When an administrator selects the **Do not enable BitLocker until recovery information is stored in AD DS for (operating system | fixed data | removable data) drives** check box in any of the **Choose how BitLocker-protected operating system drives can be recovered**, **Choose how BitLocker-protected fixed data drives can be recovered**, and **Choose how BitLocker-protected removable data drives can be recovered** policy settings, users can't enable BitLocker unless the computer is connected to the domain and the backup of BitLocker recovery information to AD DS succeeds. With these settings configured if the backup fails, BitLocker can't be enabled, ensuring that administrators will be able to recover BitLocker-protected drives in the organization.

For more info, see BitLocker policy settings.

When an administrator clears these check boxes, the administrator is allowing a drive to be BitLocker-protected without having the recovery information successfully backed up to AD DS; however, BitLocker won't automatically retry the backup if it fails. Instead, administrators can create a backup script, as described earlier in What if BitLocker is enabled on a computer before the computer joins the domain? to capture the information after connectivity is restored.

Security

What form of encryption does BitLocker use? Is it configurable?

BitLocker uses Advanced Encryption Standard (AES) as its encryption algorithm with configurable key lengths of 128 bits or 256 bits. The default encryption setting is AES-128, but the options are configurable by using policy settings.

What is the best practice for using BitLocker on an operating system drive?

The recommended practice for BitLocker configuration on an operating system drive is to implement BitLocker on a computer with a TPM version 1.2 or higher.

What are the implications of using the sleep or hibernate power management options?

BitLocker on operating system drives in its basic configuration provides extra security for the hibernate mode. In sleep mode, the computer is vulnerable to direct memory access attacks,

since unprotected data remains in RAM. Therefore, for improved security, it's recommended to disable sleep mode. Startup authentication can be configured by using a policy setting.

What are the advantages of a TPM?

Most operating systems use a shared memory space and rely on the operating system to manage physical memory. A TPM is a hardware component that uses its own internal firmware and logic circuits for processing instructions, thus shielding it from external software vulnerabilities. Attacking the TPM requires physical access to the computer. Additionally, the tools and skills necessary to attack hardware are often more expensive, and usually aren't as available as the ones used to attack software. And because each TPM is unique to the computer that contains it, attacking multiple TPM computers would be difficult and time-consuming.

> **Note**
>
> Configuring BitLocker with an additional factor of authentication provides even more protection against TPM hardware attacks.

Network Unlock

What is BitLocker Network Unlock?

BitLocker Network Unlock enables easier management for BitLocker-enabled clients and servers that use the TPM+PIN protection method in a domain environment. When a computer that is connected to a wired corporate network is rebooted, Network Unlock allows the PIN entry prompt to be bypassed. It automatically unlocks BitLocker-protected operating system volumes by using a trusted key that is provided by the Windows Deployment Services server as its secondary authentication method.

To use Network Unlock, a PIN must be configured for the computer. When the computer isn't connected to the network, a PIN will need to be provided to unlock it.

BitLocker Network Unlock has software and hardware requirements for both client computers, Windows Deployment services, and domain controllers that must be met before it can be used.

Network Unlock uses two protectors - the TPM protector and the protector provided by the network or by the PIN. Automatic unlock uses a single protector - the one stored in the TPM. If the computer is joined to a network without the key protector, it will prompt to enter a PIN. If

the PIN isn't available, the recovery key will need to be used to unlock the computer if it can't be connected to the network.

For more info, see BitLocker: How to enable Network Unlock.

Use BitLocker with other programs

Can I use EFS with BitLocker?

Yes, Encrypting File System (EFS) can be used to encrypt files on a BitLocker-protected drive. BitLocker helps protect the entire operating system drive against offline attacks, whereas EFS can provide additional user-based file level encryption for security separation between multiple users of the same computer. EFS can also be used in Windows to encrypt files on other drives that aren't encrypted by BitLocker. The root secrets of EFS are stored by default on the operating system drive; therefore, if BitLocker is enabled for the operating system drive, data that is encrypted by EFS on other drives is also indirectly protected by BitLocker.

Can I run a kernel debugger with BitLocker?

Yes. However, the debugger should be turned on before enabling BitLocker. Turning on the debugger ensures that the correct measurements are calculated when sealing to the TPM, allowing the computer to start properly. If debugging needs to be turned on or off when using BitLocker, be sure to suspend BitLocker first to avoid putting the computer into recovery mode.

How does BitLocker handle memory dumps?

BitLocker has a storage driver stack that ensures memory dumps are encrypted when BitLocker is enabled.

Can BitLocker support smart cards for pre-boot authentication?

BitLocker doesn't support smart cards for pre-boot authentication. There's no single industry standard for smart card support in the firmware, and most computers either don't implement firmware support for smart cards, or only support specific smart cards and readers. This lack of standardization makes supporting them difficult.

Can I use a non-Microsoft TPM driver?

Microsoft doesn't support non-Microsoft TPM drivers and strongly recommends against using them with BitLocker. Attempting to use a non-Microsoft TPM driver with BitLocker may cause BitLocker to report that a TPM isn't present on the computer and not allow the TPM to be used with BitLocker.

Can other tools that manage or modify the master boot record work with BitLocker?

We don't recommend modifying the master boot record on computers whose operating system drives are BitLocker-protected for several security, reliability, and product support reasons. Changes to the master boot record (MBR) could change the security environment and prevent the computer from starting normally and complicate any efforts to recover from a corrupted MBR. Changes made to the MBR by anything other than Windows might force the computer into recovery mode or prevent it from booting entirely.

Why is the system check failing when I'm encrypting my operating system drive?

The system check is designed to ensure the computer's BIOS or UEFI firmware is compatible with BitLocker and that the TPM is working correctly. The system check can fail for several reasons:

- The computer's BIOS or UEFI firmware can't read USB flash drives
- The computer's BIOS, uEFI firmware, or boot menu doesn't have reading USB flash drives enabled
- There are multiple USB flash drives inserted into the computer
- The PIN wasn't entered correctly
- The computer's BIOS or UEFI firmware only supports using the function keys (F1-F10) to enter numerals in the pre-boot environment
- The startup key was removed before the computer finished rebooting
- The TPM has malfunctioned and fails to unseal the keys

What can I do if the recovery key on my USB flash drive can't be read?

Some computers can't read USB flash drives in the pre-boot environment. First, check the BIOS or UEFI firmware and boot settings to ensure that the use of USB drives is enabled. If it isn't enabled, enable the use of USB drives in the BIOS or UEFI firmware and boot settings, and then try to read the recovery key from the USB flash drive again. If the USB flash drive still can't be read, the hard drive will need to be mounted as a data drive on another computer so that

there's an operating system to attempt to read the recovery key from the USB flash drive. If the USB flash drive has been corrupted or damaged, a recovery password may need to be supplied or use the recovery information that was backed up to AD DS. Also, if the recovery key is being used in the pre-boot environment, ensure that the drive is formatted by using the NTFS, FAT16, or FAT32 file system.

Why am I unable to save my recovery key to my USB flash drive?

The **Save to USB** option isn't shown by default for removable drives. If the option is unavailable, it means that a system administrator has disallowed the use of recovery keys.

Why am I unable to automatically unlock my drive?

Automatic unlocking for fixed data drives requires the operating system drive to also be protected by BitLocker. If a computer is being used that doesn't have a BitLocker-protected operating system drive, then the fixed drive can't be automatically unlocked. For removable data drives, automatic unlocking can be added by right-clicking the drive in Windows Explorer and selecting **Manage BitLocker**. Password or smart card credentials that were supplied when BitLocker was turned on can still be used to unlock the removable drive on other computers.

Can I use BitLocker in Safe Mode?

Limited BitLocker functionality is available in Safe Mode. BitLocker-protected drives can be unlocked and decrypted by using the **BitLocker Drive Encryption** Control Panel item. Right-clicking to access BitLocker options from Windows Explorer isn't available in Safe Mode.

How do I "lock" a data drive?

Both fixed and removable data drives can be locked by using the Manage-bde command-line tool and the -lock command.

> ⓘ **Note**
>
> Ensure all data is saved to the drive before locking it. Once locked, the drive will become inaccessible.

The syntax of this command is:

Windows Command Prompt

```
manage-bde.exe <driveletter> -lock
```

Outside of using this command, data drives will be locked on shutdown and restart of the operating system. A removable data drive will also be locked automatically when the drive is removed from the computer.

Can I use BitLocker with the Volume Shadow Copy Service?

Yes. However, shadow copies made prior to enabling BitLocker will be automatically deleted when BitLocker is enabled on software-encrypted drives. If a hardware encrypted drive is being used, the shadow copies are retained.

Does BitLocker support virtual hard disks (VHDs)?

BitLocker should work like any specific physical machine within its hardware limitations as long as the environment (physical or virtual) meets Windows Operating System requirements to run.

- With TPM: Yes, it's supported.
- Without TPM: Yes, it's supported (with password protector).

BitLocker is also supported on data volume VHDs, such as those used by clusters.

Can I use BitLocker with virtual machines (VMs)?

Yes, BitLocker can be used with virtual machines (VMs) if the environment meets BitLocker's hardware and software requirements.

Guidelines for troubleshooting BitLocker

Applies to: Supported versions of Windows Client

This article addresses common issues in BitLocker and provides guidelines to troubleshoot these issues. This article also provides information such as what data to collect and what settings to check. This information makes the troubleshooting process much easier.

Review the event logs

Open **Event Viewer** and review the following logs under **Applications and Services Logs** > **Microsoft** > **Windows**:

- **BitLocker-API**. Review the **Management** log, the **Operational** log, and any other logs that are generated in this folder. The default logs have the following unique names:
  - **Microsoft-Windows-BitLocker-API/Management**
  - **Microsoft-Windows-BitLocker-API/Operational**
  - **Microsoft-Windows-BitLocker-API/Tracing** - only displayed when **Show Analytic and Debug Logs** is enabled

- **BitLocker-DrivePreparationTool**. Review the **Admin** log, the **Operational** log, and any other logs that are generated in this folder. The default logs have the following unique names:
  - **Microsoft-Windows-BitLocker-DrivePreparationTool/Admin**
  - **Microsoft-Windows-BitLocker-DrivePreparationTool/Operational**

Additionally, review the **Windows Logs** > **System** log for events that were produced by the TPM and TPM-WMI event sources.

To filter and display or export logs, the wevtutil.exe command-line tool or the Get-WinEvent PowerShell cmdlet can be used.

For example, to use *wevtutil.exe* to export the contents of the operational log from the BitLocker-API folder to a text file that is named *BitLockerAPIOpsLog.txt*, open a Command Prompt window, and run the following command:

Windows Command Prompt

```
wevtutil.exe qe "Microsoft-Windows-BitLocker/BitLocker Operational" /f:text >
BitLockerAPIOpsLog.txt
```

To use the **Get-WinEvent** cmdlet to export the same log to a comma-separated text file, open a Windows PowerShell window and run the following command:

PowerShell

```
Get-WinEvent -logname "Microsoft-Windows-BitLocker/BitLocker Operational"  |
Export-Csv -Path Bitlocker-Operational.csv
```

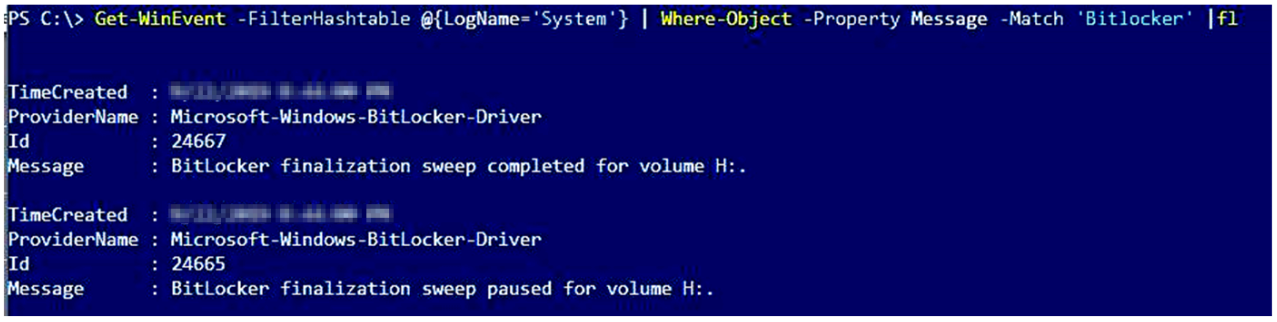
The Get-WinEvent can be used in an elevated PowerShell window to display filtered information from the system or application log by using the following syntax:

- To display BitLocker-related information:

  PowerShell

  ```
  Get-WinEvent -FilterHashtable @{LogName='System'} | Where-Object -Property
  Message -Match 'BitLocker' | fl
  ```

  The output of such a command resembles the following:

- To export BitLocker-related information:

  PowerShell

  ```
  Get-WinEvent -FilterHashtable @{LogName='System'} | Where-Object -Property
  Message -Match 'BitLocker' | Export-Csv -Path System-BitLocker.csv
  ```

- To display TPM-related information:

  PowerShell

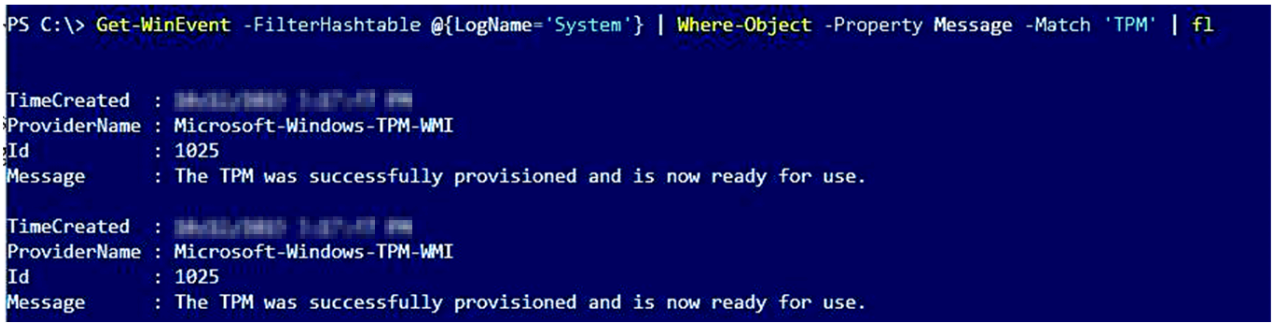
  ```
  Get-WinEvent -FilterHashtable @{LogName='System'} | Where-Object -Property
  Message -Match 'TPM' | fl
  ```

- To export TPM-related information:

  PowerShell

  ```
  Get-WinEvent -FilterHashtable @{LogName='System'} | Where-Object -Property
  Message -Match 'TPM' | Export-Csv -Path System-TPM.csv
  ```

  The output of such a command resembles the following.



> **Note**
>
> When contacting Microsoft Support, it is recommended to export the logs listed in this section.

Gather status information from the BitLocker technologies

Open an elevated Windows PowerShell window, and run each of the following commands:

Expand table

Command	Notes	More Info
`Get-Tpm > C:\TPM.txt`	PowerShell cmdlet that exports information about the local computer's Trusted Platform Module (TPM). This cmdlet shows different values depending on whether the TPM chip is version 1.2 or 2.0. This cmdlet isn't supported in Windows 7.	Get-Tpm
`manage-bde.exe -status > C:\BDEStatus.txt`	Exports information about the general encryption status of all drives on the computer.	manage-bde.exe status
`manage-bde.exe c: -protectors -get > C:\Protectors`	Exports information about the protection methods that are used for the BitLocker encryption key.	manage-bde.exe protectors

Command	Notes	More Info
`reagentc.exe /info > C:\reagent.txt`	Exports information about an online or offline image about the current status of the Windows Recovery Environment (WindowsRE) and any available recovery image.	reagentc.exe
`Get-BitLockerVolume | fl`	PowerShell cmdlet that gets information about volumes that BitLocker Drive Encryption can protect.	Get-BitLockerVolume

Review the configuration information

1. Open an elevated Command Prompt window, and run the following commands:

   Expand table

Command	Notes	More Info
`gpresult.exe /h <Filename>`	Exports the Resultant Set of Policy information, and saves the information as an HTML file.	gpresult.exe
`msinfo.exe /report <Path> /computer <ComputerName>`	Exports comprehensive information about the hardware, system components, and software environment on the local computer. The **/report** option saves the information as a .txt file.	msinfo.exe

2. Open Registry Editor, and export the entries in the following subkeys:

   - `HKLM\SOFTWARE\Policies\Microsoft\FVE`
   - `HKLM\SYSTEM\CurrentControlSet\Services\TPM\`

Check the BitLocker prerequisites

Common settings that can cause issues for BitLocker include the following scenarios:

- The TPM must be unlocked. Check the output of the *get-tpm* PowerShell cmdlet command for the status of the TPM.
- Windows RE must be enabled. Check the output of the *reagentc.exe* command for the status of WindowsRE.
- The system-reserved partition must use the correct format.
  - On Unified Extensible Firmware Interface (UEFI) computers, the system-reserved partition must be formatted as FAT32.

    - On legacy computers, the system-reserved partition must be formatted as NTFS.

- If the device being troubleshot is a slate or tablet PC, use https://gpsearch.azurewebsites.net/#8153 to verify the status of the **Enable use of BitLocker authentication requiring preboot keyboard input on slates** option.

For more information about the BitLocker prerequisites, see BitLocker basic deployment: Using BitLocker to encrypt volumes

Next steps

If the information examined so far indicates a specific issue (for example, WindowsRE isn't enabled), the issue may have a straightforward fix.

Resolving issues that don't have obvious causes depends on exactly which components are involved and what behavior is being see. The gathered information helps narrow down the areas to investigate.

- If the device being troubleshot is managed by Microsoft Intune, see Enforcing BitLocker policies by using Intune: known issues.
- If BitLocker doesn't start or can't encrypt a drive and errors or events that are related to the TPM are occurring, see BitLocker cannot encrypt a drive: known TPM issues.
- If BitLocker doesn't start or can't encrypt a drive, see BitLocker cannot encrypt a drive: known issues.
- If BitLocker Network Unlock doesn't behave as expected, see BitLocker Network Unlock: known issues.
- If BitLocker doesn't behave as expected when an encrypted drive is recovered, or if BitLocker unexpectedly recovered a drive, see BitLocker recovery: known issues.
- If BitLocker or the encrypted drive doesn't behave as expected, and errors or events that are related to the TPM are occurring, see BitLocker and TPM: other known issues.
- If BitLocker or the encrypted drive doesn't behave as expected, see BitLocker configuration: known issues.

It's recommended to keep the gathered information handy in case Microsoft Support is contacted for help with resolving the issue.

Last updated on 02/12/2026

BitLocker cannot encrypt a drive: known issues

Applies to: Supported versions of Windows Client

This article describes common issues that prevent BitLocker from encrypting a drive. This article also provides guidance to address these issues.

> **Note**
>
> If it is determined that the BitLocker issue involves the trusted platform module (TPM), see BitLocker cannot encrypt a drive: known TPM issues.

Error 0x80310059: BitLocker drive encryption is already performing an operation on this drive

When BitLocker Drive Encryption is turned on a computer that is running Windows 10 Professional or Windows 11, the following message may appear:

> ERROR: An error occurred (code 0x80310059): BitLocker Drive Encryption is already performing an operation on this drive. Please complete all operations before continuing.
> NOTE: If the -on switch has failed to add key protectors or start encryption, you may need to call manage-bde -off before attempting -on again.

Cause of Error 0x80310059

This issue may be caused by settings that are controlled by group policy objects (GPOs).

Resolution for Error 0x80310059

> **Important**
>
> Follow the steps in this section carefully. Serious problems might occur if the registry is modified incorrectly. Before modifying the registry, back up the registry for restoration in case problems occur.

To resolve this issue, follow these steps:

1. Start Registry Editor, and navigate to the following subkey:

   `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE`

2. Delete the following entries:

   - `OSPlatformValidation_BIOS`
   - `OSPlatformValidation_UEFI`
   - `PlatformValidation`

3. Exit registry editor, and turn on BitLocker drive encryption again.

Last updated on 02/12/2026

Enforcing BitLocker policies by using Intune: known issues

Applies to: Supported versions of Windows Client

This article helps troubleshooting issues that may be experienced if using Microsoft Intune policy to manage silent BitLocker encryption on devices. The Intune portal indicates whether BitLocker has failed to encrypt one or more managed devices.



To start narrowing down the cause of the problem, review the event logs as described in Troubleshoot BitLocker. Concentrate on the **Management** and **Operations** logs in the **Applications and Services logs** > **Microsoft** > **Windows** > **BitLocker-API** folder. The following sections provide more information about how to resolve the indicated events and error messages:

- Event ID 853: Error: A compatible Trusted Platform Module (TPM) Security Device cannot be found on this computer
- Event ID 853: Error: BitLocker Drive Encryption detected bootable media (CD or DVD) in the computer
- Event ID 854: WinRE is not configured
- Event ID 851: Contact manufacturer for BIOS upgrade
- Error message: The UEFI variable 'SecureBoot' could not be read
- Event ID 846, 778, and 851: Error 0x80072f9a
- Error message: There are conflicting group policy settings for recovery options on operating system drives

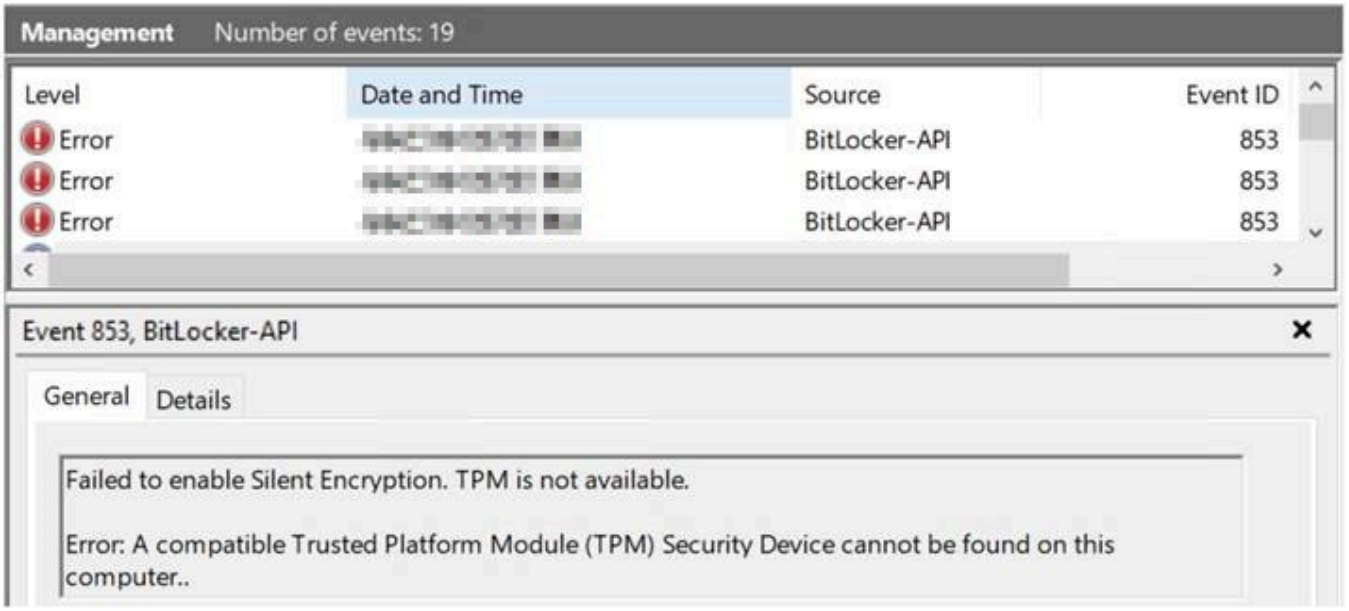
If there's no clear trail of events or error messages to follow, other areas to investigate include the following areas:

- Review the hardware requirements for using Intune to manage BitLocker on devices
- Review BitLocker policy configuration

For information about the procedure to verify whether Intune policies are enforcing BitLocker correctly, see Verifying that BitLocker is operating correctly.

Event ID 853: Error: A compatible Trusted Platform Module (TPM) Security Device cannot be found on this computer

Event ID 853 can carry different error messages, depending on the context. In this case, the Event ID 853 error message indicates that the device doesn't appear to have a TPM. The event information will be similar to the following event:



Cause of Event ID 853: Error: A compatible Trusted Platform Module (TPM) Security Device cannot be found on this computer

The device that is being secured may not have a TPM chip, or the device BIOS might have been configured to disable the TPM.

Resolution for Event ID 853: Error: A compatible Trusted Platform Module (TPM) Security Device cannot be found on this computer

To resolve this issue, verify the following configurations:

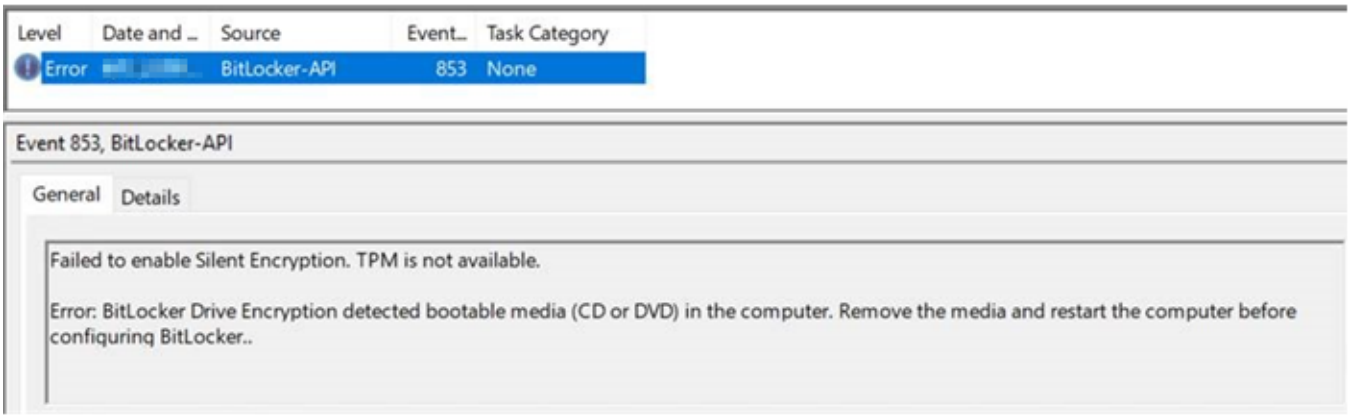
- The TPM is enabled in the device BIOS.
- The TPM status in the TPM management console is similar to the following statuses:

- Ready (TPM 2.0)
- Initialized (TPM 1.2)

For more information, see Troubleshoot the TPM.

Event ID 853: Error: BitLocker Drive Encryption detected bootable media (CD or DVD) in the computer

In this case, event ID 853 is displayed, and the error message in the event indicates that bootable media is available to the device. The event information resembles the following.



Cause of Event ID 853: Error: BitLocker Drive Encryption detected bootable media (CD or DVD) in the computer

During the provisioning process, BitLocker drive encryption records the configuration of the device to establish a baseline. If the device configuration changes later (for example, if the media is removed), BitLocker recovery mode automatically starts.

To avoid this situation, the provisioning process stops if it detects a removable bootable media.

Resolution for Event ID 853: Error: BitLocker Drive Encryption detected bootable media (CD or DVD) in the computer

Remove the bootable media, and restart the device. After the device restarts, verify the encryption status.

Event ID 854: WinRE is not configured

The event information resembles the following error message:

> Failed to enable Silent Encryption. WinRe is not configured.
>
> Error: This PC cannot support device encryption because WinRE is not properly configured.

Cause of Event ID 854: WinRE is not configured

Windows Recovery Environment (WinRE) is a minimal Windows operating system that is based on Windows Preinstallation Environment (Windows PE). WinRE includes several tools that an administrator can use to recover or reset Windows and diagnose Windows issues. If a device can't start the regular Windows operating system, the device tries to start WinRE.

The provisioning process enables BitLocker drive encryption on the operating system drive during the Windows PE phase of provisioning. This action makes sure that the drive is protected before the full operating system is installed. The provisioning process also creates a system partition for WinRE to use if the system crashes.

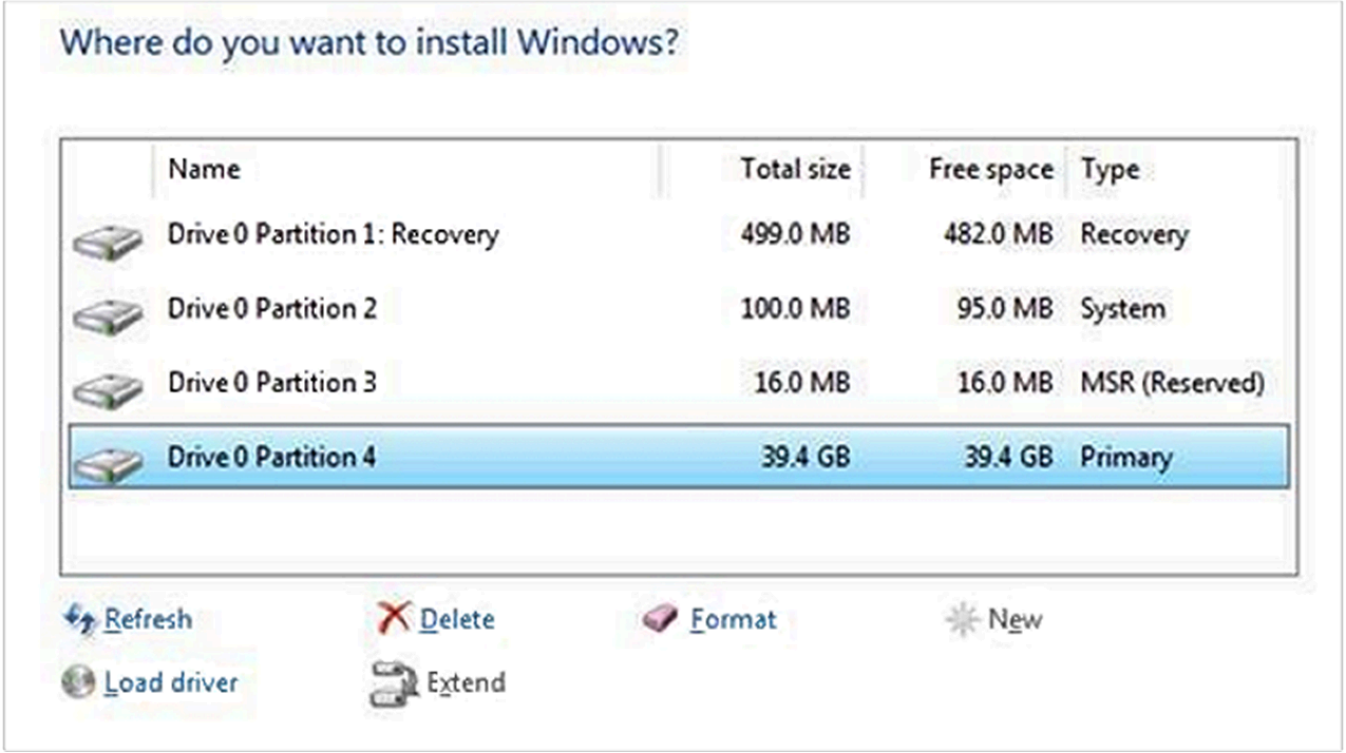
If WinRE isn't available on the device, provisioning stops.

Resolution for Event ID 854: WinRE is not configured

This issue can be resolved by verifying the configuration of the disk partitions, the status of WinRE, and the Windows Boot Loader configuration by following these steps:

Step 1: Verify the configuration of the disk partitions

The procedures described in this section depend on the default disk partitions that Windows configures during installation. Windows 11 and Windows 10 automatically create a recovery partition that contains the *Winre.wim* file. The partition configuration resembles the following.



To verify the configuration of the disk partitions, open an elevated Command Prompt window and run the following commands:

Windows Command Prompt

```
diskpart.exe
list volume
```

```
C:\WINDOWS\system32>diskpart

Microsoft DiskPart version 10.0.17763.1

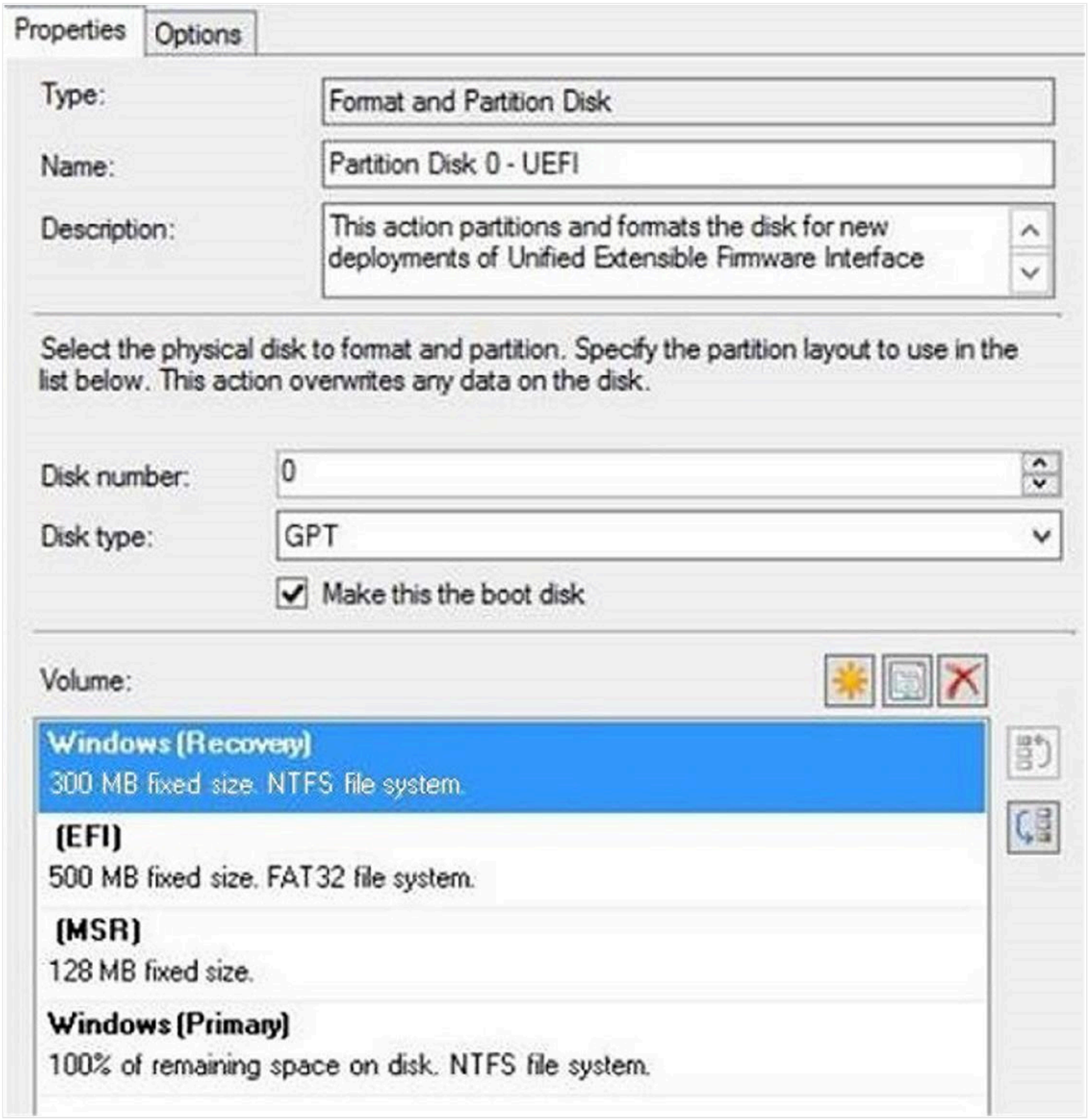
Copyright (C) Microsoft Corporation.
On computer: [redacted]

DISKPART> list vol

  Volume ###  Ltr  Label        Fs     Type        Size     Status     Info
  ----------  ---  -----------  -----  ----------  -------  ---------  --------
  Volume 0     E                       DVD-ROM         0 B  No Media
  Volume 1         System       NTFS   Partition    499 MB  Healthy    System
  Volume 2     C   OSDisk       NTFS   Partition    216 GB  Healthy    Boot
  Volume 3     D   New Volume   NTFS   Partition    248 GB  Healthy
```

If the status of any of the volumes isn't healthy or if the recovery partition is missing, Windows may need to be reinstalled. Before reinstalling Windows, check the configuration of the Windows image that is being provisioned. Make sure that the image uses the correct disk

configuration. The image configuration should resemble the following (this example is from Microsoft Configuration Manager):



Step 2: Verify the status of WinRE

To verify the status of WinRE on the device, open an elevated Command Prompt window and run the following command:

Windows Command Prompt

```
reagentc.exe /info
```

The output of this command resembles the following.

```
C:\>reagentc /info
Windows Recovery Environment (Windows RE) and system reset configuration
Information:

    Windows RE status:         Enabled
    Windows RE location:       \\?\GLOBALROOT\device\harddisk0\partition3\Recovery\WindowsRE
    Boot Configuration Data (BCD) identifier: [illegible]
    Recovery image location:
    Recovery image index:      0
    Custom image location:
    Custom image index:        0

REAGENTC.EXE: Operation Successful.
```

If the **Windows RE status** isn't **Enabled**, run the following command to enable it:

Windows Command Prompt

```
reagentc.exe /enable
```

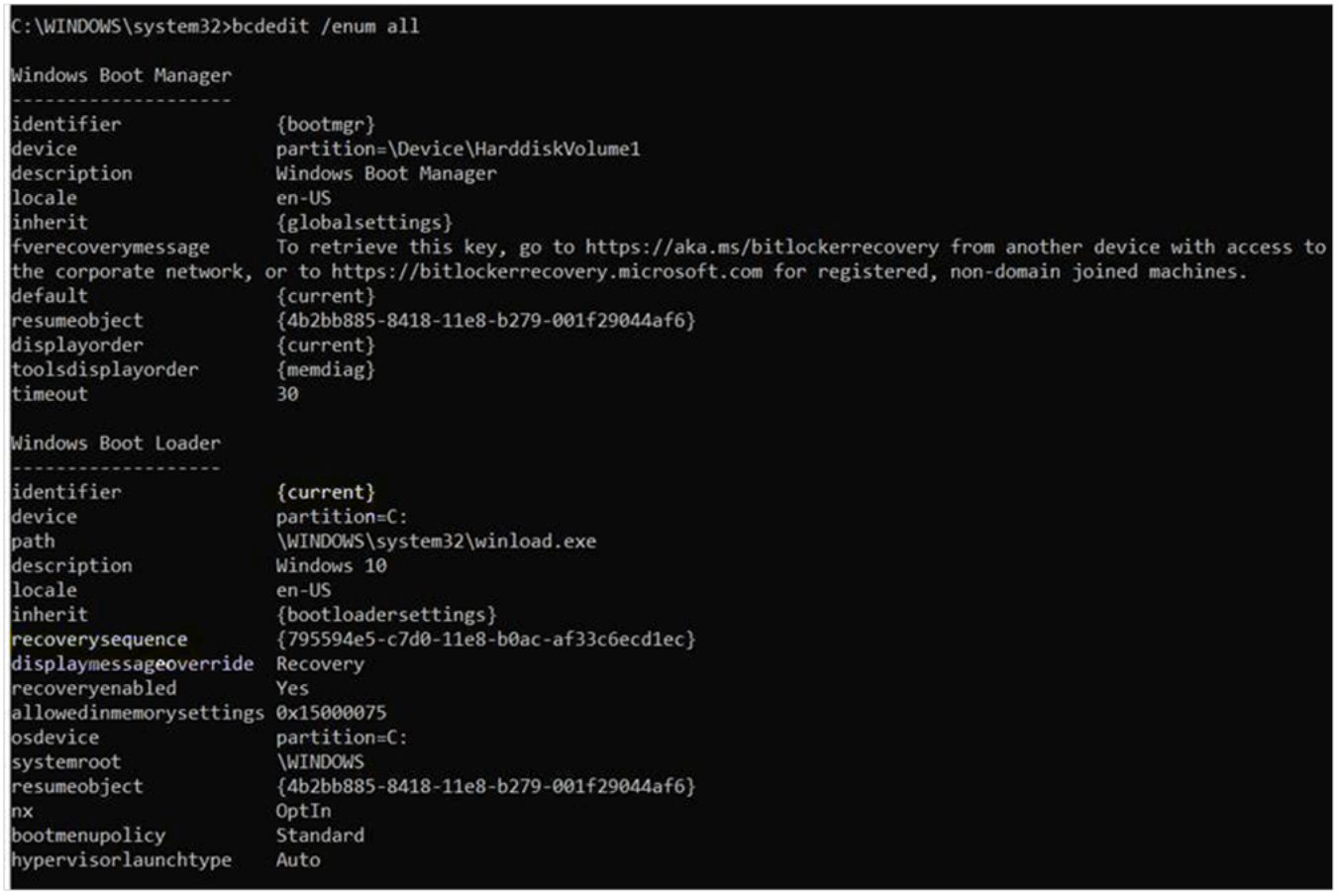
Step 3: Verify the Windows Boot Loader configuration

If the partition status is healthy, but the *reagentc.exe /enable* command results in an error, verify whether the Windows Boot Loader contains the recovery sequence GUID by running the following command in an elevated Command Prompt window:

Windows Command Prompt

```
bcdedit.exe /enum all
```

The output of this command will be similar to the following output:



```
C:\WINDOWS\system32>bcdedit /enum all

Windows Boot Manager
--------------------
identifier              {bootmgr}
device                  partition=\Device\HarddiskVolume1
description             Windows Boot Manager
locale                  en-US
inherit                 {globalsettings}
fverecoverymessage      To retrieve this key, go to https://aka.ms/bitlockerrecovery from another device with access to
the corporate network, or to https://bitlockerrecovery.microsoft.com for registered, non-domain joined machines.
default                 {current}
resumeobject            {4b2bb885-8418-11e8-b279-001f29044af6}
displayorder            {current}
toolsdisplayorder       {memdiag}
timeout                 30

Windows Boot Loader
-------------------
identifier              {current}
device                  partition=C:
path                    \WINDOWS\system32\winload.exe
description             Windows 10
locale                  en-US
inherit                 {bootloadersettings}
recoverysequence        {795594e5-c7d0-11e8-b0ac-af33c6ecd1ec}
displaymessageoverride  Recovery
recoveryenabled         Yes
allowedinmemorysettings 0x15000075
osdevice                partition=C:
systemroot              \WINDOWS
resumeobject            {4b2bb885-8418-11e8-b279-001f29044af6}
nx                      OptIn
bootmenupolicy          Standard
hypervisorlaunchtype    Auto
```

In the output, locate the **Windows Boot Loader** section that includes the line **identifier={current}**. In that section, locate the **recoverysequence** attribute. The value of this attribute should be a GUID value, not a string of zeros.

Event ID 851: Contact the manufacturer for BIOS upgrade instructions

The event information will be similar to the following error message:

> Failed to enable Silent Encryption.
>
> Error: BitLocker Drive Encryption cannot be enabled on the operating system drive. Contact the computer manufacturer for BIOS upgrade instructions.

Cause of Event ID 851: Contact the manufacturer for BIOS upgrade instructions

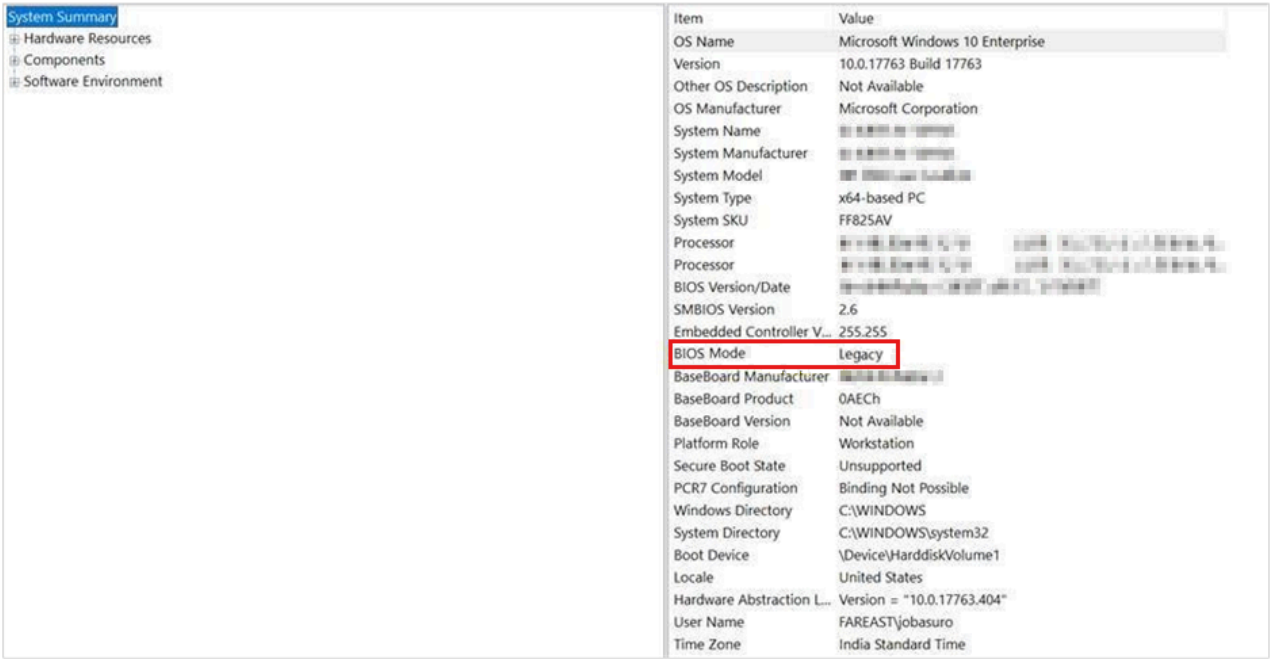
The device must have Unified Extensible Firmware Interface (UEFI) BIOS. Silent BitLocker drive encryption doesn't support legacy BIOS.

Resolution for Event ID 851: Contact the manufacturer for BIOS upgrade instructions

To verify the BIOS mode, use the System Information application by following these steps:

1. Select **Start**, and enter **msinfo32** in the **Search** box.

2. Verify that the **BIOS Mode** setting is **UEFI** and not **Legacy**.



3. If the **BIOS Mode** setting is **Legacy**, the UEFI firmware needs to be switched to **UEFI** or **EFI** mode. The steps for switching to **UEFI** or **EFI** mode are specific to the device.

> ⓘ **Note**
>
> If the device supports only Legacy mode, Intune can't be used to manage BitLocker Device Encryption on the device.

Error message: The UEFI variable 'SecureBoot' could not be read

An error message similar to the following error message is displayed:

> **Error:** BitLocker cannot use Secure Boot for integrity because the UEFI variable 'SecureBoot' could not be read. A required privilege is not held by the client.

Cause of Error message: The UEFI variable 'SecureBoot' could not be read

A platform configuration register (PCR) is a memory location in the TPM. In particular, PCR 7 measures the state of secure boot. Silent BitLocker drive encryption requires the secure boot to be turned on.

Resolution for Error message: The UEFI variable 'SecureBoot' could not be read

This issue can be resolved by verifying the PCR validation profile of the TPM and the secure boot state by following these steps:

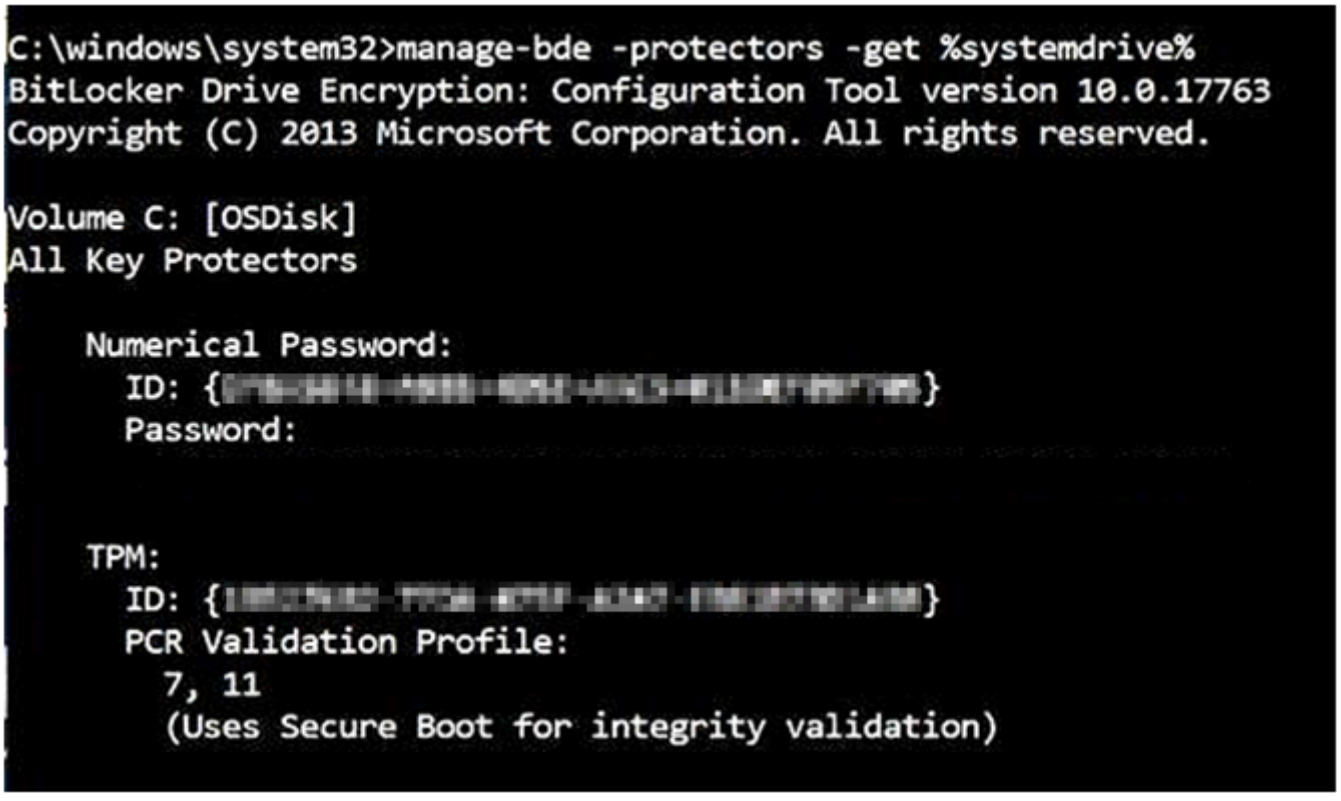
Step 1: Verify the PCR validation profile of the TPM

To verify that PCR 7 is in use, open an elevated Command Prompt window and run the following command:

Windows Command Prompt

```
Manage-bde.exe -protectors -get %systemdrive%
```

In the TPM section of the output of this command, verify whether the **PCR Validation Profile** setting includes **7**, as follows:



If **PCR Validation Profile** doesn't include **7** (for example, the values include **0**, **2**, **4**, and **11**, but not **7**), then secure boot isn't turned on.

```
C:\WINDOWS\system32>manage-bde -protectors -get %systemdrive%
BitLocker Drive Encryption: Configuration Tool version 10.0.17763
Copyright (C) 2013 Microsoft Corporation. All rights reserved.

Volume C: [OSDisk]
All Key Protectors

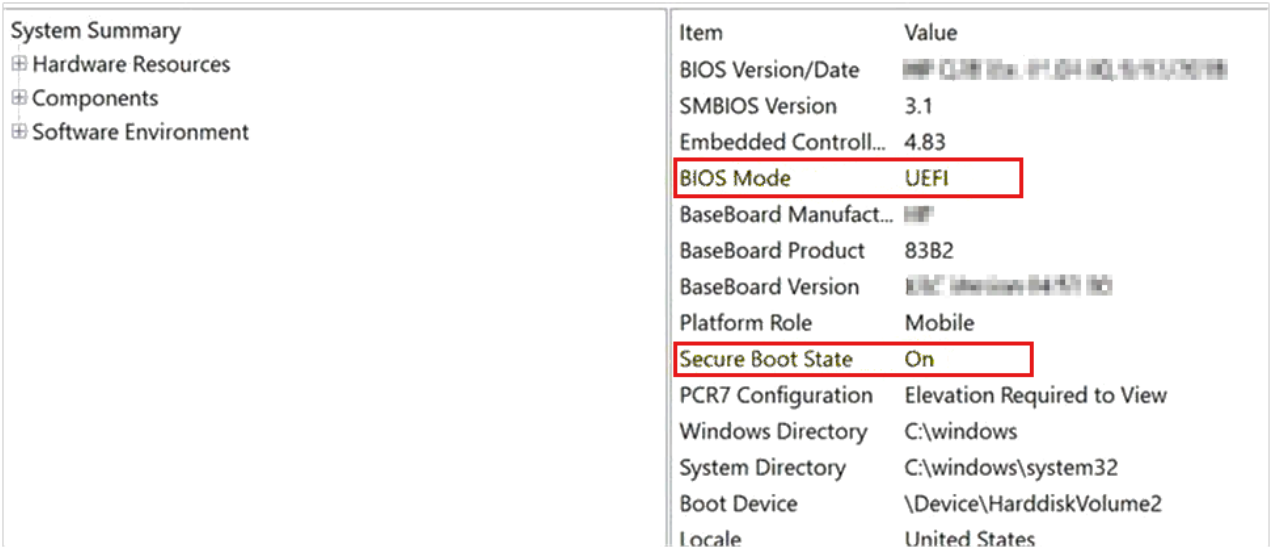
    Numerical Password:
      ID: {[redacted]}
      Password:


    TPM:
      ID: {[redacted]}
      PCR Validation Profile:
        0, 2, 4, 8, 9, 10, 11
```

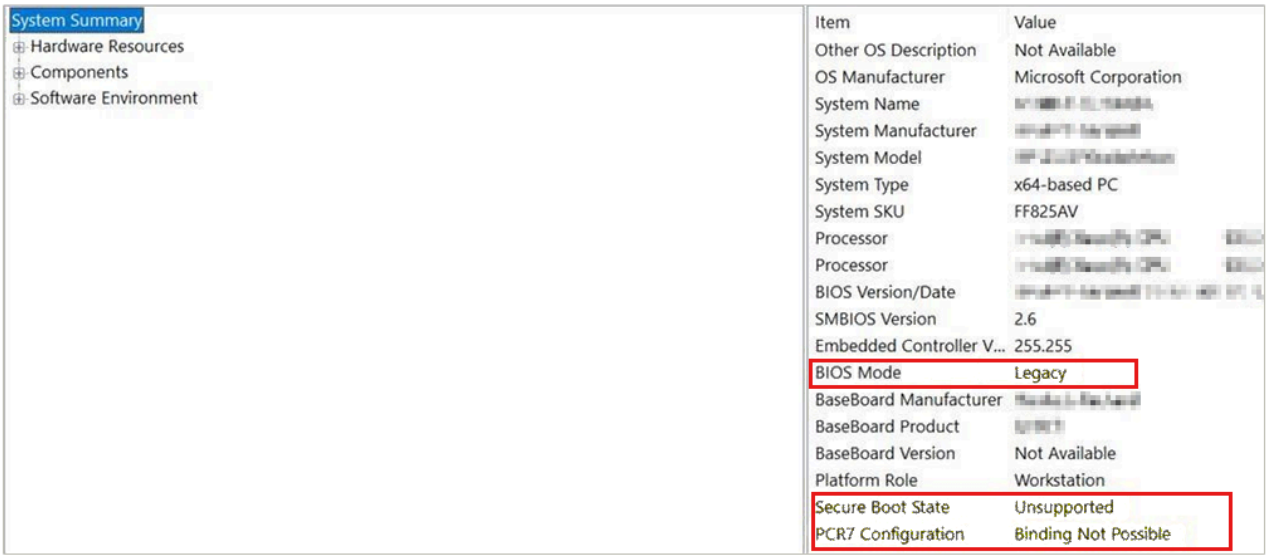
2: Verify the secure boot state

To verify the secure boot state, use the System Information application by following these steps:

1. Select **Start**, and enter **msinfo32** in the **Search** box.
2. Verify that the **Secure Boot State** setting is **On**, as follows:



3. If the **Secure Boot State** setting is **Unsupported**, Silent BitLocker Encryption can't be used on the device.



> **Note**
>
> The Confirm-SecureBootUEFI PowerShell cmdlet can also be used to verify the Secure Boot state by opening an elevated PowerShell window and running the following command:
>
> ```
> PowerShell
> Confirm-SecureBootUEFI
> ```
>
> If the computer supports Secure Boot and Secure Boot is enabled, this cmdlet returns "True."
>
> If the computer supports secure boot and secure boot is disabled, this cmdlet returns "False."
>
> If the computer does not support Secure Boot or is a BIOS (non-UEFI) computer, this cmdlet returns "Cmdlet not supported on this platform."

Event ID 846, 778, and 851: Error 0x80072f9a

Consider the following scenario:

Intune policy is being deployed to encrypt a Windows 10, version 1809 device, and the recovery password is being stored in Microsoft Entra ID. As part of the policy configuration, the **Allow standard users to enable encryption during Microsoft Entra join** option has been selected.

The policy deployment fails and the failure generates the following events in Event Viewer in the **Applications and Services Logs** > **Microsoft** > **Windows** > **BitLocker API** folder:

> Event ID:846
>
> Event: Failed to backup BitLocker Drive Encryption recovery information for volume C: to your Microsoft Entra ID.
>
> TraceId: {0000aaaa-11bb-cccc-dd22-eeeeee333333} Error: Unknown HResult Error code: 0x80072f9a

> Event ID:778
>
> Event: The BitLocker volume C: was reverted to an unprotected state.

> Event ID: 851
>
> Event: Failed to enable Silent Encryption.
>
> Error: Unknown HResult Error code: 0x80072f9a.

These events refer to Error code 0x80072f9a.

Cause of Event ID 846, 778, and 851: Error 0x80072f9a

These events indicate that the signed-in user doesn't have permission to read the private key on the certificate that is generated as part of the provisioning and enrollment process. Therefore, the BitLocker MDM policy refresh fails.

The issue affects Windows 10 version 1809.

Resolution for Event ID 846, 778, and 851: Error 0x80072f9a

To resolve this issue, install the May 21, 2019 update.

Error message: There are conflicting group policy settings for recovery options on operating system drives

An error message similar to the following error message is displayed:

> **Error:** BitLocker Drive Encryption cannot be applied to this drive because there are conflicting Group Policy settings for recovery options on operating system drives. Storing recovery information to Active Directory Domain Services cannot be required when the generation of recovery passwords is not permitted. Please have your system administrator resolve these policy conflicts before attempting to enable BitLocker...

Resolution for Error message: There are conflicting group policy settings for recovery options on operating system drives

To resolve this issue, review the group policy object (GPO) settings for conflicts. For more information, see the next section, Review BitLocker policy configuration.

For more information about GPOs and BitLocker, see BitLocker Group Policy Reference.

Review BitLocker policy configuration

For information about the procedure to use policy together with BitLocker and Intune, see the following resources:

- BitLocker management for enterprises: Managing devices joined to Microsoft Entra ID
- BitLocker Group Policy Reference
- Configuration service provider reference
- Policy CSP – BitLocker
- BitLocker CSP
- Enable ADMX-backed policies in MDM
- gpresult

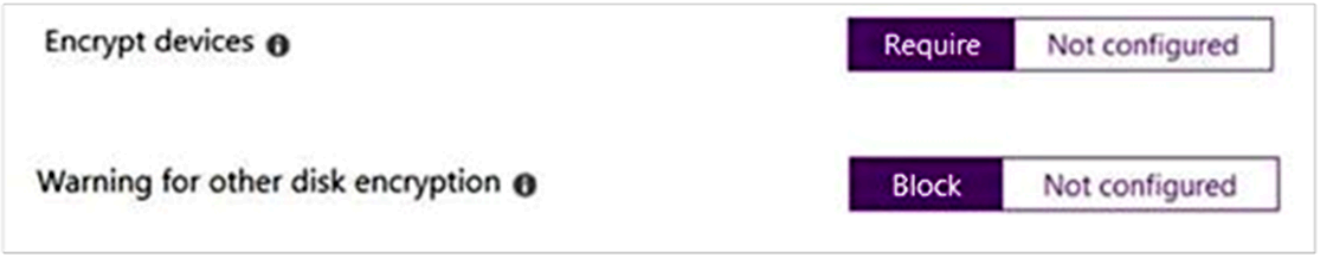
Intune offers the following enforcement types for BitLocker:

- **Automatic** (Enforced when the device joins Microsoft Entra ID during the provisioning process. This option is available in Windows 10 version 1703 and later.)
- **Silent** (Endpoint protection policy. This option is available in Windows 10 version 1803 and later.)
- **Interactive** (Endpoint policy for Windows versions that are older than Windows 10 version 1803.)

If the device runs Windows 10 version 1703 or later, supports Modern Standby (also known as Instant Go) and is HSTI-compliant, joining the device to Microsoft Entra ID triggers automatic

device encryption. A separate endpoint protection policy isn't required to enforce device encryption.

If the device is HSTI-compliant but doesn't support Modern Standby, an endpoint protection policy has to be configured to enforce silent BitLocker drive encryption. The settings for this policy should be similar to the following settings:



The OMA-URI references for these settings are as follows:

- OMA-URI: **./Device/Vendor/MSFT/BitLocker/RequireDeviceEncryption**
  Value Type: **Integer**
  Value: **1** (1 = Require, 0 = Not Configured)

- OMA-URI: **./Device/Vendor/MSFT/BitLocker/AllowWarningForOtherDiskEncryption**
  Value Type: **Integer**
  Value: **0** (0 = Blocked, 1 = Allowed)

> ⓘ **Note**
>
> Because of an update to the BitLocker Policy CSP, if the device uses Windows 10 version 1809 or later, an endpoint protection policy can be used to enforce silent BitLocker Device Encryption even if the device is not HSTI-compliant.

> ⓘ **Note**
>
> If the **Warning for other disk encryption** setting is set to **Not configured**, the BitLocker drive encryption wizard has to be manually started.

If the device doesn't support Modern Standby but is HSTI-compliant, and it uses a version of Windows that is earlier than Windows 10, version 1803, an endpoint protection policy that has the settings that are described in this article delivers the policy configuration to the device. However, Windows then notifies the user to manually enable BitLocker Drive Encryption. When the user selects the notification, it will start the BitLocker Drive Encryption wizard.

Intune provides settings that can be used to configure automatic device encryption for Autopilot devices for standard users. Each device must meet the following requirements:

- Be HSTI-compliant
- Support Modern Standby
- Use Windows 10 version 1803 or later



The OMA-URI references for these settings are as follows:

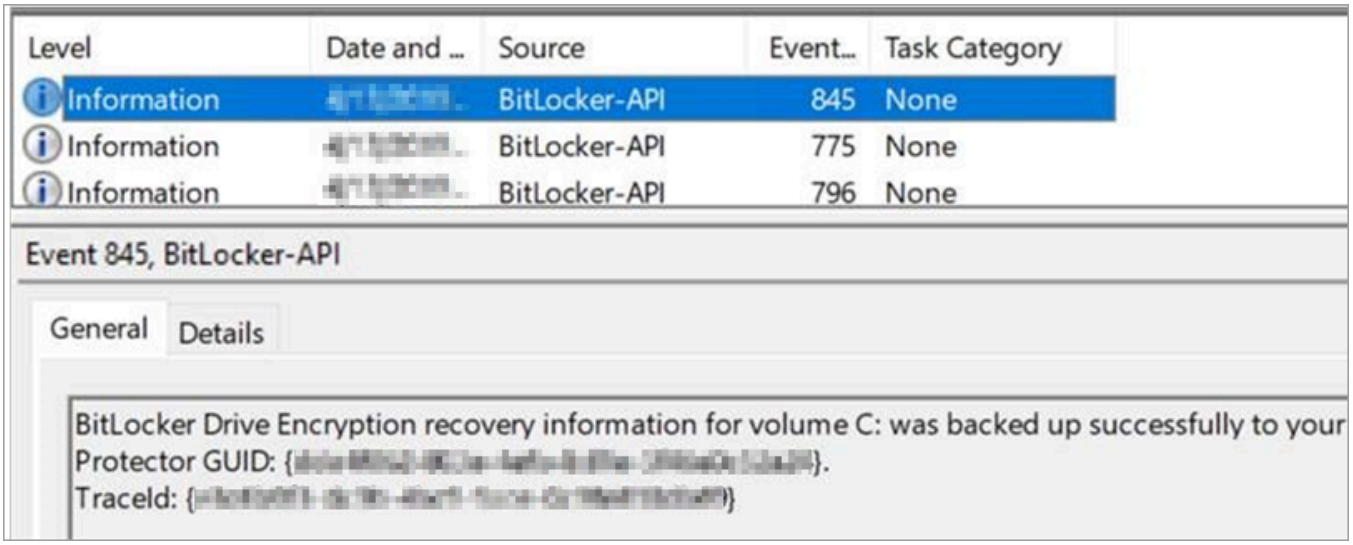
- OMA-URI: **./Device/Vendor/MSFT/BitLocker/AllowStandardUserEncryption**
  Value Type: **Integer** Value: **1**

> **Note**
>
> This node works together with the **RequireDeviceEncryption** and **AllowWarningForOtherDiskEncryption** nodes. For this reason, when the following settings are set:
>
> - **RequireDeviceEncryption** to **1**
> - **AllowStandardUserEncryption** to **1**
> - **AllowWarningForOtherDiskEncryption** to **0**
>
> Intune enforces silent BitLocker encryption for Autopilot devices that have standard user profiles.

Verifying that BitLocker is operating correctly

During regular operations, BitLocker drive encryption generates events such as Event ID 796 and Event ID 845.



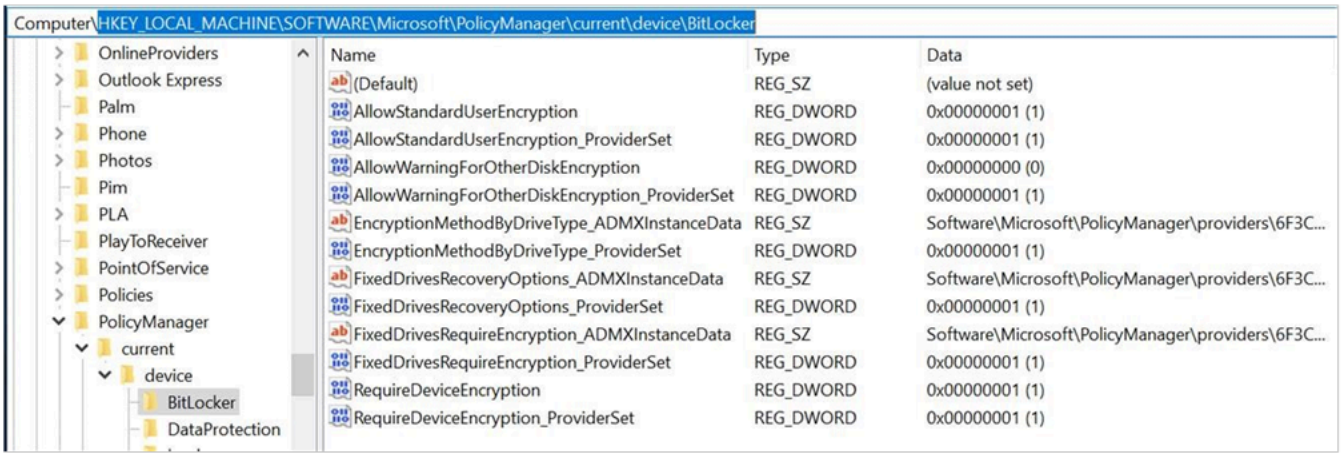


It can also be determined whether the BitLocker recovery password has been uploaded to Microsoft Entra ID by checking the device details in the Microsoft Entra Devices section.



On the device, check the Registry Editor to verify the policy settings on the device. Verify the entries under the following subkeys:

- `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PolicyManager\current\device\BitLocker`
- `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\PolicyManager\current\device`



Last updated on 02/12/2026

BitLocker Network Unlock: known issues

Applies to: Supported versions of Windows Client

By using the BitLocker Network Unlock feature, computers can be managed remotely without having to enter a BitLocker PIN when each computer starts up. To configure this behavior, the environment needs to meet the following requirements:

- Each computer belongs to a domain.
- Each computer has a wired connection to the internal network.
- The internal network uses DHCP to manage IP addresses.
- Each computer has a DHCP driver implemented in its Unified Extensible Firmware Interface (UEFI) firmware.

For general guidelines about how to troubleshoot BitLocker Network Unlock, see How to enable Network Unlock: Troubleshoot Network Unlock.

This article describes several known issues that may be encountered when BitLocker Network Unlock is used and provides guidance to address these issues.

Tip

BitLocker Network Unlock can be detected if it is enabled on a specific computer use the following steps on UEFI computers:

1. Open an elevated command prompt window and run the following command:

   Windows Command Prompt

   ```
   manage-bde.exe -protectors -get <Drive>
   ```

   For example:

   Windows Command Prompt

   ```
   manage-bde.exe -protectors -get C:
   ```

   If the output of this command includes a key protector of type **TpmCertificate (9)**, the configuration is correct for BitLocker Network Unlock.

2. Start Registry Editor, and verify the following settings:

   a. The following registry key exists and has the following value:

      - **Subkey**: `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\FVE`
      - **Type**: `REG_DWORD`
      - **Value**: `OSManageNKP` equal to `1` (True)

   b. The registry key:

      `HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\SystemCertificates\FVE_NKP\Certificates`

      has an entry whose name matches the name of the certificate thumbprint of the BitLocker Network Unlock key protector that was found in step 1.

On a Surface Pro 4 device, BitLocker Network Unlock doesn't work because the UEFI network stack is incorrectly configured

Consider the following scenario:

BitLocker Network Unlock has been configured as described in BitLocker: How to enable Network Unlock. UEFI of a Surface Pro 4 has been configured to use DHCP. However, when the Surface Pro 4 is restarted, it still prompts for a BitLocker PIN.

When testing another device, such as a different type of tablet or laptop PC that's configured to use the same infrastructure, the device restarts as expected, without prompting for the BitLocker PIN. This test confirms that the infrastructure is correctly configured, and the issue is specific to the device.

Cause of BitLocker Network Unlock not working on Surface Pro 4

The UEFI network stack on the device is incorrectly configured.

Resolution for BitLocker Network Unlock not working on Surface Pro 4

To correctly configure the UEFI network stack of the Surface Pro 4, the Microsoft Surface Enterprise Management Mode (SEMM) needs to be used. For information about SEMM, see Enroll and configure Surface devices with SEMM.

> ⓘ **Note**
>
> If SEMM can't be used, the Surface Pro 4 may be able to use BitLocker Network Unlock by configuring the Surface Pro 4 to use the network as its first boot option.

Unable to use BitLocker Network Unlock feature on a Windows client computer

Consider the following scenario:

BitLocker Network Unlock has been configured as described in BitLocker: How to enable Network Unlock. A Windows 8 client computer is connected to the internal network with an ethernet cable. However, when the device is restarted, the device still prompts for the BitLocker PIN.

Cause of unable to use BitLocker Network Unlock feature on a Windows client computer

A Windows 8-based or Windows Server 2012-based client computer sometimes doesn't receive or use the BitLocker Network Unlock protector, depending on whether the client receives unrelated BOOTP replies from a DHCP server or WDS server.

DHCP servers may send any DHCP options to a BOOTP client as allowed by the DHCP options and BOOTP vendor extensions. This behavior means that because a DHCP server supports BOOTP clients, the DHCP server replies to BOOTP requests.

The manner in which a DHCP server handles an incoming message depends in part on whether the message uses the Message Type option:

- The first two messages that the BitLocker Network Unlock client sends are DHCP DISCOVER\REQUEST messages. They use the Message Type option, so the DHCP server treats them as DHCP messages.
- The third message that the BitLocker Network Unlock client sends doesn't have the Message Type option. The DHCP server treats the message as a BOOTP request.

A DHCP server that supports BOOTP clients must interact with those clients according to the BOOTP protocol. The server must create a BOOTP BOOTREPLY message instead of a DHCP DHCPOFFER message. In other words, the server must not include the DHCP message option type and must not exceed the size limit for BOOTREPLY messages. After the server sends the BOOTP BOOTREPLY message, the server marks a binding for a BOOTP client as BOUND. A non-DHCP client doesn't send a DHCPREQUEST message, nor does that client expect a DHCPACK message.

If a DHCP server that isn't configured to support BOOTP clients receives a BOOTREQUEST message from a BOOTP client, that server silently discards the BOOTREQUEST message.

For more information about DHCP and BitLocker Network Unlock, see BitLocker: How to enable Network Unlock: Network Unlock sequence.

Resolution for unable to use BitLocker Network Unlock feature on a Windows client computer

To resolve this issue, change the configuration of the DHCP server by changing the **DHCP** option from **DHCP and BOOTP** to **DHCP**.

Last updated on 02/12/2026

BitLocker recovery: known issues

Applies to: Supported versions of Windows Client

This article describes common issues that may prevent BitLocker from behaving as expected when a drive is recovered, or that may cause BitLocker to start recovery unexpectedly. The article also provides guidance to address these issues.

> **Note**
>
> In this article, "recovery password" refers to the 48-digit recovery password and "recovery key" refers to 32-digit recovery key. For more information, see BitLocker key protectors.

For more information about device encryption, see BitLocker automatic device encryption hardware requirements.

Windows prompts for a non-existing BitLocker recovery password

Windows prompts for a BitLocker recovery password. However, a BitLocker recovery password wasn't configured.

Resolution for Windows prompts for a non-existing BitLocker recovery password

The BitLocker and Active Directory Domain Services (AD DS) FAQ address situations that may produce this symptom, and provides information about the procedure to resolve the issue:

- What if BitLocker is enabled on a computer before the computer joins the domain?
- What happens if the backup initially fails? Will BitLocker retry it?

The recovery password for a laptop wasn't backed up, and the laptop is locked

Consider the following scenario:

The hard disk of a Windows 11 or Windows 10 laptop has to be recovered. The disk was encrypted by using BitLocker Driver Encryption. However, the BitLocker recovery password wasn't backed up, and the usual user of the laptop isn't available to provide the password.

Resolution for the recovery password for a laptop wasn't backed up

You can use either of the following methods to manually back up or synchronize an online client's existing recovery information:

- Create a Windows Management Instrumentation (WMI) script that backs up the information. For more information, see BitLocker Drive Encryption Provider.
- In an elevated Command Prompt window, use the manage-bde.exe command to back up the information.

  For example, to back up all of the recovery information for the C: drive to AD DS, open an elevated Command Prompt window and run the following command:

  ***cmd manage-bde.exe -protectors -adbackup C:

> ⓘ **Note**
>
> BitLocker does not automatically manage this backup process.

Tablet devices don't support using `manage-bde.exe -forcerecovery` to test recovery mode

Consider the following scenario:

BitLocker recovery needs to be tested on a tablet or slate device by running the following command:

***cmd manage-bde.exe -forcerecovery

However, after entering the recovery password, the device can't start.

Cause of tablet devices don't support using `manage-bde.exe -forcerecovery` to test recovery mode

> **Important**
>
> Tablet devices do not support the `manage-bde.exe -forcerecovery` command.

This issue occurs because the Windows Boot Manager can't process touch-input during the pre-boot phase of startup. If Boot Manager detects that the device is a tablet, it redirects the startup process to the Windows Recovery Environment (WinRE), which can process touch-input.

If WindowsRE detects the TPM protector on the hard disk, it does a PCR reseal. However, the `manage-bde.exe -forcerecovery` command deletes the TPM protectors on the hard disk. Therefore, WinRE can't reseal the PCRs. This failure triggers an infinite BitLocker recovery cycle and prevents Windows from starting.

This behavior is by design for all versions of Windows.

Workaround for tablet devices don't support using `manage-bde.exe -forcerecovery` to test recovery mode

To resolve the restart loop, follow these steps:

1. On the BitLocker Recovery screen, select **Skip this drive**.
2. Select **Troubleshoot** > **Advanced Options** > **Command Prompt**.
3. In the Command Prompt window, run the following commands:

   Windows Command Prompt

   ```
   manage-bde.exe -unlock C: -rp <48-digit BitLocker recovery password>
   manage-bde.exe -protectors -disable C:
   ```

4. Close the Command Prompt window.
5. Shut down the device.
6. Start the device. Windows should start as usual.

After installing UEFI or TPM firmware updates on Surface, BitLocker prompts for the recovery password

Consider the following scenario:

A Surface device has BitLocker drive encryption turned on. The firmware of the Surface's TPM is updated or an update that changes the signature of the system firmware is installed. For example, the Surface TPM (IFX) update is installed.

You experience one or more of the following symptoms on the Surface device:

- At startup, the Surface device prompts for a BitLocker recovery password. The correct recovery password is entered, but Windows doesn't start up.
- Startup progresses directly into the Surface device's Unified Extensible Firmware Interface (UEFI) settings.
- The Surface device appears to be in an infinite restart loop.

Cause of after installing UEFI or TPM firmware updates on Surface, BitLocker prompts for the recovery password

This issue occurs if the Surface device TPM is configured to use Platform Configuration Register (PCR) values other than the default values of PCR 7 and PCR 11. For example, the following settings can configure the TPM this way:

- Secure boot is turned off.
- PCR values have been explicitly defined, such as by group policy.

Devices that support Connected Standby (also known as *InstantGO* or *Always On, Always Connected PCs*), including Surface devices, must use PCR 7 of the TPM. In its default configuration on such systems, BitLocker binds to PCR 7 and PCR 11 if PCR 7 and Secure Boot are correctly configured.

Resolution for after installing UEFI or TPM firmware updates on Surface, BitLocker prompts for the recovery password

To verify the PCR values that are in use on a device, open an elevated Command Prompt window and run the following command:

Windows Command Prompt

```
manage-bde.exe -protectors -get <OSDriveLetter>:
```

In this command, <*OSDriveLetter*> represents the drive letter of the operating system drive.

To resolve this issue and repair the device, follow these steps:

Step 1: Disable the TPM protectors on the boot drive

If a TPM or UEFI update has been installed and the Surface device can't start, even if the correct BitLocker recovery password has been entered, the ability to start can be restored by using the BitLocker recovery password and a Surface recovery image to remove the TPM protectors from the boot drive.

To use the BitLocker recovery password and a Surface recovery image to remove the TPM protectors from the boot drive, follow these steps:

1. Obtain the BitLocker recovery password from the Surface user's Microsoft.com account. If BitLocker is managed by a different method, such as Microsoft BitLocker Administration and Monitoring (MBAM), Configuration Manager BitLocker Management, or Intune, contact the administrator for help.
2. Use another computer to download the Surface recovery image from Surface Recovery Image Download. Use the downloaded image to create a USB recovery drive.
3. Insert the USB Surface recovery image drive into the Surface device, and start the device.
4. When prompted, select the following items:
   a. The operating system language.
   b. The keyboard layout.
5. Select **Troubleshoot** > **Advanced Options** > **Command Prompt**.
6. In the Command Prompt window, run the following commands:

   Windows Command Prompt

   ```
   manage-bde.exe -unlock -recoverypassword <Password> <DriveLetter>:
   manage-bde.exe -protectors -disable <DriveLetter>:
   ```

where:

- <*Password*> is the BitLocker recovery password that was obtained in Step 1
- <*DriveLetter*> is the drive letter that is assigned to the operating system drive

> ⓘ Note
>
> For more information about how to use this command, see manage-bde unlock.

7. Restart the computer.

8. When prompted, enter the BitLocker recovery password that was obtained in Step 1.

> ⓘ Note
>
> After the TPM protectors are disabled, BitLocker drive encryption no longer protects the device. To re-enable BitLocker drive encryption, select **Start**, type **Manage BitLocker**, and then press **Enter**. Follow the steps to encrypt the drive.

Step 2: Use Surface BMR to recover data and reset the Surface device

To recover data from the Surface device if Windows doesn't start, follow steps 1 through 5 of the section Step 1: Disable the TPM protectors on the boot drive to get to a Command Prompt window. Once a Command Prompt window is open, follow these steps:

1. At the command prompt, run the following command:

   Windows Command Prompt

   ```
   manage-bde.exe -unlock -recoverypassword <Password> <DriveLetter>:
   ```

   In this command, <*Password*> is the BitLocker recovery password that was obtained in Step 1 of the section Step 1: Disable the TPM protectors on the boot drive, and <*DriveLetter*> is the drive letter that is assigned to the operating system drive.

2. After the drive is unlocked, use the `copy` or `xcopy.exe` command to copy the user data to another drive.

   > ⓘ Note

> For more information about the these commands, see the Windows commands article.

3. To reset the device by using a Surface recovery image, follow the instructions in the article Creating and using a USB recovery drive for Surface.

Step 3: Restore the default PCR values

To prevent this issue from recurring, it's recommended to restore the default configuration of Secure Boot and the PCR values.

To enable Secure Boot on a Surface device, follow these steps:

1. Suspend BitLocker by opening an elevated Windows PowerShell window and running the following PowerShell cmdlet:

   ```PowerShell
   Suspend-BitLocker -MountPoint "<DriveLetter>:" -RebootCount 0
   ```

   In this command, *<DriveLetter>* is the letter that is assigned to the drive.

2. Restart the device, and then edit the UEFI settings to set the **Secure Boot** option to **Microsoft Only**.

3. Restart the device and sign into Windows.

4. Open an elevated PowerShell window and run the following PowerShell cmdlet:

   ```PowerShell
   Resume-BitLocker -MountPoint "<DriveLetter>:"
   ```

To reset the PCR settings on the TPM, follow these steps:

1. Disable any Group Policy Objects that configure the PCR settings, or remove the device from any groups that enforce such policies.

   For more information, see BitLocker Group Policy settings.

2. Suspend BitLocker by opening an elevated Windows PowerShell window and running the following PowerShell cmdlet:

PowerShell

```
Suspend-BitLocker -MountPoint "<DriveLetter>:" -RebootCount 0
```

In this command, <*DriveLetter*> is the letter that is assigned to the drive.

3. Run the following PowerShell cmdlet:

PowerShell

```
Resume-BitLocker -MountPoint "<DriveLetter>:"
```

Step 4: Suspend BitLocker during TPM or UEFI firmware updates

You can avoid this scenario when installing updates to system firmware or TPM firmware by temporarily suspending BitLocker before applying such updates.

> ⓘ **Important**
>
> TPM and UEFI firmware updates may require multiple restarts while they install. To keep BitLocker suspended during this process, the PowerShell cmdlet Suspend-BitLocker must be used and the **Reboot Count** parameter must be set to either of the following values:
>
> - **2** or greater: This value sets the number of times the device will restart before BitLocker Device Encryption resumes. For example, setting the value to **2** will cause BitLocker to resume after the device restarts twice.
> - **0**: This value suspends BitLocker Drive Encryption indefinitely. To resume BitLocker, the PowerShell cmdlet Resume-BitLocker or another mechanism needs to be used to resume BitLocker protection.

To suspend BitLocker while installing TPM or UEFI firmware updates:

1. Open an elevated Windows PowerShell window and run the following PowerShell cmdlet:

PowerShell

```
Suspend-BitLocker -MountPoint "<DriveLetter>:" -RebootCount 0
```

In this PowerShell cmdlet, <*DriveLetter*> is the letter that is assigned to the drive.

2. Install the Surface device driver and firmware updates.

3. After installing the firmware updates, restart the computer, open an elevated PowerShell window, and then run the following PowerShell cmdlet:

```PowerShell
Resume-BitLocker -MountPoint "<DriveLetter>:"
```

Credential Guard/Device Guard on TPM 1.2: At every restart, BitLocker prompts for the recovery password and returns error 0xC0210000

Consider the following scenario:

A device uses TPM 1.2 and runs Windows 10, version 1809. The device also uses Virtualization-based Security features such as Device Guard and Credential Guard. Every time the device is started, the device enters BitLocker Recovery mode and an error message similar to the following error message is displayed:

> Recovery
>
> Your PC/Device needs to be repaired. A required file couldn't be accessed because your BitLocker key wasn't loaded correctly.
>
> Error code 0xc0210000
>
> You'll need to use recovery tools. If you don't have any installation media (like a disc or USB device), contact your PC administrator or PC/Device manufacturer.

Cause of Credential Guard/Device Guard on TPM 1.2: At every restart, BitLocker prompts for the recovery password and returns error 0xC0210000

TPM 1.2 doesn't support Secure Launch. For more information, see System Guard Secure Launch and SMM protection: Requirements Met by System Guard Enabled Machines

For more information about this technology, see Windows Defender System Guard: How a hardware-based root of trust helps protect Windows

Resolution for Credential Guard/Device Guard on TPM 1.2: At every restart, BitLocker prompts for the recovery password and returns error 0xC0210000

To resolve this issue, use one of the following two solutions:

- Remove any device that uses TPM 1.2 from any group that is subject to GPOs that enforce secure launch.
- Edit the **Turn On Virtualization Based Security** GPO to set **Secure Launch Configuration** to **Disabled**.

BitLocker PIN unlock failure after uninstalling an update

If you uninstall the cumulative update and roll back the OS build to a version earlier than 26100.4770 after installing KB5063878 or a later update, BitLocker may fail to unlock the drive even with the correct PIN on devices running Windows 11, version 24H2.

To resolve this issue, unlock device using recovery password or other recovery mechanism and install KB5062660 or a later update.

Last updated on 02/12/2026

BitLocker configuration: known issues

Applies to: Supported versions of Windows Client

This article describes common issues that affect BitLocker's configuration and general functionality. This article also provides guidance to address these issues.

BitLocker encryption is slower in Windows 10 and Windows 11

BitLocker runs in the background to encrypt drives. However, in Windows 11 and Windows 10, BitLocker is less aggressive about requesting resources than in previous versions of Windows. This behavior reduces the chance that BitLocker will affect the computer's performance.

To compensate for these changes, BitLocker uses a conversion model called Encrypt-On-Write. This model makes sure that any new disk writes are encrypted as soon as BitLocker is enabled. This behavior happens on all client editions and for any internal drives.

> **Important**
>
> To preserve backward compatibility, BitLocker uses the previous conversion model to encrypt removable drives.

Benefits of using the new conversion model

By using the previous conversion model, an internal drive can't be considered protected and compliant with data protection standards until the BitLocker conversion is 100 percent complete. Before the process finishes, the data that existed on the drive before encryption began - that is, potentially compromised data - can still be read and written without encryption. Therefore, for data to be considered protected and compliant with data protection standards, the encryption process has to finish before sensitive data is stored on the drive. Depending on the size of the drive, this delay can be substantial.

By using the new conversion model, sensitive data can be stored on the drive as soon as BitLocker is turned on. The encryption process doesn't need to finish first, and encryption doesn't adversely affect performance. The tradeoff is that the encryption process for pre-existing data takes more time.

Other BitLocker enhancements

Several other areas of BitLocker were improved in versions of Windows released after Windows 7:

- **New encryption algorithm, XTS-AES** - Added in Windows 10 version 1511, this algorithm provides additional protection from a class of attacks on encrypted data that rely on manipulating cipher text to cause predictable changes in plain text.

  By default, this algorithm complies with the Federal Information Processing Standards (FIPS). FIPS is a United States Government standard that provides a benchmark for implementing cryptographic software.

- **Improved administration features**. BitLocker can be managed on PCs or other devices by using the following interfaces:
  - BitLocker Wizard
  - manage-bde.exe
  - Group Policy Objects (GPOs)
  - Mobile Device Management (MDM) policy
  - Windows PowerShell
  - Windows Management Interface (WMI)

- **Integration with Microsoft Entra ID** (Microsoft Entra ID) - BitLocker can store recovery information in Microsoft Entra ID to make it easier to recover.

- Direct memory access (DMA) Port Protection - By using MDM policies to manage BitLocker, a device's DMA ports can be blocked which secures the device during its startup.

- BitLocker Network Unlock - If the BitLocker-enabled desktop or server computer is connected to a wired corporate network in a domain environment, its operating system volume can be automatically unlocked during a system restart.

- **Support for** Encrypted Hard Drives - Encrypted Hard Drives are a new class of hard drives that are self-encrypting at a hardware level and allow for full disk hardware encryption. By taking on that workload, Encrypted Hard Drives increase BitLocker performance and reduce CPU usage and power consumption.

- **Support for classes of HDD/SSD hybrid disks** - BitLocker can encrypt a disk that uses a small SSD as a non-volatile cache in front of the HDD, such as Intel Rapid Storage Technology.

Hyper-V generation 2 VM: Can't access the volume after BitLocker encryption

Consider the following scenario:

1. BitLocker is turned on a generation 2 virtual machine (VM) that runs on Hyper-V.
2. Data is added to the data disk as it encrypts.
3. The VM is restarted and the following behavior is observed:
    - The system volume isn't encrypted.
    - The encrypted volume isn't accessible, and the computer lists the volume's file system as **Unknown**.
    - A message similar to the following message is displayed:

      > **You need to format the disk in <*drive_letter:*> drive before you can use it**

Cause of not being able to access the volume after BitLocker encryption on a Hyper-V generation 2 VM

This issue occurs because the third-party filter driver *Stcvsm.sys* (from StorageCraft) is installed on the VM.

Resolution for not being able to access the volume after BitLocker encryption on a Hyper-V generation 2 VM

To resolve this issue, remove the third-party software.

Production snapshots fail for virtualized domain controllers that use BitLocker-encrypted disks

Consider the following scenario:

A Windows Server 2019 or 2016 Hyper-V Server is hosting VMs (guests) that are configured as Windows domain controllers. On a domain controller guest VM, BitLocker has encrypted the disks that store the Active Directory database and log files. When a "production snapshot" of

the domain controller guest VM is attempted, the Volume Snap-Shot (VSS) service doesn't correctly process the backup.

This issue occurs regardless of any of the following variations in the environment:

- How the domain controller volumes are unlocked.
- Whether the VMs are generation 1 or generation 2.
- Whether the guest operating system is Windows Server 2019, 2016 or 2012 R2.

In the guest VM domain controller **Windows Logs** > **Application** Event Viewer log, the VSS event source records event **ID 8229**:

> ID: 8229
> Level: Warning
> Source: VSS
> Message: A VSS writer has rejected an event with error 0x800423f4. The writer experienced a non-transient error. If the backup process is retried, the error is likely to reoccur.
>
> Changes that the writer made to the writer components while handling the event will not be available to the requester.
>
> Check the event log for related events from the application hosting the VSS writer.
>
> Operation:
> PostSnapshot Event
>
> Context:
> Execution Context: Writer
> Writer Class Id: {b2014c9e-8711-4c5c-a5a9-3cf384484757}
> Writer Name: NTDS
> Writer Instance ID: {d170b355-a523-47ba-a5c8-732244f70e75}
> Command Line: C:\Windows\system32\lsass.exe
>
> Process ID: 680

In the guest VM domain controller **Applications and Services Logs** > **Directory Service** Event Viewer log, there's an event logged similar to the following event:

> Error Microsoft-Windows-ActiveDirectory_DomainService 1168
> Internal Processing Internal error: An Active Directory Domain Services error has occurred.

> Additional Data
> Error value (decimal): -1022
>
> Error value (hex): fffffc02
>
> Internal ID: 160207d9

> **Note**
>
> The internal ID of this event may differ based on the operating system release version and patch level.

When this issue occurs, the **Active Directory Domain Services (NTDS) VSS Writer** will display the following error when the `vssadmin.exe list writers` command is run:

Error

```
Writer name: 'NTDS'
 Writer Id: {b2014c9e-8711-4c5c-a5a9-3cf384484757}
 Writer Instance Id: {08321e53-4032-44dc-9b03-7a1a15ad3eb8}
 State: [11] Failed
 Last error: Non-retryable error
```

Additionally, the VMs can't be backed up until they're restarted.

Cause of production snapshots fail for virtualized domain controllers that use BitLocker-encrypted disks

After VSS creates a snapshot of a volume, the VSS writer takes "post snapshot" actions. When a "production snapshot" is initiated from the host server, Hyper-V tries to mount the snapshotted volume. However, it can't unlock the volume for unencrypted access. BitLocker on the Hyper-V server doesn't recognize the volume. Therefore, the access attempt fails and then the snapshot operation fails.

This behavior is by design.

Workaround for production snapshots fail for virtualized domain controllers that use BitLocker-encrypted disks

A supported way to perform backup and restore of a virtualized domain controller is to run **Windows Server Backup** in the guest operating system.

If a production snapshot of a virtualized domain controller needs to be taken, BitLocker can be suspended in the guest operating system before the production snapshot is started. However, this approach isn't recommended.

For more information and recommendations about backing up virtualized domain controllers, see Virtualizing Domain Controllers using Hyper-V: Backup and Restore Considerations for Virtualized Domain Controllers

More information

When the VSS NTDS writer requests access to the encrypted drive, the Local Security Authority Subsystem Service (LSASS) generates an error entry similar to the following error:

Console

```
\# for hex 0xc0210000 / decimal -1071579136
STATUS\_FVE\_LOCKED\_VOLUME ntstatus.h
\# This volume is locked by BitLocker Drive Encryption.
```

The operation produces the following call stack:

Console

```
\# Child-SP RetAddr Call Site
 00 00000086\`b357a800 00007ffc\`ea6e7a4c KERNELBASE\!FindFirstFileExW+0x1ba \
[d:\\rs1\\minkernel\\kernelbase\\filefind.c @ 872\]
 01 00000086\`b357abd0 00007ffc\`e824accb KERNELBASE\!FindFirstFileW+0x1c \
[d:\\rs1\\minkernel\\kernelbase\\filefind.c @ 208\]
 02 00000086\`b357ac10 00007ffc\`e824afa1 ESENT\!COSFileFind::ErrInit+0x10b \
[d:\\rs1\\onecore\\ds\\esent\\src\\os\\osfs.cxx @ 2476\]
 03 00000086\`b357b700 00007ffc\`e827bf02 ESENT\!COSFileSystem::ErrFileFind+0xa1 \
[d:\\rs1\\onecore\\ds\\esent\\src\\os\\osfs.cxx @ 1443\]
 04 00000086\`b357b960 00007ffc\`e82882a9 ESENT\!JetGetDatabaseFileInfoEx+0xa2 \
[d:\\rs1\\onecore\\ds\\esent\\src\\ese\\jetapi.cxx @ 11503\]
 05 00000086\`b357c260 00007ffc\`e8288166 ESENT\!JetGetDatabaseFileInfoExA+0x59 \
[d:\\rs1\\onecore\\ds\\esent\\src\\ese\\jetapi.cxx @ 11759\]
 06 00000086\`b357c390 00007ffc\`e84c64fb ESENT\!JetGetDatabaseFileInfoA+0x46 \
[d:\\rs1\\onecore\\ds\\esent\\src\\ese\\jetapi.cxx @ 12076\]
 07 00000086\`b357c3f0 00007ffc\`e84c5f23
ntdsbsrv\!CVssJetWriterLocal::RecoverJetDB+0x12f \
[d:\\rs1\\ds\\ds\\src\\jetback\\snapshot.cxx @ 2009\]
 08 00000086\`b357c710 00007ffc\`e80339e0
ntdsbsrv\!CVssJetWriterLocal::OnPostSnapshot+0x293 \
[d:\\rs1\\ds\\ds\\src\\jetback\\snapshot.cxx @ 2190\]
 09 00000086\`b357cad0 00007ffc\`e801fe6d
VSSAPI\!CVssIJetWriter::OnPostSnapshot+0x300 \
[d:\\rs1\\base\\stor\\vss\\modules\\jetwriter\\ijetwriter.cpp @ 1704\]
 0a 00000086\`b357ccc0 00007ffc\`e8022193
```

```
VSSAPI\!CVssWriterImpl::OnPostSnapshotGuard+0x1d \
[d:\\rs1\\base\\stor\\vss\\modules\\vswriter\\vswrtimp.cpp @ 5228\]
 0b 00000086\`b357ccf0 00007ffc\`e80214f0
VSSAPI\!CVssWriterImpl::PostSnapshotInternal+0xc3b \
[d:\\rs1\\base\\stor\\vss\\modules\\vswriter\\vswrtimp.cpp @ 3552\]
```

Last updated on 02/12/2026

BitLocker cannot encrypt a drive: known TPM issues

Applies to: Supported versions of Windows Client

This article describes common issues that affect the Trusted Platform Module (TPM) that might prevent BitLocker from encrypting a drive. This article also provides guidance to address these issues.

> **Note**
>
> If it's been determined that the BitLocker issue does not involve the TPM, see BitLocker cannot encrypt a drive: known issues.

The TPM is locked and the error `The TPM is defending against dictionary attacks and is in a time-out period` is displayed

It's attempted to turn on BitLocker drive encryption on a device but it fails with an error message similar to the following error message:

> **The TPM is defending against dictionary attacks and is in a time-out period.**

Cause of the TPM being locked

The TPM is locked out.

Resolution for the TPM being locked

To resolve this issue, the TPM needs to be reset and cleared. The TPM can be reset and cleared with the following steps:

1. Open an elevated PowerShell window and run the following script:

```
PowerShell
```

```
$Tpm = Get-WmiObject -class Win32_Tpm -namespace
"root\CIMv2\Security\MicrosoftTpm"
$ConfirmationStatus =
$Tpm.GetPhysicalPresenceConfirmationStatus(22).ConfirmationStatus
if($ConfirmationStatus -ne 4) {$Tpm.SetPhysicalPresenceRequest(22)}
```

2. Restart the computer. If a prompt is displayed confirming the clearing of the TPM, agree to clear the TPM.

3. Sign on to Windows and retry starting BitLocker drive encryption.

> ⚠ **Warning**
>
> Resetting and clearing the TPM can cause data loss.

The TPM fails to prepare with the error `The TPM is defending against dictionary attacks and is in a time-out period`

It's attempted to turn on BitLocker drive encryption on a device but it fails. While troubleshooting, the TPM management console (*tpm.msc*) is used to attempt to prepare the TPM on the device. The operation fails with an error message similar to the following error message:

> The TPM is defending against dictionary attacks and is in a time-out period.

Cause of TPM failing to prepare

The TPM is locked out.

Resolution for TPM failing to prepare

To resolve this issue, disable and re-enable the TPM with the following steps:

1. Enter the UEFI/BIOS configuration screens of the device by restarting the device and hitting the appropriate key combination as the device boots. Consult with the device manufacturer for the appropriate key combination for entering into the UEFI/BIOS configuration screens.

2. Once in the UEFI/BIOS configuration screens, disable the TPM. Consult with the device manufacturer for instructions on how to disable the TPM in the UEFI/BIOS configuration screens.

3. Save the UEFI/BIOS configuration with the TPM disabled and restart the device to boot into Windows.

4. Once signed into Windows, return to the TPM management console. An error message similar to the following error message is displayed:

   > Compatible TPM cannot be found
   >
   > Compatible Trusted Platform Module (TPM) cannot be found on this computer. Verify that this computer has 1.2 TPM and it is turned on in the BIOS.

   This message is expected since the TPM is currently disabled in the UEFI firmware/BIOS of the device.

5. Restart the device and enter the UEFI/BIOS configuration screens again.

6. Reenable the TPM in the UEFI/BIOS configuration screens.

7. Save the UEFI/BIOS configuration with the TPM enabled and restart the device to boot into Windows.

8. Once signed into Windows, return to the TPM management console.

If the TPM still can't be prepared, clear the existing TPM keys by following the instructions in the article Troubleshoot the TPM: Clear all the keys from the TPM.

> ⚠ **Warning**
>
> Clearing the TPM can cause data loss.

BitLocker fails to enable with the error `Access Denied: Failed to backup TPM Owner Authorization information to Active Directory Domain Services. Errorcode: 0x80070005` or `Insufficient Rights`

The **Do not enable BitLocker until recovery information is stored in AD DS** policy is enforced in the environment. It's attempted to turn on BitLocker drive encryption on a device but it fails with the error message of `Access Denied: Failed to backup TPM Owner Authorization information to Active Directory Domain Services. Errorcode: 0x80070005` or `Insufficient Rights`.

Cause of `Access Denied` or `Insufficient Rights`

The TPM didn't have sufficient permissions on the TPM devices container in Active Directory Domain Services (AD DS). Therefore, the BitLocker recovery information couldn't be backed up to AD DS, and BitLocker drive encryption couldn't turn on.

This issue appears to be limited to computers that run versions of Windows that are earlier than Windows 10.

Resolution for `Access Denied` or `Insufficient Rights`

To verify this issue is occurring, use one of the following two methods:

- Disable the policy or remove the computer from the domain followed by trying to turn on BitLocker drive encryption again. If the operation succeeds, then the issue was caused by the policy.
- Use LDAP and network trace tools to examine the LDAP exchanges between the client and the AD DS domain controller to identify the cause of the **Access Denied** or **Insufficient Rights** error. In this case, an error should be displayed when the client tries to access its object in the `CN=TPM Devices,DC=<domain>,DC=com` container.

1. To review the TPM information for the affected computer, open an elevated Windows PowerShell window and run the following command:

   PowerShell

   ```
   Get-ADComputer -Filter {Name -like "ComputerName"} -Property * | Format-Table name,msTPM-TPMInformationForComputer
   ```

   In this command, *ComputerName* is the name of the affected computer.

2. To resolve the issue, use a tool such as *dsacls.exe* to ensure that the access control list of msTPM-TPMInformationForComputer grants both **Read** and **Write** permissions to **NTAUTHORITY/SELF**.

The TPM fails to be prepared with the error `0x80072030: There is no such object on the server`

Domain controllers were upgraded from Windows Server 2008 R2 to Windows Server 2012 R2. A group policy object (GPO) exists that enforces the **Do not enable BitLocker until recovery information is stored in AD DS** policy.

It's attempted to turn on BitLocker drive encryption on a device but it fails. While troubleshooting, the TPM management console (*tpm.msc*) is used to attempt to prepare the TPM on the device. The operation fails with an error message similar to the following error message:

> 0x80072030 There is no such object on the server when a policy to back up TPM information to active directory is enabled

It's been confirmed that the **ms-TPM-OwnerInformation** and **msTPM-TpmInformationForComputer** attributes are present.

Cause of 0x80072030: There is no such object on the server

The domain and forest functional level of the environment may still be set to Windows 2008 R2. Additionally, the permissions in AD DS might not be correctly set.

Resolution for 0x80072030: There is no such object on the server

The issue can be resolved with the following steps:

1. Upgrade the functional level of the domain and forest to Windows Server 2012 R2.
2. Download Add-TPMSelfWriteACE.vbs.
3. In the script, modify the value of **strPathToDomain** to the organization's domain name.
4. Open an elevated PowerShell window, and run the following command:

   Windows Command Prompt

   ```
   cscript.exe <Path>\Add-TPMSelfWriteACE.vbs
   ```

   In this command, <*Path*> is the path to the script file.

For more information, see the following articles:

- Back up the TPM recovery information to AD DS
- Prepare your organization for BitLocker: Planning and policies

---

Last updated on 02/12/2026

BitLocker and TPM: other known issues

Applies to: Supported versions of Windows Client

This article describes common issues that relate directly to the trusted platform module (TPM), and provides guidance to address these issues.

Microsoft Entra ID: Windows Hello for Business and single sign-on don't work

Consider the following scenario:

A Microsoft Entra joined client computer can't authenticate correctly. The computer is experiencing one or more of the following symptoms:

- Windows Hello for Business doesn't work
- Conditional access fails
- Single sign-on (SSO) doesn't work

Additionally, in Event Viewer, the computer logs the following Event ID 1026 event under **Windows Logs** > **System**:

> Log Name: System
> Source: Microsoft-Windows-TPM-WMI
> Date: <Date and Time>
> Event ID: 1026
> Task Category: None
> Level: Information
> Keywords:
> User: SYSTEM
> Computer: <Computer name>
> Description:
> The Trusted Platform Module (TPM) hardware on this computer cannot be provisioned for use automatically. To set up the TPM interactively use the TPM management console (Start->tpm.msc) and use the action to make the TPM ready.
> Error: The TPM is defending against dictionary attacks and is in a time-out period.
> Additional Information: 0x840000

Cause of Microsoft Entra ID: Windows Hello for Business and single sign-on don't work

This event indicates that the TPM isn't ready or has some setting that prevents access to the TPM keys.

Additionally, the behavior indicates that the client computer can't obtain a Primary Refresh Token (PRT).

Resolution for Microsoft Entra ID: Windows Hello for Business and single sign-on don't work

To verify the status of the PRT, use the dsregcmd.exe /status command to collect information. In the tool output, verify that either **User state** or **SSO state** contains the **AzureAdPrt** attribute. If the value of this attribute is **No**, the PRT wasn't issued. If the value of the attribute is **No**, it may indicate that the computer couldn't present its certificate for authentication.

To resolve this issue, follow these steps to troubleshoot the TPM:

1. Open the TPM management console (*tpm.msc*) by selecting **Start** and entering **tpm.msc** in the **Search** box.
2. If a notice is displayed to either unlock the TPM or reset the lockout, contact the hardware vendor to determine whether there's a known fix for the issue.
3. If the issue is still not resolved after contacting the hardware vendor, clear and reinitialize the TPM by following the instructions in the article Troubleshoot the TPM: Clear all the keys from the TPM.

   > ⚠ **Warning**
   >
   > Clearing the TPM can cause data loss.

If in Step 2 there's no notice to either unlock the TPM or reset the lockout, review the UEFI firmware/BIOS settings of the computer for any setting that can be used to reset or disable the lockout.

TPM 1.2 Error: Loading the management console failed. The device that is required by the

cryptographic provider isn't ready for use

Consider the following scenario:

When trying to open the TPM management console on a Windows computer that uses TPM version 1.2, the following message is displayed:

> Loading the management console failed. The device that is required by the cryptographic provider is not ready for use.
> HRESULT 0x800900300x80090030 - NTE_DEVICE_NOT_READY
> The device that is required by this cryptographic provider is not ready for use.
> TPM Spec version: TPM v1.2

On a different device that is running the same version of Windows, the TPM management console can be opened.

Cause (suspected) of TPM 1.2 Error: Loading the management console failed. The device that is required by the cryptographic provider isn't ready for use

These symptoms indicate that the TPM has hardware or firmware issues.

Resolution for TPM 1.2 Error: Loading the management console failed. The device that is required by the cryptographic provider isn't ready for use

To resolve the issue:

- Switch the TPM operating mode from version 1.2 to version 2.0 if the device has this option available.
- If switching the TPM from version 1.2 to version 2.0 doesn't resolve the issue, or if the device doesn't have TPM version 2.0 available, contact the hardware vendor to determine whether there's a UEFI firmware update/BIOS update/TPM update for the device. If there's an update available, install the update to see if it resolves the issue.
- If updating the UEFI firmware/BIOS doesn't resolve the issue, or if there's no update available, consider replacing the device motherboard by contacting the hardware vendor.

After the motherboard has been replaced, switch the TPM operating mode from version 1.2 to version 2.0 if this option is available.

> ⚠ **Warning**
>
> Replacing the motherboard will cause data in the TPM to be lost.

Devices don't join hybrid Microsoft Entra ID because of a TPM issue

When trying to join a device to a hybrid Microsoft Entra ID, the join operation appears to fail.

To verify that the join succeeded, use the dsregcmd /status command. In the tool output, the following attributes indicate that the join succeeded:

- **AzureAdJoined: YES**
- **DomainName: <*on-prem Domain name*>**

If the value of **AzureADJoined** is **No**, the join operation failed.

Causes and resolutions for devices don't join hybrid Microsoft Entra ID because of a TPM issue

This issue may occur when the Windows operating system isn't the owner of the TPM. The specific fix for this issue depends on which errors or events are displayed, as shown in the following table:

Expand table

Message	Reason	Resolution
NTE_BAD_KEYSET (0x80090016/-2146893802)	TPM operation failed or was invalid	This issue was probably caused by a corrupted sysprep image. When creating a sysprep image, make sure to use a computer that isn't joined to or registered in Microsoft Entra ID or hybrid Microsoft Entra ID.
TPM_E_PCP_INTERNAL_ERROR (0x80290407/-2144795641)	Generic TPM error.	If the device returns this error, disable its TPM. Windows 10, version 1809 and later versions, automatically detect TPM failures

Message	Reason	Resolution
		and finish the Microsoft Entra hybrid join without using the TPM.
TPM_E_NOTFIPS (0x80280036/-2144862154)	The FIPS mode of the TPM is currently not supported.	If the device gives this error, disable its TPM. Windows 10, version 1809 and later versions, automatically detect TPM failures and finish the Microsoft Entra hybrid join without using the TPM.
NTE_AUTHENTICATION_IGNORED (0x80090031/-2146893775)	The TPM is locked out.	This error is transient. Wait for the cooldown period, and then retry the join operation.

For more information about TPM issues, see the following articles:

- TPM fundamentals: Anti-hammering
- Troubleshooting Microsoft Entra hybrid joined devices
- Troubleshoot the TPM

Last updated on 02/12/2026

Decode Measured Boot logs to track PCR changes

Applies to: Supported versions of Windows Client

Platform Configuration Registers (PCRs) are memory locations in the Trusted Platform Module (TPM). BitLocker and its related technologies depend on specific PCR configurations. Additionally, specific change in PCRs can cause a device or computer to enter BitLocker recovery mode.

By tracking changes in the PCRs, and identifying when they changed, insight can be gained into issues that occur or learn why a device or computer entered BitLocker recovery mode. The Measured Boot logs record PCR changes and other information. These logs are located in the *C:\Windows\Logs\MeasuredBoot* folder.

This article describes tools that can be used to decode these logs:

- TBSLogGenerator.exe
- PCPTool.exe

For more information about Measured Boot and PCRs, see the following articles:

- TPM fundamentals: Measured Boot with support for attestation
- Understanding PCR banks on TPM 2.0 devices

Use TBSLogGenerator.exe to decode Measured Boot logs

Use *TBSLogGenerator.exe* to decode Measured Boot logs that were collected from Windows. *TBSLogGenerator.exe* can be installed on the following systems:

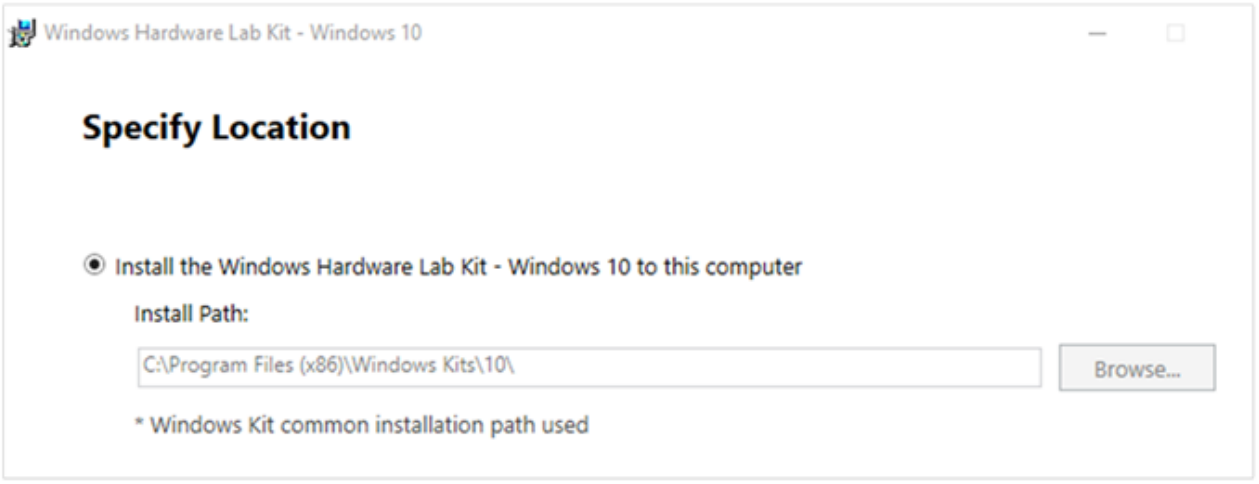
- A computer that is running Windows Server 2016 or newer and that has a TPM enabled
- A Gen 2 virtual machine running on Hyper-V that is running Windows Server 2016 or newer and is using a virtual TPM.

To install the tool, follow these steps:

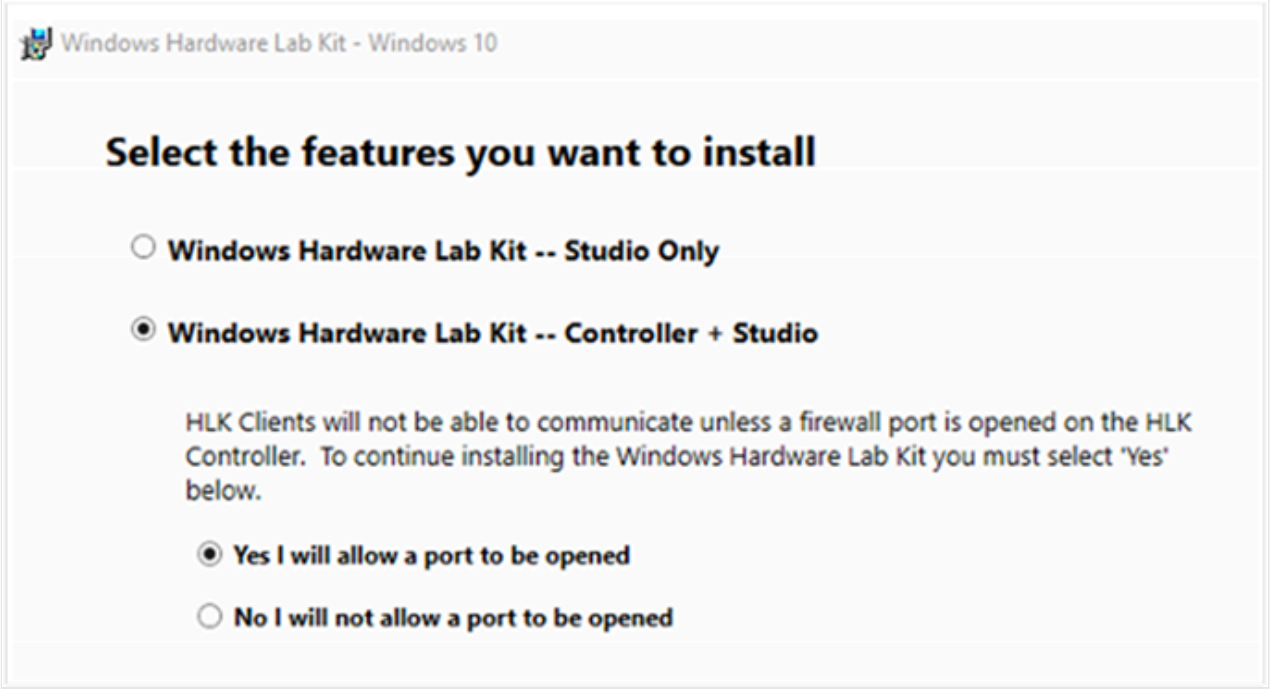
1. Download the Windows Hardware Lab Kit from Windows Hardware Lab Kit.

2. After downloading, run the installation file from the path where the install was downloaded to.

3. Accept the default installation path.



4. Under **Select the features you want to install**, select **Windows Hardware Lab Kit—Controller + Studio**.



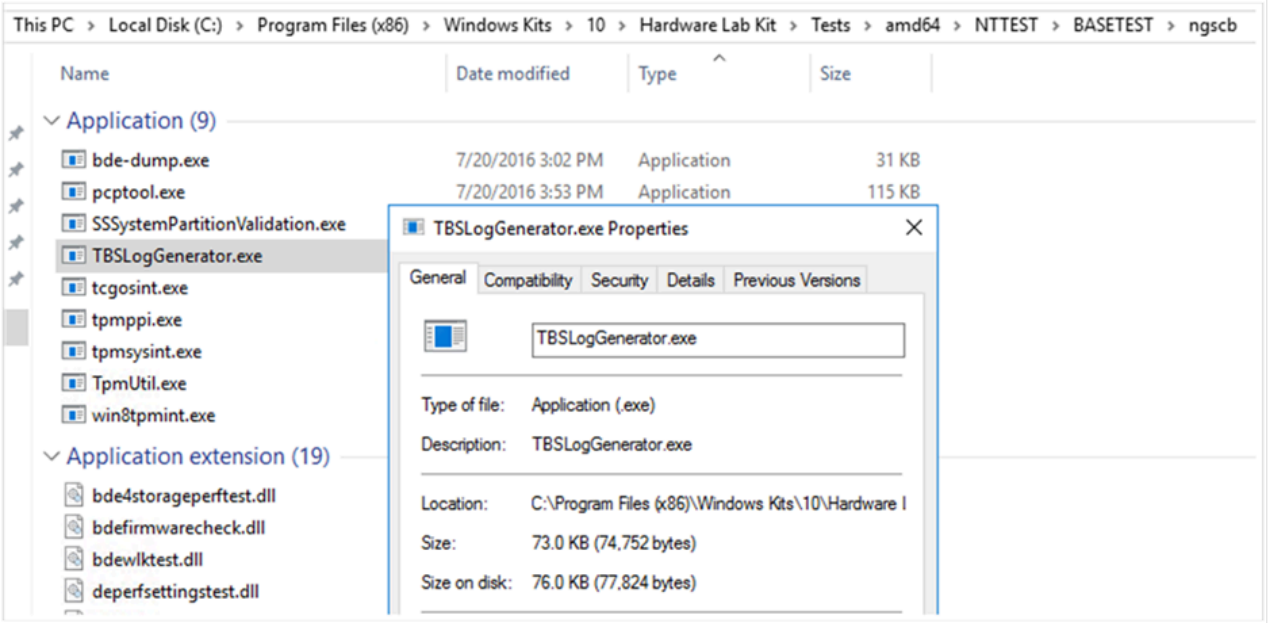
5. Finish the installation.

To use *TBSLogGenerator.exe*, follow these steps:

1. After the installation finishes, open an elevated Command Prompt window and navigate to the following folder:

   C:\Program Files (x86)\Windows Kits\10\Hardware Lab Kit\Tests\amd64\NTTEST\BASETEST\ngscb

This folder contains the *TBSLogGenerator.exe* file.



2. Run the following command:

Windows Command Prompt

```
TBSLogGenerator.exe -LF <LogFolderName>\<LogFileName>.log >
<DestinationFolderName>\<DecodedFileName>.txt
```

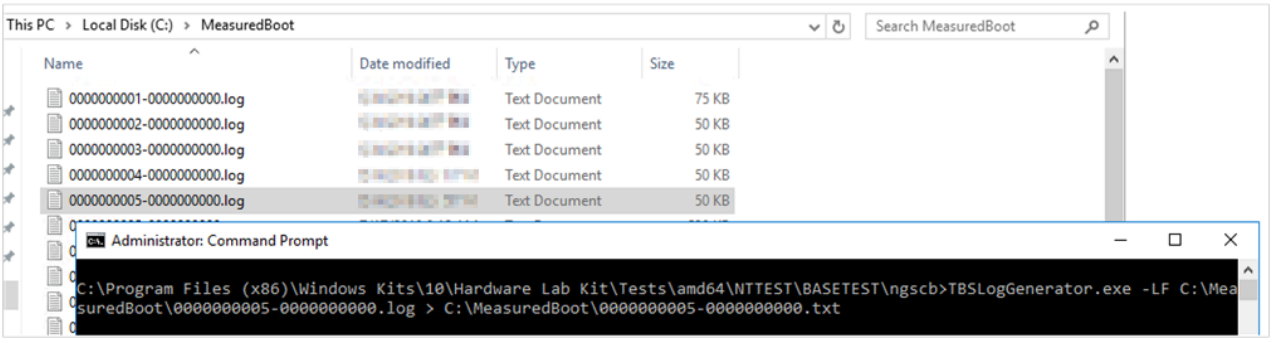
where the variables represent the following values:

- <*LogFolderName*> = the name of the folder that contains the file to be decoded
- <*LogFileName*> = the name of the file to be decoded
- <*DestinationFolderName*> = the name of the folder for the decoded text file
- <*DecodedFileName*> = the name of the decoded text file

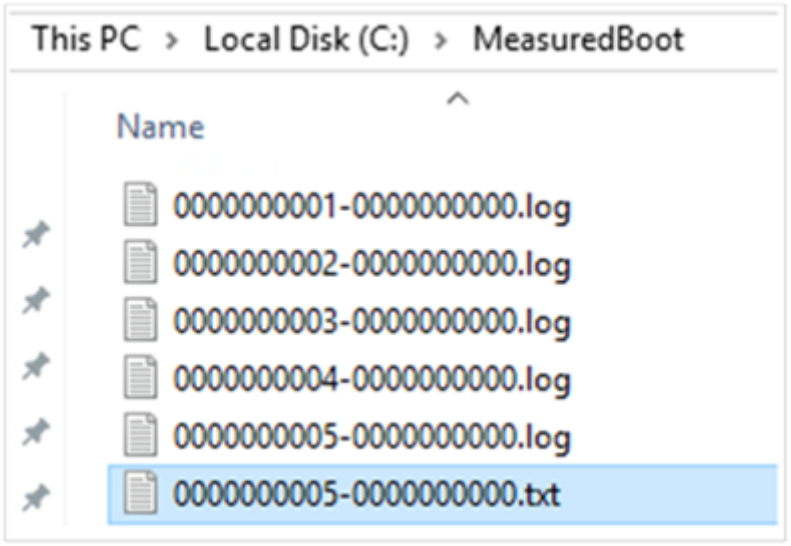
For example, the following figure shows Measured Boot logs that were collected from a Windows 10 computer and put into the *C:\MeasuredBoot* folder. The figure also shows a Command Prompt window and the command to decode the *0000000005-0000000000.log* file:

Windows Command Prompt

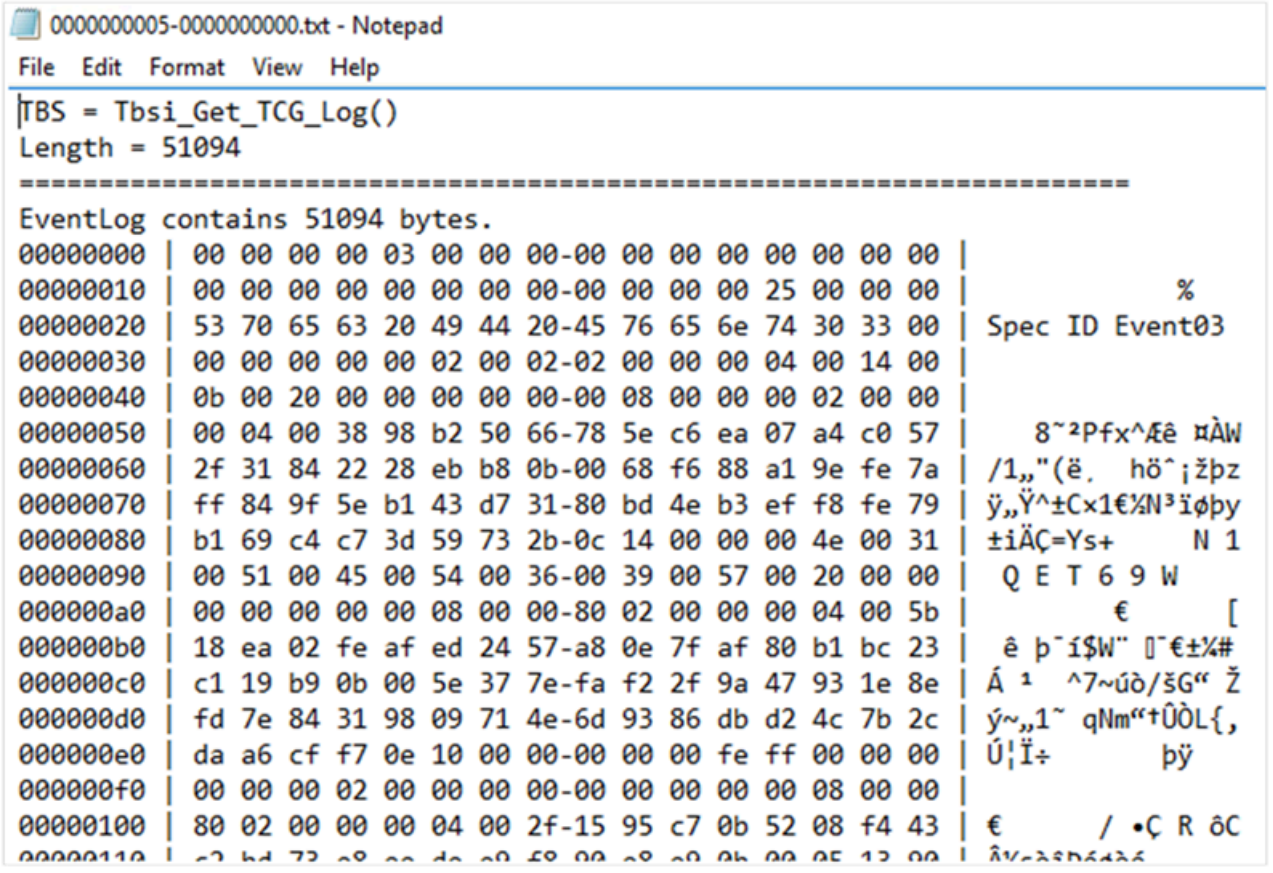
```
TBSLogGenerator.exe -LF C:\MeasuredBoot\0000000005-0000000000.log >
C:\MeasuredBoot\0000000005-0000000000.txt
```



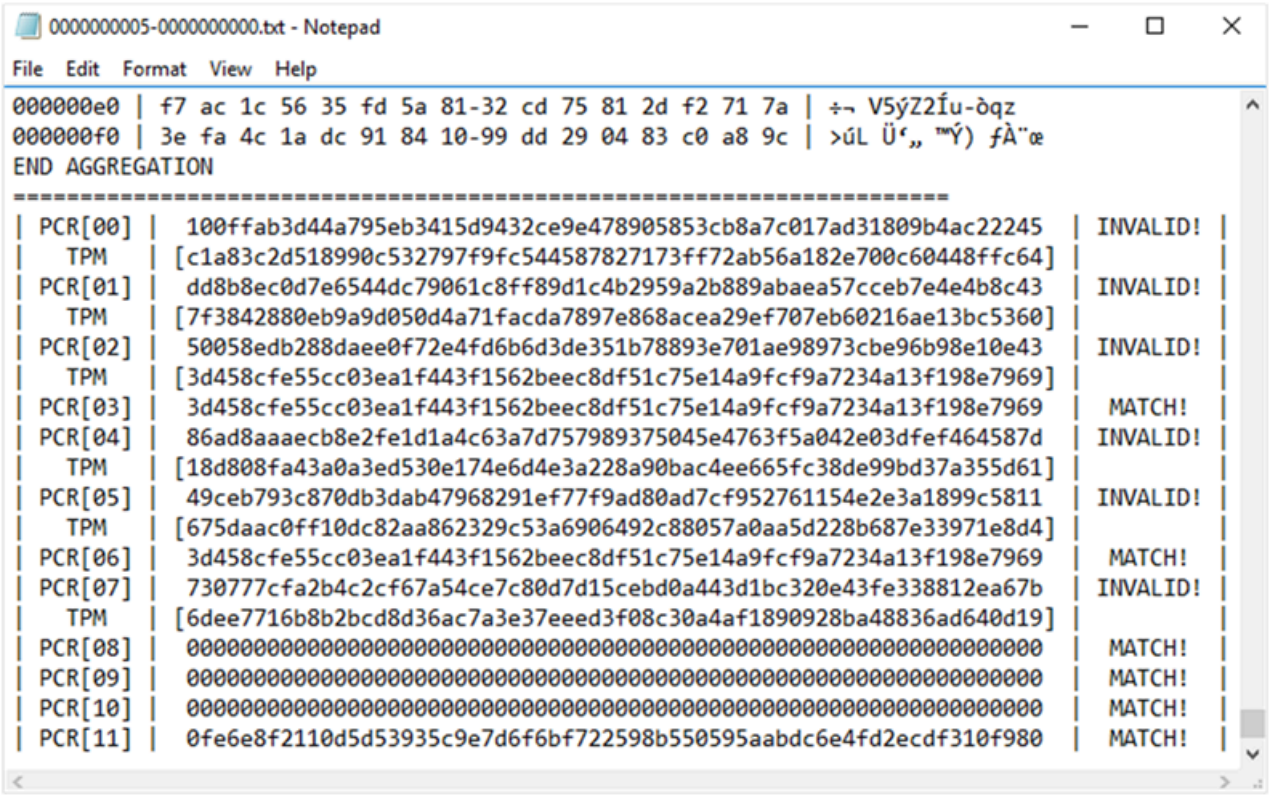
The command produces a text file that uses the specified name. In this example, the file is *0000000005-0000000000.txt*. The file is located in the same folder as the original *.log* file.



The content of this text file is similar to the following text:



To find the PCR information, go to the end of the file.



Use PCPTool.exe to decode Measured Boot logs

> ⓘ **Note**
>
> *PCPTool.exe* is a Visual Studio solution, but executable needs to be built before tool can be used.

PCPTool.exe is part of the TPM Platform Crypto-Provider Toolkit. The tool decodes a Measured Boot log file and converts it into an XML file.

To download and install *PCPTool.exe*, go to the Toolkit page, select **Download**, and follow the instructions.

To decode a log, run the following command:

Windows Command Prompt

```
PCPTool.exe decodelog <LogFolderPath>\<LogFileName>.log > <DestinationFolderName>\
<DecodedFileName>.xml
```

where the variables represent the following values:

- <*LogFolderPath*> = the path to the folder that contains the file to be decoded
- <*LogFileName*> = the name of the file to be decoded
- <*DestinationFolderName*> = the name of the folder for the decoded text file
- <*DecodedFileName*> = the name of the decoded text file

The content of the XML file will be similar to the following XML:

```
> .\PCPTool.exe decodelog C:\Windows\Logs\MeasuredBoot\0000000001-0000000000.log > TPM_log.xml
```

```
129      </EV_Event_Tag>
130      <EV_Event_Tag PCR="13" EventDigest="3db51a2ab6859f67510849226e545d7a193b66eb" Size="760">
131        <Trustboundary Size="752">
132          <SipaEvent Type="0x00040002" Size="46">
133            00ad31d5660ed001200000000b00f4247831deed0fc371f8e8a6b0746ae7b508c3c924e5b1c485c14631afa33879
134            <!-- ..1.f..........x1....q....tj..........F1..8y -->
135          </SipaEvent>
136          <LoadedModule_Aggregation Size="354">
137            <FilePath Size="86">\EFI\Microsoft\Boot\en-GB\bootmgfw.efi.MUI</FilePath>
138            <ImageSize Size="8">69632<!-- 0x0000000000011000 --></ImageSize>
139            <HashAlgorithmId Size="4">SHA-256</HashAlgorithmId>
140            <AuthenticodeHash Size="32">
141              a1cd73aee5d6f2c87e57573f0e50b8ab7f512dcf7c544df8ae231d492d5835b8
142              <!-- ..s......WW..P...Q...TM....I.X5. -->
143            </AuthenticodeHash>
144            <ImageValidated Size="1">TRUE</ImageValidated>
145            <AuthorityIssuer Size="76">Microsoft Windows Production PCA 2011</AuthorityIssuer>
146            <AuthorityPublisher Size="36">Microsoft Windows</AuthorityPublisher>
```

Last updated on 02/12/2026

Encrypted hard drives

Article • 02/25/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Overview

Encrypted hard drives are a class of hard drives that are self-encrypted at the hardware level and allow for full disk hardware encryption while being transparent to the user. These drives combine the security and management benefits provided by BitLocker Drive Encryption with the power of self-encrypting drives.

By offloading the cryptographic operations to hardware, encrypted hard drives increase BitLocker performance and reduce CPU usage and power consumption. Because encrypted hard drives encrypt data quickly, BitLocker deployment can be expanded across enterprise devices with little to no impact on productivity.

Encrypted hard drives provide:

- **Better performance**: Encryption hardware, integrated into the drive controller, allows the drive to operate at full data rate with no performance degradation
- **Strong security based in hardware**: Encryption is always *on* and the keys for encryption never leave the hard drive. User authentication is performed by the drive before it unlocks, independently of the operating system
- **Ease of use**: Encryption is transparent to the user, and the user doesn't need to enable it. Encrypted hard drives are easily erased using on-board encryption key; there's no need to re-encrypt data on the drive
- **Lower cost of ownership**: There's no need for new infrastructure to manage encryption keys, since BitLocker uses your existing infrastructure to store recovery information. Your device operates more efficiently because processor cycles don't need to be used for the encryption process

Encrypted hard drives are supported natively in the operating system through the following mechanisms:

- **Identification**: The operating system identifies that the drive is an *encrypted hard drive* device-type
- **Activation**: The operating system disk management utility activates, creates and maps volumes to ranges/bands as appropriate
- **Configuration**: The operating system creates and maps volumes to ranges/bands as appropriate

- **API**: API support for applications to manage encrypted hard drives independent of BitLocker drive encryption
- **BitLocker support**: Integration with the BitLocker Control Panel provides a seamless BitLocker user experience

> ⚠ **Warning**
>
> *Self-encrypting hard drives* and *encrypted hard drives* for Windows are not the same type of devices:
>
> - encrypted hard drives for Windows require compliance for specific TCG protocols as well as IEEE 1667 compliance
> - self-encrypting hard drives don't have these requirements
>
> It's important to confirm that the device type is an encrypted hard drive for Windows when planning for deployment.

When the operating system identifies an encrypted hard drive, it activates the *security mode*. This activation lets the drive controller generate a *media key* for every volume that the host computer creates. The media key, which is never exposed outside the disk, is used to rapidly encrypt or decrypt every byte of data that is sent or received from the disk.

If you're a storage device vendor who is looking for more info on how to implement encrypted hard drive, see the encrypted hard drive device guide.

System Requirements

To use encrypted hard drives, the following system requirements apply:

For an encrypted hard drive used as a **data drive**:

- The drive must be in an uninitialized state
- The drive must be in a security inactive state

For an encrypted hard drive used as a **startup drive**:

- The drive must be in an uninitialized state
- The drive must be in a security inactive state
- The computer must be UEFI 2.3.1 based and have the `EFI\_STORAGE\_SECURITY\_COMMAND\_PROTOCOL` defined. This protocol is used to allow

programs running in the EFI boot services environment to send security protocol commands to the drive
- The computer must have the compatibility support module (CSM) disabled in UEFI
- The computer must always boot natively from UEFI

> ⚠ **Warning**
>
> All encrypted hard drives must be attached to non-RAID controllers to function properly.

Windows edition and licensing requirements

The following table lists the Windows editions that support Encrypted hard drive:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Encrypted hard drive license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Configure encrypted hard drives as startup drives

To configure encrypted hard drives as startup drives, use the same methods as standard hard drives:

- **Deploy from media**: configuration of encrypted hard drives happens automatically through the installation process
- **Deploy from network**: this deployment method involves booting a Windows PE environment and using imaging tools to apply a Windows image from a network

share. With this method, the Enhanced Storage optional component needs to be included in the Windows PE image. Enable this component using Server Manager, Windows PowerShell, or the DISM command line tool. If the component isn't present, configuration of encrypted hard drives doesn't work

- **Deploy from server**: this deployment method involves PXE booting a client with encrypted hard drives present. Configuration of encrypted hard drives happens automatically in this environment when the Enhanced Storage component is added to the PXE boot image. During deployment, the TCGSecurityActivationDisabled setting in `unattend.xml` controls the encryption behavior of encrypted hard drives
- **Disk duplication**: this deployment method involves use of a previously configured device and disk duplication tools to apply a Windows image to an encrypted hard drive. Images made using disk duplicators don't work

Configure hardware-based encryption with policy settings

There are three policy settings to manage how BitLocker uses hardware-based encryption and which encryption algorithms to use. If these settings aren't configured or disabled on systems that are equipped with encrypted drives, BitLocker uses software-based encryption:

- Configure use of hardware-based encryption for fixed data drives
- Configure use of hardware-based encryption for removable data drives
- Configure use of hardware-based encryption for operating system drives

Encrypted hard drive architecture

Encrypted hard drives utilize two encryption keys on the device to control the locking and unlocking of data on the drive. These encryption keys are the *Data Encryption Key* (DEK) and the *Authentication Key* (AK):

- the Data Encryption Key is used to encrypt all the data on the drive. The drive generates the DEK and it never leaves the device. It's stored in an encrypted format at a random location on the drive. If the DEK is changed or erased, data encrypted using the DEK is irrecoverable.
- the AK is the key used to unlock data on the drive. A hash of the key is stored on the drive and requires confirmation to decrypt the DEK

When a device with an encrypted hard drive is in a powered-off state, the drive locks automatically. As a device powers on, the device remains in a locked state and is only unlocked after the AK decrypts the DEK. Once the AK decrypts the DEK, read-write operations can take place on the device.

When data is written to the drive, it passes through an encryption engine before the write operation completes. Likewise, reading data from the drive requires the encryption engine to decrypt the data before passing that data back to the user. If the AK needs to be changed or erased, the data on the drive doesn't need to be re-encrypted. A new Authentication Key needs to be created and it re-encrypts the DEK. Once completed, the DEK can now be unlocked using the new AK, and read-writes to the volume can continue.

Reconfigure encrypted hard drives

Many encrypted hard drive devices come preconfigured for use. If reconfiguration of the drive is required, use the following procedure after removing all available volumes and reverting the drive to an uninitialized state:

1. Open Disk Management (`diskmgmt.msc`)
2. Initialize the disk and select the appropriate partition style (MBR or GPT)
3. Create one or more volumes on the disk.
4. Use the BitLocker setup wizard to enable BitLocker on the volume.

Feedback

Was this page helpful? Yes No

Provide product feedback

Personal Data Encryption overview

Article • 03/13/2025 • Applies to: ✅ Windows 11

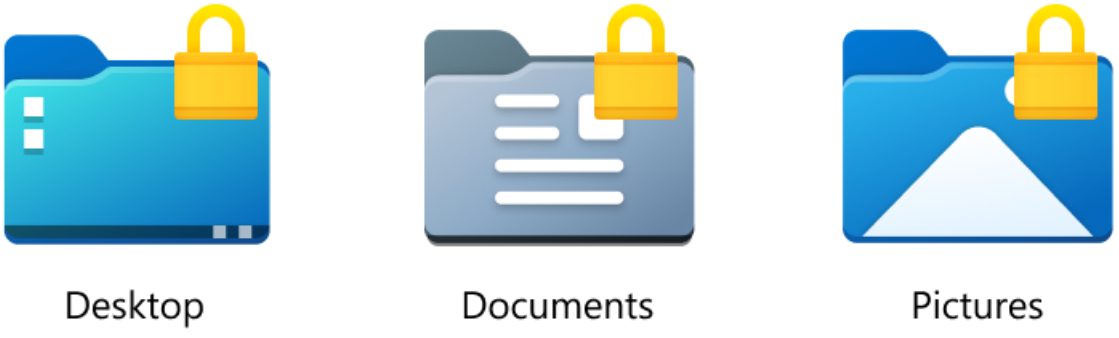
Personal Data Encryption is a security feature that provides file-based data encryption capabilities to Windows. It utilizes Windows Hello for Business to link *data encryption keys* with user credentials. When a user signs in to a device using Windows Hello, decryption keys are released, and encrypted data becomes accessible to the user. Conversely, when a user signs out, decryption keys are discarded, rendering the data inaccessible even if another user signs into the device. This ensures that sensitive information remains always protected.

The benefits of Personal Data Encryption are significant. By reducing the number of credentials needed to access encrypted content, users only need to sign in with Windows Hello. Additionally, the accessibility features available with Windows Hello extend to Personal Data Encryption protected content.

Unlike BitLocker, which encrypts entire volumes and disks, Personal Data Encryption focuses on individual files, providing another layer of security. This feature not only enhances data protection but also shows a strong commitment to safeguarding personal information.

Personal Data Encryption for known folders

Starting in Windows 11, version 24H2, Personal Data Encryption is further enhanced with *Personal Data Encryption for known folders*. Once enabled, the Windows folders **Desktop**, **Documents**, and **Pictures**, along with their contents, are automatically encrypted. This feature provides a quick and easy way to add an extra layer of security to commonly used folders.



Prerequisites

To use Personal Data Encryption, the following prerequisites must be met:

- Windows 11, version 22H2 and later
  - Personal Data Encryption for known folders is only available on Windows 11, version 24H2 and later
- The devices must be Microsoft Entra joined or Microsoft Entra hybrid joined. Domain-joined devices aren't supported
- Users must sign in using Windows Hello

> **Important**
>
> If you sign in with a password or a FIDO2 security key, you can't access Personal Data Encryption protected content.

Windows edition and licensing requirements

The following table lists the Windows editions that support Personal Data Encryption:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
No	Yes	No	Yes

Personal Data Encryption license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
No	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Personal Data Encryption protection levels

Personal Data Encryption uses *AES-CBC* with a *256-bit key* to protect content and offers two levels of protection. The level of protection is determined based on the organizational needs. These levels can be set via the Personal Data Encryption APIs.

Expand table

Item	Level 1	Level 2
Protected data accessible when user signs in via Windows Hello	Yes	Yes
Protected data is accessible at Windows lock screen	Yes	Data is accessible for one minute after lock, then it's no longer available
Protected data is accessible after user signs out of Windows	No	No
Protected data is accessible when device is shut down	No	No
Protected data is accessible via UNC paths	No	No
Protected data is accessible when signing with Windows password instead of Windows Hello	No	No
Protected data is accessible via Remote Desktop session	No	No
Decryption keys used by Personal Data Encryption discarded	After user signs out of Windows	One minute after Windows lock screen is engaged or after user signs out of Windows

Personal Data Encryption protected content accessibility

When a file is protected with Personal Data Encryption, its icon shows a padlock. If the user isn't signed in locally with Windows Hello, or an unauthorized user attempts to access protected content, they're denied access.

Scenarios where a user is denied access to Personal Data Encryption protected content include:

- User signs in with a password instead of using Windows Hello (biometrics or PIN)
- If protected via level 2 protection, when the device is locked
- When trying to access content on the device remotely. For example, UNC network paths
- Remote Desktop sessions

- Other users on the device who aren't owners of the content, even if they're signed in via Windows Hello and have permissions to navigate to the Personal Data Encryption protected content

Differences between Personal Data Encryption and BitLocker

Personal Data Encryption is meant to work alongside BitLocker. Personal Data Encryption isn't a replacement for BitLocker, nor is BitLocker a replacement for Personal Data Encryption. Using both features together provides better security than using either BitLocker or Personal Data Encryption alone. However there are differences between BitLocker and Personal Data Encryption and how they work. These differences are why using them together offers better security.

Expand table

	Personal Data Encryption	**BitLocker**
Release of decryption key	At user sign-in via Windows Hello	At boot
Decryption keys discarded	When user signs out of Windows or one minute after Windows lock screen is engaged	At shutdown
Protected content	All files in protected folders	Entire volume/drive
Authentication to access protected content	Windows Hello for Business	When BitLocker with TPM + PIN is enabled, BitLocker PIN plus Windows sign-in

Differences between Personal Data Encryption and EFS

The main difference between protecting files with Personal Data Encryption instead of EFS is the method they use to protect the file. Personal Data Encryption uses Windows Hello for Business to secure the keys that protect the files. EFS uses certificates to secure and protect the files.

To see if a file is protected with Personal Data Encryption or with EFS:

1. Open the properties of the file
2. Under the **General** tab, select **Advanced...**

3. In the **Advanced Attributes** windows, select **Details**

For Personal Data Encryption protected files, under **Protection status:** there's an item listed as **Personal Data Encryption is: On**.

For EFS protected files, under **Users who can access this file:**, there's a **Certificate thumbprint** next to the users with access to the file. There's also a section labeled **Recovery certificates for this file as defined by recovery policy:**.

Encryption information including what encryption method is being used to protect the file can be obtained with the cipher.exe /c command.

Recommendations for using Personal Data Encryption

The following are recommendations for using Personal Data Encryption:

- Enable BitLocker Drive Encryption. Although Personal Data Encryption works without BitLocker, it's recommended to enable BitLocker. Personal Data Encryption is meant to work alongside BitLocker for increased security at it isn't a replacement for BitLocker
- Backup solution such as OneDrive in Microsoft 365. In certain scenarios, such as TPM resets or destructive PIN resets, the keys used by Personal Data Encryption to protect content will be lost making any protected content inaccessible. The only way to recover such content is from a backup. If the files are synced to OneDrive, to regain access you must resync OneDrive
- Windows Hello for Business PIN reset service. Destructive PIN reset causes keys used by Personal Data Encryption to protect content to be lost, making any content protected with Personal Data Encryption inaccessible. After a destructive PIN reset, content protected with Personal Data Encryption must be recovered from a backup. For this reason, Windows Hello for Business PIN reset service is recommended since it provides nondestructive PIN resets
- Windows Hello Enhanced Sign-in Security offers more security when authenticating with Windows Hello via biometrics or PIN

Next steps

- Learn about the available options to configure Personal Data Encryption and how to configure them via Microsoft Intune or configuration Service Provider (CSP): Personal Data Encryption settings and configuration
- Review the Personal Data Encryption FAQ

Feedback

Was this page helpful? Yes No

Provide product feedback

Personal Data Encryption settings and configuration

Article • 03/13/2025 • Applies to: ✅ Windows 11

This article describes the Personal Data Encryption settings and how to configure them via Microsoft Intune or Configuration Service Providers (CSP).

> **Note**
>
> Personal Data Encryption can be configured using CSP policies. The content to be protected by Personal Data Encryption can be specified using Personal Data Encryption for known folders and Personal Data Encryption APIs.
>
> The Personal Data Encryption APIs can be used to create custom applications and scripts to specify which content to protect and at what level to protect the content. Additionally, the Personal Data Encryption APIs can't be used to protect content until the Personal Data Encryption policy is enabled.

Personal Data Encryption settings

The following table lists the required settings to enable Personal Data Encryption.

Expand table

Setting name	Description
Enable Personal Data Encryption	Personal Data Encryption isn't enabled by default. Before Personal Data Encryption can be used, you must enable it.
Sign-in and lock last interactive user automatically after a restart	Winlogon automatic restart sign-on (ARSO) isn't supported for use with Personal Data Encryption. To use Personal Data Encryption, ARSO must be disabled.

Personal Data Encryption for known folders settings

The following table lists the settings to configure Personal Data Encryption for known folders.

Expand table

Setting name	Description
Protect Desktop	Enable Personal Data Encryption on the Desktop folder.
Protect Documents	Enable Personal Data Encryption on the Documents folder.
Protect Pictures	Enable Personal Data Encryption on the Pictures folder.

Personal Data Encryption hardening recommendations

The following table lists the recommended settings to improve Personal Data Encryption's security.

Expand table

Setting name	Description
Kernel-mode crash dumps and live dumps	Kernel-mode crash dumps and live dumps can potentially cause the keys used by Personal Data Encryption to protect content to be exposed. For greatest security, disable kernel-mode crash dumps and live dumps.
Windows Error Reporting (WER)/user-mode crash dumps	Disabling Windows Error Reporting prevents user-mode crash dumps. User-mode crash dumps can potentially cause the keys used by Personal Data Encryption to protect content to be exposed. For greatest security, disable user-mode crash dumps.
Hibernation	Hibernation files can potentially cause the keys used by Personal Data Encryption to protect content to be exposed. For greatest security, disable hibernation.
Allow users to select when a password is required when resuming from connected standby	When this policy isn't configured on Microsoft Entra joined devices, users on a Connected Standby device can change the amount of time after the device ´s screen turns off before a password is required to wake the device. During the time when the screen turns off but a password isn't required, the keys used by Personal Data Encryption to protect content could potentially be exposed. It's recommended to explicitly disable this policy on Microsoft Entra joined devices.

Configure Personal Data Encryption with Microsoft Intune

If you use Microsoft Intune to manage your devices, you can configure Personal Data Encryption using a disk encryption policy, a settings catalog policy, or a custom profile.

Disk encryption policy

To configure devices using a disk encryption policy, go to **Endpoint security** > **Disk encryption** and select **Create policy**:

- **Platform** > **Windows**
- **Profile** > **Personal Data Encryption**

Provide a name, and select **Next**. In the **Configuration settings** page, select **Enable Personal Data Encryption** and configure the settings as needed.

Assign the policy to a group that contains as members the devices or users that you want to configure.

Settings catalog policy

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Personal Data Encryption	Enable Personal Data Encryption (User)	Enable Personal Data Encryption
Personal Data Encryption	Protect Desktop (User)	Enable protection for the Desktop folder
Personal Data Encryption	Protect Documents (User)	Enable protection for the Documents folder
Personal Data Encryption	Protect Pictures (User)	Enable protection for the Pictures folder
Administrative Templates > Windows Components > Windows Logon Options	Sign-in and lock last interactive user automatically after a restart	Disabled
Memory Dump	Allow Live Dump	Block
Memory Dump	Allow Crash Dump	Block
Administrative Templates > Windows Components > Windows Error Reporting	Disable Windows Error Reporting	Enabled
Power	Allow Hibernate	Block
Administrative Templates > System > Logon	Allow users to select when a password is required when resuming from connected standby	Disabled

Assign the policy to a group that contains as members the devices or users that you want to configure.

> **Tip**
>
> Use the following Graph call to automatically create the settings catalog policy in your tenant without assignments nor scope tags.
>
> When using this call, authenticate to your tenant in the Graph Explorer window. If it's the first time using Graph Explorer, you might need to authorize the application to access your tenant or to modify the existing permissions. This graph call requires *DeviceManagementConfiguration.ReadWrite.All* permissions.

msgraph

```
POST https://graph.microsoft.com/beta/deviceManagement/configurationPolicies
Content-Type: application/json

{ "id": "00-0000-0000-0000-000000000000", "name": "_MSLearn_PDE", "description":
"", "platforms": "windows10", "technologies": "mdm", "roleScopeTagIds": [ "0" ],
"settings": [ { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationSetting", "settingInstance": {
"@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId":
"device_vendor_msft_policy_config_admx_credentialproviders_allowdomaindelaylock"
, "choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_admx_credentialproviders_allowdomaindelaylock_
0", "children": [] } } }, { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationSetting", "settingInstance": {
"@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId":
"device_vendor_msft_policy_config_errorreporting_disablewindowserrorreporting",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_errorreporting_disablewindowserrorreporting_1"
, "children": [] } } }, { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationSetting", "settingInstance": {
"@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId":
"device_vendor_msft_policy_config_windowslogon_allowautomaticrestartsignon",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_windowslogon_allowautomaticrestartsignon_0",
"children": [] } } }, { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationSetting", "settingInstance": {
"@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId":
```

```
"device_vendor_msft_policy_config_memorydump_allowcrashdump",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_memorydump_allowcrashdump_0", "children": [] }
} }, { "@odata.type": "#microsoft.graph.deviceManagementConfigurationSetting",
"settingInstance": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId":
"device_vendor_msft_policy_config_memorydump_allowlivedump",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_memorydump_allowlivedump_0", "children": [] }
} }, { "@odata.type": "#microsoft.graph.deviceManagementConfigurationSetting",
"settingInstance": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId": "user_vendor_msft_pde_enablepersonaldataencryption",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"user_vendor_msft_pde_enablepersonaldataencryption_1", "children": [] } } }, {
"@odata.type": "#microsoft.graph.deviceManagementConfigurationSetting",
"settingInstance": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingInstance",
"settingDefinitionId": "device_vendor_msft_policy_config_power_allowhibernate",
"choiceSettingValue": { "@odata.type":
"#microsoft.graph.deviceManagementConfigurationChoiceSettingValue", "value":
"device_vendor_msft_policy_config_power_allowhibernate_0", "children": [] } } }
] }
```

Configure Personal Data Encryption with CSP

Alternatively, you can configure devices using the Policy CSP and Personal Data Encryption CSP.

Expand table

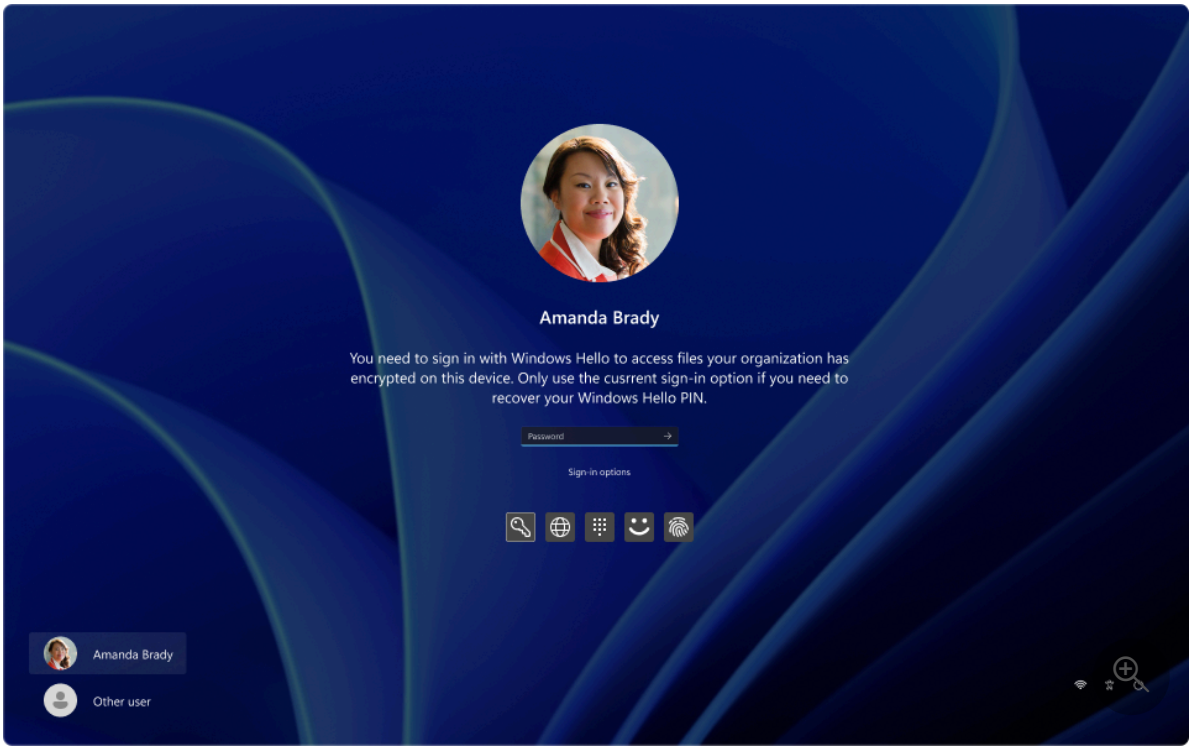
OMA-URI	Format	Value
`./User/Vendor/MSFT/PDE/EnablePersonalDataEncryption`	int	`1`
`./User/Vendor/MSFT/PDE/ProtectFolders/ProtectDesktop`	int	`1`
`./User/Vendor/MSFT/PDE/ProtectFolders/ProtectDocuments`	int	`1`
`./User/Vendor/MSFT/PDE/ProtectFolders/ProtectPictures`	int	`1`
`./Device/Vendor/MSFT/Policy/Config/WindowsLogon/AllowAutomaticRestartSignOn`	string	`<disabled/>`
`./Device/Vendor/MSFT/Policy/Config/MemoryDump/AllowCrashDump`	int	`0`
`./Device/Vendor/MSFT/Policy/Config/MemoryDump/AllowLiveDump`	int	`0`
`./Device/Vendor/MSFT/Policy/Config/ErrorReporting/DisableWindowsErrorReporting`	string	`<enabled/>`

OMA-URI	Format	Value
`./Device/Vendor/MSFT/Policy/Config/Power/AllowHibernate`	int	`0`
`./Device/Vendor/MSFT/Policy/Config/ADMX_CredentialProviders/AllowDomainDelayLock`	string	`<disabled/>`

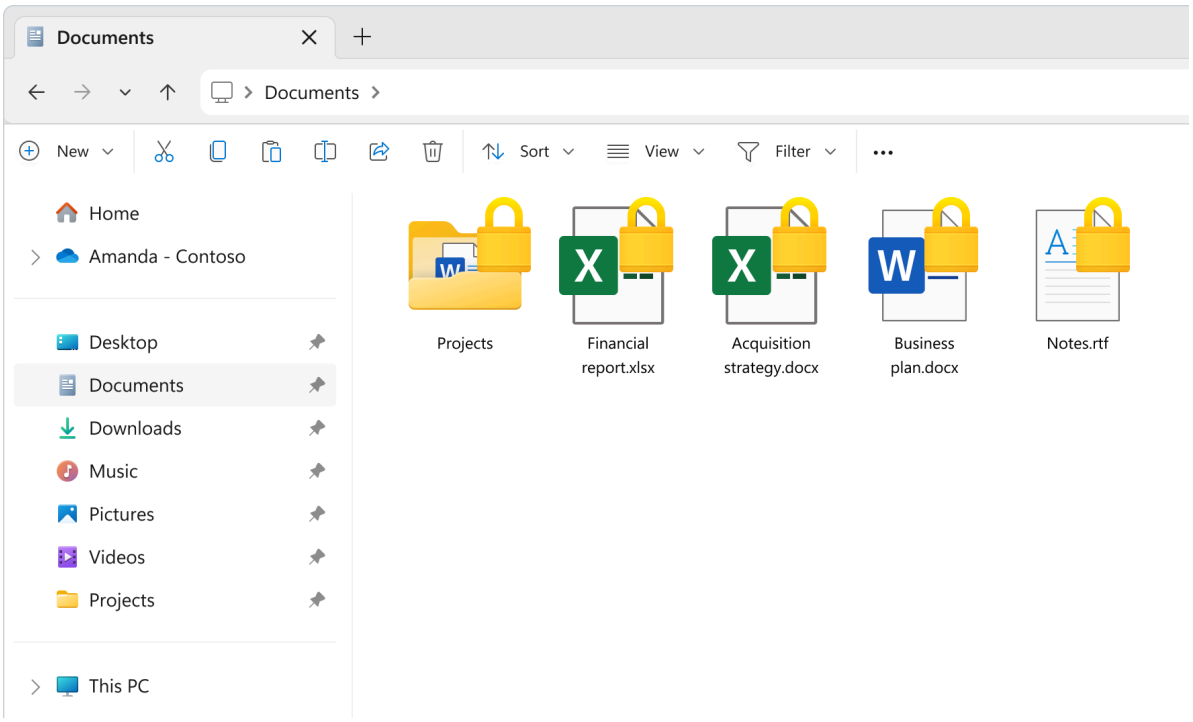
User experience

When Personal Data Encryption is enabled, the user experience is as follows:

- Access to Personal Data Encryption protected content is only possible when users sign in using Windows Hello (biometrics or PIN). If users sign in without Windows Hello, they can't open encrypted content
- If a user attempts to sign in without Windows Hello, a message appears on the sign in screen indicating that to access encrypted content the user must sign in with Windows Hello



- The data protected by Personal Data Encryption has a padlock on the file or folder's icon. The padlock icon is displayed in File Explorer and on the desktop



Disable Personal Data Encryption

Once Personal Data Encryption is enabled, it isn't recommended to disable it. However if you need to disable Personal Data Encryption, you can do so using the following steps.

Disable Personal Data Encryption with a disk encryption policy

To disable Personal Data Encryption devices using a disk encryption policy, go to **Endpoint security** > **Disk encryption** and select **Create policy**:

- **Platform** > **Windows**
- **Profile** > **Personal Data Encryption**

Provide a name, and select **Next**. In the **Configuration settings** page, select **Disable Personal Data Encryption**.

Assign the policy to a group that contains as members the devices or users that you want to configure.

Disable Personal Data Encryption with a settings catalog policy in Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Personal Data Encryption	**Enable Personal Data Encryption (User)**	Disable Personal Data Encryption

Assign the policy to a group that contains as members the devices or users that you want to configure.

Disable Personal Data Encryption with CSP

You can disable Personal Data Encryption with CSP using the following setting:

Expand table

OMA-URI	Format	Value
`./User/Vendor/MSFT/PDE/EnablePersonalDataEncryption`	int	`0`

Decrypt encrypted content

When you disable Personal Data Encryption, the content encrypted using Personal Data Encryption for known folders is automatically decrypted. However, the content encrypted using Personal Data Encryption APIs isn't decrypted automatically. To decrypt this content, follow these steps:

1. Open the properties of the file
2. Under the **General** tab, select **Advanced...**
3. Uncheck the option **Encrypt contents to secure data**
4. Select **OK**, and then **OK** again

Protected files can also be decrypted using cipher.exe, which can be helpful in the following scenarios:

- Decrypting a large number of files on a device
- Decrypting files on multiple of devices

To decrypt files on a device using `cipher.exe`:

- Decrypt all files under a directory including subdirectories:

  Windows Command Prompt

  ```
  cipher.exe /d /s:<path_to_directory>
  ```

- Decrypt a single file or all of the files in the specified directory, but not any subdirectories:

  Windows Command Prompt

  ```
  cipher.exe /d <path_to_file_or_directory>
  ```

> **Important**
>
> Once a user selects to manually decrypt a file, the user can't manually protect the file again using Personal Data Encryption.

Next steps

- Review the Personal Data Encryption FAQ

Feedback

Was this page helpful? Yes No

Provide product feedback

Frequently asked questions for Personal Data Encryption

Applies to: ✅ Windows 11

Here are some answers to common questions regarding Personal Data Encryption

General

Can Personal Data Encryption encrypt entire volumes or drives?

No, Personal Data Encryption only encrypts specified files and content.

How are files and content protected by Personal Data Encryption selected?

Personal Data Encryption APIs are used to select which files and content are protected using Personal Data Encryption.

Can users manually encrypt and decrypt files with Personal Data Encryption?

Currently users can decrypt files manually but they can't encrypt files manually. For information on how a user can manually decrypt a file, see the section Decrypt encrypted content.

Can Personal Data Encryption protected content be accessed after signing on via a Remote Desktop connection (RDP)?

No, it's not supported to access protected content over RDP.

Can Personal Data Encryption protected content be accessed via a network share?

No, Personal Data Encryption protected content can only be accessed after signing on locally to Windows with Windows Hello credentials.

What encryption method and strength does Personal Data Encryption use?

Personal Data Encryption uses AES-CBC with a 256-bit key to encrypt content.

Personal Data Encryption and other Windows features

What is the relation between Windows Hello for Business and Personal Data Encryption?

During user sign-on, Windows Hello unlocks the keys that Personal Data Encryption uses to protect content.

If a user signs into Windows with a password instead of Windows Hello, will they be able to access their Personal Data Encryption protected content?

No, the keys used by Personal Data Encryption to encrypt content are protected by Windows Hello credentials and can only be unlocked when signing on with Windows Hello (PIN or biometrics).

Can a file be protected with both Personal Data Encryption and EFS at the same time?

No, Personal Data Encryption and EFS are mutually exclusive.

Is Personal Data Encryption a replacement for BitLocker?

No, it's recommended to encrypt all volumes with BitLocker Drive Encryption for increased security.

Do I need to use OneDrive in Microsoft 365 as my backup provider?

No, Personal Data Encryption doesn't have a requirement for a backup provider, including OneDrive in Microsoft 365. However, backups are recommended in case the keys used by Personal Data Encryption to protect files are lost. OneDrive in Microsoft 365 is a recommended backup provider.

Are the files encrypted by Personal Data Encryption synced to OneDrive in an encrypted form?

Personal Data Encryption ensures that files are protected from unauthorized access by encrypting them at rest. When files are synced to OneDrive, they are transferred over a secure connection. However, Personal Data Encryption's encryption only applies to local data saved to the disk. Applications accessing the files, including OneDrive when it syncs data, get cleartext data. This means that while Personal Data Encryption protects files on the local disk, the files synced to OneDrive are not encrypted by Personal Data Encryption in the cloud.

Configure S/MIME for Windows

Article • 02/25/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Secure/Multipurpose Internet Mail Extensions (S/MIME) provides an added layer of security for email sent to and from an Exchange ActiveSync (EAS) account. S/MIME enables users to encrypt outgoing messages and attachments so that only intended recipients can read them. To read the messages, recipients must have a digital identification (ID), also known as a certificate.
Users can digitally sign a message, which provides the recipients with a way to verify the identity of the sender and that the message hasn't been tampered with.

Message encryption

Users can send encrypted message to recipients that have an encryption certificate. Users can only read encrypted messages if the message is received on their Exchange account, and they have corresponding decryption keys.

Encrypted messages can be read only by recipients who have a certificate. If you try to send an encrypted message to recipients whose encryption certificate isn't available, the app prompts you to remove these recipients before sending the email.

Digital signatures

A digitally signed message reassures the recipient that the message hasn't been tampered with, and verifies the identity of the sender. Recipients can only verify the digital signature if they're using an email client that supports S/MIME.

Windows edition and licensing requirements

The following table lists the Windows editions that support Email Encryption (S/MIME):

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Email Encryption (S/MIME) license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

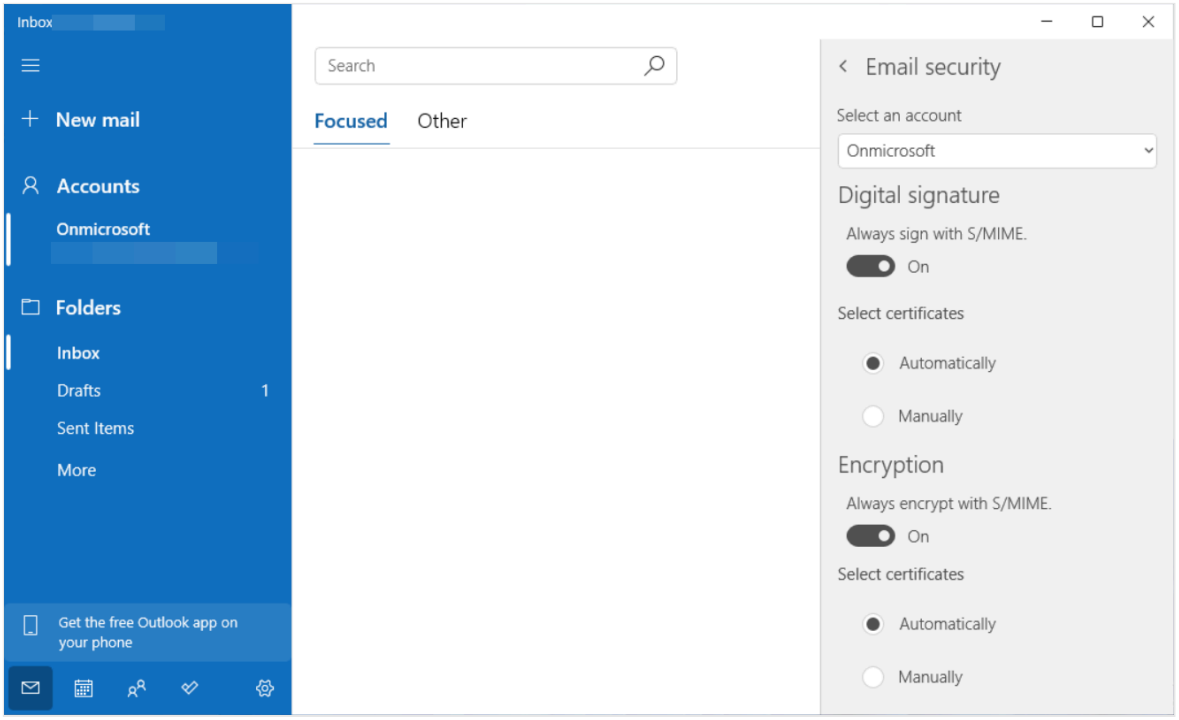
Prerequisites

- S/MIME is enabled for Exchange accounts (on-premises and Exchange Online). Users can't use S/MIME signing and encryption with a personal account such as Outlook.com
- Valid Personal Information Exchange (PFX) certificates are installed on the device
  - How to Create PFX Certificate Profiles in Configuration Manager
  - Use certificates for authentication in Microsoft Intune

Choose S/MIME settings

On the device, perform the following steps: (add select certificate)

1. Open the Mail app
2. Open **Settings** > **Email security**



3. In **Select an account**, select the account for which you want to configure S/MIME options

4. Make a certificate selection for digital signature and encryption

    - Select **Automatically** to let the app choose the certificate
    - Select **Manually** to specify the certificate yourself from the list of valid certificates on the device

5. (Optional) Select **Always sign with S/MIME**, **Always encrypt with S/MIME**, or both, to automatically digitally sign or encrypt all outgoing messages

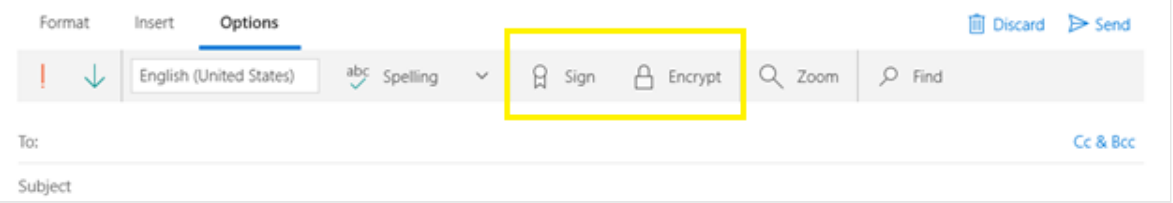
    > ⓘ **Note**
    >
    > The option to sign or encrypt can be changed for individual messages, unless EAS policies prevent it.

6. Select the back arrow

Encrypt or sign individual messages

1. While composing a message, select **Options** from the ribbon

2. Use **Sign** and **Encrypt** icons to turn on digital signature and encryption for this message



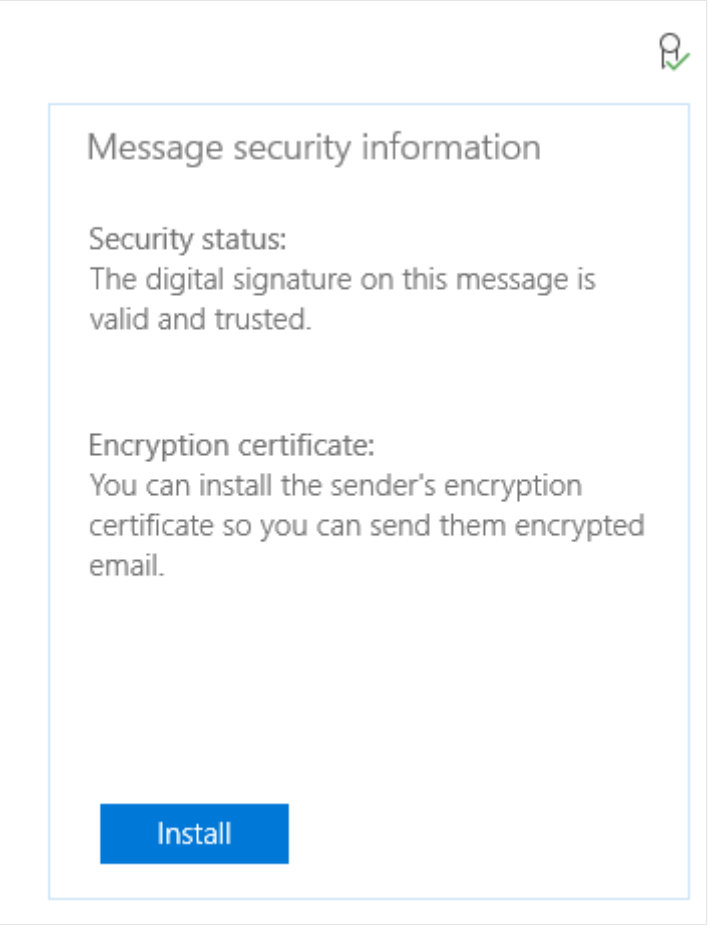
Read signed or encrypted messages

When you receive an encrypted message, the mail app checks whether there's a certificate available on your computer. If there's a certificate available, the message is decrypted when you open it. If your certificate is stored on a smartcard, you'll be prompted to insert the smartcard to read the message. Your smartcard may also require a PIN to access the certificate.

Install certificates from a received message

When you receive a signed email, the app provides a feature to install corresponding encryption certificate on your device if the certificate is available. This certificate can then be used to send encrypted email to this person.

1. Open a signed email
2. Select the digital signature icon in the reading pane
3. Select **Install.**



Feedback

Was this page helpful? Yes No

Provide product feedback

TLS/SSL overview (Schannel SSP)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article is to introduce IT professionals to TLS and SSL implementations in Windows using the Schannel Security Service Provider (SSP) by describing practical applications, changes in Microsoft's implementation, software requirements, and other resources for Windows Server and client devices.

Description

Schannel is a Security Support Provider (SSP) that implements the Secure Sockets Layer (SSL) and Transport Layer Security (TLS) Internet standard authentication protocols.

The Security Support Provider Interface (SSPI) is an API used by Windows systems to perform security-related functions including authentication. The SSPI functions as a common interface to several SSPs, including the Schannel SSP.

The Schannel authentication protocol suite provides several protocols that all use a client/server model. The following protocols are based on public key cryptography:

- TLS versions 1.0, 1.1, 1.2, and 1.3
- SSL versions 2.0 and 3.0
- Datagram Transport Layer Security (DTLS) versions 1.0 and 1.2
- Private Communications Transport (PCT)

> **Note**
>
> TLS 1.0 and TLS 1.1 is being deprecated in Windows starting with Windows 11 September Insider Preview build and later releases. To learn more, see TLS 1.0 and TLS 1.1 deprecation in Windows.

Applications

One problem when you administer a network is securing data that is being sent between applications across an untrusted network. You can use TLS and SSL to authenticate servers and client computers and then use the protocol to encrypt messages between the authenticated parties.

For example, you can use TLS/SSL for:

- SSL-secured transactions with an e-commerce website
- Authenticated client access to an SSL-secured website
- Remote access
- SQL access
- E-mail

Requirements

TLS and SSL protocols use a client/server model and are based on certificate authentication, which requires a public key infrastructure.

Server Manager information

There are no configuration steps necessary to implement TLS, SSL or Schannel.

See also

- The Schannel security package
- Secure Channel
- Transport Layer Security Protocol

Feedback

Was this page helpful? Yes No

Secure DNS Client over HTTPS (DoH)

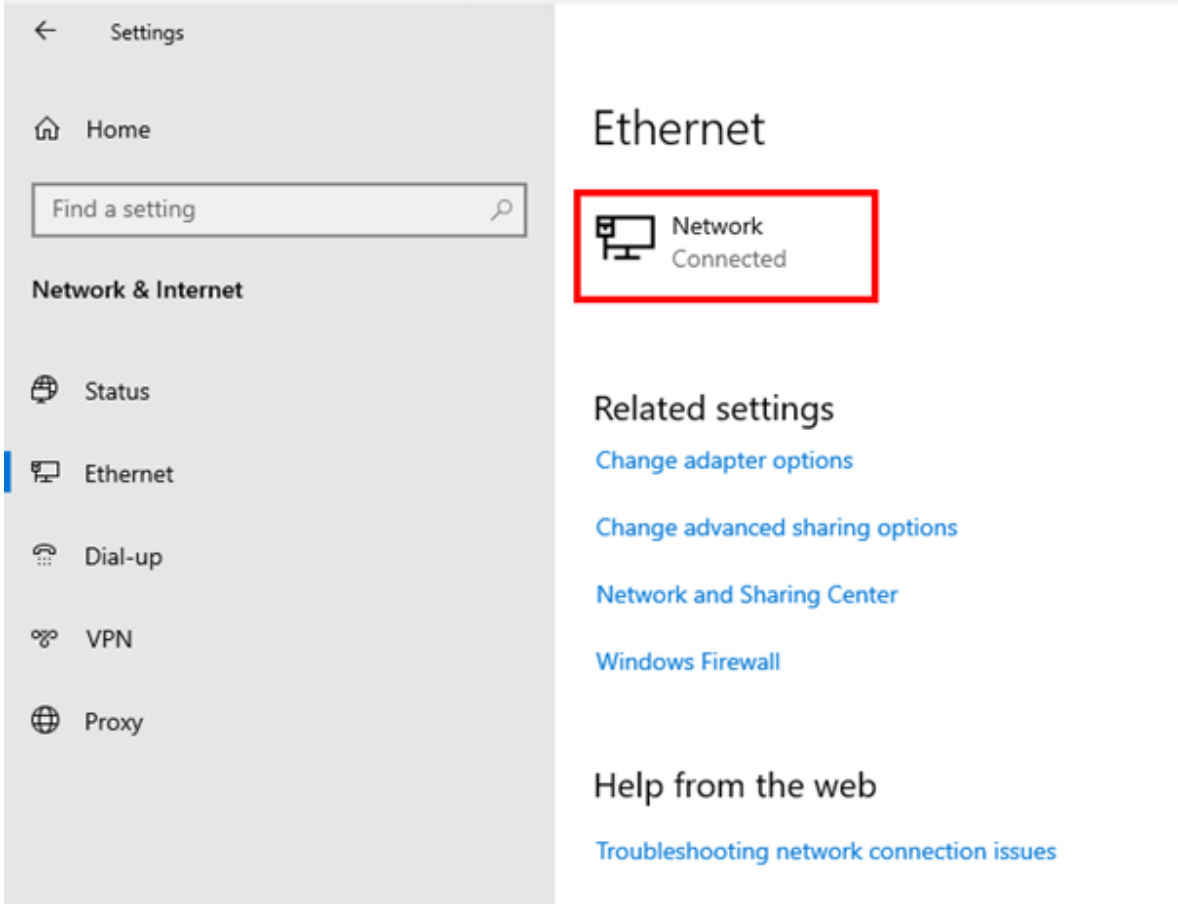
Article • 12/03/2023

Starting with Windows Server 2022, the DNS client supports DNS-over-HTTPS (DoH). When DoH is enabled, DNS queries between Windows Server's DNS client and the DNS server pass across a secure HTTPS connection rather than in plain text. By passing the DNS query across an encrypted connection, it's protected from interception by untrusted third parties.

Configure the DNS client to support DoH

You can only configure the Windows Server client to use DoH if the primary or secondary DNS server selected for the network interface is on the list of known DoH servers. You can configure the DNS client to require DoH, request DoH or only use traditional plain-text DNS queries. To configure the DNS client to support DoH on Windows Server with Desktop Experience, do the following steps:

1. From the Windows Settings control panel, select **Network & Internet**.
2. On the Network & Internet page, select **Ethernet**.
3. On the Ethernet screen, select the network interface that you want to configure for DoH.



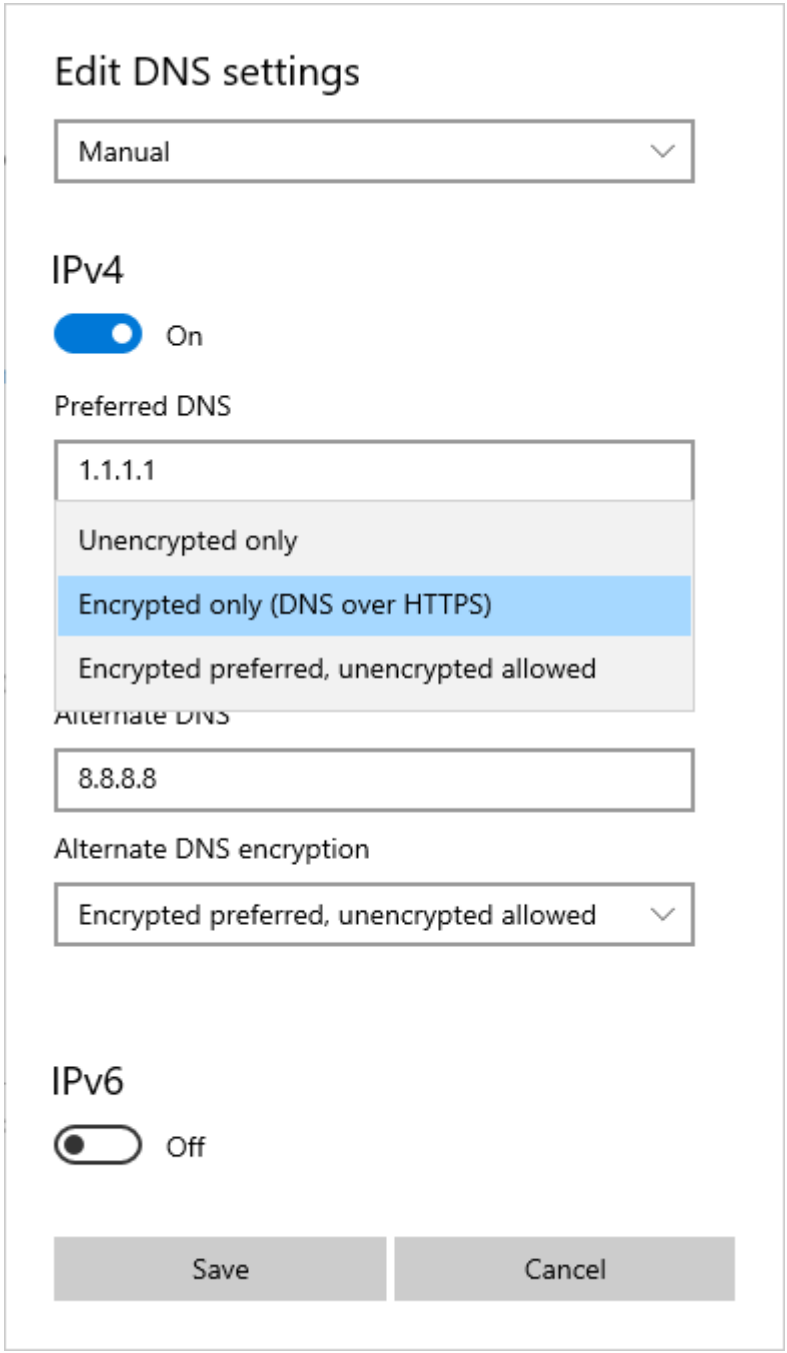
4. On the Network screen, scroll down to **DNS settings** and select the **Edit** button.

5. On the Edit DNS settings screen, select Manual from the automatic or manual IP settings dropdown. This setting allows you to configure the Preferred DNS and Alternate DNS servers. If the addresses of these servers are present in the list of known DoH servers, the **Preferred DNS encryption** dropdown will be enabled. You can choose between the following settings to set the preferred DNS encryption:

    - **Encrypted only (DNS over HTTPS)**. When this setting is chosen, all DNS query traffic will pass across HTTPS. This setting provides the best protection for DNS query traffic. However, it also means DNS resolution won't occur if the target DNS server is unable to support DoH queries.

    - **Encrypted preferred, unencrypted allowed**. When this setting is chosen, the DNS client will attempt to use DoH and then fall back to unencrypted DNS queries if that isn't possible. This setting provides the best compatibility for DoH capable DNS servers, but you won't be provided with any notification if DNS queries are switched from DoH to plain text.

    - **Unencrypted only**. All DNS query traffic to the specified DNS server is unencrypted. This setting configures the DNS client to use traditional plain text DNS queries.



6. Select **Save** to apply the DoH settings to the DNS client.

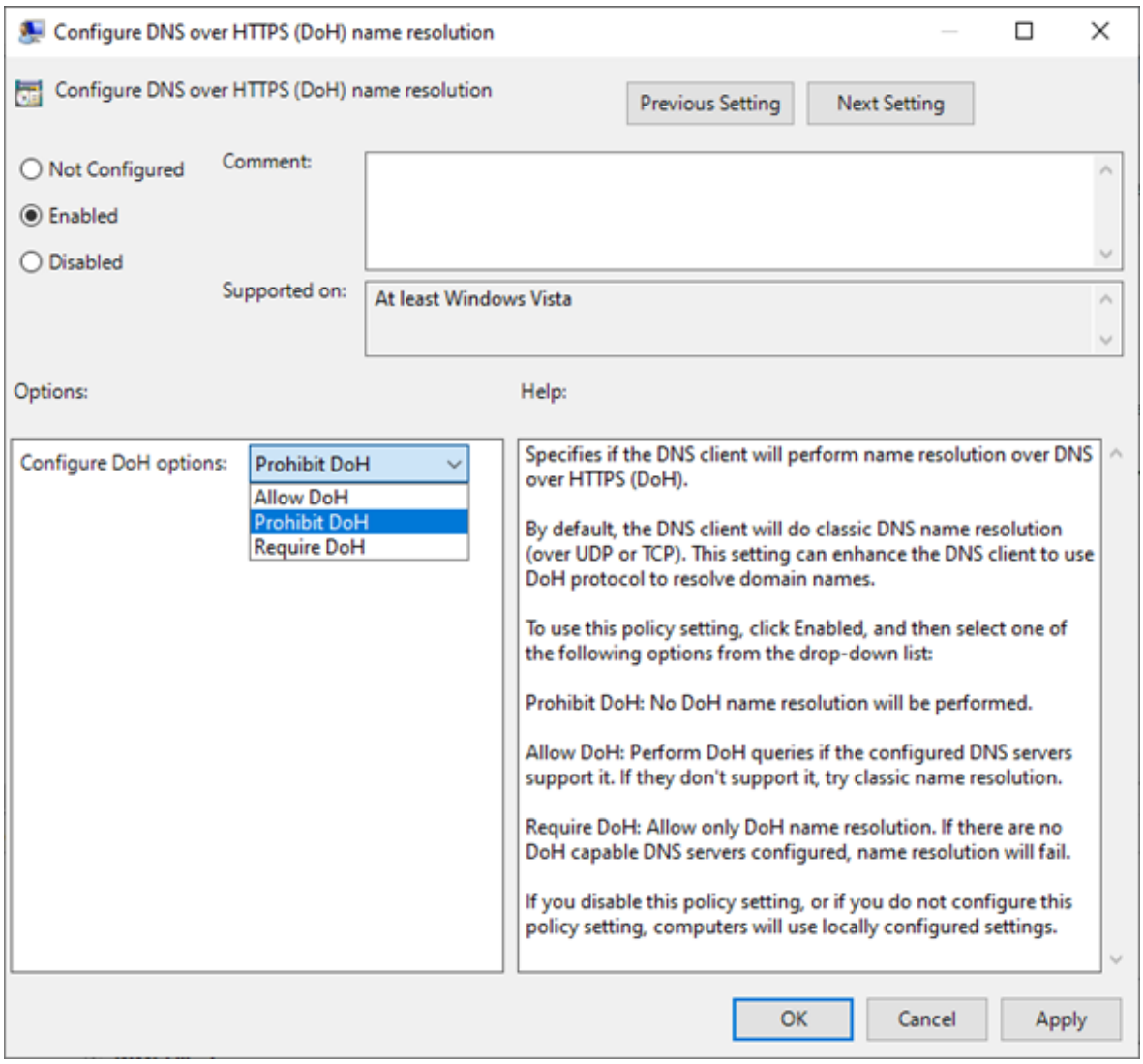
If you're configuring the DNS server address for a client using PowerShell using the `Set-DNSClientServerAddress` cmdlet, the DoH setting will depend on whether the server's fallback setting is in the list of known DoH servers table. At present you can't configure DoH settings for the DNS client on Windows Server 2022 using Windows Admin Center or *sconfig.cmd*.

Configuring DoH through Group Policy

Windows Server 2022 local and domain Group Policy settings include the **Configure DNS over HTTPS (DoH) name resolution** policy. You can use it to configure the DNS client to use DoH. This policy is found in the `Computer`

`Configuration\Policies\Administrative Templates\Network\DNS Client` node. When enabled, this policy can be configured with the following settings:

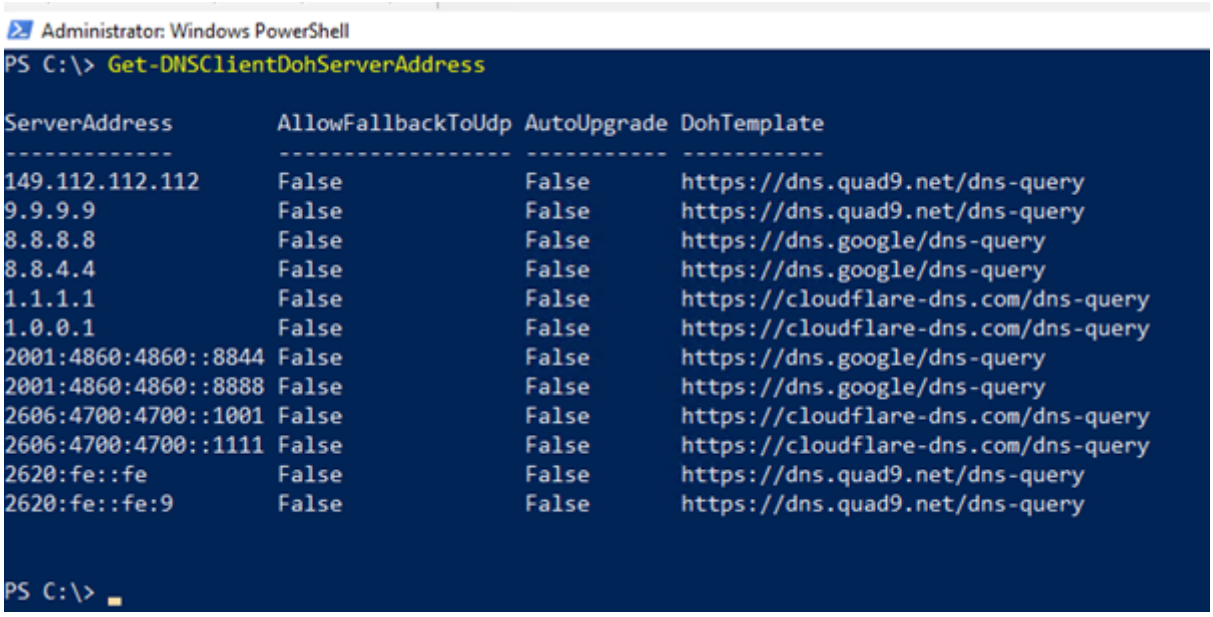
- **Allow DoH**. Queries will be performed using DoH if the specified DNS servers support the protocol. If the servers don't support DoH, non-encrypted queries will be issued.
- **Prohibit DoH**. Will prevent use of DoH with DNS client queries.
- **Require DoH**. Will require that queries are performed using DoH. If configured DNS servers don't support DoH, name resolution will fail.



Don't enable the Require DoH option for domain joined computers as Active Directory Domain Services is heavily reliant on DNS because the Windows Server DNS Server service doesn't support DoH queries. If you require DNS query traffic on Active Directory Domain Services network to be encrypted, consider implementing IPsec based connection security rules to protect this traffic. See Securing end-to-end IPsec connections by using IKEv2 for more information.

Determine which DoH servers are on the known server list

Windows Server ships with a list of servers that are known to support DoH. You can determine which DNS servers are on this list by using the `Get-DNSClientDohServerAddress` PowerShell cmdlet.



The default list of known DoH servers is as follows:

Expand table

Server Owner	DNS Server IP Addresses
Cloudflare	1.1.1.1 1.0.0.1 2606:4700:4700::1111 2606:4700:4700::1001
Google	8.8.8.8 8.8.4.4 2001:4860:4860::8888 2001:4860:4860::8844
Quad 9	9.9.9.9 149.112.112.112 2620:fe::fe 2620:fe::fe:9

Add a new DoH server to the list of known servers

You can add new DoH servers to the list of known servers using the `Add-DnsClientDohServerAddress` PowerShell cmdlet. Specify the URL of the DoH template and whether you'll allow the client to fall back to an unencrypted query should the secure query fail. The syntax of this command is:

PowerShell

```
Add-DnsClientDohServerAddress -ServerAddress '<resolver-IP-address>' -
DohTemplate '<resolver-DoH-template>' -AllowFallbackToUdp $False -
AutoUpgrade $True
```

Use Name Resolution Policy Table with DoH

You can use the Name Resolution Policy Table (NRPT) to configure queries to a specific DNS namespace to use a specific DNS server. If the DNS server is known to support DoH, queries related to that domain will be performed using DoH rather than in an unencrypted manner.

Zero Trust DNS

Applies to: ✅ Windows 11

Overview

Zero Trust DNS (ZTDNS) is a security feature that enables enterprise IT administrators to natively enforce domain-name-based network access controls on their Windows endpoints. It addresses the critical need to ensure that enterprise Windows devices only communicate with trusted network destinations, reducing the risk of a range of network attacks from malware communication to data exfiltration.

ZTDNS is an enhancement of the Windows DNS client that blocks all outbound IP traffic from the Windows device by default and only allows IP traffic to destinations resolved by the trusted DNS server or explicitly approved by the enterprise IT administrator. When paired with a policy-aware Protective DNS (PDNS) server, ZTDNS acts as the policy-enforcement point on the Windows device. This approach reduces the need for deep packet inspection or reliance on insecure signals like plain-text DNS or Server Name Indication (SNI) when attempting to determine the domain name associated with outbound traffic. Aligning with the Zero Trust principles, ZTDNS follows the "deny by default and allow by exception for a limited time" approach.

How Zero Trust DNS works

ZTDNS operates by integrating the Windows DNS client with the Windows Filtering Platform (WFP) to enable domain-name-based network lockdown. When you configure ZTDNS on a Windows device to use PDNS servers that support DNS over HTTPS (DoH) or DNS over TLS (DoT), the system ensures:

- **Encrypted DNS enforcement**: The Windows DNS client forces the use of encrypted DNS and sends queries only to the configured PDNS servers
- **Approved traffic only**: Outbound traffic is permitted only to IP addresses resolved by these trusted PDNS servers or to IP ranges with manual exceptions configured by the enterprise IT administrator
- **Default denial**: All other IPv4 and IPv6 outbound traffic is blocked by default, adhering to the "deny by default" principle of Zero Trust
- **Connection logging**: The device maintains a comprehensive log of attempted outbound connections for monitoring and troubleshooting

Traffic flow process when ZTDNS is configured on a Windows device

1. **Initial lockdown**: Windows blocks all outbound IPv4 and IPv6 traffic except for connections to the configured Protective DNS servers, explicitly allowed IP ranges, and essential network discovery traffic (DHCP, DHCPv6, and NDP)
2. **DNS resolution**: When applications need to connect to a destination, they query the trusted PDNS servers through encrypted channels (DoH or DoT)
3. **Dynamic allow listing**: DNS responses from PDNS servers that contain IP address resolutions trigger outbound allow exceptions for those specific IP addresses for a specified time
4. **Traffic enforcement**: Applications can then connect to the resolved IP addresses, while connections to any other IP addresses are blocked unless they're on the manual exceptions list

Security benefits

ZTDNS provides significant security advantages by addressing various network-based threats:

DNS hijacking protection

By ensuring that only DNS resolutions from trusted PDNS servers are used, ZTDNS helps prevent bad actors from redirecting traffic to malicious sites through DNS hijacking attacks.

Malicious communications prevention

Only allowing outbound connections to IP addresses resolved through trusted DNS queries helps disrupt phishing attempts and prevents non-administrative malware stagers and beacons from communicating with command and control servers.

Data exfiltration mitigation

Restricting outbound traffic to approved domains reduces the risk of sensitive data being transmitted to unauthorized destinations without requiring analysis of domain name resolution patterns.

Supports end-to-end encryption

Unlike traditional network filtering that relies on plain-text signals or deep packet inspection, ZTDNS is effective even when DNS traffic and SNI are encrypted, providing future-proof security controls.

Windows edition and licensing requirements

The following table lists the Windows editions that support Zero Trust DNS (ZTDNS):

Expand table

Windows 11 Home	Windows 11 Pro	Windows 11 Enterprise	Windows 11 Education
No	No	Yes	Yes

Zero Trust DNS (ZTDNS) license entitlements are granted by the following licenses:

Expand table

Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Next Steps

- Deploying Zero Trust DNS in your enterprise environment

Last updated on 11/18/2025

Deploying Zero Trust DNS

Applies to: ✅ Windows 11

This article provides comprehensive guidance for deploying Zero Trust DNS (ZTDNS) in Windows 11 Enterprise environments, including mandatory steps, configuration options, and deployment considerations.

Prerequisites

Before deploying ZTDNS, ensure your environment meets the following requirements:

System requirements

- **Windows 11 Build**: Device must run the latest build of Windows 11 Enterprise or Windows 11 Education.
- **Administrator privileges**: All ZTDNS configuration commands require administrator-level access.
- **Reachable encrypted DNS server**: Encrypted DNS server must be reachable from the device.

Protective DNS server requirements

For DNS servers to be compatible with ZTDNS, they must meet these requirements:

- **Encrypted DNS support**: Must support either DNS over HTTPS (DoH) or DNS over TLS (DoT) protocols.
- **Policy enforcement**: Should only resolve allowed domain names based on organizational policies.
- **Client authentication using mTLS (Optional)**: Should support mutual TLS (mTLS) for per-client resolution policies using client certificates.

Manual exceptions

Certain applications and services that don't use DNS to discover endpoints require manual allow exceptions for necessary IP address ranges to continue functioning after ZTDNS is enabled. Microsoft 365 applications like Exchange, SharePoint, Teams, and Office require manual exceptions for IP address ranges listed here.

Deployment modes

ZTDNS supports two operational modes:

- **Audit mode**: Logs all expected ZTDNS behavior without actually blocking outbound traffic. This mode helps administrators understand traffic patterns before full enforcement. **Strongly recommended** for initial testing and validation, it provides visibility into potential blocking scenarios without disrupting network connectivity.
- **Enforcement mode**: Actively blocks untrusted traffic while maintaining comprehensive logging. Only enable enforcement mode after thoroughly testing audit mode and confirming all required exceptions are configured.

Deploying ZTDNS using netsh

Follow these steps to deploy ZTDNS:

Step 1: Configure application DNS settings

Ensure all applications use the Windows DNS client instead of custom DNS implementations:

- **Microsoft Edge**: Configure the BuiltInDnsClientEnabled policy to disable custom DNS client
- **Google Chrome**: Configure the BuiltInDnsClientEnabled policy to disable custom DNS client
- **Other applications**: Review and configure similar settings for other browsers and applications using their custom DNS clients

Step 2: Add manual allow exceptions

Before enabling ZTDNS, configure exceptions for applications that require direct IP connections:

Microsoft 365 services

Add exceptions for Microsoft 365 services that require manual allow exceptions to maintain connectivity.

```
netsh ztdns add exception name=M365Services description="Microsoft 365 IP ranges" subnets=192.0.2.128/25,198.51.100.0/24,3fff::/48,3fff:123::/38
```

Teleconferencing applications

Teleconferencing applications like Teams use WebRTC, which negotiates IP addresses for peers within a TLS tunnel and has no DNS visibility. These IP subnets are also publicly documented and need manual allow exceptions for the application to work with ZTDNS.

```
netsh ztdns add exception name=TeamsWebRTC description="Teams WebRTC IP ranges" subnets=3fff::/48,3fff:123::/38
```

Custom application exceptions

Custom applications and services essential for employee productivity that don't discover endpoints through DNS or use hardcoded IP addresses require manual allow exceptions to function properly after ZTDNS is enabled.

```
netsh ztdns add exception name=AppName description="Description of AppName" subnets=192.0.2.128/25,198.51.100.0/24,3fff::/48,3fff:123::/38
```

Optional Step: Client authentication with mTLS

If your trusted DNS server supports client authentication by using mutual TLS (mTLS):

Configure trusted certificate authorities

Set trusted CA certificate hashes for validating the DNS server's certificate:

```
netsh ztdns set trustedca certhashes=00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff,1111223344556778899aabbccddeeff00112233445566778899aabbccddeeff
```

Configure client certificates

Set client certificate hashes and Extended Key Usage (EKU) OIDs:

```
netsh ztdns set clientcert certhashes=00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff,0011223344556778899aabbccddeeff00112233 ekus=1.3.6.1.4.1.311.64.256,1.3.6.1.4.1.311.64.255
```

Step 3: Configure trusted DNS server

Add one or more Protective DNS servers that support DoH or DoT:

DNS over HTTPS (DoH) server

```
netsh ztdns add server type=doh address=203.0.113.0
template=https://doh.resolver.example/dns-query priority=0
```

DNS over TLS (DoT) server

```
netsh ztdns add server type=dot address=2001:db8::1 hostname=dot.resolver.example
priority=1
```

Step 4: Enable ZTDNS in audit mode

Important: Always start with audit mode to validate configuration before enabling enforcement:

```
netsh ztdns set state enable=yes audit=yes
```

This command enables ZTDNS logging without blocking traffic, allowing you to:

- Identify applications that might need exceptions
- Validate DNS server configuration
- Monitor traffic patterns before enforcement

Step 5: Monitor and validate

After enabling audit mode:

1. Monitor ZTDNS logs for blocked traffic patterns
2. Add exceptions as needed
3. Verify all critical applications function correctly
4. Review and adjust name resolution policies on trusted DNS server

Step 6: Enable enforcement mode

After thorough testing in audit mode, enable enforcement:

```
netsh ztdns set state enable=yes audit=no
```

Step 7: Reboot (Recommended)

Although a reboot isn't necessary for enabling ZTDNS functionality, we recommend rebooting the device after enabling ZTDNS to ensure that applications don't continue using cached IP

addresses that ZTDNS isn't aware of or to terminate existing connections that weren't approved by policy.

Deploying ZTDNS using JSON

To streamline management of ZTDNS configuration, consider deploying ZTDNS by using a JSON file with all the necessary configurations.

Display current settings in JSON format

```
netsh ztdns show settings
```

Save configuration to JSON file (run through PowerShell)

```
netsh ztdns show settings | Out-File ztdns_config.json
```

Configure using JSON file

```
netsh ztdns set settings json=ztdns_config.json
```

Change state of ZTDNS service using JSON

```
netsh ztdns set state json=state_config.json
```

Sample JSON configuration

The following example shows a complete sample JSON configuration file:

JSON


```
{
  "config": {
    "flags": 511,
    "enableZtdns": "whenready",
    "auditMode": false,
    "blockLocalIps": false,
    "allowHostsFile": false,
    "enableForwarder": false,
    "maxRecordAge": 86400,
    "allowICSDHCPserver": false,
    "serviceNameRefreshTimeout": 1800
  },
  "rules": {
```

```
    "app_name": {
      "description": "Exception for app_name",
      "flags": 0,
      "subnets": ["192.0.2.1", "2001:db8::/64"]
    },
    "app_name2": {
      "description": "Exception for app_name2",
      "flags": 0,
      "subnets": ["192.0.2.1", "2001:db8::/64"]
    }
  },
  "clientCerts": {
    "flags": 0,
    "hashes": ["00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff",
"00112233445566778899aabbccddeeff00112233"],
    "ekus": ["1.3.6.1.4.1.311.64.256", "1.3.6.1.4.1.311.64.255"]
  },
  "trustedCas": {
    "flags": 0,
    "hashes": ["00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff",
"11112233445566778899aabbccddeeff00112233445566778899aabbccddeeff"]
  },
  "servers": {
    "1.1.1.1": {
      "flags": 0,
      "priority": 0,
      "dot": {
        "hostname": null,
        "port": 853
      }
    },
    "8.8.8.8": {
      "flags": 0,
      "priority": 0,
      "doh": {
        "template": "https://test.com/dns-query",
        "port": 443
      }
    }
  },
  "message": "The operation completed successfully.",
  "status": 0
}
```


JSON configuration parameters

- **enableZtdns**: Set to `yes`, `no`, or `whenready` (enables when prerequisites are met)
- **auditMode**: Boolean value for audit mode (true = audit, false = enforcement)
- **blockLocalIps**: Controls traffic to local machine addresses
- **allowHostsFile**: Allows traffic to entries in the Windows hosts file

- **enableForwarder**: Enables localhost to receive DNS queries made by application DNS clients and forward them to the trusted DNS server through ZTDNS
- **maxRecordAge**: Maximum time (in seconds) an IP address stays in the allowed list after DNS resolution
- **serviceNameRefreshTimeout**: Time between service name retrieval refreshes

Reboot (Recommended)

Although a reboot isn't necessary for enabling ZTDNS functionality, we recommend rebooting the device after enabling ZTDNS to ensure that applications don't continue using cached IP addresses that ZTDNS isn't aware of or to terminate existing connections that weren't approved by policy.

Disabling ZTDNS

If you experience unexpected connectivity issues while deploying ZTDNS:

```
netsh ztdns set state enable=no audit=no
```

This command immediately disables ZTDNS enforcement and restores normal network connectivity.

Last updated on 11/18/2025

Zero Trust DNS commands

Applies to: ✅ Windows 11

This article lists all the commands available for Zero Trust DNS (ZTDNS) configuration and management.

netsh ztdns add exception

Adds an IP address or range exception for ZTDNS.

Syntax:

```
syntax
netsh ztdns add exception [name=]<string> [[description=]<string>] [subnets=]
<subnets> [[json=]<filename>]
```

Description: Adds the given IP address or range to the ZTDNS allowlist. This action allows all outbound IP-based packets to be sent to this IP address or range when ZTDNS is enabled. When you use JSON, the "status" JSON key outputs the Win32 error code instead of returning it.

Example:

```
C:\> netsh ztdns add exception name=app_exception description="Exception for app_name"
subnets=192.0.2.1,2001:DB8::/64
```

Parameters:

- **name** - The name of the IP address or range exception group.
- **description** - Optional description for the IP address or range exception group.
- **subnets** - The IP addresses and/or subnets to exempt from ZTDNS enforcement.
- **json** - Path to a JSON file containing a set of exceptions to add. When present, this parameter must be the only parameter.

netsh ztdns add server

Adds a trusted DNS server for ZTDNS.

Syntax:

```
syntax
```

```
netsh ztdns add server [type=](doh|dot) [address=]<address> [[port=]<string>]
[[template=]<string>] [[hostname=]<string>] [[priority=]<priority>] [[json=]
<filename>]
```

Description: Adds a trusted DNS server for ZTDNS. This DNS server must be capable of receiving and responding to encrypted DNS queries from the given Windows 11 device by using DNS-over-HTTPS or DNS-over-TLS. Ideally, this DNS server is a Protective DNS server with allowlist and blocklist of domain names and is able to apply those policies when it receives an incoming DNS query from client devices. When you use JSON, the "status" JSON key outputs the Win32 error code instead of returning it.

Example: Example 1: Set DoH server as the trusted DNS server for ZTDNS

```
C:\> netsh ztdns add server type=doh address=203.0.113.0 port=123
template=https://doh.resolver.example/dns-query priority=0
```

Example 2: Set DoT server as the backup trusted DNS server for ZTDNS in case the primary isn't reachable or doesn't respond

```
C:\> netsh ztdns add server type=dot address=2001:db8::1 port=1234
hostname=dot.resolver.example priority=1
```

Parameters:

- **type** – Specify the encrypted DNS protocol to use for communicating with the trusted DNS server. Accepted values are "doh" or "dot".
- **address** - IP address of the trusted DNS server.
- **port** - Port to use for the trusted DNS server. Optional, uses default port of the specified encrypted DNS protocol if not defined.
- **template** – DoH template of the trusted DNS server, used only for DoH.
- **hostname** - Hostname of the trusted DNS server, used only for DoT.
- **priority** - Priority of the trusted DNS server. Higher priority servers are queried first.
- **json** - Path to a JSON file containing a set of servers to add. When present this must be the only parameter.

netsh ztdns delete clientcert

Deletes ZTDNS client certificate configuration

Syntax:

syntax

```
netsh ztdns delete clientcert
```

Description: Deletes the client certificate configured for ZTDNS client to authenticate with the trusted DNS server.

Example:

```
C:\> netsh ztdns delete clientcert
```

Parameters: N/A

netsh ztdns delete exception

Deletes an IP address or range exception for ZTDNS

Syntax:

```
syntax
netsh ztdns delete exception [[name=]<string>] [[json=]<filename>]
```

Description: Removes the IP address or subnet exception from the ZTDNS allowlist immediately. When you use JSON, the "status" JSON key outputs the Win32 error code instead of returning it.

Example:

```
C:\> netsh ztdns delete exception name=app_exception
```

Parameters:

- **name** - The name of the IP address or range exception group.
- **json** - Path to a JSON file containing a set of exceptions to delete, from a prior call to "show exception". The command deletes exceptions with matching names. When present, this parameter must be the only parameter.

netsh ztdns delete server

Deletes all trusted DNS servers for ZTDNS

Syntax:

```
syntax
netsh ztdns delete server
```

Description: Deletes all trusted DNS servers configured for ZTDNS. If you enable ZTDNS on the device, be cautious when running this command. The device can't send DNS queries to any DNS servers after this command until you disable ZTDNS.

Example:

```
C:\> netsh ztdns delete server
```

Parameters: N/A

netsh ztdns delete trustedca

Deletes all trusted certificate authorities for ZTDNS

Syntax:

```
syntax
netsh ztdns delete trustedca
```

Description: Removes all certificate hashes for trusted certificate authorities configured for ZTDNS.

Example:

```
C:\> netsh ztdns delete trustedca
```

Parameters: N/A

netsh ztdns set clientcert

Configures client certificate for ZTDNS

Syntax:

```
syntax
netsh ztdns set clientcert [certhashes=]<A list of comma-separated hashes> [ekus=]<A list of comma-separated EKU OIDs> [[json=]<filename>]
```

Description: Configures the client certificate for ZTDNS client to authenticate with the trusted DNS server.

Example:

`C:\> netsh ztdns set clientcert certhashes=00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff,0011223344556778899aabbccddeeff00112233 ekus=1.3.6.1.4.1.311.64.256,1.3.6.1.4.1.311.64.255`

Parameters:

- **certhashes** - A comma-separated list of certificate issuer hashes (SHA1 or SHA256) in lowercase hexadecimal format.
- **ekus** - A comma-separated list of EKU OIDs.
- **json** - Path to a JSON file containing a set of client certificates to add. When present, this parameter must be the only parameter.

netsh ztdns set settings

Configures ZTDNS through a JSON

Syntax:

```
syntax
netsh ztdns set settings [[json=]<filename>]
```

Description: Completes the configuration of ZTDNS, including adding a trusted DNS server, adding IP address or range exceptions, setting client certificates, setting trusted CAs, and enabling ZTDNS through a JSON file as long as the JSON file has all the necessary key value pairs.

Example:

`C:\> netsh ztdns set settings json=ztdns_config.json`

Parameters:

- **json** - Path to a JSON file containing all of the configuration information.

netsh ztdns set state

Configures the state of the ZTDNS service.

Syntax:

```
syntax
```

```
netsh ztdns set state [[enable=](yes|no|whenready)] [[audit=](yes|no)] [[localips=]
(allow|block)] [[hostsfile=](allow|block)] [[forwarder=](yes|no)] [[maxrecordage=]
<integer>] [[svcnamerefreshtimeout=]<integer>] [[json=]<filename>]
```

Description: Configures the state of the ZTDNS service on the device.

Example:

```
C:\> netsh ztdns set state enable=yes audit=no
```

Parameters:

- **enable** - Enables outbound traffic filtering through ZTDNS. If you set this parameter to `whenready`, ZTDNS is enabled once all prerequisites are met, such as the successful installation of configured certificates.
- **audit** – Enables audit mode in ZTDNS, which logs filtered outbound traffic without blocking it. The default value is `no`.
- **localips** – Allows traffic to local machine addresses. The default value is `block`.
- **hostsfile** - Honors entries configured in the hosts file and allows outbound traffic to those destinations. The default value is `block`.
- **forwarder** - **NOTE:** This is a test parameter and functionality isn't ready for use yet. Don't use it. When ready, it enables localhost to receive DNS queries made by application DNS clients and forward them to the trusted DNS server through ZTDNS. The default value is `no`.
- **maxrecordage** - Maximum time in seconds that an IP address stays in the allowed list after a successful resolution from a trusted DNS server. The default value is `86400`.
- **svcnamerefreshtimeout** - Time in seconds between each refresh of the service name retrieval.
- **json** - Path to a JSON file containing the configuration. When present, this parameter must be the only parameter.

netsh ztdns set trustedca

Configures trusted certificate authority for ZTDNS

Syntax:

syntax

```
netsh ztdns set trustedca [certhashes=]<A list of comma-separated hex strings>
[[json=]<filename>]
```

Description: Configures trusted certificate authorities for ZTDNS through their certificate hashes.

Example:

```
C:\> netsh ztdns set trustedca certhashes=00112233445566778899aabbccddeeff00112233445566778899aabbccddeeff,1111223344556778899aabbccddeeff00112233445566778899aabbccddeeff
```

Parameters:

- **certhashes** - A comma-separated list of certificate issuer hashes (SHA1 or SHA256) in lowercase hexadecimal format.
- **json** - Path to a JSON file containing a list of certificate hashes for trusted CAs. When present this parameter must be the only parameter.

netsh ztdns show clientcert

Displays client certificate configuration for ZTDNS

Syntax:

```
syntax
netsh ztdns show clientcert
```

Description: Displays client certificate used by the ZTDNS client to authenticate with the trusted DNS server.

Example:

```
C:\> netsh ztdns show clientcert
```

Parameters: N/A

netsh ztdns show exception

Displays the IP address or range exception configured for ZTDNS.

Syntax:

```
syntax
netsh ztdns show exception [[name=]<string>]
```

Description: Displays the IP address or range exceptions added to the ZTDNS allowlist. ZTDNS allows outbound IP-based packets to this IP address or range, even if the trusted DNS server doesn't resolve them.

Example: Example 1: Displays the configuration for a given IP address or range exception

```
C:\> netsh ztdns show exception name=app_exception
```

Example 2: Displays the configuration for all IP address or range exceptions

```
C:\> netsh ztdns show exception
```

Parameters:

- **name** - The name of the IP address or range exception group.

netsh ztdns show server

Displays all trusted DNS servers configured for ZTDNS.

Syntax:

```
syntax
netsh ztdns show server [[address=]<address>]
```

Description: Displays all trusted DNS servers configured for ZTDNS along with their configuration information, such as the encrypted DNS protocol being used, port number, DoH template or hostname, and priority.

Example: Example 1: Displays configuration for a given trusted DNS server

```
C:\> netsh ztdns show server address=192.0.2.1
```

Example 2: Displays configuration for all trusted DNS servers

```
C:\> netsh ztdns show server
```

Parameters:

- **address** – Optional, IP address of a specific trusted DNS server to see its configuration.

netsh ztdns show settings

Displays all ZTDNS configuration in JSON format.

Syntax:

```
syntax
netsh ztdns show settings
```

Description: Displays all configuration for ZTDNS, including ZTDNS service state, client certificates, trusted CAs, IP address or range exceptions, and trusted DNS servers in JSON format.

Example:

```
C:\> netsh ztdns show settings
```

Parameters: N/A

netsh ztdns show state

Displays the state of the ZTDNS service.

Syntax:

```
syntax
netsh ztdns show state
```

Description: Displays the state of the ZTDNS service on the device.

Example:

```
C:\> netsh ztdns show state
```

Parameters: N/A

netsh ztdns show trustedca

Displays the trusted certificate authority configured for ZTDNS.

Syntax:

```
syntax
netsh ztdns show trustedca
```

Description: Displays SHA1 and SHA256 certificate hashes for trusted certificate authorities configured for ZTDNS.

Example:

```
C:\> netsh ztdns show trustedca
```

Parameters: N/A

netsh ztdns help

Provides help documentation for ZTDNS commands.

Syntax:

syntax

```
netsh ztdns <command> help
netsh ztdns <command> ?
```

Description: To get information on syntax, parameters, or examples for a specific netsh ZTDNS command, type the command in Command Prompt followed by `help` or `?`.

Example: Example 1: Help on netsh ztdns set state command

```
C:\> netsh ztdns set state help
```

Example 2: Help on various netsh ztdns show commands

```
C:\> netsh ztdns show ?
```

Parameters: N/A

Last updated on 11/18/2025

Considerations for running Zero Trust DNS

Applies to: ✅ Windows 11

This article outlines the main technologies and solutions that may be impacted when Zero Trust DNS (ZTDNS) is deployed in an enterprise environment, along with recommended mitigations and workarounds.

Overview

Zero Trust DNS is a powerful Windows feature that enables administrators to lock down outbound IP-based traffic to endpoints resolved by a Protective DNS server. However, ZTDNS breaks certain networking features and protocols by design, as it blocks all outbound traffic that can't be associated with a domain name resolved by trusted DNS servers.

Technologies impacted by ZTDNS deployment

Printing services

Impact: ZTDNS blocks traditional network printer discovery that uses multicast DNS (mDNS).

Mitigation:

- Use Microsoft Universal Print for secure printer management.
- Universal Print resolves endpoints through ZTDNS instead of being discovered by using mDNS. For more information, see Universal Print endpoints.

File sharing

Impact:

- ZTDNS might block discovering file shares over local network.

Mitigation:

- Configure Protective DNS servers to also serve as local network DNS servers for internal file shares.
- Ensure file sharing solutions use DNS-discoverable endpoints.
- Migrate to cloud-based file hosting solutions like SharePoint, OneDrive for Business, or Azure Files.

Peer-to-peer sharing and Windows Update

Impact:

- Blocks local network peer-to-peer sharing between devices.
- Increases traffic to Windows Update services since it blocks peer-to-peer sharing of downloaded updates between devices on the same network.

Mitigation:

- Deploy Microsoft Connected Cache (MCC) for enterprises or Windows Server Update Services (WSUS) to reduce traffic to Windows Update services.

Media streaming and casting

Impact:

- Blocks local network media discovery for streaming and casting.

Mitigation:

- Migrate to cloud-based media services where possible.
- Consider physical display connections instead of wireless casting.

Teleconferencing and video communication

Impact:

- Blocks WebRTC traffic. Teleconferencing applications like Teams use WebRTC, which discovers IP addresses through STUN and TURN, bypassing DNS.

Mitigation:

- Configure IP address exceptions for vendor-documented IP ranges.

  ```
  netsh ztdns add exception name=AppWebRTC description="App WebRTC ranges" subnets=<vendor-provided-ranges>
  ```

- Add exceptions for IP ranges documented for Microsoft 365 services including Teams.

Applications with hard-coded IP addresses

Impact:

- The policy blocks applications that use hard-coded IP addresses or non-DNS discovery mechanisms.

Mitigation:

- Use audit mode first to identify affected applications.
- Create targeted IP exceptions only when necessary.
- Work with application vendors to support DNS-based discovery.
- Consider application modernization or replacement where feasible.

Network access through captive portals

Impact:

- Users can't access captive portal gated networks.

Reasoning:

- ZTDNS currently doesn't support this scenario.
- Captive portals rely on plain-text DNS query interception to redirect clients to a captive portal page. When ZTDNS is enabled, no plain-text DNS traffic is emitted.

Browser DNS clients

Impact:

- Browsers with built-in encrypted DNS clients, such as Edge and Chrome, might not resolve names or send traffic.

Mitigation:

- Set browser management policies to turn off built-in DNS clients:
  - Microsoft Edge BuiltInDnsClientEnabled policy
  - Chrome BuiltInDnsClientEnabled policy
- Make sure browsers and other custom applications use the Windows system DNS resolver.

Technologies that can bypass ZTDNS

VPN and SASE/SSE tunnels

Behavior:

- VPN and SASE/SSE solutions work when the gateway is resolved by a trusted DNS server or when the IP is manually permitted.
- All tunneled traffic appears associated with the gateway's domain name or exception, not the actual destination.

Considerations:

- Maintain domain name auditing at tunnel exit nodes.
- Implement similar lockdown functionality at the tunnel endpoint.
- Ensure tunnel gateways are properly configured in ZTDNS exceptions.

Virtualization and stack bypass technologies

Behavior:

- Virtual machines with their own networking stacks bypass ZTDNS since their traffic isn't routed through host TCP/IP stack.
- Any technology bypassing Windows Filtering Platform like Express Data Path (XDP) and Data Plane Development Kit (DPDK) also bypass ZTDNS enforcement.

Considerations:

- Plan separate network security controls for virtualized environments.
- Inventory applications using stack bypass technologies and implement alternative security controls for these applications.

Conclusion

While ZTDNS provides powerful security capabilities aligned with Zero Trust principles, successful deployment requires careful planning and consideration of the technologies in your environment. Use audit mode extensively, plan appropriate mitigations, and consider modernizing applications that rely on insecure networking protocols to maximize the security benefits of ZTDNS deployment.

Last updated on 11/18/2025

Zero Trust DNS troubleshooting guidance

Applies to: Supported versions of Windows Client

This article shows you how to diagnose common connectivity and configuration issues that affect Zero Trust Domain Name System (ZTDNS) services, and how to use ZTDNS logs. The troubleshooting steps in this article help you maintain network security while making sure that legitimate applications and services continue to work correctly.

After you deploy Zero Trust Domain Name System (ZTDNS) services, use this guidance if you experience any of the following symptoms:

- Application connectivity failures
- DNS resolution errors or timeouts
- Unexpected network blocks or service disruptions
- Network access and ZTDNS configuration issues

Best practices to avoid ZTDNS issues

- **Before you deploy and enforce ZTDNS organization-wide**
  - Test the configuration in audit mode.
  - Deploy and test ZTDNS on pilot devices.

- **Maintaining the ZTDNS deployment**
  - When you add an IP exception, document what exception was made and why.
  - To understand traffic patterns and identify issues early, monitor the Event Viewer logs.
  - To optimize ZTDNS deployment, conduct periodical reviews of the cofiguration and logs.

Troubleshooting checklist

If you experience connectivity issues after you deploy and enforce ZTDNS, follow these steps.

Step 1: Verify that the trusted DNS servers are configured correctly

Make sure that at least one trusted DNS server is configured for ZTDNS. Open a Windows Command Prompt window, and then run the following command:

Console

```
netsh ztdns show server
```

This command displays all the trusted DNS servers that are configured for Zero Trust DNS, including their protocol (DoH or DoT), IP address, and priority settings.

Step 2: Test the connectivity to the trusted DNS servers

Make sure that client computers can connect to your trusted DNS server. Open a Windows PowerShell window, and then run the following commands:

PowerShell

```
# Test basic connectivity to the DNS server
ping <dns-server-ip>
```

PowerShell

```
# Test DNS resolution using the trusted server
Resolve-DnsName -Name <domain-name> -Server <dns-server-ip>
```

If these commands don't succeed, see DNS troubleshooting guidance.

After these commands run successfully, check the end-to-end connectivity by pinging the resolved IP address.

Step 3: Test domain name resolution

To use the Windows DNS client to test DNS resolution, go to a client computer, and run the following command at a Windows command prompt:

Console

```
ping <allowed-domain-name>
```

This command uses the Windows DNS client and trusted DNS server for name resolution, testing both DNS server connectivity and endpoint reachability. Resolve any DNS issues that you find, and then check whether these changes resolve your primary issue. If the issue remains, continue to the more advanced troubleshooting steps.

Step 4 (Advanced): Review the ZTDNS configuration

To verify your current configuration, run the following commands at a Windows command prompt:

Console

```
# Check ZTDNS service state
netsh ztdns show state
```

Console

```
# View all ZTDNS settings in JSON format
netsh ztdns show settings
```

Console

```
# List configured IP exceptions
netsh ztdns show exception
```

Console

```
# Display client certificate configuration
netsh ztdns show clientcert
```

Console

```
# Show trusted certificate authorities
netsh ztdns show trustedca
```

Resolve any issues that you find.

Step 5 (Advanced): Test connectivity in audit mode

If you suspect configuration issues, temporarily enable audit mode, and then test connectivity when ZTDNS isn't enforced. In audit mode, Windows logs instances in which ZTDNS is expected to block connections if it's fully enforced. To enable audit mode for ZTDNS, run the following command:

Console

```
netsh ztdns set state enable=yes audit=yes
```

To identify services that need exceptions, review the BlockedConnections log. For more information about how to use this log, see How to find ZTDNS logs later in this article.

Step 6 (Advanced): Disable ZTDNS

If you still can't resolve your primary issue, disable ZTDNS. To disable ZTDNS enforcement and restore normal network connectivity, run the following command:

Console

```
netsh ztdns set state enable=no audit=no
```

Common issues and solutions

Applications can't connect

After you enforce ZTDNS, users experience the following symptoms:

- Applications can't reach external services.
- Network connections time out or destinations refuse connections.
- Normal web browsing works but specific applications fail.

To troubleshoot these issues, follow these steps:

1. Check the BlockedConnections log for recent blocked attempts. For more information about this log, see How to find ZTDNS logs.
2. Identify the blocked IP addresses that the application and destination use.
3. Check whether the application requires IP exceptions (for example, WebRTC applications require exceptions).
4. To add an exception (if it's necessary), run `netsh ztdns add exception`.

DNS resolution failures

After you enforce ZTDNS, users experience the following symptoms:

- Applications can't resolve domain names.
- DNS lookup operations time out.
- Applications report DNS errors.

To troubleshoot these issues, follow these steps:

1. To review the trusted DNS server configuration, run `netsh ztdns show server`.
2. To test connectivity to the trusted DNS server, run `ping <dns-server-ip>`.
3. Make sure that the trusted DNS server responds to encrypted queries.

How to find ZTDNS logs

On Windows devices that're configured to use ZTDNS, you can use Event Viewer to monitor all attempted connections and configuration changes. To locate the ZTDNS logs in Event Viewer, follow these steps:

1. In the Search bar, enter **Event Viewer**, and then select it from the search results.
2. In the left panel of Event Viewer, navigate to **Applications and Service Logs** > **Microsoft** > **Windows** > **ZTDNS**.

The ZTDNS folder lists three log categories. The following table describes the three categories, and how you can use the log information.

Expand table

Category	Purpose	Information included	Usage
BlockedConnections	Contains entries that document the connections that ZTDNS blocks.	• Time of the blocked connection • Source IP address and port • Destination IP address and port • Name of the initiating process	Review this log to identify applications or services that might need exceptions.
Operational	Contains entries that document changes in the ZTDNS configuration and service state.	• ZTDNS service status changes • Configuration changes • System events that relate to ZTDNS operations	Review this log to monitor the health and configuration of ZTDNS.
PermittedConnections	Contains entries that document any connections that ZTDNS allows.	• Time of the allowed connection • Source IP address and port	Review this log to verify that ZTDNS allows the expected connections. **Note:** By default, this category is disabled. To use it, go to the left

Category	Purpose	Information included	Usage
		• Destination IP address and port • Name of the initiating process	panel, right-click **PermittedConnections**, and then select **Enable Log**.

Last updated on 02/12/2026

Extensible Authentication Protocol (EAP) for network access

07/09/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Extensible Authentication Protocol (EAP) is an authentication framework that allows for the use of different authentication methods for secure network access technologies. Examples of these technologies include wireless access using IEEE 802.1X, wired access using IEEE 802.1X, and Point-to-Point Protocol (PPP) connections like Virtual Private Networking (VPN). EAP isn't a specific authentication method like MS-CHAP v2, but rather a framework that enables networking vendors to develop and install new authentication methods, known as EAP methods on the access client and authentication server. The EAP framework is originally defined by RFC 3748 and extended by various other RFCs and standards.

Authentication methods

EAP authentication methods that are used within tunneled EAP methods are commonly known as **inner methods** or **EAP types**. Methods that are set up as **inner methods** have the same configuration settings as they would when used as an outer method. This article contains configuration information specific to the following authentication methods in EAP.

EAP-Transport Layer Security (EAP-TLS): Standards-based EAP method that uses TLS with certificates for mutual authentication. Appears as **Smart Card or other Certificate (EAP-TLS)** in Windows. EAP-TLS can be deployed as an **inner method** for another EAP method or as a standalone EAP method.

> **Tip**
>
> EAP methods that use EAP-TLS, being certificate-based, generally offer the highest level of security. For example, EAP-TLS is the only allowed EAP method for **WPA3-Enterprise 192-bit mode**.

EAP-Microsoft Challenge Handshake Authentication Protocol version 2 (EAP-MSCHAP v2): Microsoft-defined EAP method that encapsulates the MSCHAP v2 authentication protocol, which uses username and password, for authentication. Appears as **Secure password (EAP-MSCHAP v2)** in Windows. EAP-MSCHAPv2 can be used as a standalone method for VPN, but only as an **inner method** for wired/wireless connections.

> ⚠ **Warning**
>
> MSCHAPv2-based connections are subject to similar attacks as for NTLMv1. Windows 11 Enterprise, version 22H2 (build 22621) enables **Windows Defender Credential Guard**, which might cause issues with MSCHAPv2-based connections.

Protected EAP (PEAP): Microsoft-defined EAP method that encapsulates EAP within a TLS tunnel. The TLS tunnel secures the inner EAP method, which could be unprotected otherwise. Windows supports EAP-TLS and EAP-MSCHAP v2 as inner methods.

EAP-Tunneled Transport Layer Security (EAP-TTLS): This is described by RFC 5281, encapsulates a TLS session that performs mutual authentication using another inner authentication mechanism. This inner method can be either an EAP protocol, such as EAP-MSCHAP v2, or a non-EAP protocol, such as Password Authentication Protocol (PAP). In Windows Server 2012, the inclusion of EAP-TTLS only provides support on the client-side (in Windows 8). NPS doesn't support EAP-TTLS at this time. The client support enables interoperation with commonly deployed RADIUS servers that support EAP-TTLS.

EAP-Subscriber Identity Module (EAP-SIM), EAP-Authentication and Key Agreement (EAP-AKA), and EAP-AKA Prime (EAP-AKA'): This is described by various RFCs, enables authentication by using SIM cards, and is implemented when a customer purchases a wireless broadband service plan from a mobile network operator. As part of the plan, the customer commonly receives a wireless profile that is preconfigured for SIM authentication.

Tunnel EAP (TEAP): This is described by RFC 7170, tunneled EAP method that establishes a secure TLS tunnel and executes other EAP methods inside that tunnel. Supports EAP chaining - authenticating the machine and user within one authentication session. In Windows Server 2022, the inclusion of TEAP only provides support for the client-side - Windows 10, version 2004 (build 19041). NPS doesn't support TEAP at this time. The client support enables interoperation with commonly deployed RADIUS servers that support TEAP. Windows supports EAP-TLS and EAP-MSCHAP v2 as inner methods.

The following table lists some common EAP methods and their IANA assigned method Type numbers.

Expand table

EAP method	IANA assigned Type number	Native Windows support
MD5-Challenge (EAP-MD5)	4	✗
One-Time Password (EAP-OTP)	5	✗

EAP method	IANA assigned Type number	Native Windows support
Generic Token Card (EAP-GTC)	6	❌
EAP-TLS	13	✅
EAP-SIM	18	✅
EAP-TTLS	21	✅
EAP-AKA	23	✅
PEAP	25	✅
EAP-MSCHAP v2	26	✅
Protected One-Time Password (EAP-POTP)	32	❌
EAP-FAST	43	❌
Pre-Shared Key (EAP-PSK)	47	❌
EAP-IKEv2	49	❌
EAP-AKA'	50	✅
EAP-EKE	53	❌
TEAP	55	✅
EAP-NOOB	56	❌

Configuring EAP properties

You can access the EAP properties for 802.1X authenticated wired and wireless access in the following ways:

- Configuring the Wired Network (IEEE 802.3) Policies and Wireless Network (IEEE 802.11) Policies extensions in Group Policy.
  - **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings**
- Using Mobile Device Management (MDM) software, such as Intune (Wi-Fi/Wired)
  - Wi-Fi CSP
  - WiredNetwork CSP
- Manually configuring wired or wireless connections on client computers.

You can access the EAP properties for virtual private network (VPN) connections in the following ways:

- Using Mobile Device Management (MDM) software, such as Intune

    - VPNv2 CSP
    - VPN profile options
- Manually configuring VPN connections on client computers.
- Using Connection Manager Administration Kit (CMAK) to configure VPN connections.

For more information on configuring EAP properties, see Configure EAP profiles and settings in Windows.

XML profiles for EAP

The profiles used for different connections types are XML files that contain the configuration options for that connection. Each different connection type follows a specific schema:

- Wi-Fi (WLAN) profiles
- Wired network (Ethernet) profiles
- VPN profiles

However, when configured to use EAP, each profile schema has a child element EapHostConfig element.

- Wired/Wireless: `EapHostConfig` is a child element of the EAPConfig element. **MSM** > **security** (Wired/Wireless) > OneX > **EAPConfig**
- VPN: `EapHostConfig` is a child element of **NativeProfile** > **Authentication** > **Eap** > **Configuration**

This configuration syntax is defined in the Group Policy: Wireless/Wired Protocol Extension specification.

> **Note**
>
> The various configuration GUIs don't always show every technically possible option. For example, Windows Server 2019 and earlier are unable to configure TEAP in UI. However, it's often possible to import an existing XML profile that has previously been configured.
>
> The remainder of the article is intended to provide a mapping between the EAP specific portions of the Group Policy/Control Panel UI and the XML configuration options, and providing a description of the setting.

More information about configuring XML profiles can be found in XML Profiles. An example of using an XML profile containing EAP settings can be found in Provision a Wi-Fi profile via a website.

Security settings

The following table explains the configurable security settings for a profile that uses 802.1X. These settings map to OneX.

Expand table

Setting	XML element	Description
Select a network authentication method:	EAPConfig	Allows you to select the EAP method to use for authentication. See Authentication method configuration settings and Cellular authentication configuration settings
Properties		Opens the properties dialog for the selected EAP method.
Authentication Mode	authMode	Specifies the type of credentials used for authentication. The following values are supported: 1. User or computer authentication 2. Computer authentication 3. User authentication 4. Guest authentication "Computer", in this context, means "Machine" in other references. `machineOrUser` is the default in Windows.
Max Authentication Failures	maxAuthFailures	Specifies the maximum number of authentication failures allowed for a set of credentials, defaulting to `1`.
Cache user information for subsequent connections to this network	cacheUserData	Specifies whether the user's credentials should be cached for subsequent connections to the same network, defaulting to `true`.

Advanced security settings > IEEE 802.1X

If **Enforce advanced 802.1X settings** is checked, all of the following settings are configured. If it's unchecked, default settings apply. In XML, all elements are optional, with the default values used if they aren't present.

Expand table

Setting	XML element	Description
Max Eapol-Start Msgs	maxStart	Specifies the maximum number of EAPOL-Start messages that can be sent to the authenticator (RADIUS server) before the supplicant (Windows client) assumes there's no authenticator present, defaulting to `3`.
Start Period (seconds)	startPeriod	Specifies the time period (in seconds) to wait before an EAPOL-Start message is sent to start the 802.1X authentication process, defaulting to `5`.
Held Period (seconds)	heldPeriod	Specifies the time period (in seconds) to wait after a failed authentication attempt to reattempt authentication, defaulting to `1`.
Auth Period (seconds)	authPeriod	Specifies the time period (in seconds) to wait for a response from the authenticator (RADIUS server) before assuming there's no authenticator present, defaulting to `18`.
Eapol-Start Message	supplicantMode	Specifies the method of transmission used for EAPOL-Start messages. The following values are supported: 1. Don't transmit (`inhibitTransmission`) 2. Transmit (`includeLearning`) 3. Transmit per IEEE 802.1X (`compliant`) "Computer", in this context, means "Machine" in other references. `compliant` is the default in Windows, and is the only valid option for wireless profiles.

Advanced security settings > Single Sign On

The following table explains the settings for Single Sign On (SSO), formerly known as Pre-Logon Access Provider (PLAP).

Expand table

Setting	XML element	Description
Enable Single Sign On for this network	singleSignOn	Specifies whether SSO is enabled for this network, defaulting to `false`. Don't use `singleSignOn` in a profile if the network doesn't require it.
Perform immediately before User Perform immediately after User	type	Specifies when SSO should be performed - either before or after the user logs on.

Setting	XML element	Description
Max delay for connectivity(seconds)	maxDelay	Specifies the maximum delay (in seconds) before the SSO attempt fails, defaulting to `10`.
Allow additional dialogs to be displayed during Single Sign On	allowAdditionalDialogs	Specified whether to allow EAP dialogs to be displayed during SSO, defaulting to `false`.
This network uses different VLAN for authentication with machine and user credentials	userBasedVirtualLan	Specifies whether the virtual LAN (VLAN) used by the device changes based on the user's credentials, defaulting to `false`.

Authentication method configuration settings

> **⊗ Caution**
>
> If a Network Access Server is configured to allow the same type of authentication method for a tunneled EAP method (such as PEAP) and a non-tunneled EAP method (such as EAP-MSCHAP v2), there's a potential security vulnerability. When you deploy both a tunneled EAP method and EAP (which isn't protected), don't use the same authentication type. For example, if you deploy PEAP-TLS, don't also deploy EAP-TLS because if you require the protection of the tunnel, it serves no purpose to permit the method to be executed outside of the tunnel as well.

The following table explains the configurable settings for each authentication method.

EAP-TLS

The EAP-TLS settings in the UI map to EapTlsConnectionPropertiesV1, which is extended by EapTlsConnectionPropertiesV2 and EapTlsConnectionPropertiesV3.

Expand table

Setting	XML element	Description
Use my smart card	CredentialsSource > SmartCard	Specifies that clients making authentication requests must present a smart card certificate for network authentication.
Use a certificate on this computer	CredentialsSource > CertificateStore	Specifies that authenticating clients must use a certificate located in the Current User or Local

Setting	XML element	Description
		Computer certificate stores.
Use simple certificate selection (Recommended)	SimpleCertSelection	Specifies whether Windows will automatically select a certificate for authentication without user interaction (if possible) or if Windows shows a dropdown for the user to select a certificate.
Advanced		Opens the Configure Certificate Selection dialog box.
Server validation options		
Use a different user name for the connection	DifferentUsername	Specifies whether to use a user name for authentication that is different from the user name in the certificate.

The following lists the configuration settings for **Configure Certificate Selection**. These settings define the criteria a client uses to select the appropriate certificate for authentication. This UI maps to TLSExtensions > FilteringInfo.

Expand table

Setting	XML element	Description
Certificate Issuer	**CAHashList** `Enabled="true"`	Specifies whether Certificate Issuer filtering is enabled. If both **Certificate Issuer** and **Extended Key Usage (EKU)** are enabled, only those certificates that satisfy both conditions are considered valid for authenticating the client to the server.
Root Certification Authorities	**IssuerHash**	Lists the names of all of the issuers for which corresponding certification authority (CA) certificates are present in the **Trusted Root Certification Authorities** or **Intermediate Certification Authorities** certificate store of local computer account. This includes: • All the root certification authorities and intermediate certification authorities. • Contains only those issuers for which there are corresponding valid certificates that are present on the computer (for example, certificates that aren't expired or not revoked). • The final list of certificates that are allowed for authentication contains only those certificates that were issued by any of the issuers selected in this list.

Setting	XML element	Description
		In XML, this is the SHA-1 thumbprint (hash) of the certificate.
Extended Key Usage (EKU)		Allows you to select **All Purpose**, **Client Authentication**, **AnyPurpose**, or any combination of these. Specifies that when a combination is selected, all the certificates satisfying at least one of the three conditions are considered valid certificates for authenticating the client to the server. If EKU filtering is enabled, one of the choices must be selected, otherwise the **Extended Key Usage (EKU)** checkbox becomes unchecked.
All Purpose	**AllPurposeEnabled**	When selected, this item specifies that certificates having the **All Purpose** EKU are considered valid certificates for authenticating the client to the server. The Object Identifier (OID) for **All Purpose** is `0` or empty.
Client Authentication	**ClientAuthEKUList** `Enabled="true"` (> **EKUMapInList** > **EKUName**)	Specifies that certificates having the **Client Authentication** EKU, and the specified list of EKUs are considered valid certificates for authenticating the client to the server. The Object Identifier (OID) for **Client Authentication** is `1.3.6.1.5.5.7.3.2`.
AnyPurpose	**AnyPurposeEKUList** `Enabled="true"` (> **EKUMapInList** > **EKUName**)	Specifies that all certificates having **AnyPurpose** EKU and the specified list of EKUs are considered valid certificates for authenticating the client to the server. The Object Identifier (OID) for **AnyPurpose** is `1.3.6.1.4.1.311.10.12.1`.
Add	**EKUMapping** > **EKUMap** > **EKUName/EKUOID**	Opens the **Select EKUs** dialog box, which enables you to add standard, custom, or vendor-specific EKUs to the **Client Authentication** or **AnyPurpose** list. Selecting **Add** or **Edit** within the **Select EKUs** dialog box opens the **Add/Edit EKU** dialog box, which provides two options: 1. **Enter the name of the EKU** - Provides a place to type the name of the custom EKU. 2. **Enter the EKU OID** - Provides a place to type the OID for the EKU. Only numeric digits, separators, and `.` are allowed. Wild cards are permitted, in which case all of the child OIDs in the hierarchy are allowed.

Setting	XML element	Description
		For example, entering `1.3.6.1.4.1.311.*` allows for `1.3.6.1.4.1.311.42` and `1.3.6.1.4.1.311.42.2.1`.
Edit		Enables you to edit custom EKUs that you've added. The default, predefined EKUs can't be edited.
Remove		Removes the selected EKU from the **Client Authentication** or **AnyPurpose** list.

Server certificate validation

Many EAP methods include an option for the client to validate the server's certificate. If the server certificate isn't validated, the client can't be sure that it's communicating with the correct server. This exposes the client to security risks, including the possibility that the client might unknowingly connect to a rogue network.

> **Note**
>
> Windows requires the server certificate have the **Server Authentication** EKU. The object identifier (OID) for this EKU is `1.3.6.1.5.5.7.3.1`.

The following table lists the server validation options applicable to each EAP method. Windows 11 updated the server validation logic to be more consistent. To learn more, see Updated server certificate validation behavior in Windows 11. Should they conflict, the descriptions in the following table describe the behavior for Windows 10 and earlier.

Expand table

Setting	XML element	Description
Verify the server's identity by validating the certificate	EAP-TLS: PerformServerValidation PEAP: PerformServerValidation	This item specifies that the client verifies that server certificates presented to the client computer have: • The correct signatures • The signatures aren't expired • Were issued by a trusted root certification authority (CA) **Disabling this check box causes client computers to be unable to verify the identity of your servers during the authentication process. If server authentication does not occur, users are exposed to severe security risks, including**

Setting	XML element	Description
		the possibility that users might unknowingly connect to a rogue network.
Connect to these servers	EAP-TLS: ServerValidation > ServerNames PEAP: ServerValidation > ServerNames EAP-TTLS: **ServerValidation > ServerNames** TEAP: **ServerValidation > ServerNames**	Allows you to specify the name for Remote Authentication Dial-In User Service (RADIUS) servers that provide network authentication and authorization. You must type the name **exactly** as it appears in the **subject** field of each RADIUS server certificate or use regular expressions (regex) to specify the server name. The complete syntax of the regular expression can be used to specify the server name, but to differentiate a regular expression with the literal string, you must use at least one `*` in the string specified. For example, you can specify `nps.*\.example\.com` to specify the RADIUS server `nps1.example.com` or `nps2.example.com`. You can also include a `;` to separate multiple servers. **If no RADIUS servers are specified, the client only verifies that the RADIUS server certificate was issued by a trusted root CA.**
Trusted Root Certification Authorities	EAP-TLS: ServerValidation > TrustedRootCA PEAP: ServerValidation > TrustedRootCA EAP-TTLS: **ServerValidation > TrustedRootCAHashes** TEAP: **ServerValidation > TrustedRootCAHashes**	Lists the trusted root certification authorities. The list is built from the trusted root CAs that are installed in the computer and in the user certificate stores. You can specify which trusted root CA certificates supplicants use to determine whether they trust your servers, such as your server running Network Policy Server (NPS) or your provisioning server. If no trusted root CAs are selected, the 802.1X client verifies that the computer certificate of the RADIUS server is issued by an installed trusted root CA. If one or multiple trusted root CAs are selected, the 802.1X client verifies that the computer certificate of the RADIUS server is issued by a selected trusted root CA. **If no trusted root CAs are selected, the client verifies that the RADIUS server certificate was issued by any trusted root CA.** If you have a public key infrastructure (PKI) on your network, and you use your CA to issue certificates to your RADIUS servers, your CA certificate is automatically added to the list of trusted root CAs. You can also purchase a CA certificate from a non-Microsoft vendor. Some non-Microsoft trusted root CAs provide software with your purchased certificate that automatically installs the purchased certificate into the **Trusted Root Certification Authorities** certificate store. In

Setting	XML element	Description
		this case, the trusted root CA automatically appears in the list of trusted root CAs. Don't specify a trusted root CA certificate that isn't already listed in client computers' **Trusted Root Certification Authorities** certificate stores for **Current User** and **Local Computer**. If you designate a certificate that isn't installed on client computers, authentication fails. In XML, this is the SHA-1 thumbprint (hash) of the certificate (or SHA-256 for TEAP).

Server validation user prompt

The following table describes the server validation user prompt options available for each EAP method. When a server certificate is untrusted, these options determine whether:

- The connection fails immediately.
- The user is prompted to manually accept or reject the connection.

EAP-TLS

Expand table

Setting	XML element
Don't prompt user to authorize new servers or trusted certification authorities	ServerValidation > DisableUserPromptForServerValidation

Prevents the user from being prompted to trust a server certificate if that certificate is incorrectly configured, isn't already trusted, or both (if enabled). To simplify the user experience and prevent users from mistakenly trusting a server deployed by an attacker, it's recommended that you select this checkbox.

Cellular authentication configuration settings

The following lists the configuration settings for EAP-SIM, EPA-AKA, and EPA-AKA' respectively.

EAP-SIM

EAP-SIM is defined in RFC 4186. EAP Subscriber Identity Module (SIM) is used for authentication and session key distribution using the 2nd generation mobile network Global System for Mobile Communications (GSM) Subscriber Identity Module (SIM).

The EAP-SIM settings in the UI map to EapSimConnectionPropertiesV1.

Expand table

Item	XML element	Description
Use strong cipher keys	**UseStrongCipherKeys**	Specifies that if selected, the profile uses strong encryption.
Don't reveal real identity to server when pseudonym identity is available	**DontRevealPermanentID**	When enabled, forces the client to fail the authentication if server requests for permanent identity though the client have a pseudonym identity with it. Pseudonym identities are used for identity privacy so that the actual or permanent identity of a user isn't revealed during authentication.
	ProviderName	Only available in XML, a string that indicates the provider name that is allowed for authentication.
Enable usage of realms	**Realm**=`true`	Provides a place to type the realm name. If this field is left blank with **Enable usage of realms** selected, the realm is derived from the International Mobile Subscriber Identity (IMSI) using the realm 3gpp.org, as described in the 3rd Generation Partnership Project (3GPP) standard 23.003 V6.8.0.
Specify a realm	**Realm**	Provides a place to type a realm name. If **Enable usage of realms** is enabled, this string is used. If this field is empty, the derived realm is used.

WPA3-Enterprise 192-bit mode

WPA3-Enterprise 192-bit mode is a special mode for WPA3-Enterprise that enforces certain high security requirements on the wireless connection to provide a minimum of 192 bits of security. These requirements align with the Commercial National Security Algorithm (CNSA) Suite, CNSSP 15, which is a set of cryptographic algorithms that is approved to protect classified and top secret information by the United States National Security Agency (NSA). 192-bit mode can sometimes be referred to as "Suite B mode," which is a reference to the NSA Suite B Cryptography specification, which was replaced by CNSA in 2016.

Both WPA3-Enterprise and WPA3-Enterprise 192-bit mode are available starting in Windows 10, version 2004 (build 19041) and Windows Server 2022. However, WPA3-Enterprise was singled out as a separate authentication algorithm in Windows 11. In XML, this is specified in the authEncryption element.

The following table lists the algorithms required by the CNSA Suite.

Expand table

Algorithm	Description	Parameters
Advanced Encryption Standard (AES)	Symmetric block cipher used for encryption	256-bit key (AES-256)
Elliptic Curve Diffie-Hellman (ECDH) Key Exchange	Asymmetric algorithm used to establish a shared secret (key)	384-bit prime modulus curve (P-384)
Elliptic Curve Digital Signature Algorithm (ECDSA)	Asymmetric algorithm used for digital signatures	384-bit prime modulus curve (P-384)
Secure Hash Algorithm (SHA)	Cryptographic hash function	SHA-384
Diffie-Hellman (DH) Key Exchange	Asymmetric algorithm used to establish a shared secret (key)	3072-bit modulus
Rivest-Shamir-Adleman (RSA)	Asymmetric algorithm used for digital signatures or key-establishment	3072-bit modulus

To comply with CNSA requirements, WPA3-Enterprise 192-bit mode mandates the use of EAP-TLS with these restricted cipher suites:

- `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384`
  - ECDHE and ECDSA using the 384-bit prime modulus curve P-384
- `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384` / `TLS_DHE_RSA_AES_256_GCM_SHA384`
  - ECDHE using the 384-bit prime modulus curve P-384
  - RSA >= 3072-bit modulus

> **Note**
>
> P-384 is also known as `secp384r1` or `nistp384`. Other elliptic curves, such as P-521 aren't permitted.

SHA-384 is in the SHA-2 family of hash functions. Other algorithms and variants, such as SHA-512 or SHA3-384, aren't permitted.

Windows supports only the `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384` and `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384` cipher suites for WPA3-Enterprise 192-bit mode. The `TLS_DHE_RSA_AES_256_GCM_SHA384` cipher suite isn't supported.

TLS 1.3 uses new simplified TLS suites, of which only `TLS_AES_256_GCM_SHA384` is compatible with WPA3-Enterprise 192-bit mode. As TLS 1.3 requires (EC)DHE and allows ECDSA or RSA certificates, along with the AES-256 AEAD and SHA384 hash, `TLS_AES_256_GCM_SHA384` is equivalent to `TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384` and `TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384`. However, RFC 8446 requires that TLS 1.3-compliant applications support P-256, which is forbidden by CNSA. Therefore, WPA3-Enterprise 192-bit mode can't be fully compliant with TLS 1.3. However, there are no known interoperability issues with TLS 1.3 and WPA3-Enterprise 192-bit mode.

To configure a network for WPA3-Enterprise 192-bit mode, Windows requires EAP-TLS be used with a certificate that meets the requirements described previously.

See also

- Managing the New Wireless Network (IEEE 802.11) Policies Settings
- Managing the New Wired Network (IEEE 802.3) Policies Settings
- Advanced Security Settings for Wired and Wireless Network Policies

Windows Firewall overview

Article • 04/07/2025 •

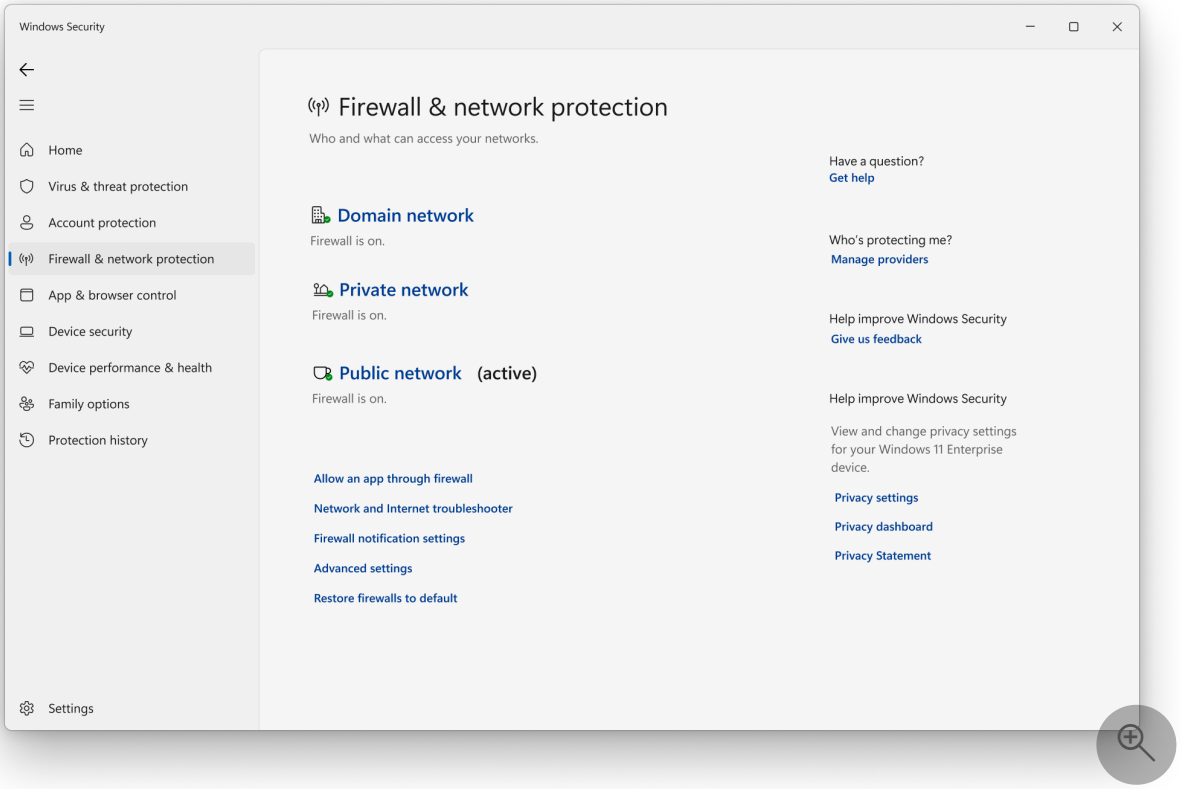
Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows Firewall is a security feature that helps to protect your device by filtering network traffic that enters and exits your device. This traffic can be filtered based on several criteria, including source and destination IP address, IP protocol, or source and destination port number. Windows Firewall can be configured to block or allow network traffic based on the services and applications that are installed on your device. This allows you to restrict network traffic to only those applications and services that are explicitly allowed to communicate on the network.

Windows Firewall is a host-based firewall that is included with the operating system and enabled by default on all Windows editions.

Windows Firewall supports Internet Protocol security (IPsec), which you can use to require authentication from any device that is attempting to communicate with your device. When authentication is required, devices that can't be authenticated as a *trusted device* can't communicate with your device. You can use IPsec to require that certain network traffic is encrypted to prevent it from being read by network packet analyzers that could be attached to the network by a malicious user.

Windows Firewall also works with Network Location Awareness so that it can apply security settings appropriate to the types of networks to which the device is connected. For example, Windows Firewall can apply the *public network* profile when the device is connected a coffee shop wi-fi, and the *private network* profile when the device is connected to the home network. This allows you to apply more restrictive settings to public networks to help keep your device secure.



Practical applications

Windows Firewall offers several benefits to address your organization's network security challenges:

- Reduced risk of network security threats: By reducing the attack surface of a device, Windows Firewall provides an additional layer of defense to the defense-in-depth model. This increases manageability and decreases the likelihood of a successful attack
- Protection of sensitive data and intellectual property: Windows Firewall integrates with IPsec to provide a simple way to enforce authenticated, end-to-end network communications. This allows for scalable, tiered access to trusted network resources, helping to enforce data integrity and, if necessary, protect data confidentiality
- Extended value of existing investments: Windows Firewall is a host-based firewall included with the operating system, so no additional hardware or software is required. It's also designed to complement existing non-Microsoft network security solutions through a documented API

Windows edition and licensing requirements

The following table lists the Windows editions that support Windows Firewall:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Windows Firewall license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Concepts

The default behavior of Windows Firewall is to:

- block all incoming traffic, unless solicited or matching a *rule*
- allow all outgoing traffic, unless matching a *rule*

Firewall rules

Firewall rules identify allowed or blocked network traffic, and the conditions for this to happen. The rules offer an extensive selection of conditions to identify traffic, including:

- Application, service or program name
- Source and destination IP addresses
- Can make use dynamic values, like default gateway, DHCP servers, DNS servers and local subnets
- Protocol name or type. For transport layer protocols, TCP and UDP, you can specify ports or port ranges. For custom protocols, you can use a number between 0 and 255 representing the IP protocol
- Interface type
- ICMP/ICMPv6 traffic type and code

Firewall profiles

Windows Firewall offers three network profiles: domain, private and public. The network profiles are used to assign rules. For example, you can allow a specific application to

communicate on a private network, but not on a public network.

Domain network

The *domain network* profile is automatically applied to a device that is joined to an Active Directory domain, when it detects the availability of a domain controller. This network profile cannot be set manually.

> **Tip**
>
> Another option to detect the *domain network* is to configure the policy settings in the NetworkListManager Policy CSP, which applies to Microsoft Entra joined devices too.

Private network

The *private network* profile is designed for private networks such as a home network. It can be set manually on a network interface by an administrator.

Public network

The *public network* profile is designed with higher security in mind for public networks, like Wi-Fi hotspots, coffee shops, airports, hotels, etc. It's the default profile for unidentified networks.

> **Tip**
>
> Use the PowerShell cmdlet `Get-NetConnectionProfile` to retrieve the active network category (`NetworkCategory`). Use the PowerShell cmdlet `Set-NetConnectionProfile` to switch the category between *private* and *public*.

Disable Windows Firewall

Microsoft recommends that you don't disable Windows Firewall because you lose other benefits, such as the ability to use Internet Protocol security (IPsec) connection security rules, network protection from attacks that employ network fingerprinting, Windows Service Hardening, and boot time filters. Non-Microsoft firewall software can programmatically disable only the rule types of Windows Firewall that need to be disabled for compatibility. You shouldn't disable the firewall yourself for this purpose. If

disabling Windows Firewall is required, don't disable it by stopping the Windows Firewall service (in the Services snap-in, the display name is Windows Defender Firewall and the service name is MpsSvc). Stopping the Windows Firewall service isn't supported by Microsoft and can cause problems, including:

- Start menu can stop working
- Modern applications can fail to install or update
- Activation of Windows via phone fails
- Application or OS incompatibilities that depend on Windows Firewall

The proper method to disable the Windows Firewall is to disable the Windows Firewall Profiles and leave the service running. See Manage Windows Firewall with the command line for detailed steps.

Next steps

Learn about Windows Firewall rules and design recommendations:

Windows Firewall rules >

Provide feedback

To provide feedback for Windows Firewall, open **Feedback Hub** (`WIN` + `F`) and use the category **Security and Privacy** > **Network protection**.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Firewall rules

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

In many cases, a first step for administrators is to customize the firewall profiles using *firewall rules*, so that they can work with applications or other types of software. For example, an administrator or user may choose to add a rule to accommodate a program, open a port or protocol, or allow a predefined type of traffic.

This article describes the concepts and recommendations for creating and managing firewall rules.

Rule precedence for inbound and outbound rules

In many cases, allowing specific types of inbound traffic is required for applications to function in the network. Administrators should keep the following rule precedence behaviors in mind when configuring inbound exceptions:

1. Explicitly defined allow rules take precedence over the default block setting.
2. Explicit block rules take precedence over any conflicting allow rules.
3. More specific rules take precedence over less specific rules, except if there are explicit block rules as mentioned in 2. For example, if the parameters of rule 1 include an IP address range, while the parameters of rule 2 include a single IP host address, rule 2 takes precedence.

Because of 1 and 2, when designing a set of policies, you should make sure that there are no other explicit block rules that could inadvertently overlap, thus preventing the traffic flow you wish to allow.

Outbound rules follow the same precedence behaviors.

> **Note**
>
> Windows Firewall doesn't support weighted, administrator-assigned rule ordering. An effective policy set with expected behaviors can be created by keeping in mind the few, consistent, and logical rule behaviors as described.

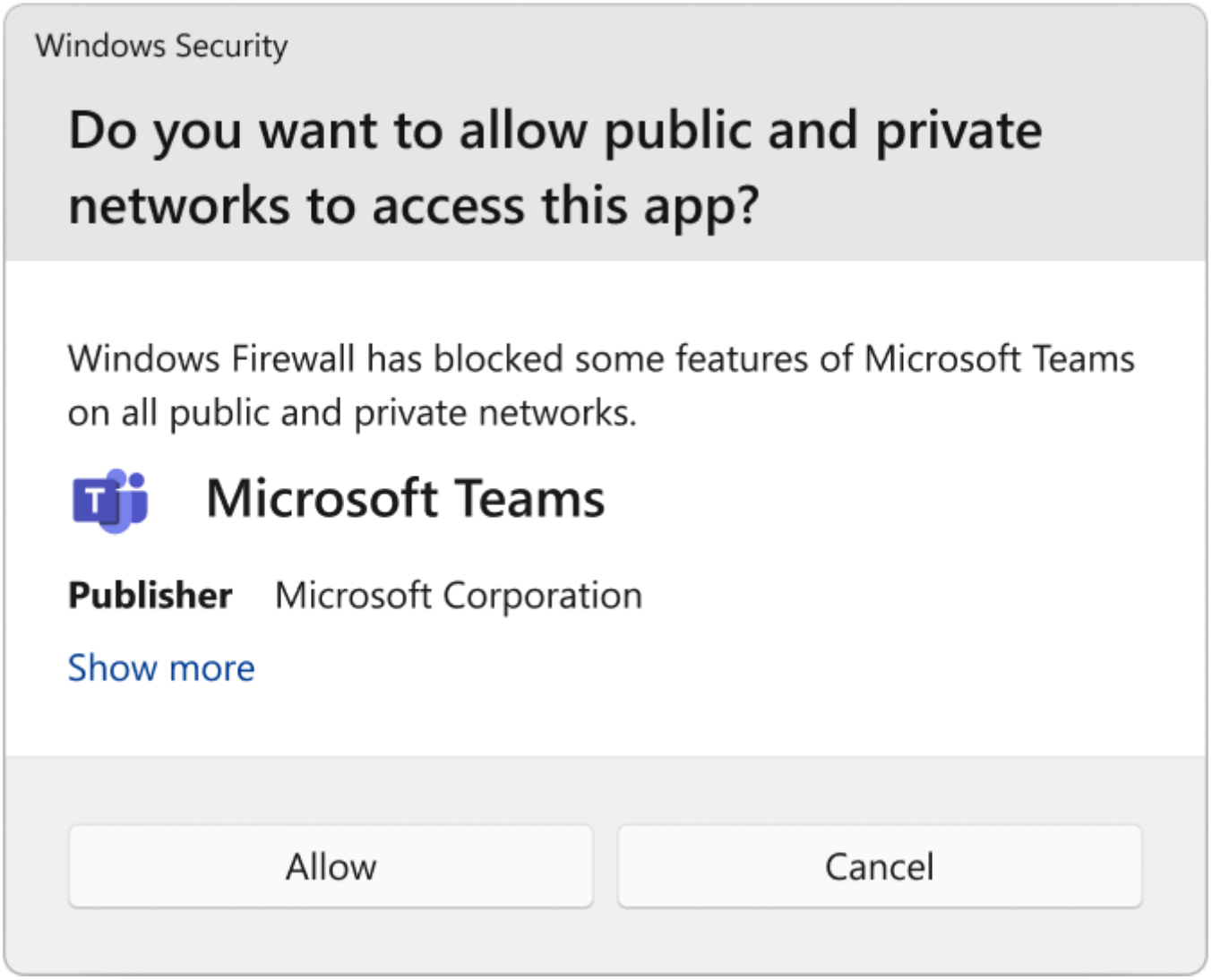
Applications rules

When first installed, network applications and services issue a *listen call* specifying the protocol/port information required for them to function properly. Since there's a default *block* action in Windows Firewall, you must create inbound exception rules to allow the traffic. It's common for the app or the app installer itself to add this firewall rule. Otherwise, the user (or firewall admin on behalf of the user) needs to manually create a rule.

If there's no active application or administrator-defined allow rule(s), a dialog box prompts the user to either allow or block an application's packets the first time the app is launched or tries to communicate in the network:

- If the user has admin permissions, they're prompted. If they respond *No* or cancel the prompt, block rules are created. Two rules are typically created, one each for TCP and UDP traffic.
- If the user isn't a local admin and they are prompted, block rules are created. It doesn't matter what option is selected.

To disable the notification prompt, you can use the command line or the **Windows Firewall with Advanced Security** console



In either of these scenarios, once the rules are added, they must be deleted to generate the prompt again. If not, the traffic continues to be blocked.

> **Note**
>
> The firewall's default settings are designed for security. Allowing all inbound connections by default introduces the network to various threats. Therefore, creating exceptions for inbound connections from non-Microsoft software should be determined by trusted app developers, the user, or the admin on behalf of the user.

> **Note**
>
> The use of wildcard patterns, such as `C:\*\teams.exe` isn't supported in application rules. You can only create rules using the full path to the application(s).

App Control tagging policies

Windows Firewall supports the use of App Control for Business Application ID (AppID) tags in firewall rules. With this capability, Windows Firewall rules can be scoped to an application or a group of applications by referencing process tags, without using absolute path or sacrificing security. There are two steps for this configuration:

1. Deploy *App Control AppId tagging policies*: an App Control for Business policy must be deployed, which specifies individual applications or groups of applications to apply a *PolicyAppId tag* to the process token(s). Then, the admin can define firewall rules that are scoped to all processes tagged with the matching *PolicyAppId*. For more information, see the App Control AppId tagging guide to create, deploy, and test an AppID policy to tag applications.
2. Configure firewall rules using *PolicyAppId tags* using one of the two methods:

    - Using the PolicyAppId node of the Firewall CSP with an MDM solution like Microsoft Intune. If you use Microsoft Intune, you can deploy the rules from Microsoft Intune Admin center, under the path **Endpoint security** > **Firewall** > **Create policy** > **Windows 10, Windows 11, and Windows Server** > **Windows Firewall Rules**. When creating the rules, provide the *AppId tag* in the **Policy App ID** setting.
    - Create local firewall rules with PowerShell: use the New-NetFirewallRule cmdlet and specify the `-PolicyAppId` parameter. You can specify one tag at a time while creating firewall rules. Multiple User Ids are supported.

Local policy merge and application rules

Rule merging policy settings control how rules from different policy sources can be combined. Administrators can configure different merge behaviors for *Domain*, *Private*, and *Public profiles*.

The rule-merging policy settings either allow or prevent local administrators from creating their own firewall rules in addition to those rules obtained from CSP or GPO.

Expand table

	Path
CSP	Domain Profile: `./Vendor/MSFT/Firewall/MdmStore/DomainProfile/`AllowLocalPolicyMerge Private Profile`./Vendor/MSFT/Firewall/MdmStore/PrivateProfile/`AllowLocalPolicyMerge Public Profile `./Vendor/MSFT/Firewall/MdmStore/PublicProfile/`AllowLocalPolicyMerge
GPO	**Computer Configuration** > **Windows Settings** > **Security Settings** > **Windows Defender Firewall with Advanced Security**

Administrators may disable *LocalPolicyMerge* in high-security environments to maintain tighter control over endpoints. This setting can impact some applications and services that automatically generate a local firewall policy upon installation.

> **Important**
>
> If merging of local policies is disabled, centralized deployment of rules is required for any app that needs inbound connectivity.

It's important to create and maintain a list of such apps, including the network ports used for communications. Typically, you can find what ports must be open for a given service on the app's website. For more complex deployments, a thorough analysis might be needed using network packet capture tools.

In general, to maintain maximum security, admins should only deploy firewall exceptions for apps and services determined to serve legitimate purposes.

> **Note**
>
> Local Group Policy firewall rules and settings are applied even when rule merging is disabled through MDM or Group Policy. Local Group Policies are configured using either **secpol.msc** (Local Security Policy) or **gpedit.msc** (Group Policy Editor), which must be run locally on the client device. Only users with local administrator privileges can access these

tools. The settings configured through Local Group Policy are stored in the same location as those managed centrally through Group Policy.

Firewall rules recommendations

Here's a list of recommendations when designing your firewall rules:

- Maintain the default Windows Firewall settings whenever possible. The settings are designed to secure your device for use in most network scenarios. One key example is the default *block behavior* for inbound connections.
- Create your rules in all three profiles, but only enable the firewall rule group on the profiles that suit your scenarios. For example, if you are installing a sharing application that is only used on a private network, then it would be best to create firewall rules in all three profiles, but only enable the firewall rule group containing your rules on the private profile.
- Configure restrictions on your firewall rules depending on which profile the rules are applied to. For applications and services that are designed to only be accessed by devices within a home or small business network, it's best to modify the remote address restriction to specify *Local Subnet* only. The same application or service wouldn't have this restriction when used in an enterprise environment. This can be done by adding the remote address restriction to rules that are added to the private and public profiles, while leaving them unrestricted in the domain profile. This remote address restriction shouldn't apply to applications or services that require global Internet connectivity.
- A general security recommended practice when creating inbound rules is to be as specific as possible. However, when new rules must be made that use ports or IP addresses, consider using consecutive ranges or subnets instead of individual addresses or ports where possible. This approach avoids creation of multiple filters under the hood, reduces complexity, and helps to avoid performance degradation.
- When creating an inbound or outbound rule, you should specify details about the app itself, the port range used, and important notes like creation date. Rules must be well-documented for ease of review both by you and other admins.
- To maintain maximum security, admins should only deploy firewall exceptions for apps and services determined to serve legitimate purposes.

Known issues with automatic rule creation

When designing a set of firewall policies for your network, it's a recommended practice to configure *allow rules* for any networked applications deployed on the host. Having the rules in place before the user first launches the application helps to ensure a seamless experience.

The absence of these staged rules doesn't necessarily mean that in the end an application will be unable to communicate on the network. However, the behaviors involved in the automatic creation of application rules at runtime require user interaction and administrative privilege.

If the device is expected to be used by non-administrative users, you should follow best practices and:

- Provide these rules before the application's first launch to avoid unexpected networking issues.
- Disable inbound notifications on all profiles. This disables the automatic creation of firewall rules.

To determine why some applications are blocked from communicating in the network, check for the following instances:

1. A user with sufficient privileges receives a query notification advising them that the application needs to make a change to the firewall policy. Not fully understanding the prompt, the user cancels or dismisses the prompt. Block rules are created.
2. A user with sufficient privileges is **not** prompted because notifications are disabled. No Allow rules are created, the traffic is blocked by the default block rule.
3. A user lacks sufficient privileges and is prompted to allow the application to make the appropriate policy changes. No matter what he clicks, Block rules get created for the application.
4. A user lacks sufficient privileges and is **not** prompted because notifications are disabled. No Allow rules are created, the traffic is blocked by the default block rule.
5. Local policy merge is disabled, preventing the application or network service from creating local rules.

Outbound rules considerations

What follows are a few general guidelines for configuring outbound rules.

- Changing the outbound rules to *blocked* can be considered for certain highly secure environments. However, the inbound rule configuration should never be changed in a way that allows all traffic by default.
- It's recommended to *allow outbound* by default for most deployments for the sake of simplification with app deployments, unless the organization prefers tight security controls over ease-of-use.
- In high security environments, an inventory of all apps should be logged and maintained. Records must include whether an app used requires network connectivity. Administrators need to create new rules specific to each app that needs network connectivity and push those rules centrally, via GPO or CSP.

Next steps

Learn about the tools to configure Windows Firewall and firewall rules:

Configuration tools >

Last updated on 06/06/2025

Windows Firewall tools

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows offers different tools to view the status and configure Windows Firewall. All tools interact with the same underlying services, but provide different levels of control over those services:

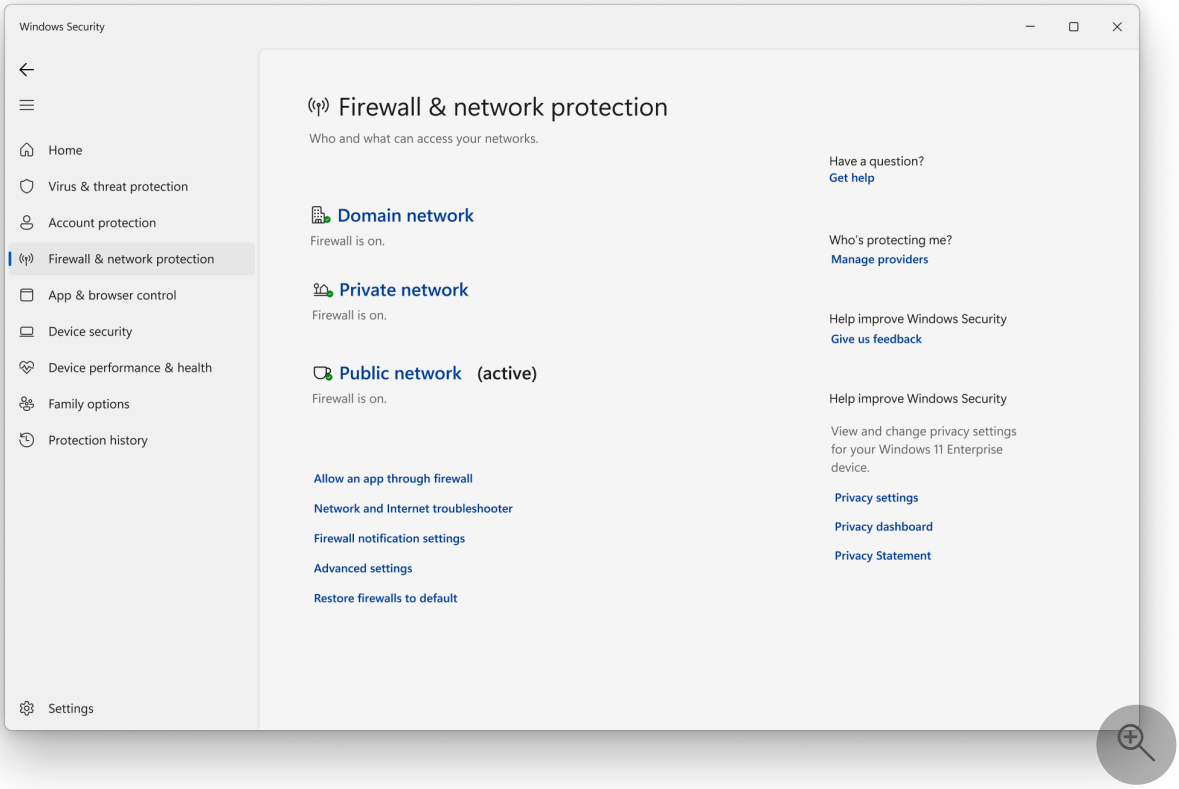
- Windows Security
- Control Panel
- Windows Defender Firewall with Advanced Security (WFAS)
- Configuration Service Provider (CSP)
- Command line tools

> **Note**
>
> To change the configuration of Windows Firewall on a device, you must have administative rights.

Windows Security

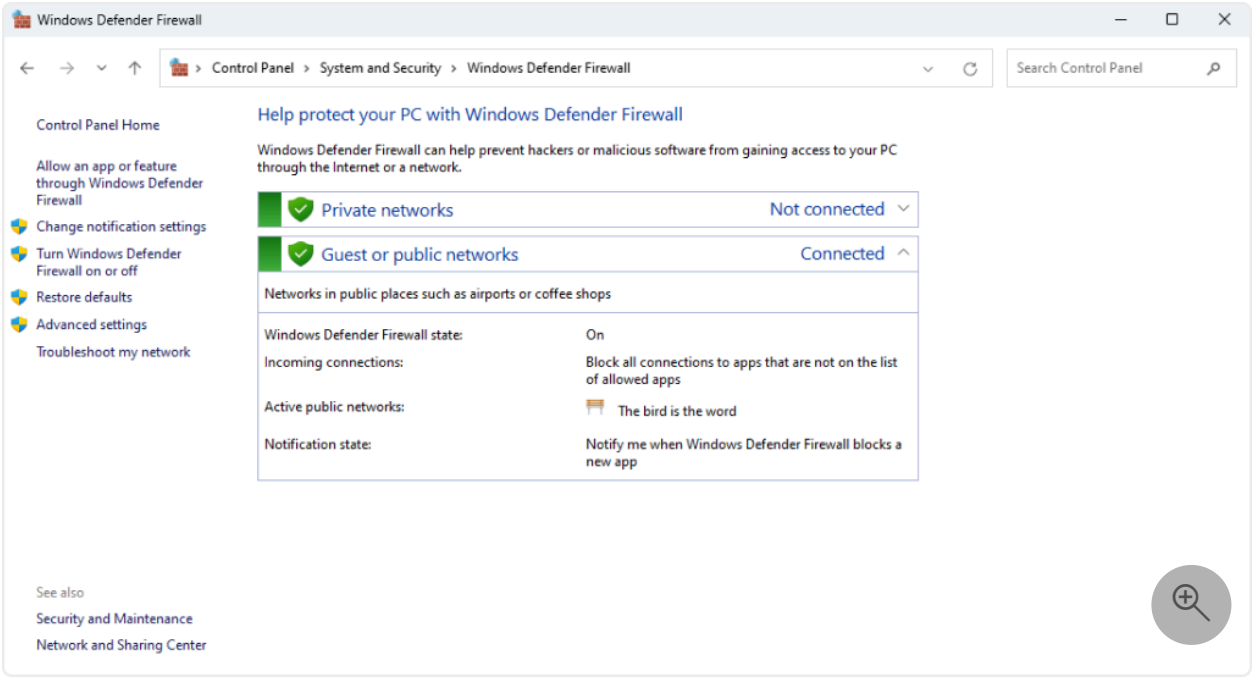
The *Windows Security* app can be used to view the Windows Firewall status and access advanced tools to configure it. Select `START`, type `Windows Security`, and press `ENTER`. Once Windows Security is open, select the tab **Firewall & network protection**. Or use the following shortcut:

Open Firewall & network protection



Control Panel

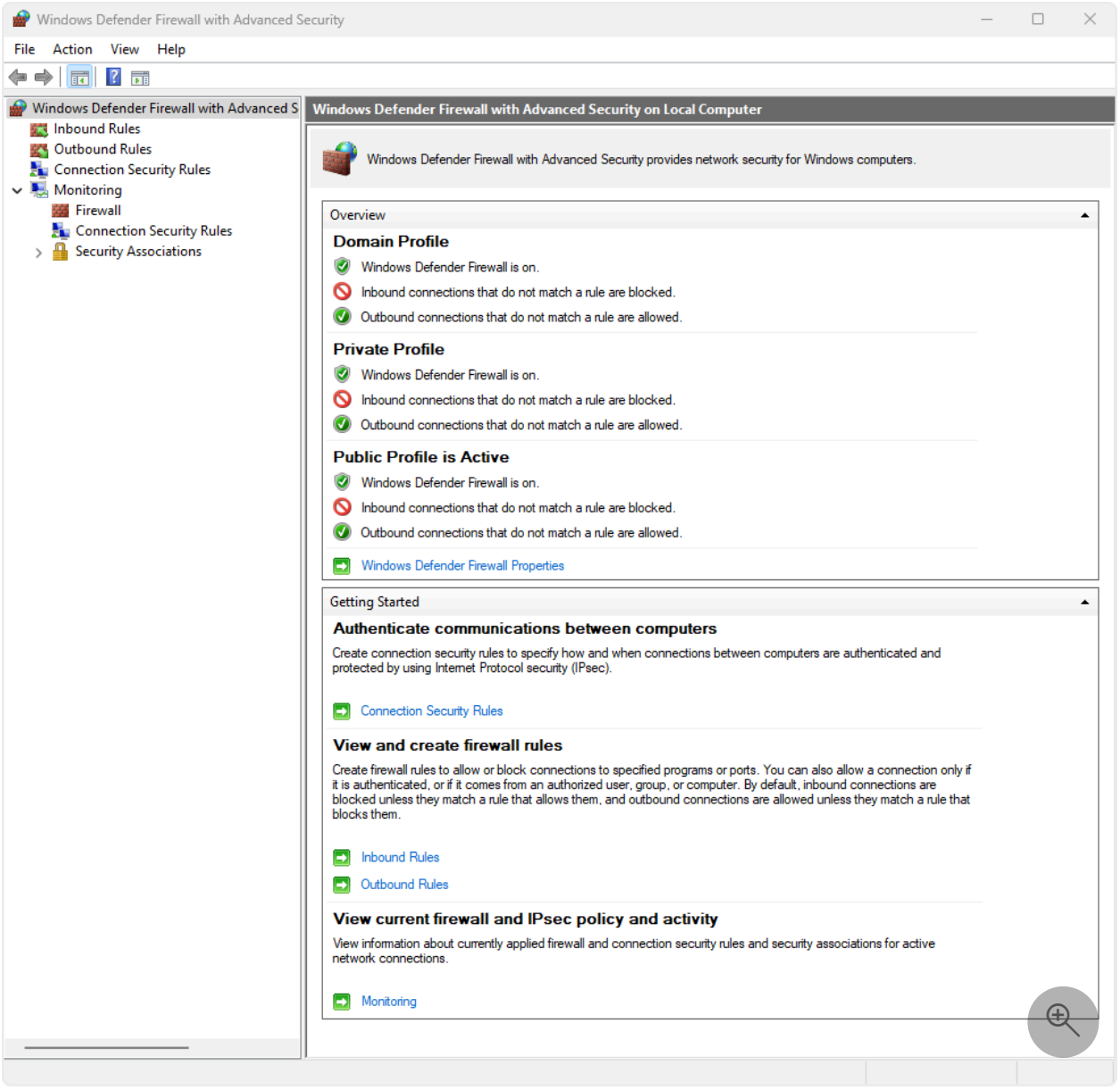
The *Windows Defender Firewall* Control Panel applet provides basic functionalities to configure Windows Firewall. Select `START`, type `firewall.cpl`, and press `ENTER`.



Windows Defender Firewall with Advanced Security

The *Windows Defender Firewall with Advanced Security* (WFAS) is a Microsoft Management Console (MMC) snap-in that provides advanced configuration functionalities. It can be used locally and in group policy (GPO) implementations.

- If you are configuring a single device, select `START`, type `wf.msc`, and press `ENTER`
- If you're configuring devices joined to an Active Directory domain, create or edit a group policy object (GPO) and expand the nodes **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Windows Firewall with Advanced Security**



Configuration Service Provider (CSP)

The Firewall CSP provides an interface to configure and query the status of Windows Firewall, which can be used with a mobile device management (MDM) solution like Microsoft Intune.

Command line tools

The `NetSecurity` PowerShell module and `Network Command Shell (netsh.exe)` are command line utilities that can be used to query the status and configure Windows Firewall.

Group policy processing considerations

The Windows Firewall policy settings are stored in the registry. By default, group policies are refreshed in the background every 90 minutes, with a random offset between 0 and 30 minutes.

Windows Firewall monitors the registry for changes, and if something is written to the registry it notifies the *Windows Filtering Platform (WFP)*, which performs the following actions:

1. Reads all firewall rules and settings
2. Applies any new filters
3. Removes the old filters

> **Note**
>
> The actions are triggered whenever something is written to, or deleted from the registry location the GPO settings are stored, regardless if there's really a configuration change. During the process, IPsec connections are disconnected.

Many policy implementations specify that they're updated only when changed. However, you might want to update unchanged policies, such as reapplying a desired policy setting in case a user has changed it. To control the behavior of the registry group policy processing, you can use the policy **Computer Configuration** > **Administrative Templates** > **System** > **Group Policy** > **Configure registry policy processing**. The **Process even if the Group Policy objects haven't changed** option updates and reapplies the policies even if the policies haven't changed. This option is disabled by default.

If you enable the option **Process even if the Group Policy objects haven't changed**, the WFP filters get reapplied at **every** background refresh. In case you have 10 group policies, the WFP filters get reapplied 10 times during the refresh interval. If an error happens during policy processing, the applied settings might be incomplete, resulting in issues like:

- Windows Firewall blocks inbound or outbound traffic allowed by group policies
- Local Firewall settings are applied instead of group policy settings
- IPsec connections can't establish

The temporary solution is to refresh the group policy settings, using the command `gpupdate.exe /force`, which requires connectivity to a domain controller.

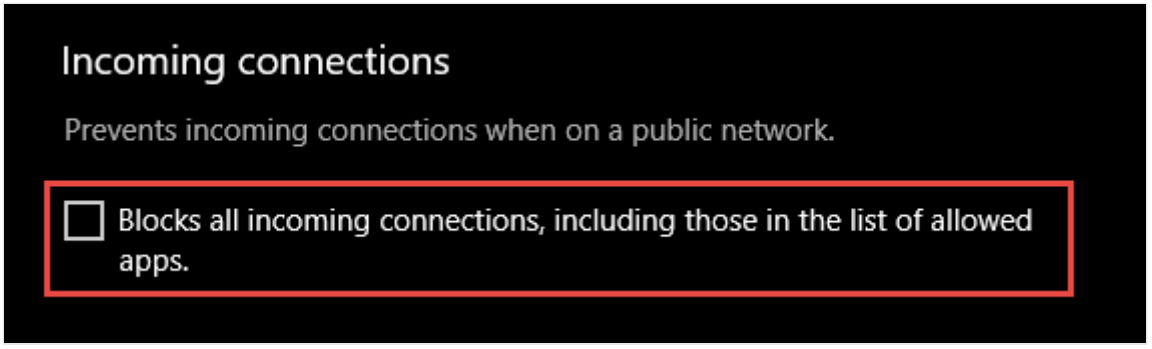
To avoid the issue, leave the policy **Configure registry policy processing** to the default value of **Not Configured** or, if already configured, configure it **Disabled**.

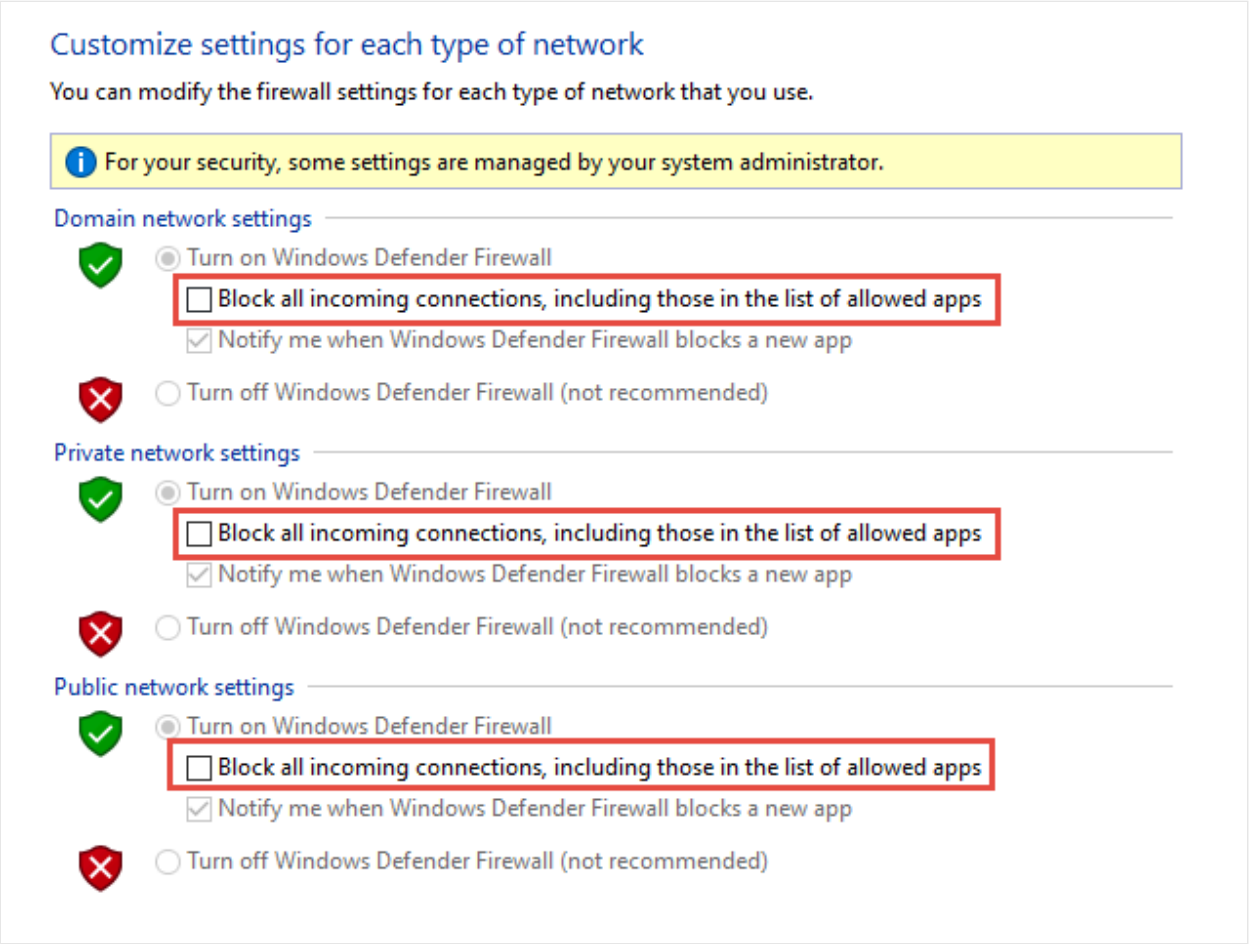
> **ⓘ Important**
>
> The checkbox next to **Process even if the Group Policy objects have not changed** must be unchecked. If you leave it unchecked, WFP filters are written only in case there's a configuration change.
>
> If there's a requirement to force registry deletion and rewrite, then disable background processing by checking the checkbox next to **Do not apply during periodic background processing**.

Shields up mode for active attacks

An important Windows Firewall feature you can use to mitigate damage during an active attack is the *shields up* mode. It's an informal term referring to an easy method a firewall administrator can use to temporarily increase security in the face of an active attack.

Shields up can be achieved by checking **Block all incoming connections, including those in the list of allowed apps** setting found in either the Windows Settings app or Control Panel.





By default, the Windows Firewall blocks everything unless there's an exception rule created. The *shield up* option overrides the exceptions. For example, the Remote Desktop feature automatically creates firewall rules when enabled. However, if there's an active exploit using multiple ports and services on a host, you can, instead of disabling individual rules, use the shields up mode to block all inbound connections, overriding previous exceptions, including the rules for Remote Desktop. The Remote Desktop rules remain intact but remote access can't work as long as shields up is active.

Once the emergency is over, uncheck the setting to restore regular network traffic.

Next steps

From the following dropdown, select one of tools to learn how to configure Windows Firewall:

Feedback

Was this page helpful? Yes No

Provide product feedback

Firewall policy for endpoint security in Intune

Use the endpoint security Firewall policy in Intune to configure a devices built-in firewall for devices that run macOS and Windows devices.

While you can configure the same firewall settings by using Endpoint Protection profiles for device configuration, the device configuration profiles include additional categories of settings. These additional settings are unrelated to firewalls and can complicate the task of configuring only firewall settings for your environment.

Find the endpoint security policies for firewalls under *Manage* in the **Endpoint security** node of the Microsoft Intune admin center.

Prerequisites for Firewall profiles

- Windows
- Windows Server 2012 R2 or later *(through the Microsoft Defender for Endpoint Security settings management scenario)*
- Any supported version of macOS

> **Important**
>
> On October 14, 2025, Windows 10 reached end of support and won't receive quality and feature updates. Windows 10 is an **allowed** version in Intune. Devices running this version can still enroll in Intune and use eligible features, but functionality won't be guaranteed and can vary.

> **Important**
>
> Windows has updated how the Windows Firewall configuration service provider (CSP) enforces rules from Atomic blocks of firewall rules. The *Windows Firewall CSP* on a device implements the firewall rule settings from your Intune endpoint security Firewall policies. Beginning with the following versions of Windows, the updated CSP behavior now enforces an all-or-nothing application of firewall rules from each Atomic block of rules:

- Windows 11 21H2
- Windows 11 22H2
- Windows 10 21H2

On devices that run an earlier version of Windows, the CSP processes firewall rules in an Atomic block of rules, one rule (or setting) at a time. The intent is to apply all the rules in that Atomic block, or none of them. However, if the CSP encounters an issue with applying any rule from the block, the CSP stops applying subsequent rules but doesn't roll back a rule from that block that has already been applied successfully. This behavior can result in a partial deployment of firewall rules on a device.

Role-based access controls (RBAC)

For guidance on assigning the right level of permissions and rights to manage Intune firewall policy, see Role-based access control for endpoint security.

Firewall profiles

Find guidance for creating endpoint security profiles at Create an endpoint security policy.

Devices managed by Intune

Platform: macOS:

- **macOS firewall** – Enable and configure settings for the built-in firewall on macOS.

Platform: Windows:

For information about configuring settings in the following profiles, see the Firewall configuration service provider (CSP).

> ⓘ **Note**
>
> Beginning on April 5, 2022, the *Windows 10 and later* platform was replaced by the *Windows* platform.
>
> The *Windows* platform supports devices communicating through Microsoft Intune or Microsoft Defender for Endpoint. These profiles also add support for the Windows Server platform which is not supported through Microsoft Intune natively.

Profiles for this new platform use the settings format as found in the Settings Catalog. Each new profile template for this new platform includes the same settings as the older profile template it replaces. With this change you can no longer create new versions of the old profiles. Your existing instances of the old profile remain available to use and edit.

- **Windows Firewall** – Configure settings for Windows Firewall with Advanced Security. Windows Firewall provides host-based, two-way network traffic filtering for a device and can block unauthorized network traffic flowing into or out of the local device.
- **Windows Firewall rules** - Define granular Firewall rules, including specific ports, protocols, applications and networks, and to allow or block network traffic. Each instance of this profile supports up to 150 custom rules.

  > **Tip**
  >
  > Use of the **Policy App Id** setting, which is described in the ***MdmStore/FirewallRules/{FirewallRuleName}/PolicyAppId*** CSP, requires that your environment use *Windows Defender Application Control* (WDAC) tagging. For more information, see the following Windows Defender articles:
  > - **About application control for Windows**
  > - **WDAC Application ID (AppId) Tagging guide**

- **Windows Hyper-V Firewall Rules** The Windows Hyper-V Firewall Rules template allows you to control firewall rules that will apply to specific Hyper-V containers on Windows, including applications like the Windows Subsystem for Linux (WSL) and the Windows Subsystem for Android (WSA)

Add reusable settings groups to profiles for Firewall rules

In public preview, Windows Firewall rule profiles support use of reusable settings groups for the following platforms:

- Windows

The following firewall rule profile settings are available in reusable settings groups:

- Remote IP address ranges
- FQDN definitions and auto-resolution

When you configure a firewall rule to add one or more reusable settings groups, you'll also configure the rules Action to define how the settings in those groups are used.

Each rule you add to the profile can include both reusable settings groups and individual settings that are added directly to the rule. However, consider using each rule for either reusable settings groups or to manage settings you add directly to the rule. This separation can help simplify future configurations or changes you might make.

> ⓘ **Note**
>
> Inbound FQDN rules aren't natively supported. However, it's possible to use *pre-hydration* scripts to generate inbound IP entries for the rule. For more information, see **Windows Firewall dynamic keywords** in the Windows Firewall documentation.

For prerequisites and guidance on configuring reusable groups, and then adding them to this profile, see Use reusable groups of settings with Intune policies.

Devices managed by Configuration Manager

Firewall

Support for devices managed by Configuration Manager is in Preview.

Manage Firewall policy settings for Configuration Manager devices, when you use tenant attach.

Policy path:

- Endpoint security > Firewall

Profiles:

- Windows Firewall (ConfigMgr)

Required version of Configuration Manager:

- Configuration Manager current branch version 2006 or later, with in-console update Configuration Manager 2006 Hotfix (KB4578605)

Supported Configuration Manager device platforms:

- Windows (x86, x64, ARM64)

> **ⓘ Important**
>
> On October 14, 2025, Windows 10 reached end of support and won't receive quality and feature updates. Windows 10 is an **allowed** version in Intune. Devices running this version can still enroll in Intune and use eligible features, but functionality won't be guaranteed and can vary.

Firewall rule mergers and policy conflicts

Plan for Firewall policies to be applied to a device using only one policy. Use of a single policy instance and policy type helps avoid having two separate policies apply different configurations to the same setting, which creates conflicts. When a conflict exists between two policy instances or types of policy that manage the same setting with different values, the setting isn't sent to the device.

- That form of policy conflict applies to the **Windows Firewall** profile, which can conflict with other Windows Firewall profiles, or a firewall configuration that's delivered by a different policy type, like device configuration.

  Windows Firewall profiles don't conflict with *Windows Firewall rules* profiles.

When you use **Windows Firewall rules** profiles, you can apply multiple rules profiles to the same device. However, when different rules exist for the same thing with different configurations, both are sent to the device and create a conflict, on that device.

- For example, if one rule blocks *Teams.exe* through the firewall and a second rule allows *Teams.exe*, both rules are delivered to the client. This result is different from conflicts created through other policies for Firewall settings.

When rules from multiple rules profiles don't conflict with each other, devices merge the rules from each profile to create a combined firewall rule configuration on the device. This behavior enables you to deploy more than the 150 rules that each individual profile supports to a device.

- For example, you have two Windows Firewall rules profiles. The first profile allows *Teams.exe* through the firewall. The second profile allows *Outlook.exe* through the firewall. When a device receives both profiles, the device is configured to allow both apps through the firewall.

Firewall policy reports

The reports for Firewall policy display status details about the firewall status for your managed devices. Firewall reports support managed devices that run the following operating systems.

- Windows

> **Important**
>
> On October 14, 2025, Windows 10 reached end of support and won't receive quality and feature updates. Windows 10 is an **allowed** version in Intune. Devices running this version can still enroll in Intune and use eligible features, but functionality won't be guaranteed and can vary.

Summary

Summary is the default view when you open the Firewall node. Open the Microsoft Intune admin center, and then go to **Endpoint security** > **Firewall** > **Summary**.

This view provides:

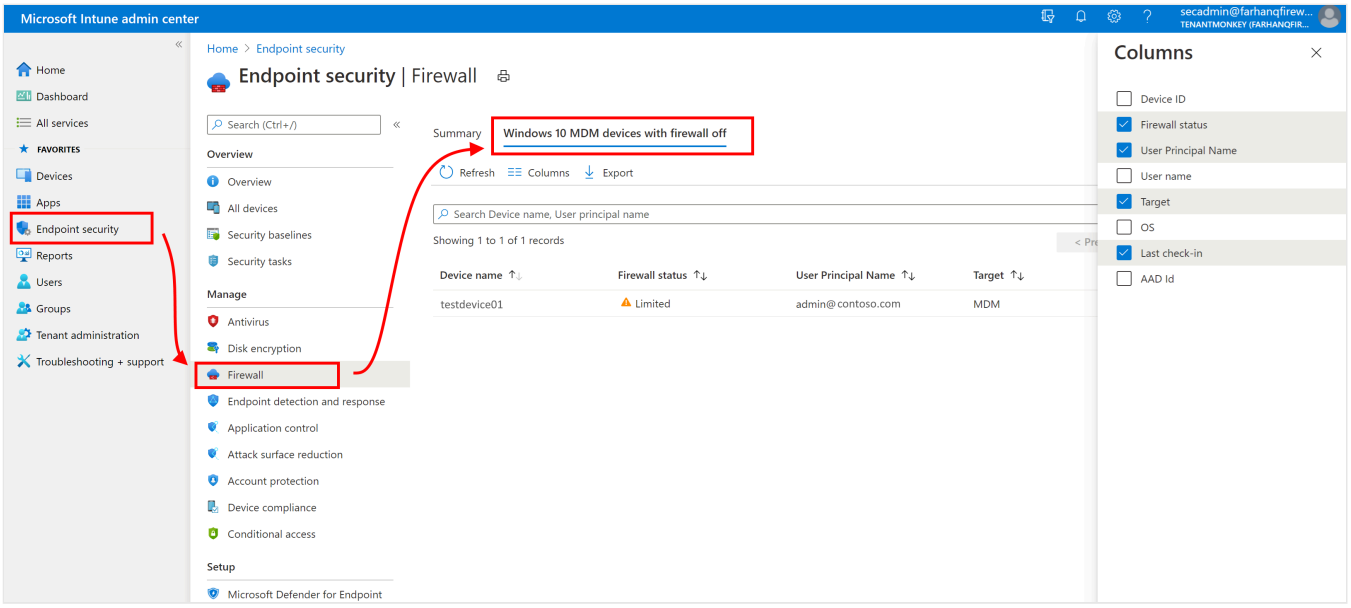
- An aggregate count of devices that have the firewall turned off.
- A list of your Firewall policies, including the name, type, if it's assigned, and when it was last modified.

MDM devices running Windows with firewall off

This report is located in the Endpoint security node. Open the Microsoft Intune admin center, and then go to **Endpoint security** > **Firewall** > **MDM devices running Windows 10 or later with firewall off**.

Data is reported through the Windows DeviceStatus CSP, and identifies each device where the Firewall is off. By default, visible details include:

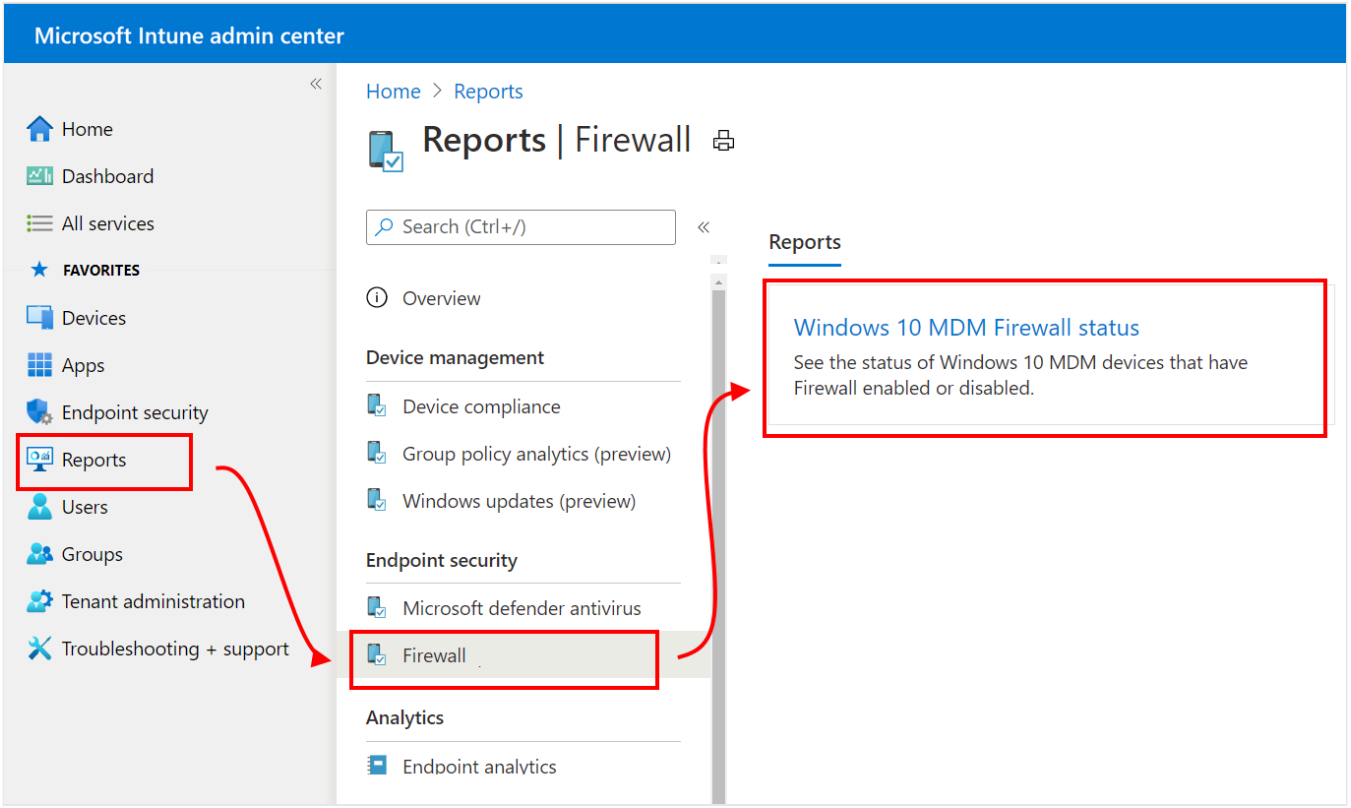
- Device name
- Firewall status
- User principal name
- Target (The method of device management)
- Last check in time



MDM Firewall status for Windows

This organizational report is also described in Intune Reports.

As an organizational report, this report is available from the **Reports** node. Open the Microsoft Intune admin center, and then go to **Reports** > **Firewall** > **MDM Firewall status for Windows**.

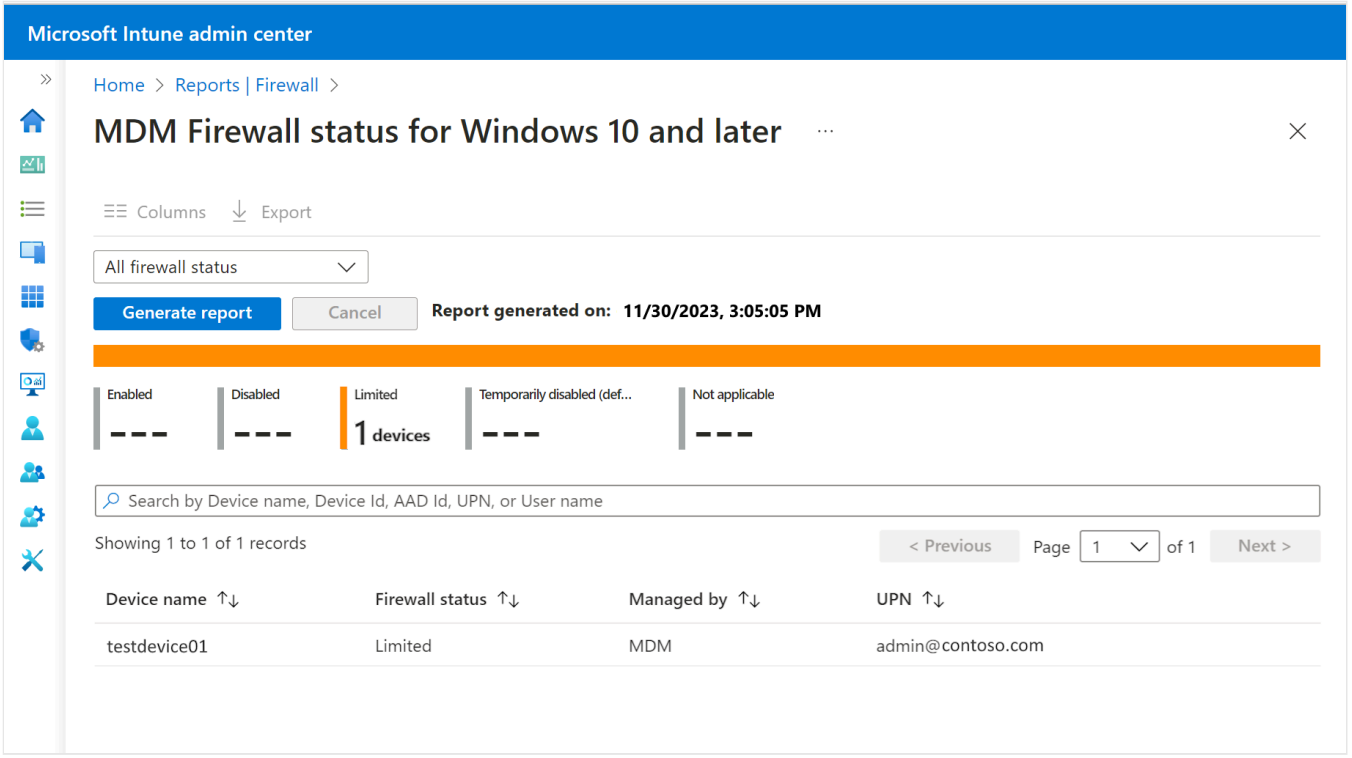


Data is reported through the Windows DeviceStatus CSP, and reports on the status of the firewall on your managed devices. You can filter returns for this report by using one or more of the status detail categories.

Status details include:

- **Enabled** – The firewall on, and successfully reporting.
- **Disabled** - The firewall is turned off.
- **Limited** – The firewall isn't monitoring all networks, or some rules are turned off.
- **Temporarily Disabled (default)** – The firewall is temporarily not monitoring all networks
- **Not applicable** – The device doesn't support firewall reporting.

You can filter returns for this report by using one or more of the status detail categories.



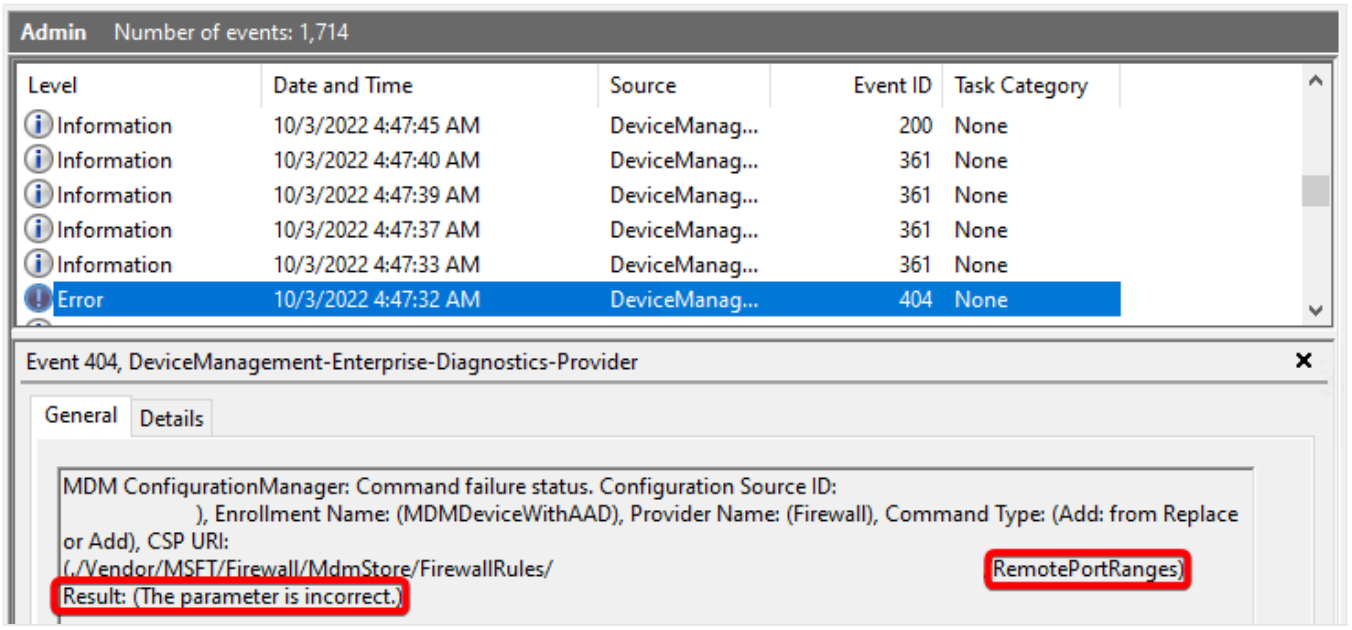
Investigate issues for Firewall rules

To learn more about Firewall rules in Intune, and how to troubleshoot common problems, see the following *Intune Customer Success* blog:

- How to trace and troubleshoot the Intune Endpoint Security Firewall rule creation process

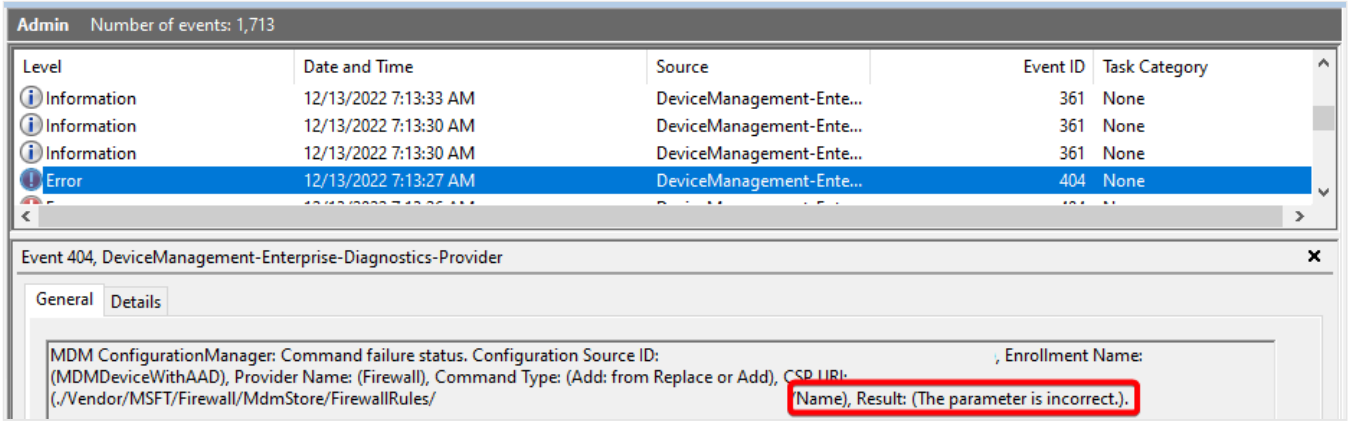
Additional common firewall rule issues:

Event Viewer: RemotePortRanges or LocalPortRanges "The parameter is incorrect"



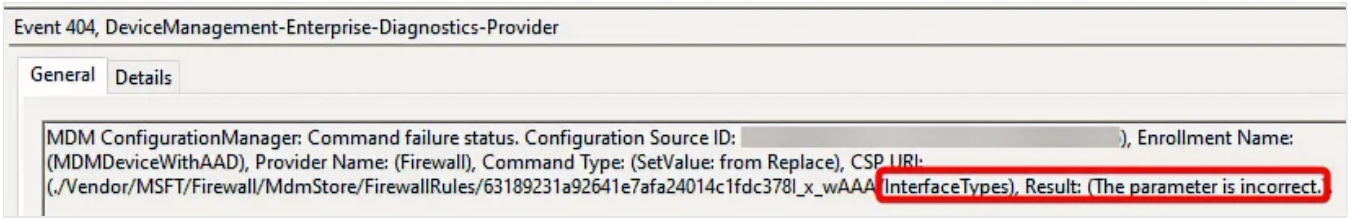
- Verify configured ranges are ascending (Example: 1-5 is correct, 5-1 will cause this error)
- Verify configured ranges are within the overall port range of 0-65535
- If either remote port ranges or local port ranges are configured in a rule, protocol **must** also be configured with 6 (TCP) **or** 17 (UDP)

Event Viewer: "...Name), Result: (The parameter is incorrect)"



- If edge traversal is enabled in a rule, the rule direction **must** be set to "This rule applies to inbound traffic".

Event Viewer: "...InterfaceTypes), Result: (The parameter is incorrect)"



- If "All" interface type is enabled in a rule, the other interface types **must not** be selected.

Next steps

Configure Endpoint security policies

View details for the settings in the deprecated Firewall profiles for the deprecated *Windows 10 and later* platform:

- Firewall profile settings.

Last updated on 04/15/2026

Configure rules with group policy

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article contains examples how to configure Windows Firewall rules using the *Windows Firewall with Advanced Security* console.

Access the Windows Firewall with Advanced Security console

If you're configuring devices joined to an Active Directory domain, to complete these procedures you must be a member of the Domain Administrators group, or otherwise have delegated permissions to modify the GPOs in the domain. To access the *Windows Firewall with Advanced Security* console, create or edit a group policy object (GPO) and expand the nodes **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Windows Firewall with Advanced Security**. Pay attention to the Group policy processing considerations when using Group Policy.

If you are configuring a single device, you must have administrative rights on the device. In which case, to access the *Windows Firewall with Advanced Security* console, select `START`, type `wf.msc`, and press `ENTER`.

Create an inbound ICMP rule

This type of rule allows ICMP requests and responses to be received by devices on the network. To create an inbound ICMP rule:

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Inbound Rules**
3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**
5. On the **Program** page, select **All programs**, and then select **Next**
6. On the **Protocol and Ports** page, select **ICMPv4** or **ICMPv6** from the **Protocol type** list. If you use both IPv4 and IPv6 on your network, you must create a separate ICMP rule for each
7. Select **Customize**
8. In the **Customize ICMP Settings** dialog box, do one of the following:

- To allow all ICMP network traffic, select **All ICMP types**, and then select **OK**
- To select one of the predefined ICMP types, select **Specific ICMP types**, and then select each type in the list that you want to allow. Select **OK**
- To select an ICMP type that does not appear in the list, select **Specific ICMP types**, select the **Type** number from the list, select the **Code** number from the list, select **Add**, and then select the newly created entry from the list. Select **OK**

9. Select **Next**
10. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
11. On the **Action** page, select **Allow the connection**, and then select **Next**
12. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**
13. On the **Name** page, type a name and description for your rule, and then select **Finish**

Create an inbound port rule

This type of rule allows any program that listens on a specified TCP or UDP port to receive network traffic sent to that port. To create an inbound port rule:

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Inbound Rules**
3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**

> ⓘ **Note**
>
> Although you can create rules by selecting **Program** or **Port**, those choices limit the number of pages presented by the wizard. If you select **Custom**, you see all of the pages, and have the most flexibility in creating your rules.

5. On the **Program** page, select **All programs**, and then select **Next**

> ⓘ **Note**
>
> This type of rule is often combined with a program or service rule. If you combine the rule types, you get a firewall rule that limits traffic to a specified port and allows the traffic only when the specified program is running. The

specified program cannot receive network traffic on other ports, and other programs cannot receive network traffic on the specified port. If you choose to do this, follow the steps in the Create an Inbound Program or Service Rule procedure in addition to the steps in this procedure to create a single rule that filters network traffic using both program and port criteria.

6. On the **Protocol and Ports** page, select the protocol type that you want to allow.
   To restrict the rule to a specified port number, you must select either **TCP** or **UDP**.
   Because this is an incoming rule, you typically configure only the local port number
   If you select another protocol, then only packets whose protocol field in the IP header match this rule are permitted through the firewall.
   To select a protocol by its number, select **Custom** from the list, and then type the number in the **Protocol number** box.
   When you have configured the protocols and ports, select **Next**.
7. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
8. On the **Action** page, select **Allow the connection**, and then select **Next**
9. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**

   > ⓘ **Note**
   >
   > If this GPO is targeted at server computers running Windows Server 2008 that never move, consider modifying the rules to apply to all network location type profiles. This prevents an unexpected change in the applied rules if the network location type changes due to the installation of a new network card or the disconnection of an existing network card's cable. A disconnected network card is automatically assigned to the Public network location type.

10. On the **Name** page, type a name and description for your rule, and then select **Finish**

Create an outbound port rule

By default, Windows Firewall allows all outbound network traffic, unless it matches a rule that prohibits the traffic. This type of rule blocks any outbound network traffic that matches the specified TCP or UDP port numbers. To create an outbound port rule:

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Outbound Rules**

3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Outbound Rule wizard, select **Custom**, and then select **Next**

   > ⓘ **Note**
   >
   > Although you can create rules by selecting **Program** or **Port**, those choices limit the number of pages presented by the wizard. If you select **Custom**, you see all of the pages, and have the most flexibility in creating your rules.

5. On the **Program** page, select **All programs**, and then select **Next**
6. On the **Protocol and Ports** page, select the protocol type that you want to block. To restrict the rule to a specified port number, you must select either **TCP** or **UDP**. Because this rule is an outbound rule, you typically configure only the remote port number If you select another protocol, then only packets whose protocol field in the IP header matches this rule are blocked by Windows Defender Firewall. Network traffic for protocols is allowed as long as other rules that match don't block it. To select a protocol by its number, select **Custom** from the list, and then type the number in the **Protocol number** box. When you've configured the protocols and ports, select **Next**
7. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
8. On the **Action** page, select **Block the connection**, and then select **Next**
9. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**
10. On the **Name** page, type a name and description for your rule, and then select **Finish**

Create an inbound program or service rule

This type of rule allows the program to listen and receive inbound network traffic on any port.

> ⓘ **Note**
>
> This type of rule is often combined with a program or service rule. If you combine the rule types, you get a firewall rule that limits traffic to a specified port and allows the traffic only when the specified program is running. The program cannot receive network traffic on other ports, and other programs cannot receive network traffic on the specified port. To combine the program and port rule types into a single

rule, follow the steps in the Create an Inbound Port Rule procedure in addition to the steps in this procedure.

To create an inbound firewall rule for a program or service:

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Inbound Rules**
3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**

   > ⓘ **Note**
   >
   > Information the user should notice even if skimmingAlthough you can create rules by selecting **Program** or **Port**, those choices limit the number of pages presented by the wizard. If you select **Custom**, you see all of the pages, and have the most flexibility in creating your rules.

5. On the **Program** page, select **This program path**
6. Type the path to the program in the text box. Use environment variables, where applicable, to ensure that programs installed in different locations on different computers work correctly.
7. Do one of the following:

   - If the executable file contains a single program, select **Next**
   - If the executable file is a container for multiple services that must all be allowed to receive inbound network traffic, select **Customize**, select **Apply to services only**, select **OK**, and then select **Next**
   - If the executable file is a container for a single service or contains multiple services but the rule only applies to one of them, select **Customize**, select **Apply to this service**, and then select the service from the list. If the service does not appear in the list, select **Apply to service with this service short name**, and then type the short name for the service in the text box. Select **OK**, and then select **Next**

   > ⓘ **Important**

> To use the **Apply to this service** or **Apply to service with this service short name** options, the service must be configured with a security identifier (SID) with a type of **RESTRICTED** or **UNRESTRICTED**. To check the SID type of a service, run the following command: `sc qsidtype <ServiceName>`
>
> If the result is `NONE`, then a firewall rule cannot be applied to that service.

To set a SID type on a service, run the following command: `sc sidtype <ServiceName> <Type>`

In the preceding command, the value of `<Type>` can be `UNRESTRICTED` or `RESTRICTED`. Although the command also permits the value of `NONE`, that setting means the service cannot be used in a firewall rule as described here. By default, most services in Windows are configured as `UNRESTRICTED`. If you change the SID type to `RESTRICTED`, the service might fail to start. We recommend that you change the SID type only on services that you want to use in firewall rules, and that you change the SID type to `UNRESTRICTED`.

8. It is a best practice to restrict the firewall rule for the program to only the ports it needs to operate. On the **Protocols and Ports** page, you can specify the port numbers for the allowed traffic. If the program tries to listen on a port different from the one specified here, it is blocked. For more information about protocol and port options, see Create an Inbound Port Rule. After you have configured the protocol and port options, select **Next**

9. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**

10. On the **Action** page, select **Allow the connection**, and then select **Next**

11. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**

12. On the **Name** page, type a name and description for your rule, and then select **Finish**

Create an outbound program or service rule

By default, Windows Defender Firewall allows all outbound network traffic unless it matches a rule that prohibits the traffic. This type of rule prevents the program from

sending any outbound network traffic on any port. To create an outbound firewall rule for a program or service:

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Outbound Rules**
3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Outbound Rule Wizard, select **Custom**, and then select **Next**

   > ⓘ **Note**
   >
   > Although you can create many rules by selecting **Program** or **Port**, those choices limit the number of pages presented by the wizard. If you select **Custom**, you see all of the pages, and have the most flexibility in creating your rules.

5. On the **Program** page, select **This program path**
6. Type the path to the program in the text box. Use environment variables as appropriate to ensure that programs installed in different locations on different computers work correctly
7. Do one of the following:

   - If the executable file contains a single program, select **Next**
   - If the executable file is a container for multiple services that must all be blocked from sending outbound network traffic, select **Customize**, select **Apply to services only**, select **OK**, and then select **Next**
   - If the executable file is a container for a single service or contains multiple services but the rule only applies to one of them, select **Customize**, select **Apply to this service**, and then select the service from the list. If the service does not appear in the list, then select **Apply to service with this service short name**, and type the short name for the service in the text box. Select **OK**, and then select **Next**

8. If you want the program to be allowed to send on some ports, but blocked from sending on others, then you can restrict the firewall rule to block only the specified ports or protocols. On the **Protocols and Ports** page, you can specify the port numbers or protocol numbers for the blocked traffic. If the program tries to send to or from a port number different from the one specified here, or by using a protocol number different from the one specified here, then the default outbound firewall behavior allows the traffic. For more information about the protocol and port options, see Create an Outbound Port Rule. When you have configured the protocol and port options, select **Next**

9. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
10. On the **Action** page, select **Block the connection**, and then select **Next**
11. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**
12. On the **Name** page, type a name and description for your rule, and then select **Finish**

Create inbound rules to support RPC

To allow inbound remote procedure call (RPC) network traffic, you must create two firewall rules:

- the first rule allows incoming network packets on TCP port 135 to the RPC Endpoint Mapper service. The incoming traffic consists of requests to communicate with a specified network service. The RPC Endpoint Mapper replies with a dynamically assigned port number that the client must use to communicate with the service
- the second rule allows the network traffic that is sent to the dynamically assigned port number

Using the two rules configured as described in this topic helps to protect your device by allowing network traffic only from devices that have received RPC dynamic port redirection and to only those TCP port numbers assigned by the RPC Endpoint Mapper.

RPC Endpoint Mapper service

1. Open the *Windows Firewall with Advanced Security* console
2. In the navigation pane, select **Inbound Rules**
3. Select **Action**, and then select **New rule**
4. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**
5. On the **Program** page, select **This Program Path**, and then type `%systemroot%\system32\svchost.exe`
6. Select **Customize**.
7. In the **Customize Service Settings** dialog box, select **Apply to this service**, select **Remote Procedure Call (RPC)** with a short name of **RpcSs**, select **OK**, and then select **Next**
8. On the warning about Windows service-hardening rules, select **Yes**
9. On the **Protocol and Ports** dialog box, for **Protocol type**, select **TCP**

10. For **Local port**, select **RPC Endpoint Mapper**, and then select **Next**
11. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
12. On the **Action** page, select **Allow the connection**, and then select **Next**
13. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**
14. On the **Name** page, type a name and description for your rule, and then select **Finish**

RPC-enabled network services

1. On the same GPO you edited in the preceding procedure, select **Action**, and then select **New rule**
2. On the **Rule Type** page of the New Inbound Rule Wizard, select **Custom**, and then select **Next**
3. On the **Program** page, select **This Program Path**, and then type the path to the executable file that hosts the network service. Select **Customize**
4. In the **Customize Service Settings** dialog box, select **Apply to this service**, and then select the service that you want to allow. If the service doesn't appear in the list, then select **Apply to service with this service short name**, and then type the short name of the service in the text box
5. Select **OK**, and then select **Next**
6. On the **Protocol and Ports** dialog box, for **Protocol type**, select **TCP**
7. For **Local port**, select **RPC Dynamic Ports**, and then select **Next**
8. On the **Scope** page, you can specify that the rule applies only to network traffic to or from the IP addresses entered on this page. Configure as appropriate for your design, and then select **Next**
9. On the **Action** page, select **Allow the connection**, and then select **Next**
10. On the **Profile** page, select the network location types to which this rule applies, and then select **Next**
11. On the **Name** page, type a name and description for your rule, and then select **Finish**

Feedback

Was this page helpful? Yes No

Provide product feedback

Manage Windows Firewall with the command line

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article provides examples how to manage Windows Firewall with PowerShell and `netsh.exe`, which can be used to automate the management of Windows Firewall.

Set profile global defaults

Global defaults set the device behavior in a per-profile basis. Windows Firewall supports Domain, Private, and Public profiles.

Windows Firewall drops traffic that doesn't correspond to allowed unsolicited traffic, or traffic that is sent in response to a request by the device. If you find that the rules you create aren't enforced, you might need to enable Windows Firewall. Here's how to enable Windows Firewall on a local device:

PowerShell

PowerShell

```powershell
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled True
```

Control Windows Firewall behavior

The global default settings can be defined through the command-line interface. These modifications are also available through the Windows Firewall console. The following scriptlets set the default inbound and outbound actions, specifies protected network connections, and disallows notifications to be displayed to the user when a program is blocked from receiving inbound connections. It allows unicast response to multicast or broadcast network traffic, and it specifies logging settings for troubleshooting.

PowerShell

PowerShell

```
Set-NetFirewallProfile -DefaultInboundAction Block -
DefaultOutboundAction Allow -NotifyOnListen False -
AllowUnicastResponseToMulticast True -LogFileName
%SystemRoot%\System32\LogFiles\Firewall\pfirewall.log
```

Disable Windows Firewall

Microsoft recommends that you don't disable Windows Firewall because you lose other benefits, such as the ability to use Internet Protocol security (IPsec) connection security rules, network protection from attacks that employ network fingerprinting, Windows Service Hardening, and boot time filters. Non-Microsoft firewall software can programmatically disable only the rule types of Windows Firewall that need to be disabled for compatibility. You shouldn't disable the firewall yourself for this purpose. If disabling Windows Firewall is required, don't disable it by stopping the Windows Firewall service (in the Services snap-in, the display name is Windows Defender Firewall and the service name is MpsSvc). Stopping the Windows Firewall service isn't supported by Microsoft and can cause problems, including:

- Start menu can stop working
- Modern applications can fail to install or update
- Activation of Windows via phone fails
- Application or OS incompatibilities that depend on Windows Firewall

The proper method to disable the Windows Firewall is to disable the Windows Firewall Profiles and leave the service running. Use the following procedure to turn off the firewall, or disable the Group Policy setting **Computer Configuration|Administrative Templates|Network|Network Connections|Windows Firewall|Domain Prolfile|Windows Firewall:Protect all network connections**. For more information, see Windows Firewall deployment guide. The following example disables Windows Firewall for all profiles.

PowerShell

```
PowerShell
Set-NetFirewallProfile -Profile Domain,Public,Private -Enabled False
```

Deploy basic firewall rules

This section provides scriptlet examples for creating, modifying, and deleting firewall rules.

Create firewall rules

Adding a firewall rule in Windows PowerShell looks a lot like it did in Netsh, but the parameters and values are specified differently. Here's an example of how to allow the Telnet application to listen on the network. This firewall rule is scoped to the local subnet by using a keyword instead of an IP address. Just like in Netsh, the rule is created on the local device, and it becomes effective immediately.

PowerShell

```
PowerShell

New-NetFirewallRule -DisplayName "Allow Inbound Telnet" -Direction
Inbound -Program %SystemRoot%\System32\tlntsvr.exe -RemoteAddress
LocalSubnet -Action Allow
```

The following scriptlet shows how to add a basic firewall rule that blocks outbound traffic from a specific application and local port to a Group Policy Object (GPO) in Active Directory. In Windows PowerShell, the policy store is specified as a parameter within the **New-NetFirewall** cmdlet. In Netsh, you must first specify the GPO that the commands in a Netsh session should modify. The commands you enter are run against the contents of the GPO, and the execution remains in effect until the Netsh session is ended or until another set store command is executed. Here, **domain.contoso.com** is the name of your Active Directory Domain Services (AD DS), and **gpo_name** is the name of the GPO that you want to modify. Quotation marks are required if there are any spaces in the GPO name.

PowerShell

```
PowerShell

New-NetFirewallRule -DisplayName "Block Outbound Telnet" -Direction
Outbound -Program %SystemRoot%\System32\tlntsvr.exe -Protocol TCP -
LocalPort 23 -Action Block -PolicyStore domain.contoso.com\gpo_name
```

GPO Caching

To reduce the burden on busy domain controllers, Windows PowerShell allows you to load a GPO to your local session, make all your changes in that session, and then save it back at all once. The following command performs the same actions as the previous example (by adding a Telnet rule to a GPO), but we do so by applying GPO caching in PowerShell. Changing the GPO by loading it onto your local session and using the *-GPOSession* parameter aren't supported in Netsh

PowerShell

```
$gpo = Open-NetGPO -PolicyStore domain.contoso.com\gpo_name
New-NetFirewallRule -DisplayName "Block Outbound Telnet" -Direction Outbound
-Program %SystemRoot%\System32\telnet.exe -Protocol TCP -LocalPort 23 -
Action Block -GPOSession $gpo
Save-NetGPO -GPOSession $gpo
```

This command doesn't batch your individual changes, it loads and saves the entire GPO at once. So if any other changes are made by other administrators, or in a different Windows PowerShell window, saving the GPO overwrites those changes.

Modify an existing firewall rule

When a rule is created, Netsh and Windows PowerShell allow you to change rule properties and influence, but the rule maintains its unique identifier (in Windows PowerShell, this identifier is specified with the *-Name* parameter). For example, you could have a rule **Allow Web 80** that enables TCP port 80 for inbound unsolicited traffic. You can change the rule to match a different remote IP address of a Web server whose traffic will be allowed by specifying the human-readable, localized name of the rule.

PowerShell

PowerShell

```
Set-NetFirewallRule -DisplayName "Allow Web 80" -RemoteAddress
192.168.0.2
```

Netsh requires you to provide the name of the rule for it to be changed and we don't have an alternate way of getting the firewall rule. In Windows PowerShell, you can query for the rule using its known properties. When you run `Get-NetFirewallRule`, you may notice that common conditions like addresses and ports don't appear. These conditions are represented in separate objects called Filters. As shown before, you can set all the conditions in New-NetFirewallRule and Set-NetFirewallRule. If you want to query for firewall rules based on these fields (ports, addresses, security, interfaces, services), you'll

need to get the filter objects themselves. You can change the remote endpoint of the **Allow Web 80** rule (as done previously) using filter objects. Using Windows PowerShell, you query by port using the port filter, then assuming other rules exist affecting the local port, you build with further queries until your desired rule is retrieved. In the following example, we assume the query returns a single firewall rule, which is then piped to the `Set-NetFirewallRule` cmdlet utilizing Windows PowerShell's ability to pipeline inputs.

PowerShell

```
Get-NetFirewallPortFilter | ?{$_.LocalPort -eq 80} | Get-NetFirewallRule | ?
{ $_.Direction -eq "Inbound" -and $_.Action -eq "Allow"} | Set-
NetFirewallRule -RemoteAddress 192.168.0.2
```

You can also query for rules using the wildcard character. The following example returns an array of firewall rules associated with a particular program. The elements of the array can be modified in subsequent `Set-NetFirewallRule` cmdlets.

PowerShell

```
Get-NetFirewallApplicationFilter -Program "*svchost*" | Get-NetFirewallRule
```

Multiple rules in a group can be simultaneously modified when the associated group name is specified in a Set command. You can add firewall rules to specified management groups in order to manage multiple rules that share the same influences. In the following example, we add both inbound and outbound Telnet firewall rules to the group **Telnet Management**. In Windows PowerShell, group membership is specified when the rules are first created so we re-create the previous example rules. Adding rules to a custom rule group isn't possible in Netsh.

PowerShell

```
New-NetFirewallRule -DisplayName "Allow Inbound Telnet" -Direction Inbound -
Program %SystemRoot%\System32\tlntsvr.exe -RemoteAddress LocalSubnet -Action
Allow -Group "Telnet Management"
New-NetFirewallRule -DisplayName "Block Outbound Telnet" -Direction Outbound
-Program %SystemRoot%\System32\tlntsvr.exe -RemoteAddress LocalSubnet -
Action Allow -Group "Telnet Management"
```

If the group isn't specified at rule creation time, the rule can be added to the rule group using dot notation in Windows PowerShell. You can't specify the group using `Set-NetFirewallRule` since the command allows querying by rule group.

PowerShell

```
$rule = Get-NetFirewallRule -DisplayName "Allow Inbound Telnet"
$rule.Group = "Telnet Management"
$rule | Set-NetFirewallRule
```

With the help of the `Set` command, if the rule group name is specified, the group membership isn't modified but rather all rules of the group receive the same modifications indicated by the given parameters. The following scriptlet enables all rules in a predefined group containing remote management influencing firewall rules.

PowerShell

PowerShell

```
Set-NetFirewallRule -DisplayGroup "Windows Firewall Remote Management" -
Enabled True
```

There's also a separate `Enable-NetFirewallRule` cmdlet for enabling rules by group or by other properties of the rule.

PowerShell

```
Enable-NetFirewallRule -DisplayGroup "Windows Firewall Remote Management" -
Verbose
```

Delete a firewall rule

Rule objects can be disabled so that they're no longer active. In Windows PowerShell, the **Disable-NetFirewallRule** cmdlet will leave the rule on the system, but put it in a disabled state so the rule no longer is applied and impacts traffic. A disabled firewall rule can be re-enabled by **Enable-NetFirewallRule**. This cmdlet is different from the **Remove-NetFirewallRule**, which permanently removes the rule definition from the device. The following cmdlet deletes the specified existing firewall rule from the local policy store.

PowerShell

PowerShell

```
Remove-NetFirewallRule -DisplayName "Allow Web 80"
```

Like with other cmdlets, you can also query for rules to be removed. Here, all blocking firewall rules are deleted from the device.

PowerShell

```
Remove-NetFirewallRule -Action Block
```

It may be safer to query the rules with the **Get** command and save it in a variable, observe the rules to be affected, then pipe them to the **Remove** command, just as we did for the **Set** commands. The following example shows how you can view all the blocking firewall rules, and then delete the first four rules.

PowerShell

```
$x = Get-NetFirewallRule -Action Block
$x
$x[0-3] | Remove-NetFirewallRule
```

Manage remotely

Remote management using WinRM is enabled by default. The cmdlets that support the *CimSession* parameter use WinRM and can be managed remotely by default. The following example returns all firewall rules of the persistent store on a device named **RemoteDevice**.

PowerShell

```
Get-NetFirewallRule -CimSession RemoteDevice
```

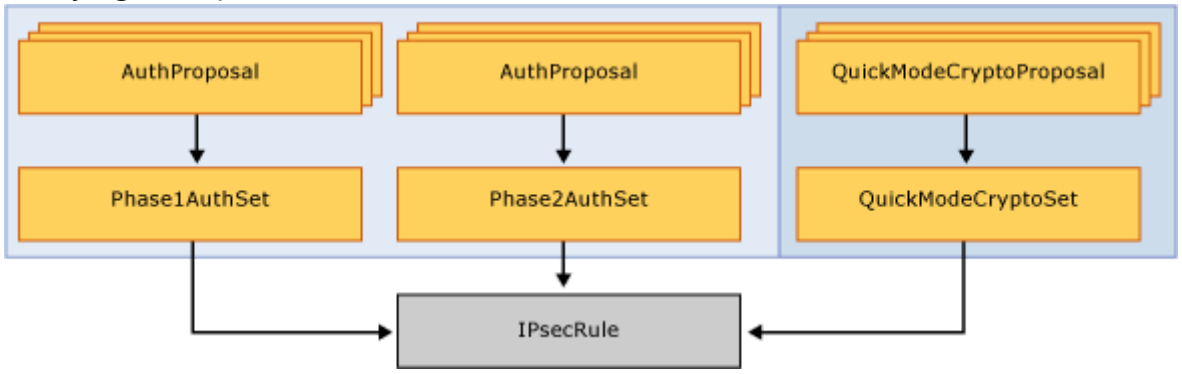
We can perform any modifications or view rules on remote devices by using the *-CimSession* parameter. Here we remove a specific firewall rule from a remote device.

PowerShell

```
$RemoteSession = New-CimSession -ComputerName RemoteDevice
Remove-NetFirewallRule -DisplayName "AllowWeb80" -CimSession $RemoteSession
-Confirm
```

Deploy basic IPsec rule settings

An Internet Protocol security (IPsec) policy consists of rules that determine IPsec behavior. IPsec supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. Windows PowerShell can create powerful, complex IPsec policies like in Netsh and the Windows Firewall console. However, because Windows PowerShell is object-based rather than string token-based, configuration in Windows PowerShell offers greater control and flexibility. In Netsh, the authentication and cryptographic sets were specified as a list of comma-separated tokens in a specific format. In Windows PowerShell, rather than using default settings, you first create your desired authentication or cryptographic proposal objects and bundle them into lists in your preferred order. Then, you create one or more IPsec rules that reference these sets. The benefit of this model is that programmatic access to the information in the rules is much easier. See the following sections for clarifying examples.



Create IPsec rules

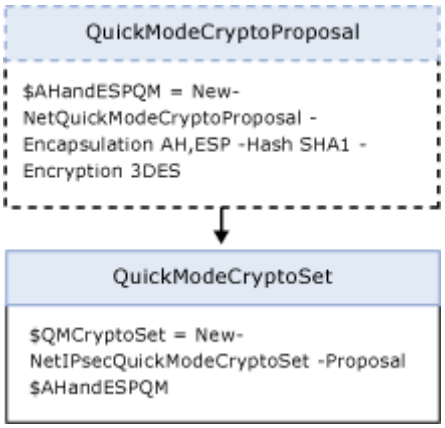
The following cmdlet creates basic IPsec transport mode rule in a Group Policy Object. An IPsec rule is simple to create; all that is required is the display name, and the remaining properties use default values. Inbound traffic is authenticated and integrity checked using the default quick mode and main mode settings. These default settings can be found in the console under Customize IPsec Defaults.

PowerShell

```
New-NetIPsecRule -DisplayName "Require Inbound Authentication" -
PolicyStore domain.contoso.com\gpo_name
```

Add custom authentication methods to an IPsec rule

If you want to create a custom set of quick-mode proposals that includes both AH and ESP in an IPsec rule object, you create the associated objects separately and link their associations. For more information about authentication methods, see Choosing the IPsec Protocol. You can then use the newly created custom quick-mode policies when you create IPsec rules. The cryptography set object is linked to an IPsec rule object.



In this example, we build on the previously created IPsec rule by specifying a custom quick-mode crypto set. The final IPsec rule requires outbound traffic to be authenticated by the specified cryptography method.

PowerShell

PowerShell

```
$AHandESPQM = New-NetIPsecQuickModeCryptoProposal -Encapsulation AH,ESP
-AHHash SHA1 -ESPHash SHA1 -Encryption DES3
$QMCryptoSet = New-NetIPsecQuickModeCryptoSet -DisplayName
"ah:sha1+esp:sha1-des3" -Proposal $AHandESPQM -PolicyStore
domain.contoso.com\gpo_name
New-NetIPsecRule -DisplayName "Require Inbound Authentication" -
InboundSecurity Require -OutboundSecurity Request -QuickModeCryptoSet
$QMCryptoSet.Name -PolicyStore domain.contoso.com\gpo_name
```

IKEv2 IPsec transport rules

A corporate network may need to secure communications with another agency. But, you discover the agency runs non-Windows operating systems and requires the use of the Internet Key Exchange Version 2 (IKEv2) standard. You can apply IKEv2 capabilities in Windows Server 2012 by specifying IKEv2 as the key module in an IPsec rule. This capability specification can only be done using computer certificate authentication and can't be used with phase-2 authentication.

PowerShell

```
New-NetIPsecRule -DisplayName "Require Inbound Authentication" -
InboundSecurity Require -OutboundSecurity Request -Phase1AuthSet
```

```
MyCertAuthSet -KeyModule IKEv2 -RemoteAddress $nonWindowsGateway
```

For more info about IKEv2, including scenarios, see Securing End-to-End IPsec Connections by Using IKEv2.

Copy an IPsec rule from one policy to another

Firewall and IPsec rules with the same rule properties can be duplicated to simplify the task of re-creating them within different policy stores. To copy the previously created rule from one policy store to another, the associated objects must also be copied separately. There's no need to copy associated firewall filters. You can query rules to be copied in the same way as other cmdlets. Copying individual rules is a task that isn't possible through the Netsh interface. Here's how you can accomplish it with Windows PowerShell.

PowerShell

```
$Rule = Get-NetIPsecRule -DisplayName "Require Inbound Authentication"
$Rule | Copy-NetIPsecRule -NewPolicyStore domain.costoso.com\new_gpo_name
$Rule | Copy-NetPhase1AuthSet -NewPolicyStore
domain.costoso.com\new_gpo_name
```

Handling Windows PowerShell errors

To handle errors in your Windows PowerShell scripts, you can use the *-ErrorAction* parameter. This parameter is especially useful with the **Remove** cmdlets. If you want to remove a particular rule, you'll notice that it fails if the rule isn't found. When rules are being removed, if the rule isn't already there, it's acceptable to ignore that error. In this case, you can do the following to suppress any "rule not found" errors during the remove operation.

PowerShell

```
Remove-NetFirewallRule -DisplayName "Contoso Messenger 98" -ErrorAction
SilentlyContinue
```

The use of wildcards can also suppress errors, but they could potentially match rules that you didn't intend to remove. These wildcards can be a useful shortcut, but should only be used if you know there aren't any extra rules that will be accidentally deleted. So the following cmdlet will also remove the rule, suppressing any "not found" errors.

PowerShell

```
Remove-NetFirewallRule -DisplayName "Contoso Messenger 98*"
```

When using wildcards, if you want to double-check the set of rules that is matched, you can use the *-WhatIf* parameter.

PowerShell

```
Remove-NetFirewallRule -DisplayName "Contoso Messenger 98*" -WhatIf
```

If you only want to delete some of the matched rules, you can use the *-Confirm* parameter to get a rule-by-rule confirmation prompt.

PowerShell

```
Remove-NetFirewallRule -DisplayName "Contoso Messenger 98*" -Confirm
```

You can also just perform the whole operation, displaying the name of each rule as the operation is performed.

PowerShell

```
Remove-NetFirewallRule -DisplayName "Contoso Messenger 98*" -Verbose
```

Monitor

The following Windows PowerShell commands are useful in the update cycle of a deployment phase. To allow you to view all the IPsec rules in a particular store, you can use the following commands. In Netsh, this command doesn't show rules where profile=domain,public or profile=domain,private. It only shows rules that have the single entry domain that is included in the rule. The following command examples will show the IPsec rules in all profiles.

PowerShell

PowerShell

```
Show-NetIPsecRule -PolicyStore ActiveStore
```

You can monitor main mode security associations for information such as which peers are currently connected to the device and which protection suite is used to form the

security associations. Use the following cmdlet to view existing main mode rules and their security associations:

PowerShell

```
Get-NetIPsecMainModeSA
```

Find the source GPO of a rule

To view the properties of a particular rule or group of rules, you query for the rule. When a query returns fields that are specified as **NotConfigured**, you can determine which policy store a rule originates from. For objects that come from a GPO (the *-PolicyStoreSourceType* parameter is specified as **GroupPolicy** in the **Show** command), if *-TracePolicyStore* is passed, the name of the GPO is found and returned in the **PolicyStoreSource** field.

PowerShell

```
Get-NetIPsecRule -DisplayName "Require Inbound Authentication" -
TracePolicyStore
```

It's important to note that the revealed sources don't contain a domain name.

Deploy a basic domain isolation policy

IPsec can be used to isolate domain members from non-domain members. Domain isolation uses IPsec authentication to require that the domain-joined devices positively establish the identities of the communicating devices to improve security of an organization. One or more features of IPsec can be used to secure traffic with an IPsec rule object. To implement domain isolation on your network, the devices in the domain receive IPsec rules that block unsolicited inbound network traffic that isn't protected by IPsec. Here we create an IPsec rule that requires authentication by domain members. Through this authentication, you can isolate domain-joined devices from devices that aren't joined to a domain. In the following examples, Kerberos authentication is required for inbound traffic and requested for outbound traffic.

PowerShell

PowerShell

```
$kerbprop = New-NetIPsecAuthProposal -Machine -Kerberos
$Phase1AuthSet = New-NetIPsecPhase1AuthSet -DisplayName "Kerberos Auth
Phase1" -Proposal $kerbprop -PolicyStore
domain.contoso.com\domain_isolation
New-NetIPsecRule -DisplayName "Basic Domain Isolation Policy" -Profile
Domain -Phase1AuthSet $Phase1AuthSet.Name -InboundSecurity Require -
OutboundSecurity Request -PolicyStore
domain.contoso.com\domain_isolation
```

Configure IPsec tunnel mode

The following command creates an IPsec tunnel that routes traffic from a private network (192.168.0.0/16) through an interface on the local device (1.1.1.1) attached to a public network to a second device through its public interface (2.2.2.2) to another private network (192.157.0.0/16). All traffic through the tunnel is checked for integrity by using ESP/SHA1, and it's encrypted by using ESP/DES3.



PowerShell

PowerShell

```
$QMProposal = New-NetIPsecQuickModeCryptoProposal -Encapsulation ESP -
ESPHash SHA1 -Encryption DES3
$QMCryptoSet = New-NetIPsecQuickModeCryptoSet -DisplayName "esp:sha1-
des3" -Proposal $QMProposal
New-NetIPSecRule -DisplayName "Tunnel from HQ to Dallas Branch" -Mode
Tunnel -LocalAddress 192.168.0.0/16 -RemoteAddress 192.157.0.0/16 -
LocalTunnelEndpoint 1.1.1.1 -RemoteTunnelEndpoint 2.2.2.2 -
InboundSecurity Require -OutboundSecurity Require -QuickModeCryptoSet
$QMCryptoSet.Name
```

Deploy secure firewall rules with IPsec

In situations where only secure traffic can be allowed through the Windows Firewall, a combination of manually configured firewall and IPsec rules are necessary. The firewall rules determine the level of security for allowed packets, and the underlying IPsec rules secure the traffic. The scenarios can be accomplished in Windows PowerShell and in Netsh, with many similarities in deployment.

Create a secure firewall rule (allow if secure)

Configuring firewalls rule to allow connections if they're secure requires the corresponding traffic to be authenticated and integrity protected, and then optionally encrypted by IPsec. The following example creates a firewall rule that requires traffic to be authenticated. The command permits inbound Telnet network traffic only if the connection from the remote device is authenticated by using a separate IPsec rule.

PowerShell

```
PowerShell

New-NetFirewallRule -DisplayName "Allow Authenticated Telnet" -Direction
Inbound -Program %SystemRoot%\System32\tlntsvr.exe -Authentication
Required -Action Allow
```

The following command creates an IPsec rule that requires a first (computer) authentication and then attempts an optional second (user) authentication. Creating this rule secures and allows the traffic through the firewall rule requirements for the messenger program.

PowerShell

```
PowerShell

$mkerbauthprop = New-NetIPsecAuthProposal -Machine -Kerberos
$mntlmauthprop = New-NetIPsecAuthProposal -Machine -NTLM
$P1Auth = New-NetIPsecPhase1AuthSet -DisplayName "Machine Auth" -
Proposal $mkerbauthprop,$mntlmauthprop
$ukerbauthprop = New-NetIPsecAuthProposal -User -Kerberos
$unentlmauthprop = New-NetIPsecAuthProposal -User -NTLM
$anonyauthprop = New-NetIPsecAuthProposal -Anonymous
$P2Auth = New-NetIPsecPhase2AuthSet -DisplayName "User Auth" -Proposal
$ukerbauthprop,$unentlmauthprop,$anonyauthprop
New-NetIPSecRule -DisplayName "Authenticate Both Computer and User" -
InboundSecurity Require -OutboundSecurity Require -Phase1AuthSet
$P1Auth.Name -Phase2AuthSet $P2Auth.Name
```

Isolate a server by requiring encryption and group membership

To improve the security of the devices in an organization, you can deploy domain isolation in which domain-members are restricted. They require authentication when communicating among each other and reject non-authenticated inbound connections.

To improve the security of servers with sensitive data, this data must be protected by allowing access only to a subset of devices within the enterprise domain. IPsec can provide this extra layer of protection by isolating the server. In server isolation, sensitive data access is restricted to users and devices with legitimate business need, and the data is additionally encrypted to prevent eavesdropping.

Create a firewall rule that requires group membership and encryption

To deploy server isolation, we layer a firewall rule that restricts traffic to authorized users or devices on the IPsec rule that enforces authentication. The following firewall rule allows Telnet traffic from user accounts that are members of a custom group called "Authorized to Access Server." This access can additionally be restricted based on the device, user, or both by specifying the restriction parameters. A Security Descriptor Definition Language (SDDL) string is created by extending a user or group's security identifier (SID). For more information about finding a group's SID, see: Finding the SID for a group account. Restricting access to a group allows administrations to extend strong authentication support through Windows Firewall and/or IPsec policies. The following example shows you how to create an SDDL string that represents security groups.

PowerShell

```
$user = new-object System.Security.Principal.NTAccount
("corp.contoso.com\Administrators")
$SIDofSecureUserGroup =
$user.Translate([System.Security.Principal.SecurityIdentifier]).Value
$secureUserGroup = "D:(A;;CC;;;$SIDofSecureUserGroup)"
```

By using the previous scriptlet, you can also get the SDDL string for a secure computer group as shown here:

PowerShell

```
$secureMachineGroup = "D:(A;;CC;;;$SIDofSecureMachineGroup)"
```

For more information about how to create security groups or how to determine the SDDL string, see Working with SIDs. Telnet is an application that doesn't provide encryption. This application can send data, such as names and passwords, over the network. This data can be intercepted by malicious users. If an administrator would like to allow the use of Telnet, but protect the traffic, a firewall rule that requires IPsec encryption can be created. This firewall rule is necessary so that the administrator can be

certain that when this application is used, all of the traffic sent or received by this port is encrypted. If IPsec fails to authorize the connection, no traffic is allowed from this application. In this example, we allow only authenticated and encrypted inbound Telnet traffic from a specified secure user group through the creation of the following firewall rule.

PowerShell

PowerShell

```
New-NetFirewallRule -DisplayName "Allow Encrypted Inbound Telnet to
Group Members Only" -Program %SystemRoot%\System32\tlntsvr.exe -Protocol
TCP -Direction Inbound -Action Allow -LocalPort 23 -Authentication
Required -Encryption Required -RemoteUser $secureUserGroup -PolicyStore
domain.contoso.com\Server_Isolation
```

Endpoint security enforcement

The previous example showed end to end security for a particular application. In situations where endpoint security is required for many applications, having a firewall rule per application can be cumbersome and difficult to manage. Authorization can override the per-rule basis and be done at the IPsec layer. In this example, we set the global IPsec setting to only allow transport mode traffic to come from an authorized user group with the following cmdlet. Consult the previous examples for working with security groups.

PowerShell

```
Set-NetFirewallSetting -RemoteMachineTransportAuthorizationList
$secureMachineGroup
```

Create firewall rules that allow IPsec-protected network traffic (authenticated bypass)

Authenticated bypass allows traffic from a specified trusted device or user to override firewall block rules. This override is helpful when an administrator wants to use scanning servers to monitor and update devices without the need to use port-level exceptions. For more information, see How to enable authenticated firewall bypass. In this example, we assume that a blocking firewall rule exists. This example permits any network traffic on any port from any IP address to override the block rule, if the traffic is authenticated

as originating from a device or user account that is a member of the specified device or user security group.

PowerShell

PowerShell

```
New-NetFirewallRule -DisplayName "Inbound Secure Bypass Rule" -Direction
Inbound -Authentication Required -OverrideBlockRules $true -
RemoteMachine $secureMachineGroup -RemoteUser $secureUserGroup -
PolicyStore domain.contoso.com\domain_isolation
```

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Firewall dynamic keywords

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **ⓘ Important**
>
> This article describes features or settings that are in preview. The content is subject to change and may have dependencies on other features or services in preview.

Windows Firewall includes a functionality called *dynamic keywords*, which simplifies the configuration and management of Windows Firewall.

With dynamic keywords, you can define a set of IP address ranges, fully qualified domain names (FQDNs), and **autoresolution** options, to which one or more Firewall rules can refer.

Configure dynamic keywords

To configure dynamic keywords, you can use:

- Firewall CSP, which can be used with a Mobile Device Management (MDM) solution like Microsoft Intune
- Windows PowerShell

> **Tip**
>
> Microsoft Intune offers a simplified management experience called *reusable settings groups*. For more information, see **Add reusable settings groups to profiles for Firewall rules**.

This article describes how to configure dynamic keywords using Windows PowerShell.

Dynamic keywords and Fully Qualified Domain Names (FQDN)

Dynamic keywords can be configured by defining a set of IP address ranges or FQDNs. Here are important things to consider when using FQDNs:

- FQDN support is for reducing the overhead of managing IP rules where IP addresses are dynamic and change frequently
- FQDNs aren't a replacement for IP addresses in all scenarios. IP addresses should be used when possible, for security and performance reasons
  - FQDN rules can affect performance on the endpoint, caused by DNS latency and other factors
  - FQDN isn't a secure DNS service. The FQDN resolution uses the default DNS configuration of the endpoint
- An FQDN rule requires a DNS query to happen for that FQDN to be resolved to an IP address. Traffic to IP addresses must generate a DNS query for FQDN rules
  - Limitations include: websites accessed via proxy, secure DNS services, certain VPN tunnel configurations, cached IPs on the endpoint
- While Partially Qualified Domain Names (PQDNs) are allowed, FQDNs are preferred. Wildcards `*` are supported for hosts, for example `*.contoso.com`

Two examples of FQDN rules are:

- Block all outbound and inbound by default and allow specific outbound traffic
- Block all inbound by default and block some specific outbound traffic

> ⓘ **Note**
>
> Inbound FQDN rules aren't natively supported. However, it's possible to use *pre-hydration* scripts to generate inbound IP entries for the rules.

> ⊗ **Caution**
>
> The default configuration of *Blocked for Outbound* rules can be considered for certain highly secure environments. However, the *Inbound* rule configuration should never be changed in a way that allows traffic by default.

In high security environments, an inventory of all apps should be maintained. Records should include whether an app requires network connectivity. Administrators should create new rules specific to each app that needs network connectivity, and push those rules centrally, using a device management solution.

Functions and known limitations

The Windows Firewall FQDN feature uses the Network Protection external callout driver, to inspect DNS responses where the DNS query matches FQDN rules. Some important

functions and limitations of the feature are:

- The Network Protection component doesn't periodically execute DNS queries. It requires an application to execute a DNS query
- Windows Firewall flushes all stored resolved IP addresses on device restart
- Network protection doesn't synchronously inspect the DNS response, as it doesn't hold the UDP packet during inspection. The result is a potential condition where an application, after receiving the DNS response, attempts to connect, but gets blocked if it's faster than the firewall rule update
  - Generally, applications have retry logic for an initial failed connection and as a result the issue is transparent to the end user
  - On occasion a component might not have retry logic on initial connection fail. Which is solved in two ways:
    - The user can hit *refresh* in the application they're using, and it should connect successfully
    - Administrators can use the *prehydration* scripts tactfully, where this condition is occurring in their environment

FQDN Feature requirements

The following are requirements for the FQDN feature:

- Microsoft Defender Antivirus must be turned on and running platform version `4.18.2209.7` or later.
  - To verify, open Windows Security and select **Settings** > **About**
- Network Protection must be in *block* or *audit* mode. For more information, see Check if network protection is enabled.
- DNS over HTTPS (DoH) must be disabled. To configure your preferred browser, you can use the following settings:
  - Microsoft Edge
  - Chrome
  - Firefox
- The device's default DNS resolution settings apply. This feature doesn't provide DNS security or functionality changes

> **Tip**
>
> You can also download the ADMX file from there, follow the directions, and configure it via gpedit.msc for local testing.

Manage dynamic keywords with Windows PowerShell

This section provides some examples how to manage dynamic keywords using Windows PowerShell. A few important things to consider when using dynamic keywords are:

- All dynamic keyword objects must have a unique identifier (GUID) to represent them
- A firewall rule can use dynamic keywords instead of explicitly defining IP addresses for its conditions
- A firewall rule can use both dynamic keywords and statically defined address ranges
- A dynamic keyword object can be reused across multiple firewall rules
- If a firewall rule doesn't have any configured remote addresses, then the rule isn't enforced. For example, if a rule is configured with only `AutoResolve` objects that aren't yet resolved
- If a rule uses multiple dynamic keywords, then the rule is enforced for all addresses that are *currently* resolved. The rule is enforced even if there are unresolved objects. When a dynamic keyword address is updated, all associated rule objects have their remote addresses updated
- Windows doesn't enforce any dependencies between a rule and a dynamic keyword address, and either object can be created first. A rule can reference dynamic keyword IDs that don't yet exist, in which case the rule isn't enforced
- You can delete a dynamic keyword address, even if it's in use by a firewall rule

Allow Outbound

Here's an example script to allow an FQDN from PowerShell. Replace the `$fqdn` variable value with the FQDN you wish to block (line #1):

PowerShell

```
$fqdn = 'contoso.com'
$id = '{' + (new-guid).ToString() + '}'
New-NetFirewallDynamicKeywordAddress -id $id -Keyword $fqdn -AutoResolve
$true
New-NetFirewallRule -DisplayName "allow $fqdn" -Action Allow -Direction
Outbound -RemoteDynamicKeywordAddresses $id
```

Dynamic keyword addresses can be created with the `AutoResolve` parameter set to `$true` or `$false`. If `AutoResolve` is set to `$true`, then Windows attempts to resolve the keyword to an IP address.

Block Outbound

Here's an example script to block an FQDN from PowerShell. Replace the `$fqdn` variable value with the FQDN you wish to block (line #1):

PowerShell

```
$fqdn = 'contoso.com'
$id = '{' + (new-guid).ToString() + '}'
New-NetFirewallDynamicKeywordAddress -id $id -Keyword $fqdn -AutoResolve
$true
New-NetFirewallRule -DisplayName "block $fqdn" -Action Block -Direction
Outbound -RemoteDynamicKeywordAddresses $id
```

Display Auto resolve rules and associated resolved IP addresses

This example shows how to display all dynamic keyword addresses that have the `AutoResolve` parameter set to `$true` and the associated resolved IP addresses.

PowerShell

```
Get-NetFirewallDynamicKeywordAddress -AllAutoResolve
```

> **Note**
>
> IP addresses will not populate until DNS query is observed.

Hydrate FQDN rules

The following sample scripts read the current Windows Firewall configuration, extract FQDN-based rules, and perform DNS resolution on each domain. The result is that the IP addresses for those rules get "prehydrated."

PowerShell

```
Get-NetFirewallDynamicKeywordAddress -AllAutoResolve |`
ForEach-Object {
  if(!$_.Keyword.Contains("*")) {
    Write-Host "Getting" $_.Keyword
    resolve-dnsname -Name $_.Keyword -DNSOnly | out-null
```

```
  }
}
```

A similar script can be used to perform DNS resolution using `nslookup.exe`:

PowerShell

```
Get-NetFirewallDynamicKeywordAddress -AllAutoResolve |`
ForEach-Object {
  if(!$_.Keyword.Contains("*")) {
    Write-Host "Getting" $_.Keyword
    nslookup $_.Keyword
  }
}
```

If using `nslookup.exe`, you must create an outbound firewall rule when using the *block all outbound* posture. Here's the command to create the outbound rule for `nslookup.exe`:

PowerShell

```
$appName = 'nslookup'
$appPath = 'C:\Windows\System32\nslookup.exe'
New-NetFirewallRule -DisplayName "allow $appName" -Program $appPath -Action
Allow -Direction Outbound -Protocol UDP -RemotePort 53
```

Block all outbound and allow some FQDNs

In the next example, a list of applications is parsed for FQDN evaluation. The FQDNs listed in the scripts were observed when inspecting traffic on the first launch of Microsoft Edge.

> **ⓘ Important**
>
> This is not a complete list nor a recommendation. It's an example of how an application should be evaluated to ensure proper connectivity and function.

To learn more about Microsoft Edge requirements for Internet connectivity, see allowlist for Microsoft Edge endpoints.

PowerShell

```
$domains = @(
    '*.microsoft.com',
```

```
    '*.msftconnecttest.com',
    'assets.msn.com',
    'client.wns.windows.com',
    'config.edge.skype.com',
    'ctldl.windowsupdate.com',
    'dns.msftncsi.com',
    'login.live.com',
    'ntp.msn.com'
)

foreach ($domain in $domains) {
    $id = '{' + (New-Guid).ToString() + '}'
    New-NetFirewallDynamicKeywordAddress -Id $id -Keyword $domain -
AutoResolve $true
    New-NetFirewallRule -DisplayName "allow $domain" -Action Allow -
Direction Outbound -RemoteDynamicKeywordAddresses $id
}
```

For more information about the PowerShell cmdlets used to manage dynamic keywords, see:

- Get-NetFirewallDynamicKeywordAddress
- New-NetFirewallDynamicKeywordAddress
- Remove-NetFirewallDynamicKeywordAddress
- Update-NetFirewallDynamicKeywordAddress

For information about the API structure, see Firewall dynamic keywords.

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure Hyper-V firewall

Article • 04/07/2025 • Applies to: ✅ Windows 11

Starting in Windows 11, version 22H2, Hyper-V firewall is a network firewall solution that enables filtering of inbound and outbound traffic to/from containers hosted by Windows, including the Windows Subsystem for Linux (WSL).
This article describes how to configure Hyper-V firewall rules and settings using PowerShell or configuration service provider (CSP).

> **ⓘ Important**
>
> The configuration of Hyper-V firewall is not available via group policy (GPO). If Windows Firewall settings are configured via GPO and Hyper-V firewall settings aren't configured via CSP, then the applicable rules and settings are automatically mirrored from the GPO configuration.

Configure Hyper-V firewall with PowerShell

This section describes the steps to manage Hyper-V firewall using PowerShell.

Obtain the WSL GUID

Hyper-V firewall rules are enabled per *VMCreatorId*. To obtain the VMCreatorId, use the cmdlet:

```PowerShell
Get-NetFirewallHyperVVMCreator
```

The output contains a VmCreator object type, which has unique identifier `VMCreatorId` and `friendly name` properties. For example, the following output shows the properties of WSL:

```PowerShell
PS C:\> Get-NetFirewallHyperVVMCreator
VMCreatorId : {40E0AC32-46A5-438A-A0B2-2B479E8F2E90}
FriendlyName : WSL
```

> **Note**
>
> The WSL VMCreatorId is `{40E0AC32-46A5-438A-A0B2-2B479E8F2E90}`.

Verify Hyper-V firewall settings

Hyper-V firewall has settings that apply in general to a VMCreatorId. Use the Get-NetFirewallHyperVVMSetting cmdlet to check the settings. For example, you can obtain the policies applied to WSL with the command:

PowerShell

```
Get-NetFirewallHyperVVMSetting -PolicyStore ActiveStore -Name '{40E0AC32-46A5-438A-A0B2-2B479E8F2E90}'
```

> **Note**
>
> `-PolicyStore ActiveStore` returns the *applied* settings.

The output contains the following values:

Expand table

Value	Description
`Enabled` (True/False)	True if Hyper-V Firewall is enabled for WSL VMs.
`DefaultInboundAction`, `DefaultOutboundAction`	These are default rule policies applied to packets entering or leaving the WSL container. The rule policies can be modified, as described in this article.
`LoopbackEnabled`	Tracks if loopback traffic between the host and the container is allowed, without requiring any Hyper-V Firewall rules. WSL enables it by default, to allow the Windows Host to talk to WSL, and WSL to talk to the Windows Host.
`AllowHostPolicyMerge`	Determines how Windows Host Firewall Enterprise Settings (GPO), Hyper-V Firewall Enterprise Settings (CSP), Windows Host Firewall Enterprise Settings (CSP), local Hyper-V Firewall settings, and local Host Firewall settings interact. This setting is detailed with the Set-NetFirewallHyperVVMSetting cmdlet.

Configure Hyper-V firewall settings

To configure Hyper-V firewall, use the Set-NetFirewallHyperVVMSetting command. For example, the following command sets the default inbound connection to *Allow*:

```PowerShell
Set-NetFirewallHyperVVMSetting -Name '{40E0AC32-46A5-438A-A0B2-2B479E8F2E90}' -DefaultInboundAction Allow
```

Firewall Rules

Hyper-V firewall rules can be enumerated and created from PowerShell. To view rules, use the Get-NetFirewallHyperVRule cmdlet. For example, to view firewall rules that only pertain to WSL, use the following command:

```PowerShell
Get-NetFirewallHyperVRule -VMCreatorId '{40E0AC32-46A5-438A-A0B2-2B479E8F2E90}'
```

To configure specific rules, use the Set-NetFirewallHyperVRule cmdlet.

For example, to create an inbound rule to allow TCP traffic to WSL on port 80, use the following command:

```PowerShell
New-NetFirewallHyperVRule -Name MyWebServer -DisplayName "My Web Server" -Direction Inbound -VMCreatorId '{40E0AC32-46A5-438A-A0B2-2B479E8F2E90}' -Protocol TCP -LocalPorts 80
```

Target Hyper-V firewall rules and settings to specific profiles

Hyper-V firewall rules and settings can be targeted to the *Firewall profiles*, which are based on the type of network the device is connected to:

- Public profile
- Private profile
- Domain profile

The policy options are similar to the ones already described, but are applied to specific profiles for the connected Windows Host network adapter.

To view the settings per profile, use the following command:

PowerShell

```
Get-NetFirewallHyperVProfile -PolicyStore ActiveStore
```

> **Note**
>
> `-PolicyStore ActiveStore` returns the *applied* settings.

The output contains an extra value compared to the ones described in the previous section:

Expand table

Value	Description
`AllowLocalFirewallRules` (True/False)	This setting determines how enterprise Hyper-V firewall rules (CSP or GPO) interact with the locally defined Hyper-V firewall rules: - if the value is *True*, both the enterprise Hyper-V firewall rules and the locally defined rules are applied - if the value is *False*, the locally defined Hyper-V firewall rules aren't applied, and only enterprise rules are applied.

> **Note**
>
> To configure these **settings** per profile, use the Set-NetFirewallHyperVProfile cmdlet.
>
> To configure these **rules** per profile using the Set-NetFirewallHyperVRule cmdlet with the `-Profile` option.

Configure Hyper-V firewall with CSP

You can configure Hyper-V firewall using the Firewall CSP, for example with an MDM solution like Microsoft Intune.

To learn more about the CSP options, follow these links:

- Configure Hyper-V firewall settings: to configure the Hyper-V firewall settings
- Configure Hyper-V firewall rules: to configure list of rules controlling traffic through the Hyper-V firewall

To learn how to configure the firewall with Microsoft Intune, see Firewall policy for endpoint security.

Provide feedback

To provide feedback for Hyper-V firewall, open **Feedback Hub** and use the category **Security and Privacy > Microsoft Defender Firewall and network protection**.

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure Windows Firewall logging

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

To configure Windows Firewall to log dropped packets or successful connections, you can use:

- Configuration Service Provider (CSP), using an MDM solution like Microsoft Intune
- Group policy (GPO)

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

1. Sign into the Microsoft Intune admin center
2. Go to **Endpoint security** > **Firewall** > **Create policy** > **Windows 10, Windows 11, and Windows Server** > **Windows Firewall** > **Create**
3. Enter a name and, optionally, a description > **Next**
4. Under **Configuration settings**, for each network location type (*Domain*, *Private*, *Public*), configure:
   - **Log file path**
   - **Enable log dropped packets**
   - **Enable log success connections**
   - **Log max file size**
5. Select **Next** > **Next**
6. Assign the policy to a group that contains as members the devices or users that you want to configure > **Next** > **Create**

> **Tip**
>
> If you prefer you can also use a Settings catalog policy to configure Windows Firewall logging.

Alternatively, you can configure devices using a custom policy with the Firewall CSP.

Expand table

Network profile	Setting
Domain	Setting name: EnableLogDroppedPackets OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/DomainProfile/EnableLogDroppedPackets`
Domain	Setting name: LogFilePath OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/DomainProfile/LogFilePath`
Domain	Setting name: EnableLogSuccessConnections OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/DomainProfile/EnableLogSuccessConnections`
Domain	Setting name: LogMaxFileSize OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/DomainProfile/LogMaxFileSize`
Private	Setting name: EnableLogDroppedPackets OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PrivateProfile/EnableLogDroppedPackets`
Private	Setting name: LogFilePath OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PrivateProfile/LogFilePath`
Private	Setting name: EnableLogSuccessConnections OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PrivateProfile/EnableLogSuccessConnections`
Private	Setting name: LogMaxFileSize OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PrivateProfile/LogMaxFileSize`
Public	Setting name: EnableLogDroppedPackets OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PublicProfile/EnableLogDroppedPackets`
Public	Setting name: LogFilePath OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PublicProfile/LogFilePath`
Public	Setting name: EnableLogSuccessConnections OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PublicProfile/EnableLogSuccessConnections`
Public	Setting name: LogMaxFileSize OMA-URI: `./Vendor/MSFT/Firewall/MdmStore/PublicProfile/LogMaxFileSize`

ⓘ Important

The location you specify must have permissions assigned that permit the Windows Firewall service to write to the log file.

Recommendations

Here are some recommendations for configuring Windows Firewall logging:

- Change the logging size to at least **20,480 KB (20 MB)** to ensure that the log file doesn't fill up too quickly. The maximum log size is 32,767 KB (32 MB)
- For each profile (Domain, Private, and Public) change the default log file name from `%windir%\system32\logfiles\firewall\pfirewall.log` to:
  - `%windir%\system32\logfiles\firewall\pfirewall_Domain.log`
  - `%windir%\system32\logfiles\firewall\pfirewall_Private.log`
  - `%windir%\system32\logfiles\firewall\pfirewall_Public.log`
- Log dropped packets to **Yes**
- Log successful connections to **Yes**

On a single system, you can use the following commands to configure logging:

Windows Command Prompt

```
netsh advfirewall>set allprofiles logging allowedconnections enable
netsh advfirewall>set allprofiles logging droppedconnections enable
```

Parsing methods

There are several methods to parse the Windows Firewall log files. For example:

- Enable *Windows Event Forwarding* (WEF) to a *Windows Event Collector* (WEC). To learn more, see Use Windows Event Forwarding to help with intrusion detection
- Forward the logs to your SIEM product such as our Azure Sentinel. To learn more, see Windows Firewall connector for Microsoft Sentinel
- Forward the logs to Azure Monitor and use KQL to parse the data. To learn more, see Azure Monitor agent on Windows client devices

Tip

If logs are slow to appear in your SIEM solution, you can decrease the log file size. Just beware that the downsizing results in more resource usage due to the increased log rotation.

Troubleshoot if the log file is not created or modified

Sometimes the Windows Firewall log files aren't created, or the events aren't written to the log files. Some examples when this condition might occur include:

- Missing permissions for the *Windows Defender Firewall Service* (`mpssvc`) on the folder or on the log files
- You want to store the log files in a different folder and the permissions are missing, or aren't set automatically
- if firewall logging is configured via policy settings, it can happen that
  - the log folder in the default location `%windir%\System32\LogFiles\firewall` doesn't exist
  - the log folder in a custom path doesn't exist

In both cases, you must create the folder manually or via script, and add the permissions for `mpssvc`.

PowerShell

```
New-Item -ItemType Directory -Path $env:windir\System32\LogFiles\Firewall
```

Verify if `mpssvc` has *FullControl* on the folder and the files. From an elevated PowerShell session, use the following commands, ensuring to use the correct path:

PowerShell

```
$LogPath = Join-Path -path $env:windir -ChildPath
"System32\LogFiles\Firewall"
(Get-ACL -Path $LogPath).Access | Format-Table
IdentityReference,FileSystemRights,AccessControlType,IsInherited,Inheritance
Flags -AutoSize
```

The output should show `NT SERVICE\mpssvc` having *FullControl*:

PowerShell

```
IdentityReference      FileSystemRights AccessControlType IsInherited
InheritanceFlags
-----------------      ---------------- ----------------- ----------- ------
----------
NT AUTHORITY\SYSTEM         FullControl             Allow       False
ObjectInherit
BUILTIN\Administrators      FullControl             Allow       False
ObjectInherit
```

```
NT SERVICE\mpssvc           FullControl             Allow       False
ObjectInherit
```

If not, add *FullControl* permissions for `mpssvc` to the folder, subfolders and files. Make sure to use the correct path.

PowerShell

```
$LogPath = Join-Path -path $env:windir -ChildPath
"System32\LogFiles\Firewall"
$NewAcl = Get-Acl -Path $LogPath

$identity = "NT SERVICE\mpssvc"
$fileSystemRights = "FullControl"
$inheritanceFlags = "ContainerInherit,ObjectInherit"
$propagationFlags = "None"
$type = "Allow"

$fileSystemAccessRuleArgumentList = $identity, $fileSystemRights,
$inheritanceFlags, $propagationFlags, $type
$fileSystemAccessRule = New-Object -TypeName
System.Security.AccessControl.FileSystemAccessRule -ArgumentList
$fileSystemAccessRuleArgumentList

$NewAcl.SetAccessRule($fileSystemAccessRule)
Set-Acl -Path $LogPath -AclObject $NewAcl
```

Restart the device to restart the *Windows Defender Firewall* service.

Feedback

Was this page helpful? Yes No

Provide product feedback

Troubleshooting UWP App Connectivity Issues

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This document is intended to help network admins, support engineers, and developers to investigate UWP app network connectivity issues.

This document guides you through steps to debug Universal Windows Platform (UWP) app network connectivity issues by providing practical examples.

Typical causes of connectivity issues

UWP app network connectivity issues are typically caused by:

1. The UWP applications not being permitted to receive loopback traffic. This permission must be configured. By default, UWP applications aren't allowed to receive loopback traffic
2. The UWP app is missing the proper capability tokens
3. The private range is configured incorrectly. For example, the private range is set incorrectly through GP/MDM policies, etc.

To understand these causes more thoroughly, there are several concepts to review.

The traffic of network packets (what's permitted and what's not) on Windows is determined by the Windows Filtering Platform (WFP). When a UWP app or the private range is configured incorrectly, it affects how the UWP app's network traffic will be processed by WFP.

When a packet is processed by WFP, the characteristics of that packet must explicitly match all the conditions of a filter to either be permitted or dropped to its target address. Connectivity issues typically happen when the packet doesn't match any of the filter conditions, leading the packet to be dropped by a default block filter. The presence of the default block filters ensures network isolation for UWP applications. Specifically, it guarantees a network drop for a packet that doesn't have the correct capabilities for the resource it's trying to reach. Such a packet drop ensures the application's granular access to each resource type and preventing the application from escaping its environment.

For more information on the filter arbitration algorithm and network isolation, see Filter Arbitration and Isolation.

The following sections cover debugging case examples for loopback and non-loopback UWP app network connectivity issues.

> **Note**
>
> As improvements to debugging and diagnostics in the Windows Filtering Platform are made, the trace examples in this document may not exactly match the traces collected on previous releases of Windows.

Debugging UWP App Loopback scenarios

If you need to establish a TCP/IP connection between two processes on the same host where one of them is a UWP app, you must enable loopback.

To enable loopback for client outbound connections, run the following command at a command prompt:

Windows Command Prompt

```
CheckNetIsolation.exe LoopbackExempt -a -n=<AppContainer or Package Family>
```

To enable loopback for server inbound connections, run the following command at a command prompt:

Windows Command Prompt

```
CheckNetIsolation.exe LoopbackExempt -is -n=<AppContainer or Package Family>
```

You can ensure loopback is enabled by checking the appx manifests of both the sender and receiver.

For more information about loopback scenarios, see Communicating with localhost (loopback).

> **Note**

If you are in the middle of developing a UWA application and want to test its loopback, ensure to uninstall and re-install the UWA app if the network capabilities change for whatever reason.

Also, see How to enable loopback and troubleshoot network isolation (Windows Runtime apps).

Debugging Live Drops

If the issue happened recently, but you find you aren't able to reproduce the issue, go to Debugging Past Drops for the appropriate trace commands.

If you can consistently reproduce the issue, then you can run the following command in an admin command prompt to gather a fresh trace:

Console

```
Netsh wfp capture start keywords=19
<Run UWP app>
Netsh wfp capture stop
```

These commands generate a wfpdiag.cab. Inside the .cab exists a wfpdiag.xml, which contains any allow or drop netEvents and filters that existed during that repro. Without "keywords=19", the trace will only collect drop netEvents.

Inside the wfpdiag.xml, search for netEvents that have FWPM_NET_EVENT_TYPE_CLASSIFY_DROP as the netEvent type. To find the relevant drop events, search for the drop events with matching destination IP address, package SID, or application ID name. The characters in the application ID name will be separated by periods:

XML

```
(ex)

<asString>
\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.w.i.n.d.o.w.s.\\.s.y.s.t.
e.m.3.2.\\.s.v.c.h.o.s.t...e.x.e...

</asString>
```

The netEvent will have more information about the packet that was dropped including information about its capabilities, the filter that dropped the packet, and much more.

Case 1: UWP app connects to Internet target address with all capabilities

In this example, the UWP app successfully connects to bing.com [2620:1ec:c11::200].

A packet from a UWP app needs the correct networking capability token for the resource it's trying to reach.

In this scenario, the app could successfully send a packet to the Internet target because it had an Internet capability token.

The following code shows the allow netEvent of the app connecting to the target IP. The netEvent contains information about the packet including its local address, remote address, capabilities, etc.

Classify Allow netEvent, `Wfpdiag-Case-1.xml`

XML

```
<netEvent>
    <header>
    <timeStamp>2020-05-21T17:25:59.070Z</timeStamp>
    <flags numItems="9">
        <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
        <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
        <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
        <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
        <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
        <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
        <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
        <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
        <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
    </flags>
    <ipVersion>FWP_IP_VERSION_V6</ipVersion>
    <ipProtocol>6</ipProtocol>

<localAddrV6.byteArray16>2001:4898:30:3:256c:e5ba:12f3:beb1</localAddrV6.byt
eArray16>
    <remoteAddrV6.byteArray16>2620:1ec:c11::200</remoteAddrV6.byteArray16>
<localPort>52127</localPort>
<remotePort>443</remotePort>
<scopeId>0</scopeId>
<appId>

<data>5c0064006500760069006300650050c00680061007200640064006900730060b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f007700730061007000700073005c006100660036003900320062
00660066002d003600370037003900 2d003400320034006600 2d003800370030006500 2d0066
00360065003500390063003500300032003300340039005f0031002e0031002e00310030002e
```

```
0030005f007800360034005f005f00350063003000330037006a006100720035003800330039
0072005c007500770070007300 6f0063006b006500740063006c00690065006e0074002e0065
00780065000000</data>

<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.

.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...1.0...0._.x.6.4._._.5.c.0.3.7.j
.a.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
</appId>
<userId>S-1-5-21-2993214446-1947230185-131795049-1000</userId>
<addressFamily>FWP_AF_INET6</addressFamily>
<packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-3020241727-
522149888-4067122936</packageSid>
<enterpriseId/>
<policyFlags>0</policyFlags>
<effectiveName/>
</header>
<type>FWPM_NET_EVENT_TYPE_CLASSIFY_ALLOW</type>
<classifyAllow>
    <filterId>125918</filterId>
    <layerId>50</layerId>
    <reauthReason>0</reauthReason>
    <originalProfile>1</originalProfile>
    <currentProfile>1</currentProfile>
</classifyAllow>
<internalFields>
<internalFlags/>
<remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
<capabilities numItems="3">
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT</item>
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT_SERVER</item>
    <item>FWP_CAPABILITIES_FLAG_PRIVATE_NETWORK</item>
</capabilities>
<fqbnVersion>0</fqbnVersion>
<fqbnName/>
<terminatingFiltersInfo numItems="2">
    <item>
        <filterId>125918</filterId>
        <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
        <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
    <item>
        <filterId>121167</filterId>
        <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
        <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
</terminatingFiltersInfo>
</internalFields>
</netEvent>
```

The following is the filter that permitted the packet to be sent to the target address according to the **terminatingFiltersInfo** in the **netEvent**. This packet was allowed by

Filter #125918, from the InternetClient Default Rule.

InternetClient Default Rule Filter #125918, `Wfpdiag-Case-1.xml`

XML

```
<item>
    <filterKey>{3389708e-f7ae-4ebc-a61a-f659065ab24e}</filterKey>
    <displayData>
    <name>InternetClient Default Rule</name>
    <description>InternetClient Default Rule</description>
    </displayData>
    <flags/>
    <providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
    <providerData>
    <data>ad2b000000000000</data>
    <asString>.+......</asString>
    </providerData>
    <layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V6</layerKey>
    <subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey
    <weight>
    <type>FWP_EMPTY</type>
    </weight>
    <filterCondition numItems="5">
    <item>
    <fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
    <matchType>FWP_MATCH_NOT_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SID</type>
    <sid>S-1-0-0</sid>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_IP_REMOTE_ADDRESS</fieldKey>
     <matchType>FWP_MATCH_RANGE</matchType>
      <conditionValue>
      <type>FWP_RANGE_TYPE</type>
      <rangeValue>
     <valueLow>
    <type>FWP_BYTE_ARRAY16_TYPE</type>
    <byteArray16>::</byteArray16>
    </valueLow>
    <valueHigh>
    <type>FWP_BYTE_ARRAY16_TYPE</type>
    <byteArray16>ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff</byteArray16>
    </valueHigh>
    </rangeValue>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ORIGINAL_PROFILE_ID<fieldKey>
```

```
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_CURRENT_PROFILE_ID<fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
    <sd>O:LSD:(A;;CC;;;S-1-15-3-1)(A;;CC;;;WD)(A;;CC;;;AN)</sd>
    </conditionValue>
    </item>
    </filterCondition>
    <action>
    <type>FWP_ACTION_PERMIT</type>
    <filterType/>
    </action>
    <rawContext>0</rawContext>
    <reserved/>
    <filterId>125918</filterId>
    <effectiveWeight>
    <type>FWP_UINT64</type>
    <uint64>103079219136</uint64>
    </effectiveWeight>
</item>
```

Capabilities Condition in Filter #125918, `Wfpdiag-Case-1.xml`

XML

```
<item>
    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
    <sd>O:LSD:(A;;CC;;;S-1-15-3-1)(A;;CC;;;WD)(A;;CC;;;AN)</sd>
    </conditionValue>
</item>
```

This condition enables checking capabilities in this filter.

The important part of this condition is **S-1-15-3-1**, which is the capability SID for **INTERNET_CLIENT** privileges.

From the **netEvent** capabilities section, capabilities from netEvent, Wfpdiag-Case-1.xml.

XML

```
<capabilities numItems="3">
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT</item>
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT_SERVER</item>
    <item>FWP_CAPABILITIES_FLAG_PRIVATE_NETWORK</item>
</capabilities>
```

These capabilities show the packet came from an app with an Internet client token (**FWP_CAPABILITIES_FLAG_INTERNET_CLIENT**) which matches the capability SID in the filter. All the other conditions are also met for the filter, so the packet is allowed. Something to note is that the only capability token required for the packet to reach bing.com was the Internet client token, even though this example showed the packet having all capabilities.

Case 2: UWP APP can't reach Internet target address and has no capabilities

In this example, the UWP app is unable to connect to bing.com [2620:1ec:c11::200].

The following example is that of a drop netEvent that was captured in the trace.

Classify Drop netEvent, `Wfpdiag-Case-2.xml`

XML

```
<netEvent>
<header>
<timeStamp>2020-03-30T23:53:09.720Z</timeStamp>
<flags numItems="9">
    <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
    <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
```

```
</flags>
<ipVersion>FWP_IP_VERSION_V6</ipVersion>
<ipProtocol>6</ipProtocol>
<localAddrV6.byteArray16>2001:4898:1a:1045:8469:3351:e6e2:543</localAddrV6.b
yteArray16>
<remoteAddrV6.byteArray16>2620:1ec:c11::200</remoteAddrV6.byteArray16>
<localPort>63187</localPort>
<remotePort>443</remotePort>
<scopeId>0</scopeId>
<appId>
<data>5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f007700730061007000700073005c006100660036003900320062
00660066002d003600370037003900 2d003400320034006600 2d003800370030006500 2d0066
00360065003500390063003500300032003300340039005f0031002e0031002e0034002e0030
005f007800360034005f005f003500630030003300370006a006100720035003800330039007 2
005c007500770070007300 6f0063006b006500740063006c00690065006e0074002e00650078
0065000000</data>
<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.
.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...4...0._.x.6.4._._.5.c.0.3.7.j.a
.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
</appId>
<userId>S-1-5-21-2788718703-1626973220-3690764900-1000</userId>
<addressFamily>FWP_AF_INET6</addressFamily>
<packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-3020241727-
522149888-4067122936</packageSid>
<enterpriseId/>
<policyFlags>0</policyFlags>
<effectiveName/>
</header>
<type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
<classifyDrop>
<filterId>68893</filterId>
<layerId>50</layerId>
<reauthReason>0</reauthReason>
<originalProfile>1</originalProfile>
<currentProfile>1</currentProfile>
<msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
<isLoopback>false</isLoopback>
<vSwitchId/>
<vSwitchSourcePort>0</vSwitchSourcePort>
<vSwitchDestinationPort>0</vSwitchDestinationPort>
</classifyDrop>
<internalFields>
<internalFlags/>
<remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
<capabilities/>
<fqbnVersion>0</fqbnVersion>
<fqbnName/>
<terminatingFiltersInfo numItems="2">
<item>
<filterId>68893</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
<actionType>FWP_ACTION_BLOCK</actionType>
```

```
</item>
<item>
<filterId>68879</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
<actionType>FWP_ACTION_PERMIT</actionType>
</item>
</terminatingFiltersInfo>
</internalFields>
</netEvent>
```

The first thing that you should check in the **netEvent** is the capabilities field. In this example, the capabilities field is empty, indicating that the UWP app wasn't configured with any capability tokens to allow it to connect to a network.

Internal Fields from netEvent, `Wfpdiag-Case-2.xml`

XML

```
<internalFields>
<internalFlags/>
<remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
<capabilities/>
<fqbnVersion>0</fqbnVersion>
<fqbnName/>
<terminatingFiltersInfo numItems="2">
<item>
<filterId>68893</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
<actionType>FWP_ACTION_BLOCK</actionType>
</item>
<item>
<filterId>68879</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
<actionType>FWP_ACTION_PERMIT</actionType>
</item>
</terminatingFiltersInfo>
</internalFields>
```

The **netEvent** also shows information about the filter that explicitly dropped this packet, like the **FilterId**, listed under classify drop.

Classify Drop from netEvent, `Wfpdiag-Case-2.xml`

XML

```
<classifyDrop>
<filterId>68893</filterId>
<layerId>50</layerId>
```

```
<reauthReason>0</reauthReason>
<originalProfile>1</originalProfile>
<currentProfile>1</currentProfile>
<msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
<isLoopback>false</isLoopback>
<vSwitchId/>
<vSwitchSourcePort>0</vSwitchSourcePort>
<vSwitchDestinationPort>0</vSwitchDestinationPort>
</classifyDrop>
```

If you search for the filter #68893 in Wfpdiag-Case2.xml, you'll see that the packet was dropped by a Block Outbound Default Rule filter.

Block Outbound Default Rule Filter #68893, `Wfpdiag-Case-2.xml`

XML

```
<item>
    <filterKey>{6d51582f-bcf8-42c4-afc9-e2ce7155c11b}</filterKey>
/t<displayData>
    <name>**Block Outbound Default Rule**</name>
    <description>Block Outbound Default Rule</description>
    </displayData>
    <flags/>
    <providerKey>{4b153735-1049-4480-aab4-d1b9bdc03710}</providerKey>
    <providerData>
    <data>b001000000000000</data>
    <asString>........</asString>
    </providerData>
    <layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V6</layerKey>
    <subLayerKey>{b3cdd441-af90-41ba-a745-7c6008ff2300}</subLayerKey>
    <weight>
    <type>FWP_EMPTY</type>
    </weight>
    <filterCondition numItems="1">
    <item>
    <fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
    <matchType>FWP_MATCH_NOT_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SID</type>
    <sid>S-1-0-0</sid>
    </conditionValue>
    </item>
    </filterCondition>
    <action>
    <type>FWP_ACTION_BLOCK</type>
    <filterType/>
    </action>
    <rawContext>0</rawContext>
    <reserved/>
```

```
    <filterId>68893</filterId>
    <effectiveWeight>
    <type>FWP_UINT64</type>
    <uint64>68719476736</uint64>
    </effectiveWeight>
</item>
```

A packet will reach a default block filter if the packet was unable to match any of the conditions of other filters, and not allowed by the other filters in the same sublayer.

If the packet had the correct capability token, **FWP_CAPABILITIES_FLAG_INTERNET_CLIENT**, it would have matched a condition for a non-default block filter, and would have been permitted to reach bing.com. Without the correct capability tokens, the packet will be explicitly dropped by a default block outbound filter.

Case 3: UWP app can't reach Internet target address without Internet Client capability

In this example, the app is unable to connect to bing.com [2620:1ec:c11::200].

The app in this scenario only has private network capabilities (Client and Server). The app is trying to connect to an Internet resource (bing.com), but only has a private network token. Therefore, the packet will be dropped.

Classify Drop netEvent, `Wfpdiag-Case-3.xml`

XML

```
<netEvent>
<header>
<timeStamp>2020-03-31T16:57:18.570Z</timeStamp>
<flags numItems="9">
<item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
<item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
<item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
<item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
<item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
<item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
<item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
<item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
<item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
</flags>
<ipVersion>FWP_IP_VERSION_V6</ipVersion>
<ipProtocol>6</ipProtocol>
<localAddrV6.byteArray16>2001:4898:1a:1045:9c65:7805:dd4a:cc4b</localAddrV6.
```

```
byteArray16>
<remoteAddrV6.byteArray16>2620:1ec:c11::200</remoteAddrV6.byteArray16>
<localPort>64086</localPort>
<remotePort>443</remotePort>
<scopeId>0</scopeId>
<appId>
<data>5c00640065007600690063006500 5c00680061007200640064006900 73006b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f007700730061007000700073005c00610066003600390032006 2
00660066002d00360037003700390 02d00340032003400660 02d00380037003000650 02d0066
00360065003500390063003500300032003300340039005f0031002e0031002e0035002e0030
005f007800360034005f005f00350063003000330037006a00610072003500380033003900 72
005c00750077007000730 06f0063006b006500740063006c00690065006e0074002e00650078
0065000000</data>
<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.
.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...5...0._.x.6.4._._.5.c.0.3.7.j.a
.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
</appId>
<userId>S-1-5-21-2788718703-1626973220-3690764900-1000</userId>
<addressFamily>FWP_AF_INET6</addressFamily>
<packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-3020241727-
522149888-4067122936</packageSid>
<enterpriseId/>
<policyFlags>0</policyFlags>
<effectiveName/>
</header>
<type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
<classifyDrop>
<filterId>68893</filterId>
<layerId>50</layerId>
<reauthReason>0</reauthReason>
<originalProfile>1</originalProfile>
<currentProfile>1</currentProfile>
<msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
<isLoopback>false</isLoopback>
<vSwitchId/>
<vSwitchSourcePort>0</vSwitchSourcePort>
<vSwitchDestinationPort>0</vSwitchDestinationPort>
</classifyDrop>
<internalFields>
<internalFlags/>
<remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
**<capabilities numItems="1">**
**<item>FWP_CAPABILITIES_FLAG_PRIVATE_NETWORK</item>**
**</capabilities>**
<fqbnVersion>0</fqbnVersion>
<fqbnName/>
<terminatingFiltersInfo numItems="2">
<item>
<filterId>68893</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
<actionType>FWP_ACTION_BLOCK</actionType>
</item>
<item>
```

```
<filterId>68879</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
<actionType>FWP_ACTION_PERMIT</actionType>
</item>
</terminatingFiltersInfo>
</internalFields>
</netEvent>
```

Case 4: UWP app can't reach Intranet target address without Private Network capability

In this example, the UWP app is unable to reach the Intranet target address, 10.50.50.50, because it doesn't have a Private Network capability.

Classify Drop netEvent, `Wfpdiag-Case-4.xml`

XML

```
<netEvent>
<header>
    <timeStamp>2020-05-22T21:29:28.601Z</timeStamp>
    <flags numItems="9">
    <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
    <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
    </flags>
    <ipVersion>FWP_IP_VERSION_V4</ipVersion>
    <ipProtocol>6</ipProtocol>
    <localAddrV4>10.216.117.17</localAddrV4>
    <remoteAddrV4>10.50.50.50</remoteAddrV4>
    <localPort>52998</localPort>
    <remotePort>53</remotePort>
    <scopeId>0</scopeId>
    <appId>

<data>5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f00770073006100700070007300 5c006100660036003900320062
00660066002d003600370037003900 2d003400320034006600 2d003800370030006500 2d0066
00360065003500390063003500300032003300340039005f0031002e0031002e00310031002e
0030005f00780036003400 5f005f00350063003000330037006a006100720035003800330039
0072005c00750077007000730063006f006300 6b006500740063006c00690065006e0074002e0065
00780065000000</data>
```

```
<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.

.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...1.1...0._.x.6.4._._.5.c.0.3.7.j
.a.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
    </appId>
    <userId>S-1-5-21-2993214446-1947230185-131795049-1000</userId>
    <addressFamily>FWP_AF_INET</addressFamily>
    <packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-
3020241727-522149888-4067122936</packageSid>
    <enterpriseId/>
    <policyFlags>0</policyFlags>
    <effectiveName/>
</header>
    <type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
    <classifyDrop>
    <filterId>121180</filterId>
    <layerId>48</layerId>
    <reauthReason>0</reauthReason>
    <originalProfile>1</originalProfile>
    <currentProfile>1</currentProfile>
    <msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
    <isLoopback>false</isLoopback>
    <vSwitchId/>
    <vSwitchSourcePort>0</vSwitchSourcePort>
    <vSwitchDestinationPort>0</vSwitchDestinationPort>
</classifyDrop>
<internalFields>
    <internalFlags/>
    <remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
    <capabilities numItems="2">
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT</item>
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT_SERVER</item>
    </capabilities>
    <fqbnVersion>0</fqbnVersion>
    <fqbnName/>
    <terminatingFiltersInfo numItems="2">
    <item>
    <filterId>121180</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
    <actionType>FWP_ACTION_BLOCK</actionType>
    </item>
    <item>
    <filterId>121165</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
    <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
    </terminatingFiltersInfo>
</internalFields>
</netEvent>
```

Case 5: UWP app can't reach "Intranet" target address with Private Network capability

In this example, the UWP app is unable to reach the Intranet target address, 10.1.1.1, even though it has a Private Network capability token.

Classify Drop netEvent, `Wfpdiag-Case-5.xml`

XML


```xml
<netEvent>
    <header>
    <timeStamp>2020-05-22T20:54:53.499Z</timeStamp>
    <flags numItems="9">
    <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
    <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
    </flags>
    <ipVersion>FWP_IP_VERSION_V4</ipVersion>
    <ipProtocol>6</ipProtocol>
    <localAddrV4>10.216.117.17</localAddrV4>
    <remoteAddrV4>10.1.1.1</remoteAddrV4>
    <localPort>52956</localPort>
    <remotePort>53</remotePort>
    <scopeId>0</scopeId>
    <appId>

<data>5c006400650076006900630065005c006800610072006400640069007300 6b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f00770073006100700070007300 5c006100660036003900320062
00660066002d003600370037003900 2d003400320034006600 2d003800370030006500 2d0066
00360065003500390063003500300032003300340039005f0031002e0031002e00310033002e
0030005f007800360034005f005f003500630030003300370 06a006100720035003800330039
0072005c007500770070007300 6f0063006b006500740063006c00690065006e0074002e0065
00780065000000</data>

<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.

.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...1.3...0._.x.6.4._._.5.c.0.3.7.j
.a.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
    </appId>
    <userId>S-1-5-21-2993214446-1947230185-131795049-1000</userId>
    <addressFamily>FWP_AF_INET</addressFamily>
    <packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-
```

```
3020241727-522149888-4067122936</packageSid>
    <enterpriseId/>
    <policyFlags>0</policyFlags>
    <effectiveName/>
</header>
    <type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
    <classifyDrop>
    <filterId>121180</filterId>
    <layerId>48</layerId>
    <reauthReason>0</reauthReason>
    <originalProfile>1</originalProfile>
    <currentProfile>1</currentProfile>
    <msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
    <isLoopback>false</isLoopback>
    <vSwitchId/>
    <vSwitchSourcePort>0</vSwitchSourcePort>
    <vSwitchDestinationPort>0</vSwitchDestinationPort>
    </classifyDrop>
    <internalFields>
    <internalFlags/>
    <remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
    <capabilities numItems="1">
    <item>FWP_CAPABILITIES_FLAG_PRIVATE_NETWORK</item>
    </capabilities>
    <fqbnVersion>0</fqbnVersion>
    <fqbnName/>
    <terminatingFiltersInfo numItems="2">
    <item>
    <filterId>121180</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
    <actionType>FWP_ACTION_BLOCK</actionType>
    </item>
    <item>
    <filterId>121165</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
    <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
    </terminatingFiltersInfo>
    </internalFields>
</netEvent>
```

The following shows the filter that blocked the event:

Block Outbound Default Rule Filter #121180, `Wfpdiag-Case-5.xml`

XML

```
<item>
    <filterKey>{e62a1a22-c80a-4518-a7f8-e7d1ef3a9ff6}</filterKey>
    <displayData>
```

```
    <name>Block Outbound Default Rule</name>
    <description>Block Outbound Default Rule</description>
    </displayData>
    <flags/>
    <providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
    <providerData>
    <data>c029000000000000</data>
    <asString>.)......</asString>
    </providerData>
    <layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
    <subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
    <weight>
    <type>FWP_EMPTY</type>
    </weight>
    <filterCondition numItems="1">
    <item>
    <fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
    <matchType>FWP_MATCH_NOT_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SID</type>
    <sid>S-1-0-0</sid>
    </conditionValue>
    </item>
    </filterCondition>
    <action>
    <type>FWP_ACTION_BLOCK</type>
    <filterType/>
    </action>
    <rawContext>0</rawContext>
    <reserved/>
    <filterId>121180</filterId>
    <effectiveWeight>
    <type>FWP_UINT64</type>
    <uint64>274877906944</uint64>
    </effectiveWeight>
</item>
```

If the target was in the private range, then it should have been allowed by a PrivateNetwork Outbound Default Rule filter.

The following PrivateNetwork Outbound Default Rule filters have conditions for matching Intranet IP addresses. Since the expected Intranet target address, 10.1.1.1, isn't included in these filters it becomes clear that the address isn't in the private range. Check the policies that configure the private range on the device (MDM, Group Policy, etc.) and make sure it includes the private target address you wanted to reach.

PrivateNetwork Outbound Default Rule Filters, `Wfpdiag-Case-5.xml`

```
```

XML

```
<item>
    <filterKey>{fd65507b-e356-4e2f-966f-0c9f9c1c6e78}</filterKey>
    <displayData>
    <name>PrivateNetwork Outbound Default Rule</name>
    <description>PrivateNetwork Outbound Default Rule</description>
    </displayData>
    <flags/>
    <providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
    <providerData>
    <data>f22d000000000000</data>
    <asString>.-......</asString>
    </providerData>
    <layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
    <subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
    <weight>
    <type>FWP_EMPTY</type>
    </weight>
    <filterCondition numItems="5">
    <item>
    <fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
    <matchType>FWP_MATCH_NOT_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SID</type>
    <sid>S-1-0-0</sid>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_IP_REMOTE_ADDRESS</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1.1.1.1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ORIGINAL_PROFILE_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_CURRENT_PROFILE_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
```

```
<matchType>FWP_MATCH_EQUAL</matchType>
<conditionValue>
<type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
<sd>O:LSD:(A;;CC;;;S-1-15-3-3)(A;;CC;;;WD)(A;;CC;;;AN)</sd>
</conditionValue>
</item>
</filterCondition>
<action>
<type>FWP_ACTION_PERMIT</type>
<filterType/>
</action>
<rawContext>0</rawContext>
<reserved/>
<filterId>129656</filterId>
<effectiveWeight>
<type>FWP_UINT64</type>
<uint64>144115600392724416</uint64>
</effectiveWeight>
</item>
<item>
<filterKey>{b11b4f8a-222e-49d6-8d69-02728681d8bc}</filterKey>
<displayData>
<name>PrivateNetwork Outbound Default Rule</name>
<description>PrivateNetwork Outbound Default Rule</description>
</displayData>
<flags/>
<providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
<providerData>
<data>f22d000000000000</data>
<asString>.-......</asString>
</providerData>
<layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
<subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
<weight>
<type>FWP_EMPTY</type>
</weight>
<filterCondition numItems="5">
<item>
<fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
<matchType>FWP_MATCH_NOT_EQUAL</matchType>
<conditionValue>
<type>FWP_SID</type>
<sid>S-1-0-0</sid>
</conditionValue>
</item>
<item>
<fieldKey>FWPM_CONDITION_IP_REMOTE_ADDRESS</fieldKey>
<matchType>FWP_MATCH_RANGE</matchType>
<conditionValue>
<type>FWP_RANGE_TYPE</type>
<rangeValue>
<valueLow>
<type>FWP_UINT32</type>
<uint32>172.16.0.0</uint32>
</valueLow>
```

```
    <valueHigh>
    <type>FWP_UINT32</type>
    <uint32>172.31.255.255</uint32>
    </valueHigh>
    </rangeValue>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ORIGINAL_PROFILE_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_CURRENT_PROFILE_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_UINT32</type>
    <uint32>1</uint32>
    </conditionValue>
    </item>
    <item>
    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
    <matchType>FWP_MATCH_EQUAL</matchType>
    <conditionValue>
    <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
    <sd>O:LSD:(A;;CC;;;S-1-15-3-3)(A;;CC;;;WD)(A;;CC;;;AN)</sd>
    </conditionValue>
    </item>
    </filterCondition>
    <action>
    <type>FWP_ACTION_PERMIT</type>
    <filterType/>
    </action>
    <rawContext>0</rawContext>
    <reserved/>
    <filterId>129657</filterId>
    <effectiveWeight>
    <type>FWP_UINT64</type>
    <uint64>36029209335832512</uint64>
    </effectiveWeight>
</item>
<item>
    <filterKey>{21cd82bc-6077-4069-94bf-750e5a43ca23}</filterKey>
    <displayData>
    <name>PrivateNetwork Outbound Default Rule</name>
    <description>PrivateNetwork Outbound Default Rule</description>
    </displayData>
    <flags/>
    <providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
    <providerData>
    <data>f22d000000000000</data>
    <asString>.-......</asString>
```

```
</providerData>
<layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
<subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
<weight>
<type>FWP_EMPTY</type>
</weight>
<filterCondition numItems="5">
<item>
<fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
<matchType>FWP_MATCH_NOT_EQUAL</matchType>
<conditionValue>
<type>FWP_SID</type>
<sid>S-1-0-0</sid>
</conditionValue>
</item>
<item>
<fieldKey>FWPM_CONDITION_IP_REMOTE_ADDRESS</fieldKey>
<matchType>FWP_MATCH_RANGE</matchType>
<conditionValue>
<type>FWP_RANGE_TYPE</type>
<rangeValue>
<valueLow>
<type>FWP_UINT32</type>
<uint32>192.168.0.0</uint32>
</valueLow>
<valueHigh>
<type>FWP_UINT32</type>
<uint32>192.168.255.255</uint32>
</valueHigh>
</rangeValue>
</conditionValue>
</item>
<item>
<fieldKey>FWPM_CONDITION_ORIGINAL_PROFILE_ID</fieldKey>
<matchType>FWP_MATCH_EQUAL</matchType>
<conditionValue>
<type>FWP_UINT32</type>
<uint32>1</uint32>
</conditionValue>
</item>
<item>
<fieldKey>FWPM_CONDITION_CURRENT_PROFILE_ID</fieldKey>
<matchType>FWP_MATCH_EQUAL</matchType>
<conditionValue>
<type>FWP_UINT32</type>
<uint32>1</uint32>
</conditionValue>
</item>
<item>
<fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
<matchType>FWP_MATCH_EQUAL</matchType>
<conditionValue>
<type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
<sd>O:LSD:(A;;CC;;;S-1-15-3-3)(A;;CC;;;WD)(A;;CC;;;AN)</sd>
</conditionValue>
```

```
    </item>
    </filterCondition>
    <action>
    <type>FWP_ACTION_PERMIT</type>
    <filterType/>
    </action>
    <rawContext>0</rawContext>
    <reserved/>
    <filterId>129658</filterId>
    <effectiveWeight>
    <type>FWP_UINT64</type>
    <uint64>36029209335832512</uint64>
    </effectiveWeight>
</item>
```

Debugging Past Drops

If you're debugging a network drop from the past or from a remote machine, you may have traces already collected from Feedback Hub, such as nettrace.etl and wfpstate.xml. Once nettrace.etl is converted, nettrace.txt will have the netEvents of the reproduced event, and wfpstate.xml will contain the filters that were present on the machine at the time.

If you don't have a live repro or traces already collected, you can still collect traces after the UWP network connectivity issue has happened by running these commands in an admin command prompt:

XML

```
    <Run UWP app>
    Netsh wfp show netevents
    Netsh wfp show state
```

`Netsh wfp show netevents` creates `netevents.xml`, which contains the past net events. `Netsh wfp show state` creates wfpstate.xml, which contains the current filters present on the machine.

Unfortunately, collecting traces after the UWP network connectivity issue isn't always reliable.

NetEvents on the device are stored in a buffer. Once that buffer has reached maximum capacity, the buffer will overwrite older net events. Due to the buffer overwrite, it's possible that the collected netevents.xml won't contain the net event associated with the UWP network connectivity issue. It could have been overwritten. Additionally, filters on the device can get deleted and re-added with different filterIds due to miscellaneous

events on the device. Because of these implications, a **filterId** from **netsh wfp show netevents** may not necessarily match any filter in **netsh wfp show state** because that **filterId** may be outdated.

If you can reproduce the UWP network connectivity issue consistently, we recommend using the commands from Debugging Live Drops instead.

Additionally, you can still follow the examples from Debugging Live Drops section using the trace commands in this section, even if you don't have a live repro. The **netEvents** and filters are stored in one file in Debugging Live Drops as opposed to two separate files in the following Debugging Past Drops examples.

Case 7: Debugging Past Drop - UWP app can't reach Internet target address and has no capabilities

In this example, the UWP app is unable to connect to bing.com.

Classify Drop Net Event, `NetEvents-Case-7.xml`

XML

```
<item>
<header>
<timeStamp>2020-05-04T22:04:07.039Z</timeStamp>
<flags numItems="9">
<item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
<item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
<item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
<item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
<item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
<item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
<item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
<item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
<item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
</flags>
<ipVersion>FWP_IP_VERSION_V4</ipVersion>
<ipProtocol>6</ipProtocol>
<localAddrV4>10.195.36.30</localAddrV4>
<remoteAddrV4>204.79.197.200</remoteAddrV4>
<localPort>57062</localPort>
<remotePort>443</remotePort>
<scopeId>0</scopeId>
<appId>
<data>5c00640065007600690063006500
5c00680061007200640064006900730
06b0076006f006c0075006d00650031005c00700072006f00670072006100
6d00200066006900
6c00650073
```

```
005c00770069006e0064006f007700730061007000700073005c006100660036003900320062
00660066002d003600370037003900 2d003400320034006600 2d003800370030006500 2d0066
00360065003500390063003500300032003300340039005f0031002e0031002e00310032002e
0030005f007800360034005f005f003500630030003300370 06a006100720035003800330039
0072005c007500770070007300 6f0063006b006500740063006c00690065006e0074002e0065
00780065000000</data>
<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.
.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...1.2...0._.x.6.4._._.5.c.0.3.7.j
.a.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
</appId>
<userId>S-1-5-21-1578316205-4060061518-881547182-1000</userId>
<addressFamily>FWP_AF_INET</addressFamily>
<packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-3020241727-
522149888-4067122936</packageSid>
<enterpriseId/>
<policyFlags>0</policyFlags>
<effectiveName/>
</header>
<type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
<classifyDrop>
<filterId>206064</filterId>
<layerId>48</layerId>
<reauthReason>0</reauthReason>
<originalProfile>1</originalProfile>
<currentProfile>1</currentProfile>
<msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
<isLoopback>false</isLoopback>
<vSwitchId/>
<vSwitchSourcePort>0</vSwitchSourcePort>
<vSwitchDestinationPort>0</vSwitchDestinationPort>
</classifyDrop>
<internalFields>
<internalFlags/>
<remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
<capabilities/>
<fqbnVersion>0</fqbnVersion>
<fqbnName/>
<terminatingFiltersInfo numItems="2">
<item>
<filterId>206064</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
<actionType>FWP_ACTION_BLOCK</actionType>
</item>
<item>
<filterId>206049</filterId>
<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
<actionType>FWP_ACTION_PERMIT</actionType>
</item>
</terminatingFiltersInfo>
</internalFields>
</item>
```

The Internal fields list no active capabilities, and the packet is dropped at nfilter 206064.

This filter is a default block rule filter, meaning the packet passed through every filter that could have allowed it, but because conditions didn't match for any of those filters, the packet fell to the filter that blocks any packet that the Security Descriptor doesn't match.

Block Outbound Default Rule Filter #206064, `FilterState-Case-7.xml`

XML

```
<item>
<filterKey>{f138d1ad-9293-478f-8519-c3368e796711}</filterKey>
<displayData>
<name>Block Outbound Default Rule</name>
<description>Block Outbound Default Rule</description>
</displayData>
<flags/>
<providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
<providerData>
<data>2e65000000000000</data>
<asString>.e......</asString>
</providerData>
<layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
<subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
<weight>
<type>FWP_EMPTY</type>
</weight>
<filterCondition numItems="1">
<item>
<fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
<matchType>FWP_MATCH_NOT_EQUAL</matchType>
<conditionValue>
<type>FWP_SID</type>
<sid>S-1-0-0</sid>
</conditionValue>
</item>
</filterCondition>
<action>
<type>FWP_ACTION_BLOCK</type>
<filterType/>
</action>
<rawContext>0</rawContext>
<reserved/>
<filterId>206064</filterId>
<effectiveWeight>
<type>FWP_UINT64</type>
<uint64>274877906944</uint64>
</effectiveWeight>
</item>
```

Case 8: Debugging Past Drop - UWP app connects to Internet target address with all capabilities

In this example, the UWP app successfully connects to bing.com [204.79.197.200].

Classify Allow Net Event, `NetEvents-Case-8.xml`

XML

```
<item>
    <header>
    <timeStamp>2020-05-04T18:49:55.101Z</timeStamp>
    <flags numItems="9">
    <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
    <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
    <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
    <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
    <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
    </flags>
    <ipVersion>FWP_IP_VERSION_V4</ipVersion>
    <ipProtocol>6</ipProtocol>
    <localAddrV4>10.195.36.30</localAddrV4>
    <remoteAddrV4>204.79.197.200</remoteAddrV4>
    <localPort>61673</localPort>
    <remotePort>443</remotePort>
    <scopeId>0</scopeId>
    <appId>

<data>5c00640065007600690063006500 5c006800610072006400640069007300 6b0076006f
006c0075006d00650031005c00700072006f006700720061006d002000660069006c00650073
005c00770069006e0064006f00770073006100700070007300 5c00610066003600390032006 2
00660066002d00360037003700390020 0034003200340066002d00380037003000650020 0066
00360065003500390063003500300032003300340039005f0031002e0031002e00310030002e
0030005f00780036003400 5f005f00350063003000330037006a00610072003500380033003 9
0072005c00750077007000730 06f0063006b006500740063006c00690065006e0074002e0065
00780065000000</data>

<asString>\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.p.r.o.g.r.a.m.

.f.i.l.e.s.\\.w.i.n.d.o.w.s.a.p.p.s.\\.a.f.6.9.2.b.f.f.-.6.7.7.9.-.4.2.4.f.-
.8.7.0.e.-.f.6.e.5.9.c.5.0.2.3.4.9._.1...1...1.0...0._.x.6.4._._.5.c.0.3.7.j
.a.r.5.8.3.9.r.\\.u.w.p.s.o.c.k.e.t.c.l.i.e.n.t...e.x.e...</asString>
    </appId>
    <userId>S-1-5-21-1578316205-4060061518-881547182-1000</userId>
    <addressFamily>FWP_AF_INET</addressFamily>
```

```
    <packageSid>S-1-15-2-4163697451-3176919390-1155390458-2883473650-
3020241727-522149888-4067122936</packageSid>
    <enterpriseId/>
    <policyFlags>0</policyFlags>
    <effectiveName/>
    </header>
    <type>FWPM_NET_EVENT_TYPE_CLASSIFY_ALLOW</type>
    <classifyAllow>
    <filterId>208757</filterId>
    <layerId>48</layerId>
    <reauthReason>0</reauthReason>
    <originalProfile>1</originalProfile>
    <currentProfile>1</currentProfile>
    </classifyAllow>
    <internalFields>
    <internalFlags/>
    <remoteAddrBitmap>0000000000000000</remoteAddrBitmap>
    <capabilities numItems="3">
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT</item>
    <item>FWP_CAPABILITIES_FLAG_INTERNET_CLIENT_SERVER</item>
    <item>FWP_CAPABILITIES_FLAG_PRIVATE_NETWORK</item>
    </capabilities>
    <fqbnVersion>0</fqbnVersion>
    <fqbnName/>
    <terminatingFiltersInfo numItems="2">
    <item>
    <filterId>208757</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
    <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
    <item>
    <filterId>206049</filterId>
    <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
    <actionType>FWP_ACTION_PERMIT</actionType>
    </item>
    </terminatingFiltersInfo>
    </internalFields>
</item>
```

All capabilities are enabled and the resulting filter determining the flow of the packet is 208757.

The filter stated above with action permit:

InternetClient Default Rule Filter #208757, `FilterState-Case-8.xml`

XML

```
<item>
    <filterKey>{e0f6f24e-1f0a-4f1a-bdd8-b9277c144fb5}</filterKey>
```

```
    <displayData>
    <name>InternetClient Default Rule</name>
        <description>InternetClient Default Rule</description>
                    </displayData>
                    <flags/>
                    <providerKey>FWPM_PROVIDER_MPSSVC_WSH</providerKey>
                    <providerData>
                        <data>e167000000000000</data>
                        <asString>.g......</asString>
                    </providerData>
                    <layerKey>FWPM_LAYER_ALE_AUTH_CONNECT_V4</layerKey>
                    <subLayerKey>FWPM_SUBLAYER_MPSSVC_WSH</subLayerKey>
                    <weight>
                        <type>FWP_EMPTY</type>
                    </weight>
                    <filterCondition numItems="5">
                        <item>

<fieldKey>FWPM_CONDITION_ALE_PACKAGE_ID</fieldKey>
                            <matchType>FWP_MATCH_NOT_EQUAL</matchType>
                            <conditionValue>
                                <type>FWP_SID</type>
                                <sid>S-1-0-0</sid>
                            </conditionValue>
                        </item>
                        <item>

<fieldKey>FWPM_CONDITION_IP_REMOTE_ADDRESS</fieldKey>
                            <matchType>FWP_MATCH_RANGE</matchType>
                            <conditionValue>
                                <type>FWP_RANGE_TYPE</type>
                                <rangeValue>
                                    <valueLow>
                                        <type>FWP_UINT32</type>
                                        <uint32>0.0.0.0</uint32>
                                    </valueLow>
                                    <valueHigh>
                                        <type>FWP_UINT32</type>
                                        <uint32>255.255.255.255</uint32>
                                    </valueHigh>
                                </rangeValue>
                            </conditionValue>
                        </item>
                        <item>

<fieldKey>FWPM_CONDITION_ORIGINAL_PROFILE_ID</fieldKey>
                            <matchType>FWP_MATCH_EQUAL</matchType>
                            <conditionValue>
                                <type>FWP_UINT32</type>
                                <uint32>1</uint32>
                            </conditionValue>
                        </item>
                        <item>

<fieldKey>FWPM_CONDITION_CURRENT_PROFILE_ID</fieldKey>
```

```
                            <matchType>FWP_MATCH_EQUAL</matchType>
                            <conditionValue>
                                <type>FWP_UINT32</type>
                                <uint32>1</uint32>
                            </conditionValue>
                        </item>
                        <item>
                            <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                            <matchType>FWP_MATCH_EQUAL</matchType>
                            <conditionValue>
                                <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
                                <sd>O:LSD:(A;;CC;;;S-1-15-3-1)(A;;CC;;;WD)
(A;;CC;;;AN)</sd>
                            </conditionValue>
                        </item>
                    </filterCondition>
                    <action>
                        <type>FWP_ACTION_PERMIT</type>
                        <filterType/>
                    </action>
                    <rawContext>0</rawContext>
                    <reserved/>
                    <filterId>208757</filterId>
                    <effectiveWeight>
                        <type>FWP_UINT64</type>
                        <uint64>412316868544</uint64>
                    </effectiveWeight>
                </item>
```

Feedback

Was this page helpful? Yes No

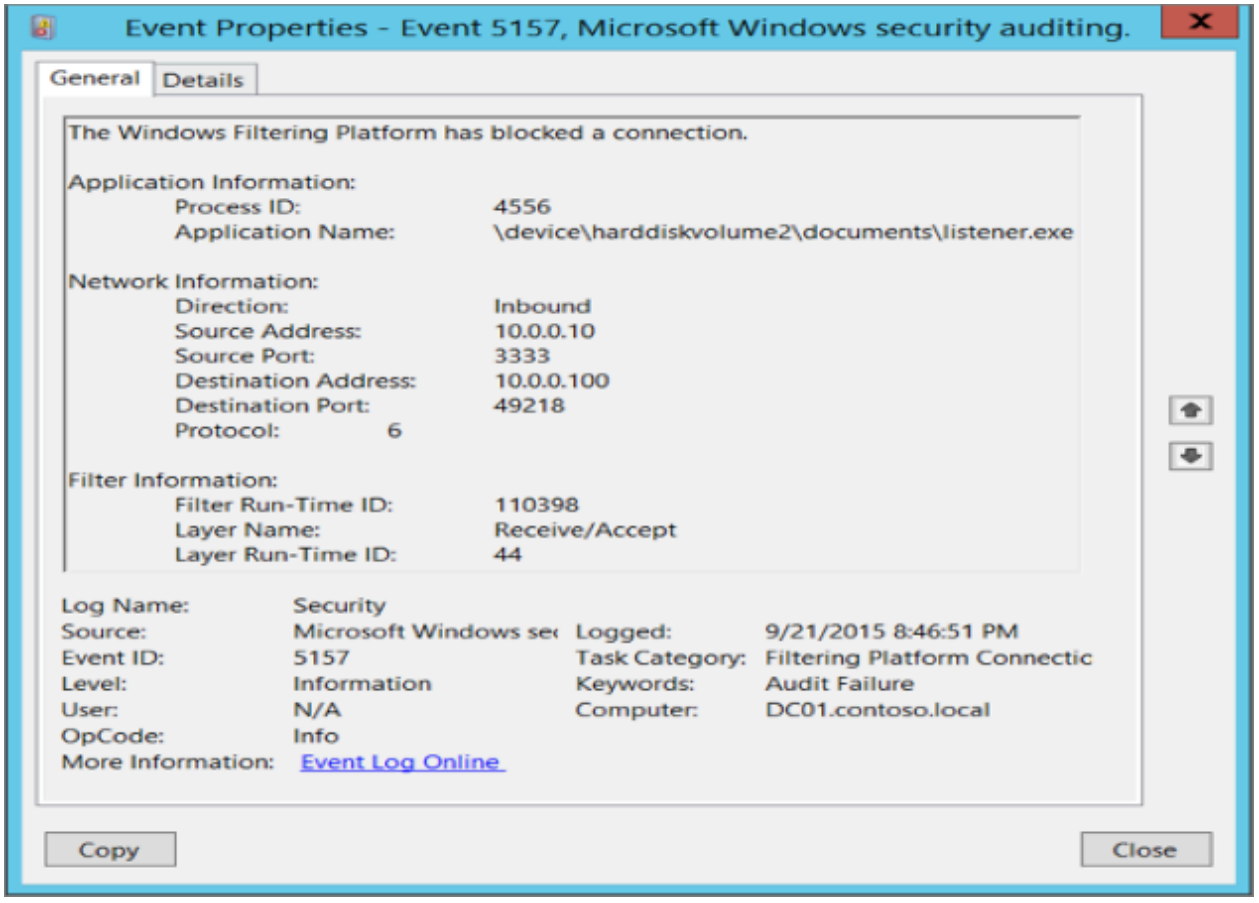
Provide product feedback

Filter origin audit log

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

When investigating packet drop events, you can use the field `Filter Run-Time ID` from Windows Filtering Platform (WFP) audits `5157` or `5152`.



The *filter ID* uniquely identifies the filter that caused the packet drop. The filter ID can be searched in the WFP state dump output to trace back to the Firewall rule where the filter originated from. However, the filter ID isn't a reliable source for tracing back to the filter or the rule, as the filter ID can change for many reasons despite the rule not changing at all. The change in ID makes the diagnosis process error-prone and difficult.

To debug packet drop events correctly and efficiently, you need more context about the blocking filter, such as its origin. The blocking filters can be categorized under these filter origins:

1. Firewall rules
2. Firewall default block filters
    a. AppContainer loopback
    b. Boot time default

c. Quarantine default
d. Query user default
e. Stealth
f. Universal Windows Platform (UWP) default
g. Windows Service Hardening (WSH) default

The next section describes the improvements made to audits `5157` and `5152` in Windows 11 and Windows Server 2022, and how the filter origins are used in these events.

Improved firewall audit

Starting in Windows 11 and Windows Server 2022, two new fields added to the audit `5157` and `5152` events are *Filter Origin* and *Interface Index*:

- The *Filter Origin* field helps identify the cause of the drop. Packet drops from firewall are explicitly dropped by default block filters created by the Windows Firewall service or a firewall rule that may be created by users, policies, services, apps, etc. Filter Origin` specifies either the *rule ID* (a unique identifier of a Firewall rule) or the name of one of the default block filters
- The *Interface Index* field specifies the network interface in which the packet was dropped. This field helps to identify which interface was quarantined, if the *Filter Origin* is a *Quarantine Default*

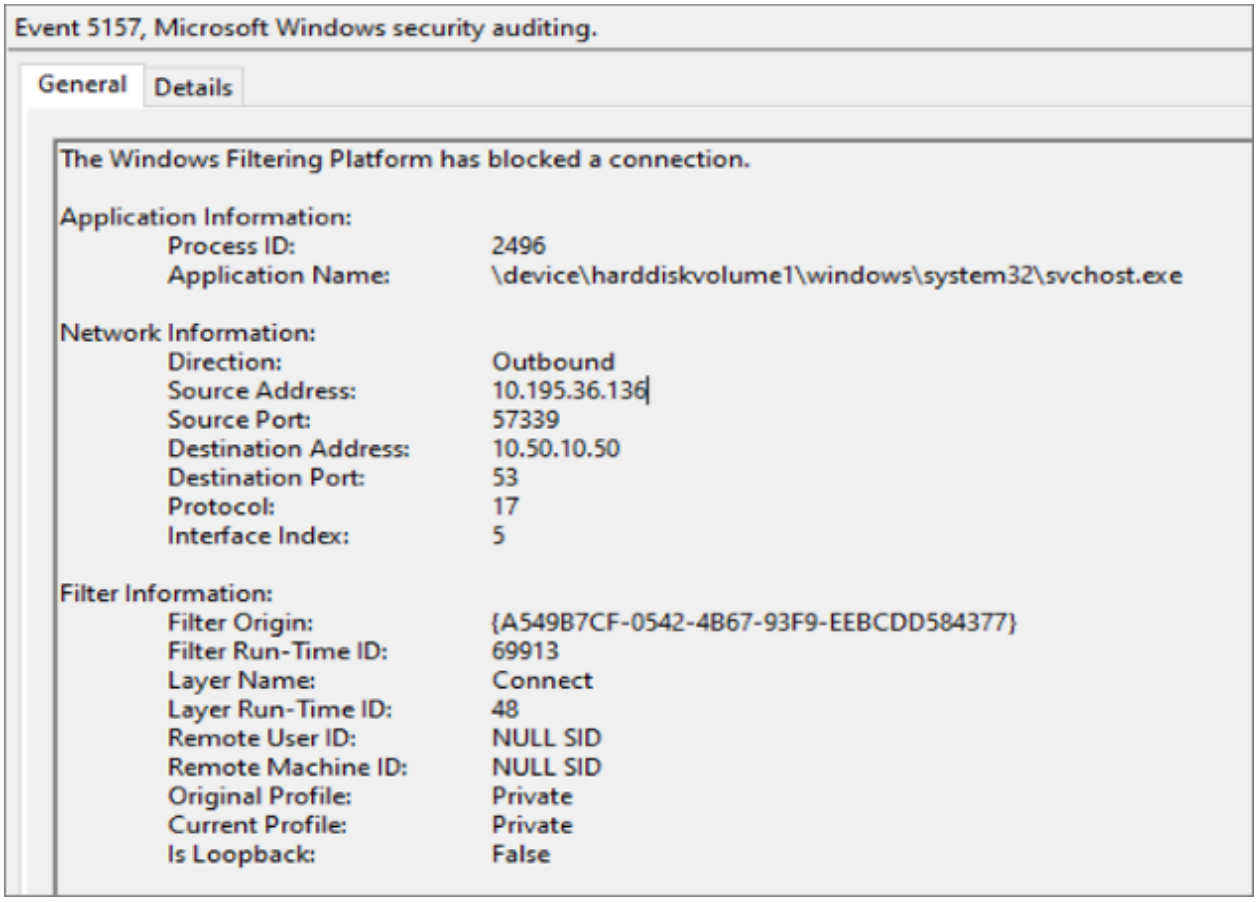
To enable a specific audit event, run the corresponding command in an administrator command prompt:

Expand table

Audit #	Enable command	Link
5157	`Auditpol /set /category:"System" /SubCategory:"Filtering Platform Connection" /success:enable /failure:enable`	5157(F): The Windows Filtering Platform has blocked a connection.
5152	`Auditpol /set /category:"System" /SubCategory:"Filtering Platform Packet Drop" /success:enable /failure:enable`	5152(F): The Windows Filtering Platform blocked a packet.

Example flow of debugging packet drops with filter origin

As the audit surfaces *Filter Origin* and *Interface Index*, the network admin can determine the root cause of the network packet drop, and the interface it happened on.



The next sections are divided by *Filter Origin* type, the value is either a rule name or the name of one of the default block filters. If the filter origin is one of the default block filters, skip to the section, Firewall default block filters.

Firewall rules

Run the following PowerShell command to generate the rule information using `Filter Origin`.

Powershell

```
Get-NetFirewallRule -Name "<Filter Origin>"
Get-NetFirewallRule -Name " {A549B7CF-0542-4B67-93F9-EEBCDD584377} "
```

```
PS C:\Windows\system32> Get-NetFirewallRule -Name "{A549B7CF-0542-4B67-93F9-EEBCDD584377}"


Name                  : {A549B7CF-0542-4B67-93F9-EEBCDD584377}
DisplayName           : test
Description           :
DisplayGroup          :
Group                 :
Enabled               : True
Profile               : Any
Platform              : {}
Direction             : Outbound
Action                : Block
EdgeTraversalPolicy   : Block
LooseSourceMapping    : False
LocalOnlyMapping      : False
Owner                 :
PrimaryStatus         : OK
Status                : The rule was parsed successfully from the store. (65536)
EnforcementStatus     : NotApplicable
PolicyStoreSource     : PersistentStore
PolicyStoreSourceType : Local
```

After identifying the rule that caused the drop, the network admin can modify or disable the rule to allow the traffic they want through one of the available tools. The network admin can find the rule in the UI with the rule's *DisplayName*.

> ⓘ **Note**
>
> Firewall rules from Mobile Device Management (MDM) store cannot be searched using the Windows Firewall UI. Additionally, the above method doesn't work when the *Filter Origin* is one of the default block filters, as they don't correspond to any firewall rules.

Firewall default block filters

AppContainer loopback

Network drop events from the AppContainer loopback block filter origin occur when localhost loopback isn't enabled properly for the Universal Windows Platform (UWP) app:

- To enable localhost loopback in a local debugging environment, see Communicating with localhost

- To enable localhost loopback for a published app that requires loopback access to communicate with another UWP or packaged Win32 app, see uap4:LoopbackAccessRules

Boot time default

Network drop events from the boot time default block filter origin occur when the computer is booting up and the firewall service isn't yet running. Services need to create a boot time allow filter to allow the traffic. It should be noted that it's not possible to add boot time filters through firewall rules.

Quarantine default

Network drops from the quarantine default *block filter* occur when the interface is temporarily quarantined by the Firewall service. The firewall service quarantines an interface when it detects a change on the network, and based on several other factors, the firewall service might put the interface in quarantine as a safeguard. When an interface quarantined, the quarantine default *block filter* blocks any new non-loopback inbound connections.

Run the following PowerShell command to generate more information about the interface:

```
Powershell

Get-NetIPInterface -InterfaceIndex <Interface Index>
```

To learn more about the quarantine feature, see Quarantine behavior.

> **Note**
>
> Quarantine-related packet drops are often transient and signify nothing more than a network change on the interface.

Query user default

Network packet drops from query user default block filters occur when there's no explicit rule created to allow an inbound connection for the packet. When an application binds to a socket but doesn't have a corresponding inbound rule to allow packets on that port, Windows generates a pop-up for the user to allow or deny the app to receive

packets on the available network categories. If the user selects to deny the connection in the pop-up, subsequent inbound packets to the app will be dropped. To resolve the drops:

1. Create an inbound firewall rule to allow the packet for this application. The rule allows the packet to bypass any query user default block filters
2. Delete any block query user rules that may have been auto generated by the firewall service

To generate a list of all the query user block rules, you can run the following PowerShell command:

Powershell

```
Get-NetFirewallRule | Where {$_.Name -like "*Query User*"}
```

The query user pop-up feature is enabled by default. To disable the query user pop-up, you can run the following command in administrative command prompt:

Windows Command Prompt

```
Netsh set allprofiles inboundusernotification disable
```

Or in PowerShell:

Powershell

```
Set-NetFirewallProfile -NotifyOnListen False
```

Stealth

Network drops from stealth filters are typically made to prevent port scanning.

To disable stealth-mode, see Disable stealth mode in Windows.

UWP default

Network drops from Universal Windows Platform (UWP) default inbound/outbound block filters are often caused by the UWP app not being configured correctly (that is, the UWP app is missing the correct capability tokens or loopback isn't enabled) or the private range is configured incorrectly.

For more information on how to debug drops caused by UWP default block filters, see Troubleshooting UWP App Connectivity Issues.

WSH default

Network drops from Windows Service Hardening (WSH) default filters indicate that there wasn't an explicit Windows Service Hardening allow rule to allow network traffic for the protected service. The service owner needs to configure *allow rules* for the service if the block isn't expected.

Feedback

Was this page helpful? Yes No

Provide product feedback

Quarantine behavior

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

One of the security challenges that network admins face is configuring a device properly after a network change.

Network changes can happen frequently. The operations required to recategorize the network after a change, and apply the correct security policies on a device, are nontrivial and might require considerable CPU time. This requirement by operations is especially true for devices that are part of a domain. The delay in applying security policies during network recategorization can be exploited for vulnerabilities.

To counter the potential exploitation, Windows Firewall quarantines an interface until the system successfully recategorizes the network, and Windows Filtering Platform (WFP) has the correct filters applied for the updated interface configuration. During quarantine, all new inbound connections without exceptions are blocked.

This document describes the quarantine feature and explains why the application traffic could be blocked by quarantine.

Quarantine filters

The quarantine feature creates filters that can be split into three categories:

- Quarantine default inbound block filter
- Quarantine default exception filters
- Interface unquarantine filters

These filters are added in the `FWPM_SUBLAYER_MPSSVC_QUARANTINE` sublayer and these layers are:

1. `FWPM_LAYER_ALE_AUTH_CONNECT_V4`
2. `FWPM_LAYER_ALE_AUTH_CONNECT_V6`
3. `FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4`
4. `FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V6`

> ⓘ **Note**

Any firewall rules added by policy settings don't affect the filters in the quarantine sublayer. Filters from firewall rules are added in the `FWPM_SUBLAYER_MPSSVC_WF` sublayer. In other words, you can't add your own exception filters to prevent packets from being evaluated by quarantine filters.

For more information about WFP layers and sublayers, see WFP Operation.

Quarantine default inbound block filter

The *quarantine default inbound block filter* blocks any new nonloopback inbound connections, unless the packet isn't explicitly permitted by another filter in the quarantine sublayer.

Quarantine default exception filters

When the interface is in quarantine state, the quarantine default exception filters permit new inbound connections given that they meet the conditions of an exception filter. One example of the exception filters is the quarantine default inbound loopback exception filter. This exception filter allows all loopback packets when the interface is in quarantine state.

Interface unquarantine filter

The interface unquarantine filters allow all nonloopback packets if the interface is successfully categorized.

Quarantine flow

The following events describe the general flow of quarantine:

1. There's some change on the current network interface
2. The interface unquarantine filters don't permit new inbound connections. The interface is now in quarantine state
3. All nonloopback inbound connections are either permitted by quarantine default exception filters or dropped by the quarantine default inbound block filter
4. The WFP filters applicable to the old interface state are removed
5. The WFP filters applicable to the new interface state are added, which include the unquarantine filters for this interface. These filters are updated to match the interface's current state

6. The interface has now exited quarantine state as the interface unquarantine filters permit any new nonloopback packets

Quarantine diagnostics

There are two methods of identifying packet drops from the quarantine default inbound block filter.

Given that the network connectivity issue is reproducible, diagnostic traces can be collected by running the following in an administrative command prompt:

Windows Command Prompt

```
Netsh wfp cap start
<Reproduce network connectivity issue>
Netsh wfp cap stop
```

These commands generate a wfpdiag.cab. Inside the .cab exists a wfpdiag.xml, which contains drop `netEvents` and filters that existed during that reproduction.

Inside the wfpdiag.xml, search for `netEvents` that have `FWPM_NET_EVENT_TYPE_CLASSIFY_DROP` as the `netEvent` type. To find the relevant drop events, search for the drop events with matching destination IP address, package SID, or application ID name.

The characters in the application ID name are separated by periods:

XML

```
 <asString>
\\.d.e.v.i.c.e.\\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\\.w.i.n.d.o.w.s.\\.s.y.s.t.
e.m.3.2.\\.s.v.c.h.o.s.t...e.x.e... </asString>
```

The `netEvent` contains more information about the dropped packet, including information about its capabilities, the filter that dropped the packet, and much more.

If the filter that dropped that packet was by the quarantine default inbound block filter, then the drop `netEvent` contains `filterOrigin` as `Quarantine Default`.

The following code is a sample `netEvent` with `filterOrigin` as `Quarantine Default`.

XML

```
<netEvent>
    <header>
        <timeStamp>2020-10-07T01:03:56.281Z</timeStamp>
        <flags numItems="9">
            <item>FWPM_NET_EVENT_FLAG_IP_PROTOCOL_SET</item>
            <item>FWPM_NET_EVENT_FLAG_LOCAL_ADDR_SET</item>
            <item>FWPM_NET_EVENT_FLAG_REMOTE_ADDR_SET</item>
            <item>FWPM_NET_EVENT_FLAG_LOCAL_PORT_SET</item>
            <item>FWPM_NET_EVENT_FLAG_REMOTE_PORT_SET</item>
            <item>FWPM_NET_EVENT_FLAG_APP_ID_SET</item>
            <item>FWPM_NET_EVENT_FLAG_USER_ID_SET</item>
            <item>FWPM_NET_EVENT_FLAG_IP_VERSION_SET</item>
            <item>FWPM_NET_EVENT_FLAG_PACKAGE_ID_SET</item>
        </flags>
        <ipVersion>FWP_IP_VERSION_V4</ipVersion>
        <ipProtocol>17</ipProtocol>
        <localAddrV4>255.255.255.255</localAddrV4>
        <remoteAddrV4>10.195.33.252</remoteAddrV4>
        <localPort>21</localPort>
        <remotePort>61706</remotePort>
        <scopeId>0</scopeId>
        <appId>

<data>5c00640065006d00330032005c00730076006300680006f00730074002e006500780065
000000</data>

<asString>\.d.e.v.i.c.e.\.h.a.r.d.d.i.s.k.v.o.l.u.m.e.1.\.w.i.n.d.o.w.s.\.s.
y.s.t.e.m.3.2.\.s.v.c.h.o.s.t...e.x.e...</asString>
        </appId>
        <userId>S-1-5-19</userId>
        <addressFamily>FWP_AF_INET</addressFamily>
        <packageSid>S-1-0-0</packageSid>
        <enterpriseId/>
        <policyFlags>0</policyFlags>
        <effectiveName/>
    </header>
    <type>FWPM_NET_EVENT_TYPE_CLASSIFY_DROP</type>
    <classifyDrop>
        <filterId>66241</filterId>
        <layerId>44</layerId>
        <reauthReason>0</reauthReason>
        <originalProfile>0</originalProfile>
        <currentProfile>0</currentProfile>
        <msFwpDirection>MS_FWP_DIRECTION_OUT</msFwpDirection>
        <isLoopback>false</isLoopback>
        <vSwitchId/>
        <vSwitchSourcePort>0</vSwitchSourcePort>
        <vSwitchDestinationPort>0</vSwitchDestinationPort>
    </classifyDrop>
    <internalFields>
        <internalFlags numItems="1">
            <item>FWPM_NET_EVENT_INTERNAL_FLAG_FILTER_ORIGIN_SET</item>
        </internalFlags>
        <capabilities/>
```

```
        <fqbnVersion>0</fqbnVersion>
        <fqbnName/>
        <terminatingFiltersInfo numItems="3">
            <item>
                <filterId>66241</filterId>

<subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_QUARANTINE</subLayer>
                <actionType>FWP_ACTION_BLOCK</actionType>
            </item>
            <item>
                <filterId>74045</filterId>
                <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WSH</subLayer>
                <actionType>FWP_ACTION_BLOCK</actionType>
            </item>
            <item>
                <filterId>73602</filterId>
                <subLayer>FWPP_SUBLAYER_INTERNAL_FIREWALL_WF</subLayer>
                <actionType>FWP_ACTION_BLOCK</actionType>
            </item>
        </terminatingFiltersInfo>
        <filterOrigin>Quarantine Default</filterOrigin>
         <interfaceIndex>5</interfaceIndex>
    </internalFields>
</netEvent>
```

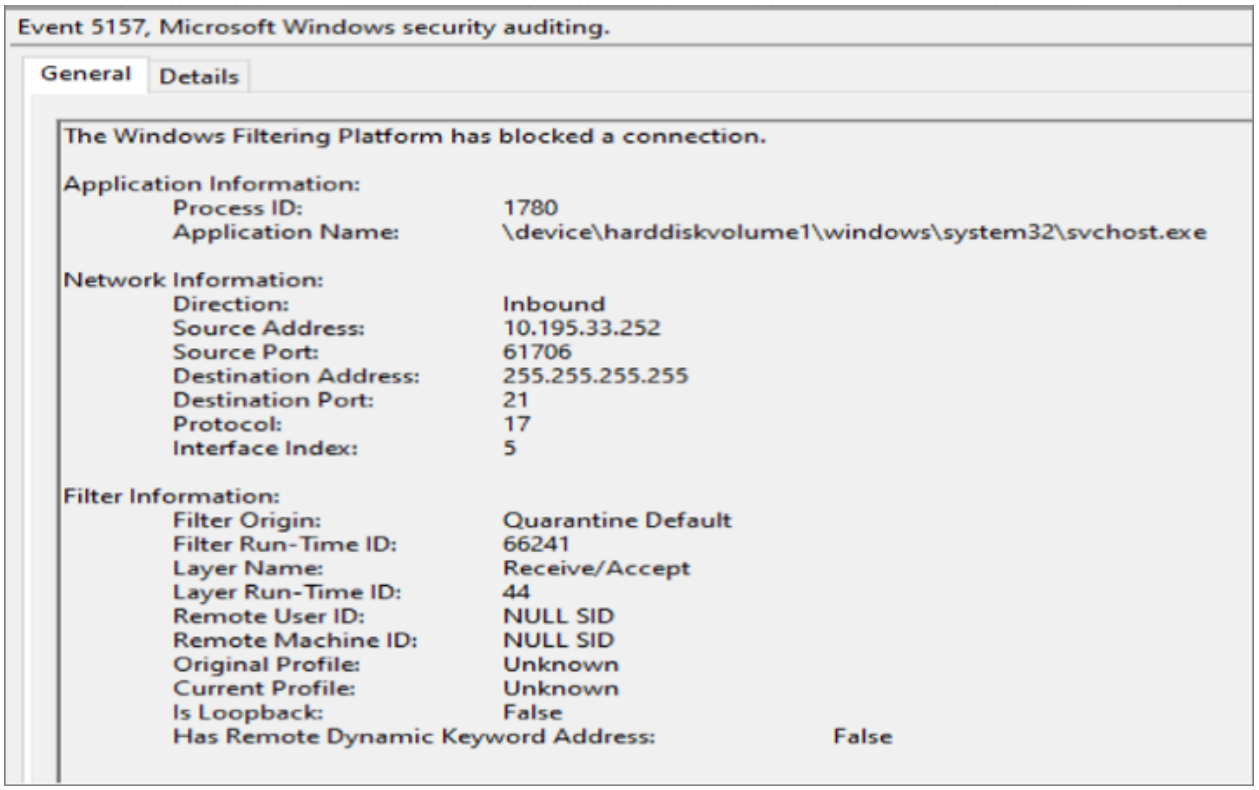
Alternatively, If the Filtering Platform Connection failure auditing is enabled, the drop event is logged in Windows Event Viewer.

To enable Filtering Platform Connection audits, run the following command in an administrative command prompt:

Windows Command Prompt

```
Auditpol /set /category:"System" /SubCategory:"Filtering Platform
Connection" /success:enable /failure:enable
```

Sample drop audit with `filterOrigin` as `Quarantine Default`.



Once the drop's filter origin has been identified as the quarantine default inbound block filter, the interface should be further investigated. To find the relevant interface, use the `InterfaceIndex` value from the `netEvent` or event audit in the following PowerShell command to generate more information about the interface:

Powershell

```
Get-NetIPInterface -InterfaceIndex <Interface Index>
Get-NetIPInterface -InterfaceIndex 5
```

With the help of the interface name, event viewer can be searched for any interface related changes.

To enable more networking audit events, see Enable IPsec and Windows Firewall Audit Events.

Packet drops from the quarantine default inbound block filter are often transient and don't signify anything more than a network change on the interface.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows VPN technical guide

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This guide walks you through the decisions to make for Windows clients in your organization's VPN solution, and how to configure your devices. This guide references the VPNv2 Configuration Service Provider (CSP) and provides mobile device management (MDM) configuration instructions using Microsoft Intune.

To create a Windows VPN device configuration profile see: Windows device settings to add VPN connections using Intune.

> **Note**
>
> This guide does not explain server deployment.

Windows edition and licensing requirements

The following table lists the Windows editions that support Virtual private network (VPN):

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Virtual private network (VPN) license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

In this guide

Expand table

Article	Description
VPN connection types	Select a VPN client and tunneling protocol
VPN routing decisions	Choose between split tunnel and force tunnel configuration
VPN authentication options	Select a method for Extensible Authentication Protocol (EAP) authentication.
VPN and conditional access	Use Microsoft Entra policy evaluation to set access policies for VPN connections.
VPN name resolution	Decide how name resolution should work
VPN auto-triggered profile options	Set a VPN profile to connect automatically by app or by name, to be "always on", and to not trigger VPN on trusted networks
VPN security features	Configure traffic filtering, connect a VPN profile to Windows Information Protection (WIP), and more
VPN profile options	Combine settings into single VPN profile using XML

Learn more

- Create VPN profiles to connect to VPN servers in Intune

Feedback

Was this page helpful? Yes No

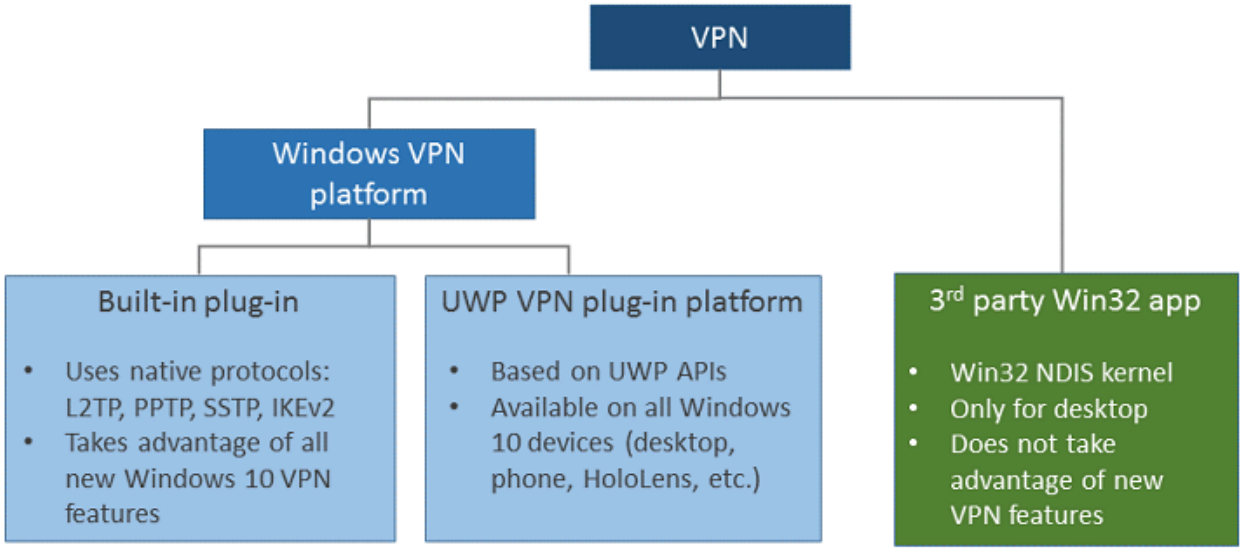
Provide product feedback

VPN connection types

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

VPNs are point-to-point connections across a private or public network, like the Internet. A VPN client uses special TCP/IP or UDP-based protocols, called *tunneling protocols*, to make a virtual call to a virtual port on a VPN server. In a typical VPN deployment, a client initiates a virtual point-to-point connection to a remote access server over the Internet. The remote access server answers the call, authenticates the caller, and transfers data between the VPN client and the organization's private network.

There are many options for VPN clients. In Windows, the built-in plug-in and the Universal Windows Platform (UWP) VPN plug-in platform are built on top of the Windows VPN platform. This article focuses on the Windows VPN platform clients and the features that can be configured.



Built-in VPN client

Tunneling protocols:

- Internet Key Exchange version 2 (IKEv2): configure the IPsec/IKE tunnel cryptographic properties using the **Cryptography Suite** setting in the VPNv2 Configuration Service Provider (CSP).
- L2TP: L2TP with pre-shared key (PSK) authentication can be configured using the **L2tpPsk** setting in the VPNv2 CSP.
- PPTP

- SSTP: SSTP can't be configured using MDM, but it's one of the protocols attempted in the **Automatic** option

  > ⓘ **Note**
  >
  > When a VPN plug-in is used, the adapter will be listed as an SSTP adapter, even though the VPN protocol used is the plug-in's protocol.

- Automatic: the **Automatic** option means that the device tries each of the built-in tunneling protocols until one succeeds. It attempts from most secure to least secure. Configure **Automatic** for the **NativeProtocolType** setting in the VPNv2 CSP.

Universal Windows Platform VPN plug-in

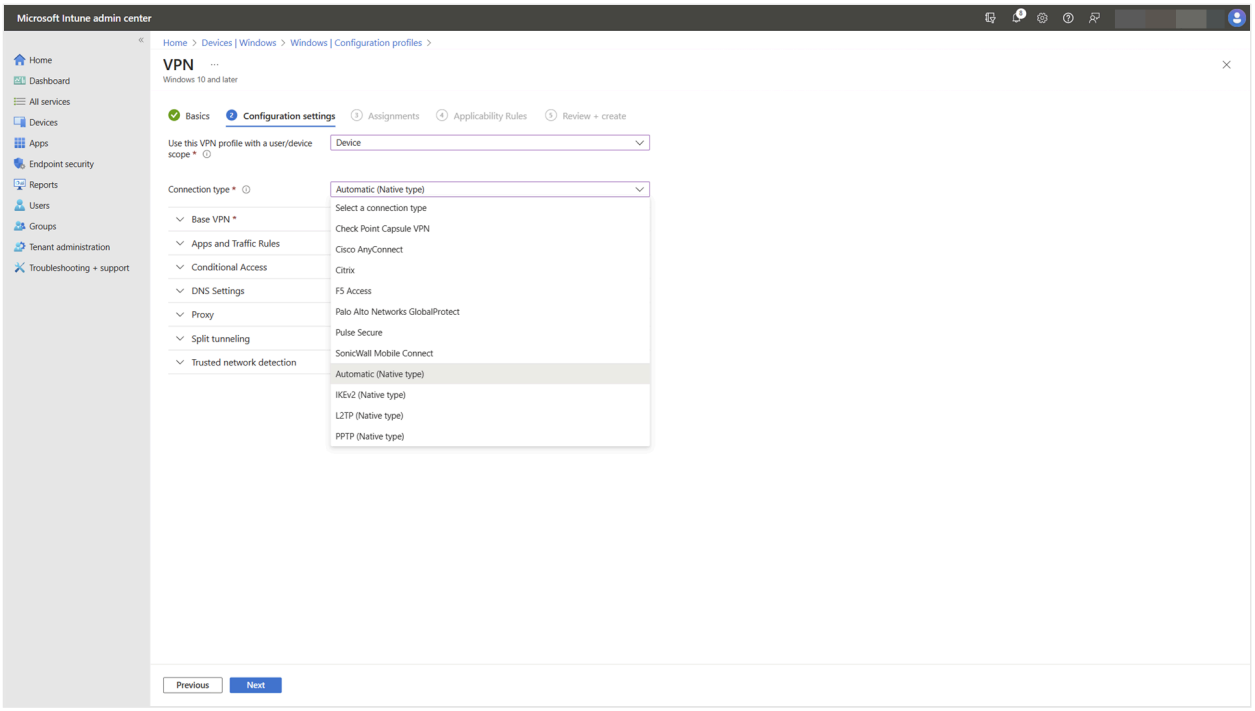
Using the UWP platform, non-Microsoft VPN providers can create app-containerized plug-ins using WinRT APIs, eliminating the complexity and problems often associated with writing to system-level drivers.

There are many Universal Windows Platform VPN applications, such as Pulse Secure, Cisco AnyConnect, F5 Access, SonicWall Mobile Connect, and Check Point Capsule. If you want to use a UWP VPN plug-in, work with your vendor for any custom settings needed to configure your VPN solution.

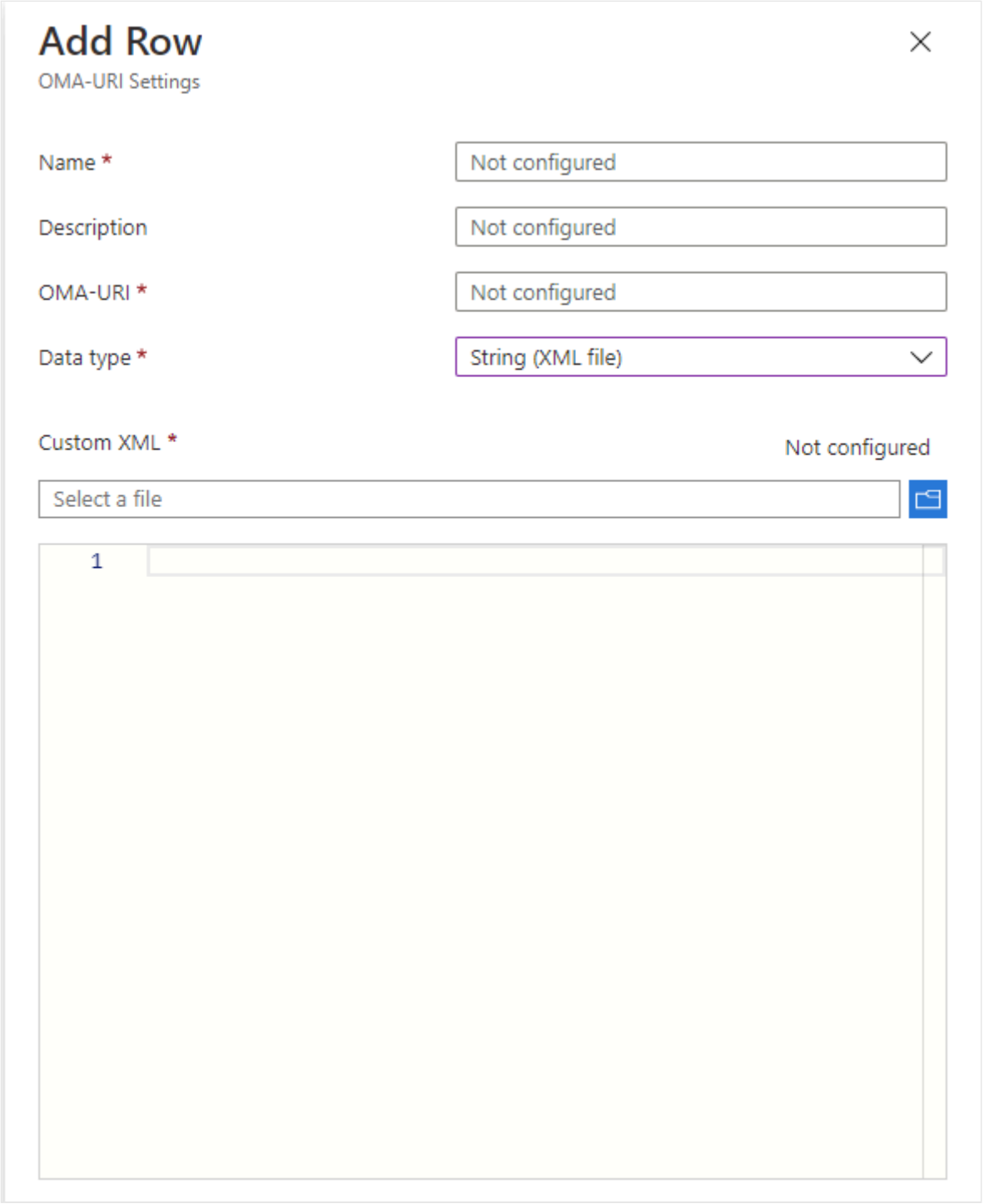
Configure connection type

See VPN profile options and VPNv2 CSP for XML configuration.

The following image shows connection options in a VPN Profile configuration policy using Microsoft Intune:



In Intune, you can also include custom XML for non-Microsoft plug-in profiles:



Feedback

Was this page helpful? Yes No

Provide product feedback

VPN routing decisions

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

Network routes are required for the stack to understand which interface to use for outbound traffic. One of the most important decision points for VPN configuration is whether you want to send all the data through VPN (*force tunnel*) or only some data through the VPN (*split tunnel*). The decision impacts the configuration, capacity planning, and security expectations from the connection.

Split tunnel configuration

In a split tunnel configuration, routes can be specified to go over VPN and all other traffic will go over the physical interface.

Routes can be configured using the `VPNv2/<ProfileName>/RouteList` setting in the VPNv2 Configuration Service Provider (CSP).

For each route item in the list, you can configure the following options:

- **Address**: `VPNv2/<ProfileName>/RouteList/<routeRowId>/Address`
- **Prefix size**: `VPNv2/<ProfileName>/RouteList/<routeRowId>/Prefix`
- **Exclusion route**: V`VPNv2/<ProfileName>/RouteList/<routeRowId>/ExclusionRoute`

With Windows VPN, you can specify exclusion routes that shouldn't go over the physical interface.

Routes can also be added at connect time through the server for UWP VPN apps.

Force tunnel configuration

In a force tunnel configuration, all traffic will go over VPN. Force tunnel is the default configuration, and takes effect when no routes are specified.

The only implication of force tunnel is the manipulation of routing entries: VPN V4 and V6 default routes (for example *0.0.0.0/0*) are added to the routing table with a lower metric than ones for other interfaces. This configuration sends traffic through the VPN as long as there isn't a specific route on the physical interface:

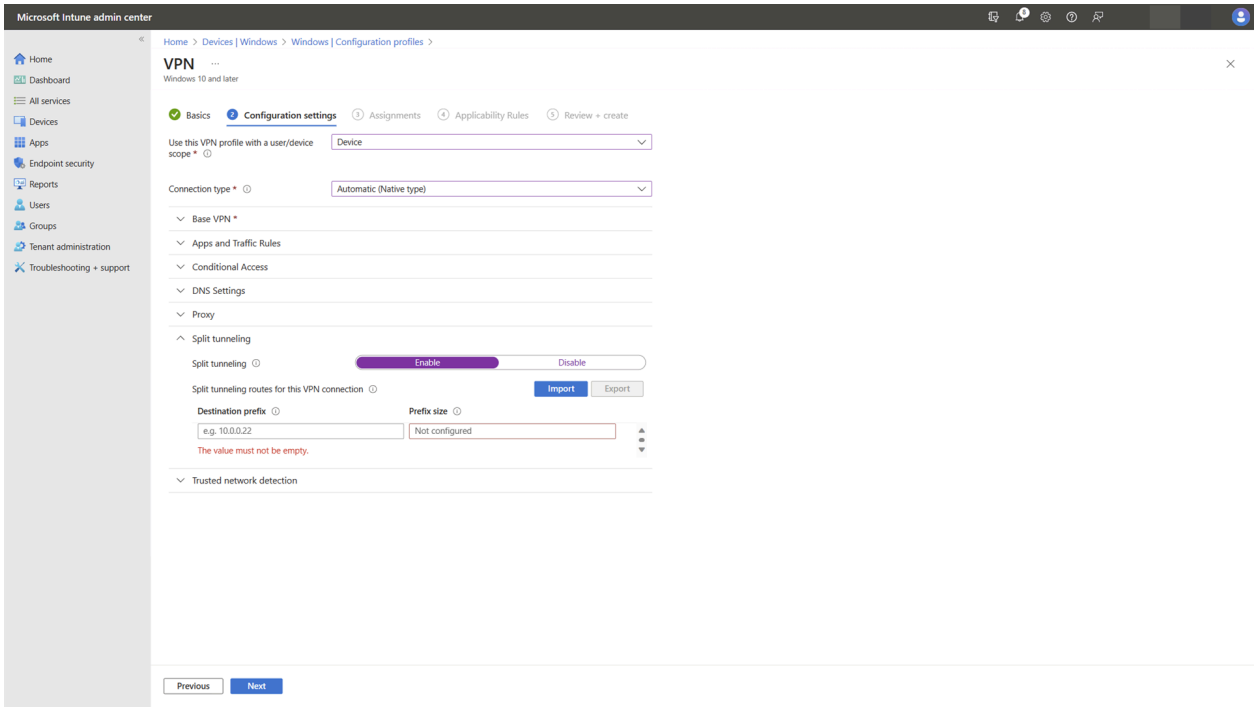
- For built-in VPN, the decision is controlled using the MDM setting `VPNv2/ProfileName/NativeProfile/RoutingPolicyType`

- For a UWP VPN plug-in, the app controls the property. If the VPN plug-in indicates the default route for IPv4 and IPv6 as the only two Inclusion routes, the VPN platform marks the connection as Force Tunneled

Configure routing

See VPN profile options and VPNv2 CSP for XML configuration.

When you configure a VPN profile in Microsoft Intune, you can enable split tunnel configuration:



Once enabled, you can add the routes that should use the VPN connection.

Feedback

Was this page helpful? Yes No

Provide product feedback

VPN authentication options

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

In addition to older and less-secure password-based authentication methods (which should be avoided), the built-in VPN solution uses Extensible Authentication Protocol (EAP) to provide secure authentication using both user name and password, and certificate-based methods. You can only configure EAP-based authentication if you select a built-in VPN type (IKEv2, L2TP, PPTP or Automatic).

Windows supports a number of EAP authentication methods.

- EAP-Microsoft Challenge Handshake Authentication Protocol version 2 (EAP-MSCHAPv2):
  - User name and password authentication
  - Winlogon credentials - can specify authentication with computer sign-in credentials

- EAP-Transport Layer Security (EAP-TLS):

  - Supports the following types of certificate authentication:
    - Certificate with keys in the software Key Storage Provider (KSP)
    - Certificate with keys in Trusted Platform Module (TPM) KSP
    - Smart card certificates
    - Windows Hello for Business certificate

  - Certificate filtering:
    - Certificate filtering can be enabled to search for a particular certificate to use to authenticate with
    - Filtering can be Issuer-based or extended key usage (EKU)-based

  - Server validation - with TLS, server validation can be toggled on or off:
    - Server name - specify the server to validate
    - Server certificate - trusted root certificate to validate the server
    - Notification - specify if the user should get a notification asking whether to trust the server or not

- Protected Extensible Authentication Protocol (PEAP):

  - Server validation - with PEAP, server validation can be toggled on or off:
    - Server name - specify the server to validate
    - Server certificate - trusted root certificate to validate the server
    - Notification - specify if the user should get a notification asking whether to trust the server or not

    - Inner method - the outer method creates a secure tunnel inside while the inner method is used to complete the authentication:
        - EAP-MSCHAPv2
        - EAP-TLS
    - Fast Reconnect: reduces the delay between an authentication request by a client and the response by the Network Policy Server (NPS) or other Remote Authentication Dial-in User Service (RADIUS) server. This reduces resource requirements for both client and server, and minimizes the number of times that users are prompted for credentials.
    - Cryptobinding: By deriving and exchanging values from the PEAP phase 1 key material (**Tunnel Key**) and from the PEAP phase 2 inner EAP method key material (**Inner Session Key**), it's possible to prove that the two authentications terminate at the same two entities (PEAP peer and PEAP server). This process, termed "cryptobinding", is used to protect the PEAP negotiation against "Man in the Middle" attacks.
- Tunneled Transport Layer Security (TTLS)
    - Inner method
        - Non-EAP
            - Password Authentication Protocol (PAP)
            - CHAP
            - MSCHAP
            - MSCHAPv2
        - EAP
            - MSCHAPv2
            - TLS
    - Server validation: in TTLS, the server must be validated. The following can be configured:
        - Server name
        - Trusted root certificate for server certificate
        - Whether there should be a server validation notification

For a UWP VPN plug-in, the app vendor controls the authentication method to be used. The following credential types can be used:

- Smart card
- Certificate
- Windows Hello for Business
- User name and password
- One-time password

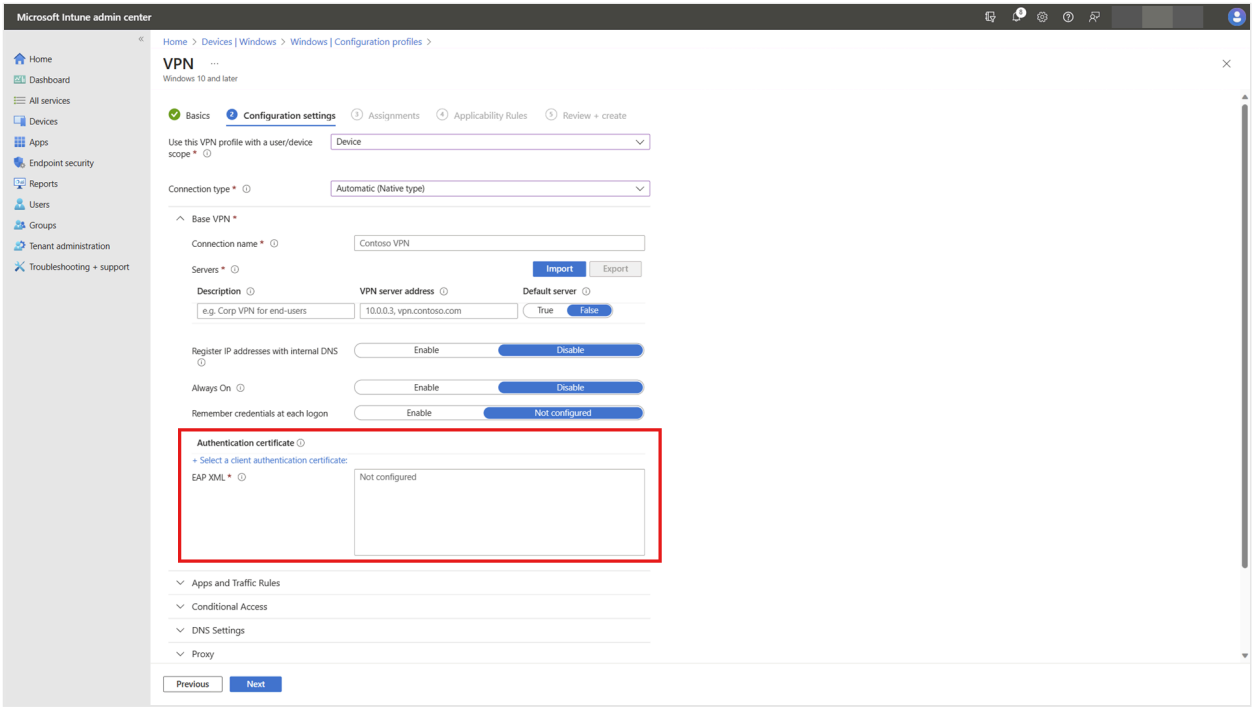
- Custom credential type

Configure authentication

See EAP configuration for EAP XML configuration.

> **Note**
>
> To configure Windows Hello for Business authentication, follow the steps in EAP configuration to create a smart card certificate. Learn more about Windows Hello for Business..

The following image shows the field for EAP XML in a Microsoft Intune VPN profile. The EAP XML field only appears when you select a built-in connection type (automatic, IKEv2, L2TP, PPTP).



Feedback

Was this page helpful? Yes No

Provide product feedback

VPN and conditional access

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

The VPN client is now able to integrate with the cloud-based Conditional Access Platform to provide a device compliance option for remote clients. Conditional Access is a policy-based evaluation engine that lets you create access rules for any Microsoft Entra connected application.

> ⓘ **Note**
>
> Conditional Access is a Microsoft Entra ID P1 or P2 feature.

Conditional Access Platform components used for Device Compliance include the following cloud-based services:

- Conditional Access Framework
- Microsoft Entra Connect Health
- Windows Health Attestation Service (optional)
- Microsoft Entra Certificate Authority - It's a requirement that the client certificate used for the cloud-based device compliance solution be issued by a Microsoft Entra ID-based Certificate Authority (CA). A Microsoft Entra CA is essentially a mini-CA cloud tenant in Azure. The Microsoft Entra CA can't be configured as part of an on-premises Enterprise CA. See also Always On VPN deployment for Windows Server and Windows 10.
- Microsoft Entra ID-issued short-lived certificates - When a VPN connection attempt is made, the Microsoft Entra Token Broker on the local device communicates with Microsoft Entra ID, which then checks for health based on compliance rules. If compliant, Microsoft Entra ID sends back a short-lived certificate that is used to authenticate the VPN. Certificate authentication methods such as EAP-TLS can be used. When the client reconnects and determines that the certificate has expired, the client will again check with Microsoft Entra ID for health validation before a new certificate is issued.
- Microsoft Intune device compliance policies: Cloud-based device compliance uses Microsoft Intune Compliance Policies, which are capable of querying the device state and define compliance rules for the following, among other things.
  - Antivirus status
  - Auto-update status and update compliance
  - Password policy compliance
  - Encryption compliance
  - Device health attestation state (validated against attestation service after query)

The following client-side components are also required:

- HealthAttestation Configuration Service Provider (CSP)
- VPNv2 CSP DeviceCompliance node settings
- Trusted Platform Module (TPM)

VPN device compliance

At this time, the Microsoft Entra certificates issued to users don't contain a CRL Distribution Point (CDP) and aren't suitable for Key Distribution Centers (KDCs) to issue Kerberos tokens. For users to gain access to on-premises resources such as files on a network share, client authentication certificates must be deployed to the Windows profiles of the users, and their VPNv2 profiles must contain the `<SSO>` section.

Server-side infrastructure requirements to support VPN device compliance include:

- The VPN server should be configured for certificate authentication.
- The VPN server should trust the tenant-specific Microsoft Entra CA.
- For client access using Kerberos/NTLM, a domain-trusted certificate is deployed to the client device and is configured to be used for single sign-on (SSO).

After the server side is set up, VPN admins can add the policy settings for conditional access to the VPN profile using the VPNv2 DeviceCompliance node.

Two client-side configuration service providers are leveraged for VPN device compliance.

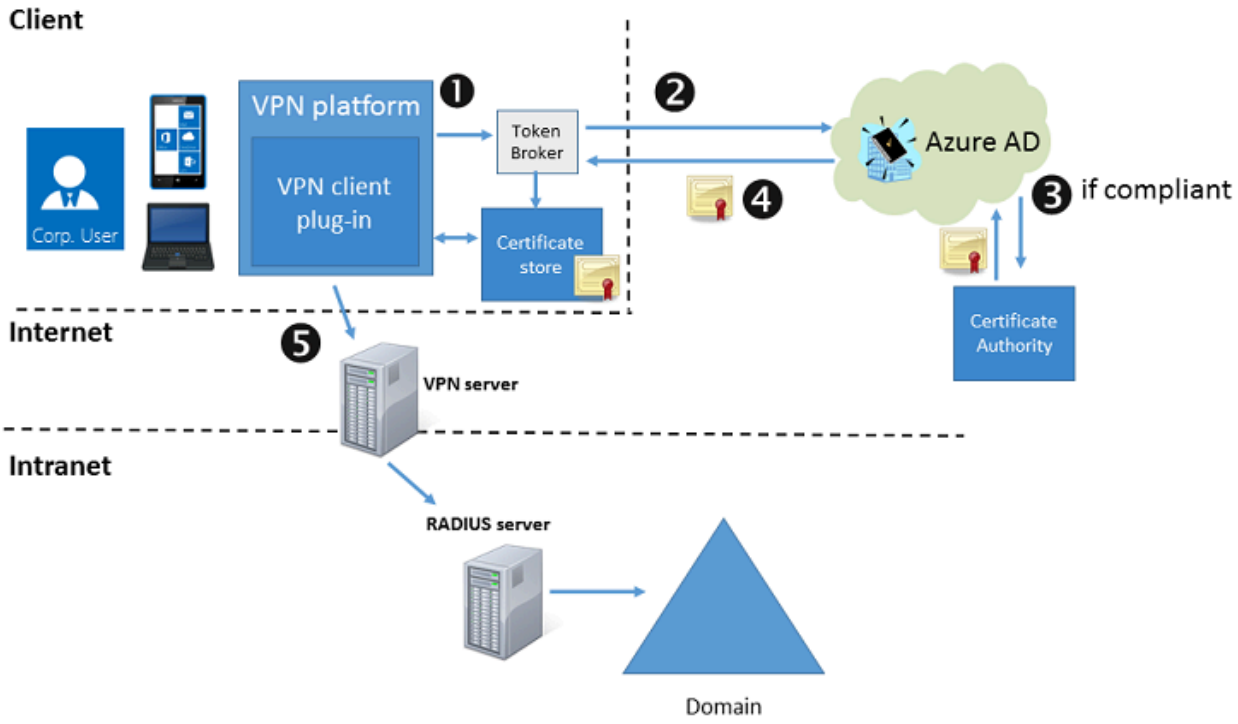
- VPNv2 CSP DeviceCompliance settings:
  - **Enabled**: enables the Device Compliance flow from the client. If marked as **true**, the VPN client attempts to communicate with Microsoft Entra ID to get a certificate to use for authentication. The VPN should be set up to use certificate authentication and the VPN server must trust the server returned by Microsoft Entra ID.
  - **Sso**: entries under SSO should be used to direct the VPN client to use a certificate other than the VPN authentication certificate when accessing resources that require Kerberos authentication.
  - **Sso/Enabled**: if this field is set to **true**, the VPN client looks for a separate certificate for Kerberos authentication.
  - **Sso/IssuerHash**: hashes for the VPN client to look for the correct certificate for Kerberos authentication.
  - **Sso/Eku**: comma-separated list of extended key usage (EKU) extensions for the VPN client to look for the correct certificate for Kerberos authentication.

- HealthAttestation CSP (not a requirement) - functions performed by the HealthAttestation CSP include:
  - Collects TPM data used to verify health states
  - Forwards the data to the Health Attestation Service (HAS)
  - Provisions the Health Attestation Certificate received from the HAS
  - Upon request, forward the Health Attestation Certificate (received from HAS) and related runtime information to the MDM server for verification

> **Note**
>
> It's required that certificates used for obtaining Kerberos tickets to be issued from an on-premises CA, and that SSO to be enabled in the user's VPN profile. This allows the user to access on-premises resources. In the case of Microsoft Entra joined devices (not hybrid joined devices), if the user certificate issued by the on-premises CA has the user UPN from Microsoft Entra in Subject and SAN (Subject Alternative Name), the VPN profile must be modified to ensure that the client doesn't cache the credentials used for VPN authentication. To do this, after deploying the VPN profile to the client, modify the *Rasphone.pbk* on the client by changing the entry **UseRasCredentials** from 1 (default) to 0 (zero).

Client connection flow

The VPN client side connection flow works as follows:



When a VPNv2 Profile is configured with <DeviceCompliance> <Enabled>true</Enabled> the VPN client uses this connection flow:

1. The VPN client calls into Windows 10 or Windows 11 Microsoft Entra Token Broker, identifying itself as a VPN client.
2. The Microsoft Entra Token Broker authenticates to Microsoft Entra ID and provides it with information about the device trying to connect. The Microsoft Entra Server checks if the device is in compliance with the policies.
3. If compliant, Microsoft Entra ID requests a short-lived certificate.
4. Microsoft Entra ID pushes down a short-lived certificate to the Certificate Store via the Token Broker. The Token Broker then returns control back over to the VPN client for further connection processing.
5. The VPN client uses the Microsoft Entra ID-issued certificate to authenticate with the VPN server.

Configure conditional access

See VPN profile options and VPNv2 CSP for XML configuration.

Learn more about Conditional Access and Microsoft Entra Health

- Microsoft Entra Conditional Access
- Getting started with Microsoft Entra Conditional Access
- Control the health of Windows devices
- Tip of the Day: The Conditional Access Framework and Device Compliance for VPN (Part 1)
- Tip of the Day: The Conditional Access Framework and Device Compliance for VPN (Part 2)
- Tip of the Day: The Conditional Access Framework and Device Compliance for VPN (Part 3)
- Tip of the Day: The Conditional Access Framework and Device Compliance for VPN (Part 4)

Feedback

Was this page helpful? Yes No

Provide product feedback

VPN name resolution

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

When the VPN client establishes a connection, it receives an IP address and, optionally, the IP address of one or more DNS servers.

The name resolution setting in the VPN profile determines how name resolution works on the system when the VPN connection is established:

1. The network stack looks at the Name Resolution Policy table (NRPT) for any matches, and tries a resolution if a match is found
2. If no match is found, the DNS suffix on the most preferred interface based on the interface metric is appended to the name (if a short name is used). A DNS query is sent to the preferred interface
3. If the query times out, the DNS suffix search list is used in order and DNS queries are sent on all interfaces

Name Resolution Policy table (NRPT)

The NRPT is a table of namespaces that determines the DNS client's behavior when issuing name resolution queries and processing responses. It's the first place that the stack will look after the DNSCache.

There are three types of name matches that can set up for NRPT:

- Fully qualified domain name (FQDN) that can be used for direct matching to a name
- Suffix match results in either a comparison of suffixes (for FQDN resolution) or the appending of the suffix (if using short name)
- Any resolution should attempt to first resolve with the proxy server/DNS server with this entry

NRPT is set using the `VPNv2/<ProfileName>/DomainNameInformationList` node of the VPNv2 CSP. You can use the same node to configure a Web proxy server or DNS.

To learn more about NRPT, see Introduction to the NRPT.

DNS suffix

The DNS suffix setting is used to configure the primary DNS suffix for the VPN interface and the suffix search list after the VPN connection is established.

Primary DNS suffix is set using the `VPNv2/<ProfileName>/DnsSuffix` node.

Learn more about primaryDNS suffix

Persistent name resolution rules

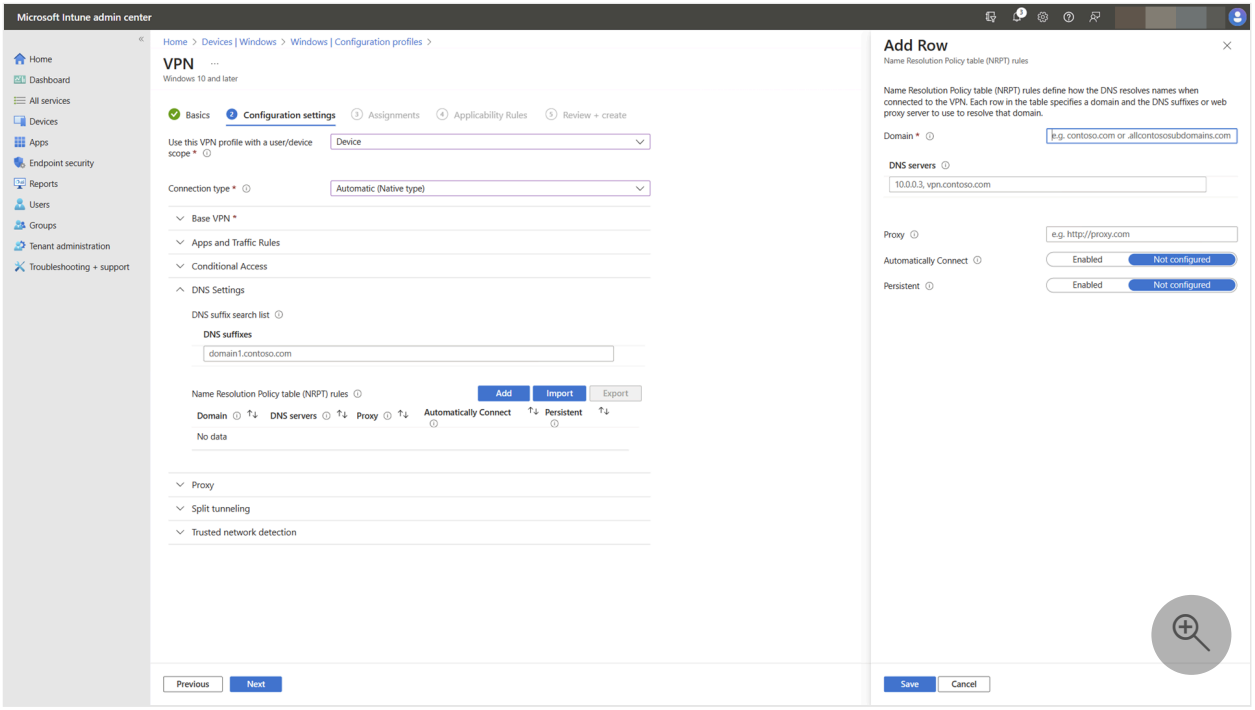
You can configure *persistent* name resolution rules. Name resolution for the specified items is done over the VPN.

Persistent name resolution is set using the `VPNv2/<ProfileName>/DomainNameInformationList/<dniRowId>/Persistent` node.

Configure name resolution

See VPN profile options and VPNv2 CSP for XML configuration.

The following image shows name resolution options in a VPN Profile configuration policy using Microsoft Intune.



The fields in **Add or edit DNS rule** in the Intune profile correspond to the XML settings shown in the following table.

Expand table

Field	XML
Name	VPNv2/*ProfileName*/DomainNameInformationList/*dniRowId*/DomainName

Field	XML
Servers (comma separated)	VPNv2/*ProfileName*/DomainNameInformationList/*dniRowId*/DnsServers
Proxy server	VPNv2/*ProfileName*/DomainNameInformationList/*dniRowId*/WebServers

Feedback

Was this page helpful? Yes No

Provide product feedback

VPN auto-triggered profile options

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

Windows can use different features to auto-trigger VPN, avoiding users to manually connect when VPN is needed to access necessary resources. There are three different types of auto-trigger rules:

- Application trigger
- Name-based trigger
- Always On

> ⓘ **Note**
>
> Auto-triggered VPN connections won't work if **Folder Redirection** for **AppData** is enabled. Either Folder Redirection for AppData must be disabled, or the auto-triggered VPN profile must be deployed in SYSTEM context, which changes the path to where the *rasphone.pbk* file is stored.

Application trigger

VPN profiles can be configured to automatically connect on the execution of certain applications:

- You can configure desktop or Universal Windows Platform (UWP) apps to trigger a VPN connection
- You can configure per-app VPN and specify traffic rules for each app

> ⓘ **Note**
>
> The app identifier for a desktop app is a file path. The app identifier for a UWP app is a package family name.
>
> Find a package family name (PFN) for per-app VPN configuration

For more information, see Traffic filters.

Name-based trigger

You can configure a domain name-based rule so that a specific domain name triggers the VPN connection.
Name-based auto-trigger can be configured using the `VPNv2/<ProfileName>/DomainNameInformationList/dniRowId/AutoTrigger` setting in the VPNv2 Configuration Service Provider (CSP).

There are four types of name-based triggers:

- Short name: for example, if *HRweb* is configured as a trigger, and the stack sees a DNS resolution request for *HRweb*, the VPN triggers
- Fully qualified domain name (FQDN): for example, if *HRweb.corp.contoso.com* is configured as a trigger, and the stack sees a DNS resolution request for *HRweb.corp.contoso.com*, the VPN triggers
- Suffix: for example, if *.corp.contoso.com* is configured as a trigger, and the stack sees a DNS resolution request with a matching suffix (such as *HRweb.corp.contoso.com*), the VPN triggers. For any short name resolution, VPN triggers, and the DNS servers are queried for the *<ShortName>.corp.contoso.com*
- All: if used, all DNS resolution triggers VPN

Always On

Always On is a Windows feature that enables the active VPN profile to connect automatically on the following triggers:

- User sign-in
- Network change
- Device screen on

When the trigger occurs, VPN tries to connect. If an error occurs, or any user input is needed, the user sees a toast notification for more interaction.

When a device has multiple profiles with Always On triggers, the user can specify the active profile in **Settings** > **Network & Internet** > **VPN** > **<VPN profile>** by selecting the **Let apps automatically use this VPN connection** checkbox. By default, the first MDM-configured profile is marked as **Active**. Devices with multiple users have the same restriction: only one profile, and therefore only one user, is able to use the Always On triggers.

Preserving user Always On preference

Another Windows feature is to preserve a user's Always On preference. If a user manually unchecks the **Connect automatically** checkbox, Windows remembers the user

preference for the profile name by adding the profile name to the registry value *AutoTriggerDisabledProfilesList*.

If a management tool removes or adds the same profile name back and set **AlwaysOn** to **true**, Windows doesn't check the box if the profile name exists in the following registry value, in order to preserve user preference.

Key: `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\RasMan\Config`
Value: `AutoTriggerDisabledProfilesList`
Type: `REG_MULTI_SZ`

Trusted network detection

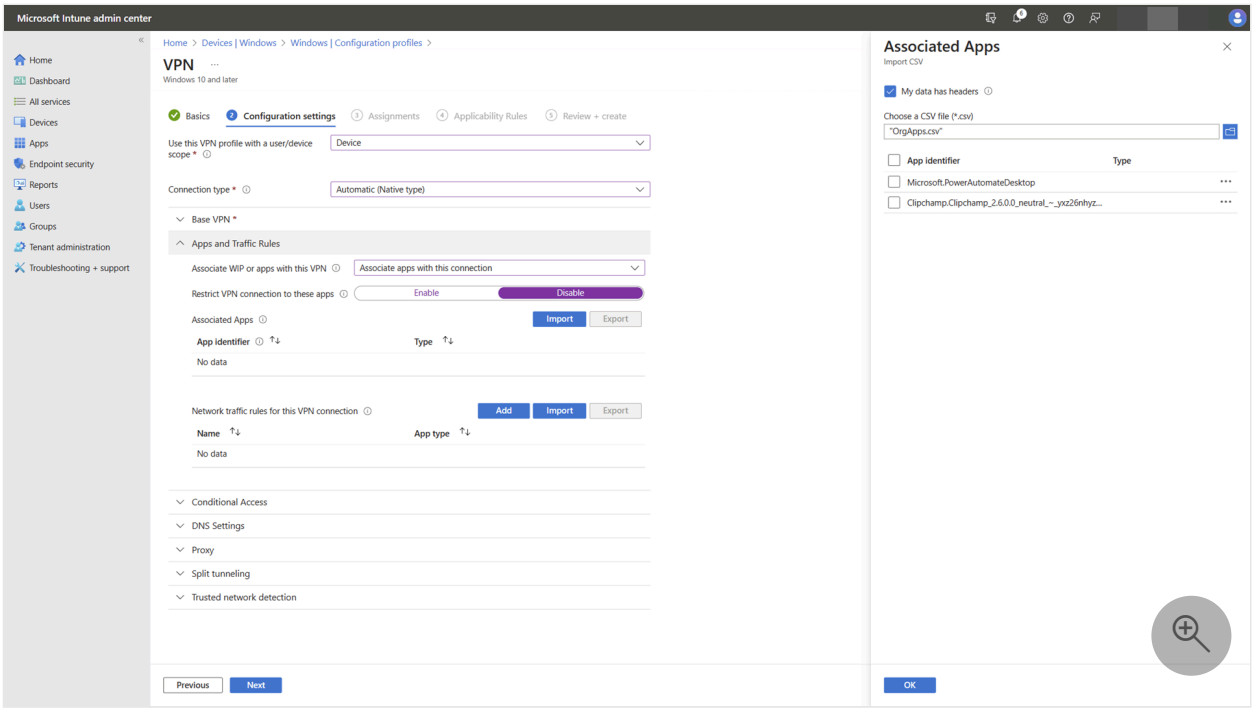
The **Trusted network detection** feature configures the VPN so that connection isn't triggered when a device is on a trusted network. To configure Trusted network detection, you must provide a list of DNS suffixes. The VPN stack verifies the network name of the physical interface connection profile: if it matches any of the suffixes configured in the list and the network is private or provisioned by MDM, then VPN doesn't trigger.

Trusted network detection can be configured using the `VPNv2/<ProfileName>/TrustedNetworkDetection` setting in the VPNv2 CSP.

Configure app-triggered VPN

See VPN profile options and VPNv2 CSP for XML configuration.

The following image shows associating apps to a VPN connection in a VPN Profile configuration policy using Microsoft Intune.



Feedback

Was this page helpful? Yes No

Provide product feedback

VPN security features

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

Hyper-V based containers and VPN

Windows supports different kinds of Hyper-V based containers, like Microsoft Defender Application Guard and Windows Sandbox. When you use a non-Microsoft VPN solution, the Hyper-V based containers may not be able to seamlessly connect to the internet, and configuration changes may be needed to resolve connectivity issues.

For example, read about the workaround for Cisco AnyConnect VPN: Cisco AnyConnect Secure Mobility Client Administrator Guide: Connectivity issues with VM-based subsystems.

Traffic Filters

Traffic Filters enables organizations to decide what traffic is allowed into the corporate network based on policy. IT admins can use Traffic Filters to apply interface-specific firewall rules to the VPN Interface.

There are two types of Traffic Filter rules:

- **App-based rules** consist of a list of applications that can be marked to only allow traffic originating from the apps to the VPN interface
- **Traffic-based rules** consist of 5-tuple policies (ports, addresses, protocol) that can be specified to only allow traffic matching the rules to go through the VPN interface

There can be sets of rules linked by *OR*. Within each set, there can be app-based rules and traffic-based rules.
All the properties within the set are linked by *AND*. The rules can be applied at a per-app level or a per-device level.

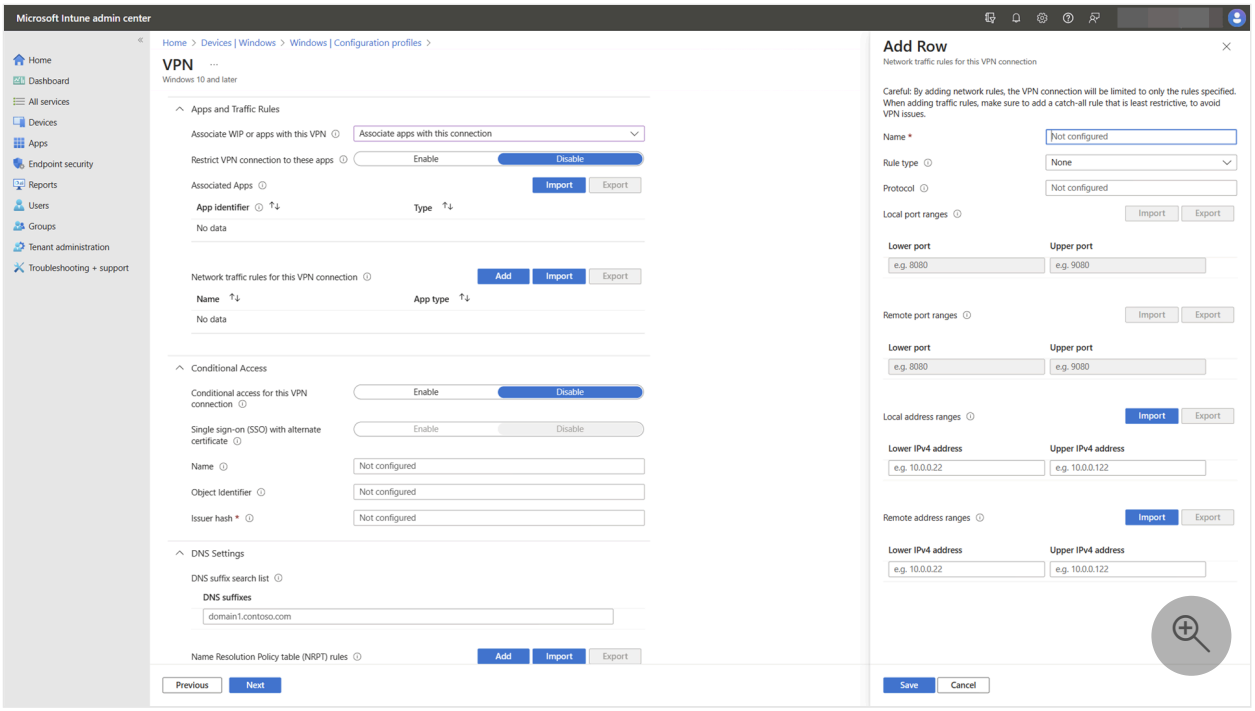
For example, an IT admin could define rules that specify:

- An *HR App* is allowed to go through the VPN and only access port *4545*
- The *Finance apps* are allowed to through the VPN and only access the Remote IP ranges of *10.10.0.40 - 10.10.0.201* on port *5889*
- All other apps on the device can only access ports *80* or *443*

Configure traffic filters

See VPN profile options and VPNv2 CSP for XML configuration.

The following image shows the interface to configure traffic rules in a VPN Profile configuration policy, using Microsoft Intune.



LockDown VPN

A VPN profile configured with LockDown secures the device to only allow network traffic over the VPN interface. It has the following features:

- The system attempts to always keep the VPN connected
- The user can't disconnect the VPN connection
- The user can't delete or modify the VPN profile
- The VPN LockDown profile uses forced tunnel connection
- If the VPN connection isn't available, outbound network traffic is blocked
- Only one VPN LockDown profile is allowed on a device

> **Note**
>
> For built-in VPN, LockDown VPN is only available for the Internet Key Exchange version 2 (IKEv2) connection type.

> **Caution**

Be careful when deploying LockDown VPN, as the resultant connection won't be able to send or receive any network traffic without the VPN connection being established.

Feedback

Was this page helpful? Yes No

Provide product feedback

VPN profile options

Article • 01/28/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

Most of the VPN settings in Windows can be configured in VPN profiles using Microsoft Intune or Microsoft Configuration Manager. VPN settings can be configured using the **ProfileXML** node in the VPNv2 configuration service provider (CSP).

> **Note**
>
> If you're not familiar with CSPs, read Introduction to configuration service providers (CSPs) first.

The following table lists the VPN settings and whether the setting can be configured in Intune and Configuration Manager, or can only be configured using **ProfileXML**.

Expand table

Profile setting	Can be configured in Intune and Configuration Manager
Connection type	Yes
Routing: split-tunnel routes	Yes, except exclusion routes
Routing: forced-tunnel	Yes
Authentication (EAP)	Yes, if connection type is built in
Conditional access	Yes
Name resolution: NRPT	Yes
Name resolution: DNS suffix	No
Name resolution: persistent	No
Auto-trigger: app trigger	Yes
Auto-trigger: name trigger	Yes
Auto-trigger: Always On	Yes
Auto-trigger: trusted network detection	No
LockDown	No

Profile setting	Can be configured in Intune and Configuration Manager
Windows Information Protection (WIP)	Yes
Traffic filters	Yes
Proxy settings	Yes, by PAC/WPAD file or server and port

> **Note**
>
> VPN proxy settings are only used on Force Tunnel Connections. On Split Tunnel Connections, the general proxy settings are used.

The ProfileXML node was added to the VPNv2 CSP to allow users to deploy VPN profile as a single blob. This node is useful for deploying profiles with features that aren't yet supported by MDMs. You can get more examples in the ProfileXML XSD article.

Sample Native VPN profile

The following sample is a sample Native VPN profile. This blob would fall under the ProfileXML node.

XML

```
<VPNProfile>
  <ProfileName>TestVpnProfile</ProfileName>
  <NativeProfile>
    <Servers>testServer.VPN.com</Servers>
    <NativeProtocolType>IKEv2</NativeProtocolType>

    <!--Sample EAP profile (PEAP)-->
    <Authentication>
      <UserMethod>Eap</UserMethod>
      <Eap>
       <Configuration>
          <EapHostConfig
xmlns="http://www.microsoft.com/provisioning/EapHostConfig">
            <EapMethod>
              <Type
xmlns="http://www.microsoft.com/provisioning/EapCommon">25</Type>
              <VendorId
xmlns="http://www.microsoft.com/provisioning/EapCommon">0</VendorId>
              <VendorType
xmlns="http://www.microsoft.com/provisioning/EapCommon">0</VendorType>
              <AuthorId
xmlns="http://www.microsoft.com/provisioning/EapCommon">0</AuthorId>
```

```
            </EapMethod>
            <Config
xmlns="http://www.microsoft.com/provisioning/EapHostConfig">
              <Eap
xmlns="http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1">
                <Type>25</Type>
                <EapType
xmlns="http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV1">
                  <ServerValidation>

<DisableUserPromptForServerValidation>true</DisableUserPromptForServerValida
tion>
                    <ServerNames></ServerNames>
                    <TrustedRootCA>d2 d3 8e ba 60 ca a1 c1 20 55 a2 e1 c8 3b
15 ad 45 01 10 c2 </TrustedRootCA>
                    <TrustedRootCA>d1 76 97 cc 20 6e d2 6e 1a 51 f5 bb 96 e9
35 6d 6d 61 0b 74 </TrustedRootCA>
                  </ServerValidation>
                  <FastReconnect>true</FastReconnect>
                  <InnerEapOptional>false</InnerEapOptional>
                  <Eap
xmlns="http://www.microsoft.com/provisioning/BaseEapConnectionPropertiesV1">
                    <Type>13</Type>
                    <EapType
xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV1">
                      <CredentialsSource>
                        <CertificateStore>
                          <SimpleCertSelection>true</SimpleCertSelection>
                        </CertificateStore>
                      </CredentialsSource>
                      <ServerValidation>

<DisableUserPromptForServerValidation>true</DisableUserPromptForServerValida
tion>
                        <ServerNames></ServerNames>
                        <TrustedRootCA>d2 d3 8e ba 60 ca a1 c1 20 55 a2 e1
c8 3b 15 ad 45 01 10 c2 </TrustedRootCA>
                        <TrustedRootCA>d1 76 97 cc 20 6e d2 6e 1a 51 f5 bb
96 e9 35 6d 6d 61 0b 74 </TrustedRootCA>
                      </ServerValidation>
                      <DifferentUsername>false</DifferentUsername>
                      <PerformServerValidation
xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2">t
rue</PerformServerValidation>
                      <AcceptServerName
xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2">f
alse</AcceptServerName>
                      <TLSExtensions
xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV2">
                        <FilteringInfo
xmlns="http://www.microsoft.com/provisioning/EapTlsConnectionPropertiesV3">
                          <EKUMapping>
                            <EKUMap>
                              <EKUName>AAD Conditional Access</EKUName>
                              <EKUOID>1.3.6.1.4.1.311.87</EKUOID>
```

```
                            </EKUMap>
                          </EKUMapping>
                          <ClientAuthEKUList Enabled="true">
                            <EKUMapInList>
                              <EKUName>AAD Conditional Access</EKUName>
                            </EKUMapInList>
                          </ClientAuthEKUList>
                        </FilteringInfo>
                      </TLSExtensions>
                    </EapType>
                  </Eap>
                  <EnableQuarantineChecks>false</EnableQuarantineChecks>
                  <RequireCryptoBinding>true</RequireCryptoBinding>
                  <PeapExtensions>
                    <PerformServerValidation
xmlns="http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2">t
rue</PerformServerValidation>
                    <AcceptServerName
xmlns="http://www.microsoft.com/provisioning/MsPeapConnectionPropertiesV2">f
alse</AcceptServerName>
                  </PeapExtensions>
                </EapType>
              </Eap>
            </Config>
          </EapHostConfig>
        </Configuration>
      </Eap>
    </Authentication>

    <!--Sample routing policy: in this case, this is a split tunnel
configuration with two routes configured-->
    <RoutingPolicyType>SplitTunnel</RoutingPolicyType>
    <DisableClassBasedDefaultRoute>true</DisableClassBasedDefaultRoute>
  </NativeProfile>
    <Route>
    <Address>192.168.0.0</Address>
    <PrefixSize>24</PrefixSize>
  </Route>
  <Route>
    <Address>10.10.0.0</Address>
    <PrefixSize>16</PrefixSize>
  </Route>

  <!--VPN will be triggered for the two apps specified here-->
  <AppTrigger>
    <App>
      <Id>Microsoft.MicrosoftEdge_8wekyb3d8bbwe</Id>
    </App>
  </AppTrigger>
  <AppTrigger>
    <App>
      <Id>C:\windows\system32\ping.exe</Id>
    </App>
  </AppTrigger>
```

```
  <!--Example of per-app VPN. This configures traffic filtering rules for
two apps. Internet Explorer is configured for force tunnel, meaning that all
traffic allowed through this app must go over VPN. Microsoft Edge is
configured as split tunnel, so whether data goes over VPN or the physical
interface is dictated by the routing configuration.-->
  <TrafficFilter>
    <App>
      <Id>%ProgramFiles%\Internet Explorer\iexplore.exe</Id>
    </App>
    <Protocol>6</Protocol>
    <LocalPortRanges>10,20-50,100-200</LocalPortRanges>
    <RemotePortRanges>20-50,100-200,300</RemotePortRanges>
    <RemoteAddressRanges>30.30.0.0/16,10.10.10.10-
20.20.20.20</RemoteAddressRanges>
    <RoutingPolicyType>ForceTunnel</RoutingPolicyType>
  </TrafficFilter>
  <TrafficFilter>
    <App>
      <Id>Microsoft.MicrosoftEdge_8wekyb3d8bbwe</Id>
    </App>
    <LocalAddressRanges>3.3.3.3/32,1.1.1.1-2.2.2.2</LocalAddressRanges>
  </TrafficFilter>

  <!--Name resolution configuration. The AutoTrigger node configures name-
based triggering. In this profile, the domain "hrsite.corporate.contoso.com"
triggers VPN.-->
  <DomainNameInformation>
    <DomainName>hrsite.corporate.contoso.com</DomainName>
    <DnsServers>1.2.3.4,5.6.7.8</DnsServers>
    <WebProxyServers>5.5.5.5</WebProxyServers>
    <AutoTrigger>true</AutoTrigger>
  </DomainNameInformation>
  <DomainNameInformation>
    <DomainName>.corp.contoso.com</DomainName>
    <DnsServers>10.10.10.10,20.20.20.20</DnsServers>
    <WebProxyServers>100.100.100.100</WebProxyServers>
  </DomainNameInformation>

  <!--EDPMode is turned on for the enterprise ID "corp.contoso.com". When a
user accesses an app with that ID, VPN will be triggered.-->
  <EdpModeId>corp.contoso.com</EdpModeId>
  <RememberCredentials>true</RememberCredentials>

  <!--Always On is turned off, and triggering VPN for the apps and domain
name specified earlier in the profile will not occur if the user is
connected to the trusted network "contoso.com".-->
  <AlwaysOn>false</AlwaysOn>
  <DnsSuffix>corp.contoso.com</DnsSuffix>
  <TrustedNetworkDetection>contoso.com</TrustedNetworkDetection>
  <Proxy>
    <Manual>
      <Server>HelloServer</Server>
    </Manual>
    <AutoConfigUrl>Helloworld.Com</AutoConfigUrl>
  </Proxy>
```

```
  <!--Device compliance is enabled and an alternate certificate is specified
for domain resource authentication.-->
  <DeviceCompliance>
        <Enabled>true</Enabled>
        <Sso>
            <Enabled>true</Enabled>
            <Eku>This is my Eku</Eku>
            <IssuerHash>This is my issuer hash</IssuerHash>
        </Sso>
    </DeviceCompliance>
</VPNProfile>
```

Sample plug-in VPN profile

The following sample is a sample plug-in VPN profile. This blob would fall under the ProfileXML node.

XML

```
<VPNProfile>
    <ProfileName>TestVpnProfile</ProfileName>
    <PluginProfile>

<ServerUrlList>testserver1.contoso.com;testserver2.contoso..com</ServerUrlLi
st>

<PluginPackageFamilyName>JuniperNetworks.JunosPulseVpn_cw5n1h2txyewy</Plugin
PackageFamilyName>
        <CustomConfiguration>&lt;pulse-
schema&gt;&lt;isSingleSignOnCredential&gt;true&lt;/isSingleSignOnCredential&
gt;&lt;/pulse-schema&gt;</CustomConfiguration>
    </PluginProfile>
    <Route>
        <Address>192.168.0.0</Address>
        <PrefixSize>24</PrefixSize>
    </Route>
    <Route>
        <Address>10.10.0.0</Address>
        <PrefixSize>16</PrefixSize>
    </Route>
    <AppTrigger>
        <App>
            <Id>Microsoft.MicrosoftEdge_8wekyb3d8bbwe</Id>
        </App>
    </AppTrigger>
    <AppTrigger>
        <App>
            <Id>%ProgramFiles%\Internet Explorer\iexplore.exe</Id>
        </App>
    </AppTrigger>
```

```
    <TrafficFilter>
        <App>
            <Id>%ProgramFiles%\Internet Explorer\iexplore.exe</Id>
        </App>
        <Protocol>6</Protocol>
        <LocalPortRanges>10,20-50,100-200</LocalPortRanges>
        <RemotePortRanges>20-50,100-200,300</RemotePortRanges>
        <RemoteAddressRanges>30.30.0.0/16,10.10.10.10-
20.20.20.20</RemoteAddressRanges>
        <!--<RoutingPolicyType>ForceTunnel</RoutingPolicyType>-->
    </TrafficFilter>
    <TrafficFilter>
        <App>
            <Id>Microsoft.MicrosoftEdge_8wekyb3d8bbwe</Id>
        </App>
        <LocalAddressRanges>3.3.3.3/32,1.1.1.1-2.2.2.2</LocalAddressRanges>
    </TrafficFilter>
    <TrafficFilter>
        <App>
            <Id>Microsoft.MicrosoftEdge_8wekyb3d8bbwe</Id>
        </App>
        <Claims>O:SYG:SYD:(A;;CC;;;AU)</Claims>
        <!--<RoutingPolicyType>SplitTunnel</RoutingPolicyType>-->
    </TrafficFilter>
    <DomainNameInformation>
        <DomainName>corp.contoso.com</DomainName>
        <DnsServers>1.2.3.4,5.6.7.8</DnsServers>
        <WebProxyServers>5.5.5.5</WebProxyServers>
        <AutoTrigger>false</AutoTrigger>
    </DomainNameInformation>
    <DomainNameInformation>
        <DomainName>corp.contoso.com</DomainName>
        <DnsServers>10.10.10.10,20.20.20.20</DnsServers>
        <WebProxyServers>100.100.100.100</WebProxyServers>
    </DomainNameInformation>
    <!--<EdpModeId>corp.contoso.com</EdpModeId>-->
    <RememberCredentials>true</RememberCredentials>
    <AlwaysOn>false</AlwaysOn>
    <DnsSuffix>corp.contoso.com</DnsSuffix>

<TrustedNetworkDetection>contoso.com,test.corp.contoso.com</TrustedNetworkDe
tection>
    <Proxy>
        <Manual>
            <Server>HelloServer</Server>
        </Manual>
        <AutoConfigUrl>Helloworld.Com</AutoConfigUrl>
    </Proxy>
</VPNProfile>
```

Apply ProfileXML using Intune

After you configure the settings that you want using ProfileXML, you can create a custom profile in the Microsoft Intune admin center. After it's created, you deploy this profile to your devices.

1. Sign in to the Microsoft Intune admin center.
2. Select **Devices** > **Configuration profiles** > **Create profile**.
3. Enter the following properties:
   - **Platform**: Select **Windows 10 and later**
   - **Profile**: Select **Templates** > **Custom**.
4. Select **Create**.
5. In **Basics**, enter the following properties:
   - **Name**: Enter a descriptive name for the profile. Name your profiles so you can easily identify them later.
   - **Description**: Enter a description for the profile. This setting is optional, but recommended.
6. Select **Next**.
7. In **Configuration settings**, enter the following properties:
   - **OMA-URI**: Enter `./user/vendor/MSFT/VPNv2/Your_VPN profile name_/ProfileXML`.
   - **Data type**: Select `String (XML file)`.
   - **Value**: Browse to, and select your XML file.

   For more information on these settings, see Use custom settings for Windows devices in Intune.
8. Select **Next**, and continue configuring the policy. For the specific steps and recommendations, see Create a profile with custom settings in Intune.

Learn more

- Create VPN profiles to connect to VPN servers in Intune
- VPNv2 configuration service provider (CSP) reference
- How to Create VPN Profiles in Configuration Manager

Feedback

Was this page helpful? Yes No

Provide product feedback

How to configure cryptographic settings for IKEv2 VPN connections

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

In IKEv2 VPN connections, the default setting for IKEv2 cryptographic settings are:

- Encryption Algorithm: DES3
- Integrity, Hash Algorithm: SHA1
- Diffie Hellman Group (Key Size): DH2

These settings aren't secure for IKE exchanges.

To secure the connections, update the configuration of VPN servers and clients by running VPN cmdlets.

VPN server

For VPN servers that run Windows Server 2012 R2 or later, you need to run Set-VpnServerConfiguration to configure the tunnel type. These settings are effective for all IKEv2 VPN connections.

PowerShell

```
Set-VpnServerConfiguration -TunnelType IKEv2 -CustomPolicy
```

On an earlier version of Windows Server, run Set-VpnServerIPsecConfiguration. Since `Set-VpnServerIPsecConfiguration` doesn't have `-TunnelType`, the configuration applies to all tunnel types on the server.

PowerShell

```
Set-VpnServerIPsecConfiguration -CustomPolicy
```

VPN client

For VPN client, you need to configure each VPN connection. For example, run Set-VpnConnectionIPsecConfiguration (version 4.0) and specify the name of the connection:

PowerShell

```
Set-VpnConnectionIPsecConfiguration -ConnectionName <String>
```

IKEv2 Crypto Settings Example

The following commands configure the IKEv2 cryptographic settings to:

- Encryption Algorithm: AES128
- Integrity, Hash Algorithm: SHA256
- Diffie Hellman Group (Key Size): DH14

IKEv2 VPN Server

PowerShell

```
Set-VpnServerConfiguration -TunnelType IKEv2 -CustomPolicy -
AuthenticationTransformConstants SHA256128 -CipherTransformConstants AES128
-DHGroup Group14 -EncryptionMethod AES128 -IntegrityCheckMethod SHA256 -
PFSgroup PFS2048 -SALifeTimeSeconds 28800 -MMSALifeTimeSeconds 86400 -
SADataSizeForRenegotiationKilobytes 1024000
restart-service RemoteAccess -PassThru
```

If you need to switch back to the default IKEv2 settings, use this command:

PowerShell

```
Set-VpnServerConfiguration -TunnelType IKEv2 -RevertToDefault
restart-service RemoteAccess -PassThru
```

IKEv2 VPN Client

PowerShell

```
Set-VpnConnectionIPsecConfiguration -ConnectionName <String - your VPN
connection name> -AuthenticationTransformConstants SHA256128 -
CipherTransformConstants AES128 -DHGroup Group14 -EncryptionMethod AES128 -
IntegrityCheckMethod SHA256 -PfsGroup PFS2048 -Force
```

If you need to switch back to the default IKEv2 settings, use this command:

PowerShell

```
Set-VpnConnectionIPsecConfiguration -ConnectionName <String - your VPN
```

```
connection name> -RevertToDefault -Force
```

> **Tip**
>
> If you're configuring a all-user VPN connection or a Device Tunnel you must use the `-AllUserConnection` parameter in the `Set-VpnConnectionIPsecConfiguration` command.

Feedback

Was this page helpful? Yes No

Provide product feedback

How to use single sign-on (SSO) over VPN and Wi-Fi connections

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article explains requirements to enable single sign-on (SSO) to on-premises domain resources over Wi-Fi or VPN connections. The following scenarios are typically used:

- Connecting to a network using Wi-Fi or VPN
- Use credentials for Wi-Fi or VPN authentication to also authenticate requests to access domain resources, without being prompted for domain credentials

For example, you want to connect to a corporate network and access an internal website that requires Windows integrated authentication.

The credentials that are used for the connection authentication are placed in *Credential Manager* as the default credentials for the **logon session**. Credential Manager stores credentials that can be used for specific domain resources. These are based on the target name of the resource:

- For VPN, the VPN stack saves its credential as the **session default**
- For Wi-Fi, Extensible Authentication Protocol (EAP) provides support

The credentials are placed in Credential Manager as a *session credential*:

- A *session credential* implies that it's valid for the current user session
- The credentials are cleaned up when the Wi-Fi or VPN connection is disconnected

> ⓘ **Note**
>
> In Windows 10, version 21H2 and later, the *session credential* isn't visible in Credential Manager.

For example, if someone using Microsoft Edge tries to access a domain resource, Microsoft Edge has the right Enterprise Authentication capability. This allows WinInet to release the credentials that it gets from Credential Manager to the SSP that is requesting it. For more information about the Enterprise Authentication capability, see App capability declarations.

The local security authority looks at the device application to determine if it has the right capability. This includes items such as a Universal Windows Platform (UWP) application. If the app isn't a UWP, it doesn't matter. But, if the application is a UWP app,

it evaluates at the device capability for Enterprise Authentication. If it does have that capability and if the resource that you're trying to access is in the Intranet zone in the Internet Options (ZoneMap), then the credential is released. This behavior helps prevent credentials from being misused by untrusted third parties.

Intranet zone

For the Intranet zone, by default it only allows single-label names, such as `http://finance`. If the resource that needs to be accessed has multiple domain labels, then the workaround is to use the Registry CSP.

Setting the ZoneMap

The ZoneMap is controlled using a registry that can be set through MDM. By default, single-label names such as `http://finance` are already in the intranet zone. For multi-label names, such as `http://finance.net`, the ZoneMap needs to be updated.

MDM Policy

OMA URI example:

`./Vendor/MSFT/Registry/HKU/S-1-5-21-2702878673-795188819-444038987-2781/Software/Microsoft/Windows/CurrentVersion/Internet%20Settings/ZoneMap/Domains/<domain name>` as an `Integer` value of `1` for each of the domains that you want to SSO into from your device. This adds the specified domains to the Intranet Zone of the Microsoft Edge browser.

Credential requirements

For VPN, the following types of credentials will be added to credential manager after authentication:

- Username and password
- Certificate-based authentication:
  - TPM Key Storage Provider (KSP) Certificate
  - Software Key Storage Provider (KSP) Certificates
  - Smart Card Certificate
  - Windows Hello for Business Certificate

The username should also include a domain that can be reached over the connection (VPN or WiFi).

User certificate templates

If the credentials are certificate-based, then the elements in the following table need to be configured for the certificate templates to ensure they can also be used for Kerberos client authentication.

Expand table

Template element	Configuration
SubjectName	The user's distinguished name (DN) where the domain components of the distinguished name reflect the internal DNS namespace when the SubjectAlternativeName doesn't have the fully qualified UPN required to find the domain controller. This requirement is relevant in multi-forest environments as it ensures a domain controller can be located.
SubjectAlternativeName	The user's fully qualified UPN where a domain name component of the user's UPN matches the organizations internal domain's DNS namespace. This requirement is relevant in multi-forest environments as it ensures a domain controller can be located when the SubjectName doesn't have the DN required to find the domain controller.
Key Storage Provider (KSP)	If the device is joined to Microsoft Entra ID, a discrete SSO certificate is used.
EnhancedKeyUsage	One or more of the following EKUs is required: • Client Authentication (for the VPN) • EAP Filtering OID (for Windows Hello for Business) • SmartCardLogon (for Microsoft Entra joined devices) If the domain controllers require smart card EKU either: • SmartCardLogon • id-pkinit-KPClientAuth (1.3.6.1.5.2.3.4) Otherwise: • TLS/SSL Client Authentication (1.3.6.1.5.5.7.3.2)

NDES server configuration

The NDES server is required to be configured so that incoming SCEP requests can be mapped to the correct template to be used. For more information, see Configure certificate infrastructure for SCEP.

Active Directory requirements

You need IP connectivity to a DNS server and domain controller over the network interface so that authentication can succeed as well.

Domain controllers must have appropriate KDC certificates for the client to trust them as domain controllers. Because phones aren't domain-joined, the root CA of the KDC's certificate must be in the Third-Party Root CA or Smart Card Trusted Roots store.

Domain controllers must be using certificates based on the updated *KDC certificate template* Kerberos Authentication.

Feedback

Was this page helpful? Yes No

Provide product feedback

Optimize Microsoft 365 traffic for remote workers with the Windows VPN client

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes how to configure the recommendations in the article VPN split tunneling for Microsoft 365 for the Windows VPN client. This guidance enables VPN administrators to optimize Microsoft 365 usage while ensuring that all other traffic goes over the VPN connection and through existing security gateways or tooling.

The recommendations can be implemented for the built-in Windows VPN client using a *Force Tunneling with Exclusions* approach, defining IP-based exclusions even when using *force tunneling*. Certain traffic can be *split* to use the physical interface, while still forcing all other traffic via the VPN interface. Traffic addressed to defined destinations (like those listed in the Microsoft 365 optimized categories) follows a much more direct and efficient path, without the need to traverse or *hairpin* via the VPN tunnel and back out of the organization's network. For cloud-services like Microsoft 365, this makes a significant difference in performance and usability for remote users.

> **Note**
>
> The term *force tunneling with exclusions* is sometimes confusingly called *split tunnels* by other vendors and in some online documentation. For Windows VPN, the term *split tunneling* is defined differently, as described in the article VPN routing decisions.

Solution Overview

The solution is based upon the use of a VPN Configuration Service Provider Reference profile (VPNv2 CSP) and the embedded ProfileXML. These are used to configure the VPN profile on the device. Various provisioning approaches can be used to create and deploy the VPN profile as discussed in the article Step 6. Configure Windows 10 client Always On VPN connections.

Typically, these VPN profiles are distributed using a Mobile Device Management solution like Intune, as described in VPN profile options and Configure the VPN client by using Intune.

To enable the use of force tunneling in Windows 10 or Windows 11 VPN, the `<RoutingPolicyType>` setting is typically configured with a value of *ForceTunnel* in your existing Profile XML (or script) by way of the following entry, under the `<NativeProfile>` `</NativeProfile>` section:

```
XML
```

```
<RoutingPolicyType>ForceTunnel</RoutingPolicyType>
```

In order to define specific force tunnel exclusions, you then need to add the following lines to your existing Profile XML (or script) for each required exclusion, and place them outside of the `<NativeProfile></NativeProfile>` section as follows:

XML

```
<Route>
 <Address>[IP addresses or subnet]</Address>
 <PrefixSize>[IP Prefix]</PrefixSize>
 <ExclusionRoute>true</ExclusionRoute>
</Route>
```

Entries defined by the `[IP Addresses or Subnet]` and `[IP Prefix]` references will consequently be added to the routing table as *more specific route entries* that will use the Internet-connected interface as the default gateway, as opposed to using the VPN interface. You must define a unique and separate `<Route></Route>` section for each required exclusion.

An example of a correctly formatted Profile XML configuration for force tunnel with exclusions is the following:

XML

```
<VPNProfile>
 <NativeProfile>
  <RoutingPolicyType>ForceTunnel</RoutingPolicyType>
 </NativeProfile>
 <Route>
  <Address>203.0.113.0</Address>
  <PrefixSize>24</PrefixSize>
  <ExclusionRoute>true</ExclusionRoute>
 </Route>
 <Route>
  <Address>198.51.100.0</Address>
  <PrefixSize>22</PrefixSize>
  <ExclusionRoute>true</ExclusionRoute>
 </Route>
</VPNProfile>
```

> **Note**
>
> The IP addresses and prefix size values in this example are used purely as examples only and should not be used.

Solution Deployment

For Microsoft 365, it's therefore necessary to add exclusions for all IP addresses documented within the optimize categories described in Office 365 URLs and IP address ranges to ensure that they're excluded from VPN force tunneling.

This can be achieved manually by adding the IP addresses defined within the *optimize* category entries to an existing Profile XML (or script) file, or alternatively the following script can be used which dynamically adds the required entries to an existing PowerShell script, or XML file, based upon directly querying the REST-based web service to ensure the correct IP address ranges are always used.

An example of a PowerShell script that can be used to update a force tunnel VPN connection with Microsoft 365 exclusions is provided below.

PowerShell

```
# Copyright (c) Microsoft Corporation.  All rights reserved.
#
# THIS SAMPLE CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
KIND,
# WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
# IF THIS CODE AND INFORMATION IS MODIFIED, THE ENTIRE RISK OF USE OR RESULTS IN
# CONNECTION WITH THE USE OF THIS CODE AND INFORMATION REMAINS WITH THE USER.

<#
.SYNOPSIS
    Applies or updates recommended Microsoft 365 optimize IP address exclusions to
an existing force tunnel Windows 10 and Windows 11 VPN profile
.DESCRIPTION
    Connects to the Microsoft 365 worldwide commercial service instance endpoints
to obtain the latest published IP address ranges
    Compares the optimized IP addresses with those contained in the supplied VPN
Profile (PowerShell or XML file)
    Adds or updates IP addresses as necessary and saves the resultant file with "-
NEW" appended to the file name
.PARAMETERS
    Filename and path for a supplied Windows 10 or Windows 11 VPN profile file in
either PowerShell or XML format
.NOTES
    Requires at least Windows 10 Version 1803 with KB4493437, 1809 with KB4490481,
or later
.VERSION
    1.0
#>

param (
    [string]$VPNprofilefile
)
```

```
$usage=@"

This script uses the following parameters:

VPNprofilefile - The full path and name of the VPN profile PowerShell script or
XML file

EXAMPLES

To check a VPN profile PowerShell script file:

Update-VPN-Profile-Office365-Exclusion-Routes.ps1 -VPNprofilefile [FULLPATH AND
NAME OF POWERSHELL SCRIPT FILE]

To check a VPN profile XML file:

Update-VPN-Profile-Office365-Exclusion-Routes.ps1 -VPNprofilefile [FULLPATH AND
NAME OF XML FILE]

"@

# Check if filename has been provided #
if ($VPNprofilefile -eq "")
{
   Write-Host "`nWARNING: You must specify either a PowerShell script or XML
filename!" -ForegroundColor Red

    $usage
    exit
}

$FileExtension = [System.IO.Path]::GetExtension($VPNprofilefile)

# Check if XML file exists and is a valid XML file #
if ( $VPNprofilefile -ne "" -and $FileExtension -eq ".xml")
{
    if ( Test-Path $VPNprofilefile )
    {
        $xml = New-Object System.Xml.XmlDocument
        try
        {
            $xml.Load((Get-ChildItem -Path $VPNprofilefile).FullName)

        }
        catch [System.Xml.XmlException]
        {
            Write-Verbose "$VPNprofilefile : $($_.toString())"
            Write-Host "`nWARNING: The VPN profile XML file is not a valid xml
file or incorrectly formatted!" -ForegroundColor Red
            $usage
            exit
        }
    }else
    {
```

```
        Write-Host "`nWARNING: VPN profile XML file does not exist or cannot be
found!" -ForegroundColor Red
        $usage
        exit
    }
}

# Check if VPN profile PowerShell script file exists and contains a VPNPROFILE XML
section #
if ( $VPNprofilefile -ne "" -and $FileExtension -eq ".ps1")
{
    if ( (Test-Path $VPNprofilefile) )
    {
        if (-Not $(Select-String -Path $VPNprofilefile -Pattern "<VPNPROFILE>") )
        {
            Write-Host "`nWARNING: PowerShell script file does not contain a valid
VPN profile XML section or is incorrectly formatted!" -ForegroundColor Red
            $usage
            exit
        }
    }else
    {
        Write-Host "`nWARNING: PowerShell script file does not exist or cannot be
found!"-ForegroundColor Red
        $usage
        exit
    }
}

# Define Microsoft 365 endpoints and service URLs #
$ws = "https://endpoints.office.com"
$baseServiceUrl = "https://endpoints.office.com"

# Path where client ID and latest version number will be stored #
$datapath = $Env:TEMP + "\endpoints_clientid_latestversion.txt"

# Fetch client ID and version if data file exists; otherwise create new file #
if (Test-Path $datapath)
{
    $content = Get-Content $datapath
    $clientRequestId = $content[0]
    $lastVersion = $content[1]

}else
{
    $clientRequestId = [GUID]::NewGuid().Guid
    $lastVersion = "0000000000"
    @($clientRequestId, $lastVersion) | Out-File $datapath
}

# Call version method to check the latest version, and pull new data if version
number is different #
$version = Invoke-RestMethod -Uri ($ws + "/version?clientRequestId=" +
$clientRequestId)
```

```
if ($version[0].latest -gt $lastVersion)
{

    Write-Host
    Write-Host "A new version of Microsoft 365 worldwide commercial service
instance endpoints has been detected!" -ForegroundColor Cyan

    # Write the new version number to the data file #
    @($clientRequestId, $version[0].latest) | Out-File $datapath
}

# Invoke endpoints method to get the new data #
$uri = "$baseServiceUrl" + "/endpoints/worldwide?clientRequestId=$clientRequestId"

# Invoke endpoints method to get the data for the VPN profile comparison #
$endpointSets = Invoke-RestMethod -Uri ($uri)
$Optimize = $endpointSets | Where-Object { $_.category -eq "Optimize" }
$optimizeIpsv4 = $Optimize.ips | Where-Object { ($_).contains(".") } | Sort-Object
-Unique

# Temporarily include additional IP address until Teams client update is released
$optimizeIpsv4 += "13.107.60.1/32"

# Process PowerShell script file start #
if ($VPNprofilefile -ne "" -and $FileExtension -eq ".ps1")
{
    Write-host "`nStarting PowerShell script exclusion route check...`n" -
ForegroundColor Cyan

    # Clear Variables to allow re-run testing #

    $ARRVPN=$null              # Array to hold VPN addresses from VPN profile
PowerShell file #
    $In_Opt_Only=$null         # Variable to hold IP addresses that only appear in
the optimize list #
    $In_VPN_Only=$null         # Variable to hold IP addresses that only appear in
the VPN profile PowerShell file #

    # Extract the Profile XML from the ps1 file #

    $regex = '(?sm).*^*.<VPNProfile>\r?\n(.*?)\r?\n</VPNProfile>.*'

    # Create xml format variable to compare with the optimize list #

    $xmlbody=(Get-Content -Raw $VPNprofilefile) -replace $regex, '$1'
    [xml]$VPNprofilexml="<VPNProfile>"+$xmlbody+"</VPNProfile>"

        # Loop through each address found in VPNPROFILE XML section #
        foreach ($Route in $VPNprofilexml.VPNProfile.Route)
        {
        $VPNIP=$Route.Address+"/"+$Route.PrefixSize
        [array]$ARRVPN=$ARRVPN+$VPNIP
        }

    # In optimize address list only #
```

```
    $In_Opt_Only= $optimizeIpsv4 | Where {$ARRVPN -NotContains $_}

    # In VPN list only #
    $In_VPN_only =$ARRVPN | Where {$optimizeIpsv4 -NotContains $_}
    [array]$Inpfile = get-content $VPNprofilefile

    if ($In_Opt_Only.Count -gt 0 )
    {
        Write-Host "Exclusion route IP addresses are unknown, missing, or need to
be updated in the VPN profile`n" -ForegroundColor Red

         [int32]$insline=0

            for ($i=0; $i -lt $Inpfile.count; $i++)
            {
                if ($Inpfile[$i] -match "</NativeProfile>")
                {
                $insline += $i # Record the position of the line after the
NativeProfile section ends #
                }
            }
            $OFS = "`r`n"
                foreach ($NewIP in $In_Opt_Only)
                {
                    # Add the missing IP address(es) #
                    $IPInfo=$NewIP.Split("/")
                    $InpFile[$insline] += $OFS+"    <Route>"
                    $InpFile[$insline] += $OFS+"
<Address>"+$IPInfo[0].Trim()+"</Address>"
                    $InpFile[$insline] += $OFS+"
<PrefixSize>"+$IPInfo[1].Trim()+"</PrefixSize>"
                    $InpFile[$insline] += $OFS+"
<ExclusionRoute>true</ExclusionRoute>"
                    $InpFile[$insline] += $OFS+"    </Route>"
                }
             # Update fileName and write new PowerShell file #
             $NewFileName=(Get-Item $VPNprofilefile).Basename + "-NEW.ps1"
             $OutFile=$(Split-Path $VPNprofilefile -Parent)+"\"+$NewFileName
             $InpFile | Set-Content $OutFile
             Write-Host "Exclusion routes have been added to VPN profile and
output to a separate PowerShell script file; the original file has not been
modified`n" -ForegroundColor Green
    }else
    {
        Write-Host "Exclusion route IP addresses are correct and up to date in the
VPN profile`n" -ForegroundColor Green
        $OutFile=$VPNprofilefile
    }

if ( $In_VPN_Only.Count -gt 0 )
{
    Write-Host "Unknown exclusion route IP addresses have been found in the VPN
profile`n" -ForegroundColor Yellow

        foreach ($OldIP in $In_VPN_Only)
```

```
        {
            [array]$Inpfile = get-content $Outfile
            $IPInfo=$OldIP.Split("/")
            Write-Host "Unknown exclusion route IP address"$IPInfo[0]"has been
found in the VPN profile - Do you wish to remove it? (Y/N)`n" -ForegroundColor
Yellow
            $matchstr="<Address>"+$IPInfo[0].Trim()+"</Address>"
            $DelAns=Read-host
                if ($DelAns.ToUpper() -eq "Y")
                {
                    [int32]$insline=0
                        for ($i=0; $i -lt $Inpfile.count; $i++)
                        {
                            if ($Inpfile[$i] -match $matchstr)
                            {
                                $insline += $i # Record the position of the line
for the string match #
                            }
                        }
                        # Remove entries from XML #
                        $InpFile[$insline-1]="REMOVETHISLINE"
                        $InpFile[$insline]="REMOVETHISLINE"
                        $InpFile[$insline+1]="REMOVETHISLINE"
                        $InpFile[$insline+2]="REMOVETHISLINE"
                        $InpFile[$insline+3]="REMOVETHISLINE"
                        $InpFile=$InpFile | Where-Object {$_ -ne "REMOVETHISLINE"}

                        # Update filename and write new PowerShell file #
                        $NewFileName=(Get-Item $VPNprofilefile).Basename + "-
NEW.xml"
                        $OutFile=$(Split-Path $VPNprofilefile -
Parent)+"\"+$NewFileName
                        $Inpfile | Set-content $OutFile
                        Write-Host "`nAddress"$IPInfo[0]"exclusion route has been
removed from the VPN profile and output to a separate PowerShell script file; the
original file has not been modified`n" -ForegroundColor Green

                }else
                {
                    Write-Host "`nExclusion route IP address has *NOT* been
removed from the VPN profile`n" -ForegroundColor Green
                }
        }
 }
}

# Process XML file start #
if ($VPNprofilefile -ne "" -and $FileExtension -eq ".xml")
{
    Write-host "`nStarting XML file exclusion route check...`n" -ForegroundColor
Cyan

    # Clear variables to allow re-run testing #
    $ARRVPN=$null               # Array to hold VPN addresses from the XML file #
    $In_Opt_Only=$null          # Variable to hold IP Addresses that only appear in
```

```
optimize list #
    $In_VPN_Only=$null          # Variable to hold IP Addresses that only appear in
the VPN profile XML file #

    # Extract the Profile XML from the XML file #
    $regex = '(?sm).*^*.<VPNProfile>\r?\n(.*?)\r?\n</VPNProfile>.*'

    # Create xml format variable to compare with optimize list #
    $xmlbody=(Get-Content -Raw $VPNprofilefile) -replace $regex, '$1'
    [xml]$VPNRulesxml="$xmlbody"

        # Loop through each address found in VPNPROFILE file #
        foreach ($Route in $VPNRulesxml.VPNProfile.Route)
        {
            $VPNIP=$Route.Address+"/"+$Route.PrefixSize
            [array]$ARRVPN=$ARRVPN+$VPNIP
        }

    # In optimize address list only #
    $In_Opt_Only= $optimizeIpsv4 | Where {$ARRVPN -NotContains $_}

    # In VPN list only #
    $In_VPN_only =$ARRVPN | Where {$optimizeIpsv4 -NotContains $_}
    [System.Collections.ArrayList]$Inpfile = get-content $VPNprofilefile

        if ($In_Opt_Only.Count -gt 0 )
        {
            Write-Host "Exclusion route IP addresses are unknown, missing, or need
to be updated in the VPN profile`n" -ForegroundColor Red

                foreach ($NewIP in $In_Opt_Only)
                {
                    # Add the missing IP address(es) #
                    $IPInfo=$NewIP.Split("/")
                    $routes += "<Route>`n"+"`t<Address>"+$IPInfo[0].Trim()+"
</Address>`n"+"`t<PrefixSize>"+$IPInfo[1].Trim()+"
</PrefixSize>`n"+"`t<ExclusionRoute>true</ExclusionRoute>`n"+"</Route>`n"
                }
            $inspoint = $Inpfile.IndexOf("</VPNProfile>")
            $Inpfile.Insert($inspoint,$routes)

            # Update filename and write new XML file #
            $NewFileName=(Get-Item $VPNprofilefile).Basename + "-NEW.xml"
            $OutFile=$(Split-Path $VPNprofilefile -Parent)+"\"+$NewFileName
            $InpFile | Set-Content $OutFile
            Write-Host "Exclusion routes have been added to VPN profile and output
to a separate XML file; the original file has not been modified`n`n" -
ForegroundColor Green

        }else
        {
            Write-Host "Exclusion route IP addresses are correct and up to date in
the VPN profile`n" -ForegroundColor Green
            $OutFile=$VPNprofilefile
        }
```

```
        if ( $In_VPN_Only.Count -gt 0 )
        {
            Write-Host "Unknown exclusion route IP addresses found in the VPN
profile`n" -ForegroundColor Yellow

            foreach ($OldIP in $In_VPN_Only)
            {
                [array]$Inpfile = get-content $OutFile
                $IPInfo=$OldIP.Split("/")
                Write-Host "Unknown exclusion route IP address"$IPInfo[0]"has been
found in the VPN profile - Do you wish to remove it? (Y/N)`n" -ForegroundColor
Yellow
                $matchstr="<Route>"+"<Address>"+$IPInfo[0].Trim()+"</Address>"+"
<PrefixSize>"+$IPInfo[1].Trim()+"</PrefixSize>"+"
<ExclusionRoute>true</ExclusionRoute>"+"</Route>"
                $DelAns=Read-host
                    if ($DelAns.ToUpper() -eq "Y")
                    {
                        # Remove unknown IP address(es) #
                        $inspoint = $Inpfile[0].IndexOf($matchstr)
                        $Inpfile[0] = $Inpfile[0].Replace($matchstr,"")

                        # Update filename and write new XML file #
                        $NewFileName=(Get-Item $VPNprofilefile).Basename + "-
NEW.xml"
                        $OutFile=$(Split-Path $VPNprofilefile -
Parent)+"\"+$NewFileName
                        $Inpfile | Set-content $OutFile
                        Write-Host "`nAddress"$IPInfo[0]"exclusion route has been
removed from the VPN profile and output to a separate XML file; the original file
has not been modified`n" -ForegroundColor Green

                    }else
                    {
                        Write-Host "`nExclusion route IP address has *NOT* been
removed from the VPN profile`n" -ForegroundColor Green
                    }
            }
        }
}
```

Other Considerations

You should also be able to adapt this approach to include necessary exclusions for other cloud-services that can be defined by known/static IP addresses; exclusions required for Cisco WebEx or Zoom are good examples.

Examples

An example of a PowerShell script that can be used to create a force tunnel VPN connection with Microsoft 365 exclusions is provided below, or refer to the guidance in Create the ProfileXML configuration files to create the initial PowerShell script:

PowerShell

```
# Copyright (c) Microsoft Corporation.  All rights reserved.
#
# THIS SAMPLE CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY
KIND,
# WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
# IF THIS CODE AND INFORMATION IS MODIFIED, THE ENTIRE RISK OF USE OR RESULTS IN
# CONNECTION WITH THE USE OF THIS CODE AND INFORMATION REMAINS WITH THE USER.

<#
.SYNOPSIS
    Configures an AlwaysOn IKEv2 VPN Connection using a basic script
.DESCRIPTION
    Configures an AlwaysOn IKEv2 VPN Connection with proxy PAC information and
force tunneling
.PARAMETERS
    Parameters are defined in a ProfileXML object within the script itself
.NOTES
    Requires at least Windows 10 Version 1803 with KB4493437, 1809 with KB4490481,
or later
.VERSION
    1.0
#>

<#-- Define Key VPN Profile Parameters --#>
$ProfileName = 'Contoso VPN with Microsoft 365 Exclusions'
$ProfileNameEscaped = $ProfileName -replace ' ', '%20'

<#-- Define VPN ProfileXML --#>
$ProfileXML = '<VPNProfile>
      <RememberCredentials>true</RememberCredentials>
    <DnsSuffix>corp.contoso.com</DnsSuffix>
    <AlwaysOn>true</AlwaysOn>
    <TrustedNetworkDetection>corp.contoso.com</TrustedNetworkDetection>
<NativeProfile>
        <Servers>edge1.contoso.com</Servers>
        <RoutingPolicyType>ForceTunnel</RoutingPolicyType>
        <NativeProtocolType>IKEv2</NativeProtocolType>
        <Authentication>
            <MachineMethod>Certificate</MachineMethod>
        </Authentication>
    </NativeProfile>
    <Route>
      <Address>13.107.6.152</Address>
      <PrefixSize>31</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
```

```
    <Route>
      <Address>13.107.18.10</Address>
      <PrefixSize>31</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>13.107.128.0</Address>
      <PrefixSize>22</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>23.103.160.0</Address>
      <PrefixSize>20</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>40.96.0.0</Address>
      <PrefixSize>13</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>40.104.0.0</Address>
      <PrefixSize>15</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>52.96.0.0</Address>
      <PrefixSize>14</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>131.253.33.215</Address>
      <PrefixSize>32</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>132.245.0.0</Address>
      <PrefixSize>16</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>150.171.32.0</Address>
      <PrefixSize>22</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>191.234.140.0</Address>
      <PrefixSize>22</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>204.79.197.215</Address>
      <PrefixSize>32</PrefixSize>
    <ExclusionRoute>true</ExclusionRoute>
    </Route>
```

```
    <Route>
      <Address>13.107.136.0</Address>
      <PrefixSize>22</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>40.108.128.0</Address>
      <PrefixSize>17</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>52.104.0.0</Address>
      <PrefixSize>14</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>104.146.128.0</Address>
      <PrefixSize>17</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>150.171.40.0</Address>
      <PrefixSize>22</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>13.107.60.1</Address>
      <PrefixSize>32</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>13.107.64.0</Address>
     <PrefixSize>18</PrefixSize>
    <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>52.112.0.0</Address>
      <PrefixSize>14</PrefixSize>
      <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Route>
      <Address>52.120.0.0</Address>
      <PrefixSize>14</PrefixSize>
    <ExclusionRoute>true</ExclusionRoute>
    </Route>
    <Proxy>

<AutoConfigUrl>http://webproxy.corp.contoso.com/proxy.pac</AutoConfigUrl>
      </Proxy>
</VPNProfile>'

<#-- Convert ProfileXML to Escaped Format --#>
$ProfileXML = $ProfileXML -replace '<', '&lt;'
$ProfileXML = $ProfileXML -replace '>', '&gt;'
$ProfileXML = $ProfileXML -replace '"', '&quot;'
```

```
<#-- Define WMI-to-CSP Bridge Properties --#>
$nodeCSPURI = './Vendor/MSFT/VPNv2'
$namespaceName = "root\cimv2\mdm\dmmap"
$className = "MDM_VPNv2_01"

<#-- Define WMI Session --#>
$session = New-CimSession

<#-- Detect and Delete Previous VPN Profile --#>
try
{
    $deleteInstances = $session.EnumerateInstances($namespaceName, $className,
$options)
    foreach ($deleteInstance in $deleteInstances)
    {
        $InstanceId = $deleteInstance.InstanceID
        if ("$InstanceId" -eq "$ProfileNameEscaped")
        {
            $session.DeleteInstance($namespaceName, $deleteInstance, $options)
            $Message = "Removed $ProfileName profile $InstanceId"
            Write-Host "$Message"
        } else {
            $Message = "Ignoring existing VPN profile $InstanceId"
            Write-Host "$Message"
        }
    }
}
catch [Exception]
{
    $Message = "Unable to remove existing outdated instance(s) of $ProfileName
profile: $_"
    Write-Host "$Message"
    exit
}

<#-- Create VPN Profile --#>
try
{
    $newInstance = New-Object Microsoft.Management.Infrastructure.CimInstance
$className, $namespaceName
    $property =
[Microsoft.Management.Infrastructure.CimProperty]::Create("ParentID",
"$nodeCSPURI", 'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property =
[Microsoft.Management.Infrastructure.CimProperty]::Create("InstanceID",
"$ProfileNameEscaped", 'String', 'Key')
    $newInstance.CimInstanceProperties.Add($property)
    $property =
[Microsoft.Management.Infrastructure.CimProperty]::Create("ProfileXML",
"$ProfileXML", 'String', 'Property')
    $newInstance.CimInstanceProperties.Add($property)

    $session.CreateInstance($namespaceName, $newInstance, $options)
```

```
    $Message = "Created $ProfileName profile."
    Write-Host "$Message"
    Write-Host "$ProfileName profile summary:"
    $session.EnumerateInstances($namespaceName, $className, $options)
}
catch [Exception]
{
    $Message = "Unable to create $ProfileName profile: $_"
    Write-Host "$Message"
    exit
}

$Message = "Script Complete"
Write-Host "$Message"
```

An example of an Intune-ready XML file that can be used to create a force tunnel VPN connection with Microsoft 365 exclusions is provided below, or refer to the guidance in Create the ProfileXML configuration files to create the initial XML file.

> **Note**
>
> This XML is formatted for use with Intune and cannot contain any carriage returns or whitespace.

XML

```
<VPNProfile><RememberCredentials>true</RememberCredentials>
<DnsSuffix>corp.contoso.com</DnsSuffix><AlwaysOn>true</AlwaysOn>
<TrustedNetworkDetection>corp.contoso.com</TrustedNetworkDetection><NativeProfile>
<Servers>edge1.contoso.com</Servers>
<RoutingPolicyType>ForceTunnel</RoutingPolicyType>
<NativeProtocolType>IKEv2</NativeProtocolType><Authentication>
<MachineMethod>Certificate</MachineMethod></Authentication></NativeProfile><Route>
<Address>13.107.6.152</Address><PrefixSize>31</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>13.107.18.10</Address><PrefixSize>31</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>13.107.128.0</Address><PrefixSize>22</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>23.103.160.0</Address><PrefixSize>20</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>40.96.0.0</Address>
<PrefixSize>13</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>40.104.0.0</Address><PrefixSize>15</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>52.96.0.0</Address>
<PrefixSize>14</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>131.253.33.215</Address><PrefixSize>32</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>132.245.0.0</Address>
<PrefixSize>16</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>150.171.32.0</Address><PrefixSize>22</PrefixSize>
```

```
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>191.234.140.0</Address><PrefixSize>22</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>204.79.197.215</Address><PrefixSize>32</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>13.107.136.0</Address><PrefixSize>22</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>40.108.128.0</Address><PrefixSize>17</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>52.104.0.0</Address>
<PrefixSize>14</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>104.146.128.0</Address><PrefixSize>17</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>150.171.40.0</Address><PrefixSize>22</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>13.107.60.1</Address>
<PrefixSize>32</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>13.107.64.0</Address><PrefixSize>18</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Route><Address>52.112.0.0</Address>
<PrefixSize>14</PrefixSize><ExclusionRoute>true</ExclusionRoute></Route><Route>
<Address>52.120.0.0</Address><PrefixSize>14</PrefixSize>
<ExclusionRoute>true</ExclusionRoute></Route><Proxy>
<AutoConfigUrl>http://webproxy.corp.contoso.com/proxy.pac</AutoConfigUrl></Proxy>
</VPNProfile>
```

Always On VPN Overview

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Always On VPN allows you to:

- **Create advanced scenarios** by integrating Windows operating systems and third-party solutions. For a list of supported integrations, see Supported integrations.
- **Maintain network security**, restricting connection by traffic types, applications, and authentication methods. For a list of Always On VPN security features, see Security features.
- **Configure auto-triggering** for user and device authenticated connections. For more information, see Connectivity features.
- **Control your network** by creating routing policies at a granular level; even down to the individual application. For more information, see Networking features.
- **Configure your VPN settings** with a standard XML profile (ProfileXML) which is defined by an industry standard configuration template. You can deploy and manage your VPN settings with Windows PowerShell, Microsoft Endpoint Configuration Manager, Intune, Windows Configuration Designer, or any third-party mobile device management (MDM) tool.

Supported integrations

Always On VPN supports domain-joined, nondomain-joined (workgroup), or Microsoft Entra ID–joined devices to allow for both enterprise and BYOD scenarios. Always On VPN is available in all Windows editions, and the platform features are available to third parties by way of UWP VPN plug-in support.

Always On VPN supports integration with the following platforms:

- **Windows Information Protection (WIP).** Integration with WIP allows network policy enforcement to determine whether traffic is permitted to go over the VPN. If the user profile is active and WIP policies are applied, Always On VPN is automatically triggered to connect. Also, when you use WIP, there's no need to specify `AppTriggerList` and `TrafficFilterList` rules separately in the VPN profile (unless you want more advanced configuration) because the WIP policies and application lists automatically take effect.

- **Windows Hello for Business.** Always On VPN natively supports Windows Hello for Business in certificate-based authentication mode. The native Windows Hello support provides a seamless single sign-on experience for both sign-in to the machine, as well as connection to the VPN. No secondary authentication (user credentials) is needed for the VPN connection.

- **Microsoft Azure conditional access platform.** The Always On VPN client can integrate with the Azure conditional access platform to enforce multifactor authentication (MFA), device compliance, or a combination of the two. When compliant with conditional access policies, Microsoft Entra ID issues a short-lived (by default, sixty minutes) IP Security (IPsec) authentication certificate. The IPSec certificate can then be used to authenticate to the VPN gateway. Device compliance uses Configuration Manager/Intune compliance policies, which can include the device health attestation state as part of the connection compliance check. For more details, see VPN and conditional access

- **Microsoft Entra multifactor authentication platform.** When combined with Remote Authentication Dial-In User Service (RADIUS) services and the Network Policy Server (NPS) extension for Microsoft Entra multifactor authentication, VPN authentication can use strong MFA.

- **Third-party VPN plug-ins.** With the Universal Windows Platform (UWP), third-party VPN providers can create a single application for the full range of Windows devices. The UWP provides a guaranteed core API layer across devices, eliminating the complexity of and problems often associated with writing kernel-level drivers. Currently, Windows UWP VPN plug-ins exist for Pulse Secure, F5 Access, Check Point Capsule VPN, FortiClient, SonicWall Mobile Connect, and GlobalProtect.

Security features

Always On VPN provides connectivity to corporate resources by using tunnel policies that require authentication and encryption until they reach the VPN gateway. By default, the tunnel sessions terminate at the VPN gateway, which also functions as the IKEv2 gateway, providing end-to-edge security.

For details about standard VPN authentication options, see VPN authentication options.

Always On VPN supports the following security features:

- **Industry-standard IKEv2 VPN protocol support.** The Always On VPN client supports IKEv2, one of today's most widely used industry-standard tunneling protocols. This compatibility maximizes interoperability with third-party VPN gateways.

- **Interoperability with third-party IKEv2 VPN gateways.** The Always On VPN client supports interoperability with third-party IKEv2 VPN gateways. You can also achieve interoperability with third-party VPN gateways by using a UWP VPN plug-in combined with a custom tunneling type without sacrificing Always On VPN platform features and benefits.

  > ⓘ **Note**
  >
  > Consult with your gateway or third-party back-end appliance vendor on configurations and compatibility with Always On VPN and Device Tunnel using IKEv2.

- **Fall back to SSTP from IKEv2.** You can configure fall back for clients that are behind firewalls or proxy servers by using the automatic tunnel/protocol type within the VPN profile.

  > ⓘ **Note**
  >
  > User Tunnel supports SSTP and IKEv2, and Device Tunnel supports IKEv2 only, with no support for SSTP fallback.

- **Support for machine certificate authentication.** The IKEv2 protocol type available as part of the Always On VPN platform specifically supports the use of machine or computer certificates for VPN authentication.

  > ⓘ **Note**
  >
  > IKEv2 is the only supported protocol for Device Tunnel and there is no support option for SSTP fallback. For more information see, Configure an Always On VPN device tunnel.

- **Traffic and app filters.** With traffic and app firewall rules, you can specify client-side policies that determine which traffic and apps are allowed to connect to the VPN interface.

  Two types of filtering rules are available:

  - **App-based rules.** App-based firewall rules are based on a list of specified applications so that only traffic originating from these apps are permitted to go over the VPN interface.

- **Traffic-based rules.** Traffic-based firewall rules are based on network requirements like ports, addresses, and protocols. Use these rules only for traffic that matches these specific conditions are permitted to go over the VPN interface.

> ⓘ Note
>
> These rules apply only to traffic outbound from the device. Use of traffic filters blocks inbound traffic from the corporate network to the client.

- **VPN conditional access.** Conditional access and device compliance can require managed devices to meet standards before they can connect to the VPN. VPN conditional access allows you to restrict the VPN connections to the devices whose client authentication certificate contains the Microsoft Entra Conditional Access OID of `1.3.6.1.4.1.311.87`. To learn how to restrict the VPN connections directly on the NPS server, see Configure VPN conditional access on the Network Policy Server. To learn how to restrict the VPN connections with Microsoft Entra Conditional Access, see Conditional access for VPN connectivity using Microsoft Entra ID.

- **Limit remote access to specific users and devices.** You can configure Always On VPN to support granular authorization when using RADIUS, which includes the use of security groups to control VPN access.

- **Define accessible management servers before user sign-in.** Use the Device Tunnel feature (available in version 1709 – for IKEv2 only) in the VPN profile combined with traffic filters to control which management systems on the corporate network are accessible through the Device Tunnel.

> ⓘ Note
>
> If you turn on traffic filters in the Device Tunnel profile, then the Device Tunnel denies inbound traffic (from the corporate network to the client).

- **Per-app VPN.** Per-app VPN is like having an app-based traffic filter, but it goes farther to combine application triggers with an app-based traffic filter so that VPN connectivity is constrained to a specific application as opposed to all applications on the VPN client. The feature automatically initiates when the app starts.

- **Customized IPsec cryptography algorithms.** Always On VPN supports the use of both RSA and elliptic curve cryptography–based custom cryptographic algorithms to meet stringent government or organizational security policies.

- **Native Extensible Authentication Protocol (EAP) support.** Always On VPN natively supports EAP, which allows you to use a diverse set of Microsoft and third-party EAP types as part of the authentication workflow. EAP provides secure authentication based on the following authentication types:
  - Username and password
  - Smart card (both physical and virtual)
  - User certificates
  - Windows Hello for Business
  - MFA support by way of EAP RADIUS integration

  The application vendor controls third-party UWP VPN plug-in authentication methods, although they have an array of available options, including custom credential types and OTP support.

- **Windows Hello for Business two-factor authentication on PCs and mobile devices.** In Windows 10, Windows Hello for Business replaces passwords by providing strong two-factor authentication on PCs and mobile devices. For more information, see Enabling Remote Access with Windows Hello for Business in Windows 10

- **Azure multifactor authentication (MFA).** Microsoft Entra multifactor authentication has cloud and on-premises versions that you can integrate with the Windows VPN authentication mechanism. For more information, see Integrate RADIUS authentication with Azure Multi-Factor Authentication Server.

- **Trusted Platform Module (TPM) Key Attestation.** A user certificate that has a TPM-attested key provides higher security assurance, backed up by non-exportability, anti-hammering, and isolation of keys provided by the TPM.

For more information about TPM key attestation in Windows 10, see TPM Key Attestation.

Connectivity features

Always On VPN supports the following connectivity features:

- **Application auto-triggering.** You can configure Always On VPN to support auto-triggering based on application launch or namespace resolution requests. For more information on how to configure auto-triggering, see VPN auto-triggered profile options.

- **Name-based auto-triggering.** With Always On VPN, you can define rules so that specific domain name queries trigger the VPN connection. Windows devices support name-based triggering for domain-joined and nondomain-joined machines (previously, only nondomain-joined machines were supported).

- **Trusted network detection.** Always On VPN includes this feature to ensure that VPN connectivity isn't triggered if a user is connected to a trusted network within the corporate boundary. You can combine this feature with any of the triggering methods mentioned earlier to provide a seamless "only connect when needed" user experience.

- **Device Tunnel.** Always On VPN gives you the ability to create a dedicated VPN profile for device or machine. Unlike *User Tunnel*, which only connects after a user logs on to the device or machine, *Device Tunnel* allows the VPN to establish connectivity before user sign-in. Both Device Tunnel and User Tunnel operate independently with their VPN profiles, can be connected at the same time, and can use different authentication methods and other VPN configuration settings as appropriate. For information on how to configure a device tunnel, including information on how to use manage-out to dynamically register client IP addresses in DNS, see Configure an Always On VPN device tunnel.

  > **Note**
  >
  > Device Tunnel can only be configured on domain-joined devices running Windows 10 Enterprise or Education version 1709 or later. There's no support for third-party control of the Device Tunnel.

- **Connectivity Assistant** Always On VPN is fully integrated with the native Network Connectivity Assistant and provides connectivity status from the View All Networks interface. With the advent of Windows 10 Creators Update (version 1703), VPN connection status and VPN connection control for User Tunnel are available through the Network flyout (for the Windows built-in VPN client).

Networking features

Always On VPN supports the following networking features:

- **Dual-stack support for IPv4 and IPv6.** Always On VPN natively supports the use of both IPv4 and IPv6 in a dual-stack approach. It has no specific dependency on one protocol over the other, which allows for maximum IPv4/IPv6 application compatibility combined with support for future IPv6 networking needs.

- **Application-specific routing policies.** In addition to defining global VPN connection routing policies for internet and intranet traffic separation, it's possible to add routing policies to control the use of split tunnel or force tunnel configurations on a per-application basis. This option gives you more granular control over which apps are allowed to interact with which resources through the VPN tunnel.

- **Exclusion routes.** Always On VPN supports the ability to specify exclusion routes that specifically control routing behavior to define which traffic should traverse the VPN only and not go over the physical network interface.

  > ⓘ Note
  >
  > Exclusion routes work for traffic within the same subnet as the client such as LinkLocal. Exclusion routes only work in a Split Tunnel setup.

- **Support for multiple domains and forests.** The Always On VPN platform has no dependency on Active Directory Domain Services (AD DS) forests or domain topology (or associated functional/schema levels) because it doesn't require the VPN client to be domain joined to function. Group Policy is therefore not a dependency to define VPN profile settings because you don't use it during client configuration. Where Active Directory authorization integration is required, you can achieve it through RADIUS as part of the EAP authentication and authorization process.

- **Name resolution of corporate resources** using short-name, fully qualified domain name (FQDN), and DNS suffix.Always On VPN can natively define one or more DNS suffixes as part of the VPN connection and IP address assignment process, including corporate resource name resolution for short names, FQDNs, or entire DNS namespaces. Always On VPN also supports the use of Name Resolution Policy Tables to provide namespace-specific resolution granularity.

  > ⓘ Note
  >
  > Avoid the use of Global Suffixes as they interfere with shortname resolution when using Name Resolution Policy tables.

High availability features

The following are more options for high availability.

Server resilience and load balancing. In environments that require high availability or support large numbers of requests, you can increase the performance and resiliency of Remote Access by configuring load balancing between Network Policy Servers (NPS) and by enabling Remote Access server clustering.

Geographic site resilience. For IP-based geolocation, you can use Global Traffic Manager with DNS in Windows Server. For more robust geographic load balancing, you can use Global Server Load Balancing solutions, such as Microsoft Azure Traffic Manager.

Always On VPN connection process

The process is composed of the following steps:

1. The Windows VPN client uses a public DNS server to perform a name resolution query for the IP address of the VPN gateway.
2. The VPN client uses the IP address returned by DNS to send a connection request to the VPN gateway.
3. The VPN server is also configured as a Remote Authentication Dial-In User Service (RADIUS) Client; the VPN RADIUS Client sends the connection request to the NPS server for connection request processing.
4. The NPS server processes the connection request, including performing authorization and authentication, and determines whether to allow or deny the connection request.
5. The NPS server forwards an Access-Accept or Access-Deny response to the VPN server.
6. The connection is initiated or terminated based on the response that the VPN server received from the NPS server.

Next steps

- Install Remote Access as a VPN server
- Tutorial: Deploy Always On VPN
- VPN security features: This topic provides an overview of VPN security guidelines for LockDown VPN, Windows Information Protection (WIP) integration with VPN, and traffic filters.
- VPN auto-triggered profile options: This topic provides an overview of VPN auto-triggered profile options, such as app trigger, name-based trigger, and Always On.
- Troubleshoot Always On VPN

Last updated on 05/07/2025

DirectAccess

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

> **ⓘ Important**
>
> Microsoft highly recommends that you use Always On VPN instead of DirectAccess for new deployments. For more information, see Always on VPN.

You can use this topic for a brief overview of DirectAccess, including the server and client operating systems that support DirectAccess, and for links to additional DirectAccess documentation for Windows Server.

> **ⓘ Note**
>
> In addition to this topic, the following DirectAccess documentation is available.
>
> - DirectAccess Deployment Paths in Windows Server
> - Prerequisites for Deploying DirectAccess
> - DirectAccess Unsupported Configurations
> - DirectAccess Test Lab Guides
> - DirectAccess Known Issues
> - DirectAccess Capacity Planning
> - DirectAccess Offline Domain Join
> - Troubleshooting DirectAccess
> - Deploy a Single DirectAccess Server Using the Getting Started Wizard
> - Deploy a Single DirectAccess Server with Advanced Settings
> - Add DirectAccess to an Existing Remote Access (VPN) Deployment

DirectAccess allows connectivity for remote users to organization network resources without the need for traditional Virtual Private Network (VPN) connections. With DirectAccess connections, remote client computers are always connected to your organization - there is no need for remote users to start and stop connections, as is required with VPN connections. In addition, your IT administrators can manage DirectAccess client computers whenever they are running and Internet connected.

> **Important**
>
> Do not attempt to deploy Remote Access on a virtual machine (VM) in Microsoft Azure. Using Remote Access in Microsoft Azure is not supported. You cannot use Remote Access in an Azure VM to deploy VPN, DirectAccess, or any other Remote Access feature in Windows Server. For more information, see Microsoft server software support for Microsoft Azure virtual machines.

DirectAccess provides support only for domain-joined clients that include operating system support for DirectAccess.

The following client operating systems support DirectAccess.

- Windows 11 Enterprise
- Windows 10 Enterprise
- Windows 10 Enterprise 2015 Long Term Servicing Branch (LTSB)
- Windows 8.1 Enterprise

Feedback

Was this page helpful? Yes No

What is SMB File Sharing for Windows and Windows Server?

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10, ✅ Azure Local 2311.2 and later

This article provides an overview of the Server Message Block (SMB) protocol in Windows. It covers practical applications, features, and key concepts for file sharing.

The SMB protocol is a network file sharing protocol. SMB allows you to read and write to files over the network on top of TCP/IP or other protocols. With SMB, an application (or the user of an application) can access files or other resources at a remote server, allowing applications to read, create, and update files on the remote server. SMB can also communicate with any server program that is set up to receive an SMB client request.

SMB is also a fabric protocol for hyperconverged and clustered storage solutions such as Storage Spaces Direct and Storage Replica. For more information, see SMB as a foundation for Windows Server services.

Usage

This section discusses some practical ways to use the SMB protocol.

- **File storage for virtualization (Hyper-V™ over SMB)**. Hyper-V can store virtual machine files, such as configuration, Virtual hard disk (VHD) files, and snapshots, in file shares over the SMB 3.0 protocol. You can use this feature for both stand-alone file servers and clustered file servers that use Hyper-V together with shared file storage for the cluster.
- **Microsoft SQL Server over SMB**. SQL Server can store user database files on SMB file shares. Currently, SQL Server over SMB is supported with SQL Server 2008 R2 for stand-alone SQL servers. Upcoming versions of SQL Server add support for clustered SQL servers and system databases.
- **Traditional storage for end-user data**. The SMB 3.0 protocol provides enhancements to the Information Worker (or client) workloads. These enhancements include reducing the application latencies experienced by branch office users when accessing data over wide area networks (WAN) and protecting data from eavesdropping attacks.

> **Note**
>
> If you need to conserve storage space on an SMB file share, consider using Azure File Sync with cloud tiering enabled. Azure File Sync allows you to cache your most frequently

accessed files locally and tier your least frequently accessed files to the cloud, saving local storage space while maintaining performance. For details, see Planning for an Azure File Sync deployment.

SMB features

SMB includes a wide range of features organized into the following categories:

- **Security and authentication** - SMB provides multiple layers of protection including encryption, signing, NTLM blocking, and authentication rate limiting. These features protect data in transit and help prevent unauthorized access and brute-force attacks.
- **Networking and connectivity** - SMB supports multiple transport options including traditional TCP, QUIC for secure internet access without VPN, and RDMA for high-performance datacenter workloads. Features like alternative ports and firewall hardening provide other flexibility and security.
- **Performance** - SMB includes features to optimize file transfer speeds and reduce network overhead, including compression, bandwidth limits, directory caching, and WAN optimizations.
- **High availability and scale** - For enterprise environments, SMB supports transparent failover, scale-out file servers, and rolling cluster upgrades to ensure continuous access to file shares during maintenance or failures.
- **Management and monitoring** - Administrators can manage SMB using PowerShell cmdlets, control dialect versions, and monitor performance through dedicated counters.

For detailed information about individual features and their availability across Windows versions, see SMB features in Windows and Windows Server.

SMB components

SMB communication involves two components: the SMB client and the SMB server. These terms refer to the SMB protocol roles, not the Windows operating system editions.

- **SMB server** is the component that shares resources (files, printers, named pipes) and responds to requests from SMB clients. When you create a file share on a computer, that computer's SMB server handles incoming connections and file access requests.
- **SMB client** is the component that connects to remote shares and sends requests to SMB servers. When you map a network drive or access a UNC path like `\\server\share`, your

computer's SMB client initiates and manages that connection.

Both Windows Server and Windows client operating systems include the SMB client and SMB server components. Meaning a Windows client computer, such as Windows 11, can host file shares (acting as an SMB server) while also connecting to shares on other computers (acting as an SMB client). Similarly, Windows Server can connect to shares hosted on other servers or client computers.

Feature availability varies between operating system versions. Some features are available only in the SMB server component, some only in the SMB client component, and some in both. Additionally, certain features might be available in Windows Server but not in Windows client editions, or vice versa. For more information about feature availability across different Windows versions, see SMB features in Windows and Windows Server.

SMB dialects

An SMB dialect is a specific version of the SMB protocol that defines the features, capabilities, and message formats supported during communication between a client and server. When you establish an SMB connection, the client and server negotiate to use the highest dialect version that both support. Higher dialect versions provide more features such as improved security, better performance, and new capabilities, while maintaining backward compatibility with older versions.

The following table shows which SMB dialect version is supported by each Windows operating system release.

Expand table

SMB Dialect	Windows Client	Windows Server
SMB 3.1.1	Windows 10 version 1607 and later	Windows Server 2016 and later
SMB 3.02	Windows 8.1	Windows Server 2012 R2
SMB 3.0	Windows 8	Windows Server 2012

> **Note**
>
> While multiple Windows versions support the same SMB dialect (3.1.1), each version introduces more features and enhancements beyond the base protocol. For more information about version-specific improvements and feature availability, see SMB features in Windows and Windows Server.

SMB as a foundation

SMB serves as the underlying protocol for many Windows Server storage and file services. These services build on SMB to provide more capabilities while inheriting SMB's security, performance, and reliability features.

Hyperconverged and clustered storage

SMB is a fabric protocol for hyperconverged infrastructure (HCI) and clustered storage solutions:

- **Storage Spaces Direct** uses SMB 3 for node-to-node communication, including data replication and cache synchronization in hyperconverged and disaggregated deployments. SMB Direct (RDMA) provides high-throughput, low-latency connectivity between cluster nodes. For more information, see Storage Spaces Direct overview.
- **Storage Replica** uses SMB 3 to replicate data between servers or clusters for disaster recovery. SMB encryption protects replicated data in transit. For more information, see Storage Replica overview.
- **Disaggregated storage architectures** separate compute and storage into distinct clusters, with compute clusters accessing storage over SMB 3. This architecture enables independent scaling of compute and storage resources. For more information, see Failover clustering storage architectures.

File server technologies

Several file server features rely on SMB for file access:

- **Scale-Out File Server** provides continuously available file shares across multiple cluster nodes using SMB 3. Applications like Hyper-V and SQL Server can store data on these shares with transparent failover. For more information, see Scale-Out File Server overview.
- **DFS Namespaces** organizes shared folders across multiple servers into a logical namespace. While DFS Namespaces manages the namespace and referrals, the actual file access uses SMB. For more information, see DFS Namespaces overview.
- **DFS Replication** replicates folders between servers for redundancy and branch office scenarios. The replicated content is accessed through SMB file shares. For more information, see DFS Replication overview.

Virtualization and application workloads

SMB 3 enables high-performance storage for virtualized environments:

- **Hyper-V over SMB** allows you to store virtual machine files (configuration, VHD/VHDX, and checkpoints) on SMB 3 file shares instead of local storage. This feature simplifies storage management and enables features like live migration. For more information, see Deploy Hyper-V over SMB.
- **SQL Server over SMB** supports storing database files on SMB 3 shares, so you can benefit from SMB Multichannel and SMB Direct for performance.

These services benefit from SMB features like encryption, signing, Multichannel, and Direct (RDMA). When you enable SMB security features, the services that depend on SMB inherit those protections.

Event logging and auditing

SMB provides event logging and auditing capabilities to help administrators monitor file sharing activity, troubleshoot connectivity issues, and detect potential security threats.

Windows records SMB events in the Windows Event Log under **Applications and Services Logs** > **Microsoft** > **Windows** > **SMBClient** and **SMBServer**. These logs capture connection events, authentication failures, share access, and protocol negotiation details. Administrators can use these logs to diagnose connectivity problems, identify misconfigured clients, or investigate unauthorized access attempts.

Beginning with Windows Server 2025 and Windows 11 24H2, SMB includes enhanced auditing capabilities for signing and encryption. Administrators can audit whether clients and servers support SMB signing and encryption, helping identify noncompliant devices that might pose security risks. You can configure this auditing through Group Policy or PowerShell.

For more information about SMB security auditing, see SMB security enhancements.

Related content

The following list provides more resources about SMB and related technologies in Windows Server 2012 R2, Windows Server 2012, and Windows Server 2016.

- Storage in Windows Server
- Scale-Out File Server for Application Data
- Improve Performance of a File Server with SMB Direct
- Deploy Hyper-V over SMB
- Deploy SMB Multichannel

- Deploying Fast and Efficient File Servers for Server Applications
- SMB: Troubleshooting Guide

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 11/27/2025

SMB Direct

Article • 01/16/2025

Windows Server includes a feature called Server Message Block (SMB) Direct, which supports the use of network adapters that have Remote Direct Memory Access (RDMA) capability. Network adapters that have RDMA can function at full speed with lower latency without compromising CPU utilization. For workloads such as Hyper-V or Microsoft SQL Server, this feature enables a remote file server to resemble local storage. SMB Direct is automatically configured and enabled by default in Windows Server 2012 and future iterations.

Using SMB Direct provides:

- **Increased throughput**: Applies the full throughput of high speed networks where the network adapters coordinate the transfer of large amounts of data at line speed.
- **Low latency**: Provides fast responses to network requests, which makes remote file storage feel like directly attached block storage.
- **Low CPU utilization**: Uses fewer CPU cycles when transferring data over the network, which leaves more power available to server applications.

You can use SMB Direct in a failover cluster; however, you need to make sure that the cluster networks used for client access are adequate for SMB Direct. Failover clustering supports using multiple networks for client access, along with network adapters that are RSS (Receive Side Scaling)-capable and RDMA-capable.

> **Note**
>
> You can use SMB Direct on the Hyper-V management operating system to support using Hyper-V over SMB, and to provide storage to a virtual machine that uses the Hyper-V storage stack. However, RDMA-capable network adapters aren't directly exposed to a Hyper-V client. Even if you connect an RDMA-capable network adapter to a virtual switch, the virtual network adapters created by the switch can't support RDMA.

Requirements

The following are SMB Direct requirements:

- One or more network adapters with RDMA capability.

- At least two computers running one or more of the following operating systems:
  - Windows Server 2012 and later.
  - Windows 10 Enterprise and later.
  - Windows 10 Education and later.
  - Windows 10 Pro for Workstations and later
  - Windows 10 Pro 22H2 (clean install, not upgraded from older versions) and later.

> **Note**
>
> Windows 10 and Windows 11 family are restricted to client-only and can't act as an SMB Direct server.

SMB Multichannel

SMB Multichannel is the feature responsible for detecting the RDMA capabilities of network adapters to enable SMB Direct. Without SMB Multichannel, SMB uses regular TCP/IP with the RDMA-capable network adapters (all network adapters provide a TCP/IP stack along with the new RDMA stack).

With SMB Multichannel, SMB detects whether a network adapter has RDMA capability, and creates multiple RDMA connections for that single session (two per interface). This feature allows SMB to use the high throughput, low latency, and low CPU utilization offered by RDMA-capable network adapters. It also offers fault tolerance if you're using multiple RDMA interfaces.

You can team RDMA-capable network adapters using Switch Embedded Teaming (SET) starting with Windows Server 2016. After at least one RDMA network connection is created, the TCP/IP connection used for the original protocol negotiation is no longer used. However, the TCP/IP connection is retained in case the RDMA network connections fail.

Disabling SMB Multichannel also disables SMB Direct. SMB Multichannel detects network adapter capabilities and determines whether a network adapter is RDMA-capable. Client devices can't use SMB Direct if SMB Multichannel is disabled.

SMB Encryption

Beginning in Windows Server 2022 and Windows 11, SMB Direct now supports encryption. Previously, enabling SMB encryption disabled direct data placement, making RDMA performance as slow as TCP. Now data is encrypted before placement, leading to

relatively minor performance degradation while adding AES-128 and AES-256 protected packet privacy. For more information on configuring SMB encryption, see SMB security enhancements.

Furthermore, Windows Server failover clusters now support granular control of encrypting intra-node storage communications for Cluster Shared Volumes (CSV) and the storage bus layer (SBL). This means that when using Storage Spaces Direct and SMB Direct, you can decide to encrypt east-west communications within the cluster itself for higher security.

Disabling and enabling SMB Direct features

The SMB client automatically detects and uses multiple network connections if an appropriate configuration is identified. As SMB Direct is enabled by default, once disabled, it needs to be manually re-enabled whenever needed.

Disable

Typically, you don't need to disable SMB Direct, however, you can disable it along with its features, by running the following Windows PowerShell commands.

To disable SMB Direct, type:

PowerShell

```
Disable-WindowsOptionalFeature -Online -FeatureName SMBDirect
```

To disable SMB Multichannel on the server-side, type:

PowerShell

```
Set-SmbServerConfiguration -EnableMultiChannel $false
```

To disable SMB Multichannel on the client-side, type:

PowerShell

```
Set-SmbClientConfiguration -EnableMultiChannel $false
```

To disable RDMA for a specific interface, type:

PowerShell

```
Disable-NetAdapterRdma <name>
```

To disable RDMA for all interfaces, type:

```
PowerShell
Set-NetOffloadGlobalSetting -NetworkDirect Disabled
```

When you disable RDMA on either the client or the server, the systems can't use it. *Network Direct* is the internal name for Windows Server basic networking support for RDMA interfaces.

To verify which state of operability SMB Direct is currently configured to, run the following cmdlet:

```
PowerShell
Get-WindowsOptionalFeature -Online -FeatureName SMBDirect
```

Testing SMB Direct

You can test the performance of SMB Direct by measuring the throughput when running a large file copy. Before testing, verify that the network adapter supports RDMA using PowerShell.

On the server-side, type:

```
PowerShell
Get-SmbServerNetworkInterface
```

On the client-side, type:

```
PowerShell
Get-SmbClientNetworkInterface
```

Once the network adapter is verified RDMA-capable, perform the following actions:

1. Disable RDMA on the network adapter, see Disabling and Enabling SMB Direct features.

2. Measure the amount of time taken to run a large file copy without using SMB Direct.
3. Re-enable RDMA on the network adapter, perform the same file copy, and then compare the two results.
4. To avoid the impact of caching, perform the following actions:
   a. Copy a large amount of data (more data than memory is capable of handling).
   b. Copy the data twice, with the first copy as practice and then timing the second copy.
   c. Restart both the server and the client before each test to ensure they operate under similar conditions.

Additionally, you can observe the performance counters during testing using the same scenario utilizing the Performance Monitor tool by performing the following steps:

1. Select **Start**, type **perfmon**, hit `Enter`.
2. In the left pane, under **Monitoring Tools** > select **Performance Monitor**.
3. In the right pane, select the green "+" icon to add a new counter.
4. In the **Add Counters** dialog box, expand **SMB Direct Connection**.
5. Select both **Bytes RDMA Read/sec** and **Bytes RDMA Written/sec**, select **Add**, then select **OK**.

SMB Direct failover capability

Here's how to confirm the failover capability of SMB Direct:

1. Ensure that SMB Direct is functioning in a multiple network adapter configuration.
2. Run a large file copy. During the copy process, simulate a failure of one of the network paths by disconnecting one of the cables or by disabling one of the network adapters.
3. Confirm that the file copying continues using one of the remaining network adapters, and that there are no file copy errors.

> **Tip**
>
> To avoid failures of a workload that doesn't use SMB Direct, make sure there are no other workloads using the disconnected network path.

See also

- Server Message Block overview

- Increasing Server, Storage, and Network Availability: Scenario Overview
- Deploy Hyper-V over SMB

Feedback

Was this page helpful? Yes No

Microsoft Defender Antivirus in Windows Overview

Applies to: Microsoft Defender for Business, Microsoft Defender Antivirus

Microsoft Defender Antivirus is available in Windows 10 and Windows 11, and in versions of Windows Server.

Microsoft Defender Antivirus is a major component of your next-generation protection in Microsoft Defender for Endpoint. This protection brings together machine learning, big-data analysis, in-depth threat resistance research, and the Microsoft cloud infrastructure to protect devices (or endpoints) in your organization. Microsoft Defender Antivirus is built into Windows, and it works with Microsoft Defender for Endpoint to provide protection on your device and in the cloud.

> **Tip**
>
> As a companion to this article, see our Security Analyzer setup guide to review best practices and learn to fortify defenses, improve compliance, and navigate the cybersecurity landscape with confidence. For a customized experience based on your environment, you can access the Security Analyzer automated setup guide in the Microsoft 365 admin center.

Prerequisites

Supported operating systems

- Windows

Microsoft Defender Antivirus capabilities

Microsoft Defender Antivirus provides anomaly detection, a layer of protection for malware that doesn't fit any predefined pattern. Anomaly detection monitors for process creation events or files that are downloaded from the internet. Through machine learning and cloud-delivered protection, Microsoft Defender Antivirus can stay one step ahead of attackers. Anomaly detection is on by default and can help block attacks such as 3CX Security Alert for

Electron Windows App. Microsoft Defender Antivirus started blocking this malware four days before the attack was registered in VirusTotal.

Modern malware requires modern solutions. In 2015, Microsoft Defender Antivirus moved away from using a static signature-based engine to a model that uses predictive technologies--such as machine learning, applied science, and artificial intelligence--as this switch is what's necessary to keep you and your organizations safe from the complexity of today's ever-evolving malware landscape.

Microsoft Defender Antivirus can block almost all malware at first sight, in milliseconds.

We designed our antivirus solution to work in both online and offline scenarios. For offline scenarios, the latest dynamic intelligence from the Intelligence Security Graph is provisioned to the endpoint regularly throughout the day. When connected to the cloud, real-time intelligence gets fed from the Intelligent Security Graph.

Microsoft Defender Antivirus can also stop threats based on their behaviors and process trees even when the threat has started execution. A common example of these kinds of attacks is fileless malware. Microsoft's Next-generation protection features work together to identify and block malware based on abnormal behavior. To learn more, see Behavioral blocking and containment.

Compatibility with other antivirus products

If you're using a non-Microsoft antivirus/anti-malware product on your device, you might be able to run Microsoft Defender Antivirus in passive mode alongside the non-Microsoft antivirus solution. It depends on the operating system used and whether your device is onboarded to Defender for Endpoint. To learn more, see Microsoft Defender Antivirus compatibility.

Microsoft Defender Antivirus processes and services

The following table summarizes Microsoft Defender Antivirus processes and services. You can view them in Task Manager in Windows.

Expand table

Process or service	Where to view its status
Microsoft Defender Antivirus Core service (`MdCoreSvc`)	• **Processes** tab: `Antimalware Core Service` • **Details** tab: `MpDefenderCoreService.exe` • **Services** tab: `Microsoft Defender Core Service`
Microsoft Defender Antivirus service (`WinDefend`)	• **Processes** tab: `Antimalware Service Executable` • **Details** tab: `MsMpEng.exe` • **Services** tab: `Microsoft Defender Antivirus`
Microsoft Defender Antivirus Network Realtime Inspection service (`WdNisSvc`)	• **Processes** tab: `Microsoft Network Realtime Inspection Service` • **Details** tab: `NisSrv.exe` • **Services** tab: `Microsoft Defender Antivirus Network Inspection Service`
Microsoft Defender Antivirus command-line utility	• **Processes** tab: N/A • **Details** tab: `MpCmdRun.exe` • **Services** tab: N/A
Microsoft Security Client Policy Configuration Tool	• **Processes** tab: N/A • **Details** tab: `ConfigSecurityPolicy.exe` • **Services** tab: N/A

To learn more about the Microsoft Defender Core service, visit Microsoft Defender Core service overview.

For Microsoft Endpoint Data Loss Prevention (Endpoint DLP), the following table summarizes processes and services. You can view them in Task Manager in Windows.

Expand table

Process or service	Where to view its status
Microsoft Endpoint DLP service (`MDDlpSvc`)	• **Processes** tab: `MpDlpService.exe` • **Details** tab: `MpDlpService.exe` • **Services** tab: `Microsoft Data Loss Prevention Service`
Microsoft Endpoint DLP command-line utility	• **Processes** tab: N/A • **Details** tab: `MpDlpCmd.exe` • **Services** tab: N/A

Comparing active mode, passive mode, and disabled mode

The following table describes what to expect when Microsoft Defender Antivirus is in active mode, passive mode, or disabled.

Expand table

Mode	What happens
Active mode	In active mode, Microsoft Defender Antivirus is used as the primary antivirus app on the device. Files are scanned, threats are remediated, and detected threats are listed in your organization's security reports and in your Windows Security app.
Passive mode	In passive mode, Microsoft Defender Antivirus isn't used as the primary antivirus app on the device. Files are scanned, and detected threats are reported, but threats aren't remediated by Microsoft Defender Antivirus. **IMPORTANT**: Microsoft Defender Antivirus can run in passive mode only on endpoints that are onboarded to Microsoft Defender for Endpoint. See Requirements for Microsoft Defender Antivirus to run in passive mode.
Disabled or uninstalled	When disabled or uninstalled, Microsoft Defender Antivirus isn't used. Files aren't scanned, and threats aren't remediated. In general, we don't recommend disabling or uninstalling Microsoft Defender Antivirus.

To learn more, see Microsoft Defender Antivirus compatibility.

Check the state of Microsoft Defender Antivirus on your device

You can use one of several methods, such as the Windows Security app or Windows PowerShell, to check the state of Microsoft Defender Antivirus on your device.

> **Important**
>
> Beginning with platform version 4.18.2208.0 and later: If a server has been onboarded to Microsoft Defender for Endpoint, the "Turn off Windows Defender" group policy setting no longer completely disables Windows Defender Antivirus on Windows Server 2012 R2 and later. Instead, it places it into passive mode. In addition, the tamper protection feature allows a switch to active mode but not to passive mode.

- If "Turn off Windows Defender" is already in place before onboarding to Microsoft Defender for Endpoint, there is no change and Defender Antivirus remains disabled.
- To switch Defender Antivirus to passive mode, even if it was disabled before onboarding, you can apply the ForceDefenderPassiveMode configuration with a value of `1`. To place it into active mode, switch this value to `0` instead.

Understand the modified logic for `ForceDefenderPassiveMode` when tamper protection is enabled: Once Microsoft Defender Antivirus is toggled to active mode, tamper protection prevents it from going back into passive mode even when `ForceDefenderPassiveMode` is set to `1`.

Use the Windows Security app to check the status of Microsoft Defender Antivirus

1. On your Windows device, select the **Start** menu, and begin typing `Security`. Then open the Windows Security app in the results.
2. Select **Virus & threat protection**.
3. Under **Who's protecting me?**, choose **Manage Providers**.

You'll see the name of your antivirus/anti-malware solution on the security providers page.

Use PowerShell to check the status of Microsoft Defender Antivirus

1. Select the **Start** menu, and begin typing `PowerShell`. Then open Windows PowerShell in the results.
2. Type `Get-MpComputerStatus`.
3. In the list of results, look at the **AMRunningMode** row.
    - **Normal** means Microsoft Defender Antivirus is running in active mode.
    - **Passive mode** means Microsoft Defender Antivirus running, but isn't the primary antivirus/anti-malware product on your device. Passive mode is only available for devices that are onboarded to Microsoft Defender for Endpoint and that meet certain requirements. To learn more, see Requirements for Microsoft Defender Antivirus to run in passive mode.

- **EDR Block Mode** means Microsoft Defender Antivirus is running and Endpoint detection and response (EDR) in block mode, a capability in Microsoft Defender for Endpoint, is enabled. Check the **ForceDefenderPassiveMode** registry key. If its value is 0, it's running in normal mode; otherwise, it's running in passive mode.

- **SxS Passive Mode** means Microsoft Defender Antivirus is running alongside another antivirus/anti-malware product, and limited periodic scanning is used.

> **Tip**
>
> To learn more about the Get-MpComputerStatus PowerShell cmdlet, see the reference article **Get-MpComputerStatus**.
>
> **Performance tip** Due to a variety of factors (examples listed below) Microsoft Defender Antivirus, like other antivirus software, can cause performance issues on endpoint devices. In some cases, you might need to tune the performance of Microsoft Defender Antivirus to alleviate those performance issues. Microsoft's **Performance analyzer** is a PowerShell command-line tool that helps determine which files, file paths, processes, and file extensions might be causing performance issues; some examples are:
>
> - Top paths that impact scan time
> - Top files that impact scan time
> - Top processes that impact scan time
> - Top file extensions that impact scan time
> - Combinations – for example:
>   - top files per extension
>   - top paths per extension
>   - top processes per path
>   - top scans per file
>   - top scans per file per process
>
> You can use the information gathered using Performance analyzer to better assess performance issues and apply remediation actions. See: **Performance analyzer for Microsoft Defender Antivirus**.

Get your antivirus/anti-malware platform updates

It's important to keep Microsoft Defender Antivirus (or any antivirus/anti-malware solution) up to date. Microsoft releases regular updates to help ensure that your devices have the latest technology to protect against new malware and attack techniques. To learn more, see Manage Microsoft Defender Antivirus updates and apply baselines.

> **Tip**
>
> - If you're looking for Antivirus related information for other platforms, see:
>   - Set preferences for Microsoft Defender for Endpoint on macOS
>   - Microsoft Defender for Endpoint on Mac
>   - macOS Antivirus policy settings for Microsoft Defender Antivirus for Intune
>   - Set preferences for Microsoft Defender for Endpoint on Linux
>   - Microsoft Defender for Endpoint on Linux
>   - Configure Defender for Endpoint on Android features
>   - Configure Microsoft Defender for Endpoint on iOS features
> - After you install the latest Microsoft Defender Antivirus platform or engine update, certain registry entries might not update automatically. To ensure the registry reflects the current version, admins should manually verify and update the relevant registry keys using the **Registry Editor (regedit)** or a supported deployment script.

See also

- Performance analyzer for Microsoft Defender Antivirus
- Microsoft Defender Antivirus management and configuration
- Evaluate Microsoft Defender Antivirus protection
- Exclusions for Microsoft Defender for Endpoint and Microsoft Defender Antivirus

Last updated on 03/25/2026

Attack surface reduction rules overview

Applies to: Microsoft Defender for Endpoint Plan 1, Microsoft Defender for Endpoint Plan 2

> **Tip**
>
> As a companion to this article, see our Security Analyzer setup guide to review best practices and learn to fortify defenses, improve compliance, and navigate the cybersecurity landscape with confidence. For a customized experience based on your environment, you can access the Security Analyzer automated setup guide in the Microsoft 365 admin center.

Why attack surface reduction rules are important

Your organization's attack surface includes all the places where an attacker could compromise your organization's devices or networks. Reducing your attack surface means protecting your organization's devices and network, which leaves attackers with fewer ways to perform attacks. Configuring attack surface reduction rules in Microsoft Defender for Endpoint can help!

Attack surface reduction rules target certain software behaviors, such as:

- Launching executable files and scripts that attempt to download or run files
- Running obfuscated or otherwise suspicious scripts
- Performing behaviors that apps don't usually initiate during normal day-to-day work

Such software behaviors are sometimes seen in legitimate applications. However, these behaviors are often considered risky because they're commonly abused by attackers through malware. Attack surface reduction rules can constrain software-based risky behaviors and help keep your organization safe.

For a sequential, end-to-end process of how to manage attack surface reduction rules, see:

- Attack surface reduction rules deployment overview
- Plan attack surface reduction rules deployment
- Test attack surface reduction rules
- Enable attack surface reduction rules
- Operationalize attack surface reduction rules

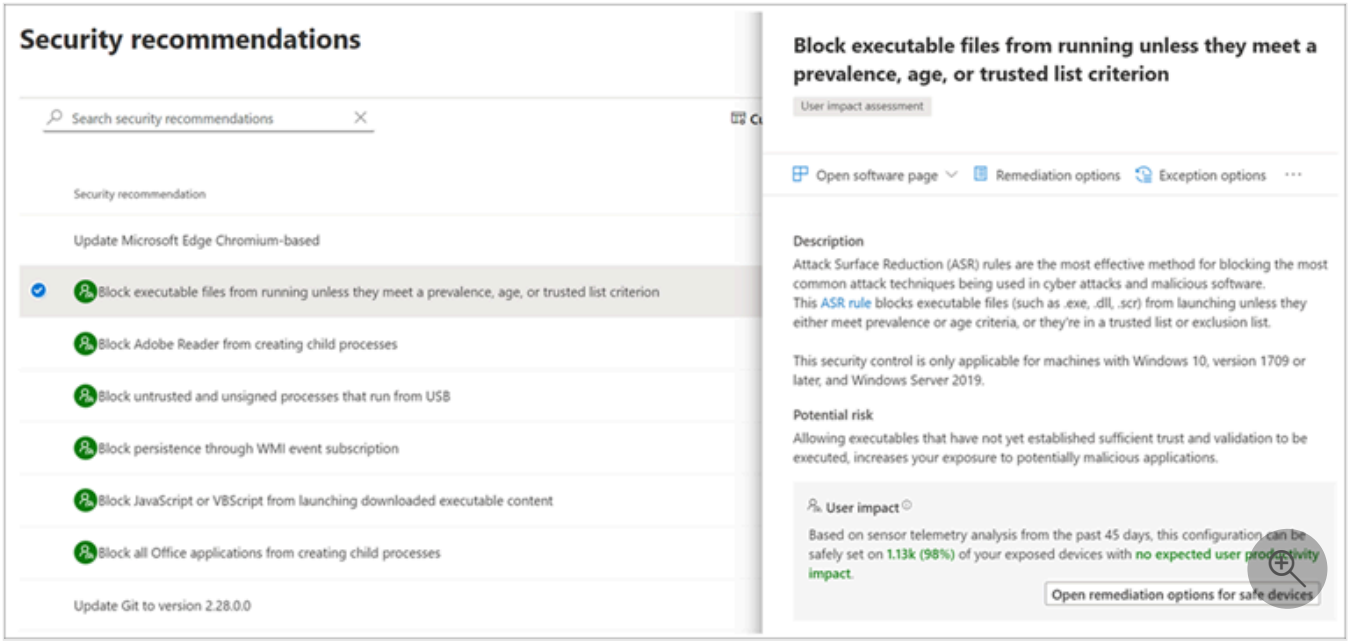
Prerequisites

Supported operating systems

- Windows

Assess rules before deployment

You can assess how an attack surface reduction rule might affect your network by opening the security recommendation for that rule in Microsoft Defender Vulnerability Management.



In the recommendation details pane, check for user impact to determine what percentage of your devices can accept a new policy enabling the rule in blocking mode without adversely affecting productivity.

See Requirements in the "Enable attack surface reduction rules" article for information about supported operating systems and other requirement information.

Audit mode for evaluation

Audit mode

Use audit mode to evaluate how attack surface reduction rules would affect your organization if enabled. Run all rules in audit mode first so you can understand how they affect your line-of-business applications. Many line-of-business applications are written with limited security concerns, and they might perform tasks in ways that seem similar to malware.

Exclusions

By monitoring audit data and adding exclusions for necessary applications, you can deploy attack surface reduction rules without reducing productivity.

Per-rule exclusions

For information about configuring per-rule exclusions, see Configure attack surface reduction per-rule exclusions.

Warn mode for users

Whenever an attack surface reduction rule blocks content, users see a dialog box that indicates the content is blocked. The dialog box also offers the user an option to unblock the content. The user can then retry their action, and the operation completes. When a user unblocks content, the content remains unblocked for 24 hours, and then blocking resumes.

Warn mode helps your organization have attack surface reduction rules in place without preventing users from accessing the content they need to perform their tasks.

Requirements for warn mode to work

Warn mode is supported on devices running the following versions of Windows:

- Windows 10, version 1809 or later
- Windows 11
- Windows Server, version 1809 or later

Microsoft Defender Antivirus must be running with real-time protection in Active mode.

Also, make sure Microsoft Defender Antivirus and anti-malware updates are installed.

- Minimum platform release requirement: `4.18.2008.9`
- Minimum engine release requirement: `1.1.17400.5`

For more information and to get your updates, see Update for Microsoft Defender anti-malware platform.

Cases where warn mode isn't supported

Warn mode isn't supported for three attack surface reduction rules when you configure them in Microsoft Intune. (If you use Group Policy to configure your attack surface reduction rules, warn mode is supported.) The three rules that don't support warn mode when you configure them in Microsoft Intune are as follows:

- Block JavaScript or VBScript from launching downloaded executable content (GUID `d3e037e1-3eb8-44c8-a917-57927947596d`)
- Block persistence through WMI event subscription (GUID `e6db77e5-3df2-4cf1-b95a-636979351e5b`)
- Use advanced protection against ransomware (GUID `c1db55ab-c21a-4637-bb3f-a12568109d35`)

Also, warn mode isn't supported on devices running older versions of Windows. In those cases, attack surface reduction rules that are configured to run in warn mode runs in block mode.

Notifications and alerts

Whenever an attack surface reduction rule is triggered, a notification is displayed on the device. You can customize the notification with your company details and contact information.

Also, when certain attack surface reduction rules are triggered, alerts are generated.

Notifications and any alerts that are generated can be viewed in the Microsoft Defender portal.

For specific details about notification and alert functionality, see: Per rule alert and notification details, in the article **Attack surface reduction rules reference**.

Advanced hunting and attack surface reduction events

You can use advanced hunting to view attack surface reduction events. To streamline the volume of incoming data, only unique processes for each hour are viewable with advanced hunting. The time of an attack surface reduction event is the first time that event is seen within the hour.

For example, suppose that an attack surface reduction event occurs on 10 devices during the 2:00 PM hour. Suppose that the first event occurred at 2:15, and the last at 2:45. With advanced hunting, you see one instance of that event (even though it actually occurred on 10 devices), and its timestamp is 2:15 PM.

For more information about advanced hunting, see Proactively hunt for threats with advanced hunting.

Attack surface reduction features across Windows versions

You can set attack surface reduction rules for devices that are running any of the following editions and versions of Windows:

- Windows 10 Pro, version 1709 or later
- Windows 10 Enterprise, version 1709 or later
- Windows 11 Pro, version 21H2 or later
- Windows 11 Enterprise, version 21H2 or later
- Windows Server, version 1803 (Semi-Annual Channel) or later
- Windows Server 2025
- Windows Server 2022
- Windows Server 2019
- Windows Server 2016
- Windows Server 2012 R2
- Azure Stack HCI OS, version 23H2 and later

> **Note**
>
> Windows Server 2016 and Windows Server 2012 R2 must be onboarded using the instructions in **Onboard Windows Server 2012 R2 and Windows Server 2016 to Microsoft Defender for Endpoint** for this feature to work.

Although attack surface reduction rules don't require a Windows E5 license, if you have Windows E5, you get advanced management capabilities. The advanced capabilities - available only in Windows E5 - include:S

- The monitoring, analytics, and workflows available in Defender for Endpoint

- The reporting and configuration capabilities in Microsoft Defender XDR.

These advanced capabilities aren't available with a Windows Professional or Windows E3 license. However, if you do have those licenses, you can use Event Viewer and Microsoft Defender Antivirus logs to review your attack surface reduction rule events.

Review attack surface reduction events in the Microsoft Defender portal

Defender for Endpoint provides detailed reporting for events and blocks as part of alert investigation scenarios.

You can query Defender for Endpoint data in Microsoft Defender XDR by using advanced hunting.

Here's an example query:

Kusto

```
DeviceEvents
| where ActionType startswith 'Asr'
```

Review attack surface reduction events in Windows Event Viewer

You can review the Windows event log to view events generated by attack surface reduction rules:

1. Download the Evaluation Package and extract the file *cfa-events.xml* to an easily accessible location on the device.
2. Enter the words, *Event Viewer*, into the Start menu to open the Windows Event Viewer.
3. Under **Actions**, select **Import custom view...**.
4. Select the file *cfa-events.xml* from where it was extracted. Alternatively, copy the XML directly.
5. Select **OK**.

You can create a custom view that filters events to only show the following events, all of which are related to controlled folder access:

Expand table

Event ID	Description
5007	Event when settings are changed
1121	Event when rule fires in Block-mode
1122	Event when rule fires in Audit-mode

Defender for Endpoint generates the "engine version" that's listed in the event log for attack surface reduction. The operating system doesn't generate this version. Defender for Endpoint is integrated with Windows 10 and Windows 11, so this feature works on all devices with Windows 10 or Windows 11 installed.

See also

- Attack surface reduction rules deployment overview
- Plan attack surface reduction rules deployment
- Test attack surface reduction rules
- Enable attack surface reduction rules
- Operationalize attack surface reduction rules
- Attack surface reduction rules report
- Exclusions for Microsoft Defender for Endpoint and Microsoft Defender Antivirus

Tip

If you're looking for Antivirus related information for other platforms, see:

- Set preferences for Microsoft Defender for Endpoint on macOS
- Microsoft Defender for Endpoint on Mac
- macOS Antivirus policy settings for Microsoft Defender Antivirus for Intune
- Set preferences for Microsoft Defender for Endpoint on Linux
- Microsoft Defender for Endpoint on Linux
- Configure Defender for Endpoint on Android features
- Configure Microsoft Defender for Endpoint on iOS features

Last updated on 10/20/2025

Protect security settings with tamper protection

Applies to: Microsoft Defender for Endpoint Plan 1, Microsoft Defender for Endpoint Plan 2, Microsoft Defender for Business, Microsoft Defender Antivirus

What is tamper protection?

Tamper protection is a capability in Microsoft Defender for Endpoint that helps protect certain security settings, such as virus and threat protection, from being disabled or changed. During some kinds of cyber attacks, bad actors try to disable security features on devices. Disabling security features provides bad actors with easier access to your data, the ability to install malware, and the ability to exploit your data, identity, and devices. Tamper protection helps guard against these types of activities.

Tamper protection is part of anti-tampering capabilities that include standard protection attack surface reduction rules. Tamper protection is an important part of built-in protection.

Prerequisites

Supported operating systems

Tamper protection is available for devices that are running one of the following versions of Windows:

- Windows
- macOS
- Windows 10 and 11 (including Enterprise multi-session)
- Windows Server 2016 and later
- Windows Server, version 1803 or later
- Windows Server 2012 R2 (using the modern, unified solution)
- Azure Stack HCI OS, version 23H2 and later.

Tamper protection is also available for Mac, although it works a little differently than on Windows. For more information, see Protect macOS security settings with tamper protection.

What happens when tamper protection is turned on?

When tamper protection is turned on, these tamper-protected settings can't be changed:

- Virus and threat protection remains enabled.
- Real-time protection remains turned on.
- Behavior monitoring remains turned on.
- Antivirus protection, including IOfficeAntivirus (IOAV) remains enabled.
- Cloud protection remains enabled.
- Security intelligence updates occur.
- Automatic actions are taken on detected threats.
- Notifications are visible in the Windows Security app on Windows devices.
- Archived files are scanned.
- Exclusions can't be modified or added

As of signature release `1.383.1159.0`, due to confusion around the default value for "Allow Scanning Network Files", tamper protection no longer locks this setting to its default value. In managed environments, the default value is `enabled`.

> **ⓘ Important**
>
> When tamper protection is turned on, tamper-protected settings can't be changed. To avoid breaking management experiences, including Intune, Microsoft Defender for Endpoint and Configuration Manager, keep in mind that changes made to tamper-protected settings might appear to succeed but are actually blocked by tamper protection. Depending on your particular scenario, you have several options available:

- If you must make changes to a device and those changes are blocked by tamper protection, you can use troubleshooting mode to temporarily disable tamper protection on the device.

> - You can use Intune, Microsoft Defender for Endpoint or Configuration Manager to exclude devices from tamper protection.

Tamper protection doesn't prevent you from viewing your security settings. And, tamper protection doesn't affect how non-Microsoft antivirus apps register with the Windows Security app. If your organization is using Defender for Endpoint, individual users can't change the

tamper protection setting; in those cases, your security team manages tamper protection. For more information, see How do I configure or manage tamper protection?

> **Tip**
>
> Built-in protection includes turning tamper protection on by default. For more information, see:
>
> - Built-in protection helps guard against ransomware (article)
> - Tamper protection is turned on for all enterprise customers (Tech Community blog post)

Tamper protection on Windows Server 2012 R2, 2016, or Windows version 1709, 1803, or 1809

If you're using Windows Server 2012 R2 using the modern unified solution, Windows Server 2016, Windows 10 version 1709, 1803, or 1809, you don't see **Tamper Protection** in the Windows Security app. Instead, you can use PowerShell to determine whether tamper protection is enabled.

> **Important**
>
> On Windows Server 2016, the Settings app doesn't accurately reflect the status of real-time protection when tamper protection is enabled.

Use PowerShell to determine whether tamper protection and real-time protection are turned on

1. Open the Windows PowerShell app.
2. Use the Get-MpComputerStatus PowerShell cmdlet.
3. In the list of results, look for `IsTamperProtected` and `RealTimeProtectionEnabled`. (A value of *true* means tamper protection is enabled.)

How do I configure or manage tamper protection?

You can use Microsoft Intune and other methods to configure or manage tamper protection, as listed in the following table:

Expand table

Method	What you can do
Use the Microsoft Defender portal.	Turn tamper protection on (or off), tenant wide, or apply tamper protection to some users/devices. You can exclude certain devices from tamper protection. See Manage tamper protection for your organization using Microsoft Defender XDR.
Use the Microsoft Intune admin center or Configuration Manager.	Turn tamper protection on (or off), tenant wide, or apply tamper protection to some users/devices. You can exclude certain devices from tamper protection. See Manage tamper protection for your organization using Intune. Protect Microsoft Defender Antivirus exclusions from tampering if you're using Intune only or Configuration Manager only. See Tamper protection for antivirus exclusions.
Use Configuration Manager with tenant attach.	Turn tamper protection on (or off), tenant wide, or apply tamper protection to some users/devices. You can exclude certain devices from tamper protection. See Manage tamper protection for your organization using tenant attach with Configuration Manager, version 2006.
Use the Windows Security app.	Turn tamper protection on (or off) on an individual device that isn't managed by a security team (such as devices for home use). See Manage tamper protection on an individual device. *This method doesn't override tamper protection settings that are set in the Microsoft Defender portal, Intune, or Configuration Manager, and it isn't intended to be used by organizations.*

Tip

If you're using Group Policy to manage Microsoft Defender Antivirus settings, keep in mind that any changes made to tamper-protected settings are ignored. If you must make changes to a device and those changes are blocked by tamper protection, use troubleshooting mode to temporarily disable tamper protection on the device. After troubleshooting mode ends, any changes made to tamper-protected settings are reverted to their configured state. To change the values on tamper-protected settings permanently, disable tamper protection temporarily before turning it back on after the settings have changed. Keep in mind that this method can pose security risks, and doesn't work on

devices that are offline when tamper protection was temporarily disabled. For this reason, we recommend using other management methods for Defender for Endpoint settings, such as Intune, instead of using Group Policy.

Protect Microsoft Defender Antivirus exclusions

Under certain conditions, tamper protection can protect exclusions that are defined for Microsoft Defender Antivirus. For more information, see Tamper protection for exclusions.

View information about tampering attempts

Tampering attempts typically indicate that a larger cyberattack has taken place. Bad actors try to change security settings as a way to persist and stay undetected. If you're part of your organization's security team, you can view information about such attempts, and then take appropriate actions to mitigate threats.

Whenever a tampering attempt is detected, an alert is raised in the Microsoft Defender portal (https://security.microsoft.com).

Using endpoint detection and response and advanced hunting capabilities in Microsoft Defender for Endpoint, your security operations team can investigate and address such attempts.

Review your security recommendations

Tamper protection integrates with Microsoft Defender Vulnerability Management capabilities. Security recommendations include making sure tamper protection is turned on. For example, in your Vulnerability Management dashboard, you can search on *tamper*. In the results, you can select **Turn on Tamper Protection** to learn more and turn it on.

To learn more about Microsoft Defender Vulnerability Management, see Dashboard insights - Defender Vulnerability Management.

See also

- Built-in protection helps guard against ransomware
- Frequently asked questions on tamper protection
- Troubleshoot problems with tamper protection

Last updated on 10/20/2025

Protect important folders with controlled folder access

Applies to: Microsoft Defender for Endpoint Plan 1, Microsoft Defender for Endpoint Plan 2

What is controlled folder access?

Controlled folder access helps protect your valuable data from malicious apps and threats, such as ransomware. Controlled folder access protects your data by checking apps against a list of known, trusted apps. Controlled folder access can be configured by using Microsoft Defender for Endpoint Security Settings Management, Microsoft Intune, Microsoft Configuration Manager, or the Windows Security App.

Controlled folder access works best with Microsoft Defender for Endpoint, which gives you detailed reporting into controlled folder access events and blocks as part of the usual alert investigation scenarios.

> **Tip**
>
> Controlled folder access blocks don't generate alerts in the **Alerts queue**. However, you can view information about controlled folder access blocks in the **device timeline view**, while using **advanced hunting**, or with **custom detection rules**.

Prerequisites

Controlled folder access requires:

- Microsoft Defender Antivirus to be the primary antivirus (active mode).
- Real-Time Protection (RTP) needs to be on.

Supported operating systems

- Windows
- Windows 11
- Windows 10
- Azure Stack HCI OS, version 23H2 and later.

- Windows Server 2016 and later
- Windows Server 2012 R2

How does controlled folder access work?

Controlled folder access works by only allowing trusted apps to access protected folders. Protected folders are specified when controlled folder access is configured. Typically, commonly used folders, such as those used for documents, pictures, downloads, and so on, are included in the list of controlled folders.

Controlled folder access works with a list of trusted apps. Apps that are included in the list of trusted software work as expected. Apps that aren't included in the list are prevented from making any changes to files inside protected folders.

Apps are added to the list based upon their prevalence and reputation. Apps that are highly prevalent throughout your organization and that haven't ever displayed any behavior deemed malicious are considered trustworthy. Those apps are added to the list automatically.

Apps can also be added manually to the trusted list by using Configuration Manager or Intune. Other actions can be performed in the Microsoft Defender portal.

Why controlled folder access is important

Controlled folder access is especially useful in helping to protect your documents and information from ransomware. In a ransomware attack, your files can get encrypted and held hostage. With controlled folder access in place, a notification appears on the computer where an app attempted to make changes to a file in a protected folder. You can customize the notification with your company details and contact information. You can also enable the rules individually to customize what techniques the feature monitors.

The protected folders include common system folders (including boot sectors), and you can add more folders. You can also allow apps to give them access to the protected folders.

You can use audit mode to evaluate how controlled folder access would impact your organization if it were enabled.

Windows system folders are protected by default

Windows system folders are protected by default, along with several other folders:

The protected folders include common system folders (including boot sectors), and you can add other folders. You can also allow apps to give them access to the protected folders. The Windows systems folders that are protected by default are:

- `c:\Users\<username>\Documents`
- `c:\Users\Public\Documents`
- `c:\Users\<username>\Pictures`
- `c:\Users\Public\Pictures`
- `c:\Users\Public\Videos`
- `c:\Users\<username>\Videos`
- `c:\Users\<username>\Music`
- `c:\Users\Public\Music`
- `c:\Users\<username>\Favorites`

Default folders appear in the user's profile, under **This PC**, as shown in the following image:

Documents
C:\Users\UserName\Documents

Documents
C:\Users\Public\Documents

Pictures
C:\Users\UserName\Pictures

Pictures
C:\Users\Public\Pictures

Videos
C:\Users\Public\Videos

Videos
C:\Users\UserName\Videos

Music
C:\Users\UserName\Music

Music
C:\Users\Public\Music

Favorites
C:\Users\UserName\Favorites

The same profile folders are also protected for system accounts, such as `LocalService`, `NetworkService`, `systemprofile`, and so on. For example, `C:\Windows\System32\config\systemprofile\Documents` is also protected (if it exists).

> **Note**
>
> You can configure more folders as protected, but you can't remove Windows system folders that are protected by default.

> **Note**
>
> Scripting engines like PowerShell aren't trusted by controlled folder access, even if you create an "allow" indicator by using certificate and file indicators. The only way to allow script engines to modify protected folders is by adding them as an allowed app. See Allow specific apps to make changes to controlled folders.

Review controlled folder access events in the Microsoft Defender portal

> **Tip**
>
> Controlled folder access blocks don't generate alerts in the Alerts queue. However, you can view information about controlled folder access blocks in the device timeline view, while using advanced hunting, or with custom detection rules.

Defender for Endpoint provides detailed reporting into events and blocks as part of its alert investigation scenarios in the Microsoft Defender portal. For more information, see Microsoft Defender for Endpoint in Microsoft Defender XDR.

You can query Microsoft Defender for Endpoint data by using Advanced hunting. If you're using audit mode, you can use advanced hunting to see how controlled folder access settings would affect your environment if they were enabled.

Example query:

```
DeviceEvents
| where ActionType in
('ControlledFolderAccessViolationAudited','ControlledFolderAccessViolationBlocked')
```

Review controlled folder access events in Windows Event Viewer

You can review the Windows event log to see events that are created when controlled folder access blocks (or audits) an app:

1. Download the Evaluation Package and extract the file *cfa-events.xml* to an easily accessible location on the device.
2. Type **Event viewer** in the Start menu to open the Windows Event Viewer.
3. On the left panel, under **Actions**, select **Import custom view...**.
4. Navigate to where you extracted *cfa-events.xml* and select it. Alternatively, copy the XML directly.
5. Select **OK**.

   The following table shows events related to controlled folder access:

Expand table

Event ID	Description
`5007`	Event when settings are changed
`1124`	Audited controlled folder access event
`1123`	Blocked controlled folder access event
`1127`	Blocked controlled folder access sector write block event
`1128`	Audited controlled folder access sector write block event

Controlled folder access experience

A user tries to install an application that triggers Controlled folder access, if the software or application has an unknown reputation, a toast notification presents the user with the following:

```
Virus & threat protection
Unauthorized changes blocked
```

```
Controlled folder access blocked C:\...
\ApplicationName... from making changes to memory.
```

And in the Protection history, you will see:

```

Protected memory access blocked
MM/DD/YEAR HH:MM AM/PM
```

View or change the list of protected folders

You can use the Windows Security app to view the list of folders that are protected by controlled folder access.

1. On your Windows 10 or Windows 11 device, open the Windows Security app.
2. Select **Virus & threat protection**.
3. Under **Ransomware protection**, select **Manage ransomware protection**.
4. If controlled folder access is turned off, you need to turn it on. Select **protected folders**.
5. Take one of the following steps:
    - To add a folder, select **+ Add a protected folder**.
    - To remove a folder, select it, and then select **Remove**.

> **ⓘ Important**
>
> Don't add local share paths (loopbacks) as protected folders. Use the local path instead. For example, if you have shared `C:\demo` as `\\mycomputer\demo`, don't add `\\mycomputer\demo` to the list of protected folders. Instead add `C:\demo`.

Windows system folders are protected by default, and you can't remove them from the list. Subfolders are also included in protection when you add a new folder to the list.

Last updated on 10/20/2025

Protect devices from exploits

Applies to: Microsoft Defender for Endpoint Plan 1, Microsoft Defender for Endpoint Plan 2

Exploit protection automatically applies many exploit mitigation techniques to operating system processes and apps. Exploit protection is supported beginning with Windows 10, version 1709, Windows 11, and Windows Server, version 1803.

Exploit protection works best with Defender for Endpoint - which gives you detailed reporting into exploit protection events and blocks as part of the usual alert investigation scenarios.

You can enable exploit protection on an individual device, and then use Group Policy to distribute the XML file to multiple devices at once.

When a mitigation is found on the device, a notification is displayed from the Action Center. You can customize the notification with your company details and contact information. You can also enable the rules individually to customize what techniques the feature monitors.

You can also use audit mode to evaluate how exploit protection would affect your organization if it were enabled.

Many of the features in the Enhanced Mitigation Experience Toolkit (EMET) are included in exploit protection. In fact, you can convert and import existing your EMET configuration profiles into exploit protection. To learn more, see Import, export, and deploy exploit protection configurations.

> **ⓘ Important**
>
> If you're currently using EMET, it's important to note that EMET reached end of support on July 31, 2018. Consider replacing EMET with exploit protection in Windows 10.

> **⚠ Warning**
>
> Some security mitigation technologies might have compatibility issues with some applications. You should test exploit protection in all target use scenarios by using audit mode before deploying the configuration across a production environment or the rest of your network.

Review exploit protection events in the Microsoft Defender portal

Defender for Endpoint provides detailed reporting into events and blocks as part of its alert investigation scenarios.

You can query Defender for Endpoint data by using Advanced hunting. If you're using audit mode, you can use advanced hunting to see how exploit protection settings could affect your environment.

Here's an example query:

Kusto

```
DeviceEvents
| where ActionType startswith 'ExploitGuard' and ActionType !contains
'NetworkProtection'
```

Exploit Protection and advanced hunting

The advanced hunting actiontypes available for Exploit Protection are as follows:

Expand table

Exploit Protection mitigation name	Exploit Protection - Advanced Hunting - ActionTypes
Arbitrary code guard	ExploitGuardAcgAudited ExploitGuardAcgEnforced
Don't allow child processes	ExploitGuardChildProcessAudited ExploitGuardChildProcessBlocked
Export address filtering (EAF)	ExploitGuardEafViolationAudited ExploitGuardEafViolationBlocked
Import address filtering (IAF)	ExploitGuardIafViolationAudited ExploitGuardIafViolationBlocked
Block low integrity images	ExploitGuardLowIntegrityImageAudited ExploitGuardLowIntegrityImageBlocked
Code integrity guard	ExploitGuardNonMicrosoftSignedAudited ExploitGuardNonMicrosoftSignedBlocked
• Simulate execution (SimExec) • Validate API invocation (CallerCheck)	ExploitGuardRopExploitAudited ExploitGuardRopExploitBlocked

Exploit Protection mitigation name	Exploit Protection - Advanced Hunting - ActionTypes
• Validate stack integrity (StackPivot)	
Block remote images	ExploitGuardSharedBinaryAudited ExploitGuardSharedBinaryBlocked
Disable Win32k system calls	ExploitGuardWin32SystemCallAudited ExploitGuardWin32SystemCallBlocked

Review exploit protection events in Windows Event Viewer

You can review the Windows event log to see events that are created when exploit protection blocks (or audits) an app:

Expand table

Provider/source	Event ID	Description
Security-Mitigations	1	ACG audit
Security-Mitigations	2	ACG enforce
Security-Mitigations	3	Don't allow child processes audit
Security-Mitigations	4	Don't allow child processes block
Security-Mitigations	5	Block low integrity images audit
Security-Mitigations	6	Block low integrity images block
Security-Mitigations	7	Block remote images audit
Security-Mitigations	8	Block remote images block
Security-Mitigations	9	Disable win32k system calls audit
Security-Mitigations	10	Disable win32k system calls block
Security-Mitigations	11	Code integrity guard audit
Security-Mitigations	12	Code integrity guard block
Security-Mitigations	13	EAF audit
Security-Mitigations	14	EAF enforce

Provider/source	Event ID	Description
Security-Mitigations	15	EAF+ audit
Security-Mitigations	16	EAF+ enforce
Security-Mitigations	17	IAF audit
Security-Mitigations	18	IAF enforce
Security-Mitigations	19	ROP StackPivot audit
Security-Mitigations	20	ROP StackPivot enforce
Security-Mitigations	21	ROP CallerCheck audit
Security-Mitigations	22	ROP CallerCheck enforce
Security-Mitigations	23	ROP SimExec audit
Security-Mitigations	24	ROP SimExec enforce
WER-Diagnostics	5	CFG Block
Win32K	260	Untrusted Font

Mitigation comparison

The mitigations available in EMET are included natively in Windows 10 (starting with version 1709), Windows 11, and Windows Server (starting with version 1803), under Exploit protection.

The table in this section indicates the availability and support of native mitigations between EMET and exploit protection.

Expand table

Mitigation	Available under exploit protection	Available in EMET
Arbitrary code guard (ACG)	Yes	Yes As "Memory Protection Check"
Block remote images	Yes	Yes As "Load Library Check"
Block untrusted fonts	Yes	Yes

Mitigation	Available under exploit protection	Available in EMET
Data Execution Prevention (DEP)	Yes	Yes
Export address filtering (EAF)	Yes	Yes
Force randomization for images (Mandatory ASLR)	Yes	Yes
NullPage Security Mitigation	Yes Included natively in Windows 10 and Windows 11 For more information, see Mitigate threats by using Windows 10 security features	Yes
Randomize memory allocations (Bottom-Up ASLR)	Yes	Yes
Simulate execution (SimExec)	Yes	Yes
Validate API invocation (CallerCheck)	Yes	Yes
Validate exception chains (SEHOP)	Yes	Yes
Validate stack integrity (StackPivot)	Yes	Yes
Certificate trust (configurable certificate pinning)	Windows 10 and Windows 11 provide enterprise certificate pinning	Yes
Heap spray allocation	Ineffective against newer browser-based exploits; newer mitigations provide better protection For more information, see Mitigate threats by using Windows 10 security features	Yes
Block low integrity images	Yes	No
Code integrity guard	Yes	No
Disable extension points	Yes	No
Disable Win32k system calls	Yes	No
Don't allow child processes	Yes	No
Import address filtering (IAF)	Yes	No
Validate handle usage	Yes	No
Validate heap integrity	Yes	No

Mitigation	Available under exploit protection	Available in EMET
Validate image dependency integrity	Yes	No

> **Note**
>
> The Advanced ROP mitigations that are available in EMET are superseded by ACG in Windows 10 and Windows 11, which other EMET advanced settings are enabled by default, as part of enabling the anti-ROP mitigations for a process. For more information on how Windows 10 employs existing EMET technology, see the Mitigation threats by using Windows 10 security features.

See also

- Configure and audit exploit protection mitigations
- Troubleshoot exploit protection
- Optimize ASR rule deployment and detections

Last updated on 01/14/2026

Microsoft Defender SmartScreen

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Microsoft Edge

Microsoft Defender SmartScreen protects against phishing or malware websites and applications, and the downloading of potentially malicious files.

Microsoft Defender SmartScreen determines whether a site is potentially malicious by:

- Analyzing visited webpages and looking for indications of suspicious behavior. If Microsoft Defender SmartScreen determines that a page is suspicious, it shows a warning page to advise caution.
- Checking the visited sites against a dynamic list of reported phishing sites and malicious software sites. If it finds a match, Microsoft Defender SmartScreen shows a warning to let the user know that the site might be malicious.

Microsoft Defender SmartScreen determines whether a downloaded app or app installer is potentially malicious by:

- Checking downloaded files against a list of reported malicious software sites and programs known to be unsafe. If it finds a match, Microsoft Defender SmartScreen shows a warning to let the user know that the site might be malicious.
- Checking downloaded files against a list of files that are well known and downloaded frequently. If the file isn't on that list, Microsoft Defender SmartScreen shows a warning, advising caution.

Benefits of Microsoft Defender SmartScreen

Microsoft Defender SmartScreen provide an early warning system against websites that might engage in phishing attacks or attempt to distribute malware through a socially engineered attack. The primary benefits are:

- **Anti-phishing and anti-malware support:** Microsoft Defender SmartScreen helps to protect users from sites that are reported to host phishing attacks or attempt to distribute malicious software. It can also help protect against deceptive advertisements, scam sites, and drive-by attacks. Drive-by attacks are web-based attacks that tend to start on a trusted site, targeting security vulnerabilities in commonly used software. Because drive-by attacks can happen even if the user doesn't select or download anything on the page, the danger often goes unnoticed. For more information about drive-by attacks, see Evolving Microsoft Defender SmartScreen to protect you from drive-by attacks.
- **Reputation-based URL and app protection:** Microsoft Defender SmartScreen evaluates a website's URLs to determine if they're known to distribute or host unsafe content. It also

provides reputation checks for apps, checking downloaded programs and the digital signature used to sign a file. If a URL, a file, an app, or a certificate has an established reputation, users don't see any warnings. If there's no reputation, the item is marked as a higher risk and presents a warning to the user.

- **Operating system integration:** Microsoft Defender SmartScreen is integrated into the Windows 10 operating system. It checks any files an app (including 3rd-party browsers and email clients) that attempts to download and run.
- **Improved heuristics and diagnostic data:** Microsoft Defender SmartScreen is constantly learning and endeavoring to stay up to date, so it can help to protect you against potentially malicious sites and files.
- **Management through group policy and Microsoft Intune:** Microsoft Defender SmartScreen supports using both group policy and Microsoft Intune settings. For more info about all available settings, see Available Microsoft Defender SmartScreen group policy and mobile device management (MDM) settings.
- **Blocking URLs associated with potentially unwanted applications:** In Microsoft Edge (based on Chromium), SmartScreen blocks URLs associated with potentially unwanted applications, or PUAs. For more information on blocking URLs associated with PUAs, see Detect and block potentially unwanted applications.

> **Important**
>
> SmartScreen protects against malicious files from the internet. It doesn't protect against malicious files on internal locations or network shares, such as shared folders with UNC paths or SMB/CIFS shares.

Windows edition and licensing requirements

The following table lists the Windows editions that support Microsoft Defender SmartScreen:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Microsoft Defender SmartScreen license entitlements are granted by the following licenses:

Expand table

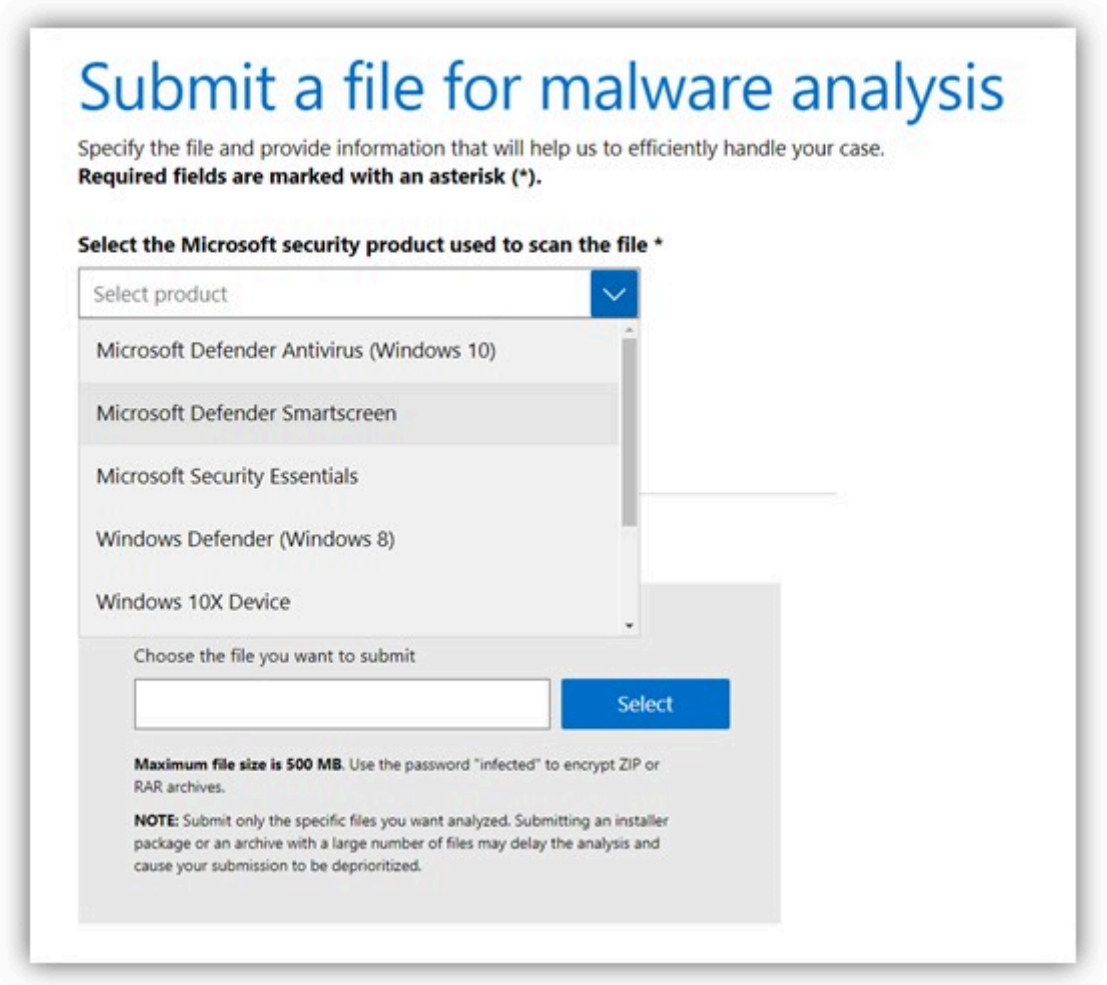
Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Submit files to Microsoft Defender SmartScreen for review

If you believe a warning or block was incorrectly shown for a file or application, or if you believe an undetected file is malware, submit a file to Microsoft for review. For more information, see Submit files for analysis.

When submitting a file for Microsoft Defender SmartScreen, make sure to select **Microsoft Defender SmartScreen** from the product menu.



Related articles

- SmartScreen frequently asked questions

- Configuration service provider reference

Last updated on 04/15/2025

Available Microsoft Defender SmartScreen settings

Applies to: ✅ Windows 11, ✅ Windows 10

Microsoft Defender SmartScreen respects Intune, Group Policy, and mobile device management (MDM) settings. You can configure Microsoft Defender SmartScreen to block suspicious content entirely, or show users a warning but allow them to continue to load the content.

See Windows settings to protect devices using Intune for the controls you can use in Intune.

Group Policy settings

SmartScreen uses registry-based Administrative Template policy settings.

Expand table

Setting	Description
Administrative Templates > Windows Components > Windows Defender SmartScreen > Explorer > Configure Windows Defender SmartScreen	This policy setting controls Microsoft Defender SmartScreen's Application Reputation ("Check apps and files") feature. If you enable this setting, it turns on Microsoft Defender SmartScreen and your users are unable to turn it off. When enabling this feature, you must pick whether users may choose to ignore warnings and run an unknown or malicious program. If you disable this setting, it turns off Microsoft Defender SmartScreen and your users are unable to turn it on. If you don't configure this setting, your users can decide whether to use Microsoft Defender SmartScreen's Application Reputation feature.
Administrative Templates > Windows Components > Windows Defender SmartScreen > Explorer > Configure App Install Control	This policy setting allows you to control whether users can install downloaded apps from outside of the Microsoft Store. This setting does not impact opening files from USB devices, local network shares, or other non-internet sources.
Administrative Templates > Microsoft Edge > SmartScreen settings > Configure Microsoft Defender SmartScreen	This policy setting lets you configure Microsoft Defender SmartScreen in the Microsoft Edge web browser. Microsoft Defender SmartScreen provides warning messages to help protect your users from potential phishing sites, tech scams, and malicious software. By default, Microsoft Defender SmartScreen is

Setting	Description
	turned on. If you enable this setting, Microsoft Defender SmartScreen is turned on, and users can't turn it off. If you disable this setting, Microsoft Defender SmartScreen is turned off, and users can't turn it on. If you don't configure this setting, users can choose whether to use Microsoft Defender SmartScreen.
Administrative Templates > Microsoft Edge > SmartScreen settings > Prevent bypassing Windows Defender SmartScreen prompts for sites	This policy setting lets you decide whether users can override Microsoft Defender SmartScreen warnings about potentially malicious websites. If you enable this setting, users can't ignore Microsoft Defender SmartScreen warnings and will be blocked from continuing to suspicious sites. If you disable or don't configure this setting, users can ignore Microsoft Defender SmartScreen warnings and continue to suspicious sites.
Administrative Templates > Microsoft Edge > SmartScreen settings > Prevent bypassing of Microsoft Defender SmartScreen warnings about downloads	This policy setting lets you decide whether users can override Microsoft Defender SmartScreen warnings about unverified (potentially malicious) downloads. If you enable this setting, users can't ignore Microsoft Defender SmartScreen warnings and will be blocked from downloading unverified files. If you disable or don't configure this setting, users can ignore Microsoft Defender SmartScreen warnings and download unverified files.

> **Note**
>
> To install the Administrative Templates ADMX file for Microsoft Edge browser policies, see Configure Microsoft Edge.

MDM settings

If you manage your policies using Microsoft Intune, use these MDM policy settings. All settings support desktop computers running Windows 10/11 Pro or Windows 10/11 Enterprise,

enrolled with Microsoft Intune.

- AllowSmartScreen
- EnableAppInstallControl
- EnableSmartScreenInShell
- PreventOverrideForFilesInShell
- PreventSmartScreenPromptOverride
- PreventSmartScreenPromptOverrideForFiles

Recommended Group Policy and MDM settings for your organization

By default, Microsoft Defender SmartScreen allows users to bypass warnings, which allows users to continue to an unsafe site or run an unsafe file, even after being warned. Because of this possibility, we strongly recommend that you set up Microsoft Defender SmartScreen to block high-risk interactions instead of providing just a warning.

To better help you protect your organization, we recommend turning on and using these specific Microsoft Defender SmartScreen Group Policy and MDM settings.

Expand table

Group Policy setting	Recommendation
Administrative Templates > Windows Components > Explorer > Configure Windows Defender SmartScreen	**Enable with the Warn and prevent bypass option.** Stops users from ignoring warning messages about malicious files downloaded from the Internet.
Administrative Templates > Microsoft Edge > SmartScreen settings > Configure Microsoft Defender SmartScreen	**Enable.** Turns on Microsoft Defender SmartScreen.
Administrative Templates > Microsoft Edge > Prevent bypassing Windows Defender SmartScreen prompts for sites	**Enable.** Stops users from ignoring warning messages and continuing to a potentially malicious website.
Administrative Templates > Microsoft Edge > Prevent bypassing of Microsoft Defender SmartScreen warnings about downloads	**Enable.** Stops users from ignoring warning messages and downloading an unverified file

Expand table

MDM setting	Recommendation
Browser/AllowSmartScreen	**1.** Turns on Microsoft Defender SmartScreen.
Browser/PreventSmartScreenPromptOverride	**1.** Stops users from ignoring warning messages and continuing to a potentially malicious website.
Browser/PreventSmartScreenPromptOverrideForFiles	**1.** Stops users from ignoring warning messages and continuing to download potentially malicious files.
SmartScreen/EnableSmartScreenInShell	**1.** Turns on Microsoft Defender SmartScreen in Windows. Requires at least Windows 10, version 1703.
SmartScreen/PreventOverrideForFilesInShell	**1.** Stops users from ignoring warning messages about malicious files downloaded from the Internet. Requires at least Windows 10, version 1703.

> **Note**
>
> For a list of settings available for Enhanced phishing protection, see Enhanced phishing protection.

Last updated on 04/28/2025

Enhanced Phishing Protection in Microsoft Defender SmartScreen

Applies to: ✅ Windows 11, version 22H2

Starting in Windows 11, version 22H2, Enhanced Phishing Protection in Microsoft Defender SmartScreen helps protect Microsoft school or work passwords against phishing and unsafe usage on sites and apps.

If a user signs into Windows using a password, Enhanced Phishing Protection works alongside Windows security protections, and helps protect typed work or school password used to sign into Windows 11 in these ways:

- If users type or paste their work or school password on any browser, into a site deemed malicious by Microsoft Defender SmartScreen, Enhanced Phishing Protection alerts them. It also alerts them to change their password so attackers can't gain access to their account.
- Reusing work or school passwords makes it easy for attackers who compromise a user's password to gain access to their other accounts. Enhanced Phishing Protection can warn users if they reuse their work or school Microsoft account password on sites and apps and alert them to change their password.
- Since it's unsafe to store plaintext passwords in text editors, Enhanced Phishing Protection can warn users if they store their work or school password in Notepad, Word, or any Microsoft 365 Office app, and recommends they delete their password from the file.
- If users type their work or school password into a website or app that SmartScreen finds suspicious, Enhanced Phishing Protection can automatically collect information from that website or app to help identify security threats. For example, the content displayed, sounds played, and application memory.

> ⓘ **Note**
>
> When a user signs in to a device using a Windows Hello for Business PIN or biometric, Enhanced Phishing Protection doesn't alert the user or send events to Microsoft Defender for Endpoint (MDE).

Benefits of Enhanced Phishing Protection in Microsoft Defender SmartScreen

Enhanced Phishing Protection provides robust phishing protections for work or school passwords that are used to sign into Windows 11. The benefits of Enhanced Phishing Protection are:

- **Anti-phishing support:** Phishing attacks trick users through convincing imitations of safe content or through credential harvesting content hosted inside trusted sites and applications. Enhanced Phishing Protection helps protect users from reported phishing sites by evaluating the URLs a site or app is connecting to, along with other characteristics, to determine if they're known to distribute or host unsafe content.

- **Secure operating system integration:** Enhanced Phishing Protection is integrated directly into the Windows 11 operating system, so it can understand users' password entry context (including process connections, URLs, certificate information) in any browser or app. Because Enhanced Phishing Protection has unparalleled insight into what is happening at the OS level, it can identify when users type their work or school password unsafely. If users do use their work or school password unsafely, the feature empowers users to change their password to minimize chances of their compromised credential being weaponized against them.

- **Unparalleled telemetry shared throughout Microsoft's security suite:** Enhanced Phishing Protection is constantly learning from phishing attacks seen throughout the entire Microsoft security stack. It works alongside other Microsoft security products, to provide a layered approach to password security, especially for organizations early in their password-less authentication journey. If your organization uses Microsoft Defender for Endpoint, you can see valuable phishing sensors data in the Microsoft 365 Defender Portal. This portal lets you view Enhanced Phishing Protection alerts and reports for unsafe password usage in your environment.

- **Easy management through Group Policy and Microsoft Intune:** Enhanced Phishing Protection works with Group Policy and mobile device management (MDM) settings to help you manage your organization's computer settings. Based on how you set up Enhanced Phishing Protection, you can customize which phishing protection scenarios show users warning dialogs. For example, the Service Enabled setting determines whether the Enhanced Phishing Protection service is on or off. The feature is in audit mode if the other settings, which correspond to notification policies, aren't enabled.

Windows edition and licensing requirements

The following table lists the Windows editions that support Enhanced phishing protection with SmartScreen:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Enhanced phishing protection with SmartScreen license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Configure Enhanced Phishing Protection for your organization

Enhanced Phishing Protection can be configured via Microsoft Intune, Group Policy Objects (GPO), or Configuration Service Providers (CSP) with an MDM service. These settings are available to configure your devices using either Microsoft Intune, GPO, or CSP.

Expand table

Setting	Description
Automatic Data Collection	This policy setting determines whether Enhanced Phishing Protection can collect additional information-such as content displayed, sounds played, and application memory-when your users enter their work or school password into a suspicious website or app. This information is used only for security purposes and helps SmartScreen determine whether the website or app is malicious. • If you enable this policy setting, Enhanced Phishing Protection may automatically collect additional content for security analysis from a suspicious website or app when your users enter their work or school password into that website or app. • If you disable this policy setting, Enhanced Phishing Protection won't collect additional content for security analysis when your users enter their work or school password into a suspicious site or app. • If this policy isn't set, Enhanced Phishing Protection automatic data collection honors the end user's settings.
Service Enabled	This policy setting determines whether Enhanced Phishing Protection is in audit mode or off. Users don't see any notifications for any protection scenarios when Enhanced

Setting	Description
	Phishing Protection is in audit mode. In audit mode, Enhanced Phishing Protection captures unsafe password entry events and sends diagnostic data through Microsoft Defender. • If you enable or don't configure this setting, Enhanced Phishing Protection is enabled in audit mode, preventing users to turn it off. • If you disable this policy setting, Enhanced Phishing Protection is off. When off, Enhanced Phishing Protection doesn't capture events, send data, or notify users. Additionally, your users are unable to turn it on.
Notify Malicious	This policy setting determines whether Enhanced Phishing Protection warns your users if they type their work or school password into one of the following malicious scenarios: into a reported phishing site, into a sign-in URL with an invalid certificate, or into an application connecting to either a reported phishing site or a sign-in URL with an invalid certificate • If you enable this policy setting, Enhanced Phishing Protection warns your users if they type their work or school password into one of the malicious scenarios described above and encourages them to change their password. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they type their work or school password into one of the malicious scenarios described above.
Notify Password Reuse	This policy setting determines whether Enhanced Phishing Protection warns your users if they reuse their work or school password. • If you enable this policy setting, Enhanced Phishing Protection warns users if they reuse their work, or school password and encourages them to change it. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they reuse their work or school password.
Notify Unsafe App	This policy setting determines whether Enhanced Phishing Protection warns your users if they type their work or school passwords in Notepad or Microsoft 365 Office Apps. • If you enable this policy setting, Enhanced Phishing Protection warns your users if they store their password in Notepad or Microsoft 365 Office Apps. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they store their password in Notepad or Microsoft 365 Office Apps.

Enhanced Phishing Protection allows organizations to add their custom identity provider sign-in URL as a recognized URL. Then Enhanced Phishing Protection doesn't consider Microsoft passwords typed into an internal identity provider (IdP) as unknown or password reuse. Without knowledge of an enterprise's custom identity provider URL, SmartScreen might not have enough information about the URL. If you configure warning dialogs for Enhanced Phishing Protection, it might show an unsafe password usage dialog to the user entering their Microsoft password into the URL.

To add your organization's custom sign-in URL to Enhanced Phishing Protection, configure the `EnableWebSignIn` policy in the Authentication Policy CSP. For more information, see Web sign-in for Windows.

Follow these instructions to configure your devices using either Microsoft Intune, GPO, or CSP.

Intune

To configure devices using Microsoft Intune, create a **Settings catalog** policy, and use the settings listed under the category `SmartScreen > Enhanced Phishing Protection`:

- Automatic Data Collection
- Service Enabled
- Notify Malicious
- Notify Password Reuse
- Notify Unsafe App

Assign the policy to a security group that contains as members the devices or users that you want to configure.

Recommended settings for your organization

By default, Enhanced Phishing Protection is deployed in audit mode, preventing notifications to the users for any protection scenarios. In audit mode, Enhanced Phishing Protection captures unsafe password entry events and sends diagnostic data through Microsoft Defender. Users aren't warned if they enter their work or school password into a phishing site, if they reuse their password, or if they unsafely store their password in applications. Because of this possibility, we recommend that you configure Enhanced Phishing Protection to warn users during all protection scenarios.

Expand table

Setting	Default Value	Recommendation
Automatic Data Collection	**Enabled** for domain joined devices or devices enrolled with MDM. **Disabled** for all other devices.	**Enabled**: Turns on collection of additional content for security analysis from a suspicious website or app to improve Microsoft's threat intelligence. This information is used only for security purposes and helps SmartScreen determine whether the website or app is malicious.
Service Enabled	**Enabled**	**Enabled**: Turns on Enhanced Phishing Protection in audit mode, which captures work or school password entry events and sends diagnostic data but doesn't show any notifications to your users.
Notify Malicious	**Disabled** for devices onboarded to MDE.	**Enabled**: Turns on Enhanced Phishing Protection notifications when users type their work or school password into one of the

Setting	Default Value	Recommendation
	Enabled for all other devices.	previously described malicious scenarios and encourages them to change their password.
Notify Password Reuse	**Disabled**	**Enabled**: Turns on Enhanced Phishing Protection notifications when users reuse their work or school password and encourages them to change their password.
Notify Unsafe App	**Disabled**	**Enabled**: Turns on Enhanced Phishing Protection notifications when users type their work or school passwords in Notepad and Microsoft 365 Office Apps.

To better help you protect your organization, we recommend turning on and using these specific Microsoft Defender SmartScreen settings.

Intune

Expand table

Settings catalog element	Recommended value
Automatic Data Collection	**Enabled**
Service Enabled	**Enabled**
Notify Malicious	**Enabled**
Notify Password Reuse	**Enabled**
Notify Unsafe App	**Enabled**

Related articles

- SmartScreen frequently asked questions
- WebThreatDefense CSP
- Threat protection

Last updated on 04/15/2025

Microsoft Defender for Endpoint documentation

Microsoft Defender for Endpoint delivers preventative protection, post-breach detection, automated investigation, and response.

Microsoft Defender for Endpoint

OVERVIEW

What is Microsoft Defender for Endpoint?

What is Defender for Endpoint plan 1?

Defender for Endpoint on Linux

Defender for Endpoint on Mac overview

Defender for Endpoint on Android

Defender for Endpoint on iOS

WHAT'S NEW

What's new in Microsoft Defender for Endpoint

Announcing Microsoft Defender for Endpoint Plan 1

VIDEO

Overview video

Evaluate & deploy the service

GET STARTED

Evaluate Microsoft Defender for Endpoint

Plan your deployment

DEPLOY

Deployment guide

Onboard supported devices

Set up and configure Defender for Endpoint Plan 1

HOW-TO GUIDE

Migration guide

VIDEO

Onboarding video

Security operations

OVERVIEW

Endpoint detection and response

Behavioral blocking and containment

Automated investigation and response (AIR)

Advanced hunting

Microsoft Threat Experts

Threat analytics

Reference

REFERENCE

Management and APIs

Partner integration

Security administration

OVERVIEW

Microsoft Defender Vulnerability Management

Attack surface reduction

Next-generation protection

Security baselines

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Using security baselines in your organization

Microsoft is dedicated to providing its customers with secure operating systems, such as Windows and Windows Server, and secure apps, such as Microsoft 365 apps for enterprise and Microsoft Edge. In addition to the security assurance of its products, Microsoft also enables you to have fine control over your environments by providing various configuration capabilities.

Even though Windows and Windows Server are designed to be secure out-of-the-box, many organizations still want more granular control over their security configurations. To navigate the large number of controls, organizations need guidance on configuring various security features. Microsoft provides this guidance in the form of security baselines.

We recommend that you implement an industry-standard configuration that is broadly known and well-tested, such as Microsoft security baselines, as opposed to creating a baseline yourself. This industry-standard configuration helps increase flexibility and reduce costs.

For more information, see the following blog post: Sticking with well-known and proven solutions.

What are security baselines?

Every organization faces security threats. However, the types of security threats that are of most concern to one organization can be different from another organization. For example, an e-commerce company might focus on protecting its internet-facing web apps, while a hospital might focus on protecting confidential patient information. The one thing that all organizations have in common is a need to keep their apps and devices secure. These devices must be compliant with the security standards (or security baselines) defined by the organization.

A security baseline is a group of Microsoft-recommended configuration settings that explains their security implication. These settings are based on feedback from Microsoft security engineering teams, product groups, partners, and customers.

Why are security baselines needed?

Security baselines are an essential benefit to customers because they bring together expert knowledge from Microsoft, partners, and customers.

For example, there are over 3,000 group policy settings for Windows 10, which doesn't include over 1,800 Internet Explorer 11 settings. Of these 4,800 settings, only some are security-related. Although Microsoft provides extensive guidance on different security features, exploring each one can take a long time. You would have to determine the security implication of each setting on your own. Then, you would still need to determine the appropriate value for each setting.

In modern organizations, the security threat landscape is constantly evolving, and IT pros and policy-makers must keep up with security threats and make required changes to security settings to help mitigate these threats. To enable faster deployments and make managing Microsoft products easier, Microsoft provides customers with security baselines that are available in consumable formats, such as group policy object backups.

Windows edition and licensing requirements

The following table lists the Windows editions that support Security baselines:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Security baselines license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Baseline principles

Our recommendations follow a streamlined and efficient approach to baseline definitions. The foundation of that approach is essentially:

- The baselines are designed for well-managed, security-conscious organizations in which standard end users don't have administrative rights.

- A baseline enforces a setting only if it mitigates a contemporary security threat and doesn't cause operational issues that are worse than the risks they mitigate.
- A baseline enforces a default only if it's otherwise likely to be set to an insecure state by an authorized user:
  - If a non-administrator can set an insecure state, enforce the default.
  - If setting an insecure state requires administrative rights, enforce the default only if it's likely that a misinformed administrator will otherwise choose poorly.

How can you use security baselines?

You can use security baselines to:

- Ensure that user and device configuration settings are compliant with the baseline.
- Set configuration settings. For example, you can use group policy, Microsoft Configuration Manager, or Microsoft Intune to configure a device with the setting values specified in the baseline.

Where can I get the security baselines?

There are several ways to get and use security baselines:

1. You can download the security baselines from the Microsoft Download Center. This download page is for the Security Compliance Toolkit (SCT), which comprises tools that can assist admins in managing baselines in addition to the security baselines. The SCT also includes tools to help you manage the security baselines. You can also get support for the security baselines

2. Mobile device management (MDM) security baselines function like the Microsoft group policy-based security baselines and can easily integrate these baselines into an existing MDM management tool.

3. MDM security baselines can easily be configured in Microsoft Intune on devices that run Windows 10 and Windows 11. For more information, see List of the settings in the Windows 10/11 MDM security baseline in Intune.

Related articles

- Microsoft Security Baselines Blog
- Microsoft Security Compliance Toolkit
- Security Baseline Policy Analyzer

Last updated on 08/18/2025

Microsoft Security Compliance Toolkit - How to use

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

What is the Security Compliance Toolkit (SCT)?

The Security Compliance Toolkit (SCT) is a set of tools that allows enterprise security administrators to download, analyze, test, edit, and store Microsoft-recommended security configuration baselines for Windows and other Microsoft products.

The SCT enables administrators to effectively manage their enterprise's Group Policy Objects (GPOs). Using the toolkit, administrators can compare their current GPOs with Microsoft-recommended GPO baselines or other baselines, edit them, store them in GPO backup file format, and apply them broadly through Active Directory or individually through local policy.

The Security Compliance Toolkit consists of:

- Windows 11 security baseline
  - Windows 11, version 24H2
  - Windows 11, version 23H2
  - Windows 11, version 22H2
  - Windows 11, version 21H2
- Windows 10 security baselines
  - Windows 10, version 22H2
  - Windows 10, version 21H2
  - Windows 10, version 1809
  - Windows 10, version 1607
  - Windows 10, version 1507
- Windows Server security baselines
  - Windows Server 2025
  - Windows Server 2022
  - Windows Server 2019
  - Windows Server 2016
- Microsoft Office security baseline
  - Microsoft 365 Apps for Enterprise Version 2412
- Microsoft Edge security baseline
  - Microsoft Edge version 128
- Tools
  - Policy Analyzer

- Local Group Policy Object (LGPO)
- Set Object Security
- GPO to Policy Rules

You can download the tools along with the baselines for the relevant Windows versions. For more information about security baseline recommendations, see the Microsoft Security Guidance blog.

What is the Policy Analyzer tool?

The Policy Analyzer is a utility for analyzing and comparing sets of Group Policy Objects (GPOs). Its main features include:

- Highlight when a set of Group Policies has redundant settings or internal inconsistencies
- Highlight the differences between versions or sets of Group Policies
- Compare GPOs against current local policy and local registry settings
- Export results to a Microsoft Excel spreadsheet

Policy Analyzer lets you treat a set of GPOs as a single unit. This treatment makes it easy to determine whether particular settings are duplicated across the GPOs or are set to conflicting values. Policy Analyzer also lets you capture a baseline and then compare it to a snapshot taken at a later time to identify changes anywhere across the set.

More information on the Policy Analyzer tool can be found on the Microsoft Security Guidance blog or by downloading the tool.

What is the Local Group Policy Object (LGPO) tool?

`LGPO.exe` is a command-line utility that is designed to help automate management of Local Group Policy. Using local policy gives administrators a simple way to verify the effects of Group Policy settings, and is also useful for managing non-domain-joined systems. `LGPO.exe` can import and apply settings from Registry Policy (Registry.pol) files, security templates, Advanced Auditing backup files, and from formatted "LGPO text" files. It can export local policy to a GPO backup. It can export the contents of a Registry Policy file to the "LGPO text" format that can then be edited, and can build a Registry Policy file from an LGPO text file.

Documentation for the LGPO tool can be found on the Microsoft Security Guidance blog or by downloading the tool.

What is the Set Object Security tool?

`SetObjectSecurity.exe` enables you to set the security descriptor for just about any type of Windows securable object, such as files, directories, registry keys, event logs, services, and SMB shares. For file system and registry objects, you can choose whether to apply inheritance rules. You can also choose to output the security descriptor in a `.reg` file compatible representation of the security descriptor for a REG_BINARY registry value.

Documentation for the Set Object Security tool can be found on the Microsoft Security Baselines blog or by downloading the tool.

What is the GPO to Policy Rules tool?

Automate the conversion of GPO backups to Policy Analyzer `.PolicyRules` files and skip the GUI. GPO2PolicyRules is a command-line tool that is included with the Policy Analyzer download.

Documentation for the GPO to PolicyRules tool can be found on the Microsoft Security Baselines blog or by downloading the tool.

Last updated on 08/18/2025

Get Support

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Frequently asked questions

What is the Microsoft Security Compliance Manager (SCM)?

The Security Compliance Manager (SCM) is now retired and is no longer supported. The reason is that SCM was an incredibly complex and large program that needed to be updated for every Windows release. It has been replaced by the Security Compliance Toolkit (SCT). To provide a better service for our customers, we've moved to SCT with which we can publish baselines through the Microsoft Download Center in a lightweight .zip file that contains GPO Backups, GPO reports, Excel spreadsheets, WMI filters, and scripts to apply the settings to local policy.

More information about this change can be found on the Microsoft Security Guidance blog.

What file formats are supported by the SCT?

The toolkit supports formats created by the Windows GPO backup feature (`.pol`, `.inf`, and `.csv`). Policy Analyzer saves its data in XML files with a `.PolicyRules` file extension. LGPO also supports its own LGPO text file format as a text-based analog for the binary registry.pol file format. For more information, see the LGPO documentation. Keep in mind that SCMs' `.cab` files are no longer supported.

Does SCT support Desired State Configuration (DSC) file format?

No. PowerShell-based DSC is rapidly gaining popularity, and more DSC tools are coming online to convert GPOs and DSC and to validate system configuration.

Does SCT support the creation of Microsoft Configuration Manager DCM packs?

No. A potential alternative is Desired State Configuration (DSC), a feature of the Windows Management Framework. For a tool that supports conversion of GPO Backups to DSC format, see BaselineManagement.

Does SCT support the creation of Security Content Automation Protocol (SCAP)-format policies?

No. SCM supported only SCAP 1.0, which wasn't updated as SCAP evolved. The new toolkit likewise doesn't include SCAP support.

Version matrix

Client versions

Expand table

Name	Build	Baseline release date	Security tools
Windows 11	24H2	October 2024	SCT 1.0
Windows 11	23H2	October 2023	SCT 1.0
Windows 11	22H2	September 2022	SCT 1.0
Windows 10	22H2 21H2 20H2 1809 1607 1507	October 2022 December 2021 December 2020 October 2018 October 2016 January 2016	SCT 1.0

Server versions

Expand table

Name	Build	Baseline Release Date	Security Tools
Windows Server 2025	SecGuide	January 2025	SCT 1.0
Windows Server 2022	SecGuide	September 2021	SCT 1.0
Windows Server 2019	SecGuide	November 2018	SCT 1.0
Windows Server 2016	SecGuide	October 2016	SCT 1.0

Microsoft products

Expand table

Name	Details	Security Tools
Microsoft 365 Apps for enterprise, version 2412	SecGuide	SCT 1.0
Microsoft Edge, version 128	SecGuide	SCT 1.0

Related articles

Windows security baselines

Last updated on 08/18/2025

What is Microsoft Baseline Security Analyzer and its uses?

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Microsoft Baseline Security Analyzer (MBSA) is used to verify patch compliance. MBSA also performed several other security checks for Windows, IIS, and SQL Server. Unfortunately, the logic behind these extra checks hadn't been actively maintained since Windows XP and Windows Server 2003. Changes in the products since then rendered many of these security checks obsolete and some of their recommendations counterproductive.

MBSA was largely used in situations where Microsoft Update a local WSUS or Configuration Manager server wasn't available, or as a compliance tool to ensure that all security updates were deployed to a managed environment. While MBSA version 2.3 introduced support for Windows Server 2012 R2 and Windows 8.1, it has since been deprecated and no longer developed. MBSA 2.3 isn't updated to fully support Windows 10 and Windows Server 2016.

> ⓘ **Note**
>
> In accordance with our SHA-1 deprecation initiative, the Wsusscn2.cab file is no longer dual-signed using both SHA-1 and the SHA-2 suite of hash algorithms (specifically SHA-256). This file is now signed using only SHA-256. Administrators who verify digital signatures on this file should now expect only single SHA-256 signatures. Starting with the August 2020 Wsusscn2.cab file, MBSA will return the following error "The catalog file is damaged or an invalid catalog." when attempting to scan using the offline scan file.

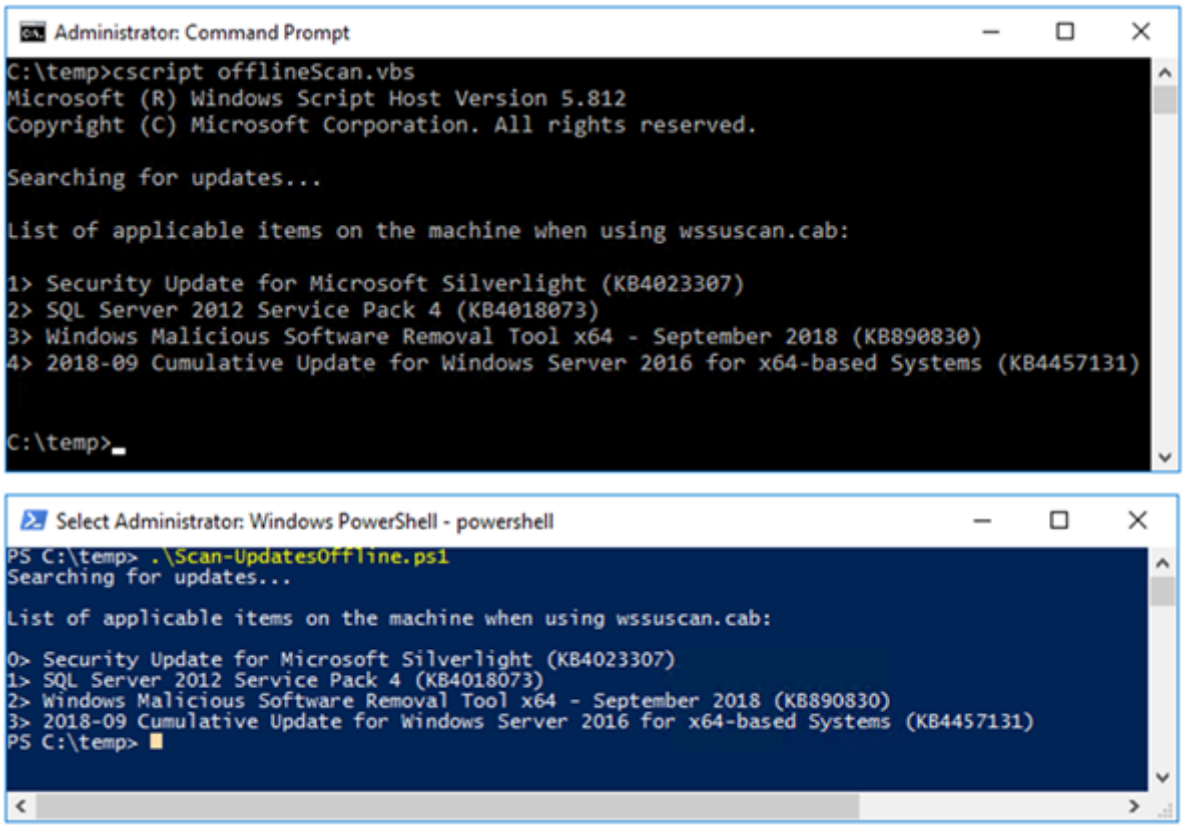
Solution

A script can help you with an alternative to MBSA's patch-compliance checking:

- Using WUA to Scan for Updates Offline, which includes a sample .vbs script.

For a PowerShell alternative, see Using WUA to Scan for Updates Offline with PowerShell.

For example:



The preceding scripts use the WSUS offline scan file (wsusscn2.cab) to perform a scan and get the same information on missing updates as MBSA supplied. MBSA also relied on the wsusscn2.cab to determine which updates were missing from a given system without connecting to any online service or server. The wsusscn2.cab file is still available and there are currently no plans to remove or replace it.

The wsusscn2.cab file contains the metadata of only security updates, update rollups, and service packs available from Microsoft Update; it doesn't contain any information on non-security updates, tools, or drivers.

More information

For security compliance and for desktop/server hardening, we recommend the Microsoft Security Baselines and the Security Compliance Toolkit.

- Windows security baselines
- Download Microsoft Security Compliance Toolkit 1.0
- Microsoft Security Guidance blog

Last updated on 08/18/2025

Override Process Mitigation Options to help enforce app-related security policies

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows includes group policy-configurable "Process Mitigation Options" that add advanced protections against memory-based attacks, that is, attacks where malware manipulates memory to gain control of a system. For example, malware might attempt to use buffer overruns to inject malicious executable code into memory, but Process Mitigation Options can prevent the running of the malicious code.

> **ⓘ Important**
>
> We recommend trying these mitigations in a test lab before deploying to your organization, to determine if they interfere with your organization's required apps.

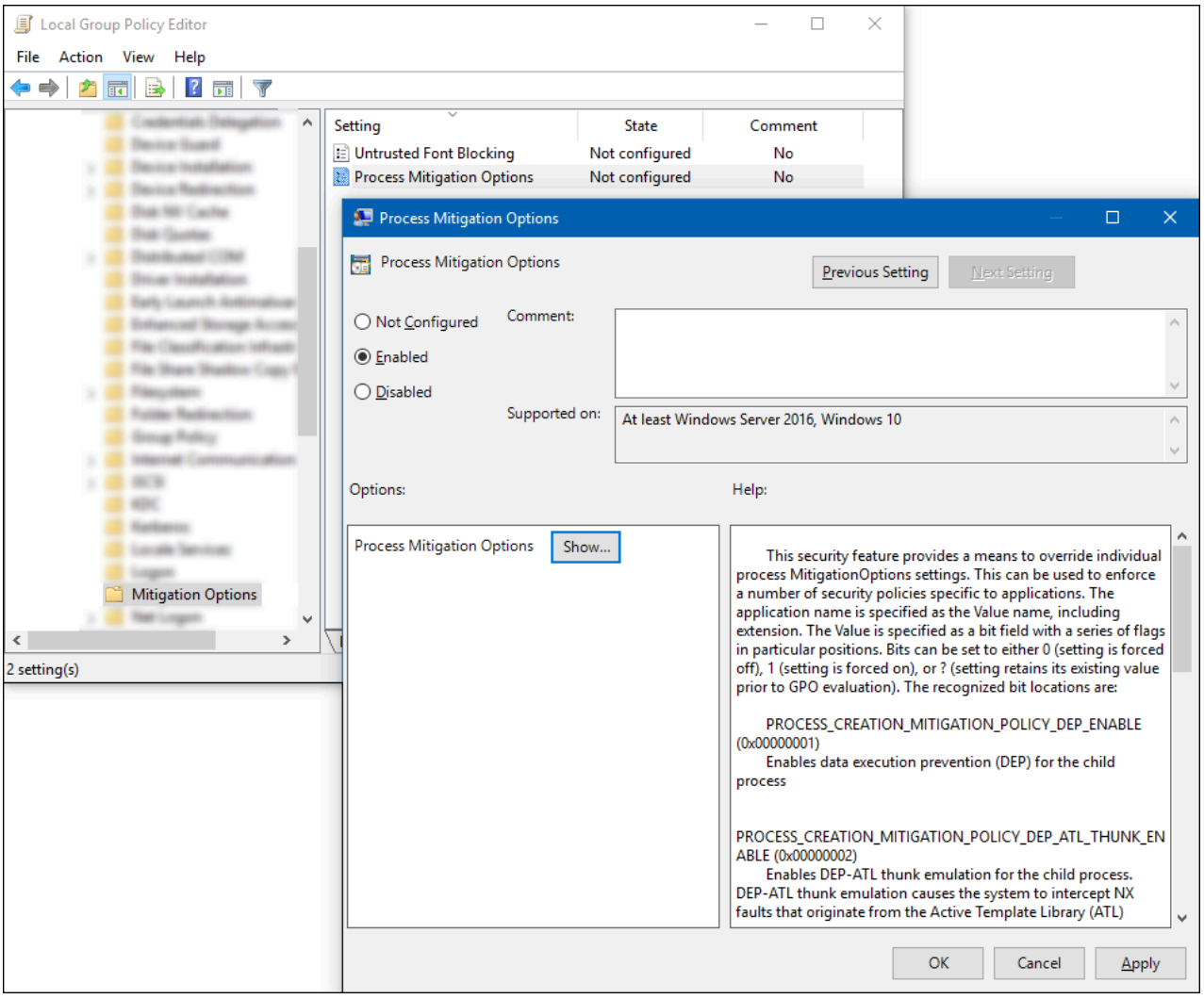
The Group Policy settings in this article are related to three types of process mitigations. All three types are on by default for 64-bit applications, but by using the Group Policy settings described in this article, you can configure more protections. The types of process mitigations are:

- **Data Execution Prevention (DEP)** is a system-level memory protection feature that enables the operating system to mark one or more pages of memory as nonexecutable, preventing code from being run from that region of memory, to help prevent exploitation of buffer overruns. DEP helps prevent code from being run from data pages such as the default heap, stacks, and memory pools. For more information, see Data Execution Prevention.
- **Structured Exception Handling Overwrite Protection (SEHOP)** is designed to block exploits that use the Structured Exception Handler (SEH) overwrite technique. Because this protection mechanism is provided at run-time, it helps to protect apps regardless of whether they're compiled with the latest improvements. For more information, see Structured Exception Handling Overwrite Protection.
- **Address Space Layout Randomization (ASLR)** loads DLLs into random memory addresses at boot time to mitigate against malware that's designed to attack specific memory locations, where specific DLLs are expected to be loaded. For more information, see Address Space Layout Randomization. To find more ASLR protections in the table below, look for `IMAGES` or `ASLR`.

The following procedure describes how to use Group Policy to override individual **Process Mitigation Options** settings.

To modify Process Mitigation Options

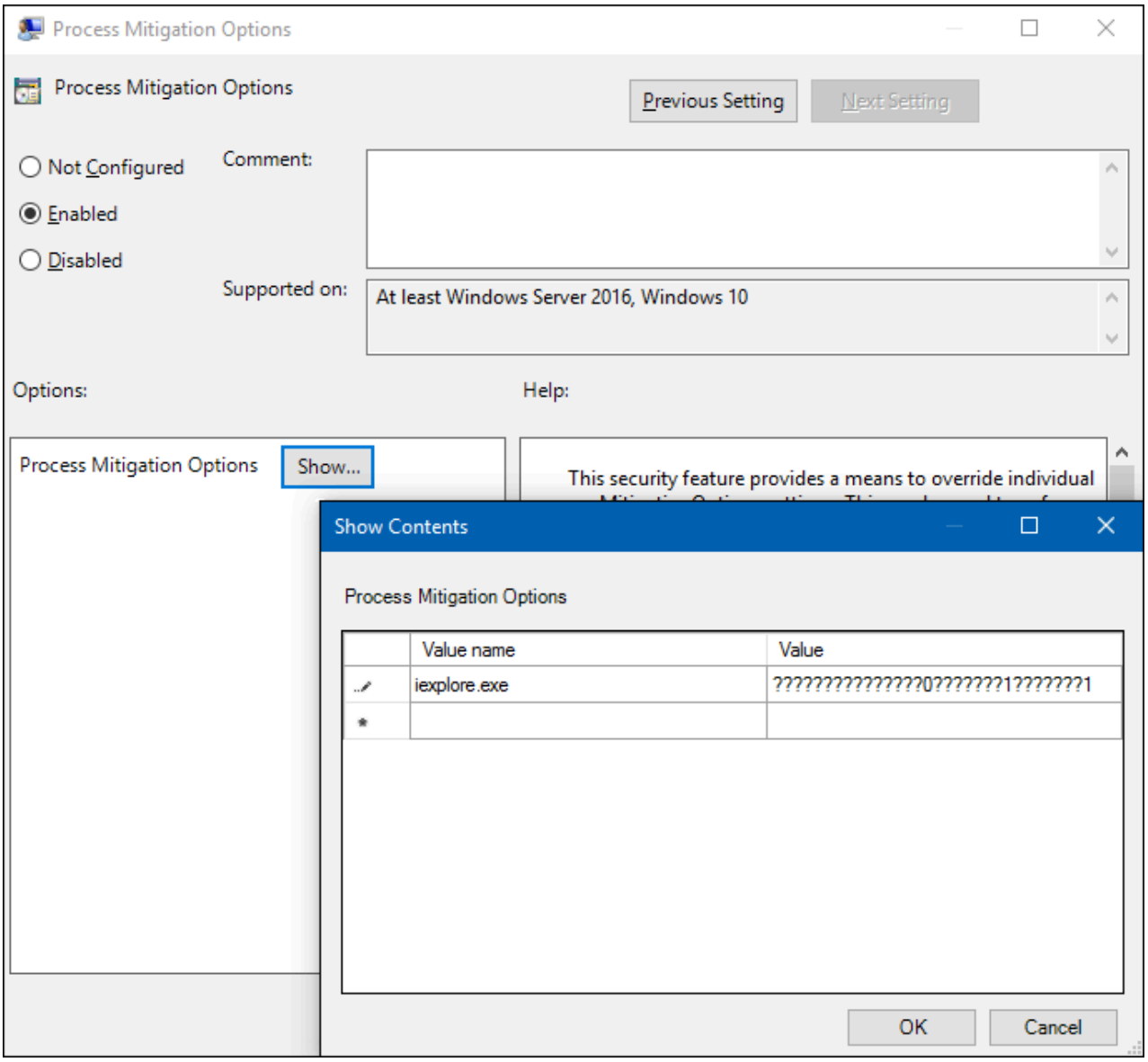
1. Open your Group Policy editor and go to the **Administrative Templates\System\Mitigation Options\Process Mitigation Options** setting.



2. Select **Enabled**, and then in the **Options** area, select **Show** to open the **Show Contents** box, where you can add your apps and the appropriate bit flag values, as shown in the Setting the bit field and Example sections of this article.

> **ⓘ Important**
>
> For each app you want to include, you must include:
>
> - **Value name.** The app file name, including the extension. For example, `iexplore.exe`.
> - **Value.** A bit field with a series of bit flags in particular positions. Bits can be set to `0` (where the setting is forced off), `1` (where the setting is forced on), or `?` (where the setting retains the previous, existing value). Setting bit flags in

positions not specified here to anything other than `?` might cause undefined behavior.



Setting the bit field

Here's a visual representation of the bit flag locations for the various Process Mitigation Options settings:

	3										2										1									0
1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0	9	8	7	6	5	4	3	2	1	0
?	?	?	?	?	?	?	?	?	?	?	?	?	?	?	F	E	?	?	?	?	?	?	D	?	?	?	?	?	C	B	A

Where the bit flags are read from right to left and are defined as:

Expand table

Flag	Bit location	Setting	Details
A	0	`PROCESS_CREATION_MITIGATION_POLICY_DEP_ENABLE (0x00000001)`	Turns on Data Execution Prevention (DEP) for child processes.
B	1	`PROCESS_CREATION_MITIGATION_POLICY_DEP_ATL_THUNK_ENABLE (0x00000002)`	Turns on DEP-ATL thunk emulation for child processes. DEP-ATL thunk emulation lets the system intercept nonexecutable (NX) faults that originate from the Active Template Library (ATL) thunk layer, and then emulate and handle the instructions so the process can continue to run.
C	2	`PROCESS_CREATION_MITIGATION_POLICY_SEHOP_ENABLE (0x00000004)`	Turns on Structured Exception Handler Overwrite Protection (SEHOP) for child processes. SEHOP helps to block exploits that use the Structured Exception Handler (SEH)

Flag	Bit location	Setting	Details
			overwrite technique.
D	8	`PROCESS_CREATION_MITIGATION_POLICY_FORCE_RELOCATE_IMAGES_ALWAYS_ON (0x00000100)`	Uses the force Address Space Layout Randomization (ASLR) setting to act as though an image base collision happened at load time, forcibly rebasing images that aren't dynamic base compatible. Images without the base relocation section aren't loaded if relocations are required.
E	15	`PROCESS_CREATION_MITIGATION_POLICY_BOTTOM_UP_ASLR_ALWAYS_ON (0x00010000)`	Turns on the bottom-up randomization policy, which includes stack randomization options and causes a random location to be used as the lowest user address.
F	16	`PROCESS_CREATION_MITIGATION_POLICY_BOTTOM_UP_ASLR_ALWAYS_OFF (0x00020000)`	Turns off the bottom-up randomization policy, which

Flag	Bit location	Setting	Details
			includes stack randomization options and causes a random location to be used as the lowest user address.

Example

If you want to turn on the **PROCESS_CREATION_MITIGATION_POLICY_DEP_ENABLE** and **PROCESS_CREATION_MITIGATION_POLICY_FORCE_RELOCATE_IMAGES_ALWAYS_ON** settings, turn off the **PROCESS_CREATION_MITIGATION_POLICY_BOTTOM_UP_ASLR_ALWAYS_OFF** setting, and leave everything else as the default values, you'd want to type a value of `????????????????0???????1???????1`.

Last updated on 08/18/2025

Use Windows Event Forwarding to help with intrusion detection

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Learn about an approach to collect events from devices in your organization. This article talks about events in both normal operations and when an intrusion is suspected.

Windows Event Forwarding (WEF) reads any operational or administrative event logged on a device in your organization and forwards the events you choose to a Windows Event Collector (WEC) server.

To accomplish this functionality, there are two different subscriptions published to client devices - the Baseline subscription and the suspect subscription. The Baseline subscription enrolls all devices in your organization, and a Suspect subscription only includes devices that have been added by you. The Suspect subscription collects more events to help build context for system activity and can quickly be updated to accommodate new events and/or scenarios as needed without impacting baseline operations.

This implementation helps differentiate where events are ultimately stored. Baseline events can be sent to devices with online analytical capability, such as Security Event Manager (SEM), while also sending events to a MapReduce system, such as HDInsight or Hadoop, for long-term storage and deeper analysis. Events from the Suspect subscription are sent directly to a MapReduce system due to volume and lower signal/noise ratio, they're largely used for host forensic analysis.

An SEM's strength lies in being able to inspect, correlate events, and generate alerts for known patterns manner and alert security staff at machine speed.

A MapReduce system has a longer retention time (years versus months for an SEM), larger ingress ability (hundreds of terabytes per day), and the ability to perform more complex operations on the data like statistical and trend analysis, pattern clustering analysis, or apply Machine Learning algorithms.

Here's an approximate scaling guide for WEF events:

Expand table

Events/second range	Data store
0 - 5,000	SQL or SEM

Events/second range	Data store
5,000 - 50,000	SEM
50,000+	Hadoop/HDInsight/Data Lake

Event generation on a device must be enabled either separately or as part of the GPO for the baseline WEF implementation, including enabling of disabled event logs and setting channel permissions. For more info, see Appendix C - Event channel settings (enable and channel access) methods. This condition is because WEF is a passive system regarding the event log. It can't change the size of event log files, enable disabled event channels, change channel permissions, or adjust a security audit policy. WEF only queries event channels for existing events. Additionally, having event generation already occurring on a device allows for more complete event collection building a complete history of system activity. Otherwise, you'll be limited to the speed of GPO and WEF subscription refresh cycles to make changes to what is being generated on the device. On modern devices, enabling more event channels and expanding the size of event log files hasn't resulted in noticeable performance differences.

For the minimum recommended audit policy and registry system ACL settings, see Appendix A - Minimum recommended minimum audit policy and Appendix B - Recommended minimum registry system ACL policy.

> ⓘ **Note**
>
> These are only minimum values need to meet what the WEF subscription selects.

From a WEF subscription management perspective, the event queries provided should be used in two separate subscriptions for ease of maintenance; only machines meeting specific criteria would be allowed access to the targeted subscription. This access would be determined by an algorithm or an analysts' direction. All devices should have access to the Baseline subscription.

This system of dual subscription means you would create two base subscriptions:

- **Baseline WEF subscription**. Events collected from all hosts; these events include some role-specific events, which will only be emitted by those machines.
- **Targeted WEF subscription**. Events collected from a limited set of hosts due to unusual activity and/or heightened awareness for those systems.

Each using the respective event query below. For the Targeted subscription, enabling the "read existing events" option should be set to true to allow collection of existing events from systems. By default, WEF subscriptions will only forward events generated after the WEF subscription was received by the client.

In Appendix E - Annotated Baseline Subscription Event Query and Appendix F - Annotated Suspect Subscription Event Query, the event query XML is included when creating WEF subscriptions. These subscriptions are annotated for query purpose and clarity. Individual `<Query>` element can be removed or edited without affecting the rest of the query.

Common WEF questions

This section addresses common questions from IT pros and customers.

Will the user notice if their machine is enabled for WEF or if WEF encounters an error?

The short answer is: No.

The longer answer is: The **Eventlog-forwardingPlugin/Operational** event channel logs the success, warning, and error events related to WEF subscriptions present on the device. Unless the user opens Event Viewer and navigates to that channel, they won't notice WEF either through resource consumption or Graphical User Interface pop-ups. Even if there's an issue with the WEF subscription, there's no user interaction or performance degradation. All success, warning, and failure events are logged to this operational event channel.

Is WEF Push or Pull?

A WEF subscription can be configured to be pushed or pulled, but not both. The simplest, most flexible IT deployment with the greatest scalability can be achieved by using a push, or source initiated, subscription. WEF clients are configured by using a GPO and the built-in forwarding client is activated. For pull, collector initiated, the subscription on the WEC server is preconfigured with the names of the WEF Client devices from which events are to be selected. Those clients are to be configured ahead of time to allow the credentials used in the subscription to access their event logs remotely (normally by adding the credential to the **Event Log Readers** built-in local security group.) A useful scenario: closely monitoring a specific set of machines.

Will WEF work over VPN or RAS?

WEF handles VPN, RAS, and DirectAccess scenarios well and will reconnect and send any accumulated backlog of events when the connection to the WEF Collector is re-established.

How is client progress tracked?

The WEC server maintains in its registry the bookmark information and last heartbeat time for each event source for each WEF subscription. When an event source reconnects to a WEC server, the last bookmark position is sent to the device to use as a starting point to resume forwarding events. If a WEF client has no events to send, the WEF client connects periodically to send a Heartbeat to the WEC server to indicate it's active. This heartbeat value can be individually configured for each subscription.

Will WEF work in an IPv4, IPv6, or mixed IPv4/IPv6 environment?

Yes. WEF is transport agnostic and will work over IPv4 or IPv6.

Are WEF events encrypted? I see an HTTP/HTTPS option!

In a domain setting, the connection used to transmit WEF events is encrypted using Kerberos, by default (with NTLM as a fallback option, which can be disabled by using a GPO). Only the WEF collector can decrypt the connection. Additionally, the connection between WEF client and WEC server is mutually authenticated regardless of authentication type (Kerberos or NTLM.) There are GPO options to force Authentication to use Kerberos Only.

This authentication and encryption is performed regardless if HTTP or HTTPS is selected.

The HTTPS option is available if certificate based authentication is used, in cases where the Kerberos based mutual authentication isn't an option. The SSL certificate and provisioned client certificates are used to provide mutual authentication.

Do WEF Clients have a separate buffer for events?

The WEF client machines local event log is the buffer for WEF for when the connection to the WEC server is lost. To increase the "buffer size", increase the maximum file size of the specific event log file where events are being selected. For more info, see Appendix C - Event Channel Settings (enable and Channel Access) methods.

When the event log overwrites existing events (resulting in data loss if the device isn't connected to the Event Collector), there's no notification sent to the WEF collector that events are lost from the client. Neither is there an indicator that there was a gap encountered in the event stream.

What format is used for forwarded events?

WEF has two modes for forwarded events. The default is "Rendered Text" that includes the textual description of the event as you would see it in Event Viewer. This description's inclusion means that the event size is effectively doubled or tripled depending on the size of the rendered description. The alternative mode is "Events" (also sometimes referred to as "Binary" format) - which is just the event XML itself sent in binary XML format (as it would be written to the evtx file.) This format is compact and can more than double the event volume a single WEC server can accommodate.

A subscription "testSubscription" can be configured to use the Events format through the WECUTIL utility:

syntax

```
@rem required to set the DeliveryMaxItems or DeliveryMaxLatencyTime
Wecutil ss "testSubscription" /cf:Events
```

How frequently are WEF events delivered?

Event delivery options are part of the WEF subscription configuration parameters - There are three built-in subscription delivery options: Normal, Minimize Bandwidth, and Minimize Latency. A fourth, catch-all called "Custom" is available but can't be selected or configured through the WEF UI by using Event Viewer. The Custom delivery option must be selected and configured using the WECUTIL.EXE command-line application. All subscription options define a maximum event count and maximum event age, if either limit is exceeded then the accumulated events are sent to the event collector.

This table outlines the built-in delivery options:

Expand table

Event delivery optimization options	Description
Normal	This option ensures reliable delivery of events and doesn't attempt to conserve bandwidth. It's the appropriate choice unless you need tighter control over bandwidth usage or need forwarded events delivered as quickly as possible. It uses pull delivery mode, batches 5 items at a time and sets a batch timeout of 15 minutes.
Minimize bandwidth	This option ensures that the use of network bandwidth for event delivery is strictly controlled. It's an appropriate choice if you want to limit the frequency of network connections made to deliver events. It uses push delivery mode and sets a batch timeout of 6 hours. In addition, it uses a heartbeat interval of 6 hours.

Event delivery optimization options	Description
Minimize latency	This option ensures that events are delivered with minimal delay. It's an appropriate choice if you're collecting alerts or critical events. It uses push delivery mode and sets a batch timeout of 30 seconds.

For more info about delivery options, see Configure Advanced Subscription Settings.

The primary difference is in the latency which events are sent from the client. If none of the built-in options meet your requirements, you can set Custom event delivery options for a given subscription from an elevated command prompt:

Windows Command Prompt

```
@rem required to set the DeliveryMaxItems or DeliveryMaxLatencyTime
Wecutil ss "SubscriptionNameGoesHere" /cm:Custom
@rem set DeliveryMaxItems to 1 event
Wecutil ss "SubscriptionNameGoesHere" /dmi:1
@rem set DeliveryMaxLatencyTime to 10 ms
Wecutil ss "SubscriptionNameGoesHere" /dmlt:10
```

How do I control which devices have access to a WEF Subscription?

For source initiated subscriptions: Each WEF subscription on a WEC server has its own ACL for machine accounts or security groups containing machine accounts (not user accounts) that are explicitly allowed to participate in that subscription or are explicitly denied access. This ACL applies to only a single WEF subscription (since there can be multiple WEF subscriptions on a given WEC server), other WEF Subscriptions have their own separate ACL.

For collector initiated subscriptions: The subscription contains the list of machines from which the WEC server is to collect events. This list is managed at the WEC server, and the credentials used for the subscription must have access to read event logs from the WEF Clients - the credentials can be either the machine account or a domain account.

Can a client communicate to multiple WEF Event Collectors?

Yes. If you desire a High-Availability environment, configure multiple WEC servers with the same subscription configuration and publish both WEC Server URIs to WEF clients. WEF Clients forward events simultaneously to the configured subscriptions on the WEC servers, if they have the appropriate access.

What are the WEC server's limitations?

There are three factors that limit the scalability of WEC servers. The general rule for a stable WEC server on commodity hardware is planning for a total of 3,000 events per second on average for all configured subscriptions.

- **Disk I/O**. The WEC server doesn't process or validate the received event, but rather buffers the received event and then logs it to a local event log file (EVTX file). The speed of logging to the EVTX file is limited by the disk write speed. Isolating the EVTX file to its own array or using high speed disks can increase the number of events per second that a single WEC server can receive.

- **Network Connections**. While a WEF source doesn't maintain a permanent, persistent connection to the WEC server, it doesn't immediately disconnect after sending its events. This leniency means that the number of WEF sources that can simultaneously connect to the WEC server is limited to the open TCP ports available on the WEC server.

- **Registry size**. For each unique device that connects to a WEF subscription, there's a registry key (corresponding to the FQDN of the WEF Client) created to store bookmark and source heartbeat information. If this information isn't pruned to remove inactive clients, this set of registry keys can grow to an unmanageable size over time.
  - When a subscription has >1000 WEF sources connect to it over its operational lifetime, also known as lifetime WEF sources, Event Viewer can become unresponsive for a few minutes when selecting the **Subscriptions** node in the left-navigation, but will function normally afterwards.
  - At >50,000 lifetime WEF sources, Event Viewer is no longer an option and wecutil.exe (included with Windows) must be used to configure and manage subscriptions.
  - At >100,000 lifetime WEF sources, the registry won't be readable and the WEC server will likely have to be rebuilt.

Subscription information

Below lists all of the items that each subscription collects, the actual subscription XML is available in an Appendix. These items are separated out into Baseline and Targeted. The intent is to subscribe all hosts to Baseline, and then enroll (and remove) hosts on an as needed basis to the Targeted subscription.

Baseline subscription

While this subscription appears to be the largest subscription, it really is the lowest volume on a per-device basis. (Exceptions should be allowed for unusual devices - a device performing

complex developer related tasks can be expected to create an unusually high volume of process create and AppLocker events.) This subscription doesn't require special configuration on client devices to enable event channels or modify channel permissions.

The subscription is essentially a collection of query statements applied to the Event Log. This subscription means that it's modular in nature and a given query statement can be removed or changed without impacting other query statement in the subscription. Additionally, suppress statements that filter out specific events, only apply within that query statement and aren't to the entire subscription.

Baseline subscription requirements

To gain the most value out of the baseline subscription, we recommend having the following requirements set on the device to ensure that the clients are already generating the required events to be forwarded off the system.

- Apply a security audit policy that is a super-set of the recommended minimum audit policy. For more info, see Appendix A - Minimum Recommended minimum Audit Policy. This policy ensures that the security event log is generating the required events.
- Apply at least an Audit-Only AppLocker policy to devices.
  - If you're already allowing or restricting events by using AppLocker, then this requirement is met.
  - AppLocker events contain useful information, such as file hash and digital signature information for executables and scripts.
- Enable disabled event channels and set the minimum size for modern event files.
- Currently, there's no GPO template for enabling or setting the maximum size for the modern event files. This threshold must be defined by using a GPO. For more info, see Appendix C - Event Channel Settings (enable and Channel Access) methods.

The annotated event query can be found in the following. For more info, see Appendix F - Annotated Suspect Subscription Event Query.

- Anti-malware events from Windows Security. These events can be configured for any given anti-malware product easily if it writes to the Windows event log.
- Security event log Process Create events.
- AppLocker Process Create events (EXE, script, packaged App installation and execution).
- Registry modification events. For more info, see Appendix B - Recommended minimum Registry System ACL Policy.

- OS startup and shutdown
  - Startup events include operating system version, service pack level, QFE version, and boot mode.
- Service install
  - Includes what the name of the service, the image path, and who installed the service.
- Certificate Authority audit events
  - These events are only applicable on systems with the Certificate Authority role installed.
  - Logs certificate requests and responses.
- User profile events
  - Use of a temporary profile or unable to create a user profile may indicate an intruder is interactively logging into a device but not wanting to leave a persistent profile behind.
- Service start failure
  - Failure codes are localized, so you have to check the message DLL for values.
- Network share access events
  - Filter out IPC$ and /NetLogon file shares, which are expected and noisy.
- System shutdown initiate requests
  - Find out what initiated the restart of a device.
- User-initiated interactive sign out event
- Remote Desktop Services sessions connect, reconnect, or disconnect.
- EMET events, if EMET is installed.
- Event forwarding plugin events
  - For monitoring WEF subscription operations, such as Partial Success events. This event is useful for diagnosing deployment issues.
- Network share creation and deletion
  - Enables detection of unauthorized share creation.

    > ⓘ **Note**
    >
    > All shares are re-created when the device starts.

- Sign-in sessions
  - Sign-in success for interactive (local and Remote Interactive/Remote Desktop)

    - Sign-in success for services for non-built-in accounts, such as LocalSystem, LocalNetwork, and so on.
    - Sign-in success for batch sessions
    - Sign-in session close, which is sign out events for non-network sessions.

- Windows Error Reporting (Application crash events only)
    - This session can help detect early signs of intruder not familiar with enterprise environment using targeted malware.

- Event log service events
    - Errors, start events, and stop events for the Windows Event Log service.

- Event log cleared (including the Security Event Log)
    - This event could indicate an intruder that is covering their tracks.

- Special privileges assigned to new sign in
    - This assignation indicates that at the time of signing in, a user is either an Administrator or has the sufficient access to make themselves Administrator.

- Outbound Remote Desktop Services session attempts
    - Visibility into potential beachhead for intruder

- System time changed

- SMB Client (mapped drive connections)

- Account credential validation
    - Local accounts or domain accounts on domain controllers

- A user was added or removed from the local Administrators security group.

- Crypto API private key accessed
    - Associated with signing objects using the locally stored private key.

- Task Scheduler task creation and delete
    - Task Scheduler allows intruders to run code at specified times as LocalSystem.

- Sign-in with explicit credentials
    - Detect credential use changes by intruders to access more resources.

- Smartcard card holder verification events
    - This event detects when a smartcard is being used.

Suspect subscription

This subscription adds some possible intruder-related activity to help analyst further refine their determinations about the state of the device.

- Sign-in session creation for network sessions
  - Enables time-series analysis of network graphs.
- RADIUS and VPN events
  - Useful if you use a Microsoft IAS RADIUS/VPN implementation. It shows user-> IP address assignment with remote IP address connecting to the enterprise.
- Crypto API X509 object and build chain events
  - Detects known bad certificate, CA, or sub-CA
  - Detects unusual process use of CAPI
- Groups assigned to local sign in
  - Gives visibility to groups that enable account-wide access
  - Allows better planning for remediation efforts
  - Excludes well known, built-in system accounts.
- Sign-in session exit
  - Specific for network sign-in sessions.
- Client DNS lookup events
  - Returns what process performed a DNS query and the results returned from the DNS server.
- Process exit
  - Enables checking for processes terminating unexpectedly.
- Local credential validation or signing in with explicit credentials
  - Generated when the local SAM is authoritative for the account credentials being authenticated.
  - Noisy on domain controllers
  - On client devices, it's only generated when local accounts sign in.
- Registry modification audit events
  - Only when a registry value is being created, modified, or deleted.
- Wireless 802.1x authentication
  - Detect wireless connection with a peer MAC address
- Windows PowerShell logging
  - Covers Windows PowerShell 2.0 and later and includes the Windows PowerShell 5.0 logging improvements for in-memory attacks using Windows PowerShell.
  - Includes Windows PowerShell remoting logging

- User Mode Driver Framework "Driver Loaded" event
  - Can possibly detect a USB device loading multiple device drivers. For example, a USB_STOR device loading the keyboard or network driver.

Appendix A - Minimum recommended minimum audit policy

If your organizational audit policy enables more auditing to meet its needs, that is fine. The policy below is the minimum audit policy settings needed to enable events collected by both baseline and targeted subscriptions.

Expand table

Category	Subcategory	Audit settings
Account Logon	Credential Validation	Success and Failure
Account Management	Security Group Management	Success
Account Management	User Account Management	Success and Failure
Account Management	Computer Account Management	Success and Failure
Account Management	Other Account Management Events	Success and Failure
Detailed Tracking	Process Creation	Success
Detailed Tracking	Process Termination	Success
Logon/Logoff	User/Device Claims	Not configured
Logon/Logoff	IPsec Extended Mode	Not configured
Logon/Logoff	IPsec Quick Mode	Not configured
Logon/Logoff	Logon	Success and Failure
Logon/Logoff	Logoff	Success
Logon/Logoff	Other Logon/Logoff Events	Success and Failure
Logon/Logoff	Special Logon	Success and Failure
Logon/Logoff	Account Lockout	Success
Object Access	Application Generated	Not configured
Object Access	File Share	Success

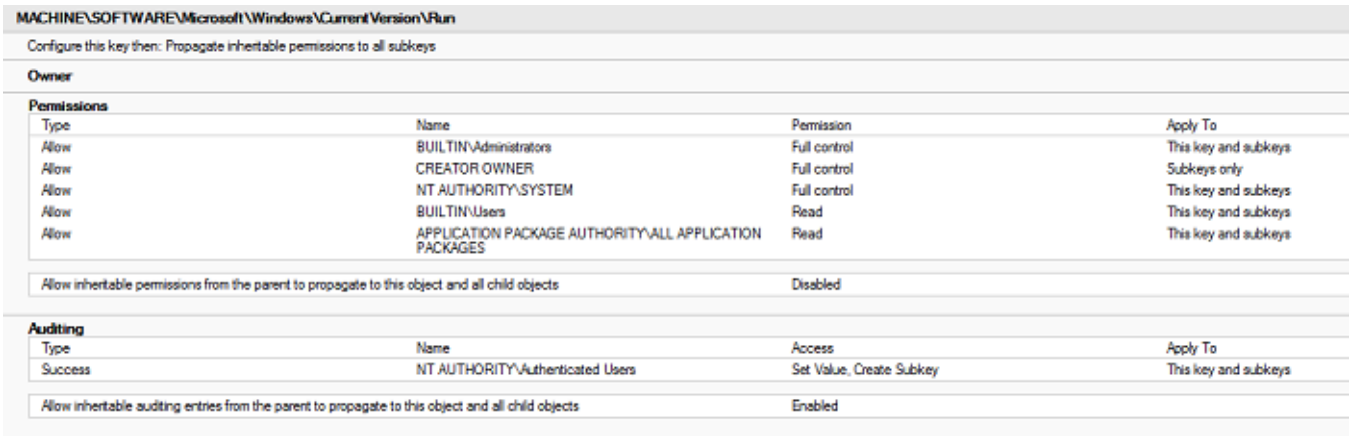
Category	Subcategory	Audit settings
Object Access	File System	Not configured
Object Access	Other Object Access Events	Not configured
Object Access	Registry	Not configured
Object Access	Removable Storage	Success
Policy Change	Audit Policy Change	Success and Failure
Policy Change	MPSSVC Rule-Level Policy Change	Success and Failure
Policy Change	Other Policy Change Events	Success and Failure
Policy Change	Authentication Policy Change	Success and Failure
Policy Change	Authorization Policy Change	Success and Failure
Privilege Use	Sensitive Privilege Use	Not configured
System	Security State Change	Success and Failure
System	Security System Extension	Success and Failure
System	System Integrity	Success and Failure

Appendix B - Recommended minimum registry system ACL policy

The Run and RunOnce keys are useful for intruders and malware persistence. It allows code to be run (or run only once then removed, respectively) when a user signs in to the system.

This implication can easily be extended to other Auto-Execution Start Points keys in the registry.

Use the following figures to see how you can configure those registry keys.



MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

Configure this key then: Propagate inheritable permissions to all subkeys

Owner

Permissions

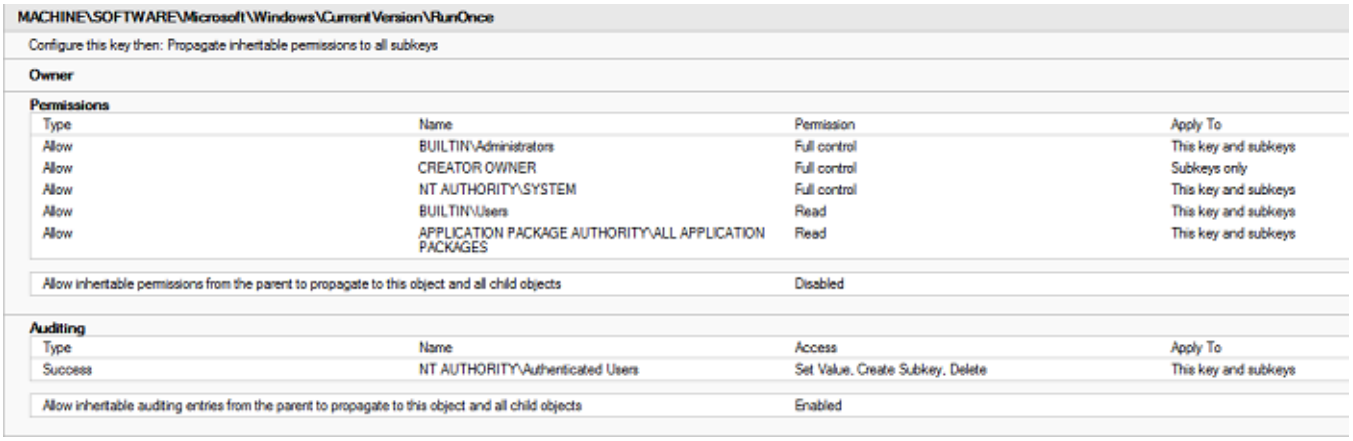
Type	Name	Permission	Apply To
Allow	BUILTIN\Administrators	Full control	This key and subkeys
Allow	CREATOR OWNER	Full control	Subkeys only
Allow	NT AUTHORITY\SYSTEM	Full control	This key and subkeys
Allow	BUILTIN\Users	Read	This key and subkeys
Allow	APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES	Read	This key and subkeys

Allow inheritable permissions from the parent to propagate to this object and all child objects	Disabled

Auditing

Type	Name	Access	Apply To
Success	NT AUTHORITY\Authenticated Users	Set Value, Create Subkey	This key and subkeys

Allow inheritable auditing entries from the parent to propagate to this object and all child objects	Enabled



MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce

Configure this key then: Propagate inheritable permissions to all subkeys

Owner

Permissions

Type	Name	Permission	Apply To
Allow	BUILTIN\Administrators	Full control	This key and subkeys
Allow	CREATOR OWNER	Full control	Subkeys only
Allow	NT AUTHORITY\SYSTEM	Full control	This key and subkeys
Allow	BUILTIN\Users	Read	This key and subkeys
Allow	APPLICATION PACKAGE AUTHORITY\ALL APPLICATION PACKAGES	Read	This key and subkeys

Allow inheritable permissions from the parent to propagate to this object and all child objects	Disabled

Auditing

Type	Name	Access	Apply To
Success	NT AUTHORITY\Authenticated Users	Set Value, Create Subkey, Delete	This key and subkeys

Allow inheritable auditing entries from the parent to propagate to this object and all child objects	Enabled

Appendix C - Event channel settings (enable and channel access) methods

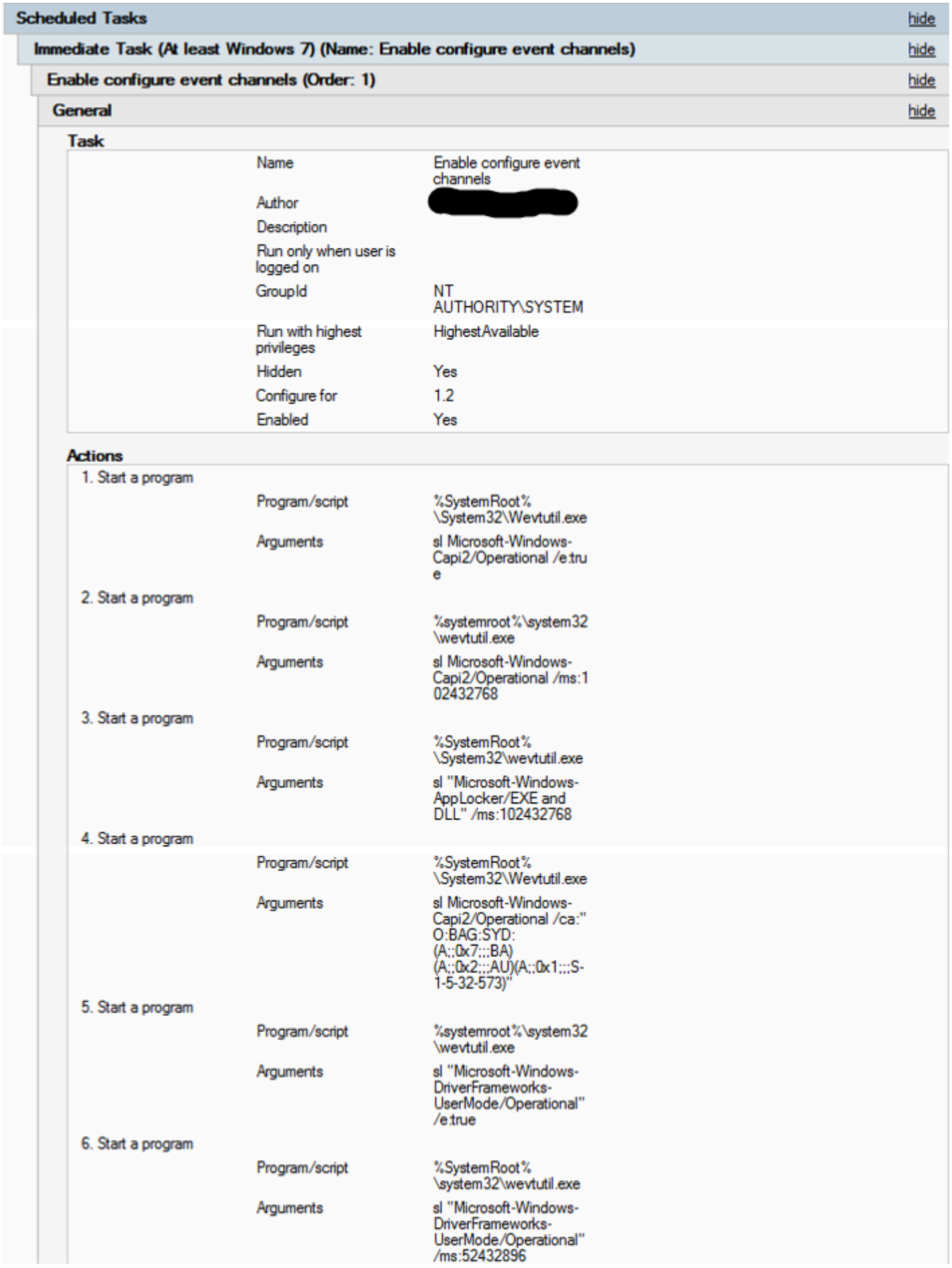
Some channels are disabled by default and have to be enabled. Others, such as Microsoft-Windows-CAPI2/Operational must have the channel access modified to allow the Event Log Readers built-in security group to read from it.

The recommended and most effective way to do this customization is configuring the baseline GPO to run a scheduled task to configure the event channels (enable, set maximum size, and adjust channel access). This configuration will take effect at the next GPO refresh cycle and has minimal impact on the client device.

The following GPO snippet performs the following tasks:

- Enables the **Microsoft-Windows-Capi2/Operational** event channel.
- Sets the maximum file size for **Microsoft-Windows-Capi2/Operational** to 100MB.
- Sets the maximum file size for **Microsoft-Windows-AppLocker/EXE and DLL** to 100 MB.
- Sets the maximum channel access for **Microsoft-Windows-Capi2/Operational** to include the built-in Event Log Readers security group.
- Enables the **Microsoft-Windows-DriverFrameworks-UserMode/Operational** event channel.

- Sets the maximum file size for **Microsoft-Windows-DriverFrameworks-UserMode/Operational** to 50 MB.



Scheduled Tasks hide

Immediate Task (At least Windows 7) (Name: Enable configure event channels) hide

Enable configure event channels (Order: 1) hide

General hide

Task

Name	Enable configure event channels
Author	[redacted]
Description	
Run only when user is logged on	
GroupId	NT AUTHORITY\SYSTEM
Run with highest privileges	HighestAvailable
Hidden	Yes
Configure for	1.2
Enabled	Yes

Actions

1. Start a program

Program/script	%SystemRoot%\System32\Wevtutil.exe
Arguments	sl Microsoft-Windows-Capi2/Operational /e:true

2. Start a program

Program/script	%systemroot%\system32\wevtutil.exe
Arguments	sl Microsoft-Windows-Capi2/Operational /ms:102432768

3. Start a program

Program/script	%SystemRoot%\System32\wevtutil.exe
Arguments	sl "Microsoft-Windows-AppLocker/EXE and DLL" /ms:102432768

4. Start a program

Program/script	%SystemRoot%\System32\Wevtutil.exe
Arguments	sl Microsoft-Windows-Capi2/Operational /ca:"O:BAG:SYD:(A;;0x7;;;BA)(A;;0x2;;;AU)(A;;0x1;;;S-1-5-32-573)"

5. Start a program

Program/script	%systemroot%\system32\wevtutil.exe
Arguments	sl "Microsoft-Windows-DriverFrameworks-UserMode/Operational" /e:true

6. Start a program

Program/script	%SystemRoot%\system32\wevtutil.exe
Arguments	sl "Microsoft-Windows-DriverFrameworks-UserMode/Operational" /ms:52432896

The following table also contains the six actions to configure in the GPO:

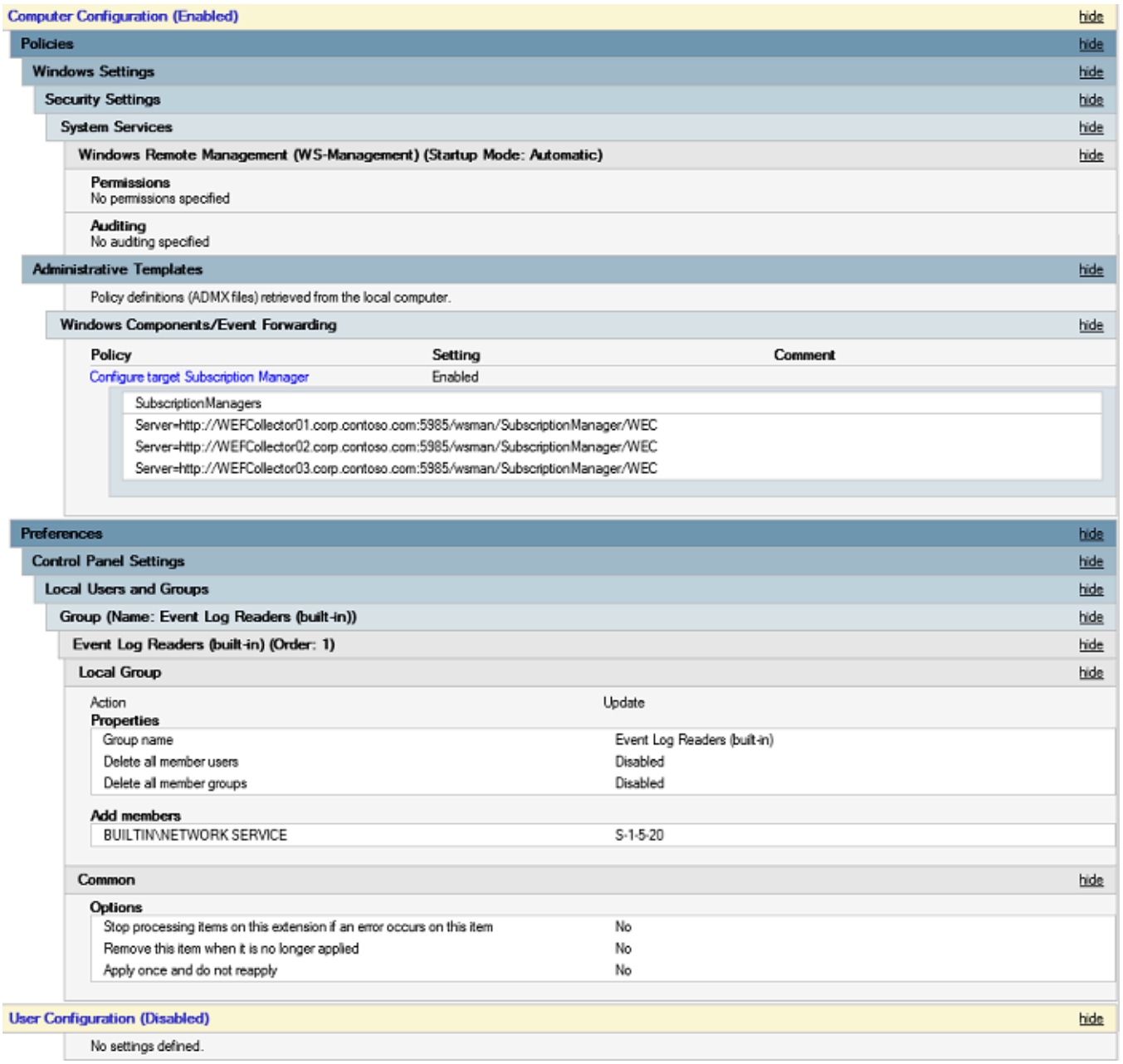
Expand table

Program/Script	Arguments
%SystemRoot%\System32\wevtutil.exe	sl Microsoft-Windows-CAPI2/Operational /e:true
%SystemRoot%\System32\wevtutil.exe	sl Microsoft-Windows-CAPI2/Operational /ms:102432768
%SystemRoot%\System32\wevtutil.exe	sl "Microsoft-Windows-AppLocker/EXE and DLL" /ms:102432768
%SystemRoot%\System32\wevtutil.exe	sl Microsoft-Windows-CAPI2/Operational /ca:"O:BAG:SYD:(A;;0x7;;;BA)(A;;0x2;;;AU)(A;;0x1;;;S-1-5-32-573)"
%SystemRoot%\System32\wevtutil.exe	sl "Microsoft-Windows-DriverFrameworks-UserMode/Operational" /e:true
%SystemRoot%\System32\wevtutil.exe	sl "Microsoft-Windows-DriverFrameworks-UserMode/Operational" /ms:52432896

Appendix D - Minimum GPO for WEF Client configuration

Here are the minimum steps for WEF to operate:

1. Configure the collector URI(s).
2. Start the WinRM service.
3. Add the Network Service account to the built-in Event Log Readers security group. This addition allows reading from secured event channel, such as the security event channel.



Appendix E - Annotated baseline subscription event query

XML

```
<QueryList>
  <Query Id="0" Path="System">
    <!-- Anti-malware *old* events, but only detect events (cuts down noise) -->
    <Select Path="System">*[System[Provider[@Name='Microsoft Antimalware'] and
(EventID &gt;= 1116 and EventID &lt;= 1119)]]</Select>
  </Query>
  <!-- AppLocker EXE events or Script events -->
  <Query Id="1" Path="Microsoft-Windows-AppLocker/EXE and DLL">
    <Select Path="Microsoft-Windows-AppLocker/EXE and DLL">*
[UserData[RuleAndFileData[PolicyName="EXE"]]]</Select>
    <Select Path="Microsoft-Windows-AppLocker/MSI and Script">*</Select>
  </Query>
```

```
  <Query Id="2" Path="Security">
    <!-- Wireless Lan 802.1x authentication events with Peer MAC address -->
    <Select Path="Security">*[System[(EventID=5632)]]</Select>
  </Query>
  <Query Id="3" Path="Microsoft-Windows-TaskScheduler/Operational">
    <!-- Task scheduler Task Registered (106),  Task Registration Deleted (141),
Task Deleted (142) -->
    <Select Path="Microsoft-Windows-TaskScheduler/Operational">*
[System[Provider[@Name='Microsoft-Windows-TaskScheduler'] and (EventID=106 or
EventID=141 or EventID=142 )]]</Select>
    <Select Path="System">*[System[Provider[@Name='Microsoft-Windows-TaskScheduler']
and (EventID=106 or EventID=141 or EventID=142 )]]</Select>
  </Query>
  <Query Id="4" Path="System">
    <!-- System startup (12 - includes OS/SP/Version) and shutdown -->
    <Select Path="System">*[System[Provider[@Name='Microsoft-Windows-Kernel-
General'] and (EventID=12 or EventID=13)]]</Select>
  </Query>
  <Query Id="5" Path="System">
    <!-- Service Install (7000), service start failure (7045), new service (4697) --
>
    <Select Path="System">*[System[Provider[@Name='Service Control Manager'] and
(EventID = 7000 or EventID=7045)]]</Select>
<Select Path="Security">*[System[(EventID=4697)]]</Select>
  </Query>
  <Query Id="6" Path="Security">
    <!-- TS Session reconnect (4778), TS Session disconnect (4779) -->
    <Select Path="Security">*[System[(EventID=4778 or EventID=4779)]]</Select>
  </Query>
  <Query Id="7" Path="Security">
    <!-- Network share object access without IPC$ and Netlogon shares -->
    <Select Path="Security">*[System[(EventID=5140)]] and (*
[EventData[Data[@Name="ShareName"]!="\\*\IPC$"]]) and (*
[EventData[Data[@Name="ShareName"]!="\\*\NetLogon"]])</Select>
  </Query>
  <Query Id="8" Path="Security">
    <!-- System Time Change (4616)  -->
    <Select Path="Security">*[System[(EventID=4616)]]</Select>
  </Query>
  <Query Id="9" Path="System">
    <!-- Shutdown initiate requests, with user, process and reason (if supplied) -->
    <Select Path="System">*[System[Provider[@Name='USER32'] and (EventID=1074)]]
</Select>
  </Query>
  <!-- AppLocker packaged (Modern UI) app execution -->
  <Query Id="10" Path="Microsoft-Windows-AppLocker/Packaged app-Execution">
    <Select Path="Microsoft-Windows-AppLocker/Packaged app-Execution">*</Select>
  </Query>
  <!-- AppLocker packaged (Modern UI) app installation -->
  <Query Id="11" Path="Microsoft-Windows-AppLocker/Packaged app-Deployment">
    <Select Path="Microsoft-Windows-AppLocker/Packaged app-Deployment">*</Select>
  </Query>
  <Query Id="12" Path="Application">
    <!-- EMET events -->
    <Select Path="Application">*[System[Provider[@Name='EMET']]]</Select>
```

```
  </Query>
  <Query Id="13" Path="System">
    <!-- Event log service events -->
    <Select Path="System">*[System[Provider[@Name='Microsoft-Windows-Eventlog']]]
</Select>
  </Query>
  <Query Id="14" Path="Security">
    <!-- Local logons without network or service events -->
    <Select Path="Security">*[System[(EventID=4624)]] and (*
[EventData[Data[@Name="LogonType"]!="3"]]) and (*
[EventData[Data[@Name="LogonType"]!="5"]])</Select>
  </Query>
  <Query Id="15" Path="Application">
    <!-- WER events for application crashes only -->
    <Select Path="Application">*[System[Provider[@Name='Windows Error Reporting']]]
and (*[EventData[Data[3] ="APPCRASH"]])</Select>
  </Query>
  <Query Id="16" Path="Security">
    <!-- Security Log cleared events (1102), EventLog Service shutdown (1100)-->
    <Select Path="Security">*[System[(EventID=1102 or EventID = 1100)]]</Select>
  </Query>
  <Query Id="17" Path="System">
    <!-- Other Log cleared events (104)-->
    <Select Path="System">*[System[(EventID=104)]]</Select>
  </Query>
  <Query Id="18" Path="Security">
    <!--  user initiated logoff -->
    <Select Path="Security">*[System[(EventID=4647)]]</Select>
  </Query>
  <Query Id="19" Path="Security">
    <!-- user logoff for all non-network logon sessions-->
    <Select Path="Security">*[System[(EventID=4634)]] and (*
[EventData[Data[@Name="LogonType"] != "3"]])</Select>
  </Query>
  <Query Id="20" Path="Security">
    <!-- Service logon events if the user account isn't LocalSystem, NetworkService,
LocalService -->
    <Select Path="Security">*[System[(EventID=4624)]] and (*
[EventData[Data[@Name="LogonType"]="5"]]) and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-18"]]) and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-19"]]) and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-20"]])</Select>
  </Query>
  <Query Id="21" Path="Security">
    <!-- Network Share create (5142), Network Share Delete (5144)  -->
    <Select Path="Security">*[System[(EventID=5142 or EventID=5144)]]</Select>
  </Query>
  <Query Id="22" Path="Security">
    <!-- Process Create (4688) -->
    <Select Path="Security">*[System[EventID=4688]]</Select>
  </Query>
  <Query Id="23" Path="Security">
    <!-- Event log service events specific to Security channel -->
    <Select Path="Security">*[System[Provider[@Name='Microsoft-Windows-Eventlog']]]
</Select>
```

```
  </Query>
  <Query Id="26" Path="Security">
    <!-- Special Privileges (Admin-equivalent Access) assigned to new logon,
excluding LocalSystem-->
    <Select Path="Security">*[System[(EventID=4672)]]</Select>
    <Suppress Path="Security">*[EventData[Data[1]="S-1-5-18"]]</Suppress>
  </Query>
  <Query Id="27" Path="Security">
    <!-- New user added to local security group-->
    <Select Path="Security">*[System[(EventID=4732)]]</Select>
  </Query>
  <Query Id="28" Path="Security">
    <!-- New user added to global security group-->
    <Select Path="Security">*[System[(EventID=4728)]]</Select>
  </Query>
  <Query Id="29" Path="Security">
    <!-- New user added to universal security group-->
    <Select Path="Security">*[System[(EventID=4756)]]</Select>
  </Query>
  <Query Id="30" Path="Security">
    <!-- User removed from local Administrators group-->
    <Select Path="Security">*[System[(EventID=4733)]] and (*
[EventData[Data[@Name="TargetUserName"]="Administrators"]])</Select>
  </Query>
  <Query Id="31" Path="Microsoft-Windows-TerminalServices-RDPClient/Operational">
    <!-- Log attempted TS connect to remote server -->
    <Select Path="Microsoft-Windows-TerminalServices-RDPClient/Operational">*
[System[(EventID=1024)]]</Select>
  </Query>
  <Query Id="32" Path="Security">
    <!-- Certificate Services received certificate request (4886), Approved and
Certificate issued (4887), Denied request (4888) -->
    <Select Path="Security">*[System[(EventID=4886 or EventID=4887 or
EventID=4888)]]</Select>
  </Query>
  <Query Id="34" Path="Security">
    <!-- New User Account Created(4720), User Account Enabled (4722), User Account
Disabled (4725), User Account Deleted (4726) -->
    <Select Path="Security">*[System[(EventID=4720 or EventID=4722 or EventID=4725
or EventID=4726)]]</Select>
  </Query>
  <Query Id="35" Path="Microsoft-Windows-SmartCard-Audit/Authentication">
    <!-- Gets all Smart-card Card-Holder Verification (CHV) events (success and
failure) performed on the host. -->
    <Select Path="Microsoft-Windows-SmartCard-Audit/Authentication">*</Select>
  </Query>
  <Query Id="36" Path="Microsoft-Windows-SMBClient/Operational">
    <!-- get all UNC/mapped drive successful connection -->
    <Select Path="Microsoft-Windows-SMBClient/Operational">*[System[(EventID=30622
or EventID=30624)]]</Select>
  </Query>
  <Query Id="37" Path="Application">
    <!-- User logging on with Temporary profile (1511), cannot create profile, using
temporary profile (1518)-->
    <Select Path="Application">*[System[Provider[@Name='Microsoft-Windows-User
```

```
Profiles Service'] and (EventID=1511 or EventID=1518)]]</Select>
  </Query>
  <Query Id="39" Path="Microsoft-Windows-Sysmon/Operational">
    <!-- Modern SysMon event provider-->
    <Select Path="Microsoft-Windows-Sysmon/Operational">*</Select>
  </Query>
  <Query Id="40" Path="Application">
    <!-- Application crash/hang events, similar to WER/1001. These include full path
to faulting EXE/Module.-->
    <Select Path="Application">*[System[Provider[@Name='Application Error'] and
(EventID=1000)]]</Select>
    <Select Path="Application">*[System[Provider[@Name='Application Hang'] and
(EventID=1002)]]</Select>
  </Query>
  <Query Id="41" Path="Microsoft-Windows-Windows Defender/Operational">
    <!-- Modern Windows Defender event provider Detection events (1006-1009) and
(1116-1119) -->
    <Select Path="Microsoft-Windows-Windows Defender/Operational">*[System[(
(EventID &gt;= 1006 and EventID &lt;= 1009) )]]</Select>
    <Select Path="Microsoft-Windows-Windows Defender/Operational">*[System[(
(EventID &gt;= 1116 and EventID &lt;= 1119) )]]</Select>
  </Query>
  <Query Id="42" Path="Security">
    <!-- An account Failed to Log on events -->
    <Select Path="Security">*[System[(EventID=4625)]] and (*
[EventData[Data[@Name="LogonType"]!="2"]]) </Select>
  </Query>

</QueryList>
```

Appendix F - Annotated Suspect Subscription Event Query

XML

```
<QueryList>
  <Query Id="0" Path="Security">
    <!-- Network logon events-->
    <Select Path="Security">*[System[(EventID=4624)]] and (*
[EventData[Data[@Name="LogonType"]="3"]])</Select>
  </Query>
  <Query Id="1" Path="System">
    <!-- RADIUS authentication events User Assigned IP address (20274), User
successfully authenticated (20250), User Disconnected (20275)  -->
    <Select Path="System">*[System[Provider[@Name='RemoteAccess'] and (EventID=20274
or EventID=20250 or EventID=20275)]]</Select>
  </Query>
  <Query Id="2" Path="Microsoft-Windows-CAPI2/Operational">
    <!-- CAPI events Build Chain (11), Private Key accessed (70), X509 object (90)--
>
    <Select Path="Microsoft-Windows-CAPI2/Operational">*[System[(EventID=11 or
```

```
EventID=70 or EventID=90)]]</Select>
  </Query>
  <Query Id="3" Path="Security">
    <!-- CA stop/Start events CA Service Stopped (4880), CA Service Started (4881),
CA DB row(s) deleted (4896), CA Template loaded (4898) -->
    <Select Path="Security">*[System[(EventID=4880 or EventID = 4881 or EventID =
4896 or EventID = 4898)]]</Select>
  </Query>
  <Query Id="4" Path="Microsoft-Windows-LSA/Operational">
    <!-- Groups assigned to new login (except for well known, built-in accounts)-->
    <Select Path="Microsoft-Windows-LSA/Operational">*[System[(EventID=300)]] and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-20"]]) and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-18"]]) and (*
[EventData[Data[@Name="TargetUserSid"] != "S-1-5-19"]])</Select>
  </Query>
  <Query Id="5" Path="Security">
    <!-- Logoff events - for Network Logon events-->
    <Select Path="Security">*[System[(EventID=4634)]] and (*
[EventData[Data[@Name="LogonType"] = "3"]])</Select>
  </Query>
  <Query Id="6" Path="Security">
    <!-- RRAS events - only generated on Microsoft IAS server -->
    <Select Path="Security">*[System[( (EventID &gt;= 6272 and EventID &lt;= 6280)
)]]</Select>
  </Query>
  <Query Id="7" Path="Microsoft-Windows-DNS-Client/Operational">
    <!-- DNS Client events Query Completed (3008) -->
    <Select Path="Microsoft-Windows-DNS-Client/Operational">*
[System[(EventID=3008)]]</Select>
<!-- suppresses local machine name resolution events -->
<Suppress Path="Microsoft-Windows-DNS-Client/Operational">*
[EventData[Data[@Name="QueryOptions"]="140737488355328"]]</Suppress>
<!-- suppresses empty name resolution events -->
<Suppress Path="Microsoft-Windows-DNS-Client/Operational">*
[EventData[Data[@Name="QueryResults"]=""]]</Suppress>
  </Query>
  <Query Id="8" Path="Security">
    <!-- Process Terminate (4689) -->
    <Select Path="Security">*[System[(EventID = 4689)]]</Select>
  </Query>
  <Query Id="9" Path="Security">
    <!-- Local credential authentication events (4776), Logon with explicit
credentials (4648) -->
    <Select Path="Security">*[System[(EventID=4776 or EventID=4648)]]</Select>
  </Query>
  <Query Id="10" Path="Security">
    <!-- Registry modified events for Operations: New Registry Value created
(%%1904), Existing Registry Value modified (%%1905), Registry Value Deleted (%%1906)
-->
    <Select Path="Security">*[System[(EventID=4657)]] and ((*
[EventData[Data[@Name="OperationType"] = "%%1904"]]) or (*
[EventData[Data[@Name="OperationType"] = "%%1905"]]) or (*
[EventData[Data[@Name="OperationType"] = "%%1906"]]))</Select>
  </Query>
  <Query Id="11" Path="Security">
```

```
    <!-- Request made to authenticate to Wireless network (including Peer MAC (5632)
-->
    <Select Path="Security">*[System[(EventID=5632)]]</Select>
  </Query>
  <Query Id="12" Path="Microsoft-Windows-PowerShell/Operational">
    <!-- PowerShell execute block activity (4103), Remote Command(4104), Start
Command(4105), Stop Command(4106) -->
    <Select Path="Microsoft-Windows-PowerShell/Operational">*[System[(EventID=4103
or EventID=4104 or EventID=4105 or EventID=4106)]]</Select>
  </Query>
  <Query Id="13" Path="Microsoft-Windows-DriverFrameworks-UserMode/Operational">
    <!-- Detect User-Mode drivers loaded - for potential BadUSB detection. -->
    <Select Path="Microsoft-Windows-DriverFrameworks-UserMode/Operational">*
[System[(EventID=2004)]]</Select>
  </Query>
<Query Id="14" Path="Windows PowerShell">
    <!-- Legacy PowerShell pipeline execution details (800) -->
    <Select Path="Windows PowerShell">*[System[(EventID=800)]]</Select>
  </Query>
</QueryList>
```

Appendix G - Online resources

You can get more info with the following links:

- Event Selection
- Event Queries and Event XML
- Event Query Schema
- Windows Event Collector

Last updated on 08/18/2025

Block untrusted fonts in an enterprise

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

To help protect your company from attacks that may originate from untrusted or attacker-controlled font files, you can block untrusted fonts. Using this feature, you can turn on a global setting that stops your employees from loading untrusted fonts processed using the Graphics Device Interface (GDI) onto your network. Untrusted fonts are any font installed outside of the `%windir%\Fonts` directory. Blocking untrusted fonts helps prevent both remote (web-based or email-based) and local EOP attacks that can happen during the font file-parsing process.

What does this mean for me?

Blocking untrusted fonts helps improve your network and employee protection against font-processing-related attacks. By default, this feature isn't turned on.

How does this feature work?

There are three ways to use this feature:

- **On.** Helps stop any font processed using GDI from loading outside of the `%windir%\Fonts` directory. It also turns on event logging.
- **Audit.** Turns on event logging, but doesn't block fonts from loading, regardless of location. The name of the apps that use untrusted fonts appear in your event log.

  > ⓘ Note
  >
  > If you aren't quite ready to deploy this feature into your organization, you can run it in Audit mode to see if not loading untrusted fonts causes any usability or compatibility issues.

- **Exclude apps to load untrusted fonts.** You can exclude specific apps, allowing them to load untrusted fonts, even while this feature is turned on. For instructions, see Fix apps having problems because of blocked fonts.

Potential reductions in functionality

After you turn on this feature, your employees might experience reduced functionality when:

- Sending a print job to a remote printer server that uses this feature and where the spooler process hasn't been excluded. In this situation, any fonts that aren't already available in the server's %windir%/Fonts folder won't be used.
- Printing using fonts provided by the installed printer's graphics .dll file, outside of the %windir%/Fonts folder. For more information, see Introduction to Printer Graphics DLLs.
- Using first or non-Microsoft apps that use memory-based fonts.
- Using Internet Explorer to look at websites that use embedded fonts. In this situation, the feature blocks the embedded font, causing the website to use a default font. However, not all fonts have all of the characters, so the website might render differently.
- Using desktop Office to look at documents with embedded fonts. In this situation, content shows up using a default font picked by Office.

Turn on and use the Blocking Untrusted Fonts feature

Use Group Policy or the registry to turn this feature on, off, or to use audit mode.

GPO

1. Open the Group Policy editor (gpedit.msc) and go to `Computer Configuration\Administrative Templates\System\Mitigation Options\Untrusted Font Blocking`.
2. Select **Enabled** to turn on the feature, and then select one of the following **Mitigation Options**:

   - **Block untrusted fonts and log events.** Turns on the feature, blocking untrusted fonts and logging installation attempts to the event log.
   - **Do not block untrusted fonts.** Turns on the feature, but doesn't block untrusted fonts nor does it log installation attempts to the event log.
   - **Log events without blocking untrusted fonts**. Turns on the feature, logging installation attempts to the event log, but not blocking untrusted fonts.

3. Select **OK**.

View the event log

After you turn on this feature, or start using Audit mode, you can look at your event logs for details.

To look at your event log

1. Open the event viewer (eventvwr.exe) and go to **Application and Service Logs/Microsoft/Windows/Win32k/Operational**.
2. Scroll down to **EventID: 260** and review the relevant events.

Event Example 1 - MS Word

> WINWORD.EXE attempted loading a font that is restricted by font-loading policy.
> FontType: Memory
> FontPath:
> Blocked: true

> **Note**
>
> Because the **FontType** is *Memory*, there's no associated **FontPath**.

Event Example 2 - Winlogon

> Winlogon.exe attempted loading a font that is restricted by font-loading policy.
> FontType: File
> FontPath: `\??\C:\PROGRAM FILES (X86)\COMMON FILES\MICROSOFT SHARED\EQUATION\MTEXTRA.TTF`
> Blocked: true

> **Note**
>
> Because the **FontType** is *File*, there's also an associated **FontPath**.

Event Example 3 - Internet Explorer running in Audit mode

> Iexplore.exe attempted loading a font that is restricted by font-loading policy.
> FontType: Memory
> FontPath:
> Blocked: false

> **Note**
>
> In Audit mode, the problem is recorded, but the font isn't blocked.

Fix apps having problems because of blocked fonts

Your company may still need apps that are having problems because of blocked fonts, so we suggest that you first run this feature in Audit mode to determine which fonts are causing the problems.

After you figure out the problematic fonts, you can try to fix your apps in two ways: by directly installing the fonts into the %windir%/Fonts directory or by excluding the underlying processes and letting the fonts load. As the default solution, we highly recommend that you install the problematic font. Installing fonts is safer than excluding apps because excluded apps can load any font, trusted or untrusted.

To fix your apps by installing the problematic fonts (recommended)

On each computer with the app installed, right-click on the font name and select **Install**. The font should automatically install into your `%windir%\Fonts` directory. If it doesn't, you need to manually copy the font files into the **Fonts** directory and run the installation from there.

To fix your apps by excluding processes

1. On each computer with the app installed, open regedit.exe and go to `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<process_image_name>`. For example, if you want to exclude Microsoft Word processes, you'd use `HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Winword.exe`.
2. Add other processes that need to be excluded here, and then turn on the Blocking untrusted fonts feature, using the steps in Turn on and use the Blocking Untrusted Fonts feature, earlier in this article.

Last updated on 08/18/2025

Application Control for Windows

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Note**
>
> Some capabilities of App Control for Business are only available on specific Windows versions. Learn more about App Control feature availability.

Your organization's data is one of its most valuable assets... and adversaries want it. No matter what security controls you apply over your data, there are no controls to fully protect your most vulnerable target: the trusted user sitting at the keyboard. When a user runs a process, that process shares the same access to your data that the user has. So your sensitive information is easily transmitted, modified, deleted, or encrypted when a user, intentionally or not, runs malicious software. And with thousands of new malicious files created every day, relying solely on traditional methods like antivirus (AV) solutions gives you an inadequate defense against new attacks.

Application control changes Windows from a place where all code runs unless your AV solution confidently predicts it's bad, to one where code runs only if your policy says so. The cyber threats you face change rapidly, and your defenses need to change too. Government and security organizations, like the Australian Signals Directorate, frequently cite application control as one of the most effective ways to address the threat of executable file-based malware (.exe, .dll, etc.). It works alongside your AV solution to help mitigate security threats by restricting the apps that users can run and even what code runs in the System Core (kernel).

> **Important**
>
> Although application control can significantly harden your computers against malicious code, it's not a replacement for antivirus. You should continue to maintain an active antivirus solution alongside App Control for a well-rounded enterprise security portfolio.

Although we call it application control, the code running on your system isn't always an app. Application control extends beyond apps to also cover scripts and Microsoft installers (MSI), command-line batch files, and even interactive sessions of Windows PowerShell, which run in Constrained Language Mode.

Windows includes two application control technologies you can use depending on your organization's specific scenarios and requirements:

- **App Control for Business (app control)**; and
- **AppLocker**

App Control and Smart App Control

Starting in Windows 11 version 22H2, Smart App Control brings robust application control to consumers and to some small businesses with simpler app portfolios. Smart App Control ensures only signed code runs or code predicted to be safe by our intelligent cloud-powered security service. When code is unsigned and the service is unable to predict with confidence that it's safe to run, then we block it. Over time, the code's reputation might change as the service processes new signals it receives. Meanwhile, code determined to be unsafe is always blocked.

While Smart App Control is designed for consumers, we believe it's the ideal starting point for most organizations. And since we built it entirely upon App Control for Business, you can create a policy with the same security and compatibility as Smart App Control that also trusts the line-of-business (LOB) apps your organization needs. The service Smart App Control uses to predict what code is safe to run is also available in App Control for Business and called the Intelligent Security Graph (ISG).

Smart App Control starts in evaluation mode and switches off within 48 hours for enterprise managed devices unless the user turns it on first. If you want to proactively turn off Smart App Control across your organization's endpoints, set the **VerifiedAndReputablePolicyState** (DWORD) registry value under `HKLM\SYSTEM\CurrentControlSet\Control\CI\Policy` as shown in the following table. After you change the registry value, you must run CiTool.exe -r for the change to take effect.

Expand table

Value	Description
0	Off
1	Enforce
2	Evaluation

> **Important**
>
> Once you turn Smart App Control off, it can't be turned on without resetting or reinstalling Windows.

The App Control policy used for Smart App Control comes bundled with the App Control Wizard policy authoring tool and is also found as an example policy at *%windir%/schemas/CodeIntegrity/ExamplePolicies/SmartAppControl.xml*. To use this example policy as a starting point for your own policy, see Use the Smart App Control Policy to build your own base policy. When using the Smart App Control example policy as the basis for your own custom policy, you must remove the option **Enabled:Conditional Windows Lockdown Policy** so it's ready for use as an App Control for Business policy.

Windows edition and licensing requirements

The following table lists the Windows editions that support App Control for Business:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

App Control license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

What you should read next

- To learn more about the two application control technologies available in Windows, read App Control for Business and AppLocker Overview.
- To jump right in and get started creating policies, go revisit Smart App Control and Use the Smart App Control policy to build your own starter policy.

Last updated on 03/29/2026

Application Control for Windows

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Note**
>
> Some capabilities of App Control for Business are only available on specific Windows versions. Learn more about App Control feature availability.

Your organization's data is one of its most valuable assets... and adversaries want it. No matter what security controls you apply over your data, there are no controls to fully protect your most vulnerable target: the trusted user sitting at the keyboard. When a user runs a process, that process shares the same access to your data that the user has. So your sensitive information is easily transmitted, modified, deleted, or encrypted when a user, intentionally or not, runs malicious software. And with thousands of new malicious files created every day, relying solely on traditional methods like antivirus (AV) solutions gives you an inadequate defense against new attacks.

Application control changes Windows from a place where all code runs unless your AV solution confidently predicts it's bad, to one where code runs only if your policy says so. The cyber threats you face change rapidly, and your defenses need to change too. Government and security organizations, like the Australian Signals Directorate, frequently cite application control as one of the most effective ways to address the threat of executable file-based malware (.exe, .dll, etc.). It works alongside your AV solution to help mitigate security threats by restricting the apps that users can run and even what code runs in the System Core (kernel).

> **Important**
>
> Although application control can significantly harden your computers against malicious code, it's not a replacement for antivirus. You should continue to maintain an active antivirus solution alongside App Control for a well-rounded enterprise security portfolio.

Although we call it application control, the code running on your system isn't always an app. Application control extends beyond apps to also cover scripts and Microsoft installers (MSI), command-line batch files, and even interactive sessions of Windows PowerShell, which run in Constrained Language Mode.

Windows includes two application control technologies you can use depending on your organization's specific scenarios and requirements:

- **App Control for Business (app control)**; and
- **AppLocker**

App Control and Smart App Control

Starting in Windows 11 version 22H2, Smart App Control brings robust application control to consumers and to some small businesses with simpler app portfolios. Smart App Control ensures only signed code runs or code predicted to be safe by our intelligent cloud-powered security service. When code is unsigned and the service is unable to predict with confidence that it's safe to run, then we block it. Over time, the code's reputation might change as the service processes new signals it receives. Meanwhile, code determined to be unsafe is always blocked.

While Smart App Control is designed for consumers, we believe it's the ideal starting point for most organizations. And since we built it entirely upon App Control for Business, you can create a policy with the same security and compatibility as Smart App Control that also trusts the line-of-business (LOB) apps your organization needs. The service Smart App Control uses to predict what code is safe to run is also available in App Control for Business and called the Intelligent Security Graph (ISG).

Smart App Control starts in evaluation mode and switches off within 48 hours for enterprise managed devices unless the user turns it on first. If you want to proactively turn off Smart App Control across your organization's endpoints, set the **VerifiedAndReputablePolicyState** (DWORD) registry value under `HKLM\SYSTEM\CurrentControlSet\Control\CI\Policy` as shown in the following table. After you change the registry value, you must run CiTool.exe -r for the change to take effect.

Expand table

Value	Description
0	Off
1	Enforce
2	Evaluation

> **Important**
>
> Once you turn Smart App Control off, it can't be turned on without resetting or reinstalling Windows.

The App Control policy used for Smart App Control comes bundled with the App Control Wizard policy authoring tool and is also found as an example policy at *%windir%/schemas/CodeIntegrity/ExamplePolicies/SmartAppControl.xml*. To use this example policy as a starting point for your own policy, see Use the Smart App Control Policy to build your own base policy. When using the Smart App Control example policy as the basis for your own custom policy, you must remove the option **Enabled:Conditional Windows Lockdown Policy** so it's ready for use as an App Control for Business policy.

Windows edition and licensing requirements

The following table lists the Windows editions that support App Control for Business:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

App Control license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

What you should read next

- To learn more about the two application control technologies available in Windows, read App Control for Business and AppLocker Overview.
- To jump right in and get started creating policies, go revisit Smart App Control and Use the Smart App Control policy to build your own starter policy.

Last updated on 03/29/2026

App Control and virtualization-based protection of code integrity

Article • 10/01/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows includes a set of hardware and OS technologies that, when configured together, allow enterprises to "lock down" Windows systems so they behave more like kiosk devices. In this configuration, **App Control for Business** is used to restrict devices to run only approved apps, while the OS is hardened against kernel memory attacks using **memory integrity**.

> ⓘ **Note**
>
> Memory integrity is sometimes referred to as **hypervisor-protected code integrity (HVCI)** or **hypervisor enforced code integrity**, and was originally released as part of **Device Guard**. Device Guard is no longer used except to locate memory integrity and VBS settings in Group Policy or the Windows registry.

App Control policies and memory integrity are powerful protections that can be used separately. However, when these two technologies are configured to work together, they present a strong protection capability for Windows devices. Using App Control to restrict devices to only authorized apps has these advantages over other solutions:

1. The Windows kernel handles enforcement of App Control policy and requires no other services or agents.
2. The App Control policy takes effect early in the boot sequence before nearly all other OS code and before traditional antivirus solutions run.
3. App Control lets you set application control policy for any code that runs on Windows, including kernel mode drivers and even code that runs as part of Windows.
4. Customers can protect the App Control policy even from local administrator tampering by digitally signing the policy. Changing signed policy requires both administrative privilege and access to the organization's digital signing process. Using signed policies makes it difficult for an attacker, including one who manages to gain administrative privilege, to tamper with App Control policy.
5. You can protect the entire App Control enforcement mechanism with memory integrity. Even if a vulnerability exists in kernel mode code, memory integrity greatly reduces the likelihood that an attacker could successfully exploit it. Without

memory integrity, an attacker who compromises the kernel could normally disable most system defenses, including application control policies enforced by App Control or any other application control solution.

There are no direct dependencies between App Control and memory integrity. You can deploy them individually or together and there's no order in which they must be deployed.

Memory integrity relies on Windows Virtualization-based security, and has hardware, firmware, and kernel driver compatibility requirements that some older systems can't meet.

App Control has no specific hardware or software requirements.

Related articles

- App Control for Business
- Memory integrity
- Driver compatibility with memory integrity

Feedback

Was this page helpful? Yes No

Provide product feedback

User Account Control overview

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

User Account Control (UAC) is a Windows security feature designed to protect the operating system from unauthorized changes. When changes to the system require administrator-level permission, UAC notifies the user, giving the opportunity to approve or deny the change. UAC improves the security of Windows devices by limiting the access that malicious code has to execute with administrator privileges. UAC empowers users to make informed decisions about actions that might affect the stability and security of their device.

Unless you disable UAC, malicious software is prevented from disabling or interfering with UAC settings. UAC is enabled by default, and you can configure it if you have administrative privileges.

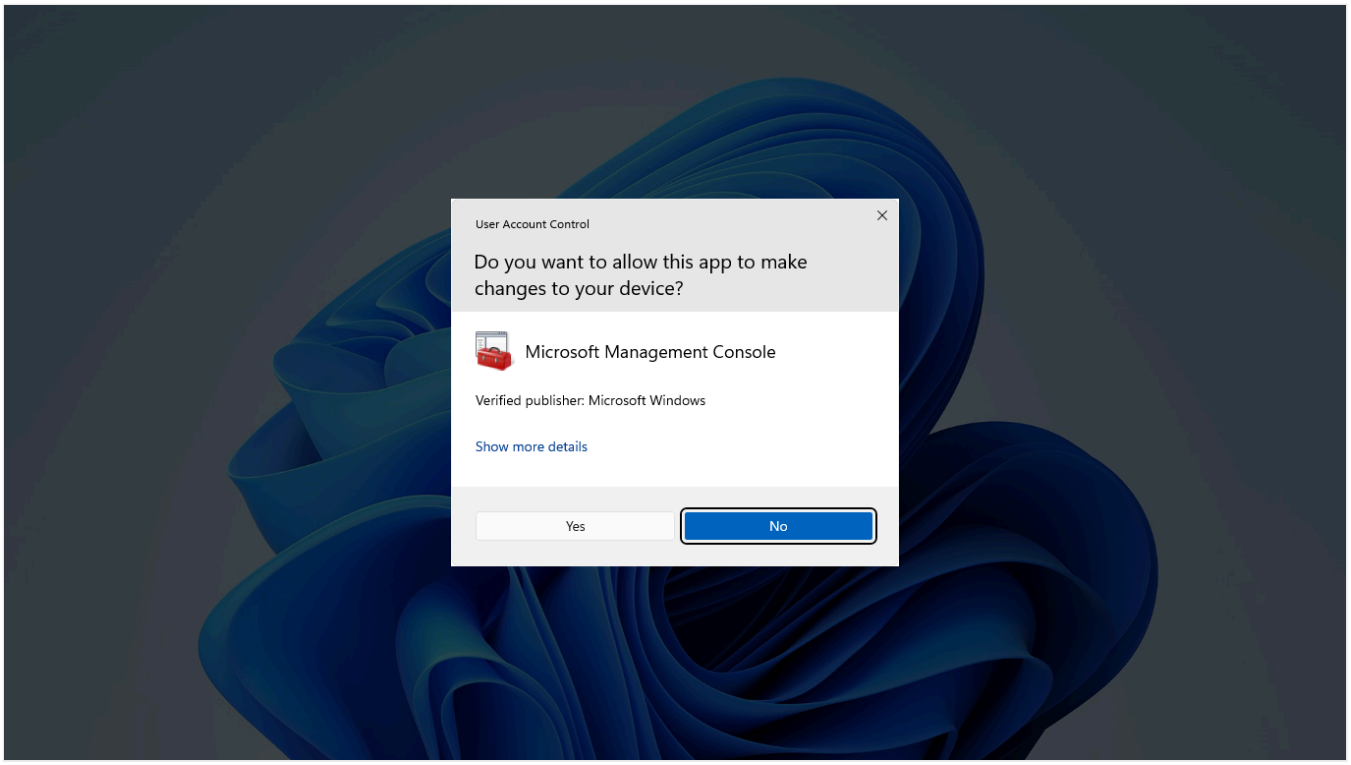
Benefits of UAC

UAC allows all users to sign in their devices using a *standard user account*. Processes launched using a *standard user token* might perform tasks using access rights granted to a standard user. For instance, Windows Explorer automatically inherits standard user level permissions. Any applications that are started using Windows Explorer (for example, by opening a shortcut) also run with the standard set of user permissions. Most applications, including the ones included with the operating system, are designed to work properly this way.

Other applications, like ones that aren't designed with security settings in mind, might require more permissions to run successfully. These applications are referred to as *legacy apps*.

When a user tries to perform an action that requires administrative privileges, UAC triggers a *consent prompt*. The prompt notifies the user that a change is about to occur, asking for their permission to proceed:

- If the user approves the change, the action is performed with the highest available privilege
- If the user doesn't approve the change, the action isn't performed and the application that requested the change is prevented from running



When an app requires to run with more than standard user rights, UAC allows users to run apps with their *administrator token* (that is, with administrative rights and permissions) instead of their default, standard user token. Users continue to operate in the standard user security context, while enabling certain apps to run with elevated privileges, if needed.

Windows edition and licensing requirements

The following table lists the Windows editions that support User Account Control (UAC):

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

User Account Control (UAC) license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Next steps

- How User Account Control works
- User Account Control settings and configuration

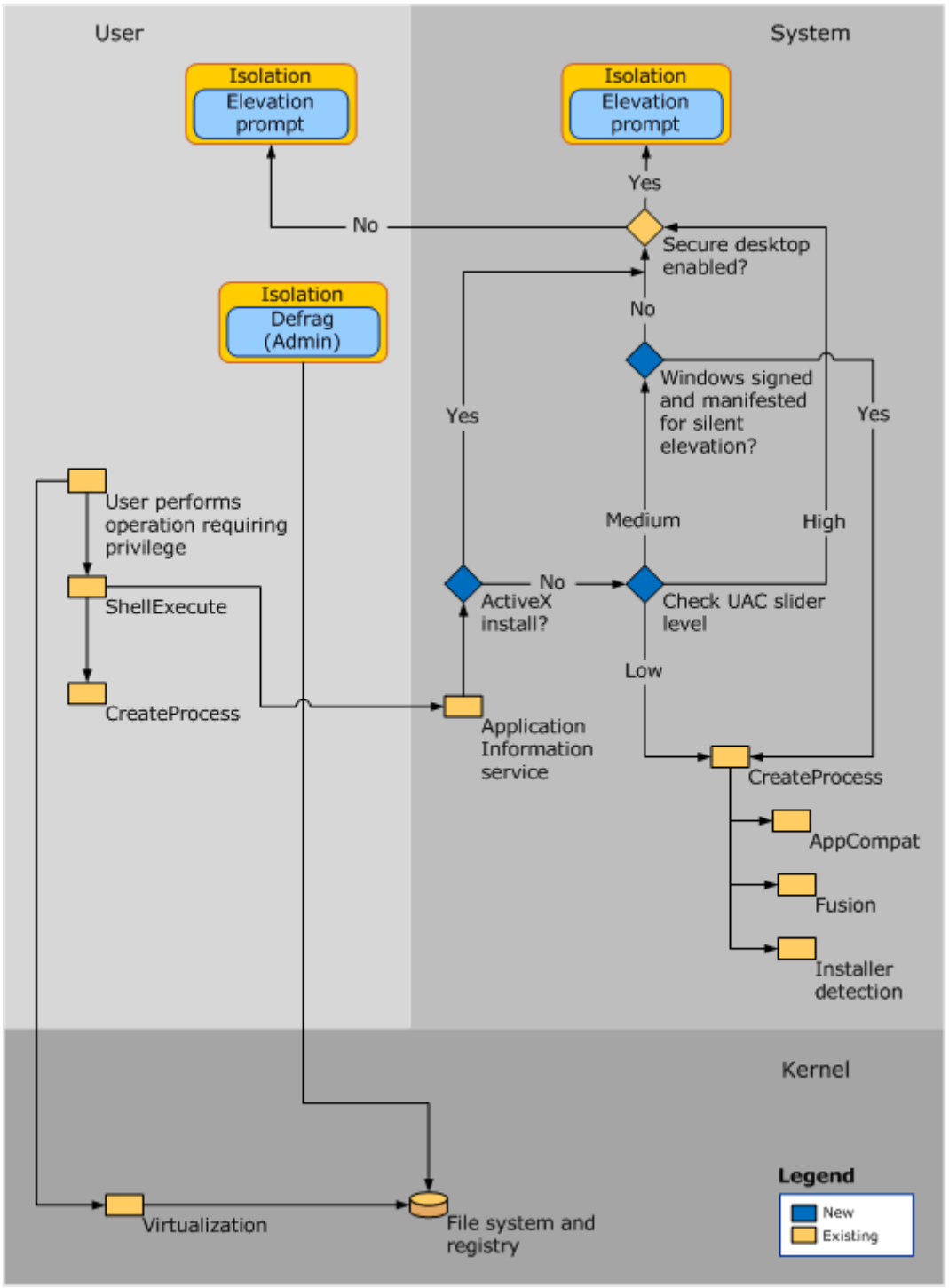
Last updated on 04/15/2025

UAC Architecture

Article • 04/15/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The following diagram details the UAC architecture.



User

- **User performs operation requiring privilege**: If the operation changes the file system or registry, Virtualization is called. All other operations call ShellExecute.

- **ShellExecute**: ShellExecute calls CreateProcess. ShellExecute looks for the `ERROR_ELEVATION_REQUIRED` error from CreateProcess. If it receives the error, ShellExecute calls the Application Information service to attempt to perform the requested task with the elevated prompt.
- **CreateProcess**: If the application requires elevation, CreateProcess rejects the call with `ERROR_ELEVATION_REQUIRED`.

System

- **Application Information service**:
  - A system service that helps start apps that require one or more elevated privileges or user rights to run.
  - The Application Information service helps start such apps by creating a new process for the application with an administrative user's full access token when elevation is required.
  - Depending on the configured policies, the user might give consent.

- **Elevating an ActiveX install**:
  - If ActiveX isn't installed, the system checks the UAC slider level.
  - If ActiveX is installed, the **User Account Control: Switch to the secure desktop when prompting for elevation** Group Policy setting is checked.

- **Check UAC slider level**: UAC has a slider to select from four levels of notification:
  - **Always notify** will:
    - Notify you when programs try to install software or make changes to your computer.
    - Notify you when you make changes to Windows settings.
    - Freeze other tasks until you respond.
    - Recommended if you often install new software or visit unfamiliar websites.
  - **Notify me only when programs try to make changes to my computer** will:
    - Notify you when programs try to install software or make changes to your computer.
    - Not notify you when you make changes to Windows settings.
    - Freeze other tasks until you respond.
    - Recommended if you don't often install apps or visit unfamiliar websites.
  - **Notify me only when programs try to make changes to my computer (do not dim my desktop)** will:
    - Notify you when programs try to install software or make changes to your computer.
    - Not notify you when you make changes to Windows settings.
    - Not freeze other tasks until you respond.

    - Not recommended. Choose this option only if it takes a long time to dim the desktop on your computer.
  - **Never notify (Disable UAC prompts)** will:
    - Not notify you when programs try to install software or make changes to your computer.
    - Not notify you when you make changes to Windows settings.
    - Not freeze other tasks until you respond.
    - Not recommended due to security concerns.

- **Secure desktop enabled**: The **User Account Control: Switch to the secure desktop when prompting for elevation** policy setting is checked:
  - If the secure desktop is enabled, all elevation requests go to the secure desktop regardless of prompt behavior policy settings for administrators and standard users.
  - If the secure desktop isn't enabled, all elevation requests go to the interactive user's desktop, and the per-user settings for administrators and standard users are used.

- **CreateProcess**:
  - CreateProcess calls AppCompat, Fusion, and Installer detection to assess if the app requires elevation.
  - The file is then inspected to determine its requested execution level, which is stored in the application manifest for the file.
  - CreateProcess fails if the requested execution level specified in the manifest doesn't match the access token and returns an error (ERROR_ELEVATION_REQUIRED) to ShellExecute.

- **AppCompat**:
  - The AppCompat database stores information in the application compatibility fix entries for an application.

- **Fusion**:
  - The Fusion database stores information from application manifests that describe the applications.
  - The manifest schema is updated to add a new requested execution level field.

- **Installer detection**:
  - Installer detection detects setup files and helps prevent installations from being run without the user's knowledge and consent.

Kernel

- **Virtualization**: Virtualization technology ensures that noncompliant apps don't silently fail to run or fail in a way that the cause can't be determined. UAC also provides file and

registry virtualization and logging for applications that write to protected areas.
- **File system and registry**: The per-user file and registry virtualization redirects per-computer registry and file write requests to equivalent per-user locations. Read requests are redirected to the virtualized per-user location first and to the per-computer location second.

The slider never turns off UAC completely. If you set it to **Never notify**, it will:

- Keep the UAC service running
- Cause all elevation request initiated by administrators to be autoapproved without showing a UAC prompt
- Automatically deny all elevation requests for standard users

> ⓘ **Important**
>
> In order to fully disable UAC, you must disable the policy **User Account Control: Run all administrators in Admin Approval Mode**. Some Universal Windows Platform apps might not work when UAC is disabled.

Virtualization

Because system administrators in enterprise environments attempt to secure systems, many line-of-business (LOB) applications are designed to use only a standard user access token. As a result, you don't need to replace most apps when UAC is turned on.

Windows includes file and registry virtualization technology for apps that aren't UAC-compliant and that requires an administrator's access token to run correctly. When an administrative app that isn't UAC-compliant attempts to write to a protected folder, such as *Program Files*, UAC gives the app its own virtualized view of the resource it's attempting to change. The virtualized copy is maintained in the user's profile. This strategy creates a separate copy of the virtualized file for each user that runs the noncompliant app.

Most app tasks operate properly by using virtualization features. Although virtualization allows most applications to run, it's a short-term fix and not a long-term solution. App developers should modify their apps to be compliant as soon as possible, rather than relying on file, folder, and registry virtualization.

Virtualization isn't an option in the following scenarios:

- Virtualization doesn't apply to apps that are elevated and run with a full administrative access token

- Virtualization supports only 32-bit apps. Nonelevated 64-bit apps receive an access denied message when they attempt to acquire a handle (a unique identifier) to a Windows object. Native Windows 64-bit apps are required to be compatible with UAC and to write data into the correct locations
- Virtualization is disabled if the app includes an app manifest with a requested execution level attribute

Request execution levels

An app manifest is an XML file that describes and identifies the shared and private side-by-side assemblies that an app should bind to at run time. The app manifest includes entries for UAC app compatibility purposes. Administrative apps that include an entry in the app manifest prompt the user for permission to access the user's access token. Although they lack an entry in the app manifest, most administrative app can run without modification by using app compatibility fixes. App compatibility fixes are database entries that enable applications that aren't UAC-compliant to work properly.

All UAC-compliant apps should have a requested execution level added to the application manifest. If the application requires administrative access to the system, marking the app with a requested execution level of *require administrator* ensures that the system identifies this program as an administrative app, and performs the necessary elevation steps. Requested execution levels specify the privileges required for an app.

Installer detection technology

Installation programs are apps designed to deploy software. Most installation programs write to system directories and registry keys. These protected system locations are typically writeable only by an administrator in Installer detection technology, which means that standard users don't have sufficient access to install programs. Windows heuristically detects installation programs and requests administrator credentials or approval from the administrator user in order to run with access privileges. Windows also heuristically detects updates and programs that uninstall applications. One of the design goals of UAC is to prevent installations from being run without the user's knowledge and consent because installation programs write to protected areas of the file system and registry.

Installer detection only applies to:

- 32-bit executable files
- Applications without a requested execution level attribute
- Interactive processes running as a standard user with UAC enabled

Before a 32-bit process is created, the following attributes are checked to determine whether it's an installer:

- File name includes keywords such as "install," "setup," or "update."
- Versioning Resource fields contain the following keywords: Vendor, Company Name, Product Name, File Description, Original Filename, Internal Name, and Export Name.
- Keywords in the side-by-side manifest are embedded in the executable file.
- Keywords in specific StringTable entries are linked in the executable file.
- Key attributes in the resource script data are linked in the executable file.
- Executable file contains targeted sequences of bytes.

> **Note**
>
> The keywords and sequences of bytes were derived from common characteristics observed from various installer technologies.

> **Note**
>
> The *User Account Control: Detect application installations and prompt for elevation* policy must be enabled for installer detection to detect installation programs. For more information, see **User Account Control settings list**.

How User Account Control works

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

User Account Control (UAC) is a key part of Windows security. UAC reduces the risk of malware by limiting the ability of malicious code to execute with administrator privileges. This article describes how UAC works and how it interacts with the end-users.

UAC process and interactions

With UAC, each application that requires the *administrator access token* must prompt the end user for consent. The only exception is the relationship that exists between parent and child processes. Child processes inherit the user's access token from the parent process. Both the parent and child processes, however, must have the same *integrity level*.

Windows protects processes by marking their integrity levels. Integrity levels are measurements of trust:

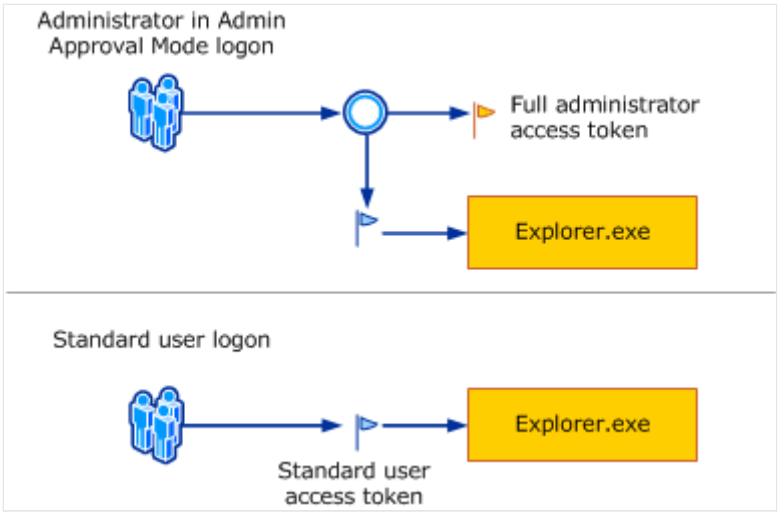
- A *high integrity application* is one that performs tasks that modify system data, such as a disk partitioning application
- A *low integrity application* is one that performs tasks that could potentially compromise the operating system, like as a Web browser

Applications with lower integrity levels can't modify data in applications with higher integrity levels. When a standard user attempts to run an app that requires an administrator access token, UAC requires that the user provides valid administrator credentials.

To better understand how this process works, let's take a closer look at the Windows sign in process.

Sign in process

The following diagram shows how the sign in process for an administrator differs from the sign in process for a standard user.



By default, both standard and administrator users access resources and execute apps in the security context of a standard user.
When a user signs in, the system creates an access token for that user. The access token contains information about the level of access that the user is granted, including specific security identifiers (SIDs) and Windows privileges.

When an administrator logs on, two separate access tokens are created for the user: a *standard user access token* and an *administrator access token*. The standard user access token:

- Contains the same user-specific information as the administrator access token, but the administrative Windows privileges and SIDs are removed
- Is used to start applications that don't perform administrative tasks (standard user apps)
- Is used to display the desktop by executing the process *explorer.exe*. Explorer.exe is the parent process from which all other user-initiated processes inherit their access token. As a result, all apps run as a standard user unless a user provides consent or credentials to approve an app to use a full administrative access token

A user that is a member of the Administrators group can sign in, browse the Web, and read e-mail while using a standard user access token. When the administrator needs to perform a task that requires the administrator access token, Windows automatically prompts the user for approval. This prompt is called an *elevation prompt*, and its behavior can be configured via policy or registry.

The UAC user experience

When UAC is enabled, the user experience for standard users is different from administrator users. The recommended and more secure method of running Windows, is to ensure your primary user account is a standard user. Running as a standard user helps to maximize security for a managed environment. With the built-in UAC elevation component, standard users can easily perform an administrative task by entering valid credentials for a local administrator account.

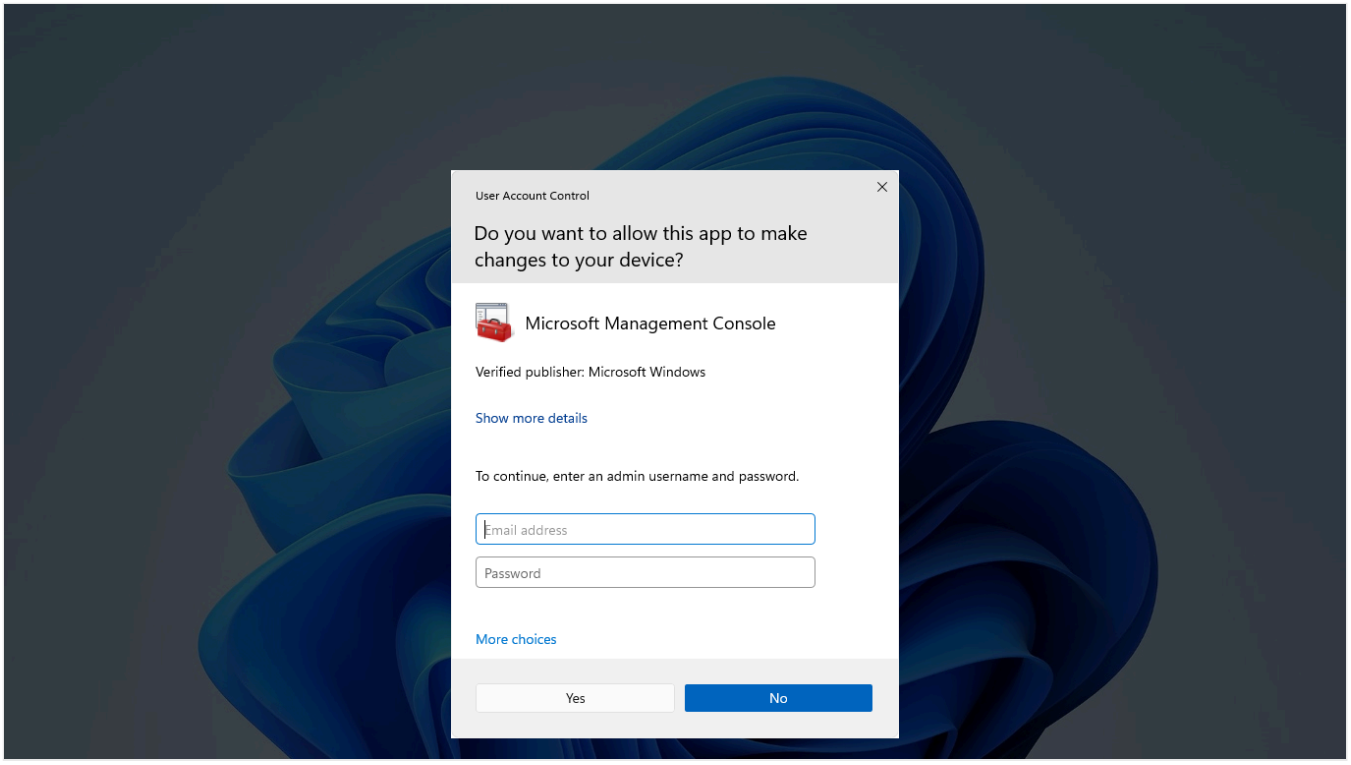
The default, built-in UAC elevation component for standard users is the *credential prompt*.

The alternative to running as a standard user is to run as an administrator in *Admin Approval Mode*. With the built-in UAC elevation component, members of the local Administrators group can easily perform an administrative task by providing approval.

The default, built-in UAC elevation component for an administrator account in Admin Approval Mode is called the *consent prompt*.

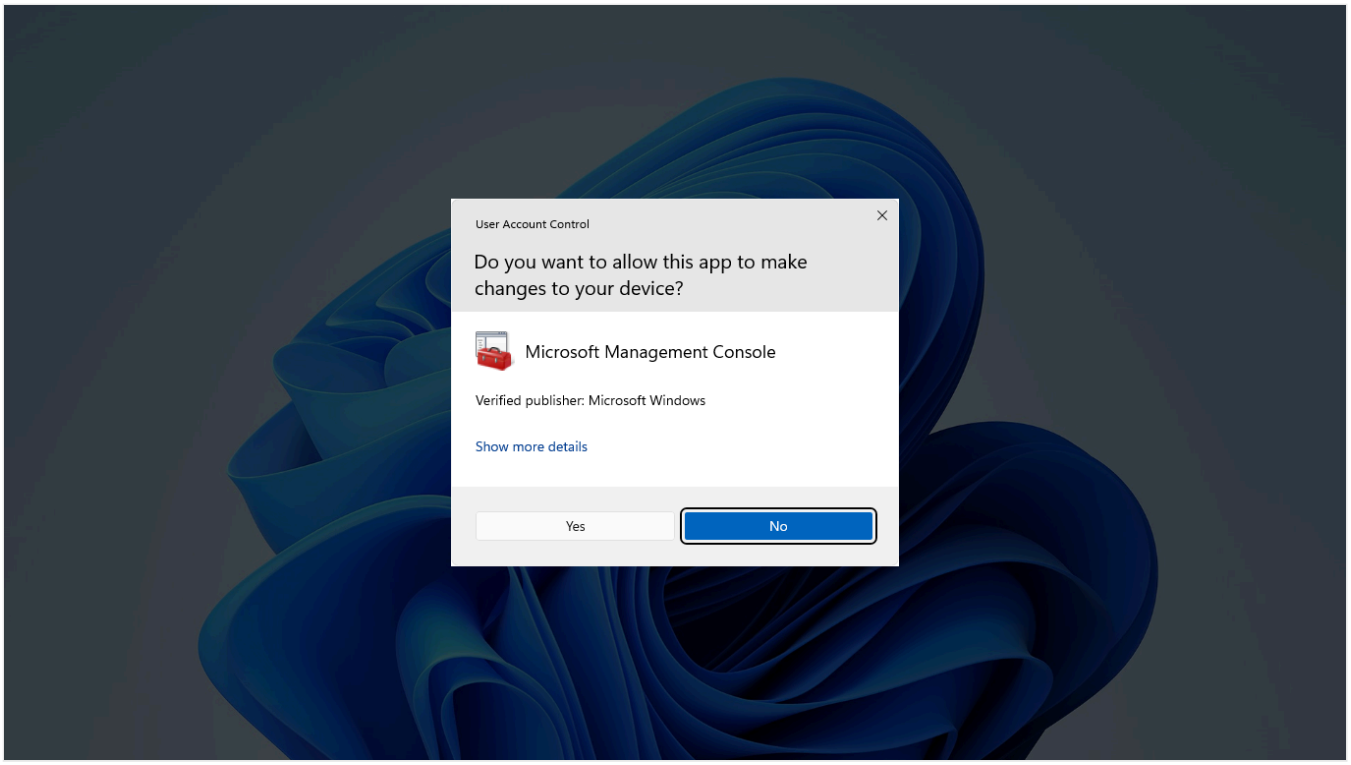
The credential prompt

The credential prompt is presented when a standard user attempts to perform a task that requires a user's administrative access token. Administrators can also be required to provide their credentials by setting the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting value to **Prompt for credentials**.



The consent prompt

The consent prompt is presented when a user attempts to perform a task that requires a user's administrative access token.



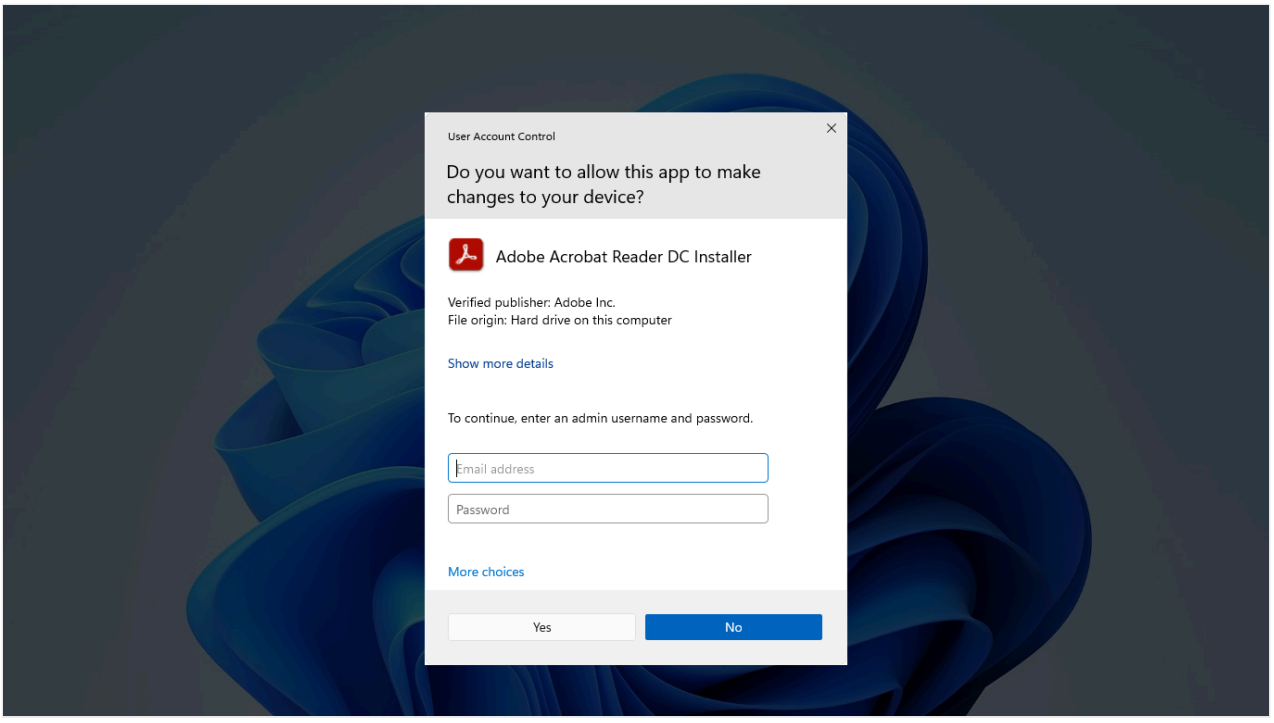
UAC elevation prompts

The UAC elevation prompts are color-coded to be app-specific, enabling for easier identification of an application's potential security risk. When an app attempts to run with an administrator's full access token, Windows first analyzes the executable file to determine its publisher. Apps are first separated into three categories based on the file's publisher:

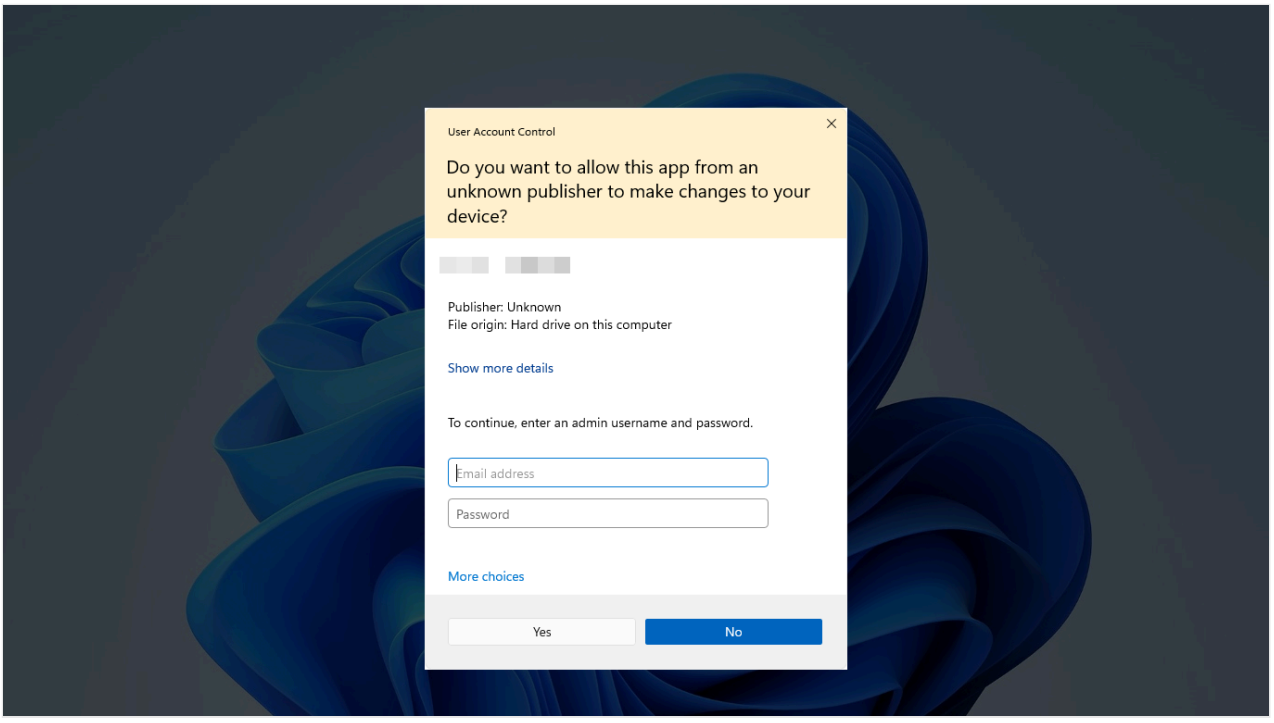
- Windows
- Publisher verified (signed)
- Publisher not verified (unsigned)

The elevation prompt color-coding is as follows:

- Gray background: The application is a Windows administrative app, such as a Control Panel item, or an application signed by a verified publisher

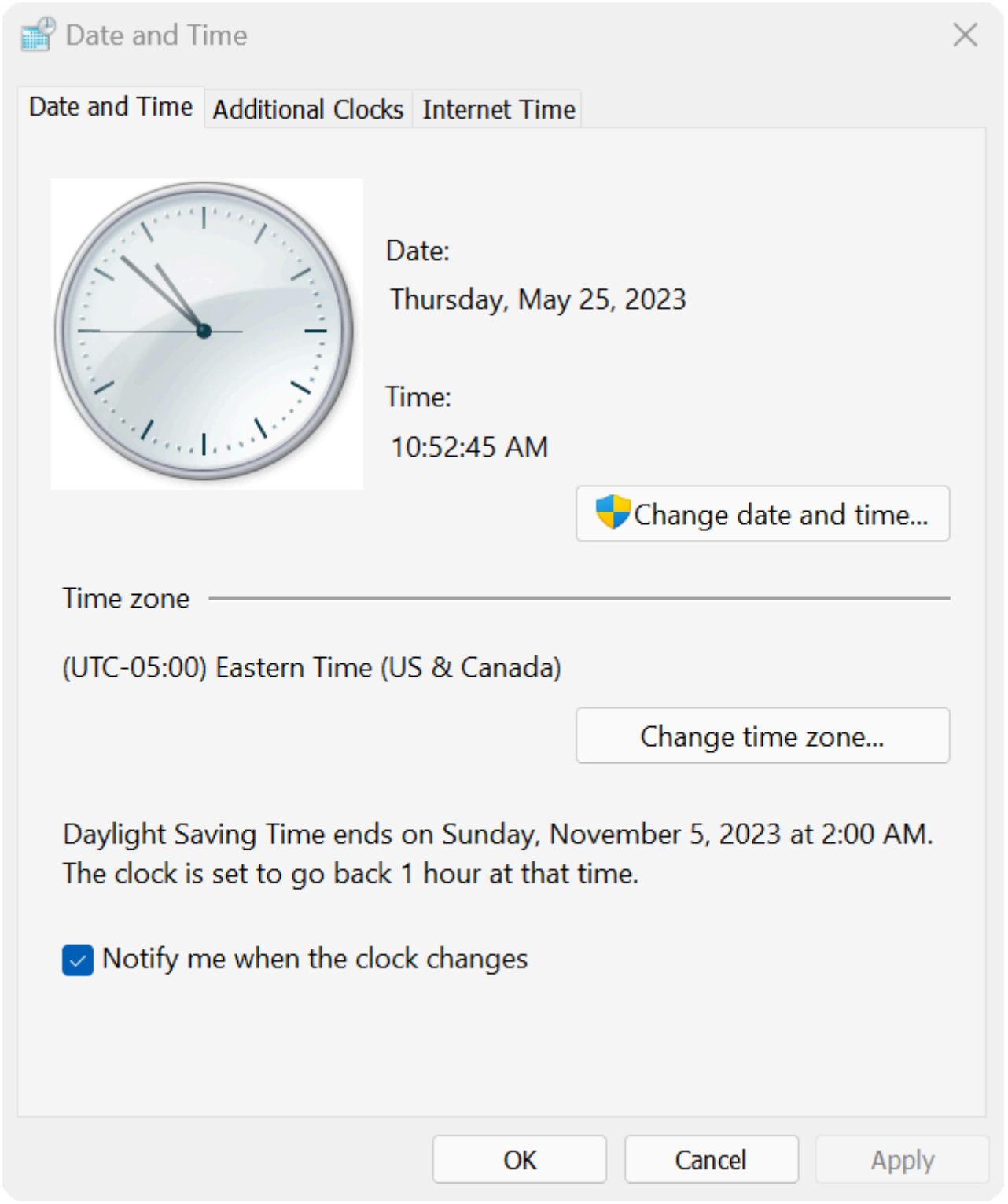


- Yellow background: the application is unsigned or signed but isn't trusted



Shield icon

Some Control Panel items, such as **Date and Time**, contain a combination of administrator and standard user operations. Standard users can view the clock and change the time zone, but a full administrator access token is required to change the local system time. The following is a screenshot of the **Date and Time** Control Panel item.



The shield icon on the **Change date and time...** button indicates that the process requires a full administrator access token.

Securing the elevation prompt

The elevation process is further secured by directing the prompt to the *secure desktop*. The consent and credential prompts are displayed on the secure desktop by default. Only Windows processes can access the secure desktop. For higher levels of security, we recommend keeping the **User Account Control: Switch to the secure desktop when prompting for elevation** policy setting enabled.

When an executable file requests elevation, the *interactive desktop*, also called the *user desktop*, is switched to the secure desktop. The secure desktop dims the user desktop and displays an elevation prompt that must be responded to before continuing. When the user selects **Yes** or **No**, the desktop switches back to the user desktop.

> **Note**
>
> Starting in **Windows Server 2019**, it's not possible to paste the content of the clipboard on the secure desktop. This behavior is the same as the currently supported Windows client OS versions.

Malware can present an imitation of the secure desktop, but when the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting is set to **Prompt for consent**, the malware doesn't gain elevation if the user selects **Yes** on the imitation. If the policy setting is set to **Prompt for credentials**, malware imitating the credential prompt might be able to gather the credentials from the user. However, the malware doesn't gain elevated privilege and the system has other protections that mitigate malware from taking control of the user interface even with a harvested password.

While malware could present an imitation of the secure desktop, this issue can't occur unless a user previously installed the malware on the PC. Because processes requiring an administrator access token can't silently install when UAC is enabled, the user must explicitly provide consent by selecting **Yes** or by providing administrator credentials. The specific behavior of the UAC elevation prompt is dependent upon security policies.

Last updated on 04/15/2025

User Account Control settings and configuration

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

User Account Control settings list

The following list shows the available settings to configure the UAC behavior, and their default values.

- **Admin Approval Mode for the built-in Administrator account**: Controls the behavior of Admin Approval Mode for the built-in Administrator account.
  - **Enabled**: The built-in Administrator account uses Admin Approval Mode. By default, any operation that requires elevation of privilege prompts the user to prove the operation.
  - **Disabled (default)**: The built-in Administrator account runs all applications with full administrative privilege.

- **Allow UIAccess applications to prompt for elevation without using the secure desktop**: Controls whether User Interface Accessibility (UIAccess or UIA) programs can automatically disable the secure desktop for elevation prompts used by a standard user.
  - **Enabled**: UIA programs, including Remote Assistance, automatically disable the secure desktop for elevation prompts. If you don't disable the **Switch to the secure desktop when prompting for elevation** policy setting, the prompts appear on the interactive user's desktop instead of the secure desktop. This setting allows the remote administrator to provide the appropriate credentials for elevation. This policy setting doesn't change the behavior of the UAC elevation prompt for administrators. If you plan to enable this policy setting, you should also review the effect of the **Behavior of the elevation prompt for standard users** policy setting; if configured as **Automatically deny elevation requests**, elevation requests aren't presented to the user.
  - **Disabled (default)**: The secure desktop can be disabled only by the user of the interactive desktop or by disabling the **Switch to the secure desktop when prompting for elevation** policy setting.

- **Behavior of the elevation prompt for administrators in Admin Approval Mode**: Controls the behavior of the elevation prompt for administrators.
  - **Elevate without prompting**: Allows privileged accounts to perform an operation that requires elevation without requiring consent or credentials. **Use this option only in the most constrained environments**.

    - **Prompt for credentials on the secure desktop**: When an operation requires elevation of privilege, the user is prompted on the secure desktop to enter a privileged user name and password. If the user enters valid credentials, the operation continues with the user's highest available privilege.
    - **Prompt for consent on the secure desktop**: When an operation requires elevation of privilege, the user is prompted on the secure desktop to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.
    - **Prompt for credentials**: When an operation requires elevation of privilege, the user is prompted to enter an administrative user name and password. If the user enters valid credentials, the operation continues with the applicable privilege.
    - **Prompt for consent**: When an operation requires elevation of privilege, the user is prompted to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.
    - **Prompt for consent for non-Windows binaries (default)**: When an operation for a non-Microsoft application requires elevation of privilege, the user is prompted on the secure desktop to select either Permit or Deny. If the user selects Permit, the operation continues with the user's highest available privilege.

- **Behavior of the elevation prompt for standard users**: Controls the behavior of the elevation prompt for standard users.
    - **Prompt for credentials (default)**: When an operation requires elevation of privilege, the user is prompted to enter an administrative user name and password. If the user enters valid credentials, the operation continues with the applicable privilege.
    - **Automatically deny elevation requests**: When an operation requires elevation of privilege, a configurable access denied error message is displayed. An enterprise that is running desktops as standard user might choose this setting to reduce help desk calls.
    - **Prompt for credentials on the secure desktop**: When an operation requires elevation of privilege, the user is prompted on the secure desktop to enter a different user name and password. If the user enters valid credentials, the operation continues with the applicable privilege.

- **Detect application installations and prompt for elevation**: Controls the behavior of application installation detection for the computer.
    - **Enabled (default)**: When an app installation package is detected that requires elevation of privilege, the user is prompted to enter an administrative user name and password. If the user enters valid credentials, the operation continues with the applicable privilege.
    - **Disabled**: App installation packages aren't detected and prompted for elevation. Enterprises that are running standard user desktops and use delegated installation

technologies, such as Microsoft Intune, should disable this policy setting. In this case, installer detection is unnecessary.

- **Only elevate executables that are signed and validated**: Enforces signature checks for any interactive applications that request elevation of privilege. IT admins can control which applications are allowed to run by adding certificates to the Trusted Publishers certificate store on local devices.
  - **Enabled**: Enforces the certificate certification path validation for a given executable file before it's permitted to run.
  - **Disabled (default)**: Doesn't enforce the certificate certification path validation before a given executable file is permitted to run.

- **Only elevate UIAccess applications that are installed in secure locations**: Controls whether applications that request to run with a User Interface Accessibility (UIAccess) integrity level must reside in a secure location in the file system.
  - **Enabled (default)**: If an app resides in a secure location in the file system, it runs only with UIAccess integrity.
  - **Disabled**: An app runs with UIAccess integrity even if it doesn't reside in a secure location in the file system.

  Secure locations are limited to the following folders:
  - `%ProgramFiles%`, including subfolders
  - `%SystemRoot%\system32\`
  - `%ProgramFiles(x86)%`, including subfolders

  > ⓘ **Note**
  >
  > Windows enforces a digital signature check on any interactive apps that request to run with a UIAccess integrity level regardless of the state of this setting.

- **Run all administrators in Admin Approval Mode**: Controls the behavior of all UAC policy settings.
  - **Enabled (default)**: Admin Approval Mode is enabled. This policy must be enabled and related UAC settings configured. The policy allows the built-in Administrator account and members of the Administrators group to run in Admin Approval Mode.
  - **Disabled**: Admin Approval Mode and all related UAC policy settings are disabled. **Note:** If this policy setting is disabled, **Windows Security** notifies you that the overall security of the operating system is reduced.

- **Switch to the secure desktop when prompting for elevation**: This policy setting controls whether the elevation request prompt is displayed on the interactive user's desktop or the secure desktop.

    - **Enabled (default)**: All elevation requests go to the secure desktop regardless of prompt behavior policy settings for administrators and standard users.
    - **Disabled**: All elevation requests go to the interactive user's desktop. Prompt behavior policy settings for administrators and standard users are used.

- **Virtualize File And Registry Write Failures To Per User Locations**: Controls whether application write failures are redirected to defined registry and file system locations. This setting mitigates applications that run as administrator and write run-time application data to `%ProgramFiles%`, `%Windir%`, `%Windir%\system32`, or `HKLM\Software`.
    - **Enabled (default)**: App write failures are redirected at run time to defined user locations for both the file system and registry.
    - **Disabled**: Apps that write data to protected locations fail.

User Account Control configuration

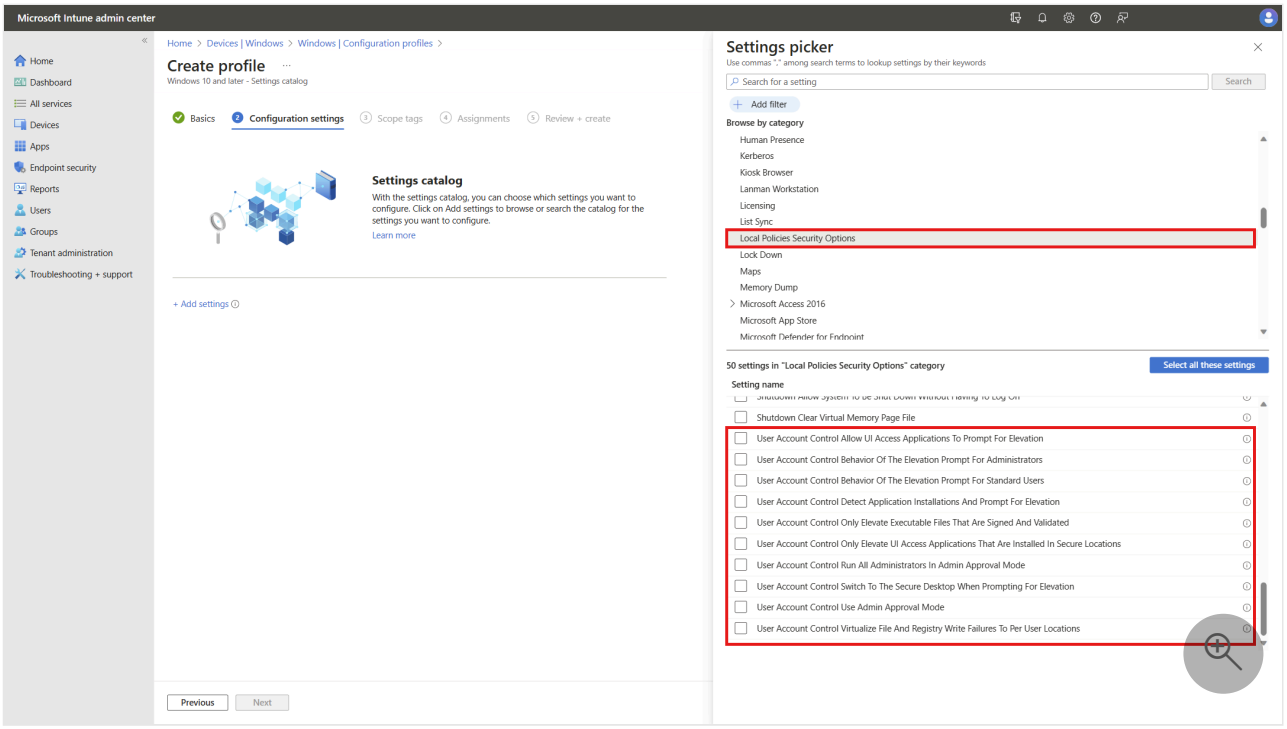
To configure UAC, you can use:

- Microsoft Intune
- CSP
- Group policy
- Registry

The following instructions provide details how to configure your devices. Select the option that best suits your needs.

Intune

To configure devices using Microsoft Intune, create a **Settings catalog** policy, and use the settings listed under the category **`Local Policies Security Options`**:



Assign the policy to a security group that contains as members the devices or users that you want to configure.

Last updated on 04/15/2025

Administrator protection (preview)

Applies to: ✅ Windows 11

Administrator protection is a platform security feature in Windows 11 designed to ensure users run with the least privilege needed, elevating to admin rights only when necessary and only with the user's explicit approval. This feature operates on the principle of least privilege, keeping users in a deprivileged state and granting just-in-time elevation rights only when needed.

In today's digital landscape, users with Administrator rights on Windows have powerful capabilities to modify configurations and make system-wide changes that might affect the overall security posture of a Windows 11 device. Administrative privileges create a significant attack vector. Malicious actors often exploit these privileges to gain unauthorized access to user data, compromise privacy, and disable OS security features without the user's knowledge.

Administrator protection addresses this challenge by requiring users to verify their identity with Windows Hello integrated authentication. It enforces this verification before allowing actions that need administrator privileges, such as installing software, changing system settings like the time or registry, and accessing sensitive data.

Benefits

Administrator protection provides several key benefits:

- **Enhanced security**: By requiring explicit authorization from the user for every administrative task, Administrator protection protects Windows from accidental changes by users and changes by malware. It helps ensure that users are aware of potentially harmful actions before they occur, providing an extra layer of defense against threats.
- **User control**: Administrator protection requires users to decide explicitly whether they want a particular application to run as elevated. This helps ensure that only authorized apps can make system changes, reducing the risk of accidental or malicious modifications.
- **Malware reduction**: Malicious software often relies on admin privileges to change device settings and execute harmful actions. Administrator protection breaks the attack kill chain since malware can no longer silently acquire admin privileges.

System requirements

Administrator protection will be available on Windows 11 devices soon. The feature previously listed in the October 2025 non-security update (KB5067036) has been reverted and will roll out at a later date.

How Administrator protection works

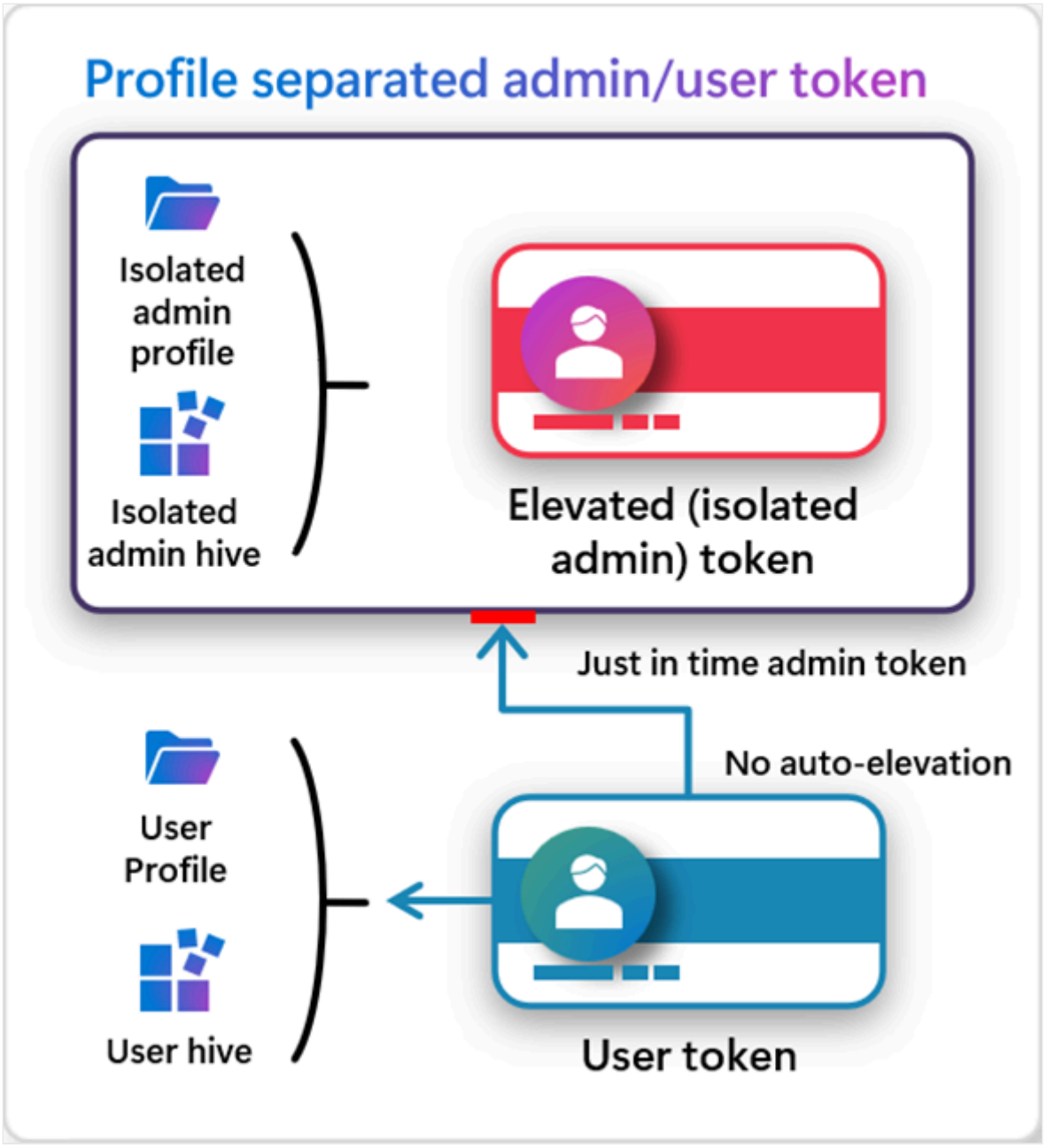
At its core, Administrator protection operates on the principle of least privilege. When a user signs in to Windows, they receive a deprivileged user token. However, when admin privileges are needed, Windows requests that the user authorize the operation. Once authorized, Windows uses a hidden, system-generated, profile-separated user account to create an isolated admin token. This token is issued to the requesting process and is destroyed once the process ends, ensuring that admin privileges don't persist.

Administrator protection introduces a new security boundary with support to fix any reported security bugs. The architectural changes ensure that no one can access or tamper with the code or data of elevated sessions without proper authorization.

Some applications may be relying on admin rights always being present, and the elevated profile being accessible when running unelevated. With this new approach, some scenarios in those apps might need updates to work smoothly with the enhanced security model. We're actively collaborating with app developers to help them adapt, ensuring your favorite experiences remain seamless while keeping your system protected. See Enhance your application security with administrator protection for guidance on developing your applications.

Ultimately, these changes are all about making Windows more secure for you, so you can enjoy powerful features with greater peace of mind.

Key architectural highlights



- **Just-in-time elevation**: Users stay deprivileged and are granted just-in-time elevation rights only during an admin operation. The admin token is discarded after use and recreated when another task requiring admin privileges is performed.
- **Profile separation**: Administrator protection uses hidden, system-generated, profile-separated user accounts to create isolated admin tokens. This helps ensure that user-level malware can't compromise the elevated session, making elevation a security boundary.
- **No auto-elevations**: Users need to interactively authorize every admin operation. This ensures that the administrator user stays in full control and that admin privileges aren't abused.
- **Windows Hello integration**: Administrator protection is integrated with Windows Hello for simple and secure authorization.

Configuration

Administrator protection can be enabled through several methods:

- Microsoft Intune Settings Catalog (Preview)
- CSP
- Group policy
- Windows Security Settings (Preview)

Intune

> **Note**
>
> - This option is currently available in preview. It will be rolled out to everyone gradually.
> - Administrator protection can be configured with Configuration Service Provider (CSP) policies using Intune. Use custom policy to configure:
>   - UserAccountControl_TypeOfAdminApprovalMode [enable]
>   - UserAccountControl_BehaviorOfTheElevationPromptForAdministratorProtection [configure prompt]

To configure devices using Microsoft Intune, create a **Settings catalog** policy, and use the settings listed under the category `Local Policies Security Options`:

- User Account Control Behavior Of the Elevation Prompt for Administrator Protection
  - User Account Control Type Of Admin Approval Mode

Assign the policy to a security group that contains as members the devices or users that you want to configure.

> **Important**
>
> A reboot is required for Administrator Protection to take effect.

Monitoring and Reporting Events

To track elevations, administrator protection has two new Event Tracing for Windows (ETW) events under the existing **Microsoft-Windows-LUA** provider with GUID `{93c05d69-51a3-485e-877f-1806a8731346}`:

Expand table

Event ID	Event Name	Description
15031	Elevation Approved	Logged when a user successfully authenticates and elevation is granted
15032	Elevation Denied/Fail	Logged when elevation is denied, fails, or times out

What Gets Logged

- The Security Identifier (SID) of the user who triggered the elevation
- The application name and path
- The elevation outcome (approved, denied, timeout)
- The system managed admin account used to perform the task
- The authentication method (e.g., password, PIN, Windows Hello)

How to capture these events

- Enable the Microsoft-Windows-LUA provider (GUID: `{93c05d69-51a3-485e-877f-1806a8731346}`)
- Use Logman or WPR (Windows Performance Recorder) to start a trace session
- Filter for Event IDs `15031` and `15032`
- Analyze the resulting .etl file using Windows Performance Analyzer or your preferred tool

Here's a sample command:

Windows Command Prompt

```
logman start AdminProtectionTrace -p {93c05d69-51a3-485e-877f-1806a8731346} -ets
```

Troubleshooting

- Use local SYSTEM or dedicated service accounts for scheduled tasks or scripts set to run "with highest privileges". Essentially, redesign scripts to avoid expecting an always-on admin token. Any workflow that assumed an admin session is available will need to be adjusted.

- When apps run elevated with administrator protection, Single Sign-On (SSO) credentials from the standard session aren't available to the elevated session. Any domain or cloud authentication must be re-established within that elevated session.
- Network Drives/Resources Inaccessible from elevated apps. Install it in user context to enable network credential prompts. If it's essential to install an application as elevated, copy installation files to a local drive before elevating.
- Settings data for applications don't carry over across the regular(unelevated) and the elevated profiles. Only elevate apps you truly need to. Consider configuring those applications to use common data directories accessible to both profiles (if possible).
- Scope of administrator protection: admin accounts on a device. Remote logon, roaming Profiles or backup admins aren't in scope.
- A few applications don't show the launch icon in the start menu after installation. If you have installed elevated, you need to navigate manually to installation location: `AppData\Roaming\Microsoft\Windows\Start Menu\Programs\<App name>`
- If updating any of your applications is blocked, temporarily disable the feature. (Reboot is needed).
- When not to enable this feature:
  - For devices that require Hyper-V or Windows Subsystem for Linux (WSL).
  - If you're using apps that can’t access Edge extensions or shared files across profiles. For example: Some installers (using WebView2 internally) may request elevated permissions even when launched normally showing “Microsoft Edge can't read and write to its data directory”.

Related articles

- User Account Control (UAC)
- Windows Hello enhanced sign-in security
- Microsoft Intune settings catalog

Last updated on 11/20/2025

Microsoft recommended driver block rules

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Note**
>
> Some capabilities of App Control for Business are only available on specific Windows versions. Learn more about App Control feature availability.

Microsoft has strict requirements for code running in kernel. So, malicious actors are turning to exploit vulnerabilities in legitimate and signed kernel drivers to run malware in kernel. One of the many strengths of the Windows platform is our strong collaboration with independent hardware vendors (IHVs) and OEMs. Microsoft works closely with our IHVs and security community to ensure the highest level of driver security for our customers. When vulnerabilities in drivers are found, we work with our partners to ensure they're quickly patched and rolled out to the ecosystem. The vulnerable driver blocklist is designed to help harden systems against non-Microsoft-developed drivers across the Windows ecosystem with any of the following attributes:

- Known security vulnerabilities that an attacker could exploit to elevate privileges in the Windows kernel
- Malicious behaviors (malware) or certificates used to sign malware
- Behaviors that aren't malicious but circumvent the Windows Security Model and an attacker could exploit to elevate privileges in the Windows kernel

Drivers can be submitted to Microsoft for security analysis at the Microsoft Security Intelligence Driver Submission page. For more information about driver submission, see Improve kernel security with the new Microsoft Vulnerable and Malicious Driver Reporting Center. To report an issue or request a change to the blocklist, including updating a block rule once a fixed version of a driver is available, visit the Microsoft Security Intelligence portal.

> **Note**
>
> Blocking drivers can cause devices or software to malfunction, and in rare cases, lead to blue screen. The vulnerable driver blocklist isn't guaranteed to block every driver found to have vulnerabilities. When we produce the blocklist, Microsoft attempts to balance the security risks from vulnerable drivers with the potential effect on compatibility and reliability. The blocklist included in this article and in the associated downloadable files usually contains a more complete set of known vulnerable drivers than the version in the OS and delivered by Windows Update. It's often necessary for us to hold back some

blocks to avoid breaking existing functionality while we work with our partners who are engaging their users to update to patched versions. As always, Microsoft recommends using an explicit allowlist approach to security wherever possible, but when that isn't feasible, the use of this blocklist is a critical tool to disrupt malicious actors.

Microsoft vulnerable driver blocklist

Since the Windows 11 2022 update, the vulnerable driver blocklist is enabled by default for all devices, and can be turned on or off via the Windows Security app. Except on Windows Server 2016, the vulnerable driver blocklist is also enforced when either memory integrity, also known as hypervisor-protected code integrity (HVCI), Smart App Control, or S mode is active. Users can opt in to HVCI using the Windows Security app, and HVCI is on by-default for most new Windows 11 devices.

The blocklist is updated with each new major release of Windows, typically 1-2 times per year. The most current blocklist is now also available as an optional update from Windows Update. Microsoft will occasionally publish future updates through regular Windows servicing.

Customers who always want the most up-to-date driver blocklist can also use App Control for Business to apply the latest recommended driver blocklist included in this article. For your convenience, we provide a download of the most up-to-date vulnerable driver blocklist along with instructions to apply it on your computer at the end of this article.

Blocking vulnerable drivers using App Control

Microsoft recommends enabling HVCI or S mode to protect your devices against security threats. If this setting isn't possible, Microsoft recommends blocking this list of drivers within your existing App Control for Business policy. Blocking kernel drivers without sufficient testing can cause devices or software to malfunction, and in rare cases, blue screen. You should first validate this policy in audit mode and review the audit block events before deploying an enforced version.

Important

Microsoft also recommends enabling the Attack Surface Reduction (ASR) rule **Block abuse of exploited vulnerable signed drivers** to prevent an application from writing a vulnerable signed driver to disk. The ASR rule doesn't block a driver already existing on the system from loading, however enabling **Microsoft vulnerable driver blocklist** or applying this App Control policy prevents the existing driver from loading.

Steps to download and apply the vulnerable driver blocklist binary

If you prefer to apply the vulnerable driver blocklist, follow these steps:

1. Download the App Control policy refresh tool
2. Download and extract the vulnerable driver blocklist binaries
3. Select either the audit only version or the enforced version and rename the file to SiPolicy.p7b
4. Copy SiPolicy.p7b to %windir%\system32\CodeIntegrity
5. Run the App Control policy refresh tool you downloaded in Step 1 above to activate and refresh all App Control policies on your computer

To check that the policy was successfully applied on your computer:

1. Open Event Viewer
2. Browse to **Applications and Services Logs - Microsoft - Windows - CodeIntegrity - Operational**
3. Select **Filter Current Log...**
4. Replace "<All Event IDs>" with "3099" and select OK.
5. You should find a 3099 event where the PolicyNameBuffer and PolicyIdBuffer match the Name and ID from PolicyInfo settings found in the blocklist App Control Policy XML in this article. NOTE: Your computer might have more than one 3099 event if other App Control policies are present.

> **Note**
>
> If any vulnerable drivers are already running that the policy would block, you must reboot your computer for those drivers to be blocked. Running processes aren't stopped when activating a new App Control policy without reboot.

Vulnerable driver blocklist XML

The recommended blocklist xml policy file can be downloaded from the Microsoft Download Center.

This policy contains **Allow All** rules. If your version of Windows supports App Control multiple policies, we recommend deploying this policy alongside any existing App Control policies. If you do plan to merge this policy with another policy, remove the **Allow All** rules before merging it if the other policy applies an explicit allowlist. For more information, see Create an App Control Deny Policy.

> **Note**
>
> To use this policy with Windows Server 2016, you must convert the policy XML on a device running a newer operating system. The policies available at the Microsoft Download Center link provided earlier in this article also include versions for Windows Server 2016.

More information

- Merge App Control for Business policies

Last updated on 08/11/2025

Microsoft Defender Application Guard overview

Applies to: ✅ Windows 11, ✅ Windows 10

> ⓘ Note
>
> - Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is deprecated for Microsoft Edge for Business and will no longer be updated. To learn more about Microsoft Edge security capabilities, see Microsoft Edge For Business Security.
> - Starting with Windows 11, version 24H2, Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is no longer available.
> - Because Application Guard is deprecated there will not be a migration to Edge Manifest V3. The corresponding browser extensions and associated Windows Store app are no longer available. If you want to block unprotected browsers until you are ready to retire MDAG usage in your enterprise, we recommend using AppLocker policies or Microsoft Edge management service. For more information, see Microsoft Edge and Microsoft Defender Application Guard.

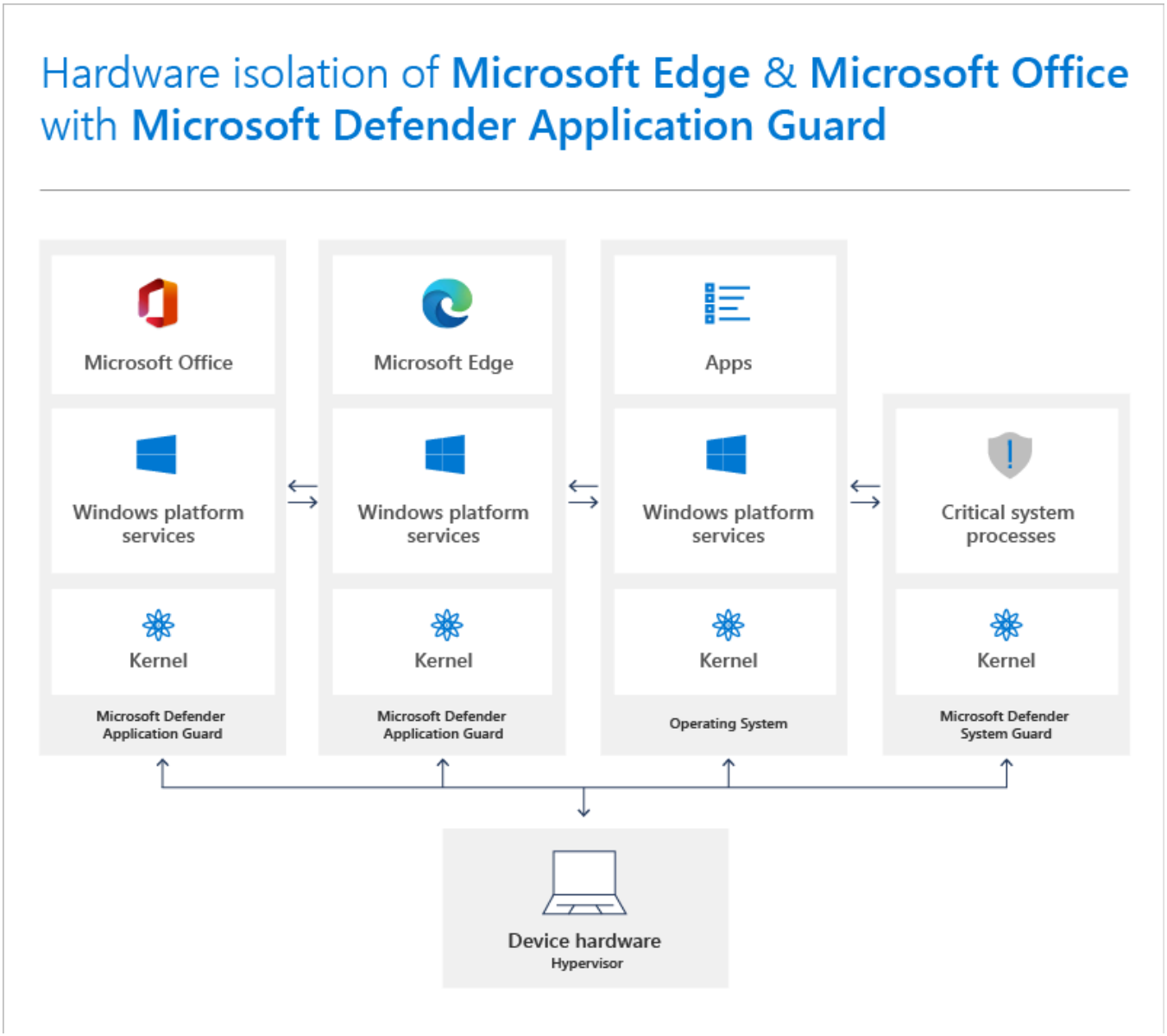
Microsoft Defender Application Guard (MDAG) is designed to help prevent old and newly emerging attacks to help keep employees productive. Using our unique hardware isolation approach, our goal is to destroy the playbook that attackers use by making current attack methods obsolete.

What is Application Guard and how does it work?

For Microsoft Edge, Application Guard helps to isolate enterprise-defined untrusted sites, protecting your company while your employees browse the Internet. As an enterprise administrator, you define what is among trusted web sites, cloud resources, and internal networks. Everything not on your list is considered untrusted. If an employee goes to an untrusted site through either Microsoft Edge or Internet Explorer, Microsoft Edge opens the site in an isolated Hyper-V-enabled container.

For Microsoft Office, Application Guard helps prevents untrusted Word, PowerPoint, and Excel files from accessing trusted resources. Application Guard opens untrusted files in an isolated Hyper-V-enabled container. The isolated Hyper-V container is separate from the host operating system. This container isolation means that if the untrusted site or file turns out to

be malicious, the host device is protected, and the attacker can't get to your enterprise data. For example, this approach makes the isolated container anonymous, so an attacker can't get to your employee's enterprise credentials.



What types of devices should use Application Guard?

Application Guard has been created to target several types of devices:

- **Enterprise desktops**. These desktops are domain-joined and managed by your organization. Configuration management is primarily done through Microsoft Configuration Manager or Microsoft Intune. Employees typically have Standard User privileges and use a high-bandwidth, wired, corporate network.

- **Enterprise mobile laptops**. These laptops are domain-joined and managed by your organization. Configuration management is primarily done through Microsoft Configuration Manager or Microsoft Intune. Employees typically have Standard User privileges and use a high-bandwidth, wireless, corporate network.

- **Bring your own device (BYOD) mobile laptops**. These personally owned laptops aren't domain-joined, but are managed by your organization through tools, such as Microsoft Intune. The employee is typically an admin on the device and uses a high-bandwidth wireless corporate network while at work and a comparable personal network while at home.
- **Personal devices**. These personally owned desktops or mobile laptops aren't domain-joined or managed by an organization. The user is an admin on the device and uses a high-bandwidth wireless personal network while at home or a comparable public network while outside.

Windows edition and licensing requirements

The following table lists the Windows editions that support Microsoft Defender Application Guard (MDAG) for Edge standalone mode:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Microsoft Defender Application Guard (MDAG) for Edge standalone mode license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

For more information about Microsoft Defender Application Guard (MDAG) for Microsoft Edge enterprise mode, Configure Microsoft Defender Application Guard policy settings.

Related articles

Expand table

Article	Description
System requirements for Microsoft Defender Application Guard	Specifies the prerequisites necessary to install and use Application Guard.
Prepare and install Microsoft Defender Application Guard	Provides instructions about determining which mode to use, either Standalone or Enterprise-managed, and how to install Application Guard in your organization.
Configure the Group Policy settings for Microsoft Defender Application Guard	Provides info about the available Group Policy and MDM settings.
Testing scenarios using Microsoft Defender Application Guard in your business or organization	Provides a list of suggested testing scenarios that you can use to test Application Guard in your organization.
Microsoft Defender Application Guard for Microsoft Office	Describes Application Guard for Microsoft Office, including minimum hardware requirements, configuration, and a troubleshooting guide
Frequently asked questions - Microsoft Defender Application Guard	Provides answers to frequently asked questions about Application Guard features, integration with the Windows operating system, and general configuration.
Use a network boundary to add trusted sites on Windows devices in Microsoft Intune	Network boundary, a feature that helps you protect your environment from sites that aren't trusted by your organization.

Last updated on 04/15/2025

Microsoft Defender Application Guard overview

Applies to: ✅ Windows 11, ✅ Windows 10

> ⓘ Note
>
> - Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is deprecated for Microsoft Edge for Business and will no longer be updated. To learn more about Microsoft Edge security capabilities, see Microsoft Edge For Business Security.
> - Starting with Windows 11, version 24H2, Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is no longer available.
> - Because Application Guard is deprecated there will not be a migration to Edge Manifest V3. The corresponding browser extensions and associated Windows Store app are no longer available. If you want to block unprotected browsers until you are ready to retire MDAG usage in your enterprise, we recommend using AppLocker policies or Microsoft Edge management service. For more information, see Microsoft Edge and Microsoft Defender Application Guard.

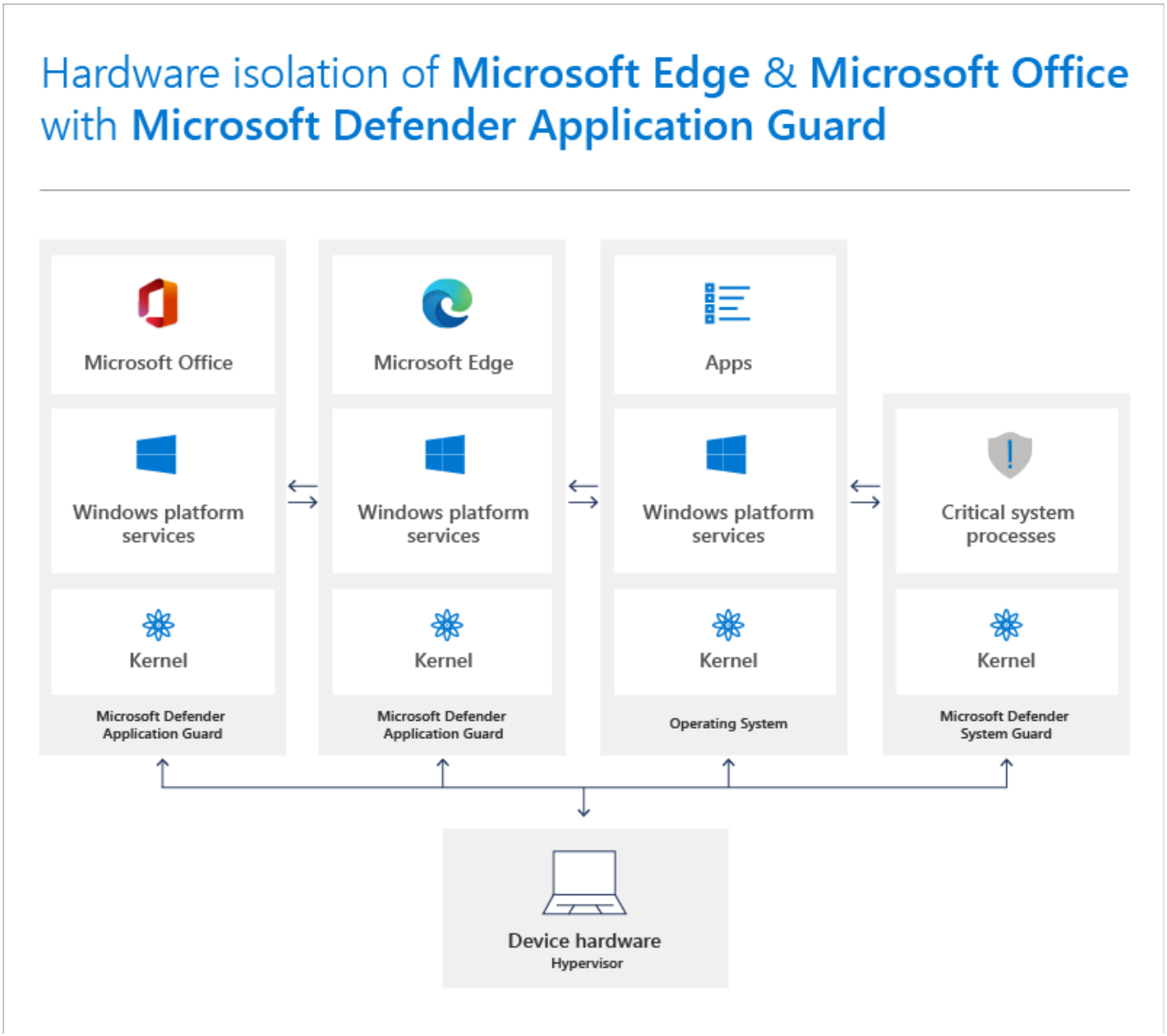
Microsoft Defender Application Guard (MDAG) is designed to help prevent old and newly emerging attacks to help keep employees productive. Using our unique hardware isolation approach, our goal is to destroy the playbook that attackers use by making current attack methods obsolete.

What is Application Guard and how does it work?

For Microsoft Edge, Application Guard helps to isolate enterprise-defined untrusted sites, protecting your company while your employees browse the Internet. As an enterprise administrator, you define what is among trusted web sites, cloud resources, and internal networks. Everything not on your list is considered untrusted. If an employee goes to an untrusted site through either Microsoft Edge or Internet Explorer, Microsoft Edge opens the site in an isolated Hyper-V-enabled container.

For Microsoft Office, Application Guard helps prevents untrusted Word, PowerPoint, and Excel files from accessing trusted resources. Application Guard opens untrusted files in an isolated Hyper-V-enabled container. The isolated Hyper-V container is separate from the host operating system. This container isolation means that if the untrusted site or file turns out to

be malicious, the host device is protected, and the attacker can't get to your enterprise data. For example, this approach makes the isolated container anonymous, so an attacker can't get to your employee's enterprise credentials.



What types of devices should use Application Guard?

Application Guard has been created to target several types of devices:

- **Enterprise desktops**. These desktops are domain-joined and managed by your organization. Configuration management is primarily done through Microsoft Configuration Manager or Microsoft Intune. Employees typically have Standard User privileges and use a high-bandwidth, wired, corporate network.

- **Enterprise mobile laptops**. These laptops are domain-joined and managed by your organization. Configuration management is primarily done through Microsoft Configuration Manager or Microsoft Intune. Employees typically have Standard User privileges and use a high-bandwidth, wireless, corporate network.

- **Bring your own device (BYOD) mobile laptops**. These personally owned laptops aren't domain-joined, but are managed by your organization through tools, such as Microsoft Intune. The employee is typically an admin on the device and uses a high-bandwidth wireless corporate network while at work and a comparable personal network while at home.
- **Personal devices**. These personally owned desktops or mobile laptops aren't domain-joined or managed by an organization. The user is an admin on the device and uses a high-bandwidth wireless personal network while at home or a comparable public network while outside.

Windows edition and licensing requirements

The following table lists the Windows editions that support Microsoft Defender Application Guard (MDAG) for Edge standalone mode:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Microsoft Defender Application Guard (MDAG) for Edge standalone mode license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

For more information about Microsoft Defender Application Guard (MDAG) for Microsoft Edge enterprise mode, Configure Microsoft Defender Application Guard policy settings.

Related articles

Expand table

Article	Description
System requirements for Microsoft Defender Application Guard	Specifies the prerequisites necessary to install and use Application Guard.
Prepare and install Microsoft Defender Application Guard	Provides instructions about determining which mode to use, either Standalone or Enterprise-managed, and how to install Application Guard in your organization.
Configure the Group Policy settings for Microsoft Defender Application Guard	Provides info about the available Group Policy and MDM settings.
Testing scenarios using Microsoft Defender Application Guard in your business or organization	Provides a list of suggested testing scenarios that you can use to test Application Guard in your organization.
Microsoft Defender Application Guard for Microsoft Office	Describes Application Guard for Microsoft Office, including minimum hardware requirements, configuration, and a troubleshooting guide
Frequently asked questions - Microsoft Defender Application Guard	Provides answers to frequently asked questions about Application Guard features, integration with the Windows operating system, and general configuration.
Use a network boundary to add trusted sites on Windows devices in Microsoft Intune	Network boundary, a feature that helps you protect your environment from sites that aren't trusted by your organization.

Last updated on 04/15/2025

Microsoft Edge support for Microsoft Defender Application Guard

Article • 09/27/2024

Important

Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is deprecated for Microsoft Edge for Business and will no longer be updated. Starting with Windows 11, version 24H2, Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, is no longer available.

Existing installations of Application Guard

This deprecation does not impact the existing installations of Microsoft Defender Application Guard (MDAG). Organizations can continue to use Application Guard on current versions of Windows, but we recommend that security admins evaluate their security requirements going forward. This feature might be removed in a future Windows release, but it will continue to be maintained for existing installations on Windows.

Deprecation considerations

Deprecation includes the following elements of Application Guard.

- If your organization requires container-based isolation, we recommend Windows Sandbox or Azure Virtual Desktop (AVD).
- Because Application Guard is deprecated there will not be a migration to Edge Manifest V3. The corresponding extensions and associated Windows Store app will not be available after May 2024. This affects the following browsers: *Chrome* and *Firefox*. If you want to block unprotected browsers until you are ready to retire MDAG usage in your enterprise, we recommend using AppLocker policies or Microsoft Edge management service.

Tip

Download the Microsoft Edge for Business Security Whitepaper to learn more about the capabilities that make Edge for Business a secure enterprise browser.

The additional security features in Edge make it very secure without needing Application Guard. The growing list of security features includes:

- Defender SmartScreen for the anti-phishing and malware support and URL scanning and blocking.
- Enhanced security mode for protecting against memory-related vulnerabilities by disabling just-in-time JavaScript compilation (and other protections).
- Website typo protection for misspelled websites.
- Data Loss Prevention to identify, monitor, and automatically protect sensitive items.

Overview

This article describes how Microsoft Edge supports Microsoft Defender Application Guard (Application Guard).

Security architects in the enterprise must deal with the tension that exists between productivity and security. It's relatively easy to lock down a browser and only allow a handful of trusted sites to load. This approach will improve the overall security posture but is arguably less productive. If you make it less restrictive to improve productivity, you increase the risk profile. It's a hard balance to strike!

It's even harder to keep up with new emerging threats in this constantly changing threat landscape. Browsers remain the primary attack surface on client devices because the browser's basic job is to let users access, download, and open untrusted content from untrusted sources. Malicious actors are constantly working to social engineer new forms of attacks against the browser. Security incident prevention or detection/response strategies can't guarantee 100% safety.

A key security strategy to consider is the Assume Breach Methodology, which means there's an acceptance that an attack is going to succeed at least once regardless of efforts to prevent it. This mindset requires building defenses to contain the damage, which ensures that corporate network and other resources remain protected in this scenario. Deploying Application Guard for Microsoft Edge fits right into this strategy.

About Application Guard

Designed for Windows 10/11 and Microsoft Edge, Application Guard uses a hardware isolation approach. This approach lets untrusted site navigation launch inside a

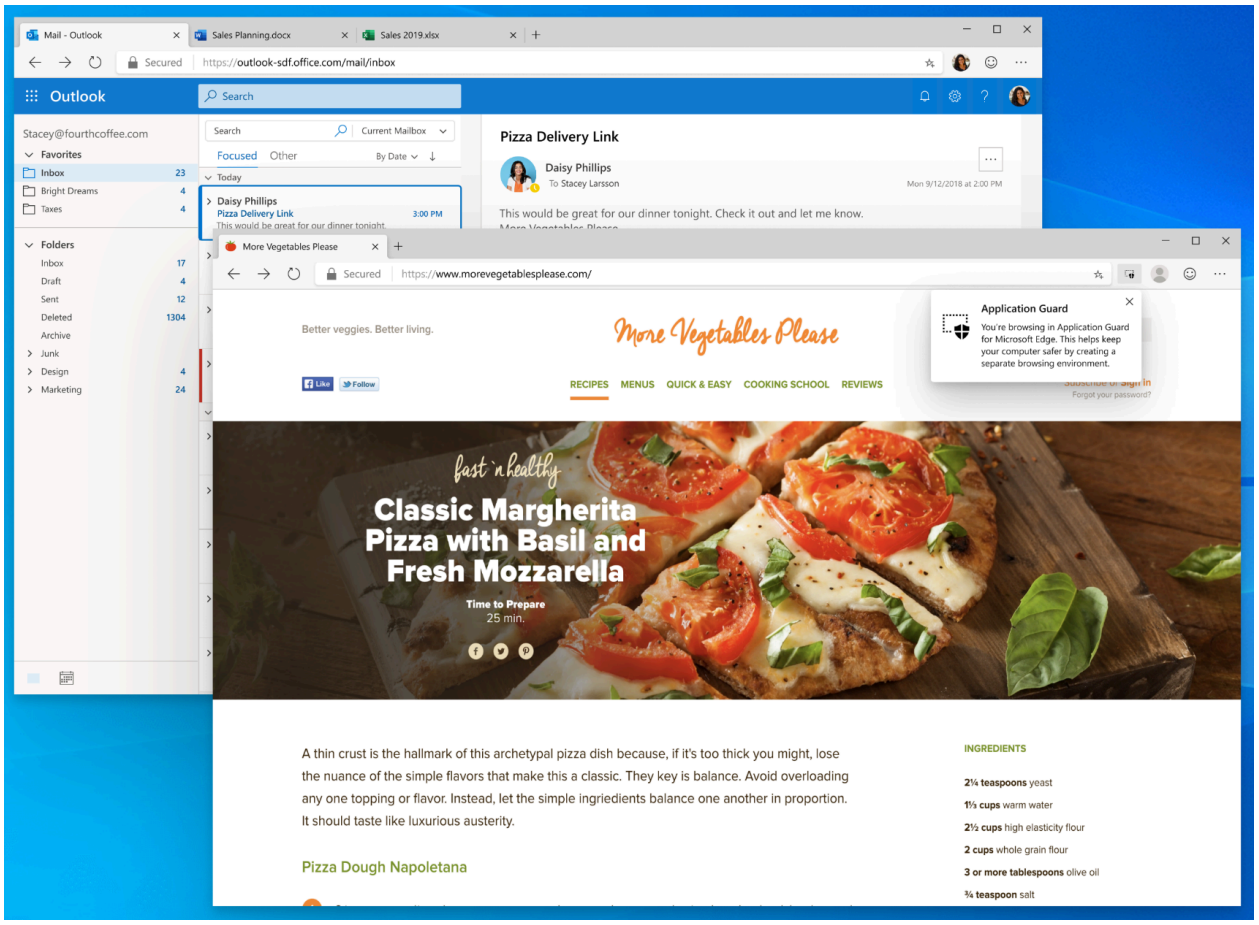
container. Hardware isolation helps enterprises safeguard their corporate network and data in case users visit a site that is compromised or is malicious.

The enterprise administrator defines what are trusted sites, cloud resources, and internal networks. Everything that's not in the trusted sites list is considered untrusted. These sites are isolated from the corporate network and data on the user's device.

For more information:

- watch our video Microsoft Edge browser isolation using Application Guard
- read What is Application Guard and how does it work?

The next screenshot shows an example of Application Guard's message showing that the user is browsing in a safe space.



What's new

Application Guard support in the new Microsoft Edge browser has functional parity with Microsoft Edge Legacy and includes several improvements.

Enable Upload Blocking

Starting from Microsoft Edge 96, admins now have the option to block uploads while in the container, meaning that users cannot upload files from their local device to their Application Guard instance. This support can be controlled via policy. You can update the Edge policy ApplicationGuardUploadBlockingEnabled to enable or disable uploads in the container.

Enable Application Guard in passive mode and browse Edge normally

Starting from Microsoft Edge 94, users now have the option to configure passive mode, meaning that Application Guard ignores the site list configuration and users can browse Edge normally. This support can be controlled via policy. You can update the Edge policy ApplicationGuardPassiveModeEnabled to enable or disable passive mode.

> ⓘ **Note**
>
> This policy ONLY impacts Edge, so navigations from other browsers might get redirected to the Application Guard Container if you have the corresponding extensions enabled.

Favorites synchronizing from the host to the container

Some of our customers have been asking for favorites sync between the host browser and the container in Application Guard. Starting from Microsoft Edge 91, users now have the option to configure Application Guard to synchronize their favorites from the host to the container. This ensures new favorites appear on the container as well.

This support can be controlled via policy. You can update the Edge policy ApplicationGuardFavoritesSyncEnabled to enable or disable favorites sync.

> ⓘ **Note**
>
> For security reasons, favorites sync is only possible from the host to the container and not the other way around. To ensure a unified list of favorites across the host and the container, we have disabled favorites management inside the container.

Identify network traffic originating from the container

Several customers are using WDAG in a specific configuration where they want to identify network traffic coming from the container. Some of the scenarios for this are:

- To restrict access to only a handful of untrusted sites
- To allow internet access from the container only

Starting with Microsoft Edge version 91, there's built in support to tag network traffic originating from Application Guard containers, allowing enterprises to use proxy to filter out traffic and apply specific policies. You can use the header to identify which traffic is through the container or the host using ApplicationGuardTrafficIdentificationEnabled.

Extension support inside the container

Extension support inside the container has been one of the top requests from the customers. Scenarios ranged from wanting to run ad-blockers inside the container to boost browser performance to having the ability to run custom home-grown extensions inside the container.

Extension installs in the container is now supported, starting from Microsoft Edge version 81. This support can be controlled via policy. The `updateURL` that gets used in ExtensionInstallForcelist policy should be added as Neutral Resources in the Network Isolation policies used by Application Guard.

Some examples of container support include the following scenarios:

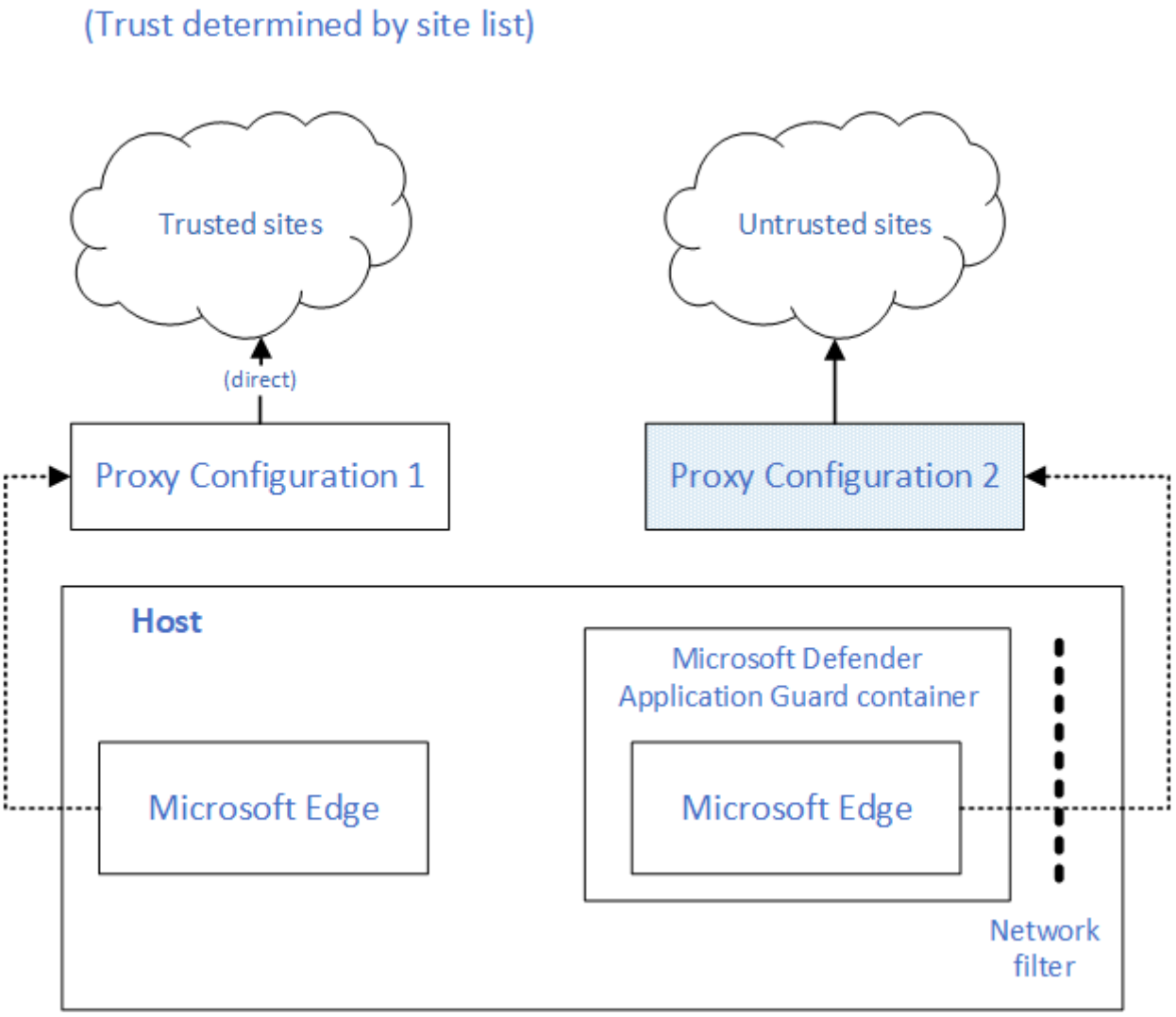
- Force installs of an extension on the host
- Removing an extension from the host
- Extensions blocked on the host

> ⓘ **Note**
>
> It's also possible to manually install individual extensions inside the container from the extension store. Manually installed extensions will only persist in the container when **Allow Persistence** policy is enabled.

Identifying Application Guard traffic via Dual Proxy

Some enterprise customers are deploying Application Guard with a specific use case where they need to identify web traffic coming out of a Microsoft Defender Application Guard container at the proxy level. Starting with Stable Channel version 84, Microsoft Edge will support dual proxy to address this requirement. You can configure this functionality using the ApplicationGuardContainerProxy policy.

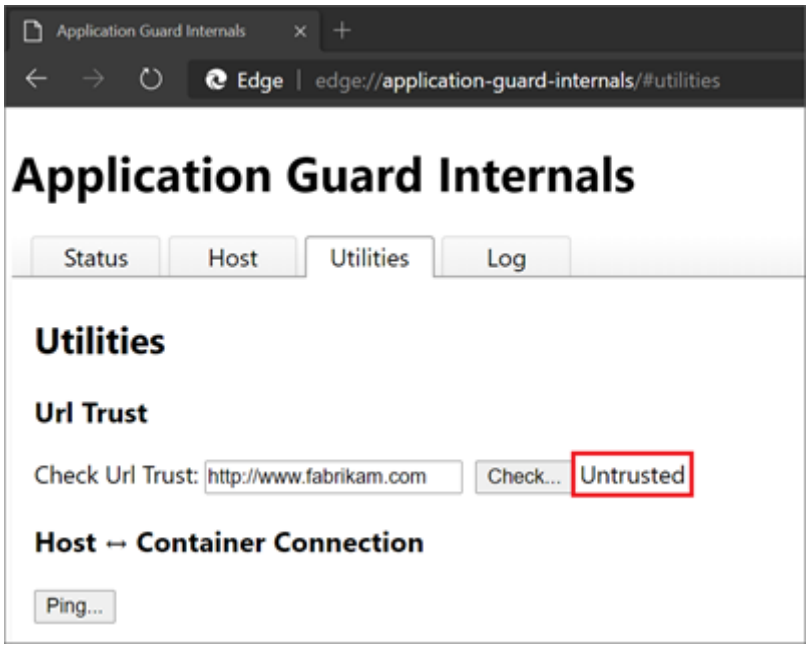
The following drawing shows the dual proxy architecture for Microsoft Edge.



Diagnostic page for troubleshooting

Another user pain point is troubleshooting the Application Guard configuration on a device when a problem is reported. Microsoft Edge has a diagnostics page (`edge://application-guard-internals`) to troubleshoot user issues. One of these diagnostics is being able to check the URL trust based on the configuration on the user's device.

The next screenshot shows a multiple tab diagnostics page to help diagnose user reported issues on the device.



Microsoft Edge updates in the container

Microsoft Edge Legacy updates in the container are part of the Windows OS update cycle. Because the new version of Microsoft Edge updates itself independent of the Windows OS, there is no longer any dependency on container updates. The channel and version of the host Microsoft Edge is replicated inside the container.

Prerequisites

The following requirements apply to devices using Application Guard with Microsoft Edge:

- Windows 10 1809 (Windows 10 October 2018 Update) and above.
- Only Windows client SKUs

  > ⓘ **Note**
  >
  > Application Guard is only supported on Windows 10/11 Pro and Windows 10/11 Enterprise SKUs.

- One of the management solutions described in Software requirements

How to install Application Guard

The following articles provide the information you need to install, configure, and test Application Guard with Microsoft Edge.

- System requirements
- Install Microsoft Defender Application Guard
- Configure Application Guard group policy settings
- Test Application Guard

Frequently Asked Questions

Is Application Guard being deprecated?

Yes, Microsoft Defender Application Guard, including the Windows Isolated App Launcher APIs, will be deprecated for Microsoft Edge for Business and will no longer be updated.

Does Application Guard work in IE mode?

IE mode supports Application Guard functionality, but we don't anticipate much use of this feature in IE Mode. IE mode is recommended to be deployed for a list of trusted internal sites, and Application Guard is for untrusted sites only. Make sure all the IE mode sites or IP addresses are also added to the Network Isolation policy to be considered as trusted resource by Application Guard.

Do I need to install the Application Guard Chrome extension?

No, the Application Guard feature is natively supported in Microsoft Edge. In fact, the Application Guard Chrome extension isn't a supported configuration in Microsoft Edge.

Can employees download documents from the Application Guard Edge session onto host devices?

In Windows 10 Enterprise edition, version 1803, users are able to download documents from the isolated Application Guard container to the host PC. This capability is managed by policy.

In Windows 10 Enterprise edition, version 1709, or Windows 10 Professional edition, version 1803, it is not possible to download files from the isolated Application Guard container to the host computer. However, employees can use the Print as PDF or Print as XPS options and save those files to the host device.

Can employees copy and paste between the host device and the Application Guard Edge session?

Depending on your organization's settings, employees can copy and paste images (.bmp) and text to and from the isolated container.

Why don't employees see their favorites in the Application Guard Edge session?

Depending on your organization's settings, it might be that Favorites Sync is turned off. To manage the policy, see: Microsoft Edge and Microsoft Defender Application Guard | Microsoft Docs.

Why aren't employees able to see their extensions in the Application Guard Edge session?

Make sure to enable the extensions policy on your Application Guard configuration.

My extension doesn't seem to work in Edge Application Guard?

If the extensions policy is enabled for MDAG in configuration, check if your extension requires Native Message Handling components, those extensions are not supported in the Application Guard container.

I'm trying to watch playback video with HDR, why is the HDR option missing?

In order for HDR video playback to work in the container, vGPU Hardware Acceleration needs to be enabled in Application Guard.

Are there any other platform related FAQs?

Yes. Frequently asked questions - Microsoft Defender Application Guard

See also

- Microsoft Edge Enterprise landing page
- Microsoft Defender Advanced Threat Protection

- Video: Microsoft Edge browser isolation using Application Guard

Feedback

Was this page helpful? Yes No

Provide product feedback

WindowsDefenderApplicationGuard CSP

Article • 03/12/2025

The WindowsDefenderApplicationGuard configuration service provider (CSP) is used by the enterprise to configure the settings in Microsoft Defender Application Guard. This CSP was added in Windows 10, version 1709.

Windows edition and licensing requirements

The following table lists the Windows editions that support Microsoft Defender Application Guard (MDAG) configure via MDM:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
No	Yes	No	Yes

Microsoft Defender Application Guard (MDAG) configure via MDM license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
No	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

The following list shows the WindowsDefenderApplicationGuard configuration service provider nodes:

- ./Device/Vendor/MSFT/WindowsDefenderApplicationGuard
  - Audit
    - AuditApplicationGuard
  - InstallWindowsDefenderApplicationGuard
  - PlatformStatus
  - Settings
    - AllowCameraMicrophoneRedirection

- AllowPersistence
- AllowVirtualGPU
- AllowWindowsDefenderApplicationGuard
- BlockNonEnterpriseContent
- CertificateThumbprints
- ClipboardFileType
- ClipboardSettings
- PrintingSettings
- SaveFilesToHost

- Status

Audit

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Audit
```

Interior node for Audit.

Description framework properties:

Expand table

Property name	Property value
Format	`node`
Access Type	Get

Audit/AuditApplicationGuard

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Audit/AuditApplicationGuard
```

This policy setting allows you to decide whether auditing events can be collected from Application Guard.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Audit event logs aren't collected for Application Guard.
1	Application Guard inherits its auditing policies from system and starts to audit security events for Application Guard container.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_AuditApplicationGuardConfig
Friendly Name	Allow auditing events in Microsoft Defender Application Guard
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	AuditApplicationGuard
ADMX File Name	AppHVSI.admx

InstallWindowsDefenderApplicationGuard

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/InstallWindowsDefenderApplicationGuard
```

Initiates remote installation of Application Guard feature.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec, Get

Allowed values:

Expand table

Value	Description
Install	Will initiate feature install.
Uninstall	Will initiate feature uninstall.

PlatformStatus

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 2004 [10.0.19041] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/PlatformStatus
```

Returns bitmask that indicates status of Application Guard platform installation and prerequisites on the device. Bit 0 - Set to 1 when Application Guard is enabled into enterprise manage mode. Bit 1 - Set to 1 when the client machine is Hyper-V capable. Bit 2 - Reserved for Microsoft. Bit 3 - Set to 1 when Application Guard is installed on the client machine. Bit 4 - Reserved for Microsoft. Bit 5 - Set to 1 when the client machine meets minimum hardware requirements.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Get

Settings

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings
```

Interior Node for Settings.

Description framework properties:

Expand table

Property name	Property value
Format	`node`
Access Type	Get

Settings/AllowCameraMicrophoneRedirection

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/AllowCameraMicrophoneRedirection
```

This policy setting allows you to determine whether applications inside Microsoft Defender Application Guard can access the device's camera and microphone when these

settings are enabled on the user's device.

- If you enable this policy setting, applications inside Microsoft Defender Application Guard will be able to access the camera and microphone on the user's device.
- If you disable or don't configure this policy setting, applications inside Microsoft Defender Application Guard will be unable to access the camera and microphone on the user's device.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Microsoft Defender Application Guard can't access the device's camera and microphone. When the policy isn't configured, it's the same as disabled (0).
1	Turns on the functionality to allow Microsoft Defender Application Guard to access the device's camera and microphone.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_AllowCameraMicrophoneRedirectionConfig
Friendly Name	Allow camera and microphone access in Microsoft Defender Application Guard
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard

Name	Value
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	AllowCameraMicrophoneRedirection
ADMX File Name	AppHVSI.admx

Settings/AllowPersistence

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/AllowPersistence
```

This policy setting allows you to decide whether data should persist across different sessions in Application Guard.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace

Allowed values:

Expand table

Value	Description
0	Application Guard discards user-downloaded files and other items (such as, cookies, Favorites, and so on) during machine restart or user log-off.
1	Application Guard saves user-downloaded files and other items (such as, cookies, Favorites, and so on) for use in future Application Guard sessions.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_AllowPersistence
Friendly Name	Allow data persistence for Microsoft Defender Application Guard
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	AllowPersistence
ADMX File Name	AppHVSI.admx

Settings/AllowVirtualGPU

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1803 [10.0.17134] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/AllowVirtualGPU
```

This policy setting allows you to determine whether Application Guard can use the virtual Graphics Processing Unit (GPU) to process graphics. If you enable this setting,

Microsoft Defender Application Guard uses Hyper-V to access supported, high-security rendering graphics hardware (GPUs). These GPUs improve rendering performance and battery life while using Microsoft Defender Application Guard, particularly for video playback and other graphics-intensive use cases. If you enable this setting without connecting any high-security rendering graphics hardware, Microsoft Defender Application Guard will automatically revert to software-based (CPU) rendering.

> ⚠ **Warning**
>
> Enabling this setting with potentially compromised graphics devices or drivers might pose a risk to the host device.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Cannot access the vGPU and uses the CPU to support rendering graphics. When the policy isn't configured, it's the same as disabled (0).
1	Turns on the functionality to access the vGPU offloading graphics rendering from the CPU. This can create a faster experience when working with graphics intense websites or watching video within the container.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_AllowVirtualGPU

Name	Value
Friendly Name	Allow hardware-accelerated rendering for Microsoft Defender Application Guard
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	AllowVirtualGPU
ADMX File Name	AppHVSI.admx

Settings/AllowWindowsDefenderApplicationGuard

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/AllowWindowsDefenderApplicationGuard
```

Turn on Microsoft Defender Application Guard in Enterprise Mode.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace

Allowed values:

Expand table

Value	Description
0	Disable Microsoft Defender Application Guard.
1	Enable Microsoft Defender Application Guard for Microsoft Edge ONLY.
2	Enable Microsoft Defender Application Guard for isolated Windows environments ONLY.
3	Enable Microsoft Defender Application Guard for Microsoft Edge AND isolated Windows environments.

Group policy mapping:

Expand table

Name	Value
Name	AllowAppHVSI
Path	Windows Components > Microsoft Defender Application Guard

Settings/BlockNonEnterpriseContent

> **Note**
>
> This policy is deprecated and may be removed in a future release.

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/BlockNonEnterpriseContent
```

This policy setting allows you to decide whether websites can load non-enterprise content in Microsoft Edge and Internet Explorer.

> **Note**
>
> This policy setting is no longer supported in the new Microsoft Edge browser. The policy will be deprecated and removed in a future release. Webpages that contain mixed content, both enterprise and non-enterprise, may load incorrectly or fail completely if this feature is enabled.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Non-enterprise content embedded in enterprise sites is allowed to open outside of the Microsoft Defender Application Guard container, directly in Internet Explorer and Microsoft Edge.
1	Non-enterprise content embedded on enterprise sites are stopped from opening in Internet Explorer or Microsoft Edge outside of Microsoft Defender Application Guard.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_BlockNonEnterpriseContentConfig
Friendly Name	Prevent enterprise websites from loading non-enterprise content in Microsoft Edge and Internet Explorer

Name	Value
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	BlockNonEnterpriseContent
ADMX File Name	AppHVSI.admx

Settings/CertificateThumbprints

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/CertificateThumbprints
```

This policy setting allows certain device level Root Certificates to be shared with the Microsoft Defender Application Guard container.

- If you enable this setting, certificates with a thumbprint matching the ones specified will be transferred into the container. Multiple certificates can be specified by using a comma to separate the thumbprints for each certificate you want to transfer. Here's an example: b4e72779a8a362c860c36a6461f31e3aa7e58c14,1b1d49f06d2a697a544a1059bd59a7b058cda924.
- If you disable or don't configure this setting, certificates aren't shared with the Microsoft Defender Application Guard container.

> **Note**
>
> To enforce this policy, device restart or user logon/logoff is required.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Add, Delete, Get, Replace
Allowed Values	List (Delimiter: `,`)

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_CertificateThumbprints
Friendly Name	Allow Microsoft Defender Application Guard to use Root Certificate Authorities from the user's device
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
ADMX File Name	AppHVSI.admx

Settings/ClipboardFileType

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education	✅ Windows 10, version 1709 [10.0.16299] and later

Scope	Editions	Applicable OS
	✅ IoT Enterprise / IoT Enterprise LTSC	

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/ClipboardFileType
```

Determines the type of content that can be copied from the host to Application Guard environment and vice versa.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace

Allowed values:

Expand table

Value	Description
1	Allow text copying.
2	Allow image copying.
3	Allow text and image copying.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_ClipboardConfig
Friendly Name	Configure Microsoft Defender Application Guard clipboard settings
Location	Computer Configuration

Name	Value
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
ADMX File Name	AppHVSI.admx

Settings/ClipboardSettings

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/ClipboardSettings
```

This policy setting allows you to decide how the clipboard behaves while in Application Guard.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Completely turns Off the clipboard functionality for the Application Guard.
1	Turns On clipboard operation from an isolated session to the host.
2	Turns On clipboard operation from the host to an isolated session.
3	Turns On clipboard operation in both the directions.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_ClipboardConfig
Friendly Name	Configure Microsoft Defender Application Guard clipboard settings
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
ADMX File Name	AppHVSI.admx

Settings/PrintingSettings

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/PrintingSettings
```

This policy setting allows you to decide how the print functionality behaves while in Application Guard.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	Disables all print functionality.
1	Enables only XPS printing.
2	Enables only PDF printing.
3	Enables both PDF and XPS printing.
4	Enables only local printing.
5	Enables both local and XPS printing.
6	Enables both local and PDF printing.
7	Enables local, PDF, and XPS printing.
8	Enables only network printing.
9	Enables both network and XPS printing.
10	Enables both network and PDF printing.
11	Enables network, PDF, and XPS printing.
12	Enables both network and local printing.
13	Enables network, local, and XPS printing.
14	Enables network, local, and PDF printing.
15	Enables all printing.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_PrintingConfig
Friendly Name	Configure Microsoft Defender Application Guard print settings
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
ADMX File Name	AppHVSI.admx

Settings/SaveFilesToHost

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1803 [10.0.17134] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Settings/SaveFilesToHost
```

This policy setting allows you to determine whether users can elect to download files from Edge in the container and persist files them from container to the host operating system.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Add, Delete, Get, Replace

Property name	Property value
Default Value	0

Allowed values:

Expand table

Value	Description
0 (Default)	The user can't download files from Edge in the container to the host file system. When the policy isn't configured, it's the same as disabled (0).
1	Turns on the functionality to allow users to download files from Edge in the container to the host file system.

Group policy mapping:

Expand table

Name	Value
Name	AppHVSI_SaveFilesToHost
Friendly Name	Allow files to download and save to the host operating system from Microsoft Defender Application Guard
Location	Computer Configuration
Path	Windows Components > Microsoft Defender Application Guard
Registry Key Name	SOFTWARE\Policies\Microsoft\AppHVSI
Registry Value Name	SaveFilesToHost
ADMX File Name	AppHVSI.admx

Status

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/WindowsDefenderApplicationGuard/Status
```

Returns bitmask that indicates status of Application Guard installation and pre-requisites on the device. Bit 0 - Set to 1 when Application Guard is enabled into enterprise manage mode. Bit 1 - Set to 1 when the client machine is Hyper-V capable. Bit 2 - Set to 1 when the client machine has a valid OS license and SKU. Bit 3 - Set to 1 when Application Guard installed on the client machine. Bit 4 - Set to 1 when required Network Isolation Policies are configured. Bit 5 - Set to 1 when the client machine meets minimum hardware requirements. Bit 6 - Set to 1 when system reboot is required.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Get

Related articles

Configuration service provider reference

Feedback

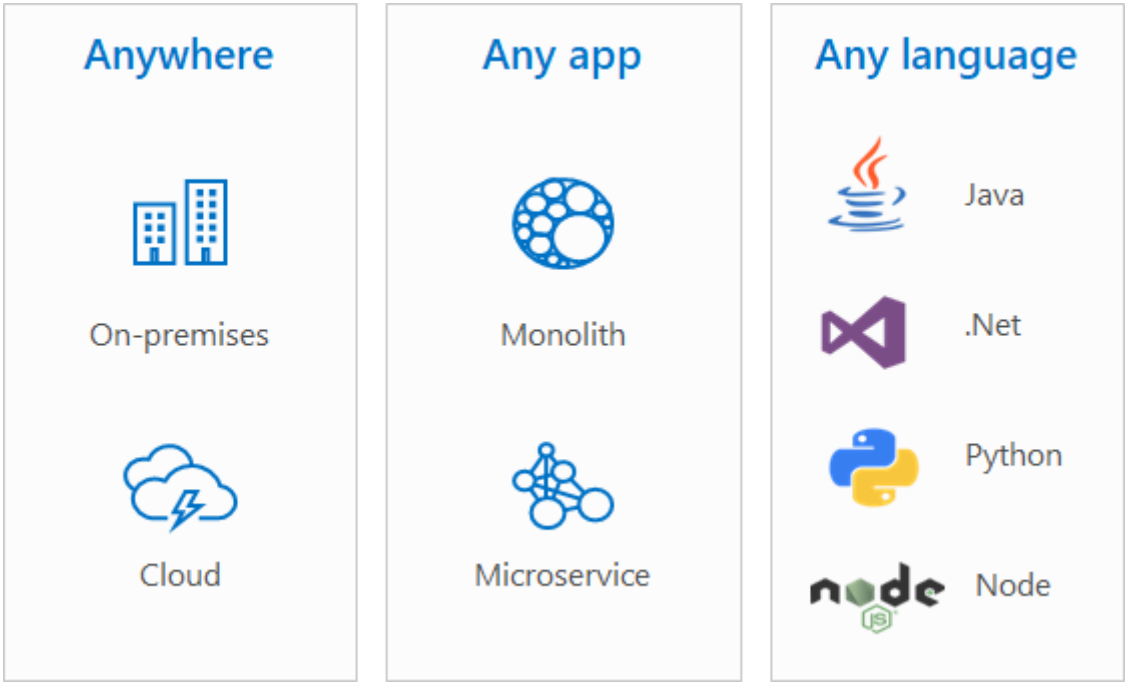
Was this page helpful? Yes No

Provide product feedback

Windows and containers

> Applies to: Windows Server 2025, Windows Server 2022, Windows Server 2019, Windows Server 2016

Containers are a technology for packaging and running Windows and Linux applications across diverse environments on-premises and in the cloud. Containers provide a lightweight, isolated environment that makes apps easier to develop, deploy, and manage. Containers start and stop quickly, making them ideal for apps that need to rapidly adapt to changing demand. The lightweight nature of containers also makes them a useful tool for increasing the density and utilization of your infrastructure.



The Microsoft container ecosystem

Microsoft provides many tools and platforms to help you develop and deploy apps in containers:

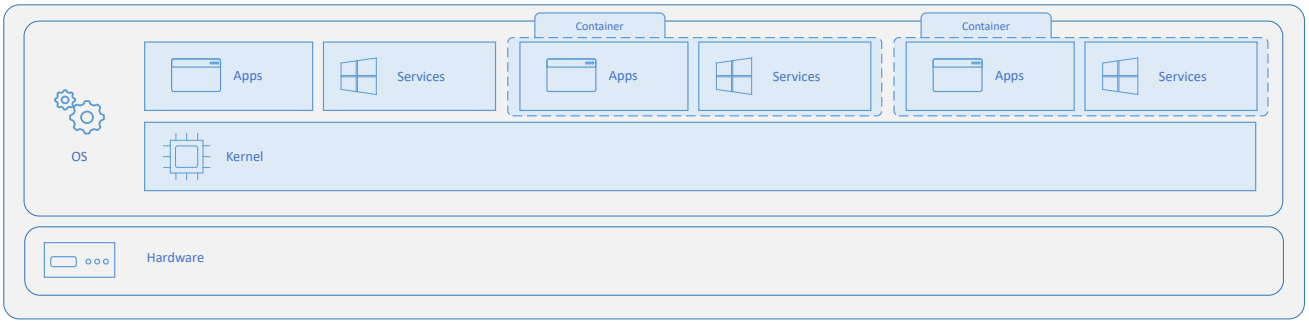
- **Run Windows-based or Linux-based containers on Windows 10** for development and testing using Docker Desktop, which makes use of containers functionality built-in to Windows. You can also run containers natively on Windows Server.
- **Develop, test, publish, and deploy Windows-based containers** using the powerful container support in Visual Studio and Visual Studio Code, which include support for Docker, Docker Compose, Kubernetes, Helm, and other useful technologies.
- **Publish your apps as container images** to the public DockerHub for others to use, or to a private Azure Container Registry for your org's own development and deployment,

pushing and pulling directly from within Visual Studio and Visual Studio Code.

- **Deploy containers at scale on Azure** or other clouds:
  - Pull your app from a container registry, such as the Azure Container Registry, and then deploy and manage it at scale using an orchestrator such as Azure Kubernetes Service (AKS).
  - Azure Kubernetes Service deploys containers to Azure virtual machines and manages them at scale. You can run dozens, hundreds, or even thousands of containers.
  - Azure virtual machines run either a customized Windows Server image for Windows-based apps, or a customized Ubuntu Linux image for Linux-based apps.
- **Deploy containers on-premises** by using Azure Kubernetes Service (AKS) enabled by Azure Arc, AKS engine on Azure Stack Hub, or Azure Stack Hub with OpenShift. You can also set up Kubernetes yourself on Windows Server. For more information, see Kubernetes on Windows. Microsoft is also developing support for Windows containers on RedHat OpenShift Container Platform.

How containers work

A container is an isolated, lightweight package for running an application on the host operating system. Containers run on top of the host operating system's kernel (which you can think of as the buried plumbing of the operating system), as shown in the following diagram.



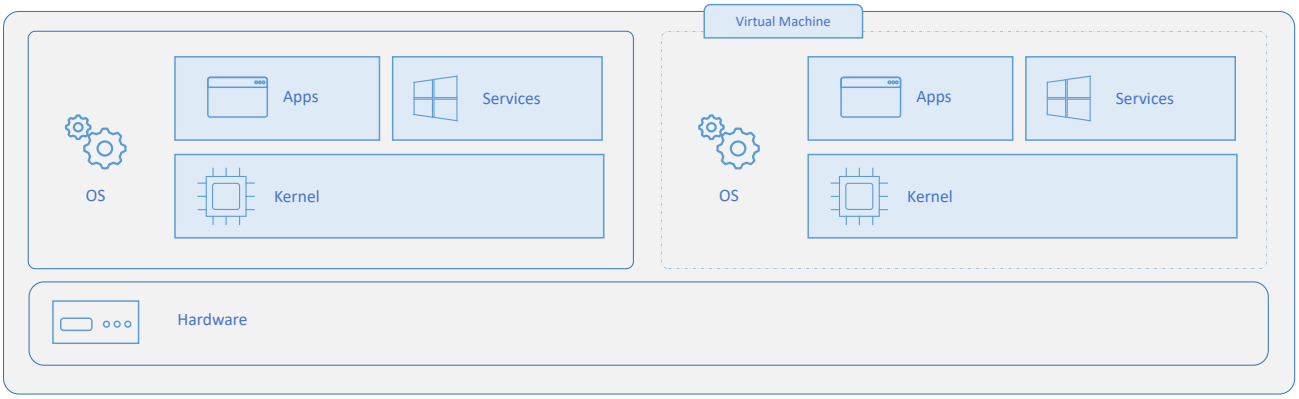
While a container shares the host operating system's kernel, the container doesn't get unfettered access to it. Instead, the container gets an isolated–and in some cases virtualized–view of the system. For example, a container can access a virtualized version of the file system and registry, but any changes affect only the container and are discarded when it stops. To save data, the container can mount persistent storage such as an Azure Disk or a file share like Azure Files.

A container builds on top of the kernel, but the kernel doesn't provide all of the APIs and services an app needs to run. Most of these APIs and services come from system files (libraries)

that run above the kernel in user mode. Because a container is isolated from the host's user mode environment, the container needs its own copy of these user mode system files. These files are packaged into something known as a base image. The base image serves as the foundational layer upon which you build your container, providing it with operating system services not provided by the kernel.

Containers vs. virtual machines

In contrast to a container, a virtual machine (VM) runs a complete operating system–including its own kernel–as shown in the following diagram.



Containers and virtual machines each have their uses. In fact, many deployments of containers use virtual machines as the host operating system rather than running directly on the hardware, especially when running containers in the cloud.

For more information on the similarities and differences of these complementary technologies, see Containers vs. virtual machines.

Container images

You create all containers from container images. A container image is a bundle of files organized into a stack of layers that resides on your local machine or in a remote container registry. A container image consists of the user mode operating system files that your app needs to run. The image also includes your app's runtimes, dependencies, and any configuration files that your app requires.

Microsoft offers several images (called base images) that you can use as a starting point to build your own container image:

- **Windows** - contains the full set of Windows APIs and system services (minus server roles).
- **Windows Server** - contains the full set of Windows APIs and system services.

- **Windows Server Core** - a smaller image that contains a subset of the Windows Server APIs–namely the full .NET framework. It also includes most but not all server roles (for example Fax Server isn't included).
- **Nano Server** - the smallest Windows Server image and includes support for the .NET Core APIs and some server roles.

Container images are composed of a series of layers. Each layer contains a set of files that, when overlaid together, represent your container image. Because of the layered nature of containers, you don't always have to target a base image to build a Windows container. Instead, you can target another image that already carries the framework you want. For example, the .NET team publishes a .NET core image that carries the .NET core runtime. It saves users from needing to duplicate the process of installing .NET core–instead they can reuse the layers of this container image. The .NET core image itself is built based upon Nano Server.

For more information, see Container Base Images.

Container users

Containers for developers

Containers help developers build and ship higher-quality apps, faster. With containers, developers can create a container image that deploys in seconds, identically across environments. Containers act as an easy mechanism to share code across teams and to bootstrap a development environment without impacting your host filesystem.

Containers are portable and versatile, can run apps written in any language, and they're compatible with any machine running Windows 10, version 1607 or later, or Windows Server 2016 or later. Developers can create and test a container locally on their laptop or desktop, and then deploy that same container image to their company's private cloud, public cloud, or service provider. The natural agility of containers supports modern app development patterns in large-scale, virtualized cloud environments. The most useful benefit to developers is perhaps the ability to isolate your environment so that your app always gets the version of libraries that you specify, avoiding conflicts with dependencies.

Containers for IT professionals

Containers help admins create infrastructure that's easier to update and maintain, and that more fully utilizes hardware resources. IT professionals can use containers to provide standardized environments for their development, QA, and production teams. By using

containers, systems administrators abstract away differences in operating system installations and the underlying infrastructure.

You can also use the interactive mode of containers to run conflicting instances of a command line tool on the same system.

Container orchestration

Orchestrators are a critical piece of infrastructure when setting up a container-based environment. Orchestrators are essential when managing containerized apps at scale. While you can manage a few containers manually using Docker and Windows, apps often make use of five, ten, or even hundreds of containers, making orchestrators indispensable.

Container orchestrators were built to help manage containers at scale and in production. Orchestrators provide functionality for:

Orchestrators help you grow containerized apps at scale, providing functionality for:

- Deploying at scale
- Workload scheduling
- Health monitoring
- Failing over when a node fails
- Scaling up or down
- Networking
- Service discovery
- Coordinating app upgrades
- Cluster node affinity

There are many different orchestrators that you can use with Windows containers; here are the options Microsoft provides:

- Azure Kubernetes Service (AKS) - use a managed Azure Kubernetes service
- Azure Kubernetes Service (AKS) on Azure Stack HCI - use Azure Kubernetes Service on-premises

Try containers on Windows

To get started with containers on Windows Server or Windows 10, see the following article:

Get started: Configure Your Environment for Containers

For help with deciding which Azure services are right for your scenario, see Azure container services and Choosing what Azure services to use to host your application.

Resources

To view resources for using Windows Server containers, see the following resources:

- For current issues and planned feature upgrades, see the Windows containers GitHub repository.
- Check out the blog: Windows Containers Blog.
- To contact the Windows Server containers team, send email to the Windows Containers Customers group.

Last updated on 11/28/2025

Windows Sandbox

Applies to: ✅ Windows 11, ✅ Windows 10

Windows Sandbox (WSB) offers a lightweight, isolated desktop environment for safely running applications. It's ideal for testing, debugging, exploring unknown files, and experimenting with tools. Applications installed within the sandbox remain isolated from the host machine using hypervisor-based virtualization. As a disposable virtual machine (VM), Windows Sandbox ensures reboot persistence, quick launch times, and a lower memory footprint compared to full VMs. Its one-click setup simplifies the user experience.

The sandbox is temporary; closing it deletes all software, files, and state. Each launch provides a fresh instance. Host-installed software isn't available in the sandbox. Applications needed within the sandbox must be installed there explicitly.

> ⓘ **Note**
>
> Starting with Windows 11, version 22H2, data persists through restarts initiated within the sandbox, useful for applications requiring a reboot.

Windows Sandbox offers the following features:

- **Part of Windows**: Everything required for this feature is included in the supported Windows editions like Pro, Enterprise, and Education. There's no need to maintain a separate VM installation.
- **Disposable**: Nothing persists on the device. Everything is discarded when the user closes the application.
- **Pristine**: Every time Windows Sandbox runs, it's as clean as a brand-new installation of Windows.
- **Secure**: Uses hardware-based virtualization for kernel isolation. It relies on the Microsoft hypervisor to run a separate kernel that isolates Windows Sandbox from the host.
- **Efficient**: Takes a few seconds to launch, supports virtual GPU, and has smart memory management that optimizes memory footprint.

> ⓘ **Important**
>
> Windows Sandbox enables network connection by default. It can be disabled using the Windows Sandbox configuration file. Enabling networking can expose untrusted applications to the internal network.

WSB can be used without any technical skills in various scenarios where users need a secure, clean environment for testing or running potentially harmful software. Here are some ways in which you can use WSB:

- **Clean environment for software testing**: Test or debug your applications in WSB's clean environment to identify and resolve bugs or compatibility issues.
- **Secure web browsing**: Use WSB for secure web browsing, especially when accessing unfamiliar or potentially dangerous websites without putting your system at risk of malware infection.
- **Running Untrusted Applications**: Mitigate security risks by opening untrusted applications or files, such as email attachments in WSB. Improve your safety and security by opening a sandbox with networking disabled and mapping the folder with the application or file you want to open to the sandbox in read-only mode. Check Sample configuration files for more details.
- **Testing or demoing new software for the first time**: Test drive or demo new software, preview versions, extensions, or add-ons without the hassle of installing and then uninstalling on your host machine.
- **Maintaining multiple dev environments**: Streamline your development process by utilizing WSB to maintain multiple sandboxes for different development environments. For example, maintain a sandbox for each python version and its dependencies!

> **Note**
>
> Windows Sandbox currently doesn't allow multiple instances to run simultaneously.

Windows edition and licensing requirements

The following table lists the Windows editions that support Windows Sandbox:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Windows Sandbox license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

> **Note**
>
> Windows Sandbox is currently not supported on Windows Home edition.

Last updated on 03/29/2026

Passwordless strategy overview

Article • 10/29/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Microsoft's passwordless strategy and how Windows security features help implementing it.

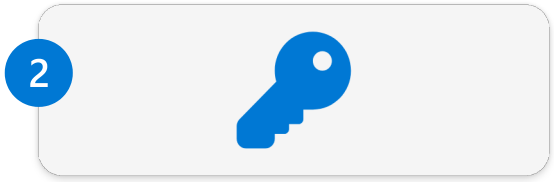
Four steps to password freedom

Microsoft is working hard to create a world where passwords are no longer needed. This is how Microsoft envisions the four steps approach to end the era of passwords for the organizations:



Deploy a password replacement option

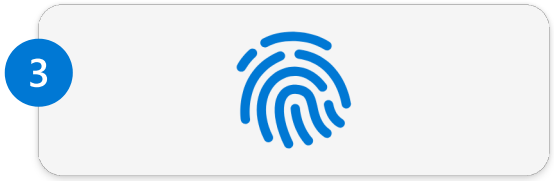
Before you move away from passwords, you need something to replace them. Windows Hello for Business and FIDO2 security keys offer a strong, hardware-protected two-factor credential that enables single sign-on to Microsoft Entra ID and Active Directory. Deploy Windows Hello for Business or FIDO2 security keys is the first step toward a passwordless environment. Users are likely to use these features because of their convenience, especially when combined with biometrics. However, some workflows and applications might still need passwords. This early stage is about implementing an alternative solution to passwords, and getting users accustomed to it.



Reduce user-visible password surface area

With a password replacement option and passwords coexisting in the environment, the next step is to reduce the password surface area. The environment and workflows need to stop asking for passwords. The goal of this step is to achieve a state where the users

know they have a password, **but they never use it**. This state helps decondition users from providing a password anytime a password prompt shows on their computer. This behavior is how passwords are phished. Users who rarely, if at all, use their password are unlikely to provide it. **Password prompts are no longer the norm**.



Transition into a passwordless deployment

Once the user-visible password surface is eliminated, your organization can begin to transition users into a passwordless environment. In this stage, users never type, change, or even know their password.
The user signs in to Windows using Windows Hello for Business or FIDO2 security keys, and enjoys single sign-on to Microsoft Entra ID and Active Directory resources. If the user is forced to authenticate, their authentication uses Windows Hello for Business or FIDO2 security keys.



Eliminate passwords from the identity directory

The final step of the passwordless journey is where passwords don't exist. At this stage, identity directories don't store any form of the password.

Prepare for the passwordless journey

The road to being passwordless is a journey. The duration of the journey varies for each organization. It's important for IT decision makers to understand the criteria influencing the length of that journey.

The most intuitive answer is the size of the organization, but what exactly defines size? We can look at these factors to get a summary of the organization's size:

Expand table

Size factor	Details
Number of departments	The number of departments within an organization varies. Most organizations have a common set of departments such as *executive leadership*, *human resources*, *accounting*, *sales*, and *marketing*. Small organizations might not explicitly segment their departments, while larger ones might. Additionally, there may be subdepartments, and subdepartments of those subdepartments as well. You need to know all the departments within your organization, and you need to know which departments use computers and which ones don't. It's fine if a department doesn't use computers (probably rare, but acceptable). This circumstance means there's one less department with which you need to concern yourself. Nevertheless, ensure this department is in your list and that it's not applicable. Your count of the departments must be thorough and accurate, as well as knowing the stakeholders for those departments that put you and your staff on the road to password freedom. Realistically, many of us lose sight of our organizational chart and how it grows or shrinks over time. This realization is why you need to inventory all of them. Also, don't forget to include external departments such as vendors or federated partners. If your organization goes passwordless, but your partners continue to use passwords to access your corporate resources, you should know about it and include them in your passwordless strategy.
Organization or department hierarchy	Organization and department hierarchy is the management layers within the departments or the organization as a whole. How the device is used, what applications and how they're used, most likely differs between each department, but also within the structure of the department. To determine the correct passwordless strategy, you need to know these differences across your organization. An executive leader is likely to use their device differently compared to a member of middle management in the sales department. Both of those user cases are probably different to how an individual contributor in the customer service department uses their device.
Number and type of applications and services	Most organizations have many applications and rarely have one centralized list that's accurate. Applications and services are the most critical items in your passwordless assessment. Applications and services take considerable effort to move to a different type of authentication. Changing policies and procedures can be a daunting task. Consider the trade-off between updating your standard operating procedures and security policies compared to changing 100 lines (or more) of authentication code in the critical path of your internally developed CRM application. Capturing the number of applications used is easier once you have the departments, their hierarchy, and their stakeholders. In this approach, you should have an organized list of departments and the hierarchy in each. You

Size factor	Details
	can now associate the applications that are used by all levels within each department. You also want to document whether the application is internally developed or commercially available off-the-shelf. If the latter, document the manufacturer and the version. Also, don't forget web-based applications or services when inventorying applications.
Number of work personas	Work personas are where the three previous efforts converge. You know the departments, the organizational levels within each department, the numbers of applications used by each, respectively, and the type of application. From this information, you want to create a work persona. A work persona classifies a category of user, title or role (individual contributor, manager, middle manager, etc.), within a specific department to a collection of applications used. There's a high probability that you have many work personas. These work personas will become units of work, and you refer to them in documentation and in meetings. You need to give them a name. Give your personas easy and intuitive names like *Amanda - Accounting*, *Mark - Marketing*, or *Sue - Sales*. If the organization levels are common across departments, then decide on a first name that represents the common levels in a department. For example, *Amanda* could be the first name of an individual contributor in any given department, while the first name *Sue* could represent someone from middle management in any given department. Additionally, you can use suffixes (such as *I*, *II*, *Senior*, etc.) to further define departmental structure for a given persona. Ultimately, create a naming convention that doesn't require your stakeholders and partners to read through a long list of tables or a secret decoder ring. Also, if possible, try to keep the references as names of people. After all, you're talking about a person who is in that department and who uses that specific software.
Organization's IT structure	IT department structures can vary more than the organization. Some IT departments are centralized while others are decentralized. Also, the road to password freedom will probably have you interacting with the *client authentication* team, the *deployment* team, the *security* team, the *PKI* team, the *identity* team, the *cloud* team, etc. Most of these teams are your partner on your journey to password freedom. Ensure there's a passwordless stakeholder on each of these teams, and that the effort is understood and funded.

Assess your organization

By now you can understand why this is a journey and not a quick task. You need to investigate user-visible password surfaces for each of your work personas. Once you've

identified the password surfaces, you need to mitigate them. Resolving some password surfaces are simple - meaning a solution already exists in the environment and it's only a matter of moving users to it. Resolution to some passwords surfaces might exist, but aren't deployed in your environment. That resolution results in a project that must be planned, tested, and then deployed. That project is likely to span multiple IT departments with multiple people, and potentially one or more distributed systems. Those types of projects take time and need dedicated cycles. This same sentiment is true with in-house software development. Even with agile development methodologies, changing the way someone authenticates to an application is critical. Without the proper planning and testing, it has the potential to severely affect productivity.

The time to complete the passwordless journey varies, depending on the organizational alignment to a passwordless strategy. Top-down agreement that a passwordless environment is the organization's goal makes conversations easier. Easier conversations mean less time spent convincing people and more time spent moving toward the goal. Top-down agreement, as a priority within the ranks of other on-going IT projects, helps everyone understand how to prioritize existing projects. Agreeing on priorities should reduce and minimize manager and executive level escalations. After these organizational discussions, modern project management techniques are used to continue the passwordless effort. The organization allocates resources based on the priority (after they agreed on the strategy). Those resources will:

- Work through the work personas
- Organize and deploy user acceptance testing
- Evaluate user acceptance testing results for user visible password surfaces
- Work with stakeholders to create solutions that mitigate user visible password surfaces
- Add the solution to the project backlog and prioritize against other projects
- Deploy the solution
- Perform user acceptance testing to confirm that the solution mitigates the user visible password surface
- Repeat the testing as needed

Your organization's journey to password freedom may take some time. Counting the number of work personas and the number of applications is a good indicator of the investment. Hopefully, your organization is growing, which means that the list of personas and the list of applications is unlikely to shrink. If the work to go passwordless today is *n*, then it's likely that to go passwordless tomorrow is *n x 2* or more, *n x n*. Don't let the size or duration of the project be a distraction. As you progress through each work persona, the actions and tasks become more familiar for you and your stakeholders. Scope the project to sizable, realistic phases, pick the correct work

personas, and soon you'll see parts of your organization transition to a passwordless state.

What's the best guidance for kicking off the journey to password freedom? **You want to show your management a proof of concept as soon as possible**. Ideally, you want to show it at each step of your passwordless journey. Keeping your passwordless strategy top of mind and showing consistent progress keeps everyone focused.

Work persona

You begin with your work personas. These were part of your preparation process. They have a persona name, such as *Amanda - Accounting II*, or any other naming convention your organization defined. That work persona includes a list of all the applications *Amanda* uses to perform her assigned duties in the accounting department. To start, you need to pick a work persona. It's the targeted work persona you enable to complete the journey.

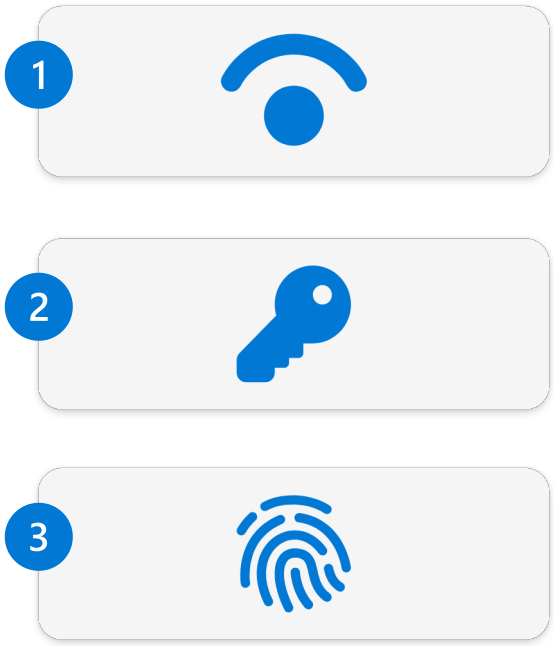
> **Tip**
>
> Avoid using any work personas from your IT department. This method is probably the worst way to start the passwordless journey. IT roles are very difficult and time consuming. IT workers typically have multiple credentials, run a multitude of scripts and custom applications, and are the worst offenders of password usage. It is better to save these work personas for the middle or end of your journey.

Review your collection of work personas. Early in your passwordless journey, identify personas with the fewest applications. These work personas could represent an entire department or two. These roles are the perfect work personas for your proof-of-concept (POC) or pilot.

Most organizations host their POC in a test lab or environment. If you do that test with a password-free strategy, it might be more challenging and take more time. To test in a lab, you must first duplicate the environment of the targeted persona. This process could take a few days or several weeks, depending on the complexity of the targeted work persona.

You want to balance lab testing with providing results to management quickly. Continuing to show forward progress on your journey to password freedom is always a good thing. If there are ways you can test in production with low or no risk, it might be advantageous to your timeline.

The journey to password freedom is to take each work persona through each step of the process. In the beginning, we encourage working with one persona at a time to ensure team members and stakeholders are familiar with the process. Once comfortable with the process, you can cover as many work personas in parallel as resources allow. The process looks something like this:



Deploy a passwordless replacement option

- Identify test users representing the targeted work persona
- Deploy Windows Hello for Business to test users
- Validate that passwords and Windows Hello for Business work

Reduce user-visible password surface

- Survey test user workflow for password usage
- Identify password usage and plan, develop, and deploy password mitigations
- Repeat until all user password usage is mitigated
- Remove password capabilities from Windows
- Validate that **none of the workflows** need passwords

Transition into a passwordless scenario

- Awareness campaign and user education
- Include remaining users who fit the work persona
- Validate that **none of the users** of the work personas need passwords
- Configure user accounts to prevent password authentication

After successfully moving a work persona to password freedom, you can prioritize the remaining work personas and repeat the process.

Next steps

Step 1: deploy a passwordless replacement option >

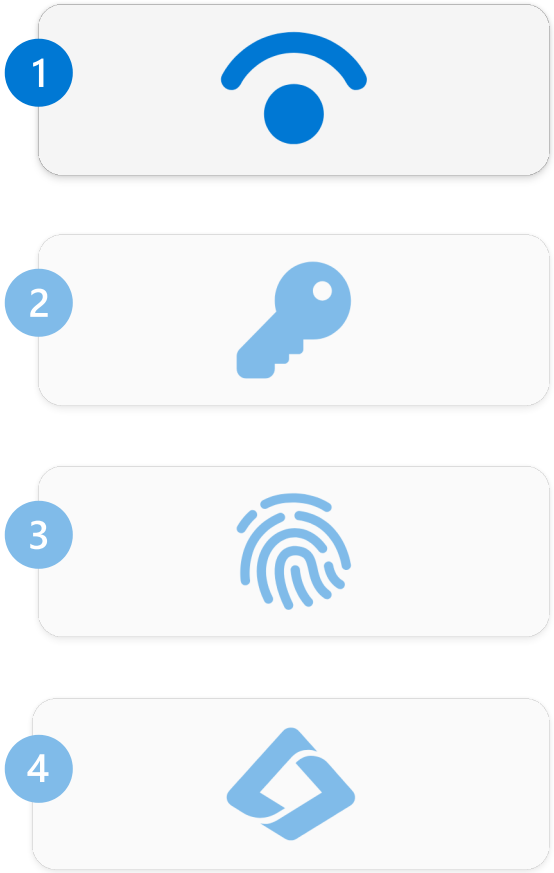
Feedback

Was this page helpful? Yes No

Provide product feedback

Deploy a passwordless replacement option

Article • 10/29/2024 • Applies to: ✅ Windows 11, ✅ Windows 10



The first step to password freedom is providing an alternative to passwords. Windows provides an affordable and easy in-box alternative to passwords, *Windows Hello for Business*. Another option is to use *FIDO2 security keys*, but they require the organization to purchase and distribute them.

Both options provide a strong, two-factor authentication to Microsoft Entra ID and Active Directory.

Identify test users representing the targeted work persona

A successful transition relies on user acceptance testing. It's impossible for you to know how every work persona goes about their day-to-day activities, or how to accurately validate them. You need to enlist the help of users who fit the targeted work persona. You only need a few users from the targeted work persona. As you cycle through step 2,

you might want to change a few of the users (or add a few) as part of your validation process.

Deploy Windows Hello for Business or FIDO2 security keys to test users

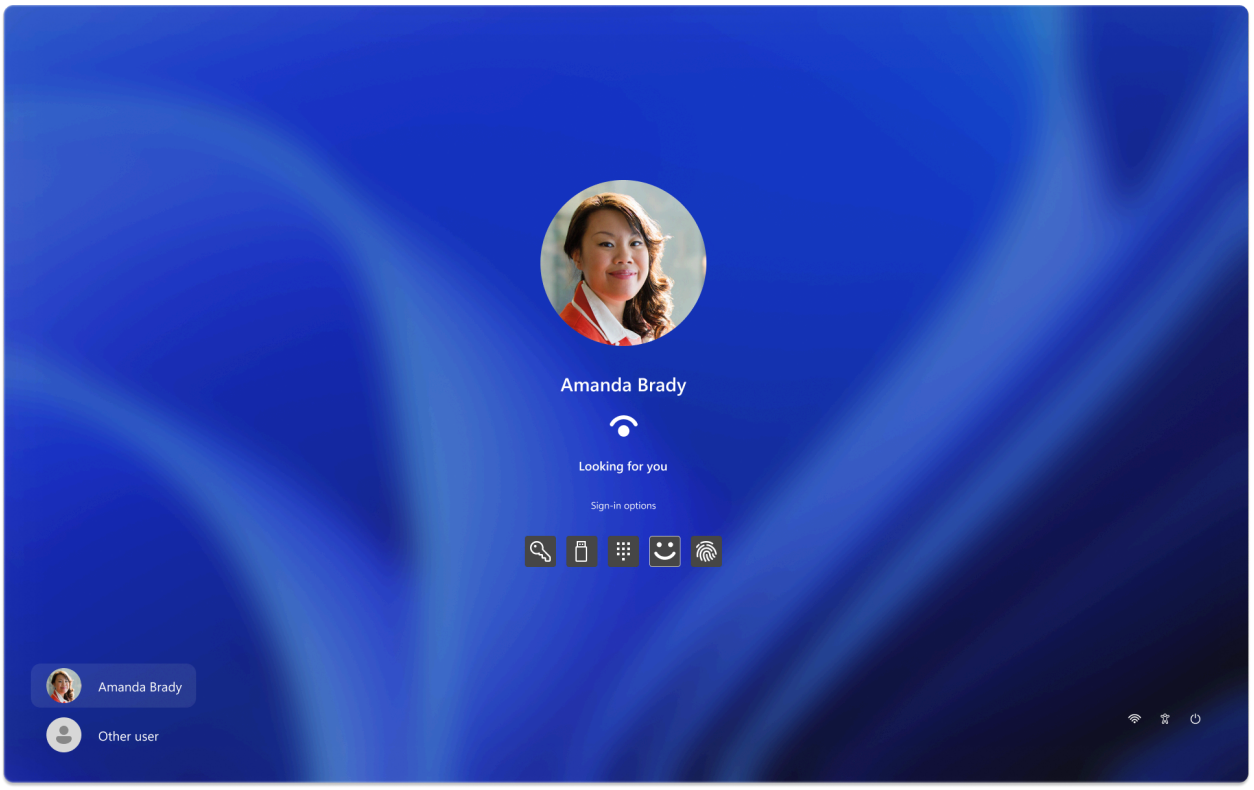
Next, you want to plan your password replacement deployment. Your test users need an alternative way to sign-in during step 2 of the journey to becoming passwordless. Use the Windows Hello for Business planning guide to help learning which deployment is best suited for your environment. Next, use one of the deployment guides to deploy Windows Hello for Business. With the Windows Hello for Business infrastructure in place, you can limit Windows Hello for Business enrollments to the targeted work personas. The great news is that you only need to deploy the infrastructure once. When other targeted work personas need to start using Windows Hello for Business, add them to a group. You use the first work persona to validate your Windows Hello for Business deployment.

If you decide to use FIDO2 security keys, follow the Enable security key sign-in to Windows guide to learn how to adopt FIDO2 security keys.

> **Note**
>
> Deployments vary based on how the device is joined to Microsoft Entra ID. Review the planning guide to learn the type of infrastructure required to support your devices.

Validate passwords and Windows Hello for Business or FIDO2 security keys

In this first step, passwords and your password replacement choice must coexist. You want to validate all scenarios while the targeted work personas can sign in and unlock using Windows Hello or security keys. Users can also sign-in, unlock, and use passwords as needed. Reducing the user-visible password surface too soon can create frustration and confusion with your targeted user personas.



Next steps

Before you move to step 2, make sure you've:

- ✓ Selected your targeted work persona
- ✓ Identified your test users who represent the targeted work persona
- ✓ Deployed Windows Hello for Business or FIDO2 security keys to test users
- ✓ Validated that both your password replacement choice and passwords work for the test users

Step 2: reduce the user-visible password surface area >

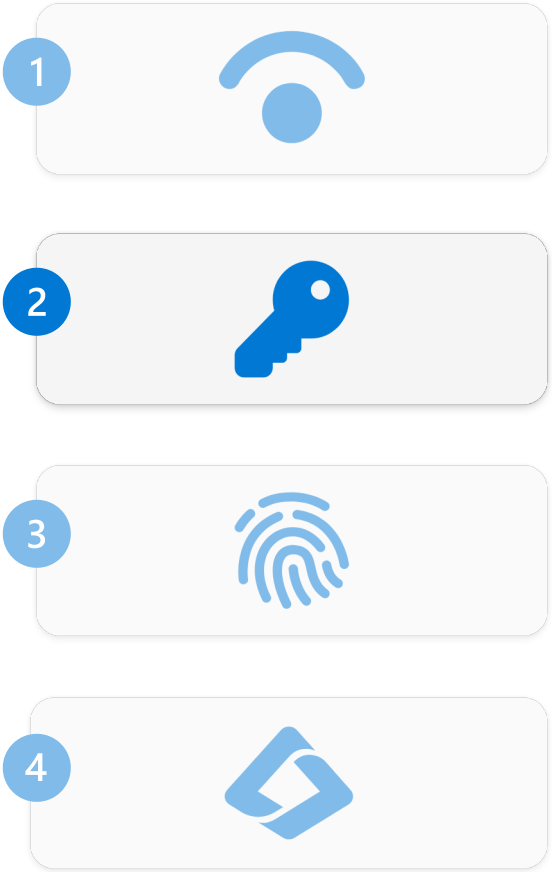
Feedback

Was this page helpful? Yes No

Provide product feedback

Reduce the user-visible password surface area

Article • 10/29/2024 • Applies to: ✅ Windows 11, ✅ Windows 10



Survey test user workflow for password usage

Now is the time to learn more about the targeted work persona. You should have a list of applications they use, but you don't know what, why, when, and how frequently. This information is important as you further your progress through step 2. Test users create the workflows associated with the targeted work persona. Their initial goal is to do one simple task: document password usage. This list isn't a comprehensive one, but it gives you an idea of the type of information you want. The goal is to learn about all the scenarios in which that work persona encounters a password. A good approach is to ask yourself the following set of questions:

Expand table

	Question
☐	*What's the name of the application that asked for a password?*

	Question
☐	*Why do they use the application that asked for a password? For example, is there more than one application that can do the same thing?*
☐	*What part of their workflow makes them use the application? Try to be as specific as possible. For example, "I use application x to issue credit card refunds for amounts over y."*
☐	*How frequently do you use the application in a given day or week?*
☐	*Is the password you type into the application the same as the password you use to sign-in to Windows?*

Some organizations empower their users to write this information, while some might insist on having a member of the IT department shadow them. An objective viewer might notice a password prompt that the user overlooks simply because of muscle memory. As previously mentioned, this information is critical. You could miss one password prompt that could delay the transition to being passwordless.

Identify password usage and plan, develop, and deploy password mitigations

Your test users provided you valuable with information that describes how, what, why, and when they use a password. It's now time for your team to identify each of these password use cases and understand why the user must use a password.
Create a list of the scenarios. Each scenario should have a clear problem statement. Name the scenario with a one-sentence summary of the problem statement. Include in the scenario the results of your team's investigation as to why the user is asked to provide a password. Include relevant, but accurate details. If the scenario is policy or procedure-driven, then include the name and section of the policy that dictates why the workflow uses a password.

Your test users won't uncover all scenarios, therefore you must force on them some uncommon scenarios. Remember to include the following:

- Provision a new user with an unknown password
- Users who forget the PIN or other remediation flows when the strong credential is unusable

Next, review your list of scenarios. You can start with the workflows that are dictated by process or policy, or you can begin with workflows that need technical solutions, whichever of the two is easier or quicker. This choice varies by organization.

Start mitigating password usages based on the workflows of your targeted personas. Document the mitigation as a solution to your scenario. Don't worry about the implementation details for the solution. An overview of the changes needed to reduce the password usages is all you need. If there are technical changes needed, either infrastructure or code changes, the exact details are likely included in the project documentation. However your organization tracks projects, create a new project in that system. Associate your scenario to that project and start the processes needed to get that project funded.

Mitigating password usage with applications is one of the more challenging obstacles in the passwordless journey. If your organization develops the application, then you are in better shape the common-off-the-shelf software (COTS).

The ideal mitigation for applications that prompt the user for a password is to enable those applications to use an existing authenticated identity, such as Microsoft Entra ID or Active Directory. Work with the applications vendors to have them add support for Microsoft Entra identities. For on-premises applications, have the application use Windows integrated authentication. The goal for your users should be a seamless single sign-on experience where each user authenticates once when they sign-in to Windows. Use this same strategy for applications that store their own identities in their own databases.

Each scenario on your list should now have a problem statement, an investigation as to why the password was used, and a mitigation plan on how to make the password usage go away. Armed with this data, one-by-one, close the gaps on user-visible passwords. Change policies and procedures as needed, make infrastructure changes where possible. Convert in-house applications to integrate in your Microsoft Entra ID tenant, use federated identities, or use Windows integrated authentication. Work with non-Microsoft software publishers to update their software to integrate in Microsoft Entra ID, support federated identities, or use Windows integrated authentication.

Repeat until all user password usage is mitigated

Some or all of your mitigations are in place. You need to validate that your solutions solved their problem statements. This stage is where you rely on your test users. You want to keep a good portion of your first test users, but this point is a good opportunity to replace or add a few. Survey test users workflow for password usage. If all goes well, you closed most or all of the gaps. A few are likely to remain. Evaluate your solutions and what went wrong, change your solution as needed until you reach a solution that removes your user's need to type a password. If you're stuck, others might be too. Use

the forums from various sources or your network of IT colleagues to describe your problem and see how others are solving it. If you're out of options, contact Microsoft for assistance.

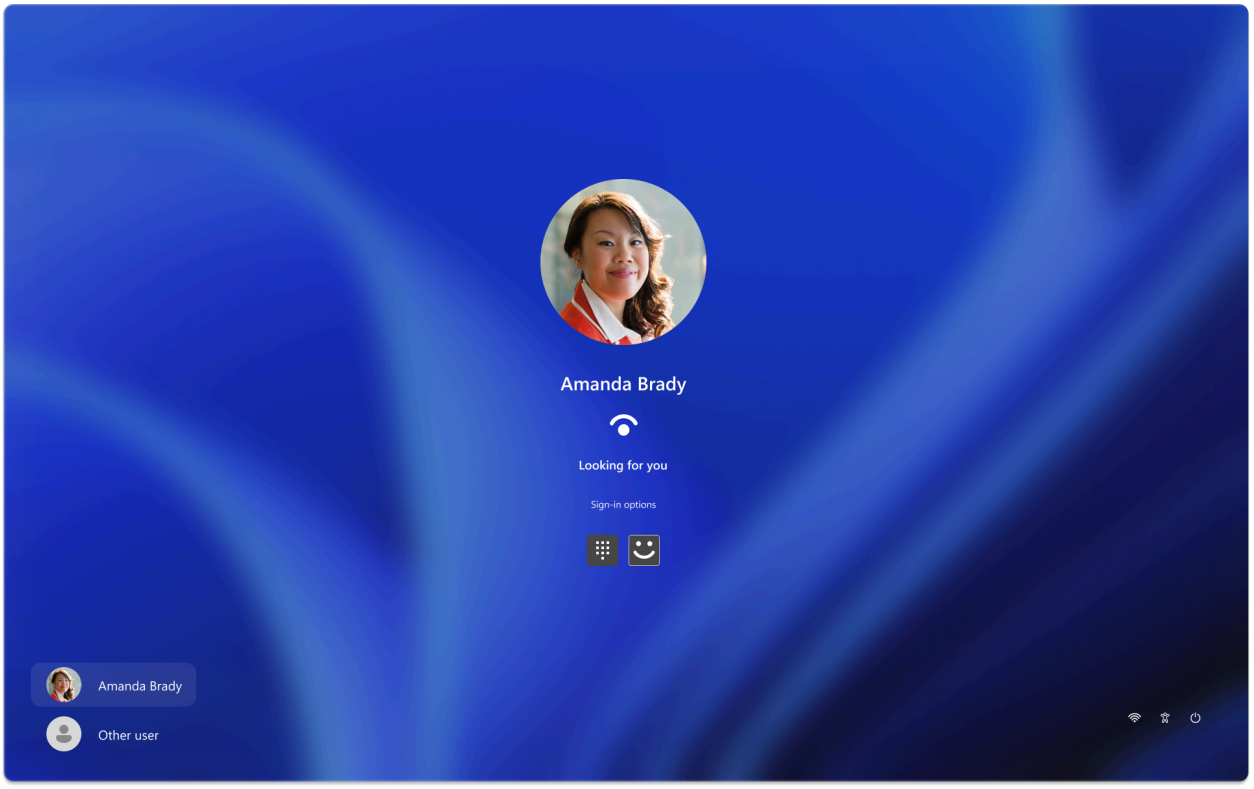
Remove password capabilities from Windows

You believe you mitigated all the password usage for the targeted work persona. Now comes the true test: configure Windows so the user can't use a password.
Windows offers three main options to reduce or eliminate the password surface area:

- Windows passwordless experience
- Exclude the password credential provider
- Require Windows Hello for Business or a smart card

Windows passwordless experience

Windows Passwordless experience is a security policy that hides the password credential provider for user accounts that sign in with Windows Hello or a FIDO2 security key. Windows Passwordless experience is the recommended option, but it's only available on Microsoft Entra joined devices. The following image shows the Windows lock screen when Windows passwordless experience is enabled. A user enrolled in Windows Hello for Business doesn't have the option to use a password to sign in:



To learn more, see Windows passwordless experience

Exclude the password credential provider

The *Exclude credential providers* policy setting can be used to disable the password credential provider. When configured, Windows disables the possibility to use passwords for *all accounts*, including local accounts. It also prevents the use of passwords for RDP and *Run as* authentication scenarios. This policy setting might impact support scenarios, such as when a user needs to sign in with a local account to troubleshoot a problem. For this reason, carefully evaluate all scenarios before you enable the setting.

- GPO: **Computer Configuration** > **Administrative Templates** > **System** > **Logon** > **Exclude credential providers**
- CSP: `./Device/Vendor/MSFT/Policy/Config/ADMX_CredentialProviders/`ExcludedCredentialProviders

The value to enter in the policy to hide the password credential provider is `{60b78e88-ead8-445c-9cfd-0b87f74ea6cd}`.

Require Windows Hello for Business or a smart card

The *Require Windows Hello for Business or a smart card* policy setting can be used to require Windows Hello for Business or a smart card for interactive logon. When enabled, Windows prevents users from signing in or unlocking with a password. The password credential provider remains visible to the user. If a user tries to use a password, Windows informs the user they must use Windows Hello for Business or a smart card. Before you enable this policy setting, the user must be enrolled in Windows Hello for Business or have a smart card. Therefore, implementing this policy requires careful planning and coordination.

- GPO: **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options** > **Interactive logon: Require Windows Hello for Business or smart card**
- CSP: not available

Validate that none of the workflows needs passwords

This stage is the significant moment. You identified password usage, developed solutions to mitigate password usage, and removed or disabled password usage from Windows. In this configuration, your users can't use a password. Users are blocked if any

of their workflows ask them for a password. Ideally, your test users should be able to complete all the work flows of the targeted work persona without any password usage. Don't forget those low percentage work flows, such as provisioning a new user or a user that forgot their PIN or can't use their strong credential. Ensure those scenarios are validated as well.

Next steps

You're ready to transition one or more portions of your organization to a passwordless deployment. You've validated that the targeted work persona is ready to go where the user no longer needs to know or use their password. You're just a few steps away from declaring success.

Step 3: transition into a passwordless deployment >

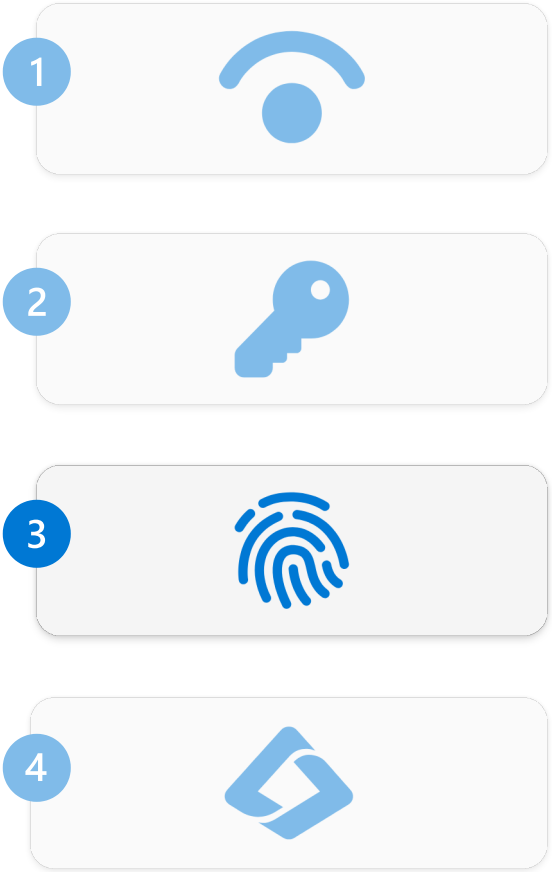
Feedback

Was this page helpful? Yes No

Provide product feedback

Transition into a passwordless deployment

Article • 01/30/2025 • Applies to: ✅ Windows 11, ✅ Windows 10



Awareness and user education

In this last step, you're going to include the remaining users that fit the targeted work persona to the passwordless deployment. Before you do this step, you want to invest in an awareness campaign.

An awareness campaign introduces the users to the new way of authenticating to their device, such as using Windows Hello for Business. The idea of the campaign is to positively promote the change to the users in advance. Explain the value and why your company is changing. The campaign should provide dates and encourage questions and feedback. This campaign can coincide with user education, where you can show the users the changes and, if your environment allows, enable the users to try out the experience.

> **Tip**

To facilitate user communication and to ensure a successful Windows Hello for Business deployment, you can find customizable material (email templates, posters, trainings, etc.) at Microsoft Entra templates.

Include remaining users that fit the work persona

You implemented the awareness campaign for the targeted users. These users are informed and ready to transition to being passwordless. Add the remaining users that match the targeted work persona to your deployment.

Validate that none of the users of the work personas need passwords

You successfully transitioned all users for the targeted work persona to being passwordless. Monitor the users within the work persona to ensure they don't encounter any issues while working in a passwordless environment.

Track all reported issues. Set priority and severity to each reported issue and have your team triage the issues appropriately. As you triage issues, consider the following questions:

Expand table

	Question
☐	*Is the reporting user performing a task outside the work persona?*
☐	*Is the reported issue affecting the entire work persona, or only specific users?*
☐	*Is the outage a result of a misconfiguration?*
☐	*Is the outage an overlooked gap from step 2?*

Each organization's priority and severity differ. However, most organizations consider work stoppages to be fairly significant. Your team should predefine levels of priority and severity. With each of these levels, create service level agreements (SLAs) for each combination of severity and priority, and hold everyone accountable to those agreements. Reactive planning enables people to spend more time on the issue and resolving it, and less time on the process.

Resolve the issues per your service level agreements. Higher severity items might require returning some or all of the user's password surface. Clearly this outcome isn't the end goal, but don't let it slow down your momentum towards becoming passwordless. Refer to how you reduced the user's password surface in step 2, and progress forward to a solution, deploying that solution and validating it.

> **Tip**
>
> Monitor your domain controllers for password authentication events. This helps to proactively identify users who are still using passwords, and to reach out to them.

Configure user accounts to prevent password authentication

You transitioned all the users for the targeted work persona to a passwordless environment and validated all their workflows. The last step to complete the passwordless transition is to remove the user's knowledge of the password.

Password scrambling

While you can't completely remove the password from the user's account, you can prevent the user from using the password to authenticate. The easiest and most effective approach is to set the password to a random value. This approach prevents the user from knowing the password and using it to authenticate, but it allows the user to reset the password whenever needed.

> **Tip**
>
> Enable Microsoft Entra self-service password reset (SSPR) to allow the users to reset their password. Once implemented, users can sign in to their Windows devices using Windows Hello for Business or a FIDO2 security key, and reset their password from https://aka.ms/sspr. Combine it with password writeback to have the password reset synchronized to your on-premises Active Directory.

The following sample PowerShell script generates a random password of 64 characters and sets it for the user specified in the variable name $userId against Microsoft Entra ID. Modify the **userId** variable of the script to match your environment (first line), and then run it in a PowerShell session. When prompted to authenticate to Microsoft Entra ID, use the credentials of an account with a role capable of resetting passwords.

Azure PowerShell

```
$userId = "<UPN of the user>"

function Generate-RandomPassword{
    [CmdletBinding()]
    param (
      [int]$Length = 64
    )
  $chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-
_=+[]{};:,.<>/?\|`~"
  $random = New-Object System.Random
  $password = ""
  for ($i = 0; $i -lt $Length; $i++) {
    $index = $random.Next(0, $chars.Length)
    $password += $chars[$index]
  }
  return $password
}

Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser -Force
Install-Module Microsoft.Graph -Scope CurrentUser
Import-Module Microsoft.Graph.Users.Actions
Connect-MgGraph -Scopes "UserAuthenticationMethod.ReadWrite.All" -NoWelcome

$passwordParams = @{
 UserId = $userId
 AuthenticationMethodId = "28c10230-6103-485e-b985-444c60001490"
 NewPassword = Generate-RandomPassword
}

Reset-MgUserAuthenticationMethodPassword @passwordParams
```

A similar script can be used to reset the password against Active Directory. Modify the **samAccountName** variable of the script to match your environment (first line), and then run it in a PowerShell session.

PowerShell

```
$samAccountName = <sAMAccountName of the user>

function Generate-RandomPassword{
    [CmdletBinding()]
    param (
      [int]$Length = 64
    )
  $chars =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789!@#$%^&*()-
_=+[]{};:,.<>/?\|`~"
  $random = New-Object System.Random
  $password = ""
```

```
  for ($i = 0; $i -lt $Length; $i++) {
    $index = $random.Next(0, $chars.Length)
    $password += $chars[$index]
  }
  return $password
}

$NewPassword = ConvertTo-SecureString -String (Generate-RandomPassword) -
AsPlainText -Force

Set-ADAccountPassword -identity $samAccountName -NewPassword $NewPassword -
Reset
```

If your organizational policies allow it, you can configure the randomized passwords to never expire, or use a long expiration period. This configuration prevents the user from being prompted to change their password.

> ⊗ **Caution**
>
> Execute the script only from a secure and trusted environment, and ensure that the script is not logged. Treat the host where the script is executed as a privileged host, with the same level of security as a domain controller.

Password age and password rotation

If your organization doesn't have password rotation requirements, it's recommended to disable password age.

If your organization has a password rotation policy, consider implementing automation to rotate the user's password regularly. This approach ensures that the user's password is always randomized and prevents the user from knowing the password.

For more password-related guidance, see the whitepaper Password Guidance.

Next steps

Microsoft is working hard to make the passwordless journey easier for you. We're working on new features and capabilities to help you transition to a passwordless environment, and to achieve the long-term security promise of a truly passwordless environment. Check back often to see what's new.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Hello for Business

Applies to: ✅ Windows 11, ✅ Windows 10

Overview

Windows Hello is an authentication technology that allows users to sign in to their Windows devices using biometric data, or a PIN, instead of a traditional password. It provides enhanced security through phish-resistant two-factor authentication, and built-in brute force protection. With FIDO/WebAuthn, Windows Hello can also be used to sign in to supported websites, reducing the need to remember multiple complex passwords.

Windows Hello for Business is an **extension** of Windows Hello that provides enterprise-grade security and management capabilities, including device attestation, certificate-based authentication, and conditional access policies. Policy settings can be deployed to devices to ensure they're secure and compliant with organizational requirements.

The following table lists the main authentication and security differences between Windows Hello and Windows Hello for business:

Expand table

	Windows Hello	Windows Hello for Business
Authentication	Users can authenticate to: - A Microsoft account (MSA) - Identity providers (IdPs) that support Fast ID Online (FIDO) v2.0 authentication	Users can authenticate to: - A Microsoft Entra ID account - An Active Directory account - Identity provider (IdP) or relying party (RP) services that support Fast ID Online (FIDO) v2.0 authentication
Security	It uses **key-based** authentication. There's no symmetric secret (password) which can be stolen from a server or phished from a user and used remotely.	It uses **key-based** or **certificate-based** authentication. There's no symmetric secret (password) which can be stolen from a server or phished from a user and used remotely.

Windows Hello can also be used with local accounts for convenient sign-ins, instead of entering a password. This configuration isn't backed by asymmetric (public/private) key, so it doesn't offer the same level of security as key-based or certificate-based authentication that is available with MSA or Microsoft Entra accounts. In all other aspects, using Windows Hello with a local account is like using it with MSA or Entra ID. For enhanced security, it's recommended to use Windows Hello with a Microsoft account (MSA) or identity providers (IdPs) that support FIDO2 authentication.

> **Note**
>
> FIDO2 (Fast Identity Online) authentication is an open standard for passwordless authentication. It allows users to sign in to their devices and apps using biometric authentication or a physical security key, without the need for a traditional password. FIDO2 support in Windows Hello and Windows Hello for Business provides an additional layer of security and convenience for users, while also reducing the risk of password-related attacks.

Benefits

Windows Hello for Business provides many benefits, including:

- It helps to strengthen protections against credential theft. An attacker must have both the device and the biometric or PIN, making it much more difficult to gain access without the user's knowledge
- Since no passwords are used, it circumvents phishing and brute force attacks. Most importantly, it prevents server breaches and replay attacks because the credentials are asymmetric and generated within isolated environments of TPMs
- Users get a simple and convenient authentication method (backed up with a PIN) that's always with them, so there's nothing to lose. The use of a PIN doesn't compromise security, since Windows Hello has built-in brute force protection, and the PIN never leaves the device
- You can add biometric devices as part of a coordinated rollout or to specific users, as needed

The following video shows a demonstration of Windows Hello for Business in action, where a user signs in with a fingerprint:

https://learn-video.azurefd.net/vod/player?id=fb5ceb53-d82b-4997-bde1-d473b620038a&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2F

Windows Hello and two factor authentication

Windows Hello for Business uses a two-factor authentication method that combines a device-specific credential with a biometric or PIN gesture. This credential is tied to your identity provider, such as Microsoft Entra ID or Active Directory, and can be used to access organization apps, websites, and services.

After an initial two-step verification of the user during provisioning, Windows Hello is set up on the user's device and Windows asks the user to set a gesture, which can be a biometric, and a PIN. The user provides the gesture to verify their identity. Windows then uses Windows Hello to authenticate users.

Windows Hello for Business is considered two-factor authentication based on the observed authentication factors of: *something you have*, *something you know*, and *something that's part of you*. Windows Hello for Business incorporates two of these factors: something you have (the user's private key protected by the device's security module) and something you know (your PIN). With the proper hardware, you can enhance the user experience by introducing biometrics. By using biometrics, you can replace the *something you know* authentication factor with the *something that is part of you* factor, with the assurances that users can fall back to the *something you know factor*.

Biometric sign-in

Windows Hello provides reliable, fully integrated biometric authentication based on facial recognition or fingerprint matching. Windows Hello uses a combination of special infrared (IR) cameras and software to increase accuracy and guard against spoofing. Major hardware vendors are shipping devices that have integrated Windows Hello-compatible cameras and fingerprint readers.

On devices that support Windows Hello, an easy biometric gesture unlocks users' credentials:

- **Facial recognition**: this type of biometric recognition uses special cameras that see in IR light, which allows them to reliably tell the difference between a photograph or scan and a living person. Several vendors offer external cameras that incorporate this technology, and many laptop manufacturers incorporate it into their devices
- **Fingerprint recognition**: this type of biometric recognition uses a capacitive fingerprint sensor to scan your fingerprint. Most existing fingerprint readers work with Windows, whether they're external or integrated into laptops or USB keyboards
- **Iris Recognition**: this type of biometric recognition uses cameras to perform scan of your iris

Windows stores biometric data that is used to implement Windows Hello securely on the local device only. The biometric data doesn't roam and is never sent to external devices or servers. Because Windows Hello only stores biometric identification data on the device, there's no single collection point an attacker can compromise to steal biometric data.

Windows edition and licensing requirements

The following table lists the Windows editions that support Windows Hello for Business:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Windows Hello for Business license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

> **Note**
>
> Windows Hello for Business doesn't work with Microsoft Entra Domain Services.

Hardware requirements

Microsoft collaborates with manufacturers to help ensuring a high-level of performance and protection is met by each sensor and device, based on the following requirements:

- **False Accept Rate (FAR):** represents the instance a biometric identification solution verifies an unauthorized person. This is normally represented as a ratio of number of instances in a given population size, for example 1 in 100,000. This can also be represented as a percentage of occurrence, for example, 0.001%. This measurement is heavily considered the most important regarding the security of the biometric algorithm
- **False Reject Rate (FRR):** represents the instances a biometric identification solution fails to verify an authorized person correctly. Represented as a percentage, the sum of the True Accept Rate and False Reject Rate is 1. Can be with or without anti-spoofing or liveness detection

Fingerprint sensor requirements

To allow fingerprint matching, devices must have fingerprint sensors and software. Fingerprint sensors can be touch sensors (large area or small area) or swipe sensors. Each type of sensor

has its own set of detailed requirements that must be implemented by the manufacturer, but all of the sensors must include anti-spoofing measures.

Acceptable performance range for small to large size touch sensors:

- False Accept Rate (FAR): <0.001 - 0.002%
- Effective, real world FRR with Anti-spoofing or liveness detection: <10%

Acceptable performance range for swipe sensors:

- False Accept Rate (FAR): <0.002%
- Effective, real world FRR with Anti-spoofing or liveness detection: <10%

Facial recognition sensors

To allow facial recognition, you must have devices with integrated special infrared (IR) sensors and software. Facial recognition sensors use special cameras that see in IR light, letting them tell the difference between a photo and a living person while scanning an employee's facial features. These sensors, like the fingerprint sensors, must also include anti-spoofing measures (required) and a way to configure them (optional).

- False Accept Rate (FAR): <0.001%
- False Reject Rate (FRR) without Anti-spoofing or liveness detection: <5%
- Effective, real world FRR with Anti-spoofing or liveness detection: <10%

> **Note**
>
> Windows Hello face authentication doesn't support wearing a mask during enrollment or authentication. If your working environment doesn't allow you to remove a mask temporarily, consider using PIN or fingerprint.

Iris recognition sensor requirements

To use Iris authentication, you need a HoloLens 2 device. All HoloLens 2 editions are equipped with the same sensors. Iris is implemented the same way as other Windows Hello technologies and achieves biometrics security FAR of 1/100K.

For more information about the hardware requirements for Windows Hello, see Windows Hello biometric requirements.

Next steps

Learn how Windows Hello for Business works >

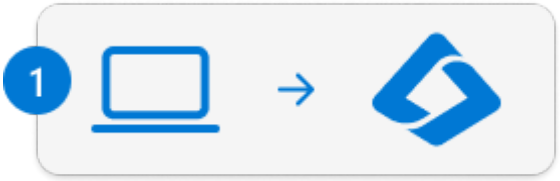
Last updated on 03/29/2026

How Windows Hello for Business works

Applies to: ✅ Windows 11, ✅ Windows 10

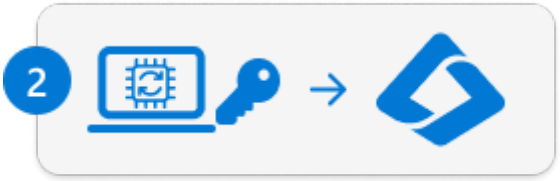
Windows Hello for Business is a distributed system that requires multiple technologies to work together. To simplify the explanation of how Windows Hello for Business works, let's break it down into five phases, which represent the chronological order of the deployment process.

> ⓘ **Note**
>
> Two of these phases are required only for certain deployment scenarios.
>
> The deployment scenarios are described in the article: Plan a Windows Hello for Business deployment.



Device registration phase

In this phase, the device registers its identity with the identity provider (IdP), so that it can be associated and authenticate to the IdP.



Provisioning phase

During this phase, the user authenticates using one form of authentication (typically, username/password) to request a new Windows Hello for Business credential. The provisioning flow requires a second factor of authentication before it can generate a public/private key pair. The public key is registered with the IdP, mapped to the user account.



Key synchronization phase

In this phase, **required by some hybrid deployments**, the user's public key is synchronized from Microsoft Entra ID to Active Directory.



Certificate enrollment phase

In this phase, **required only by deployments using certificates**, a certificate is issued to the user using the organization's public key infrastructure (PKI).



Authentication phase

In this last phase, the user can sign-in to Windows using biometrics or a PIN. Regardless of the gesture used, authentication occurs using the private portion of the Windows Hello for Business credential. The IdP validates the user identity by mapping the user account to the public key registered during the provisioning phase.

The following sections provide deeper insights into each of these phases.

Device Registration

All devices included in the Windows Hello for Business deployment must go through a process called *device registration*. Device registration enables devices to be associated and to authenticate to an IdP:

- For cloud and hybrid deployments, the identity provider is Microsoft Entra ID, and the device registers with the *Device Registration Service*
- For on-premises deployments, the identity provider is Active Directory Federation Services (AD FS), and the device registers with the *Enterprise Device Registration Service* hosted on AD FS

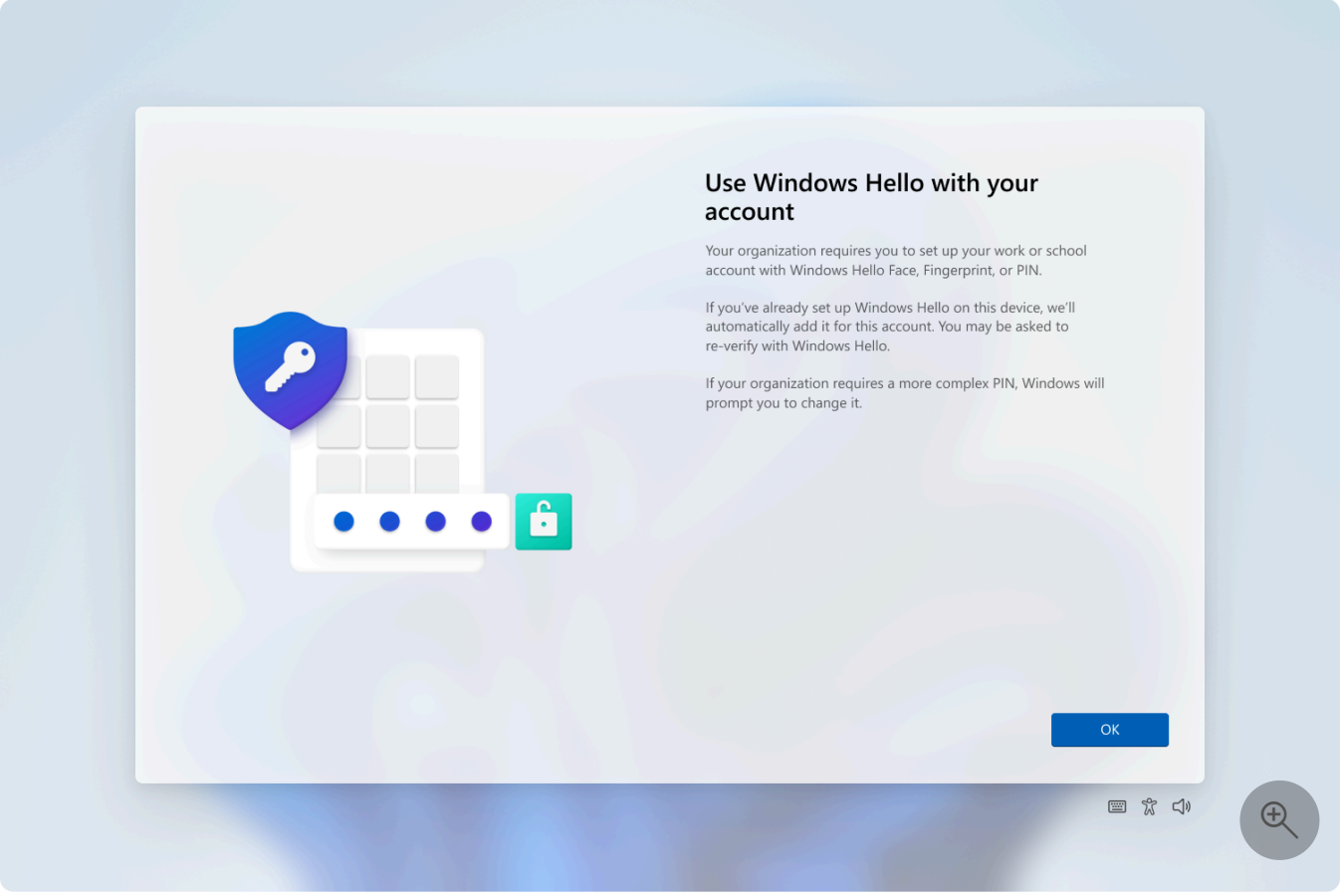
When a device is registered, the IdP provides the device with an identity that is used to authenticate the device when a user signs in.

There are different registration types, which are identified as *join type*. For more information, see What is a device identity.

For detailed sequence diagrams, see how device registration works.

Provisioning

Windows Hello provisioning is triggered once device registration completes, and after the device receives a policy that enables Windows Hello. If all the prerequisites are met, a Cloud eXperience Host (CXH) window is launched to take the user through the provisioning flow.



> **Note**

Depending on the deployment type, Windows Hello for Business provisioning is launched only if:

- The device meets the Windows Hello hardware requirements
- The device is joined to Active Directory or Microsoft Entra ID
- The user signs in with an account defined in Active Directory or Microsoft Entra ID
- The Windows Hello for Business policy is enabled
- The user is not connected to the machine via Remote Desktop

Additional prerequisites for specific deployment types are described in the article **Plan a Windows Hello for Business deployment**.

During the provisioning phase, a *Windows Hello container* is created. A Windows Hello container is a logical grouping of *key material*, or data. The container holds organization's credentials only on devices that are *registered* with the organization's IdP.

> **Note**
>
> There are no physical containers on disk, in the registry, or elsewhere. Containers are logical units used to group related items. The keys, certificates, and credentials that Windows Hello stores, are protected without the creation of actual containers or folders.

Here are the steps involved with the provisioning phase:

1. In the CXH window, the user is prompted to authenticate to the IdP with MFA
2. After successful MFA, the user must provide a bio gesture (if available), and a PIN
3. After the PIN confirmation, the Windows Hello container is created
4. A public/private key pair is generated. The key pair is bound to the Trusted Platform Module (TPM), if available, or in software
5. The private key is stored locally and protected by the TPM, and can't be exported
6. The public key is registered with the IdP, mapped to the user account
   a. The Device Registration Service writes the key to the user object in Microsoft Entra ID
   b. For on-premises scenarios, AD FS writes the key to Active Directory

The following video shows the Windows Hello for Business enrollment steps after signing in with a password:

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fhow-it-works

For more information and detailed sequence diagrams, see how provisioning works.

Windows Hello container details

During the provisioning phase, Windows Hello generates a new public/private key pair on the device. The TPM generates and protects the private key. If the device doesn't have a TPM, the private key is encrypted and stored in software. This initial key is referred to as the *protector key*. The protector key is associated with a single gesture: if a user registers a PIN, a fingerprint, and a face on the same device, each of those gestures has a unique protector key.

The protector key securely wraps the *authentication key*. The authentication key is used to unlock the *user ID keys*. The container has only one authentication key, but there can be multiple copies of that key wrapped with different unique protector keys.



Each protector encrypts its own copy of the authentication key. How the encryption is performed is up to the protector itself. For example, the PIN protector performs a TPM seal

operation using the PIN as entropy, or when no TPM is available, performs symmetric encryption of the authentication key using a key derived from the PIN itself.

> **Important**
>
> Keys can be generated in hardware (TPM 1.2 or 2.0) or software, based on the configured policy setting. To guarantee that keys are generated in hardware, you must configure a policy setting. For more information, see Use a hardware security device.

Personal (Microsoft account) and Work or School (Active Directory or Microsoft Entra ID) accounts use a single container for keys. All keys are separated by identity providers' domains to help ensure user privacy.

Windows Hello also generates an *administrative key*. The administrative key can be used to reset credentials when necessary. For example, when using the PIN reset service. In addition to the protector key, TPM-enabled devices generate a block of data that contains attestations from the TPM.

Access to the key material stored in the container, is enabled only by the PIN or biometric gesture. The two-step verification that takes place during provisioning creates a trusted relationship between the IdP and the user. This happens when the public portion of the public/private key pair is sent to an identity provider and associated with the user account. When a user enters the gesture on the device, the identity provider knows that it's a verified identity, because of the combination of Windows Hello keys and gestures. It then provides an authentication token that allows Windows to access resources and services.

A container can contain several types of key material:

- An *authentication key*, which is always an asymmetric public-private key pair. This key pair is generated during registration. It must be unlocked each time it's accessed, by using either the user's PIN or a biometric gesture. The authentication key exists until the user resets the PIN, at which time a new key is generated. When the new key is generated, all the key material that the old key previously protected must be decrypted and re-encrypted using the new key
- One or multiple *user ID keys*. These keys can be either symmetric or asymmetric, depending on which IdP you use. For certificate-based Windows Hello for Work, when the container is unlocked, applications that require access to the user ID key or key pair can request access. User ID keys are used to sign or encrypt authentication requests or tokens sent from this device to the IdP. User ID keys are typically long-lived but could have a shorter lifetime than the authentication key. Microsoft accounts, Active Directory accounts, and Microsoft Entra accounts all require the use of asymmetric key pairs. The device generates public and private keys, registers the public key with the IdP (which

stores it for later verification), and securely stores the private key. For organizations, the user ID keys can be generated in two ways:

- The user ID key pair can be associated with an organization's Certificate Authority (CA). This option lets organizations that have an existing PKI continue to use it where appropriate. Given that many applications, such as VPN solutions, require the use of certificates, when you deploy Windows Hello in this mode, it allows a faster transition away from user passwords while still preserving certificate-based functionality. This option also allows the organization to store other certificates in the protected container. For example, certificates that allow the user to authenticate via RDP
- The IdP can generate the user ID key pair directly, which allows quick, lower-overhead deployment of Windows Hello in environments that don't have or need a PKI

User ID keys are used to authenticate the user to a service. For example, by signing a nonce to prove possession of the private key, which corresponds to a registered public key. Users with an Active Directory, Microsoft Entra ID or Microsoft account have a key associated with their account. The key can be used to sign into their Windows device by authenticating to a domain controller (Active Directory scenario), or to the cloud (Microsoft Entra ID and MSA scenarios).

Windows Hello can also be used as a FIDO2 authenticator to authenticate to any website that supports WebAuthn. Websites or application can create a FIDO user ID key in the user's Windows Hello container using APIs. On subsequent visits, the user can authenticate to the website or app using their Windows Hello PIN or biometric gesture.

To learn more how Windows uses the TPM in support of Windows Hello for Business, see How Windows uses the Trusted Platform Module.

Biometric data storage

The biometric data used to support Windows Hello is stored on the local device only. It doesn't roam and is never sent to external devices or servers. This separation helps to stop potential attackers by providing no single collection point that an attacker could potentially compromise to steal biometric data. Even if an attacker could obtain the biometric data from a device, it couldn't be converted back into a raw biometric sample recognizable by the biometric sensor.

Each sensor has its own biometric database file where template data is stored (path `C:\WINDOWS\System32\WinBioDatabase`). Each database file has a unique, randomly generated key that is encrypted to the system. The template data for the sensor is encrypted with the per-database key using AES with CBC chaining mode. The hash is SHA256.

> **Note**

Some fingerprint sensors have the capability to complete matching on the fingerprint sensor module instead of in the OS. These sensors store biometric data on the fingerprint module instead of in the database file. For more information, see **Windows Hello Enhanced Security Sign-in (ESS)**.

Key synchronization

Key synchronization is required in hybrid environments. After the user provisions a Windows Hello for Business credential, the key must synchronize from Microsoft Entra ID to Active Directory.

The user's public key is written to the `msDS-KeyCredentialLink` attribute of the user object in Active Directory. The synchronization is handled by Microsoft Entra Connect Sync.

Certificate enrollment

For certificate deployments, after registering the key, the client generates a certificate request. The request is sent to the Certificate Registration Authority (CRA). The CRA is on the Active Directory Federation Services (AD FS) server, which validates the certificate request and fulfills it using the enterprise PKI.

A certificate is enrolled on the user's Hello container, which is used to authenticate to on-premises resources.

Authentication

Windows Hello credentials are based on certificate or asymmetrical key pair. Windows Hello credentials, and the token that is obtained using those credentials, are bound to the device.

Authentication is the two-factor authentication with the combination of:

- A key, or certificate, tied to a device and
  - something that the person knows (a PIN) or
  - something that the person is (biometrics)

PIN entry and biometric gesture both trigger Windows to use the private key to cryptographically sign data that is sent to the identity provider. The IdP verifies the user's identity and authenticates the user.

The PIN or the private portion of the credentials is never sent to the IdP, and the PIN isn't stored on the device. The PIN and bio gestures are *user-provided entropy* when performing

operations that use the private portion of the credential.

When a user wants to access protected key material, the authentication process begins with the user entering a PIN or biometric gesture to unlock the device, a process sometimes called *releasing the key*. Think of it like using a physical key to unlock a door: before you can unlock the door, you need to remove the key from your pocket or purse. The user's PIN unlocks the protector key for the container on the device. When that container is unlocked, applications (and thus the user) can use whatever User ID keys reside inside the container.

These keys are used to sign requests that are sent to the IdP, requesting access to specified resources.

> **Important**
>
> Although the keys are unlocked, applications cannot use them at will. Applications can use specific APIs to request operations that require key material for particular actions (for example, decrypt an email message or sign in to a website). Access through these APIs doesn't require explicit validation through a user gesture, and the key material isn't exposed to the requesting application. Rather, the application asks for authentication, encryption, or decryption, and the Windows Hello layer handles the actual work and returns the results. Where appropriate, an application can request a forced authentication even on an unlocked device. Windows prompts the user to reenter the PIN or perform an authentication gesture, which adds an extra level of protection for sensitive data or actions. For example, you can configure an application to require re-authentication anytime a specific operation is performed, even though the same account and PIN or gesture were already used to unlock the device.

For more information and detailed sequence diagrams, see how authentication works.

Primary refresh token

Single sign-on (SSO) relies on special tokens obtained to access specific applications. In the traditional Windows Integrated authentication case using Kerberos, the token is a Kerberos TGT (ticket-granting ticket). For Microsoft Entra ID and AD FS applications, this token is a *primary refresh token* (PRT). It's a JSON Web Token that contains claims about both the user and the device.

The PRT is initially obtained during sign-in or unlock in a similar way the Kerberos TGT is obtained. This behavior is true for both Microsoft Entra joined and Microsoft Entra hybrid joined devices. For personal devices registered with Microsoft Entra ID, the PRT is initially

obtained upon *Add Work or School Account*. For a personal device, the account to unlock the device isn't the work account, but a consumer account (*Microsoft account*).

The PRT is needed for SSO. Without it, users would be prompted for credentials every time they access applications. The PRT also contains information about the device. If you have any device-based conditional access policies set on an application, without the PRT access is denied.

> **Tip**
>
> The Windows Hello for Business key meets Microsoft Entra multifactor authentication (MFA) requirements and reduces the number of MFA prompts users will see when accessing resources.

For more information, see What is a Primary Refresh Token.

Windows Hello for Business and password changes

Changing a user account password doesn't affect sign-in or unlock, since Windows Hello for Business uses a key or certificate.

However, when users are required to change their password (for example, due to password expiration policies), then they won't be notified of the password change requirement when signing in with Windows Hello. This might cause failures to authenticate to Active Directory-protected resources. To mitigate the issue consider one of the following options:

- Disable password expiration for the user accounts
- As an alternative to password expiration policies, consider adopting PIN expiration policies
- If password expiration is an organization's requirement, instruct the users to change their passwords regularly or when they receive authentication failure messages. Users can reset their password by:
  - Using the `Ctrl` + `Alt` + `Del` > **Change a password** option
  - Sign in with their password. If the password must be changed, Windows prompts the user to update it

> **Important**
>
> To change a user's password, the device must be able to communicate with a domain controller.

Next steps

To accommodate the multitude of organizations needs and requirements, Windows Hello for Business offers different deployment options. To learn how to plan a Windows Hello for Business deployment, see:

Plan a Windows Hello for Business Deployment

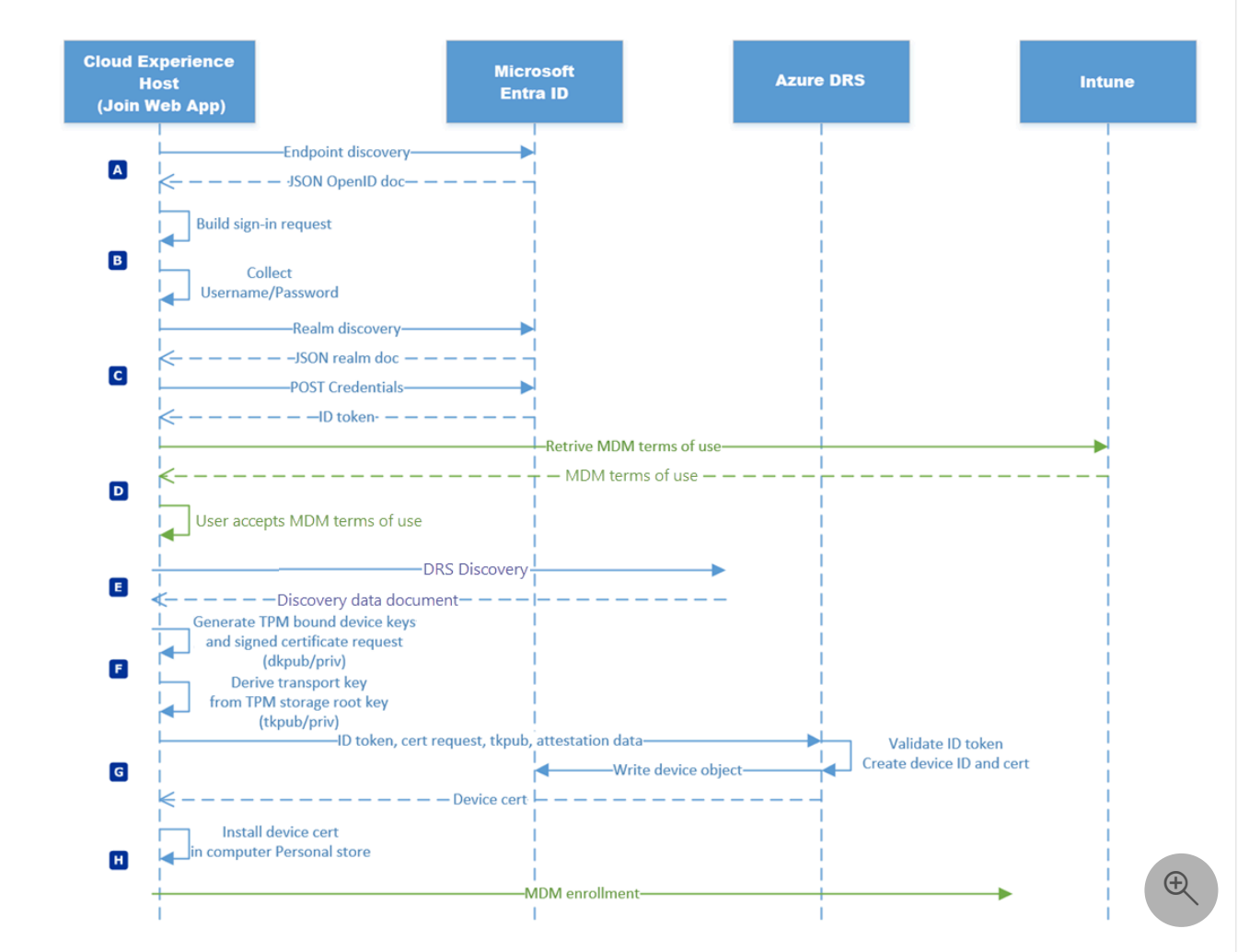
Last updated on 03/29/2026

How it works: Device registration

06/27/2025

Device Registration is a prerequisite to cloud-based authentication. Commonly, devices are Microsoft Entra ID or Microsoft Entra hybrid joined to complete device registration. This article provides details of how Microsoft Entra join and Microsoft Entra hybrid join work in managed and federated environments. For more information about how Microsoft Entra authentication works on these devices, see the article Primary refresh tokens.

Microsoft Entra joined in Managed environments

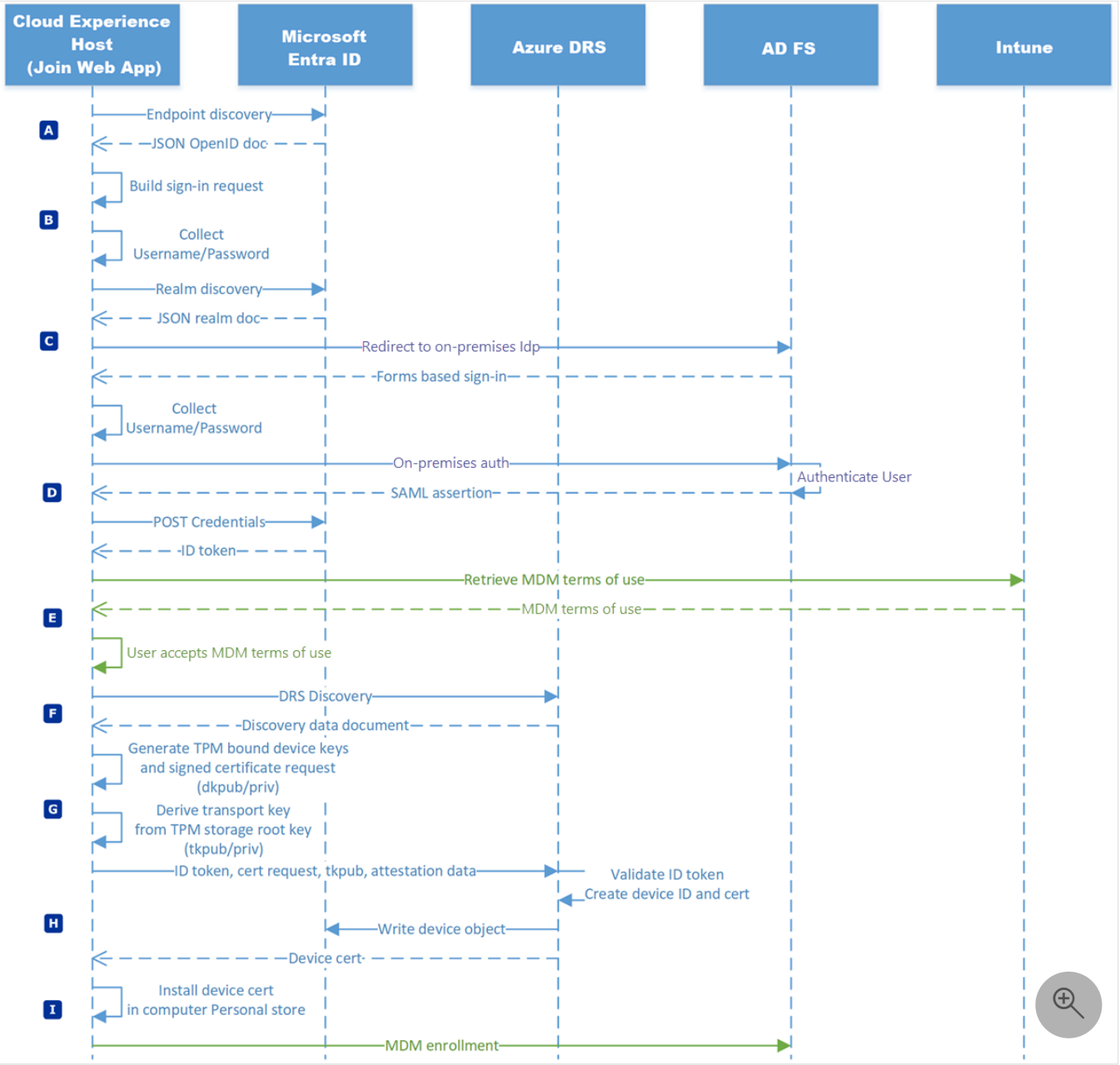


Expand table

Phase	Description
A	The most common way Microsoft Entra joined devices register is during the out-of-box-experience (OOBE) where it loads the Microsoft Entra join web application in the Cloud Experience Host (CXH) application. The application sends a GET request to the Microsoft Entra OpenID configuration endpoint to discover authorization endpoints. Microsoft Entra ID returns

Phase	Description
	the OpenID configuration, which includes the authorization endpoints, to application as JSON document.
B	The application builds a sign-in request for the authorization end point and collects user credentials.
C	After the user provides their user principal name (UPN), the application sends a GET request to Microsoft Entra ID to discover corresponding realm information for the user. This information determines if the environment is managed or federated. Microsoft Entra ID returns the information in a JSON object. The application determines the environment is managed (nonfederated). The last step in this phase has the application create an authentication buffer and if in OOBE, temporarily caches it for automatic sign-in at the end of OOBE. The application POSTs the credentials to Microsoft Entra ID where they're validated. Microsoft Entra ID returns an ID token with claims.
D	The application looks for mobile device management (MDM) terms of use (the mdm_tou_url claim). If present, the application retrieves the terms of use from the claim's value, present the contents to the user, and waits for the user to accept the terms of use. This step is optional and skipped if the claim isn't present or if the claim value is empty.
E	The application sends a device registration discovery request to the Azure Device Registration Service (DRS). Azure DRS returns a discovery data document, which returns tenant-specific URIs to complete device registration.
F	The application creates TPM bound (preferred) RSA 2048 bit key-pair known as the device key (dkpub/dkpriv). The application creates a certificate request using dkpub and the public key and signs the certificate request with using dkpriv. Next, the application derives second key pair from the TPM's storage root key. This key is the transport key (tkpub/tkpriv).
G	The application sends a device registration request to Azure DRS that includes the ID token, certificate request, tkpub, and attestation data. Azure DRS validates the ID token, creates a device ID, and creates a certificate based on the included certificate request. Azure DRS then writes a device object in Microsoft Entra ID and sends the device ID and the device certificate to the client.
H	Device registration completes by receiving the device ID and the device certificate from Azure DRS. The device ID is saved for future reference (viewable from `dsregcmd.exe /status`), and the device certificate is installed in the Personal store of the computer. With device registration complete, the process continues with MDM enrollment.

Microsoft Entra joined in Federated environments

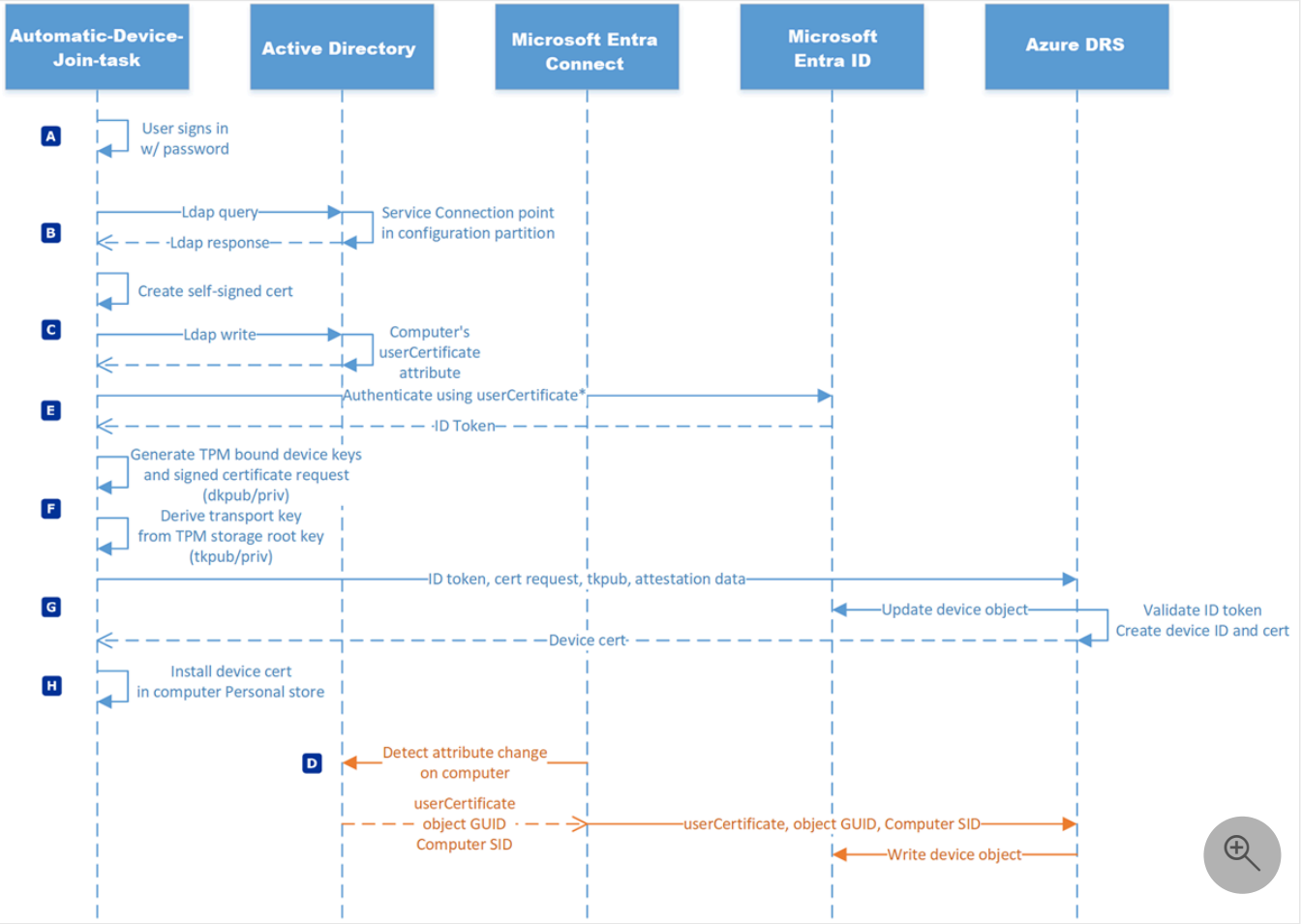


Expand table

Phase	Description
A	The most common way Microsoft Entra joined devices register is during the out-of-box-experience (OOBE) where it loads the Microsoft Entra join web application in the Cloud Experience Host (CXH) application. The application sends a GET request to the Microsoft Entra OpenID configuration endpoint to discover authorization endpoints. Microsoft Entra ID returns the OpenID configuration, which includes the authorization endpoints, to application as JSON document.
B	The application builds a sign-in request for the authorization end point and collects user credentials.
C	After the user provides their user name (in UPN format), the application sends a GET request to Microsoft Entra ID to discover corresponding realm information for the user. This information determines if the environment is managed or federated. Microsoft Entra ID returns the information in a JSON object. The application determines the environment is federated.

Phase	Description
	The application redirects to the AuthURL value (on-premises STS sign-in page) in the returned JSON realm object. The application collects credentials through the STS web page.
D	The application POST the credential to the on-premises STS, which might require extra factors of authentication. The on-premises STS authenticates the user and returns a token. The application POSTs the token to Microsoft Entra ID for authentication. Microsoft Entra ID validates the token and returns an ID token with claims.
E	The application looks for MDM terms of use (the mdm_tou_url claim). If present, the application retrieves the terms of use from the claim's value, present the contents to the user, and waits for the user to accept the terms of use. This step is optional and skipped if the claim isn't present or if the claim value is empty.
F	The application sends a device registration discovery request to the Azure Device Registration Service (DRS). Azure DRS returns a discovery data document, which returns tenant-specific URIs to complete device registration.
G	The application creates TPM bound (preferred) RSA 2048 bit key-pair known as the device key (dkpub/dkpriv). The application creates a certificate request using dkpub and the public key and signs the certificate request with using dkpriv. Next, the application derives second key pair from the TPM's storage root key. This key is the transport key (tkpub/tkpriv).
H	The application sends a device registration request to Azure DRS that includes the ID token, certificate request, tkpub, and attestation data. Azure DRS validates the ID token, creates a device ID, and creates a certificate based on the included certificate request. Azure DRS then writes a device object in Microsoft Entra ID and sends the device ID and the device certificate to the client.
I	Device registration completes by receiving the device ID and the device certificate from Azure DRS. The device ID is saved for future reference (viewable from `dsregcmd.exe /status`), and the device certificate is installed in the Personal store of the computer. With device registration complete, the process continues with MDM enrollment.

Microsoft Entra hybrid joined in Managed environments

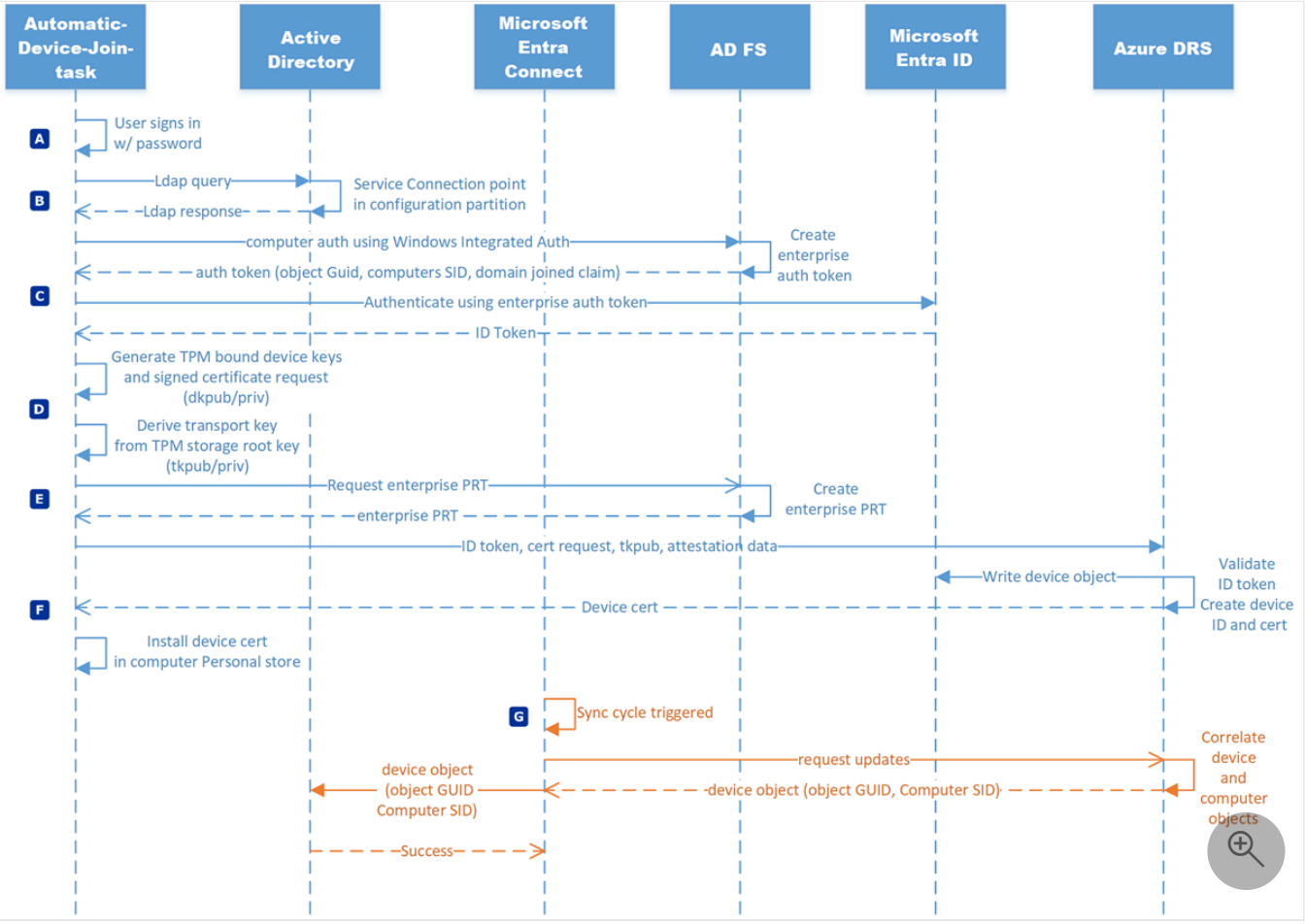


Expand table

Phase	Description
A	The user signs in to a domain joined Windows 10 or newer computer using domain credentials. This credential can be user name and password or smart card authentication. The user sign-in triggers the Automatic Device Join task. The Automatic Device Join tasks is triggered on domain join and retried every hour. It doesn't solely depend on the user sign-in.
B	The task queries Active Directory using the LDAP protocol for the keywords attribute on the service connection point stored in the configuration partition in Active Directory (`CN=62a0ff2e-97b9-4513-943f-0d221bd30080,CN=Device Registration Configuration,CN=Services,CN=Configuration,DC=corp,DC=contoso,DC=com`). The value returned in the keywords attribute determines if device registration is directed to Azure Device Registration Service (DRS) or the enterprise device registration service hosted on-premises.
C	For the managed environment, the task creates an initial authentication credential in the form of a self-signed certificate. The task writes the certificate to the userCertificate attribute on the computer object in Active Directory using LDAP.
D	The computer can't authenticate to Azure DRS until a device object representing the computer that includes the certificate on the userCertificate attribute is created in Microsoft Entra ID. Microsoft Entra Connect detects an attribute change. On the next synchronization cycle, Microsoft Entra Connect sends the userCertificate, object GUID, and computer SID to Azure DRS. Azure DRS uses the attribute information to create a device object in Microsoft Entra ID.

Phase	Description
E	The Automatic Device Join task triggers with each user sign-in or every hour, and tries to authenticate the computer to Microsoft Entra ID using the corresponding private key of the public key in the userCertificate attribute. Microsoft Entra authenticates the computer and issues an ID token to the computer.
F	The task creates TPM bound (preferred) RSA 2048 bit key-pair known as the device key (dkpub/dkpriv). The application creates a certificate request using dkpub and the public key and signs the certificate request with using dkpriv. Next, the application derives second key pair from the TPM's storage root key. This key is the transport key (tkpub/tkpriv).
G	The task sends a device registration request to Azure DRS that includes the ID token, certificate request, tkpub, and attestation data. Azure DRS validates the ID token, creates a device ID, and creates a certificate based on the included certificate request. Azure DRS then updates the device object in Microsoft Entra ID and sends the device ID and the device certificate to the client.
H	Device registration completes by receiving the device ID and the device certificate from Azure DRS. The device ID is saved for future reference (viewable from `dsregcmd.exe /status`), and the device certificate is installed in the Personal store of the computer. With device registration complete, the task exits.

Microsoft Entra hybrid joined in Federated environments



Expand table

Phase	Description
A	The user signs in to a domain joined Windows 10 or newer computer using domain credentials. This credential can be user name and password or smart card authentication. The user sign-in triggers the Automatic Device Join task. The Automatic Device Join tasks is triggered on domain join and retried every hour. It doesn't solely depend on the user sign-in.
B	The task queries Active Directory using the LDAP protocol for the keywords attribute on the service connection point stored in the configuration partition in Active Directory (`CN=62a0ff2e-97b9-4513-943f-0d221bd30080,CN=Device Registration Configuration,CN=Services,CN=Configuration,DC=corp,DC=contoso,DC=com`). The value returned in the keywords attribute determines if device registration is directed to Azure Device Registration Service (DRS) or the enterprise device registration service hosted on-premises.
C	For the federated environments, the computer authenticates the enterprise device registration endpoint using Windows Integrated Authentication. The enterprise device registration service creates and returns a token that includes claims for the object GUID, computer SID, and domain joined state. The task submits the token and claims to Microsoft Entra ID where they're validated. Microsoft Entra ID returns an ID token to the running task.
D	The application creates TPM bound (preferred) RSA 2048 bit key-pair known as the device key (dkpub/dkpriv). The application creates a certificate request using dkpub and the public key and signs the certificate request with using dkpriv. Next, the application derives second key pair from the TPM's storage root key. This key is the transport key (tkpub/tkpriv).
E	To provide SSO for on-premises federated application, the task requests an enterprise PRT from the on-premises STS. Windows Server 2016 running the Active Directory Federation Services role validate the request and return it the running task.
F	The task sends a device registration request to Azure DRS that includes the ID token, certificate request, tkpub, and attestation data. Azure DRS validates the ID token, creates a device ID, and creates a certificate based on the included certificate request. Azure DRS then writes a device object in Microsoft Entra ID and sends the device ID and the device certificate to the client. Device registration completes by receiving the device ID and the device certificate from Azure DRS. The device ID is saved for future reference (viewable from `dsregcmd.exe /status`), and the device certificate is installed in the Personal store of the computer. With device registration complete, the task exits.
G	If Microsoft Entra Connect device writeback is enabled, Microsoft Entra Connect requests updates from Microsoft Entra ID at its next synchronization cycle (device writeback is required for hybrid deployment using certificate trust). Microsoft Entra ID correlates the device object with a matching synchronized computer object. Microsoft Entra Connect receives the device object that includes the object GUID and computer SID and writes the device object to Active Directory.

Next steps

- Microsoft Entra joined devices

- Microsoft Entra registered devices
- Microsoft Entra hybrid joined devices
- What is a Primary Refresh Token?
- Microsoft Entra Connect: Device options

How Windows Hello for Business provisioning works

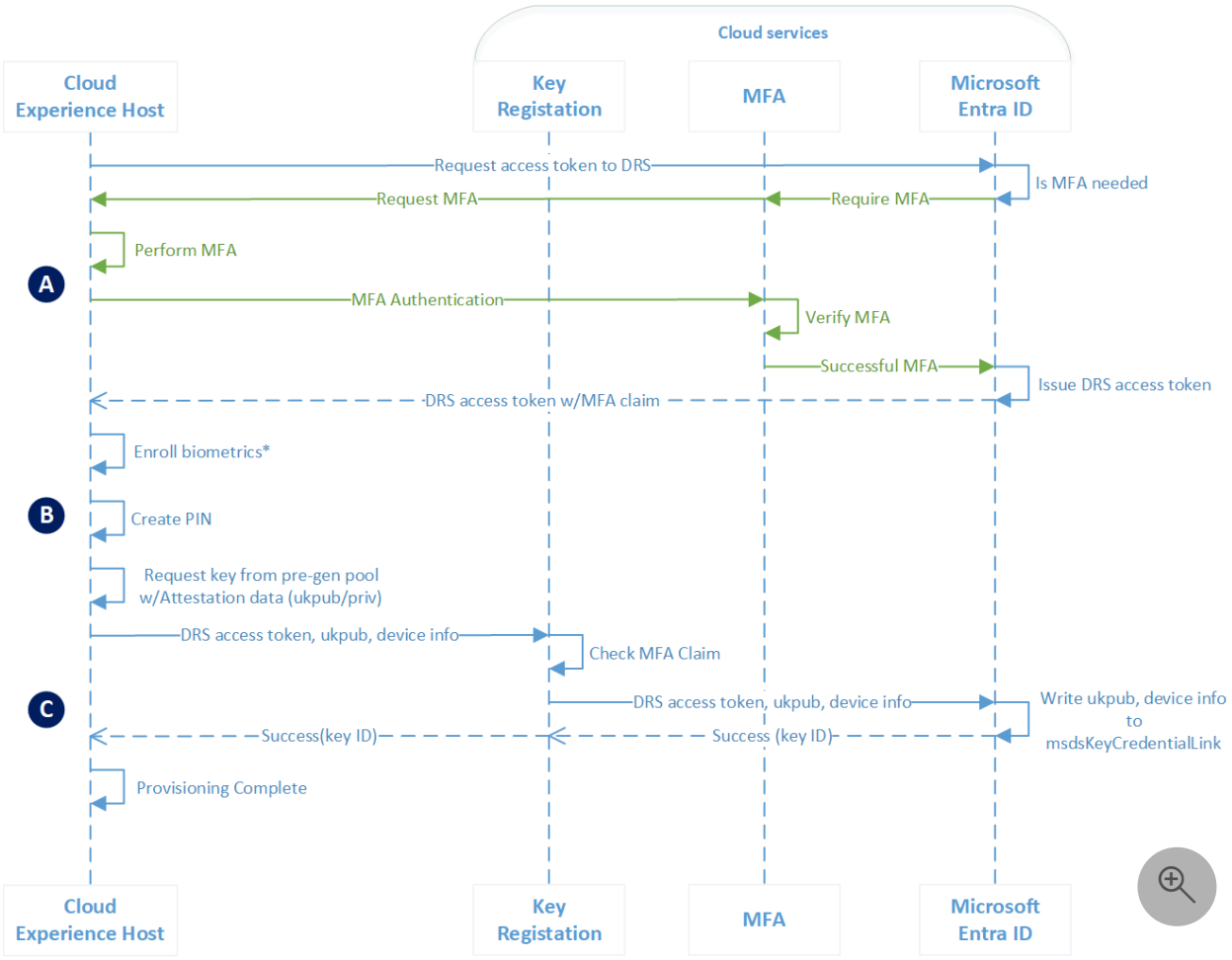
Article • 11/22/2024

Windows Hello for Business provisioning enables a user to enroll a new, strong, two-factor credential that they can use for passwordless authentication. Provisioning experience vary based on:

- How the device is joined to Microsoft Entra ID
- The Windows Hello for Business deployment type
- If the environment is managed or federated

> **Note**
>
> The flows in this section are not exhaustive for every possible scenario. For example, Federated Key Trust is also a supported configuration.

Provisioning for Microsoft Entra joined devices with managed authentication

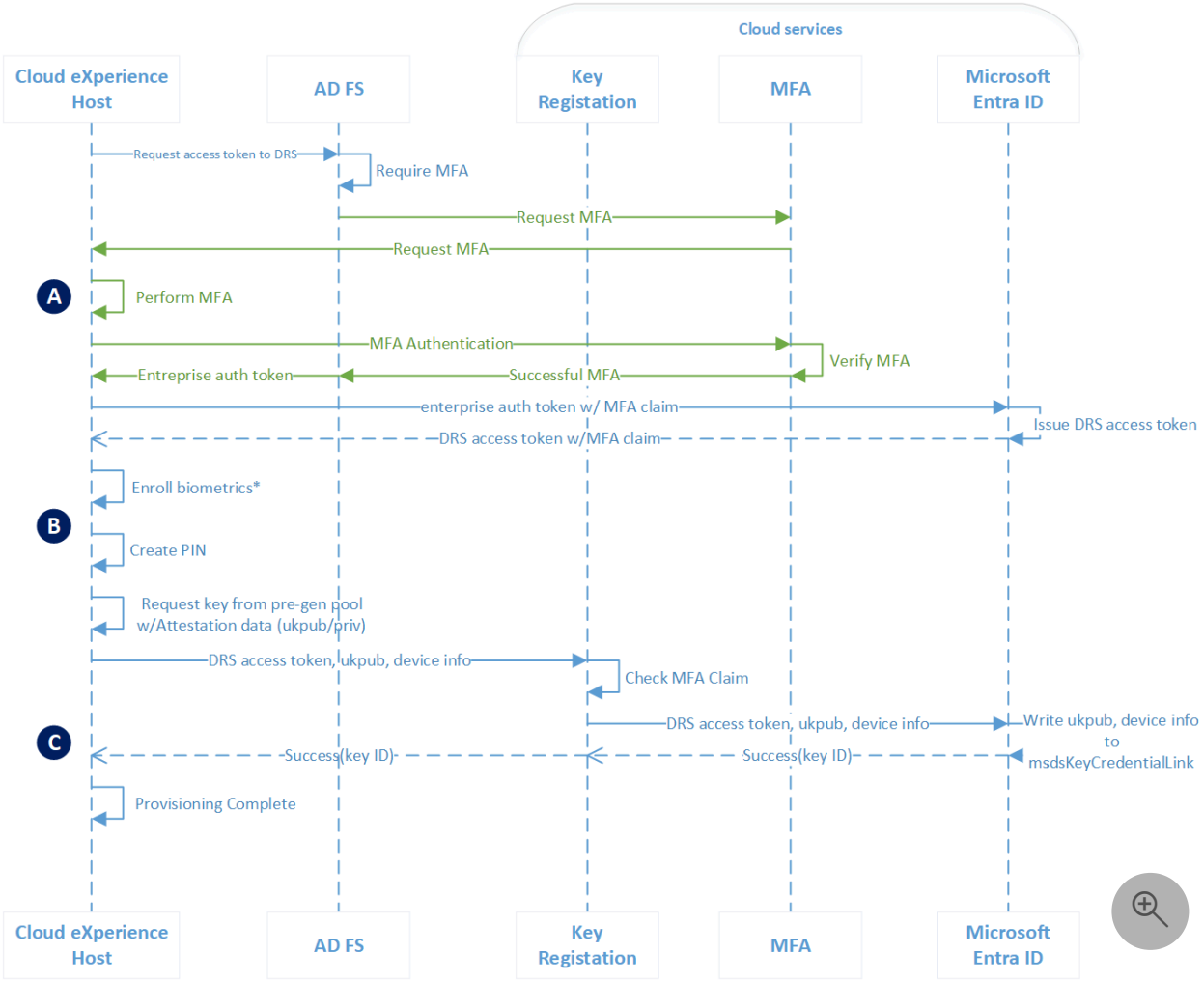


Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Azure Device Registration Service (ADRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. The Microsoft Entra multifactor authentication service provides the second factor of authentication. If the user has performed Microsoft Entra multifactor authentication within the last 10 minutes, such as when registering the device from the out-of-box-experience (OOBE), then they aren't prompted for MFA because the current MFA remains valid. Microsoft Entra ID validates the access token request and the MFA claim associated with it, creates an ADRS access token, and returns it to the application.
B	After receiving an ADRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).

Phase	Description
C	The application sends the ADRS token, ukpub, attestation data, and device information to ADRS for user key registration. Azure DRS validates the MFA claim remains current. On successful validation, Azure DRS locates the user's object in Microsoft Entra ID, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. Microsoft Entra ID returns a key ID to the application, which signals the end of user provisioning and the application exits.

Provisioning for Microsoft Entra joined devices with federated authentication

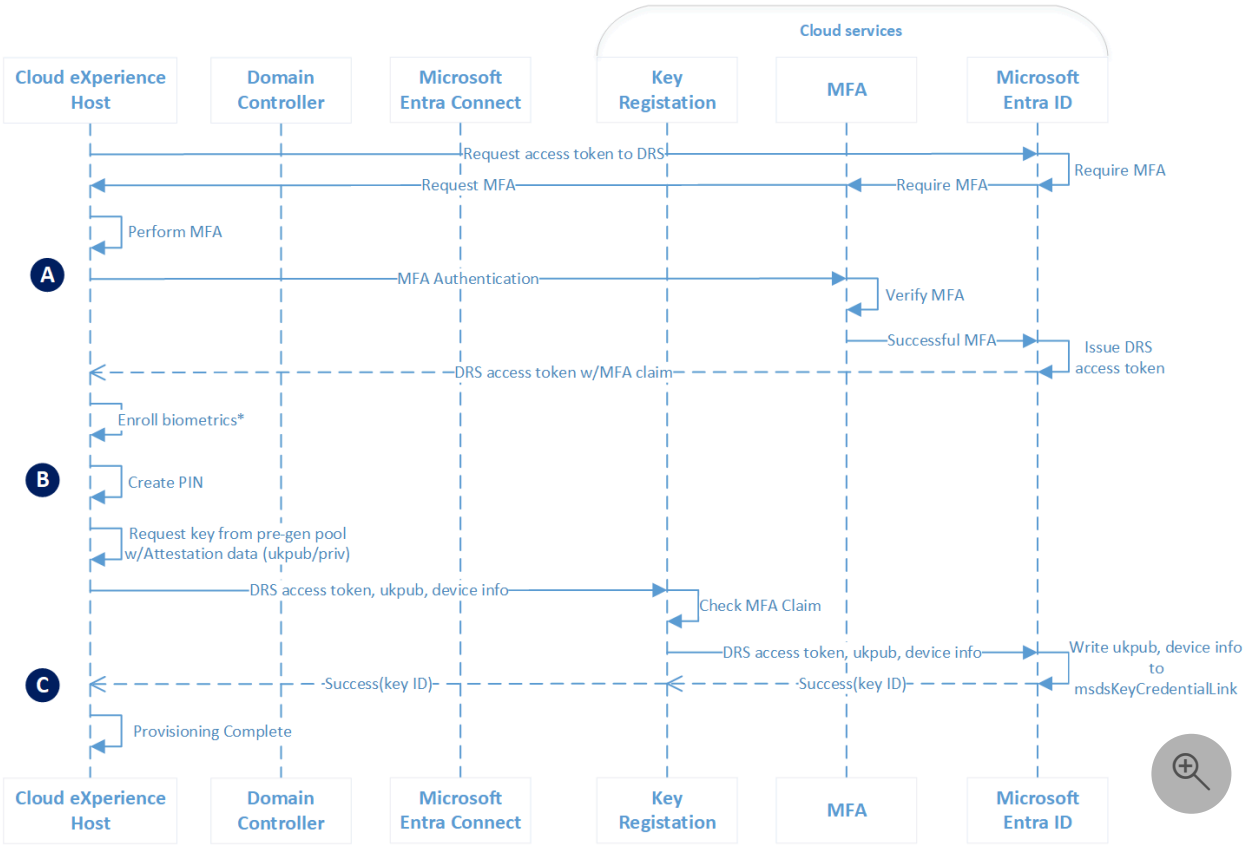


Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Azure Device Registration Service (ADRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. In a federated environment, the plug-in sends the token request to the on-premises STS,

Phase	Description
	such as Active Directory Federation Services. The on-premises STS authenticates the user and determines if the user should perform another factor of authentication. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. The Microsoft Entra multifactor authentication service provides the second factor of authentication. If the user has performed Microsoft Entra multifactor authentication within the last 10 minutes, such as when registering the device from the out-of-box-experience (OOBE), then they aren't prompted for MFA because the current MFA remains valid. The on-premises STS server issues an enterprise token on successful MFA. The application sends the token to Microsoft Entra ID. Microsoft Entra ID validates the access token request and the MFA claim associated with it, creates an ADRS access token, and returns it to the application.
B	After receiving an ADRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the ADRS token, ukpub, attestation data, and device information to ADRS for user key registration. Azure DRS validates MFA claim remains current. On successful validation, Azure DRS locates the user's object in Microsoft Entra ID, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. Microsoft Entra ID returns key ID to the application, which signals the end of user provisioning and the application exits.

Provisioning in a cloud Kerberos trust deployment model with managed authentication



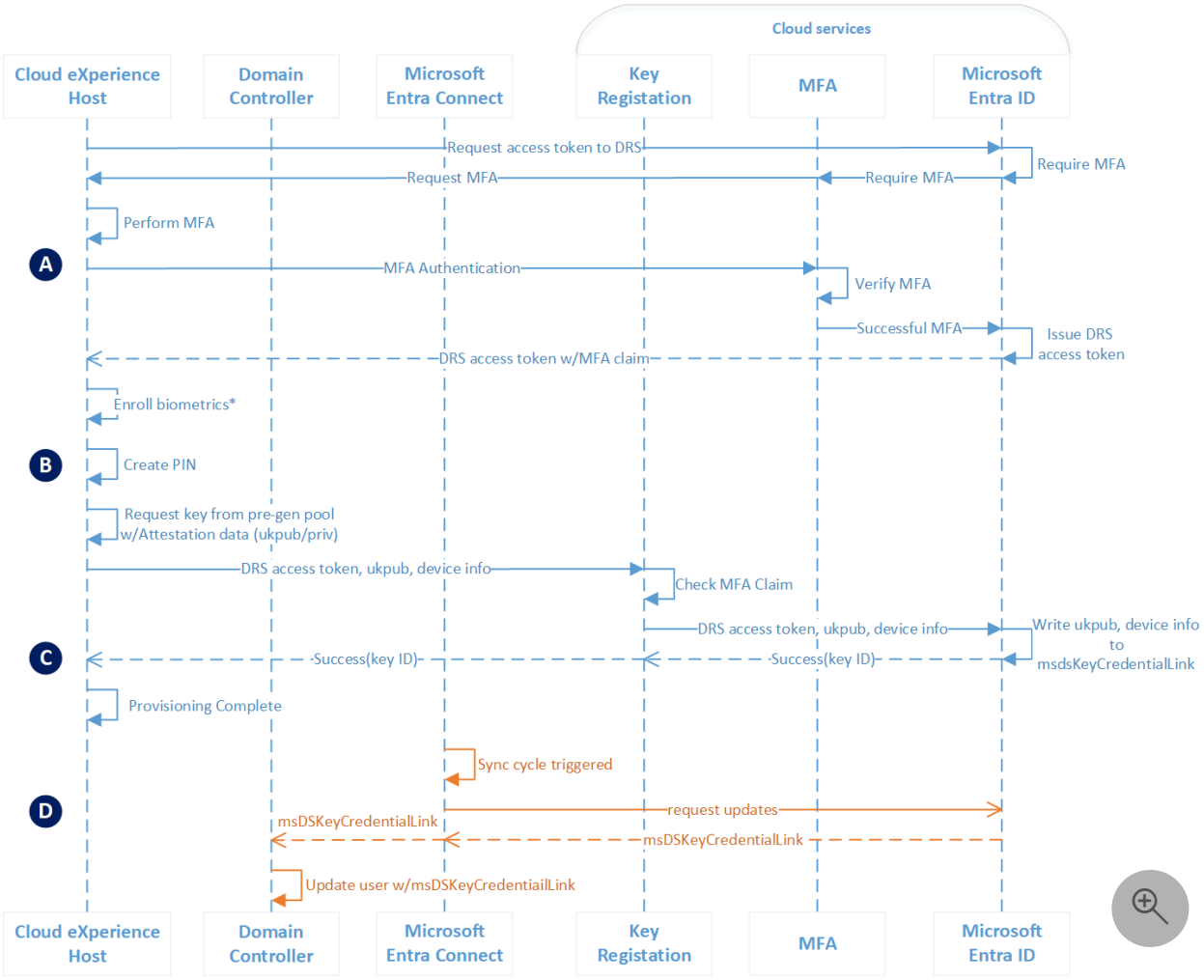
Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Azure Device Registration Service (ADRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. The Microsoft Entra multifactor authentication service provides the second factor of authentication. If the user has performed Microsoft Entra multifactor authentication within the last 10 minutes, such as when registering the device from the out-of-box-experience (OOBE), then they aren't prompted for MFA because the current MFA remains valid. Microsoft Entra ID validates the access token request and the MFA claim associated with it, creates an ADRS access token, and returns it to the application.
B	After receiving an ADRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the ADRS token, ukpub, attestation data, and device information to ADRS for user key registration. Azure DRS validates the MFA claim remains current. On successful validation, Azure DRS locates the user's object in Microsoft Entra ID, writes the

Phase	Description
	key information to a multi-values attribute. The key information includes a reference to the device from which it was created. Microsoft Entra ID returns a key ID to the application, which signals the end of user provisioning and the application exits.

> **Note**
>
> Windows Hello for Business cloud Kerberos trust does not require users' keys to be synced from Microsoft Entra ID to Active Directory. Users can immediately authenticate to Microsoft Entra ID and AD after provisioning their credential.

Provisioning in a hybrid key trust deployment model with managed authentication

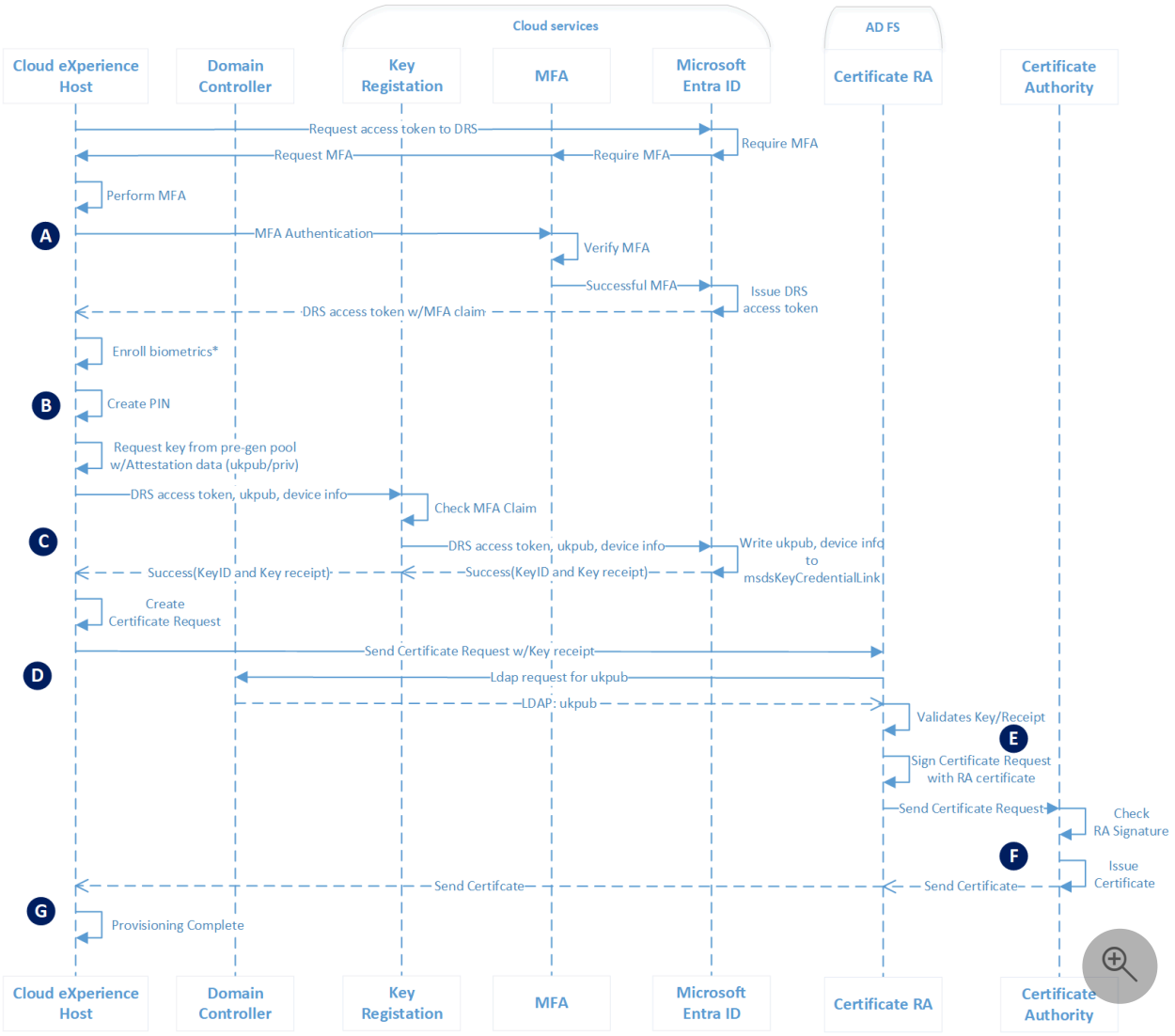


Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Azure Device Registration Service (ADRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. The Microsoft Entra multifactor authentication service provides the second factor of authentication. If the user has performed Microsoft Entra multifactor authentication within the last 10 minutes, such as when registering the device from the out-of-box-experience (OOBE), then they aren't prompted for MFA because the current MFA remains valid. Microsoft Entra ID validates the access token request and the MFA claim associated with it, creates an ADRS access token, and returns it to the application.
B	After receiving an ADRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the ADRS token, ukpub, attestation data, and device information to ADRS for user key registration. Azure DRS validates the MFA claim remains current. On successful validation, Azure DRS locates the user's object in Microsoft Entra ID, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. Microsoft Entra ID returns a key ID to the application, which signals the end of user provisioning and the application exits.
D	Microsoft Entra Connect requests updates on its next synchronization cycle. Microsoft Entra ID sends the user's public key that was securely registered through provisioning. Microsoft Entra Connect receives the public key and writes it to user's `msDS-KeyCredentialLink` attribute in Active Directory.

> **ⓘ Important**
>
> The newly provisioned user will not be able to sign in using Windows Hello for Business until Microsoft Entra Connect successfully synchronizes the public key to the on-premises Active Directory.

Provisioning in a hybrid certificate trust deployment model with federated authentication



Expand table

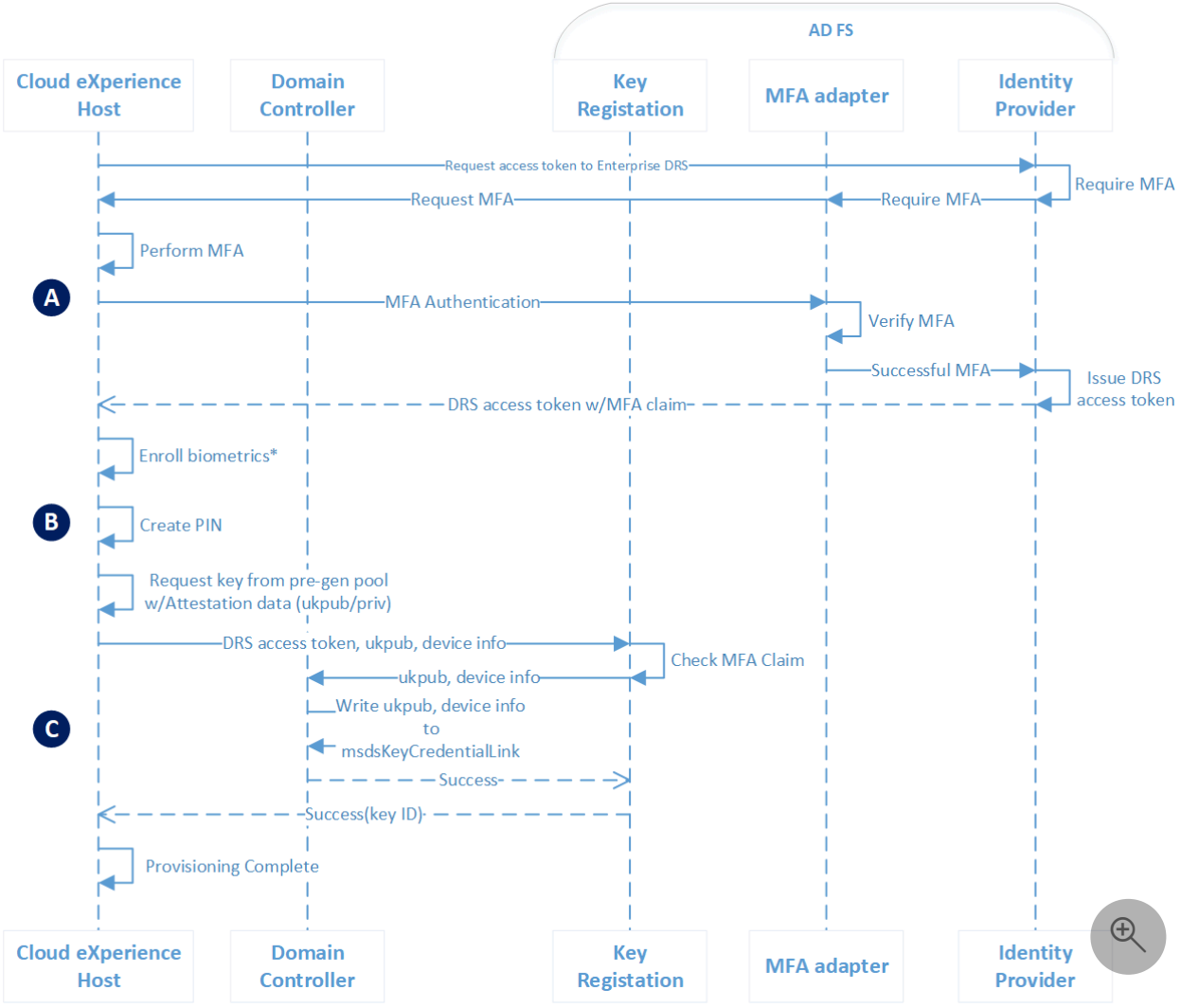
Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Azure Device Registration Service (ADRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. In a federated environment, the plug-in sends the token request to the on-premises STS, such as Active Directory Federation Services. The on-premises STS authenticates the user and determines if the user should perform another factor of authentication. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. The Microsoft Entra multifactor authentication service (or a non-Microsoft MFA service) provides the second factor of authentication. The on-premises STS server issues an enterprise token on successful MFA. The application sends the token to Microsoft Entra ID. Microsoft Entra ID validates the access token request and the MFA claim associated with it, creates an ADRS access token, and returns it to the application.
B	After receiving an ADRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric

Phase	Description
	sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the ADRS token, ukpub, attestation data, and device information to ADRS for user key registration. Azure DRS validates the MFA claim remains current. On successful validation, Azure DRS locates the user's object in Microsoft Entra ID, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. Microsoft Entra ID returns a key ID and a key receipt to the application, which represents the end of user key registration.
D	The certificate request portion of provisioning begins after the application receives a successful response from key registration. The application creates a PKCS#10 certificate request. The key used in the certificate request is the same key that was securely provisioned. The application sends the key receipt and certificate request, which includes the public key, to the certificate registration authority hosted on the Active Directory Federation Services (AD FS) farm. After receiving the certificate request, the certificate registration authority queries Active Directory for the msDS-KeyCredentialsLink for a list of registered public keys.
E	The registration authority validates the public key in the certificate request matches a registered key for the user. If the public key in the certificate isn't found in the list of registered public keys, it then validates the key receipt to confirm the key was securely registered with Azure. After validating the key receipt or public key, the registration authority signs the certificate request using its enrollment agent certificate.
F	The registration authority sends the certificate request to the enterprise issuing certificate authority. The certificate authority validates the certificate request is signed by a valid enrollment agent and, on success, issues a certificate and returns it to the registration authority that then returns the certificate to the application.
G	The application receives the newly issued certificate and installs it into the Personal store of the user. This signals the end of provisioning.

ⓘ Important

Synchronous certificate enrollment doesn't depend on Microsoft Entra Connect to synchronize the user's public key to issue the Windows Hello for Business authentication certificate. Users can sign-in using the certificate immediately after provisioning completes. Microsoft Entra Connect continues to synchronize the public key to Active Directory, but is not shown in this flow.

Provisioning in an on-premises key trust deployment model

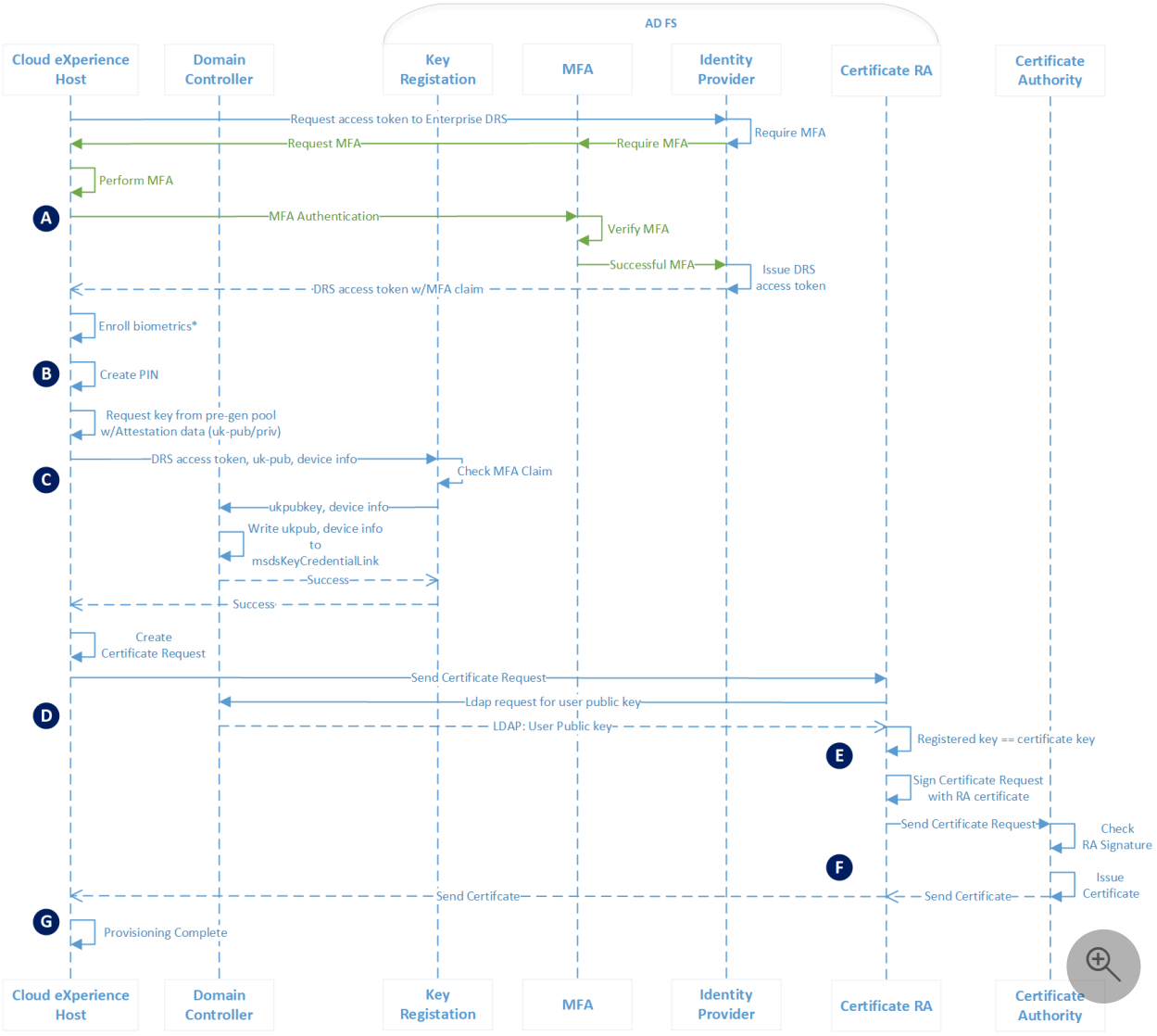


Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Enterprise Device Registration Service (EDRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. In an on-premises deployment, the plug-in sends the token request to the on-premises STS, such as Active Directory Federation Services. The on-premises STS authenticates the user and determines if the user should perform another factor of authentication. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. Microsoft Entra multifactor authentication server (or a non-Microsoft MFA service) provides the second factor of authentication. The on-premises STS server issues an enterprise DRS token on successful MFA.
B	After receiving an EDRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives

Phase	Description
	the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the EDRS token, ukpub, attestation data, and device information to the Enterprise DRS for user key registration. Enterprise DRS validates the MFA claim remains current. On successful validation, the Enterprise DRS locates the user's object in Active Directory, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. The Enterprise DRS returns a key ID to the application, which represents the end of user key registration.

Provisioning in an on-premises certificate trust deployment model



Expand table

Phase	Description
A	The provisioning application hosted in the Cloud Experience Host (CXH) starts provisioning by requesting an access token for the Enterprise Device Registration Service (EDRS). The application makes the request using the Microsoft Entra Web Account Manager plug-in. In an on-premises deployment, the plug-in sends the token request to the on-premises STS, such as Active Directory Federation Services. The on-premises STS authenticates the user and determines if the user should perform another factor of authentication. Users must provide two factors of authentication. In this phase, the user has already provided one factor of authentication, typically user name and password. Microsoft Entra multifactor authentication server (or a non-Microsoft MFA service) provides the second factor of authentication. The on-premises STS server issues an enterprise DRS token on successful MFA.
B	After receiving an EDRS access token, the application detects if the device has a Windows Hello biometric compatible sensor. If the application detects a biometric sensor, it gives the user the choice to enroll biometrics. After completing or skipping biometric enrollment, the application requires the user to create a PIN and the default (and fall-back gesture when used with biometrics). The user provides and confirms their PIN. Next, the application requests a Windows Hello for Business key pair from the key pregeneration pool, which includes attestation data. This is the user key (ukpub/ukpriv).
C	The application sends the EDRS token, ukpub, attestation data, and device information to the Enterprise DRS for user key registration. Enterprise DRS validates the MFA claim remains current. On successful validation, the Enterprise DRS locates the user's object in Active Directory, writes the key information to a multi-values attribute. The key information includes a reference to the device from which it was created. The Enterprise DRS returns a key ID to the application, which represents the end of user key registration.
D	The certificate request portion of provisioning begins after the application receives a successful response from key registration. The application creates a PKCS#10 certificate request. The key used in the certificate request is the same key that was securely provisioned. The application sends the certificate request, which includes the public key, to the certificate registration authority hosted on the Active Directory Federation Services (AD FS) farm. After receiving the certificate request, the certificate registration authority queries Active Directory for the msDS-KeyCredentialsLink for a list of registered public keys.
E	The registration authority validates the public key in the certificate request matches a registered key for the user. After validating the public key, the registration authority signs the certificate request using its enrollment agent certificate.
F	The registration authority sends the certificate request to the enterprise issuing certificate authority. The certificate authority validates the certificate request is signed by a valid enrollment agent and, on success, issues a certificate and returns it to the registration authority that then returns the certificate to the application.

Phase	Description
G	The application receives the newly issued certificate and installs it into the Personal store of the user. This signals the end of provisioning.

Feedback

Was this page helpful? Yes No

Provide product feedback

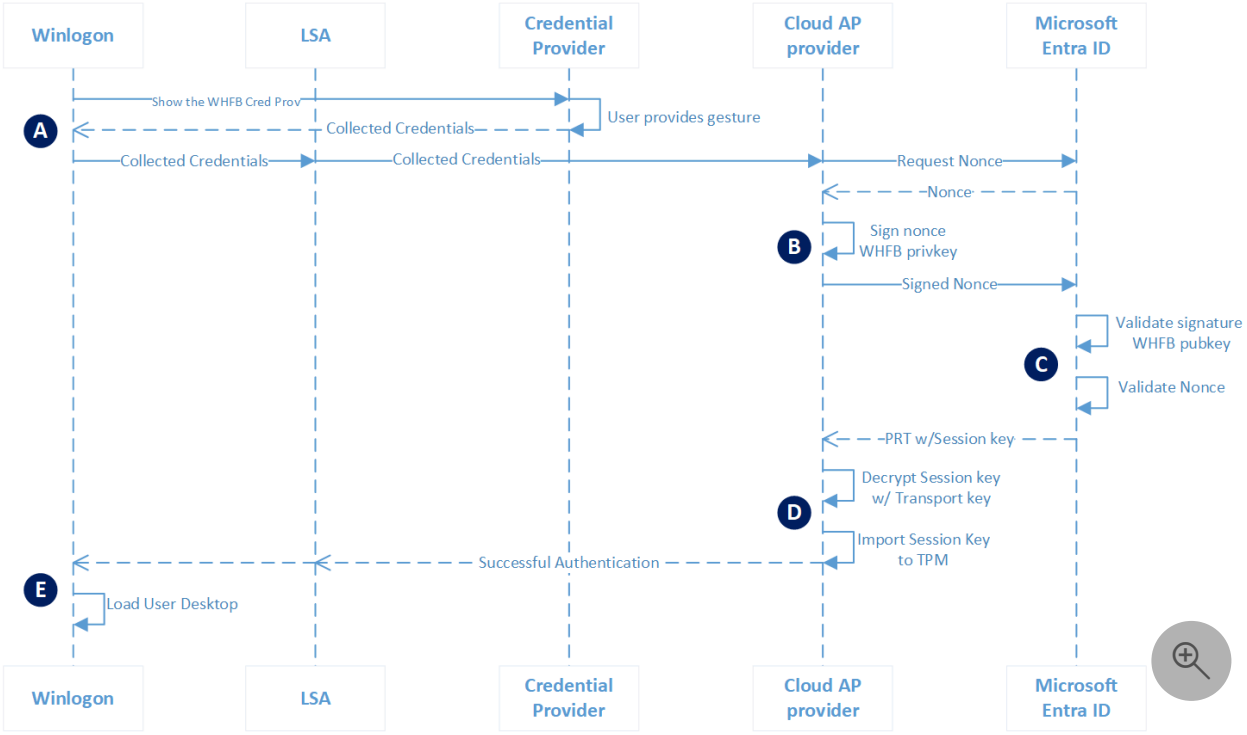
Windows Hello for Business authentication

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

Windows Hello for Business authentication is a passwordless, two-factor authentication. Authenticating with Windows Hello for Business provides a convenient sign-in experience that authenticates the user to both Microsoft Entra ID and Active Directory resources.

Microsoft Entra joined devices authenticate to Microsoft Entra ID during sign-in and can, optionally, authenticate to Active Directory. Microsoft Entra hybrid joined devices authenticate to Active Directory during sign-in, and authenticate to Microsoft Entra ID in the background.

Microsoft Entra join authentication to Microsoft Entra ID

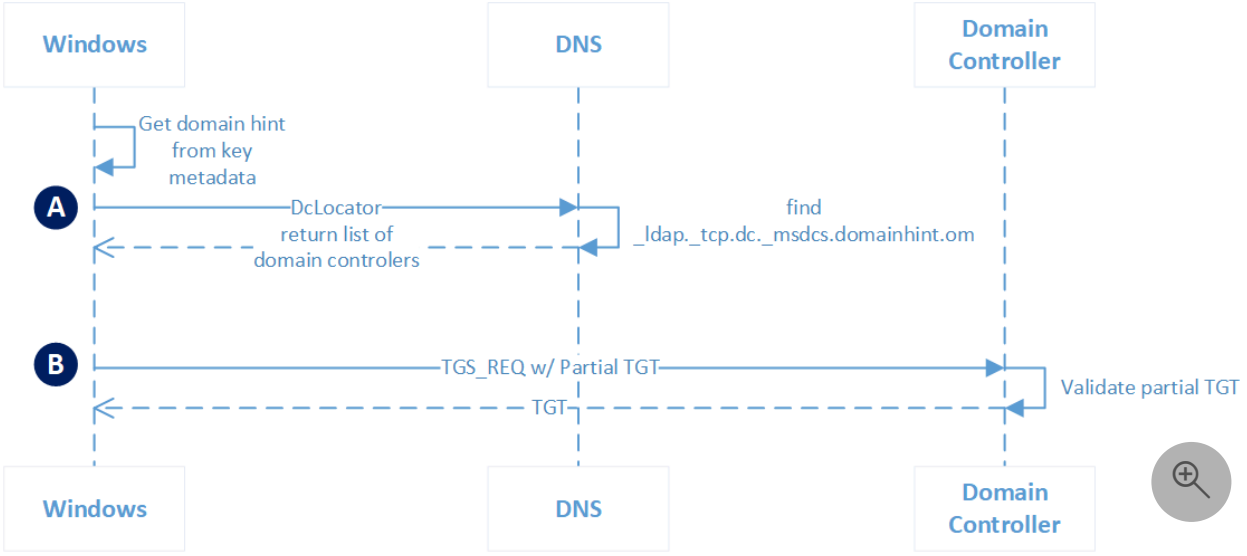


> **Note**
>
> All Microsoft Entra joined devices authenticate with Windows Hello for Business to Microsoft Entra ID the same way. The Windows Hello for Business trust type only impacts how the device authenticates to on-premises AD.

Expand table

Phase	Description
A	Authentication begins when the user dismisses the lock screen, which triggers Winlogon to show the Windows Hello for Business credential provider. The user provides their Windows Hello gesture (PIN or biometrics). The credential provider packages these credentials and returns them to Winlogon. Winlogon passes the collected credentials to LSASS. LSASS passes the collected credentials to the Cloud Authentication security support provider, referred to as the Cloud AP provider.
B	The Cloud AP provider requests a nonce from Microsoft Entra ID. Microsoft Entra ID returns a nonce. The Cloud AP provider signs the nonce using the user's private key and returns the signed nonce to the Microsoft Entra ID.
C	Microsoft Entra ID validates the signed nonce using the user's securely registered public key against the nonce signature. Microsoft Entra ID then validates the returned signed nonce, and creates a PRT with session key that is encrypted to the device's transport key and returns it to the Cloud AP provider.
D	The Cloud AP provider receives the encrypted PRT with session key. Using the device's private transport key, the Cloud AP provider decrypt the session key and protects the session key using the device's TPM.
E	The Cloud AP provider returns a successful authentication response to LSASS. LSASS caches the PRT, and informs Winlogon of the success authentication. Winlogon creates a logon session, loads the user's profile, and starts explorer.exe.

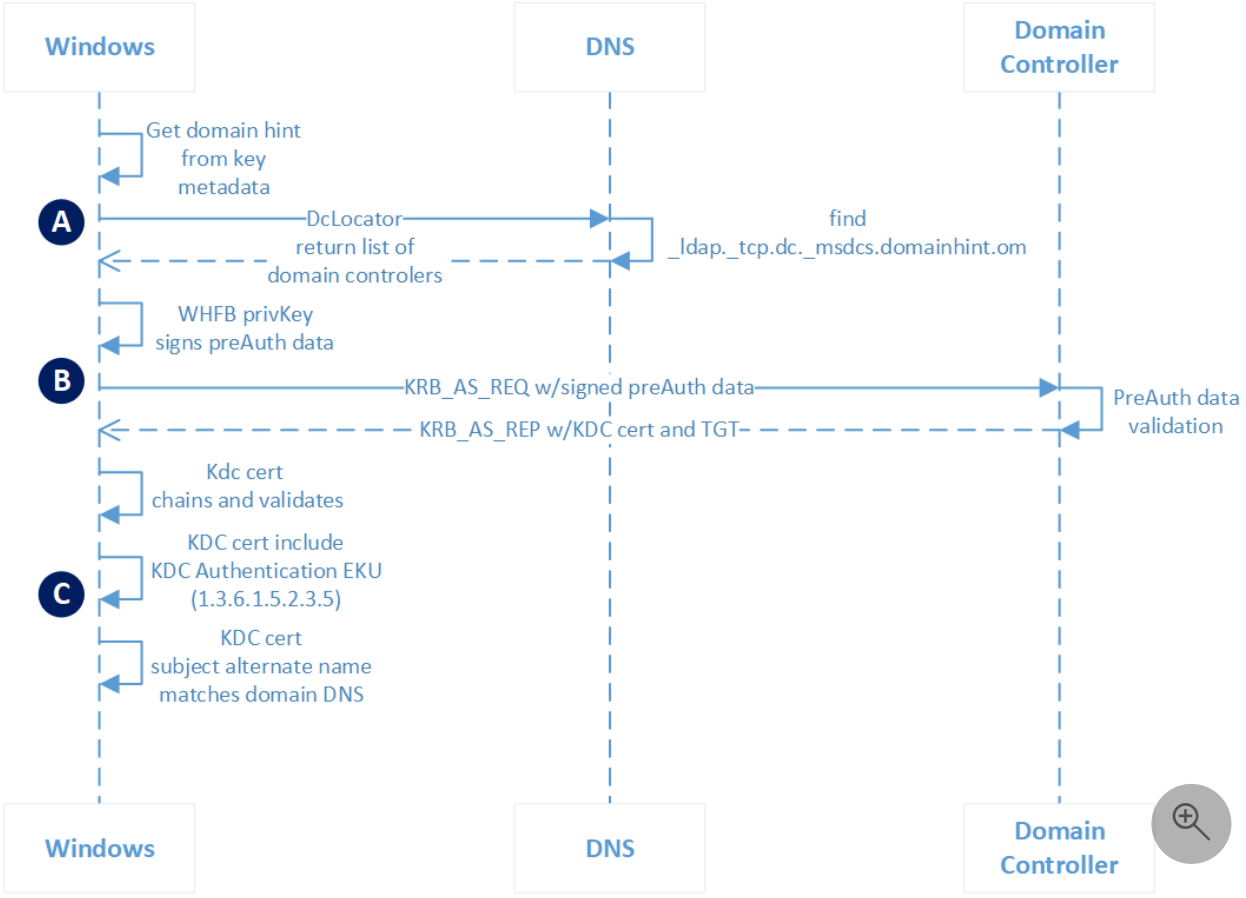
Microsoft Entra join authentication to Active Directory using cloud Kerberos trust



Expand table

Phase	Description
A	Authentication to Active Directory from a Microsoft Entra joined device begins with the user first attempts to use a resource that needs Kerberos authentication. The Kerberos security support provider, hosted in LSASS, uses metadata from the Windows Hello for Business key to get a hint of the user's domain. Using the hint, the provider uses the DClocator service to locate a domain controller.
B	After locating a domain controller, the Kerberos provider sends a partial TGT that it received from Microsoft Entra ID from a previous Microsoft Entra authentication to the domain controller. The partial TGT contains only the user SID, and it's signed by Microsoft Entra Kerberos. The domain controller verifies that the partial TGT is valid. On success, the KDC returns a TGT to the client.

Microsoft Entra join authentication to Active Directory using a key



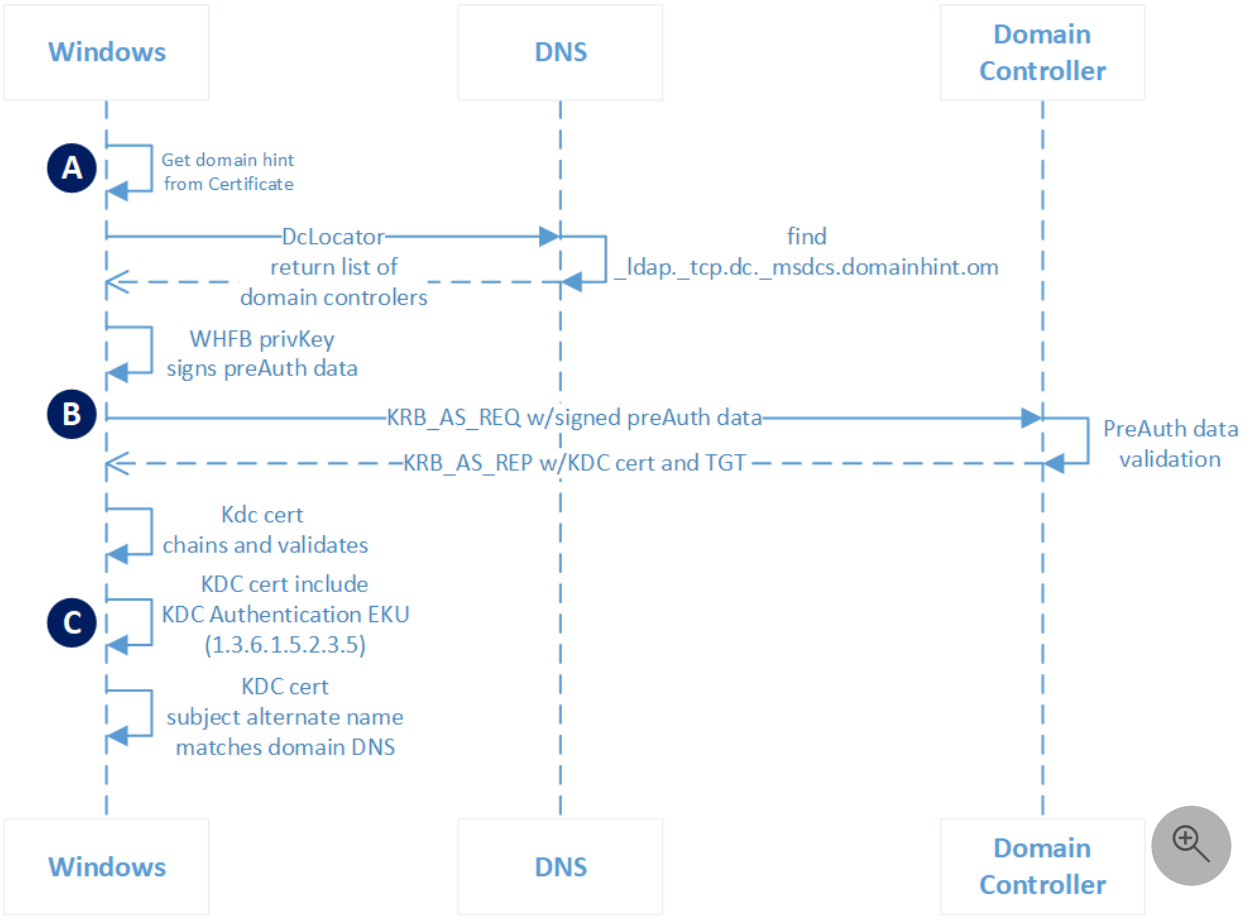
Expand table

Phase	Description
A	Authentication to Active Directory from a Microsoft Entra joined device begins with the user first attempts to use a resource that needs Kerberos authentication. The Kerberos security support provider, hosted in LSASS, uses metadata from the Windows Hello for

Phase	Description
	Business key to get a hint of the user's domain. Using the hint, the provider uses the DClocator service to locate a domain controller. After the provider locates a domain controller, the provider uses the private key to sign the Kerberos preauthentication data.
B	The Kerberos provider sends the signed preauthentication data and its public key (in the form of a self-signed certificate) to the Key Distribution Center (KDC) service running on the domain controller in the form of a KERB_AS_REQ. The domain controller determines the certificate is a self-signed certificate. It retrieves the public key from the certificate included in the KERB_AS_REQ and searches for the public key in Active Directory. It validates the UPN for authentication request matches the UPN registered in Active Directory and validates the signed preauthentication data using the public key from Active Directory. On success, the KDC returns a TGT to the client with its certificate in a KERB_AS_REP.
C	The Kerberos provider ensures it can trust the response from the domain controller. First, it ensures the KDC certificate chains to a root certificate that is trusted by the device. Next, it ensures the certificate is within its validity period and that it hasn't been revoked. The Kerberos provider then verifies the certificate has the KDC Authentication present and that the subject alternate name listed in the KDC's certificate matches the domain name to which the user is authenticating. After passing this criteria, Kerberos returns the TGT to LSASS, where it's cached and used for subsequent service ticket requests.

> **Note**
>
> You might have an on-premises domain federated with Microsoft Entra ID. Once you have successfully provisioned Windows Hello for Business PIN/Bio on the Microsoft Entra joined device, any future login of Windows Hello for Business (PIN/Bio) sign-in will directly authenticate against Microsoft Entra ID to get PRT and trigger authenticate against your DC (if LOS to DC is available) to get Kerberos. It no longer uses AD FS to authenticate for Windows Hello for Business sign-ins.

Microsoft Entra join authentication to Active Directory using a certificate



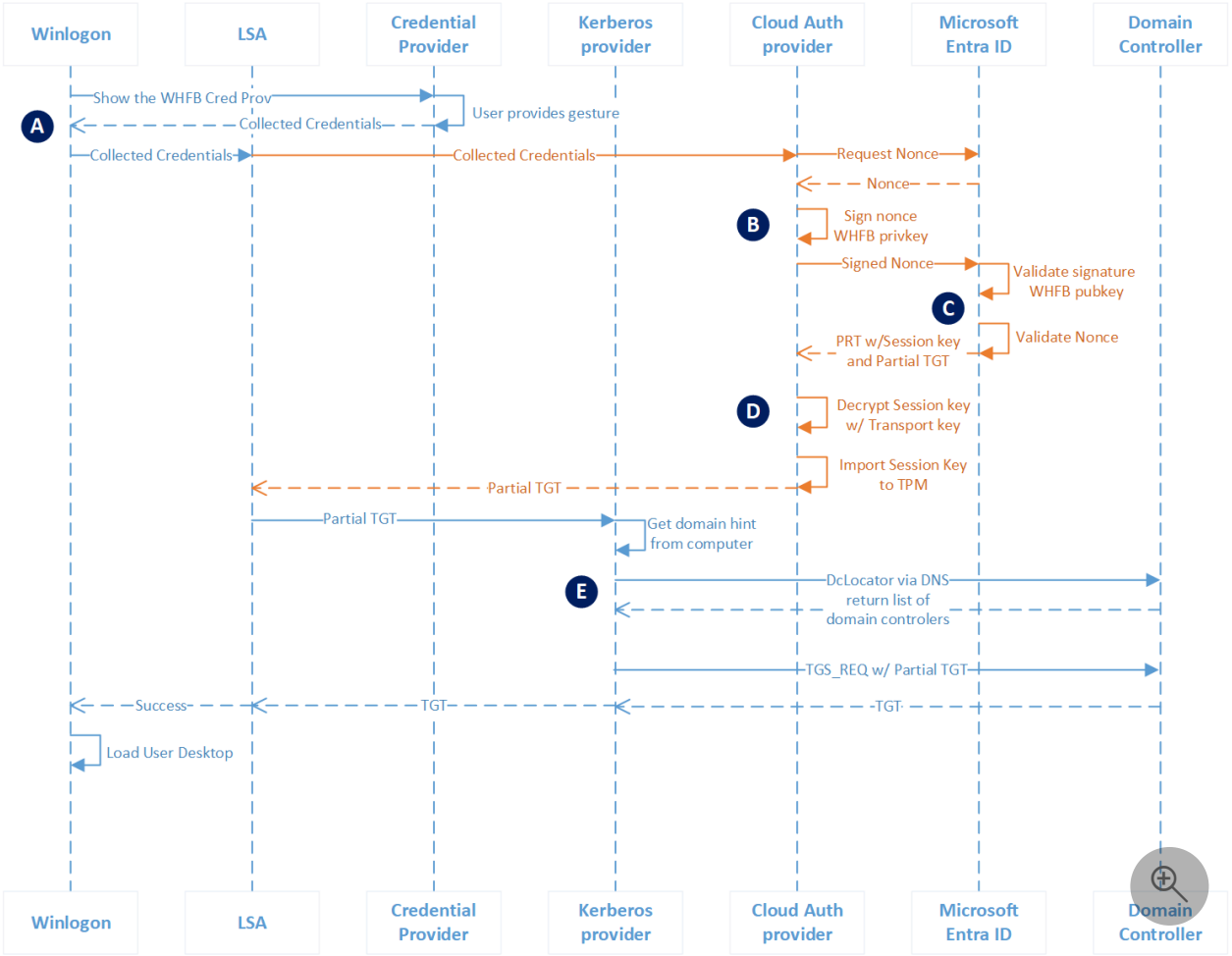
Expand table

Phase	Description
A	Authentication to Active Directory from a Microsoft Entra joined device begins with the user first attempts to use a resource that needs Kerberos authentication. The Kerberos security support provider, hosted in LSASS, uses information from the certificate to get a hint of the user's domain. Kerberos can use the distinguished name of the user found in the subject of the certificate, or it can use the user principal name of the user found in the subject alternate name of the certificate. Using the hint, the provider uses the DClocator service to locate a domain controller. After the provider locates an active domain controller, the provider uses the private key to sign the Kerberos preauthentication data.
B	The Kerberos provider sends the signed preauthentication data and user's certificate, which includes the public key, to the Key Distribution Center (KDC) service running on the domain controller in the form of a KERB_AS_REQ. The domain controller determines the certificate isn't self-signed certificate. The domain controller ensures the certificate chains to trusted root certificate, is within its validity period, can be used for authentication, and hasn't been revoked. It retrieves the public key and UPN from the certificate included in the KERB_AS_REQ and searches for the UPN in Active Directory. It validates the signed preauthentication data using the public key from the certificate. On success, the KDC returns a TGT to the client with its certificate in a KERB_AS_REP.
C	The Kerberos provider ensures it can trust the response from the domain controller. First, it ensures the KDC certificate chains to a root certificate that is trusted by the device.

Phase	Description
	Next, it ensures the certificate is within its validity period and that it hasn't been revoked. The Kerberos provider then verifies the certificate has the KDC Authentication present and that the subject alternate name listed in the KDC's certificate matches the domain name to which the user is authenticating. After passing this criteria, Kerberos returns the TGT to LSASS, where it's cached and used for subsequent service ticket requests.

> **Note**
>
> You may have an on-premises domain federated with Microsoft Entra ID. Once you have successfully provisioned Windows Hello for Business PIN/Bio on, any future login of Windows Hello for Business (PIN/Bio) sign-in will directly authenticate against Microsoft Entra ID to get PRT, as well as authenticate against your DC (if LOS to DC is available) to get Kerberos as mentioned previously. AD FS federation is used only when Enterprise PRT calls are placed from the client. You need to have device write-back enabled to get "Enterprise PRT" from your federation.

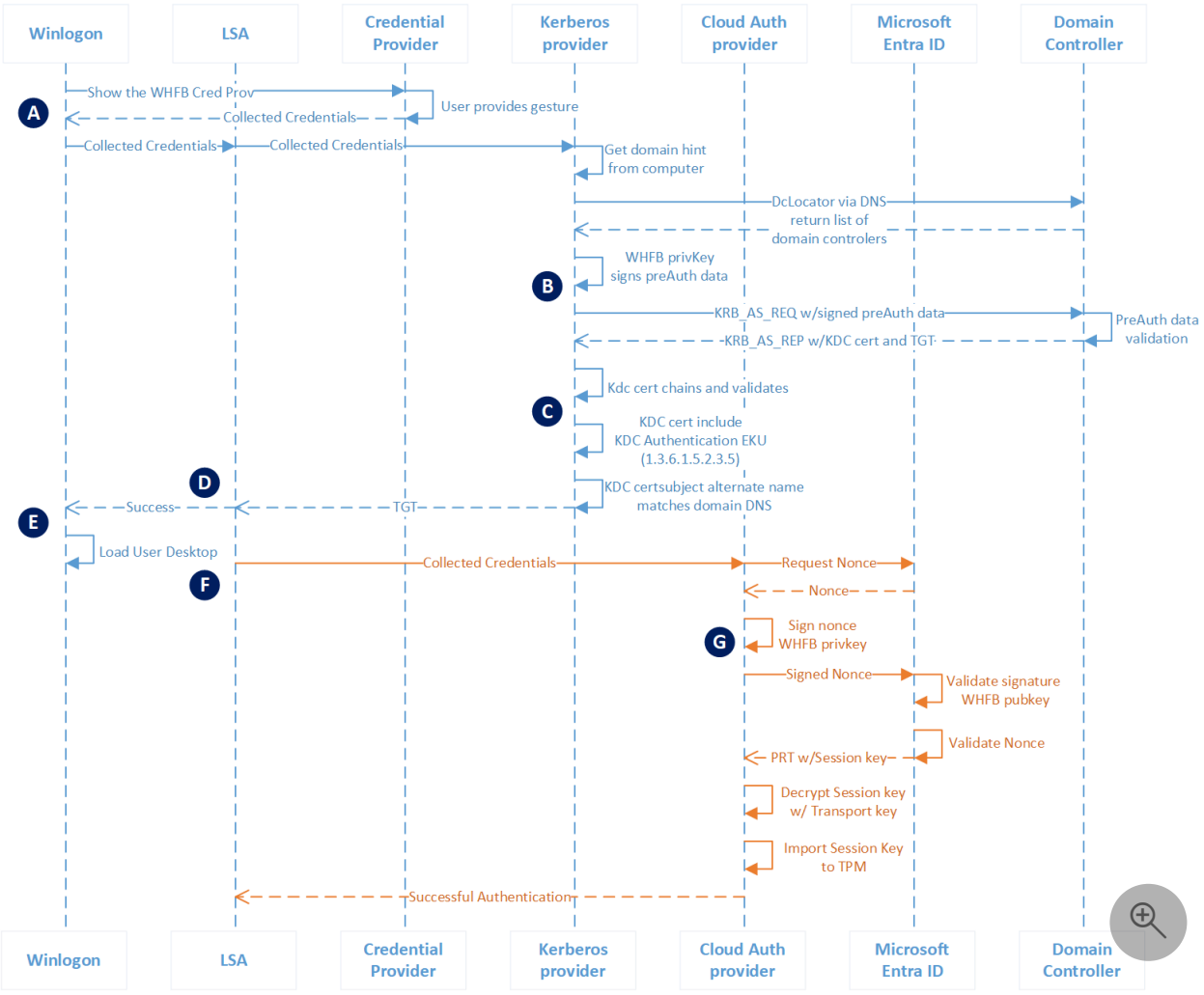
Microsoft Entra hybrid join authentication using cloud Kerberos trust



Expand table

Phase	Description
A	Authentication begins when the user dismisses the lock screen, which triggers Winlogon to show the Windows Hello for Business credential provider. The user provides their Windows Hello gesture (PIN or biometrics). The credential provider packages these credentials and returns them to Winlogon. Winlogon passes the collected credentials to LSASS. LSASS queries Windows Hello for Business policy to check if cloud Kerberos trust is enabled. If cloud Kerberos trust is enabled, LSASS passes the collected credentials to the Cloud Authentication security support provider, or Cloud AP. Cloud AP requests a nonce from Microsoft Entra ID. Microsoft Entra ID returns a nonce.
B	Cloud AP signs the nonce using the user's private key and returns the signed nonce to Microsoft Entra ID.
C	Microsoft Entra ID validates the signed nonce using the user's securely registered public key against the nonce signature. After validating the signature, Microsoft Entra ID then validates the returned signed nonce. After validating the nonce, Microsoft Entra ID creates a PRT with session key that is encrypted to the device's transport key and creates a Partial TGT from Microsoft Entra Kerberos and returns them to Cloud AP.
D	Cloud AP receives the encrypted PRT with session key. Using the device's private transport key, Cloud AP decrypts the session key and protects the session key using the device's TPM (if available). Cloud AP returns a successful authentication response to LSASS. LSASS caches the PRT and the Partial TGT.
E	The Kerberos security support provider, hosted in LSASS, uses metadata from the Windows Hello for Business key to get a hint of the user's domain. Using the hint, the provider uses the DClocator service to locate a domain controller. After locating an active domain controller, the Kerberos provider sends the partial TGT that it received from Microsoft Entra ID to the domain controller. The partial TGT contains only the user SID and is signed by Microsoft Entra Kerberos. The domain controller verifies that the partial TGT is valid. On success, the KDC returns a TGT to the client. Kerberos returns the TGT to LSASS, where it's cached and used for subsequent service ticket requests. LSASS informs Winlogon of the success authentication. Winlogon creates a logon session, loads the user's profile, and starts explorer.exe.

Microsoft Entra hybrid join authentication using a key



Expand table

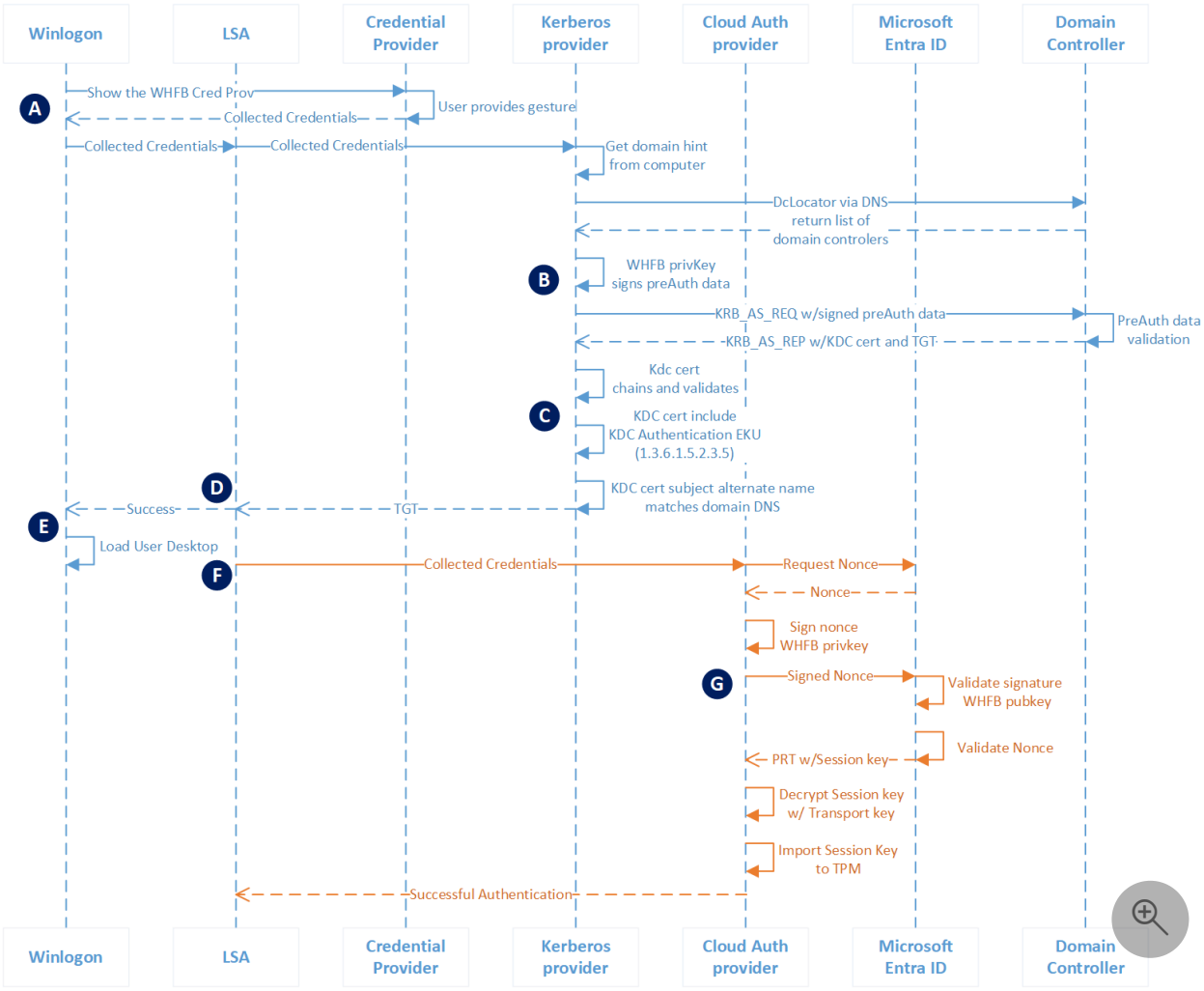
Phase	Description
A	Authentication begins when the user dismisses the lock screen, which triggers Winlogon to show the Windows Hello for Business credential provider. The user provides their Windows Hello gesture (PIN or biometrics). The credential provider packages these credentials and returns them to Winlogon. Winlogon passes the collected credentials to LSASS. LSASS passes the collected credentials to the Kerberos security support provider. The Kerberos provider gets domain hints from the domain joined workstation to locate a domain controller for the user.
B	The Kerberos provider sends the signed preauthentication data and the user's public key (in the form of a self-signed certificate) to the Key Distribution Center (KDC) service running on the domain controller in the form of a KERB_AS_REQ. The domain controller determines the certificate is a self-signed certificate. It retrieves the public key from the certificate included in the KERB_AS_REQ and searches for the public key in Active Directory. It validates the UPN for authentication request matches the UPN registered in Active Directory and validates the signed preauthentication data using the public key from Active Directory. On success, the KDC returns a TGT to the client with its certificate in a KERB_AS_REP.
C	The Kerberos provider ensures it can trust the response from the domain controller. First, it ensures the KDC certificate chains to a root certificate that is trusted by the device.

Phase	Description
	Next, it ensures the certificate is within its validity period and that it hasn't been revoked. The Kerberos provider then verifies the certificate has the KDC Authentication present and that the subject alternate name listed in the KDC's certificate matches the domain name to which the user is authenticating.
D	After passing this criteria, Kerberos returns the TGT to LSASS, where it's cached and used for subsequent service ticket requests.
E	LSASS informs Winlogon of the success authentication. Winlogon creates a logon session, loads the user's profile, and starts explorer.exe.
F	While Windows loads the user's desktop, LSASS passes the collected credentials to the Cloud Authentication security support provider, referred to as the Cloud AP provider. The Cloud AP provider requests a nonce from Microsoft Entra ID. Microsoft Entra ID returns a nonce.
G	The Cloud AP provider signs the nonce using the user's private key and returns the signed nonce to the Microsoft Entra ID. Microsoft Entra ID validates the signed nonce using the user's securely registered public key against the nonce signature. After validating the signature, Microsoft Entra ID then validates the returned signed nonce. After validating the nonce, Microsoft Entra ID creates a PRT with session key that is encrypted to the device's transport key and returns it to the Cloud AP provider. The Cloud AP provider receives the encrypted PRT with session key. Using the device's private transport key, the Cloud AP provider decrypt the session key and protects the session key using the device's TPM. The Cloud AP provider returns a successful authentication response to LSASS. LSASS caches the PRT.

ⓘ Important

In the above deployment model, a newly provisioned user will not be able to sign in using Windows Hello for Business until (a) Microsoft Entra Connect successfully synchronizes the public key to the on-premises Active Directory and (b) device has line of sight to the domain controller for the first time.

Microsoft Entra hybrid join authentication using a certificate



Expand table

Phase	Description
A	Authentication begins when the user dismisses the lock screen, which triggers Winlogon to show the Windows Hello for Business credential provider. The user provides their Windows Hello gesture (PIN or biometrics). The credential provider packages these credentials and returns them to Winlogon. Winlogon passes the collected credentials to LSASS. LSASS passes the collected credentials to the Kerberos security support provider. The Kerberos provider gets domain hints from the domain joined workstation to locate a domain controller for the user.
B	The Kerberos provider sends the signed preauthentication data and user's certificate, which includes the public key, to the Key Distribution Center (KDC) service running on the domain controller in the form of a KERB_AS_REQ. The domain controller determines the certificate isn't self-signed certificate. The domain controller ensures the certificate chains to trusted root certificate, is within its validity period, can be used for authentication, and hasn't been revoked. It retrieves the public key and UPN from the certificate included in the KERB_AS_REQ and searches for the UPN in Active Directory. It validates the signed preauthentication data using the public key from the certificate. On success, the KDC returns a TGT to the client with its certificate in a KERB_AS_REP.

Phase	Description
C	The Kerberos provider ensures it can trust the response from the domain controller. First, it ensures the KDC certificate chains to a root certificate that is trusted by the device. Next, it ensures the certificate is within its validity period and that it hasn't been revoked. The Kerberos provider then verifies the certificate has the KDC Authentication present and that the subject alternate name listed in the KDC's certificate matches the domain name to which the user is authenticating.
D	After passing this criteria, Kerberos returns the TGT to LSASS, where it's cached and used for subsequent service ticket requests.
E	LSASS informs Winlogon of the success authentication. Winlogon creates a logon session, loads the user's profile, and starts explorer.exe.
F	While Windows loads the user's desktop, LSASS passes the collected credentials to the Cloud Authentication security support provider, referred to as the Cloud AP provider. The Cloud AP provider requests a nonce from Microsoft Entra ID. Microsoft Entra ID returns a nonce.
G	The Cloud AP provider signs the nonce using the user's private key and returns the signed nonce to the Microsoft Entra ID. Microsoft Entra ID validates the signed nonce using the user's securely registered public key against the nonce signature. After validating the signature, Microsoft Entra ID then validates the returned signed nonce. After validating the nonce, Microsoft Entra ID creates a PRT with session key that is encrypted to the device's transport key and returns it to the Cloud AP provider. The Cloud AP provider receives the encrypted PRT with session key. Using the device's private transport key, the Cloud AP provider decrypt the session key and protects the session key using the device's TPM. The Cloud AP provider returns a successful authentication response to LSASS. LSASS caches the PRT.

Important

In this deployment model, a **newly provisioned** user will not be able to sign in using Windows Hello for Business unless the device has line of sight to the domain controller.

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure Windows Hello for Business

Applies to: ✅ Windows 11, ✅ Windows 10

This article describes the options to configure Windows Hello for Business in an organization, and how to implement them.

Configuration options

You can configure Windows Hello for Business by using the following options:

- Configuration Service Provider (CSP): commonly used for devices managed by a Mobile Device Management (MDM) solution, like Microsoft Intune. CSPs can also be configured with provisioning packages, which are usually used at deployment time or for unmanaged devices. To configure Windows Hello for Business, use the PassportForWork CSP
- Group policy (GPO): used for devices that are Active Directory joined or Microsoft Entra hybrid joined, and aren't managed by a device management solution

Policy precedence

Some of the Windows Hello for Business policies are available for both computer and user configuration. The following list describes the policy precedence for Windows Hello for Business:

- *User policies* take precedence over *computer policies*. If a user policy is set, the corresponded computer policy is ignored. If a user policy isn't set, the computer policy is used
- Windows Hello for Business policy settings are enforced using the following hierarchy:
  - User - GPO
  - Computer - GPO
  - User - PassportForWork CSP
  - Device - PassportForWork CSP
  - Exchange Active Sync - DeviceLock CSP

> **Important**
>
> If you configure password length and complexity settings defined by the DeviceLock CSP, and PIN length and complexity settings defined by the PassportForWork CSP, Windows enforces the strictest policy out of the set of governing policies.

The DeviceLock CSP utilizes the Exchange ActiveSync Policy (EAS) engine. For more information, see Exchange ActiveSync Policy Engine Overview.

> **Note**
>
> If a policy isn't explicitly configured to require letters or special characters, users can optionally set an alphanumeric PIN.

Retrieve the Microsoft Entra tenant ID

The configuration via CSP or registry of different Windows Hello for Business policy settings require to specify the Microsoft Entra tenant ID where the device is registered.

To look up your Tenant ID, see How to find your Microsoft Entra tenant ID or try the following, ensuring to sign in with your organization's account:

msgraph

```
GET https://graph.microsoft.com/v1.0/organization?$select=id
```

For example, the PassportForWork CSP documentation describes how to configure Windows Hello for Business options using the OMA-URI:

Device

```
./Device/Vendor/MSFT/PassportForWork/{TenantId}
```

When configuring devices, replace `TenantID` with your Microsoft Entra tenant ID. For example, if your Microsoft Entra tenant ID is `dcd219dd-bc68-4b9b-bf0b-4a33a796be35`, the OMA-URI would be:

Device

```
./Device/Vendor/MSFT/PassportForWork/{dcd219dd-bc68-4b9b-bf0b-4a33a796be35}
```

Configure Windows Hello for Business using Microsoft Intune

For Microsoft Entra joined devices and Microsoft Entra hybrid joined devices enrolled in Intune, you can use Intune policies to manage Windows Hello for Business.

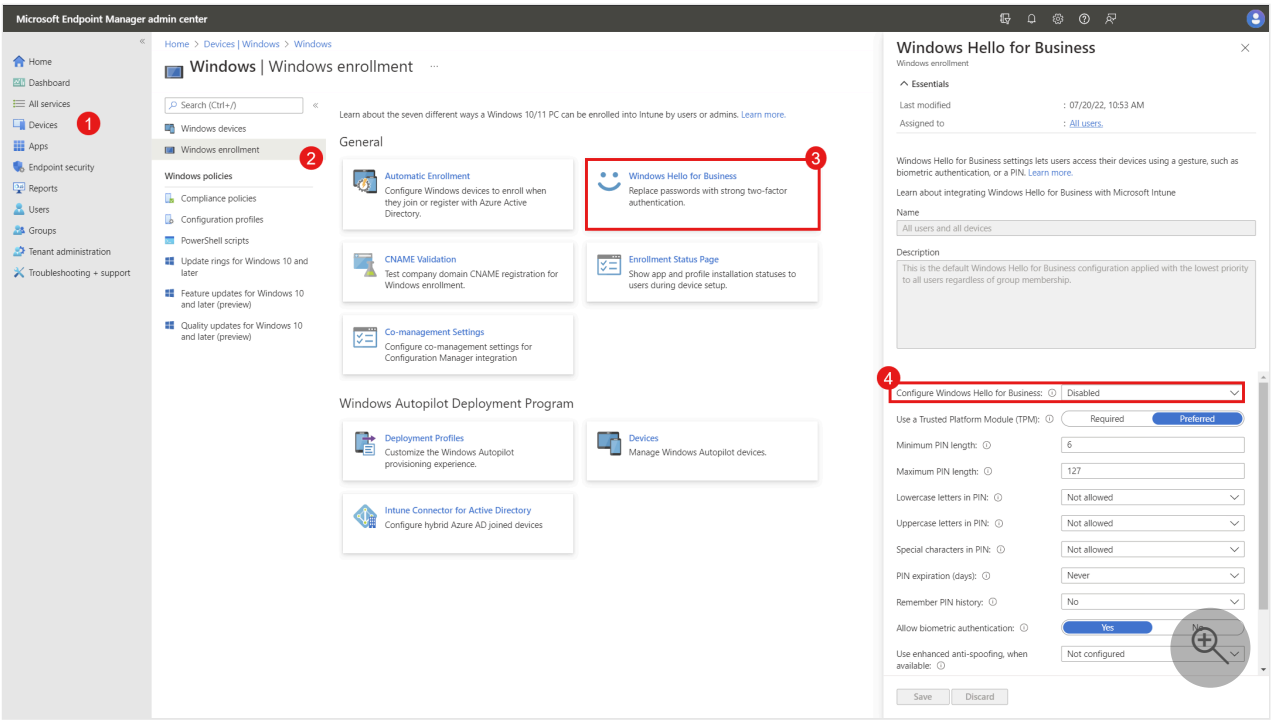
There are different ways to enable and configure Windows Hello for Business in Intune:

- Using a policy applied at the tenant level. The tenant policy:
  - Is only applied at enrollment time, and any changes to its configuration doesn't apply to devices already enrolled in Intune
  - It applies to *all devices* getting enrolled in Intune. For this reason, the policy is usually disabled and Windows Hello for Business is enabled using a policy targeted to a security group
- A device configuration policy that is applied *after* device enrollment. Any changes to the policy are applied to the devices during regular policy refresh intervals. There are different policy types to choose from:
  - Settings catalog
  - Security baselines
  - Custom policy, via the PassportForWork CSP
  - Account protection policy
  - Identity protection policy template

Verify the tenant-wide policy

To check the Windows Hello for Business policy settings applied at enrollment time:

1. Sign in to the Microsoft Intune admin center
2. Select **Devices** > **Windows** > **Windows Enrollment**
3. Select **Windows Hello for Business**
4. Verify the status of **Configure Windows Hello for Business** and any settings that might be configured



Policy conflicts from multiple policy sources

Windows Hello for Business can be configured by GPO or CSP, but not a combination of both. Avoid mixing GPO and CSP policy settings for Windows Hello for Business, as it can lead to unexpected results. If you mix GPO and CSP policy settings, the conflicting CSP settings aren't applied until the group policy settings are cleared.

> **Important**
>
> The *MDMWinsOverGP* policy setting doesn't apply to Windows Hello for Business. MDMWinsOverGP only applies to policies in the *Policy CSP*, while the Windows Hello for Business policies are in the *PassportForWork CSP*.

> **Note**
>
> For more information about deploying Windows Hello for Business configuration using Microsoft Intune, see Windows device settings to enable Windows Hello for Business in Intune and PassportForWork CSP.

Disable Windows Hello for Business enrollment

Windows Hello for Business is enabled by default for devices that are Microsoft Entra joined. If you need to disable the automatic enablement, there are different options, including:

- Disable Windows Hello using the tenant-wide policy
- Disable it using one of the policy types available in Intune, while enabling the Enrollment Status Page (ESP). The ESP can be configured to prevent a user from accessing the desktop until the device receives all the required policies. For more information, see Set up the Enrollment Status Page. The policy setting to configure is Use Windows Hello for Business
- Provision the devices using a provisioning package that disables Windows Hello for Business. For more information, see Provisioning packages for Windows
- Scripted solutions that can modify the registry settings to disable Windows Hello for Business during OS deployment

Expand table

Configuration type	Details
CSP (user)	**Key path**: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Policies\PassportForWork\<Tenant-ID>\UserSid\Policies` **Key name**: `UsePassportForWork` **Type**: `REG_DWORD` **Value**: `1` to enable `0` to disable
CSP (device)	**Key path**: `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Policies\PassportForWork\<Tenant-ID>\Device\Policies` **Key name**: `UsePassportForWork` **Type**: `REG_DWORD` **Value**: `1` to enable `0` to disable
GPO (user)	**Key path**: `HKEY_USERS\<UserSID>\SOFTWARE\Policies\Microsoft\PassportForWork` **Key name**: `Enabled` **Type**: `REG_DWORD` **Value**: `1` to enable `0` to disable
GPO (device)	**Key path**: `KEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\PassportForWork` **Key name**: `Enabled` **Type**: `REG_DWORD` **Value**: `1` to enable `0` to disable

> **Note**
>
> If there's a conflicting device policy and user policy, the user policy takes precedence. It's not recommended to create Local GPO or registry settings that could conflict with an MDM policy. This conflict could lead to unexpected results.

Next steps

For a list of Windows Hello for Business policy settings, see Windows Hello for Business policy settings.

To learn more about Windows Hello for Business features and how to configure them, see:

- PIN reset
- Dual enrollment
- Dynamic Lock
- Multi-factor Unlock
- Remote desktop (RDP) sign-in

Last updated on 03/29/2026

Plan a Windows Hello for Business deployment

Applies to: ✅ Windows 11, ✅ Windows 10

This planning guide helps you understand the different topologies, architectures, and components that encompass a Windows Hello for Business infrastructure.

This guide explains the role of each component within Windows Hello for Business and how certain deployment decisions affect other aspects of the infrastructure.

> **Tip**
>
> If you have a Microsoft Entra ID tenant, you can use our online, interactive Passwordless Wizard which walks through the same choices instead of using our manual guide below. The Passwordless Wizard is available in the Microsoft 365 admin center.

Using this guide

There are many options available for deploying Windows Hello for Business, ensuring compatibility with various organizational infrastructures. While the deployment process may appear complex, most organizations will find that they have already implemented the necessary infrastructure. It is important to note that Windows Hello for Business is a distributed system and requires proper planning across multiple teams within an organization.

This guide aims to simplify the deployment process by helping you make informed decisions about each aspect of your Windows Hello for Business deployment. It provides information on the options available and assists in selecting the deployment approach that best suits your environment.

How to proceed

Read this document and record your decisions. When finished, you should have all the necessary information to evaluate the available options and to determine requirements for your Windows Hello for Business deployment.

There are seven main areas to consider when planning a Windows Hello for Business deployment:

- ✓ Deployment options
- ✓ Public Key Infrastructure (PKI) requirements

- ✓ Authentication to Microsoft Entra ID requirements
- ✓ Device configuration options
- ✓ Licensing for cloud services requirements
- ✓ Operating System requirements
- ✓ Prepare users

Deployment options

The goal of Windows Hello for Business is to enable deployments for all organizations of any size or scenario. To provide this type of granular deployment, Windows Hello for Business offers a diverse choice of deployment options.

Deployment models

It's fundamentally important to understand which deployment model to use for a successful deployment. Some aspects of the deployment might have already been decided for you based on your current infrastructure.

There are three deployment models from which you can choose:

Expand table

	Deployment model	Description
☐	**Cloud-only**	For organizations that only have cloud identities and don't access on-premises resources. These organizations typically join their devices to the cloud and exclusively use resources in the cloud such as SharePoint Online, OneDrive, and others. Also, since the users don't use on-premises resources, they don't need certificates for things like VPN because everything they need is hosted in cloud services.
☐	**Hybrid**	For organizations that have identities synchronized from Active Directory to Microsoft Entra ID. These organizations use applications registered in Microsoft Entra ID, and want a single sign-on (SSO) experience for both on-premises and Microsoft Entra resources.
☐	**On-premises**	For organizations that don't have cloud identities or use applications hosted in Microsoft Entra ID. These organizations use on-premises applications, integrated in Active Directory, and want an SSO user experiences when accessing them.

> ⓘ Note

- Main use case of On-Premises deployment is for "Enhanced Security Administrative Environments" also known as "Red Forests"
- Migration from on-premise to hybrid deployment requires redeployment

Trust types

A deployment's trust type defines how Windows Hello for Business clients **authenticate to Active Directory**. The trust type doesn't affect authentication to Microsoft Entra ID. For this reason, the trust type isn't applicable to a cloud-only deployment model.

Windows Hello for Business authentication to Microsoft Entra ID always uses the key, not a certificate (excluding smart card authentication in a federated environment).

The trust type determines whether you issue authentication certificates to your users. One trust model isn't more secure than the other.

The deployment of certificates to users and domain controllers requires more configuration and infrastructure, which could also be a factor to consider in your decision. More infrastructure needed for certificate-trust deployments includes a certificate registration authority. In a federated environment, you must activate the Device Writeback option in Microsoft Entra Connect.

There are three trust types from which you can choose:

Expand table

	Trust type	Description
☐	**Cloud Kerberos**	Users authenticate to Active Directory by requesting a TGT from Microsoft Entra ID, using Microsoft Entra Kerberos. The on-premises domain controllers are still responsible for Kerberos service tickets and authorization. Cloud Kerberos trust uses the same infrastructure required for FIDO2 security key sign-in, and it can be used for new or existing Windows Hello for Business deployments.
☐	**Key**	Users authenticate to the on-premises Active Directory using a device-bound key (hardware or software) created during the Windows Hello provisioning experience. It requires to distribute certificates to domain controllers.
☐	**Certificate**	The certificate trust type issues authentication certificates to users. Users authenticate using a certificate requested using a device-bound key (hardware or software) created during the Windows Hello provisioning experience.

Key trust and *certificate trust* use certificate authentication-based Kerberos when requesting kerberos ticket-granting-tickets (TGTs) for on-premises authentication. This type of

authentication requires a PKI for DC certificates, and requires end-user certificates for certificate trust.

The goal of Windows Hello for Business cloud Kerberos trust is to provide a simpler deployment experience, when compared to the other trust types:

- No need to deploy a public key infrastructure (PKI) or to change an existing PKI
- No need to synchronize public keys between Microsoft Entra ID and Active Directory for users to access on-premises resources. There isn't any delay between the user's Windows Hello for Business provisioning, and being able to authenticate to Active Directory
- FIDO2 security key sign-in can be deployed with minimal extra setup

> **Tip**
>
> Windows Hello for Business cloud Kerberos trust is the recommended deployment model when compared to the *key trust model*. It is also the preferred deployment model if you do not need to support certificate authentication scenarios.

Cloud Kerberos trust requires the deployment of Microsoft Entra Kerberos. For more information about how Microsoft Entra Kerberos enables access to on-premises resources, see enabling passwordless security key sign-in to on-premises resources.

PKI requirements

Cloud Kerberos trust is the only hybrid deployment option that doesn't require the deployment of any certificates. The other hybrid and on-premises models depend on an enterprise PKI as a trust anchor for authentication:

- Domain controllers for hybrid and on-premises deployments need a certificate for Windows devices to trust the domain controller as legitimate
- Deployments using the certificate trust type require an enterprise PKI and a certificate registration authority (CRA) to issue authentication certificates to users. AD FS is used as a CRA
- Hybrid deployments might need to issue VPN certificates to users to enable connectivity on-premises resources

Expand table

	Deployment model	Trust type	PKI required?
☐	**Cloud-only**	n/a	no

	Deployment model	Trust type	PKI required?
☐	**Hybrid**	Cloud Kerberos	no
☐	**Hybrid**	Key	yes
☐	**Hybrid**	Certificate	yes
☐	**On-premises**	Key	yes
☐	**On-premises**	Certificate	yes

Authentication to Microsoft Entra ID

Users can authenticate to Microsoft Entra ID using federated authentication or cloud (nonfederated) authentication. Requirements vary based on trust type:

Expand table

	Deployment model	Trust type	Authentication to Microsoft Entra ID	Requirements
☐	**Cloud-only**	n/a	Cloud authentication	n/a
☐	**Cloud-only**	n/a	Federated authentication	Non-Microsoft federation service
☐	**Hybrid**	Cloud Kerberos trust	Cloud authentication	Password hash sync (PHS) or Pass-through authentication (PTA)
☐	**Hybrid**	Cloud Kerberos trust	Federated authentication	AD FS or non-Microsoft federation service
☐	**Hybrid**	Key trust	Cloud authentication	Password hash sync (PHS) or Pass-through authentication (PTA)
☐	**Hybrid**	Key trust	Federated authentication	AD FS or non-Microsoft federation service
☐	**Hybrid**	Certificate trust	Federated authentication	This deployment model doesn't support PTA or PHS. Active Directory must be federated with Microsoft Entra ID using AD FS

To learn more:

- Federation with Microsoft Entra ID

- Password hash synchronization (PHS)
- Pass-through authentication (PTA)

Device registration

For on-premises deployments, the server running the Active Directory Federation Services (AD FS) role is responsible for device registration. For cloud-only and hybrid deployments, devices must register in Microsoft Entra ID.

Expand table

Deployment model	Supported join type	Device registration service provider
Cloud-only	Microsoft Entra joined Microsoft Entra registered	Microsoft Entra ID
Hybrid	Microsoft Entra joined Microsoft Entra hybrid joined Microsoft Entra registered	Microsoft Entra ID
On-premises	Active Directory domain joined	AD FS

> **Important**
>
> For *Microsoft Entra hybrid joined* guidance, review **Plan your Microsoft Entra hybrid join implementation**.

Multifactor authentication

The goal of Windows Hello for Business is to move organizations away from passwords by providing them with a *strong credential* that enables easy two-factor authentication. The built-in provisioning experience accepts the user's weak credentials (username and password) as the first factor authentication. However, the user must provide a second factor of authentication before Windows provisions a strong credential:

- For cloud-only and hybrid deployments, there are different choices for multifactor authentication, including Microsoft Entra MFA
- On-premises deployments must use a multifactor option that can integrate as an AD FS multifactor adapter. Organizations can choose from non-Microsoft options that offer an AD FS MFA adapter. For more information, see Microsoft and non-Microsoft additional authentication methods

> **Important**
>
> Beginning July 1, 2019, Microsoft doesn't offer MFA Server for new deployments. New deployments that require multifactor authentication should use cloud-based Microsoft Entra multifactor authentication.
>
> Beginning September 30, 2024, Azure Multi-Factor Authentication Server deployments will no longer service MFA requests. To ensure uninterrupted authentication services and to remain in a supported state, organizations should migrate their users' authentication data to the cloud-based Azure MFA.

Expand table

	Deployment model	MFA options
☐	**Cloud-only**	Microsoft Entra MFA
☐	**Cloud-only**	Non-Microsoft MFA, via external authentication method in Microsoft Entra ID or federation
☐	**Hybrid**	Microsoft Entra MFA
☐	**Hybrid**	Non-Microsoft MFA, via external authentication method in Microsoft Entra ID or federation
☐	**On-premises**	AD FS MFA adapter

For more information:

- Configure Microsoft Entra multifactor authentication settings
- Manage an external authentication method in Microsoft Entra ID

MFA and federated authentication

It's possible for federated domains to configure the *FederatedIdpMfaBehavior* flag. The flag instructs Microsoft Entra ID to accept, enforce, or reject the MFA challenge from the federated IdP. For more information, see federatedIdpMfaBehavior values. To check this setting, use the following PowerShell command:

PowerShell

```
Connect-MgGraph
$DomainId = "<your federated domain name>"
```

```
Get-MgDomainFederationConfiguration -DomainId $DomainId |fl
```

To reject the MFA claim from the federated IdP, use the following command. This change impacts all MFA scenarios for the federated domain:

PowerShell

```
Update-MgDomainFederationConfiguration -DomainId $DomainId -FederatedIdpMfaBehavior
rejectMfaByFederatedIdp
```

If you configure the flag with a value of either `acceptIfMfaDoneByFederatedIdp` (default) or `enforceMfaByFederatedIdp`, you must verify that your federated IDP is correctly configured and working with the MFA adapter and provider used by your IdP.

Key registration

The built-in Windows Hello for Business provisioning experience creates a device-bound asymmetric key pair as the user's credentials. The private key is protected by the device's security modules. The credential is a *user key*, not a *device key*. The provisioning experience registers the user's public key with the identity provider:

Expand table

Deployment model	Key registration service provider
Cloud-only	Microsoft Entra ID
Hybrid	Microsoft Entra ID
On-premises	AD FS

Directory synchronization

Hybrid and on-premises deployments use directory synchronization, however, each for a different purpose:

- Hybrid deployments use Microsoft Entra Connect Sync to synchronize Active Directory identities (users and devices) or credentials between itself and Microsoft Entra ID. During the Window Hello for Business provisioning process, users register the public portion of their Windows Hello for Business credential with Microsoft Entra ID. Microsoft Entra Connect Sync synchronizes the Windows Hello for Business public key to Active Directory. This synchronization enables SSO to Microsoft Entra ID and its federated components.

> **Important**
>
> Windows Hello for Business is tied between a user and a device. Both the user and device object must be synchronized between Microsoft Entra ID and Active Directory.

- On-premises deployments use directory synchronization to import users from Active Directory to the Azure MFA server, which sends data to the MFA cloud service to perform the verification

Expand table

Deployment model	Directory sync options
Cloud-only	n/a
Hybrid	Microsoft Entra Connect Sync
On-premises	Azure MFA server

> **Important**
>
> Beginning September 30, 2024, Azure Multi-Factor Authentication Server deployments will no longer service MFA requests. To ensure uninterrupted authentication services and to remain in a supported state, organizations should migrate their users' authentication data to the cloud-based Azure MFA.

Device configuration options

Windows Hello for Business provides a rich set of granular policy settings. There are two main options to configure Windows Hello for Business: configuration service provider (CSP) and group policy (GPO).

- The CSP option is ideal for devices that are managed through a Mobile Device Management (MDM) solution, like Microsoft Intune. CSPs can also be configured with provisioning packages
- GPO can be used to configure domain joined devices and where devices aren't managed via MDM

Expand table

	Deployment model	Device configuration options
☐	**Cloud-only**	CSP
☐	**Cloud-only**	GPO (local)
☐	**Hybrid**	CSP
☐	**Hybrid**	GPO (Active Directory or local)
☐	**On-premises**	CSP
☐	**On-premises**	GPO (Active Directory or local)

Licensing for cloud services requirements

Here are some considerations regarding licensing requirements for cloud services:

- Windows Hello for Business doesn't require a Microsoft Entra ID P1 or P2 subscription. However, some dependencies, such as MDM automatic enrollment and Conditional Access do
  - Devices managed via MDM don't require a Microsoft Entra ID P1 or P2 subscription. By forgoing the subscription, users must manually enroll devices in the MDM solution, such as Microsoft Intune or a supported non-Microsoft MDM
- You can deploy Windows Hello for Business using the Microsoft Entra ID Free tier. All Microsoft Entra ID Free accounts can use Microsoft Entra multifactor authentication for the Windows passwordless features
  - Some Microsoft Entra multifactor authentication features require a license. For more information, see Features and licenses for Microsoft Entra multifactor authentication.
- Enrolling a certificate using the AD FS registration authority requires devices to authenticate to the AD FS server, which requires device write-back, a Microsoft Entra ID P1 or P2 feature

Expand table

	Deployment model	Trust type	Cloud services licenses (minimum)
☐	**Cloud-only**	n/a	not required
☐	**Hybrid**	Cloud Kerberos	not required
☐	**Hybrid**	Key	not required
☐	**Hybrid**	Certificate	Microsoft Entra ID P1
☐	**On-premises**	Key	Azure MFA, if used as MFA solution

	Deployment model	Trust type	Cloud services licenses (minimum)
☐	**On-premises**	Certificate	Azure MFA, if used as MFA solution

> **Important**
>
> Beginning September 30, 2024, Azure Multi-Factor Authentication Server deployments will no longer service MFA requests. To ensure uninterrupted authentication services and to remain in a supported state, organizations should migrate their users' authentication data to the cloud-based Azure MFA.

Operating System requirements

Windows requirements

All supported Windows (client) versions can be used with Windows Hello for Business. However, cloud Kerberos trust requires minimum versions:

Expand table

	Deployment model	Trust type	Windows version
☐	**Cloud-only**	n/a	All supported versions
☐	**Hybrid**	Cloud Kerberos	- Windows 10 21H2, with KB5010415 and later - Windows 11 21H2, with KB5010414 and later
☐	**Hybrid**	Key	All supported versions
☐	**Hybrid**	Certificate	All supported versions
☐	**On-premises**	Key	All supported versions
☐	**On-premises**	Certificate	All supported versions

Windows Server requirements

Windows Hello for Business can be used to authenticate against all supported Windows Server versions as a domain controller. However, cloud Kerberos trust requires minimum versions:

Expand table

	Deployment model	Trust type	Domain controller OS version
☐	**Cloud-only**	n/a	All supported versions
☐	**Hybrid**	Cloud Kerberos	- Windows Server 2016, with KB3534307 and later - Windows Server 2019, with KB4534321 and later - Windows Server 2022 - Windows Server 2025
☐	**Hybrid**	Key	All supported versions
☐	**Hybrid**	Certificate	All supported versions
☐	**On-premises**	Key	All supported versions
☐	**On-premises**	Certificate	All supported versions

The minimum required domain functional and forest functional levels are Windows Server 2008 R2 for all deployment models.

Prepare users

When you are ready to enable Windows Hello for Business in your organization, make sure to prepare the users by explaining how to provision and use Windows Hello.

To learn more, see Prepare users.

Next steps

Now that you've read about the different deployment options and requirements, you can choose the implementation that best suits your organization.

Last updated on 03/29/2026

Cloud-only deployment guide

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** cloud-only ⓘ
- **Join type:** Microsoft Entra join ⓘ

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Authentication
- ✓ Device configuration
- ✓ Licensing for cloud services
- ✓ Prepare users to use Windows Hello

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Configure Windows Hello for Business policy settings

When you Microsoft Entra join a device, the system attempts to automatically enroll you in Windows Hello for Business. If you want to use Windows Hello for Business in a cloud-only environment with its default settings, there's no extra configuration needed.

Cloud-only deployments use Microsoft Entra multifactor authentication (MFA) during Windows Hello for Business enrollment, and there's no other MFA configuration needed.

If you aren't already registered in MFA, you're guided through the MFA registration as part of the Windows Hello for Business enrollment process.

Policy settings can be configured to control the behavior of Windows Hello for Business, via configuration service provider (CSP) or group policy (GPO). In cloud-only deployments, devices are typically configured via an MDM solution like Microsoft Intune, using the PassportForWork CSP.

> **Note**
>
> Review the article Configure Windows Hello for Business using Microsoft Intune to learn about the different options offered by Microsoft Intune to configure Windows Hello for Business.

If the Intune tenant-wide policy is configured to *disable Windows Hello for Business*, or if devices are deployed with Windows Hello disabled, you must configure one policy setting to enable Windows Hello for Business:

- Use Windows Hello for Business

Another optional, but recommended, policy setting is:

- Use a hardware security device

Follow the instructions below to configure your devices using either Microsoft Intune or group policy (GPO).

Intune/CSP

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Windows Hello for Business	Use Windows Hello For Business	true
Windows Hello for Business	Require Security Device	true

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the PassportForWork CSP.

Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/UsePassportForWork` - **Data type:** `bool` - **Value:** `True`
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/RequireSecurityDevice` - **Data type:** `bool` - **Value:** `True`

Tip

If you're using Microsoft Intune, and you're not using the **tenant-wide policy**, enable the Enrollment Status Page (ESP) to ensure that the devices receive the Windows Hello for Business policy settings before users can access their desktop. For more information about ESP, see **Set up the Enrollment Status Page**.

More policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after a user signs in, if certain prerequisite checks are passed.

User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**

3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use their PIN to sign-in. The user may close the provisioning application and access their desktop

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fcloud-only

Sequence diagrams

To better understand the provisioning flows, review the following sequence diagrams based on the authentication type:

- Provisioning for Microsoft Entra joined devices with managed authentication
- Provisioning for Microsoft Entra joined devices with federated authentication

To better understand the authentication flows, review the following sequence diagram:

- Microsoft Entra join authentication to Microsoft Entra ID

Disable automatic enrollment

If you want to disable the automatic Windows Hello for Business enrollment, you can configure your devices with a policy setting or registry key. For more information, see Disable Windows Hello for Business enrollment.

> **Note**
>
> During the out-of-box experience (OOBE) flow of a Microsoft Entra join, you are guided to enroll in Windows Hello for Business when you don't have Intune. You

can cancel the PIN screen and access the desktop without enrolling in Windows Hello for Business.

Feedback

Was this page helpful? Yes No

Provide product feedback

Cloud Kerberos trust deployment guide

Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** cloud Kerberos trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Authentication
- ✓ Device configuration
- ✓ Windows requirements
- ✓ Windows Server requirements
- ✓ Prepare users to use Windows Hello

> **Important**
>
> When implementing the cloud Kerberos trust deployment model, you *must* ensure that you have an adequate number of *read-write domain controllers* in each Active Directory site where users will be authenticating with Windows Hello for Business. For more information, see Capacity planning for Active Directory.

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Deploy Microsoft Entra Kerberos
- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Deploy Microsoft Entra Kerberos

If you've already deployed on-premises SSO for passwordless security key sign-in, then Microsoft Entra Kerberos is already deployed in your organization. You don't need to redeploy or change your existing Microsoft Entra Kerberos deployment to support Windows Hello for Business, and you can skip to the Configure Windows Hello for Business policy settings section.

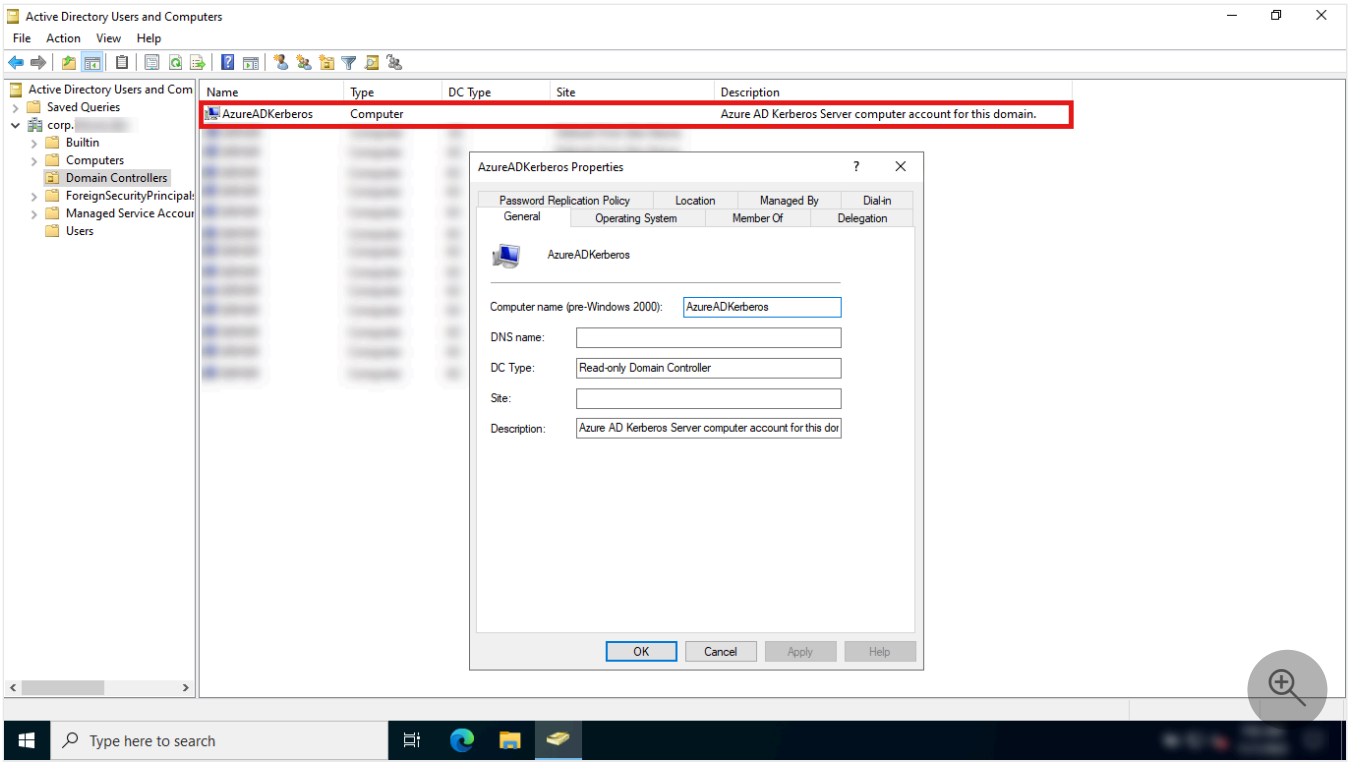
If you haven't deployed Microsoft Entra Kerberos, follow the instructions in the Enable passwordless security key sign-in documentation. This page includes information on how to install and use the Microsoft Entra Kerberos PowerShell module. Use the module to create a Microsoft Entra Kerberos server object for the domains where you want to use Windows Hello for Business cloud Kerberos trust.

Microsoft Entra Kerberos and cloud Kerberos trust authentication

When Microsoft Entra Kerberos is enabled in an Active Directory domain, an *AzureADKerberos* computer object is created in the domain. This object:

- Appears as a read only domain controller (RODC) object, but isn't associated with any physical servers
- Is only used by Microsoft Entra ID to generate TGTs for the Active Directory domain

> ⓘ **Note**
>
> Similar rules and restrictions used for RODCs apply to the AzureADKerberos computer object. For example, users that are direct or indirect members of privileged built-in security groups won't be able to use cloud Kerberos trust.



For more information about how Microsoft Entra Kerberos works with Windows Hello for Business cloud Kerberos trust, see Windows Hello for Business authentication technical deep dive.

> **Note**
>
> The default *Password Replication Policy* configured on the AzureADKerberos computer object doesn't allow to sign high privilege accounts on to on-premises resources with cloud Kerberos trust or FIDO2 security keys.
>
> Due to possible attack vectors from Microsoft Entra ID to Active Directory, it's not recommended to unblock these accounts by relaxing the Password Replication Policy of the computer object `CN=AzureADKerberos,OU=Domain Controllers,<domain-DN>`.

Configure Windows Hello for Business policy settings

After setting up the Microsoft Entra Kerberos object, Windows Hello for business must be enabled and configured to use cloud Kerberos trust. There are two policy settings required to configure Windows Hello for Business in a cloud Kerberos trust model:

- Use Windows Hello for Business
- Use cloud trust for on-premises authentication

Another optional, but recommended, policy setting is:

- Use a hardware security device

> **Important**
>
> If the **Use certificate for on-premises authentication** policy is enabled, certificate trust takes precedence over cloud Kerberos trust. Ensure that the machines that you want to enable cloud Kerberos trust have this policy **not configured**.

The following instructions explain how to configure your devices using either Microsoft Intune or group policy (GPO).

Intune/CSP

> **Note**
>
> Review the article Configure Windows Hello for Business using Microsoft Intune to learn about the different options offered by Microsoft Intune to configure Windows Hello for Business.

If the Intune tenant-wide policy is enabled and configured to your needs, you only need to enable the policy setting **Use Cloud Trust For On Prem Auth**. Otherwise, both settings must be configured.

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Windows Hello for Business	Use Windows Hello For Business	true
Windows Hello for Business	Use Cloud Trust For On Prem Auth	Enabled
Windows Hello for Business	Require Security Device	true

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the PassportForWork CSP.

Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/UsePassportForWork` - **Data type:** `bool` - **Value:** `True`
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/UseCloudTrustForOnPremAuth` - **Data type:** `bool` - **Value:** `True`
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/RequireSecurityDevice` - **Data type:** `bool` - **Value:** `True`

If you deploy Windows Hello for Business configuration using both Group Policy and Intune, Group Policy settings take precedence, and Intune settings are ignored. For more information about policy conflicts, see Policy conflicts from multiple policy sources.

More policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after a user signs in, if the prerequisite checks pass. Windows Hello for Business *cloud Kerberos trust* adds a prerequisite check for Microsoft Entra hybrid joined devices when cloud Kerberos trust is enabled by policy.

You can determine the status of the prerequisite check by viewing the **User Device Registration** admin log under **Applications and Services Logs** > **Microsoft** > **Windows**. This information is also available using the `dsregcmd.exe /status` command from a console. For more information, see dsregcmd.

The cloud Kerberos trust prerequisite check detects whether the user has a partial TGT before allowing provisioning to start. The purpose of this check is to validate whether Microsoft Entra Kerberos is set up for the user's domain and tenant. If Microsoft Entra Kerberos is set up, the user receives a partial TGT during sign-in with one of their other unlock methods. This check has three states: Yes, No, and Not Tested. The *Not Tested* state is reported if cloud Kerberos trust isn't enforced by policy or if the device is Microsoft Entra joined.

> **Note**
>
> The cloud Kerberos trust prerequisite check isn't done on Microsoft Entra joined devices. If Microsoft Entra Kerberos isn't provisioned, a user on a Microsoft Entra joined device will still be able to sign in, but won't have SSO to on-premises resources secured by Active Directory.

User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**
3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use their PIN to sign-in. The user may close the provisioning application and access their desktop

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fhybrid-cloud-kerberos-trust

Once a user completes enrollment with cloud Kerberos trust, the Windows Hello gesture can be used **immediately** for sign-in. On a Microsoft Entra hybrid joined device, the first use of the PIN requires line of sight to a DC. Once the user signs in or unlocks with the DC, cached sign-in can be used for subsequent unlocks without line of sight or network connectivity.

After enrollment, Microsoft Entra Connect synchronizes the user's key from Microsoft Entra ID to Active Directory.

Sequence diagrams

To better understand the provisioning flows, review the following sequence diagrams based on the device join and authentication type:

- Provisioning for Microsoft Entra joined devices with managed authentication
- Provisioning for Microsoft Entra joined devices with federated authentication
- Provisioning in a cloud Kerberos trust deployment model with managed authentication

To better understand the authentication flows, review the following sequence diagram:

- Microsoft Entra join authentication to Active Directory using cloud Kerberos trust

Migrate from key trust deployment model to cloud Kerberos trust

If you deployed Windows Hello for Business using the key trust model, and want to migrate to the cloud Kerberos trust model, follow these steps:

1. Set up Microsoft Entra Kerberos in your hybrid environment
2. Enable cloud Kerberos trust via Group Policy or Intune
3. For Microsoft Entra joined devices, sign out and sign in to the device using Windows Hello for Business

> **Note**
>
> For Microsoft Entra hybrid joined devices, users must perform the first sign in with new credentials while having line of sight to a DC.

Migrate from certificate trust deployment model to cloud Kerberos trust

> **Important**
>
> There is no *direct* migration path from a certificate trust deployment to a cloud Kerberos trust deployment. The Windows Hello container must be deleted before you can migrate

to cloud Kerberos trust.

If you deployed Windows Hello for Business using the certificate trust model, and want to use the cloud Kerberos trust model, you must redeploy Windows Hello for Business by following these steps:

1. Disable the certificate trust policy.
2. Enable cloud Kerberos trust via Group Policy or Intune.
3. Remove the certificate trust credential using the command `certutil.exe -deletehellocontainer` from the user context.
4. Sign out and sign back in.
5. Provision Windows Hello for Business using a method of your choice.

> **Note**
>
> For Microsoft Entra hybrid joined devices, users must perform the first sign-in with new credentials while having line of sight to a DC.

Frequently Asked Questions

For a list of frequently asked questions about Windows Hello for Business cloud Kerberos trust, see Windows Hello for Business Frequently Asked Questions.

Unsupported scenarios

The following scenarios aren't supported using Windows Hello for Business cloud Kerberos trust:

- RDP/VDI scenarios using supplied credentials (RDP/VDI can be used with Remote Credential Guard or if a certificate is enrolled into the Windows Hello for Business container)
- Using cloud Kerberos trust for *Run as*
- Signing in with cloud Kerberos trust on a Microsoft Entra hybrid joined device without previously signing in with DC connectivity

---

Last updated on 03/29/2026

Hybrid key trust deployment guide

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** key trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

> **Important**
>
> Windows Hello for Business *cloud Kerberos trust* is the recommended deployment model when compared to the *key trust model*. For more information, see **cloud Kerberos trust deployment**.

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Public Key Infrastructure
- ✓ Authentication
- ✓ Device configuration
- ✓ Prepare users to use Windows Hello

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure and validate the Public Key Infrastructure
- ✓ Configure and enroll in Windows Hello for Business
- ✓ (optional) Configure single sign-on for Microsoft Entra joined devices

Configure and validate the Public Key Infrastructure

Windows Hello for Business must have a Public Key Infrastructure (PKI) when using the *key trust* model. The domain controllers must have a certificate, which serves as a *root of trust* for clients. The certificate ensures that clients don't communicate with rogue domain controllers.

Key trust deployments don't need client-issued certificates for on-premises authentication. *Microsoft Entra Connect Sync* configures Active Directory user accounts for public key mapping, by synchronizing the public key of the Windows Hello for Business credential to an attribute on the user's Active Directory object (`msDS-KeyCredentialLink` attribute).

A Windows Server-based PKI or a non-Microsoft Enterprise certification authority can be used. For more information, see Requirements for domain controller certificates from a non-Microsoft CA.

Deploy an enterprise certification authority

This guide assumes most enterprises have an existing public key infrastructure. Windows Hello for Business depends on an enterprise PKI running the Windows Server *Active Directory Certificate Services* role.
If you don't have an existing PKI, review Certification Authority Guidance to properly design your infrastructure. Then, consult the Test Lab Guide: Deploying an AD CS Two-Tier PKI Hierarchy for instructions on how to configure your PKI using the information from your design session.

Lab-based PKI

The following instructions may be used to deploy simple public key infrastructure that is suitable **for a lab environment**.

Sign in using *Enterprise Administrator* equivalent credentials on a Windows Server where you want the certification authority (CA) installed.

> ⓘ **Note**
>
> Never install a certification authority on a domain controller in a production environment.

1. Open an elevated Windows PowerShell prompt
2. Use the following command to install the Active Directory Certificate Services role.

```
PowerShell

Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
```

3. Use the following command to configure the CA using a basic certification authority configuration

```
PowerShell

Install-AdcsCertificationAuthority
```

Configure the enterprise PKI

Configure domain controller certificates

Clients must trust the domain controllers, and the best way to enable the trust is to ensure that each domain controller has a *Kerberos Authentication* certificate. Installing a certificate on the domain controllers enables the Key Distribution Center (KDC) to prove its identity to other members of the domain. The certificates provide clients a root of trust external to the domain, namely the *enterprise certification authority*.

Domain controllers automatically request a *domain controller certificate* (if published) when they discover an enterprise CA is added to Active Directory. The certificates based on the *Domain Controller* and *Domain Controller Authentication* certificate templates don't include the *KDC Authentication* object identifier (OID), which was later added to the Kerberos RFC. Therefore, domain controllers need to request a certificate based on the *Kerberos Authentication* certificate template.

By default, the Active Directory CA provides and publishes the *Kerberos Authentication* certificate template. The cryptography configuration included in the template is based on older and less performant cryptography APIs. To ensure domain controllers request the proper certificate with the best available cryptography, use the *Kerberos Authentication* certificate template as a *baseline* to create an updated domain controller certificate template.

> **Important**

The certificates issued to the domain controllers must meet the following requirements:

- The *Certificate Revocation List (CRL) distribution point* extension must point to a valid CRL, or an *Authority Information Access (AIA)* extension that points to an Online Certificate Status Protocol (OCSP) responder
- Optionally, the certificate *Subject* section could contain the directory path of the server object (the distinguished name)
- The certificate *Key Usage* section must contain *Digital Signature* and *Key Encipherment*
- Optionally, the certificate *Basic Constraints* section should contain: `[Subject Type=End Entity, Path Length Constraint=None]`
- The certificate *extended key usage* section must contain Client Authentication (`1.3.6.1.5.5.7.3.2`), Server Authentication (`1.3.6.1.5.5.7.3.1`), and KDC Authentication (`1.3.6.1.5.2.3.5`)
- The certificate *Subject Alternative Name* section must contain the Domain Name System (DNS) name
- The certificate template must have an extension that has the value `DomainController`, encoded as a BMPstring. If you are using Windows Server Enterprise Certificate Authority, this extension is already included in the domain controller certificate template
- The domain controller certificate must be installed in the local computer's certificate store

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Kerberos Authentication** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Domain Controller Authentication (Kerberos)* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **None** from the **Subject name format** list • Select **DNS name** from the **Include this information in alternate subject** list • Clear all other items
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

> ⓘ **Note**
>
> Inclusion of the *KDC Authentication* OID in domain controller certificate is not required for Microsoft Entra hybrid joined devices. The OID is required for enabling authentication with Windows Hello for Business to on-premises resources by Microsoft Entra joined devices.

> ⓘ **Important**
>
> For Microsoft Entra joined devices to authenticate to on-premises resources, ensure to:
>
> - Install the root CA certificate in the device's trusted root certificate store. See how to deploy a trusted certificate profile via Intune

- Publish your certificate revocation list to a location that is available to Microsoft Entra joined devices, such as a web-based URL

Supersede existing domain controller certificates

The domain controllers may have an existing domain controller certificate. The Active Directory Certificate Services provides a default certificate template for domain controllers called *domain controller certificate*. Later releases of Windows Server provided a new certificate template called *domain controller authentication certificate*. These certificate templates were provided prior to the update of the Kerberos specification that stated Key Distribution Centers (KDCs) performing certificate authentication needed to include the *KDC Authentication* extension.

The *Kerberos Authentication* certificate template is the most current certificate template designated for domain controllers, and should be the one you deploy to all your domain controllers.
The *autoenrollment* feature allows you to replace the domain controller certificates. Use the following configuration to replace older domain controller certificates with new ones, using the *Kerberos Authentication* certificate template.

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the *Domain Controller Authentication (Kerberos)* (or the name of the certificate template you created in the previous section) template in the details pane and select **Properties**
4. Select the **Superseded Templates** tab. Select **Add**
5. From the **Add Superseded Template** dialog, select the *Domain Controller* certificate template and select **OK** > **Add**
6. From the **Add Superseded Template** dialog, select the *Domain Controller Authentication* certificate template and select **OK**
7. From the **Add Superseded Template** dialog, select the *Kerberos Authentication* certificate template and select **OK**
8. Add any other enterprise certificate templates that were previously configured for domain controllers to the **Superseded Templates** tab
9. Select **OK** and close the **Certificate Templates** console

The certificate template is configured to supersede all the certificate templates provided in the *superseded templates* list.

However, the certificate template and the superseding of certificate templates isn't active until the template is published to one or more certificate authorities.

> ⓘ **Note**
>
> The domain controller's certificate must chain to a root in the NTAuth store. By default, the Active Directory Certificate Authority's root certificate is added to the NTAuth store. If you are using a non-Microsoft CA, this may not be done by default. If the domain controller certificate does not chain to a root in the NTAuth store, user authentication will fail. To see all certificates in the NTAuth store, use the following command:
>
> `Certutil -viewstore -enterprise NTAuth`

Unpublish Superseded Certificate Templates

The certification authority only issues certificates based on published certificate templates. For security, it's a good practice to unpublish certificate templates that the CA isn't configured to issue, including the pre-published templates from the role installation and any superseded templates.

The newly created *domain controller authentication* certificate template supersedes previous domain controller certificate templates. Therefore, you need to unpublish these certificate templates from all issuing certificate authorities.

Sign in to the CA or management workstation with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane > **Certificate Templates**
3. Right-click the *Domain Controller* certificate template and select **Delete**. Select **Yes** on the **Disable certificate templates** window
4. Repeat step 3 for the *Domain Controller Authentication* and *Kerberos Authentication* certificate templates

Publish the certificate template to the CA

A certification authority can only issue certificates for certificate templates that are published to it. If you have more than one CA, and you want more CAs to issue certificates based on the certificate template, then you must publish the certificate template to them.

Sign in to the CA or management workstations with **Enterprise Admin** equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane
3. Select **Certificate Templates** in the navigation pane
4. Right-click the **Certificate Templates** node. Select **New** > **Certificate Template to issue**
5. In the **Enable Certificates Templates** window, select the *Domain Controller Authentication (Kerberos)* template you created in the previous steps > select **OK**
6. Close the console

> **Important**
>
> If you plan to deploy **Microsoft Entra joined** devices, and require single sign-on (SSO) to on-premises resources when signing in with Windows Hello for Business, follow the procedures to update your CA to include an http-based CRL distribution point.

Configure and deploy certificates to domain controllers

Configure automatic certificate enrollment for the domain controllers

Domain controllers automatically request a certificate from the *Domain controller certificate* template. However, domain controllers are unaware of newer certificate templates or superseded configurations on certificate templates. For domain controllers to automatically enroll and renew of certificates, configure a GPO for automatic certificate enrollment, and link it to the *Domain Controllers* OU.

1. Open the **Group Policy Management Console** (gpmc.msc)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type *Domain Controller Auto Certificate Enrollment* in the name box and select **OK**
5. Right-click the **Domain Controller Auto Certificate Enrollment** Group Policy object and select **Edit**
6. In the navigation pane, expand **Policies** under **Computer Configuration**

7. Expand **Windows Settings** > **Security Settings** > **Public Key Policies**
8. In the details pane, right-click **Certificate Services Client - Auto-Enrollment** and select **Properties**
9. Select **Enabled** from the **Configuration Model** list
10. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box
11. Select the **Update certificates that use certificate templates** check box
12. Select **OK**
13. Close the **Group Policy Management Editor**

Deploy the domain controller auto certificate enrollment GPO

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and expand the node with the Active Directory domain name. Right-click the **Domain Controllers** organizational unit and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select *Domain Controller Auto Certificate Enrollment* or the name of the domain controller certificate enrollment Group Policy object you previously created
4. Select **OK**

Validate the configuration

Windows Hello for Business is a distributed system, which on the surface appears complex and difficult. The key to a successful deployment is to validate phases of work prior to moving to the next phase.

Confirm your domain controllers enroll the correct certificates and not any superseded certificate templates. Check that each domain controller completed the certificate autoenrollment.

Use the event logs

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Using the Event Viewer, navigate to the **Application and Services** > **Microsoft** > **Windows** > **CertificateServices-Lifecycles-System** event log
2. Look for an event indicating a new certificate enrollment (autoenrollment):

    - The details of the event include the certificate template on which the certificate was issued
    - The name of the certificate template used to issue the certificate should match the certificate template name included in the event
    - The certificate thumbprint and EKUs for the certificate are also included in the event
    - The EKU needed for proper Windows Hello for Business authentication is Kerberos Authentication, in addition to other EKUs provide by the certificate template

Certificates superseded by your new domain controller certificate generate an *archive event* in the Event Log. The archive event contains the certificate template name and thumbprint of the certificate that was superseded by the new certificate.

Certificate Manager

You can use the Certificate Manager console to validate the domain controller has the properly enrolled certificate based on the correct certificate template with the proper EKUs. Use **certlm.msc** to view certificate in the local computers certificate stores. Expand the **Personal** store and view the certificates enrolled for the computer. Archived certificates don't appear in Certificate Manager.

Certutil.exe

You can use `certutil.exe` command to view enrolled certificates in the local computer. Certutil shows enrolled and archived certificates for the local computer. From an elevated command prompt, run the following command:

Windows Command Prompt

```
certutil.exe -q -store my
```

To view detailed information about each certificate in the store, and to validate automatic certificate enrollment enrolled the proper certificates, use the following command:

Windows Command Prompt

```
certutil.exe -q -v -store my
```

Troubleshooting

Windows triggers automatic certificate enrollment for the computer during boot, and when Group Policy updates. You can refresh Group Policy from an elevated command prompt using `gpupdate.exe /force`.

Alternatively, you can forcefully trigger automatic certificate enrollment using `certreq.exe -autoenroll -q` from an elevated command prompt.

Use the event logs to monitor certificate enrollment and archive. Review the configuration, such as publishing certificate templates to issuing certification authority and the *allow* auto enrollment permissions.

Section review and next steps

Before moving to the next section, ensure the following steps are complete:

- ✓ Configure domain controller certificate template
- ✓ Supersede existing domain controller certificates
- ✓ Unpublish superseded certificate templates
- ✓ Publish the certificate template to the CA
- ✓ Deploy certificates to the domain controllers
- ✓ Validate the domain controllers configuration

Next: configure and enroll in Windows Hello for Business >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure and enroll in Windows Hello for Business in a hybrid key trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** key trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

Once the prerequisites are met and the PKI configuration is validated, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Configure Windows Hello for Business policy settings

There's one policy setting required to enable Windows Hello for Business in a key trust model:

- Use Windows Hello for Business

Another optional, but recommended, policy setting is:

- Use a hardware security device

The following instructions describe how to configure your devices using either Microsoft Intune or group policy (GPO).

Intune/CSP

> **Note**
>
> Review the article Configure Windows Hello for Business using Microsoft Intune to learn about the different options offered by Microsoft Intune to configure Windows Hello for Business.

If the Intune tenant-wide policy is enabled and configured to your needs, you can skip to Enroll in Windows Hello for Business.

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Windows Hello for Business	Use Windows Hello For Business	true
Windows Hello for Business	Require Security Device	true

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the PassportForWork CSP.

Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/UsePassportForWork` - **Data type:** `bool` - **Value:** `True`
- **OMA-URI:** `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/RequireSecurityDevice` - **Data type:** `bool` - **Value:** `True`

If you deploy Windows Hello for Business configuration using both Group Policy and Intune, Group Policy settings take precedence, and Intune settings are ignored. For more information about policy conflicts, see Policy conflicts from multiple policy sources

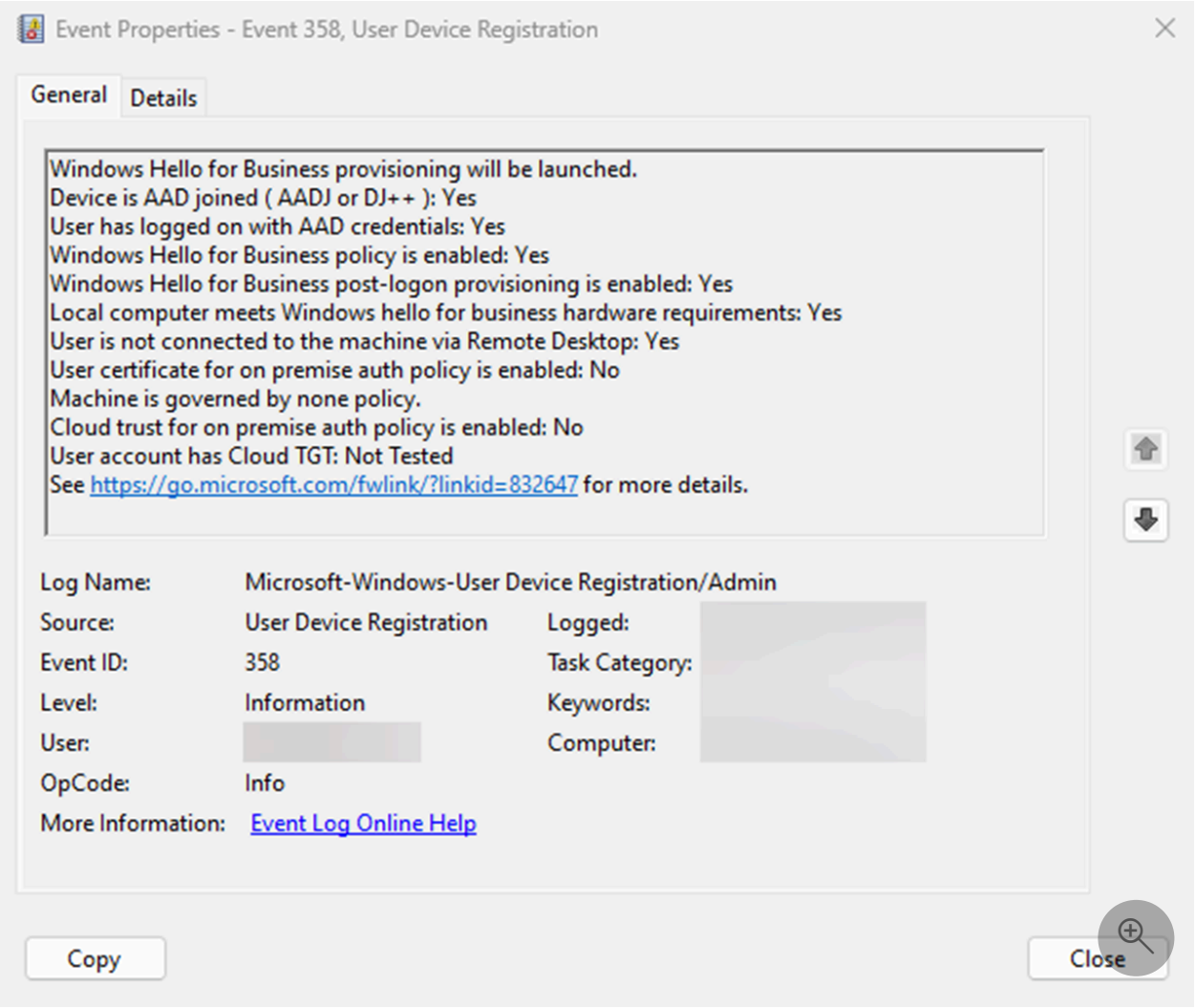
Other policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after the user profile is loaded and before the user receives their desktop. For the provisioning process

to begin, all prerequisite checks must pass.

You can determine the status of the prerequisite checks by viewing the **User Device Registration** admin log under **Applications and Services Logs** > **Microsoft** > **Windows**. This information is also available using the `dsregcmd.exe /status` command from a console. For more information, see dsregcmd.



User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**
3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't

proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use their PIN to sign-in. The user may close the provisioning application and access their desktop

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fhybrid-key-trust-enroll

After enrollment, Microsoft Entra Connect synchronizes the user's key from Microsoft Entra ID to Active Directory.

> **Important**
>
> The minimum time needed to synchronize the user's public key from Microsoft Entra ID to the on-premises Active Directory is 30 minutes. The Microsoft Entra Connect scheduler controls the synchronization interval. **This synchronization latency delays the user's ability to authenticate and use on-premises resources until the user's public key has synchronized to Active Directory.** Once synchronized, the user can authenticate and access on-premises resources. Read Microsoft Entra Connect Sync: Scheduler to view and adjust the **synchronization cycle** for your organization.

Sequence diagrams

To better understand the provisioning flows, review the following sequence diagrams based on the device join and authentication type:

- Provisioning for Microsoft Entra joined devices with managed authentication
- Provisioning for Microsoft Entra joined devices with federated authentication
- Provisioning in a hybrid key trust deployment model with managed authentication

To better understand the authentication flows, review the following sequence diagram:

- Microsoft Entra hybrid join authentication using a key

- Microsoft Entra join authentication to Active Directory using a key

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure single sign-on for Microsoft Entra joined devices

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** key trust ,certificate trust
- **Join type:** Microsoft Entra join

Windows Hello for Business combined with Microsoft Entra joined devices makes it easy for users to securely access cloud-based resources using a strong, two-factor credential. As organizations transition resources to the cloud, some resources might remain on-premises, and Microsoft Entra joined devices might need to access them. With additional configurations to the hybrid deployment, you can provide single sign-on to on-premises resources for Microsoft Entra joined devices using Windows Hello for Business, using a key or a certificate.

> **Note**
>
> These steps are not needed when using the cloud Kerberos trust model.

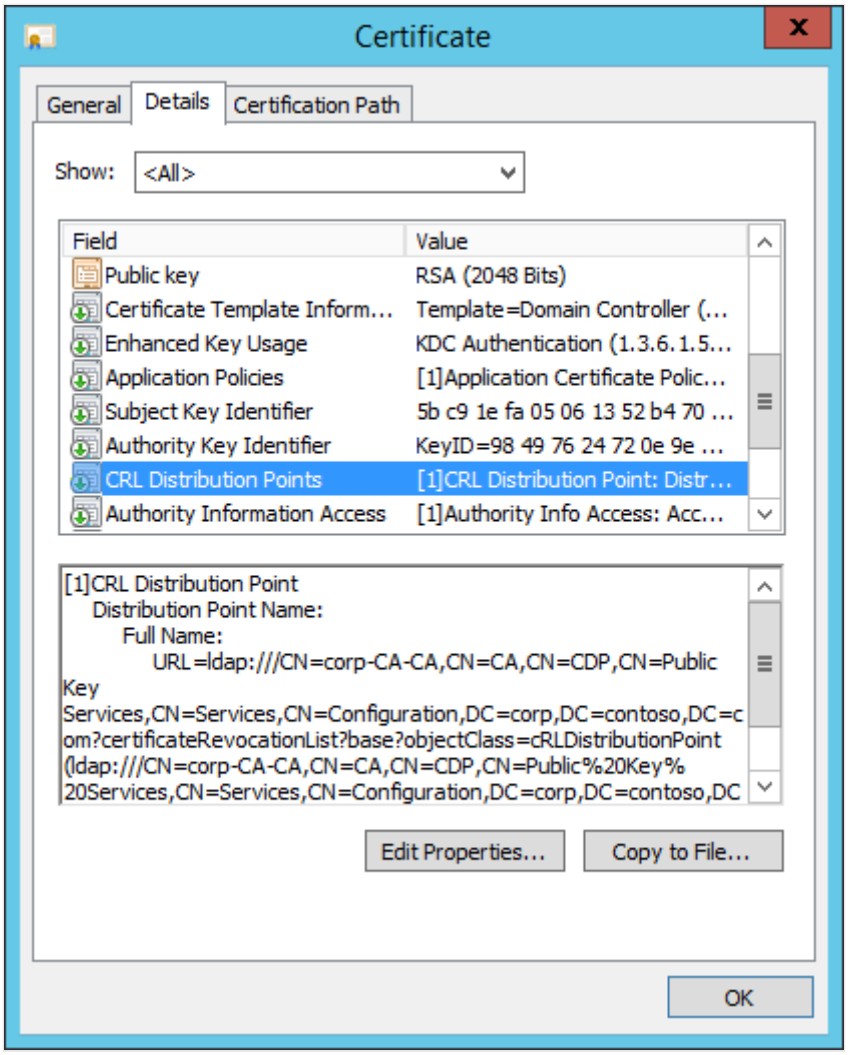
Prerequisites

Unlike Microsoft Entra hybrid joined devices, Microsoft Entra joined devices don't have a relationship with your Active Directory domain. This factor changes the way in which users authenticate to Active Directory. Validate the following configurations to ensure they support Microsoft Entra joined devices:

- ✓ Certificate Revocation List (CRL) Distribution Point
- ✓ Domain controller certificates
- ✓ Network infrastructure in place to reach the on-premises domain controllers. If the machines are external, you can use any VPN solution

CRL Distribution Point (CDP)

Certificates issued by a certificate authority can be revoked. When a certificate authority revokes a certificate, it writes information about the certificate into a *certificate revocation list* (CRL).
During certificate validation, Windows compares the current certificate with information in the CRL to determine if the certificate is valid.



In the screenshot, the CDP property of the domain controller certificate shows an LDAP path. Using Active Directory for domain joined devices provides a highly available CRL distribution point. However, Microsoft Entra joined devices can't read data from Active Directory, and certificate validation doesn't provide an opportunity to authenticate prior to reading the CRL. The authentication becomes a circular problem: the user is attempting to authenticate, but must read Active Directory to complete the authentication, but the user can't read Active Directory because they haven't authenticated.

To resolve this issue, the CRL distribution point must be a location accessible by Microsoft Entra joined devices that don't require authentication. The easiest solution is to publish the CRL distribution point on a web server that uses HTTP (not HTTPS).

If your CRL distribution point doesn't list an HTTP distribution point, then you need to reconfigure the issuing certificate authority to include an HTTP CRL distribution point, preferably first, in the list of distribution points.

> **Note**
>
> If your CA has published both the *Base* and the *Delta CRL*, make sure to publish the *Delta CRL* in the HTTP path. Include web server to fetch the *Delta CRL* by allowing **double escaping** in the (IIS) web server.

Domain controller certificates

Certificate authorities write CDP information in certificates as they're issued. If the distribution point changes, then previously issued certificates must be reissued for the certificate authority to include the new CDP. The domain controller certificate is one the critical components of Microsoft Entra joined devices authenticating to Active Directory.

Why does Windows need to validate the domain controller certificate?

Windows Hello for Business enforces the *strict KDC validation* security feature when authenticating from a Microsoft Entra joined device to a domain. This enforcement imposes more restrictive criteria that must be met by the Key Distribution Center (KDC). When authenticating using Windows Hello for Business on a Microsoft Entra joined device, the Windows client validates the reply from the domain controller by ensuring all of the following are met:

- The domain controller has the private key for the certificate provided
- The root CA that issued the domain controller's certificate is in the device's *Trusted Root Certificate Authorities*
- Use the *Kerberos Authentication certificate template* instead of any other older template
- The domain controller's certificate has the *KDC Authentication* extended key usage (EKU)
- The domain controller's certificate's subject alternate name has a DNS Name that matches the name of the domain
- The domain controller's certificate's signature hash algorithm is *sha256*
- The domain controller's certificate's public key is *RSA (2048 Bits)*

> **Important**

Authenticating from a Microsoft Entra hybrid joined device to a domain using Windows Hello for Business doesn't enforce that the domain controller certificate includes the *KDC Authentication* EKU. If you're adding Microsoft Entra joined devices to an existing domain environment, make sure to verify that your domain controller certificate has been updated to include the *KDC Authentication* EKU.

Configure a CRL distribution point for an issuing CA

Use this set of procedures to update the CA that issues domain controller certificates to include an http-based CRL distribution point.

Configure Internet Information Services to host CRL distribution point

You need to host your new certificate revocation list on a web server so Microsoft Entra joined devices can easily validate certificates without authentication. You can host these files on web servers many ways. The following steps are just one and may be useful for admins unfamiliar with adding a new CRL distribution point.

> **ⓘ Important**
>
> Do not configure the IIS server hosting your CRL distribution point to use https or a server authentication certificate. Clients should access the distribution point using http.

Install the web server

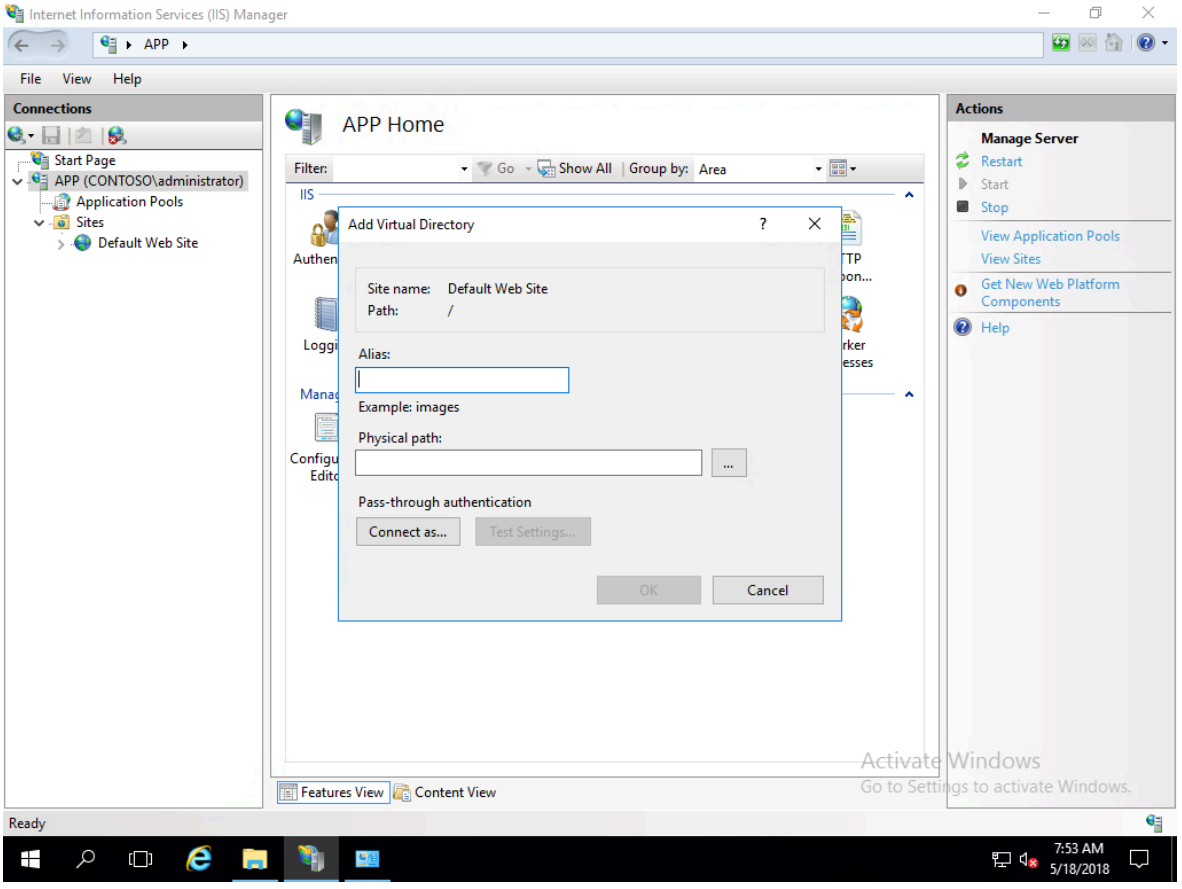
1. Sign-in to your server as a local administrator and start **Server Manager** if it didn't start during your sign in
2. Select the **Local Server** node in the navigation pane. Select **Manage** and select **Add Roles and Features**
3. In the **Add Role and Features Wizard**, select **Server Selection**. Verify the selected server is the local server. Select **Server Roles**. Select the check box next to **Web Server (IIS)**
4. Select **Next** through the remaining options in the wizard, accepting the defaults, and install the Web Server role

Configure the web server

1. From **Windows Administrative Tools**, Open **Internet Information Services (IIS) Manager**

2. Expand the navigation pane to show **Default Web Site**. Select and then right-click **Default Web site** and select **Add Virtual Directory...**

3. In the **Add Virtual Directory** dialog box, type **cdp** in **alias**. For physical path, type or browse for the physical file location where you'll host the certificate revocation list. For this example, the path `c:\cdp` is used. Select **OK**

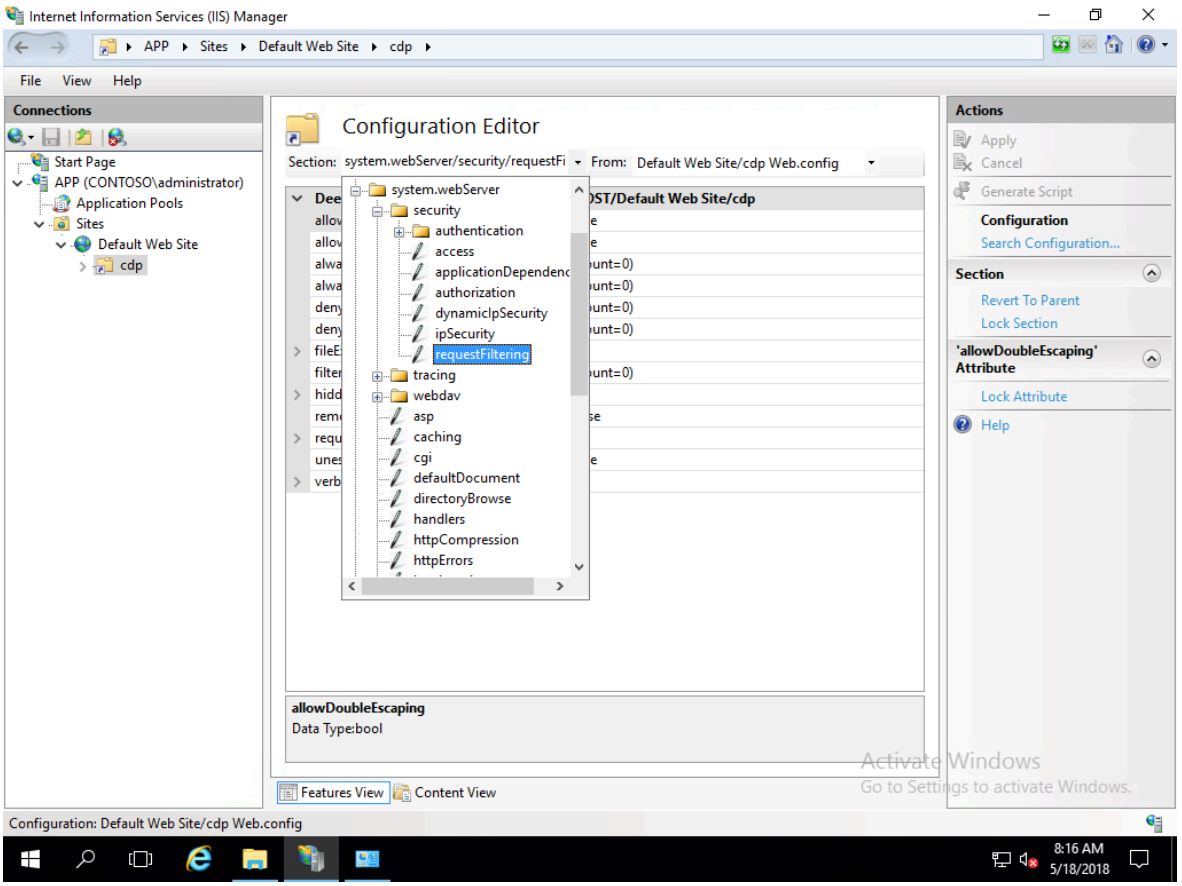


> ⓘ **Note**
>
> Make note of this path as you will use it later to configure share and file permissions.

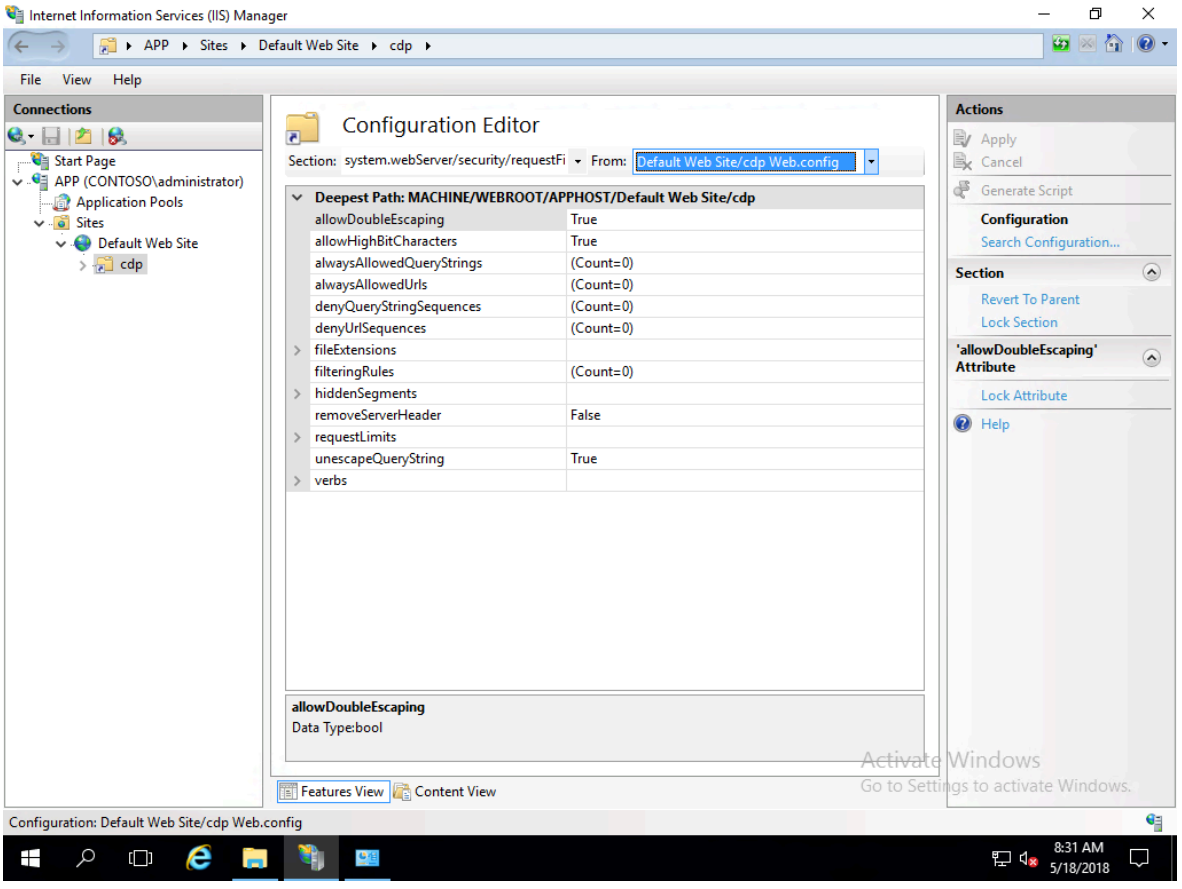
4. Select **CDP** under **Default Web Site** in the navigation pane. Open **Directory Browsing** in the content pane. Select **Enable** in the details pane

5. Select **CDP** under **Default Web Site** in the navigation pane. Open **Configuration Editor**

6. In the **Section** list, navigate to **system.webServer/security/requestFiltering**



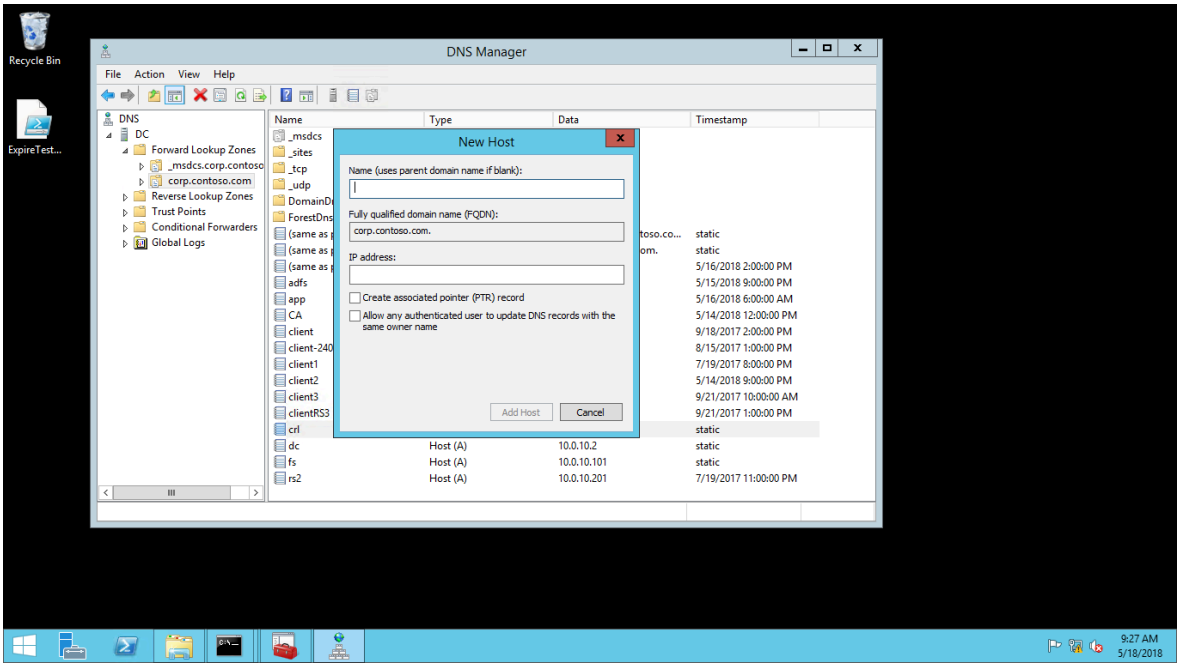
7. In the list of named value-pairs in the content pane, configure **allowDoubleEscaping** to **True**. Select **Apply** in the actions pane



8. Close **Internet Information Services (IIS) Manager**

Create a DNS resource record for the CRL distribution point URL

1. On your DNS server or from an administrative workstation, open **DNS Manager** from **Administrative Tools**
2. Expand **Forward Lookup Zones** to show the DNS zone for your domain. Right-click your domain name in the navigation pane and select **New Host (A or AAAA)...**
3. In the **New Host** dialog box, type **crl** in **Name**. Type the IP address of the web server you configured in **IP Address**. Select **Add Host**. Select **OK** to close the **DNS** dialog box. Select **Done**



4. Close the **DNS Manager**

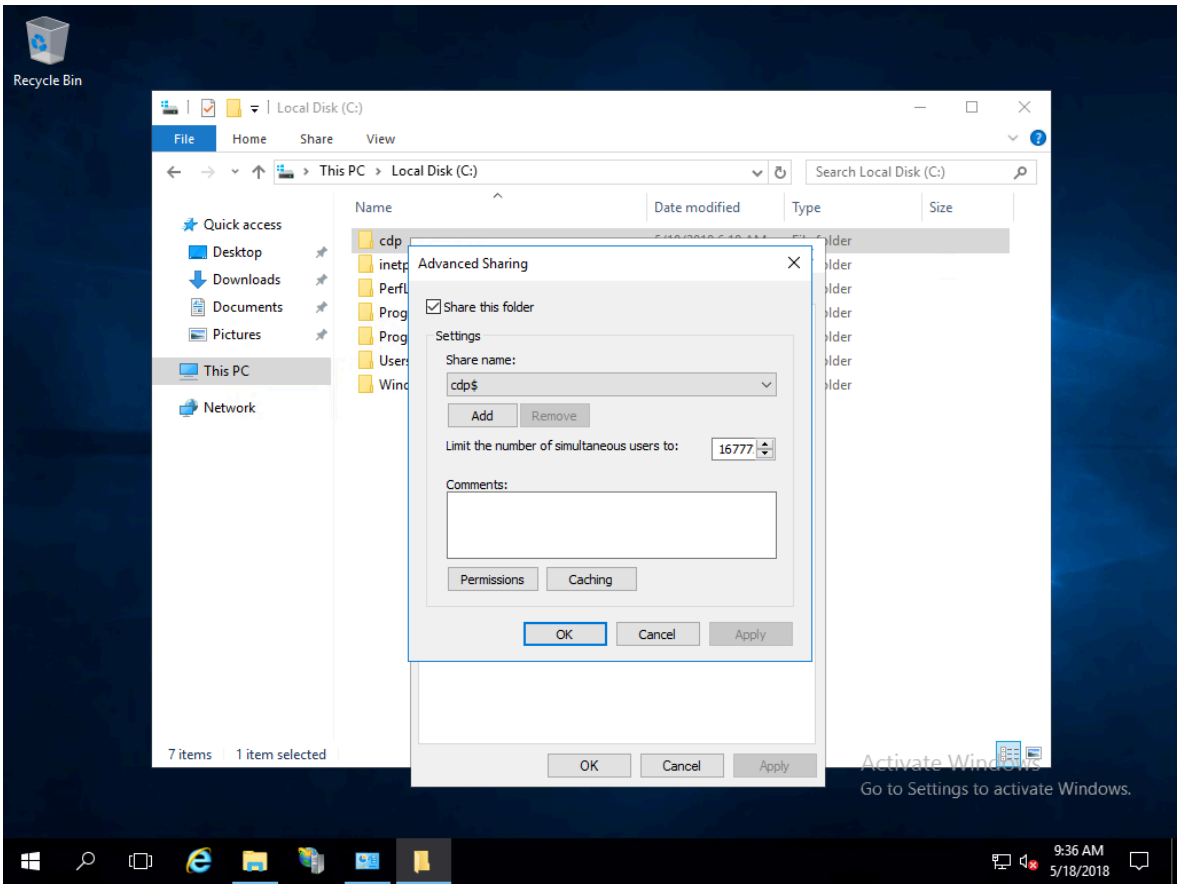
Prepare a file share to host the certificate revocation list

These procedures configure NTFS and share permissions on the web server to allow the certificate authority to automatically publish the certificate revocation list.

Configure the CDP file share

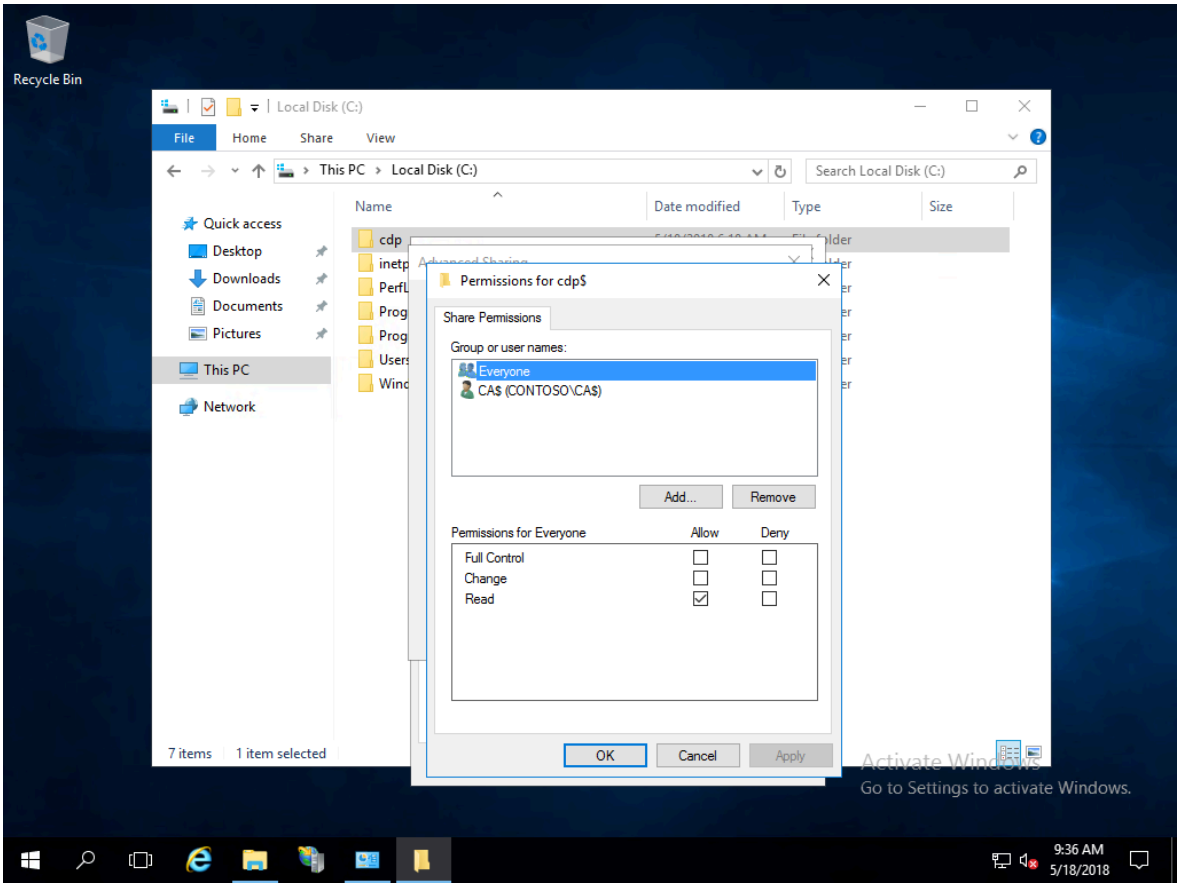
1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Sharing** tab. Select **Advanced Sharing**

3. Select **Share this folder**. Type **cdp$** in **Share name**. Select **Permissions**



4. In the **Permissions for cdp$** dialog box, select **Add**
5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Object Types**. In the **Object Types** dialog box, select **Computers**, and then select **OK**
6. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, in **Enter the object names to select**, type the name of the server running the certificate authority issuing the certificate revocation list, and then select **Check Names**. Select **OK**
7. In the **Permissions for cdp$** dialog box, select the certificate authority from the **Group or user names list**. In the **Permissions for** section, select **Allow** for **Full**

control. Select **OK**

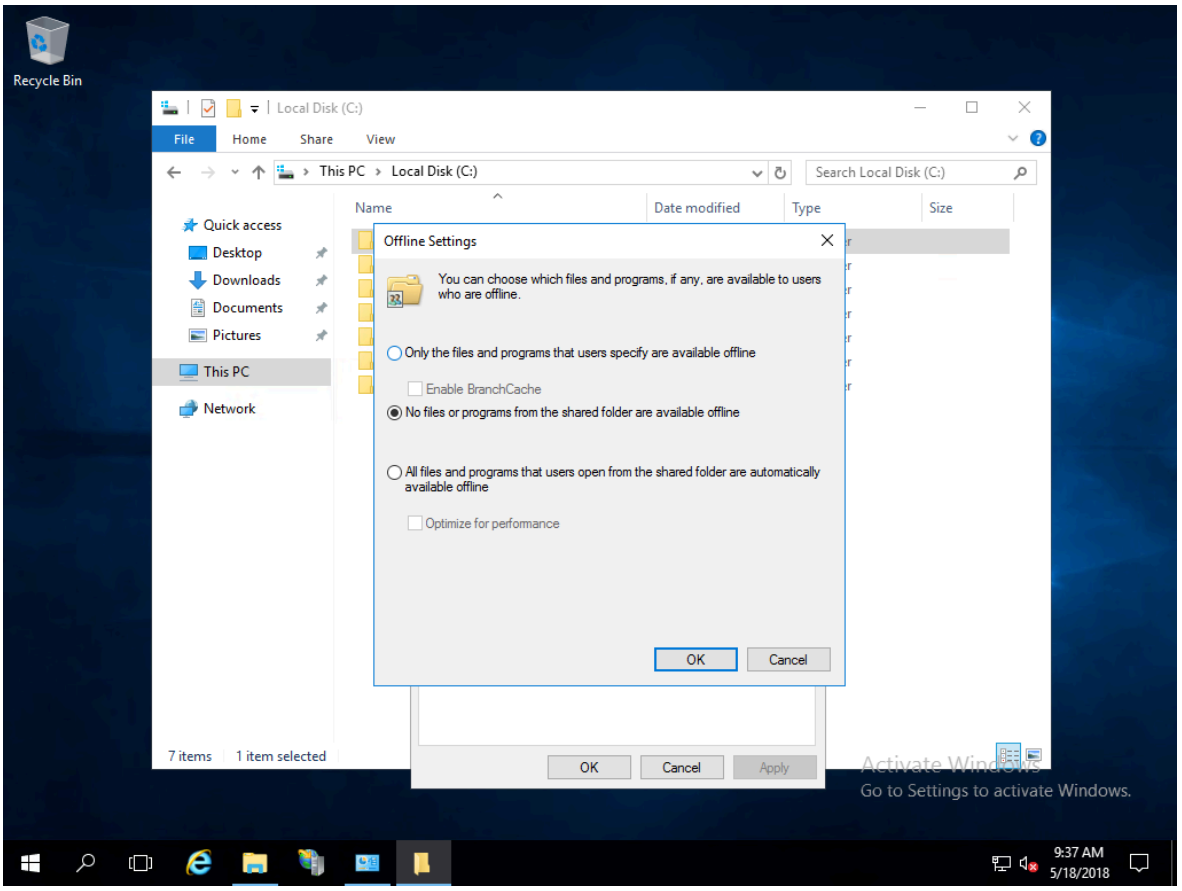


8. In the **Advanced Sharing** dialog box, select **OK**

> **Tip**
>
> Make sure that users can access `\\Server FQDN\sharename`.

Disable Caching

1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Sharing** tab. Select **Advanced Sharing**
3. Select **Caching**. Select **No files or programs from the shared folder are available offline**

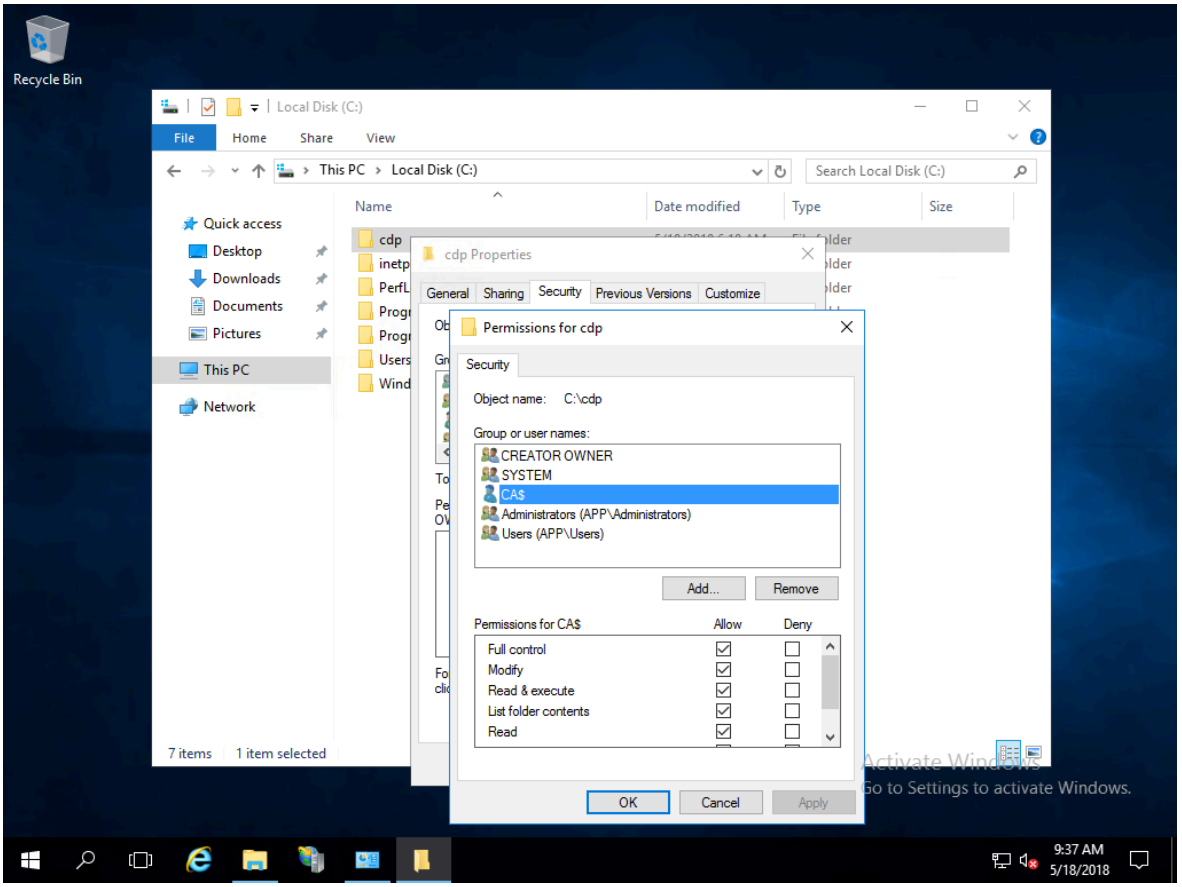


4. Select **OK**

Configure NTFS permission for the CDP folder

1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Security** tab
3. On the **Security** tab, select Edit

4. In the **Permissions for cdp** dialog box, select **Add**



5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Object Types**. In the **Object Types** dialog box, select **Computers**. Select **OK**
6. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, in **Enter the object names to select**, type the name of the certificate authority, and then select **Check Names**. Select **OK**
7. In the **Permissions for cdp** dialog box, select the name of the certificate authority from the **Group or user names** list. In the **Permissions for** section, select **Allow** for **Full control**. Select **OK**
8. Select **Close** in the **cdp Properties** dialog box

Configure the new CDP and publishing location in the issuing CA

The web server is ready to host the CRL distribution point. Now, configure the issuing certificate authority to publish the CRL at the new location and to include the new CRL distribution point.

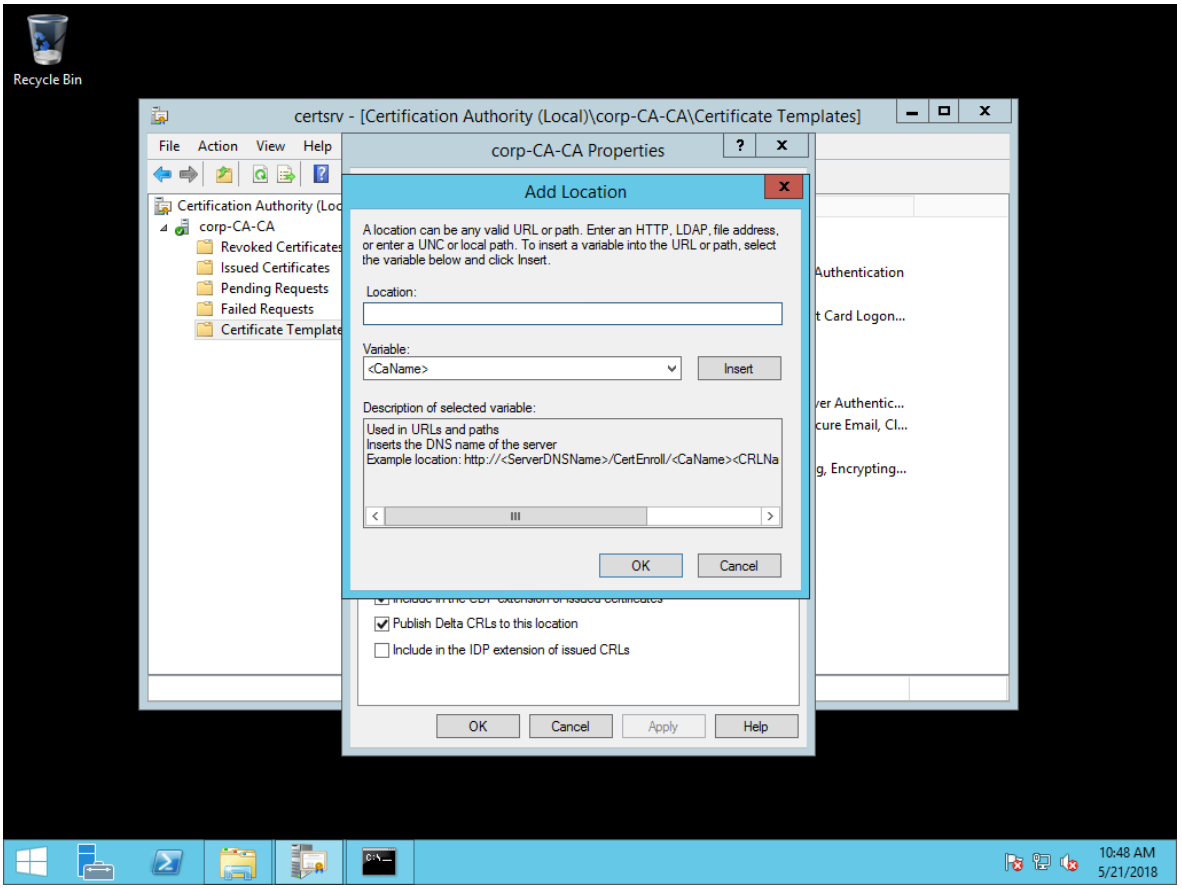
Configure the CRL distribution Point

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certification Authority** console from **Administrative Tools**

2. In the navigation pane, right-click the name of the certificate authority and select **Properties**

3. Select **Extensions**. On the **Extensions** tab, select **CRL Distribution Point (CDP)** from the **Select extension** list

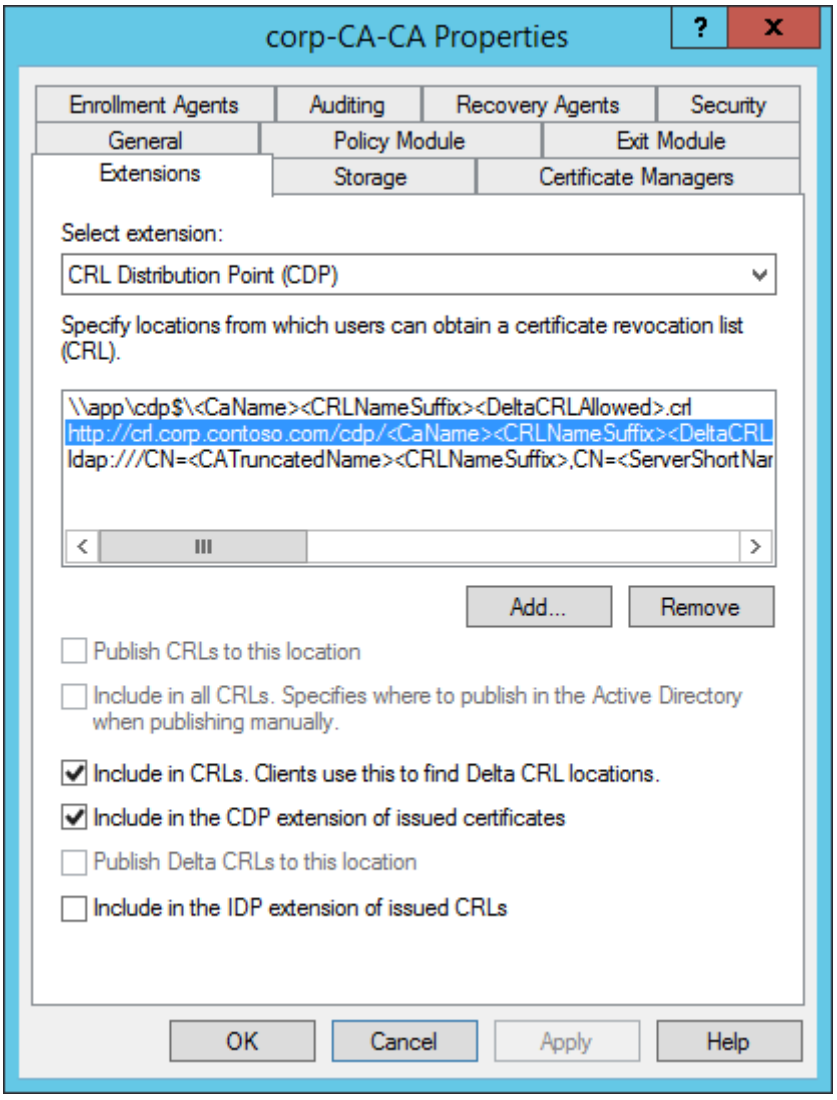
4. On the **Extensions** tab, select **Add**. Type **http://crl.[domainname]/cdp/** in **location**. For example, `<http://crl.corp.contoso.com/cdp/>` or `<http://crl.contoso.com/cdp/>` (don't forget the trailing forward slash)



5. Select <**CaName**> from the **Variable** list and select **Insert**. Select <**CRLNameSuffix**> from the **Variable** list and select **Insert**. Select <**DeltaCRLAllowed**> from the **Variable** list and select **Insert**.

6. Type **.crl** at the end of the text in **Location**. Select **OK**.

7. Select the CDP you just created.



8. Select **Include in CRLs. Clients use this to find Delta CRL locations**.

9. Select **Include in the CDP extension of issued certificates**.

10. Select **Apply** save your selections. Select **No** when ask to restart the service.

> ⓘ **Note**
>
> Optionally, you can remove unused CRL distribution points and publishing locations.

Configure the CRL publishing location

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certificate Authority** console from **Administrative Tools**

2. In the navigation pane, right-click the name of the certificate authority and select **Properties**

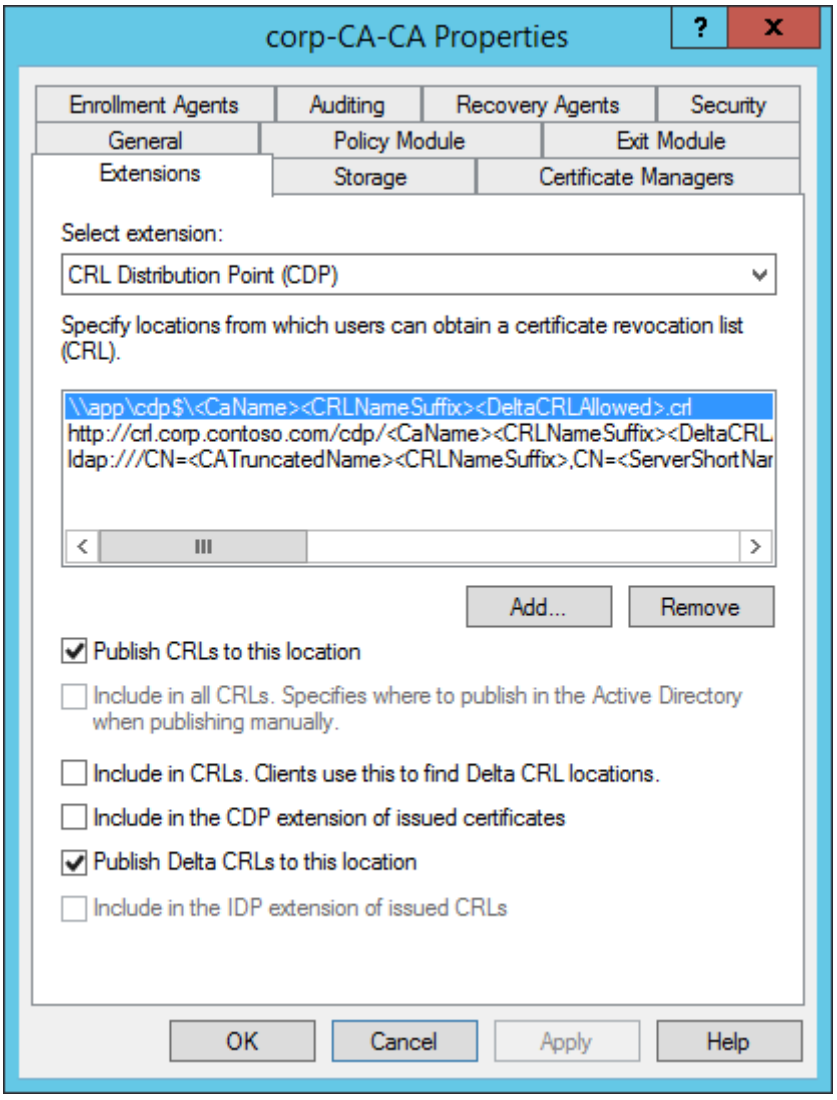
3. Select **Extensions**. On the **Extensions** tab, select **CRL Distribution Point (CDP)** from the **Select extension** list

4. On the **Extensions** tab, select **Add**. Type the computer and share name you create for your CRL distribution point in Configure the CDP file share. For example, **\\app\cdp$** (don't forget the trailing backwards slash)

5. Select **<CaName>** from the **Variable** list and select **Insert**. Select **<CRLNameSuffix>** from the **Variable** list and select **Insert**. Select **<DeltaCRLAllowed>** from the **Variable** list and select **Insert**

6. Type **.crl** at the end of the text in **Location**. Select **OK**

7. Select the CDP you just created.



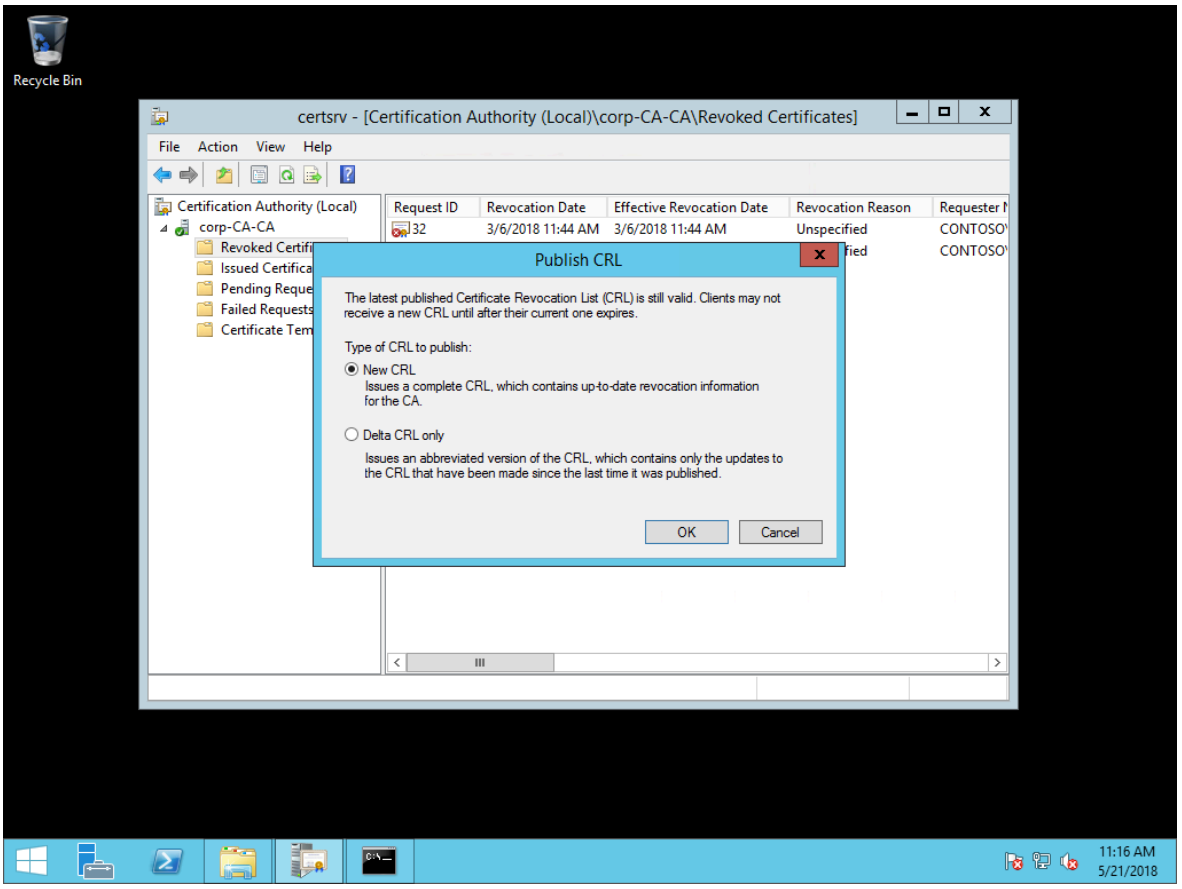
8. Select **Publish CRLs to this location**

9. Select **Publish Delta CRLs to this location**

10. Select **Apply** save your selections. Select **Yes** when ask to restart the service. Select **OK** to close the properties dialog box

Publish a new CRL

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certificate Authority** console from **Administrative Tools**.
2. In the navigation pane, right-click **Revoked Certificates**, hover over **All Tasks**, and select **Publish**



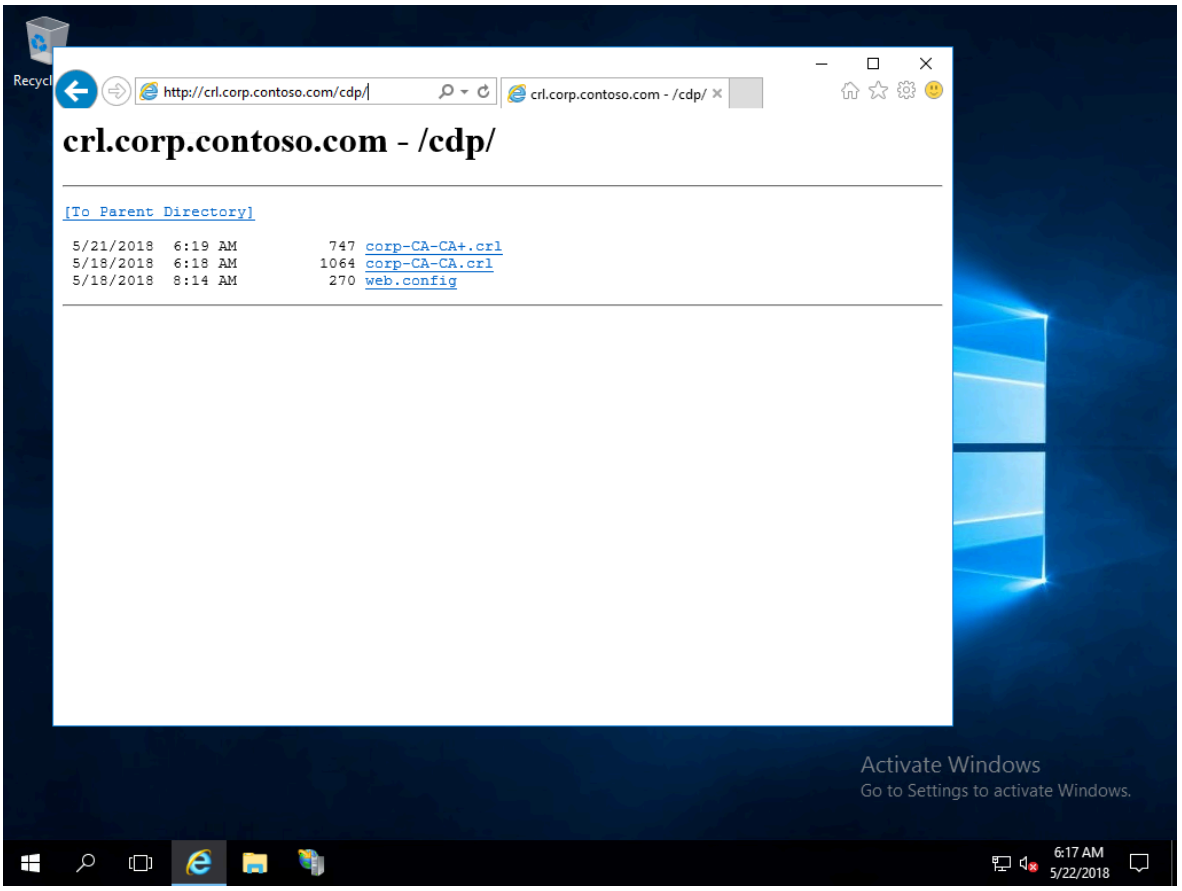
.

3. In the **Publish CRL** dialog box, select **New CRL** and select **OK**.

Validate CDP Publishing

Validate the new CRL distribution point is working.

1. Open a web browser. Navigate to `http://crl.[yourdomain].com/cdp`. You should see two files created from publishing the new CRL

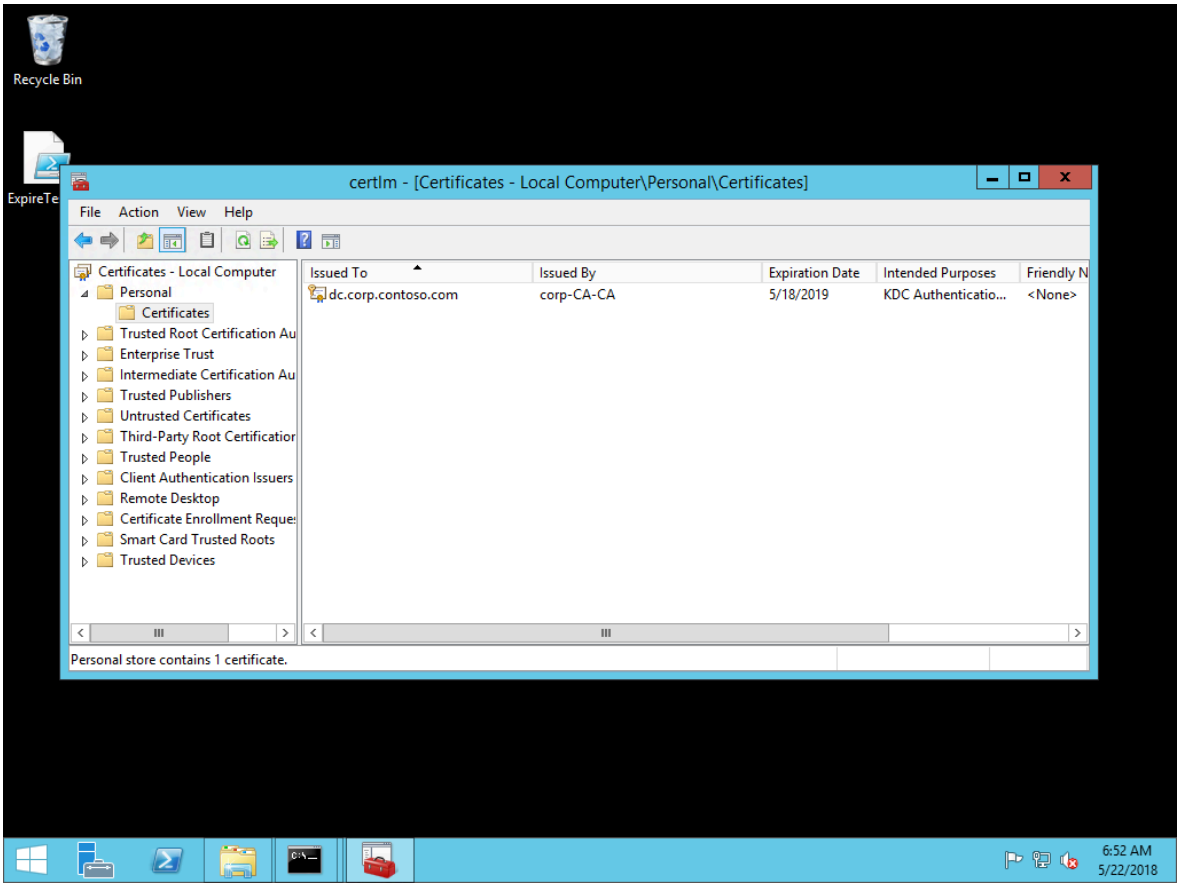


Reissue domain controller certificates

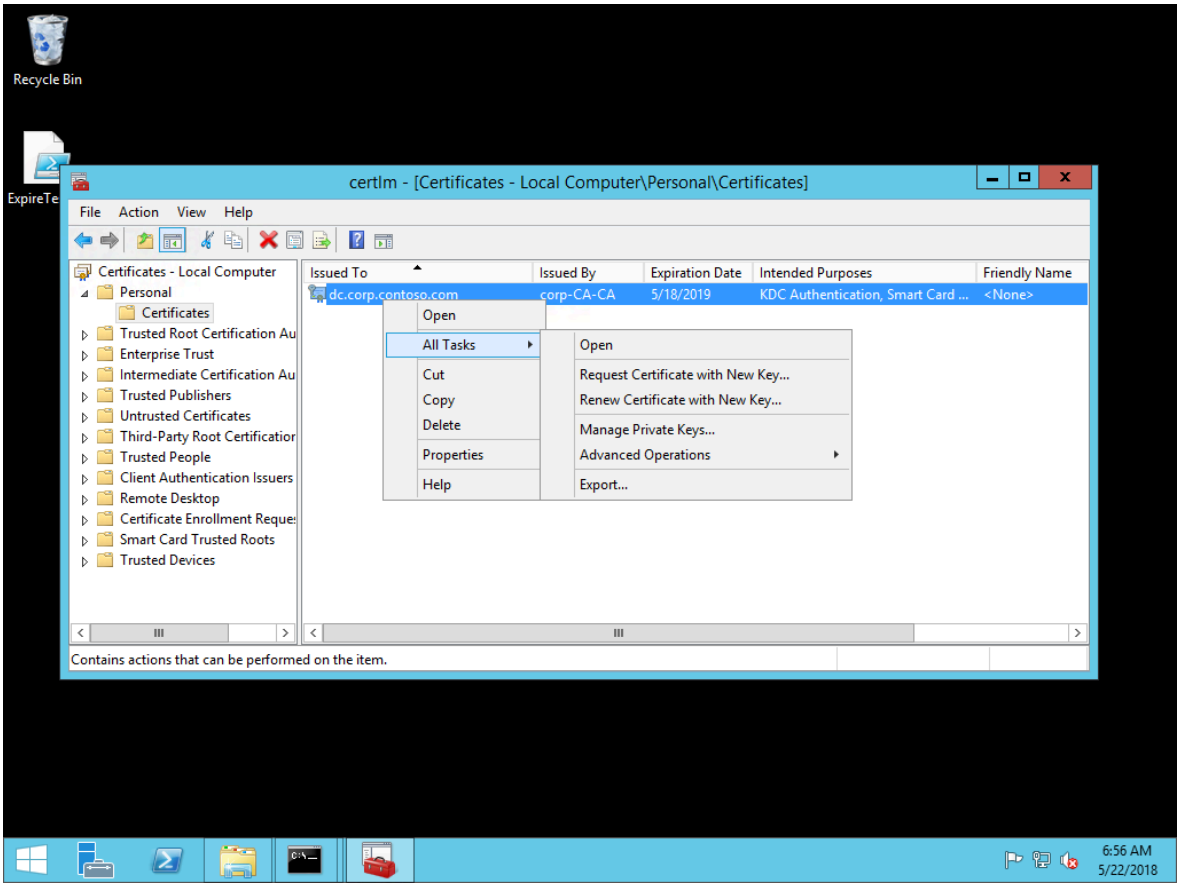
With the CA properly configured with a valid HTTP-based CRL distribution point, you need to reissue certificates to domain controllers as the old certificate doesn't have the updated CRL distribution point.

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, select the existing domain controller certificate that includes **KDC Authentication**

in the list of **Intended Purposes**



4. Right-click the selected certificate. Hover over **All Tasks** and then select **Renew Certificate with New Key...**. In the **Certificate Enrollment** wizard, select **Next**



5. In the **Request Certificates** page of the wizard, verify the selected certificate has the correct certificate template and ensure the status is available. Select **Enroll**
6. After the enrollment completes, select **Finish** to close the wizard

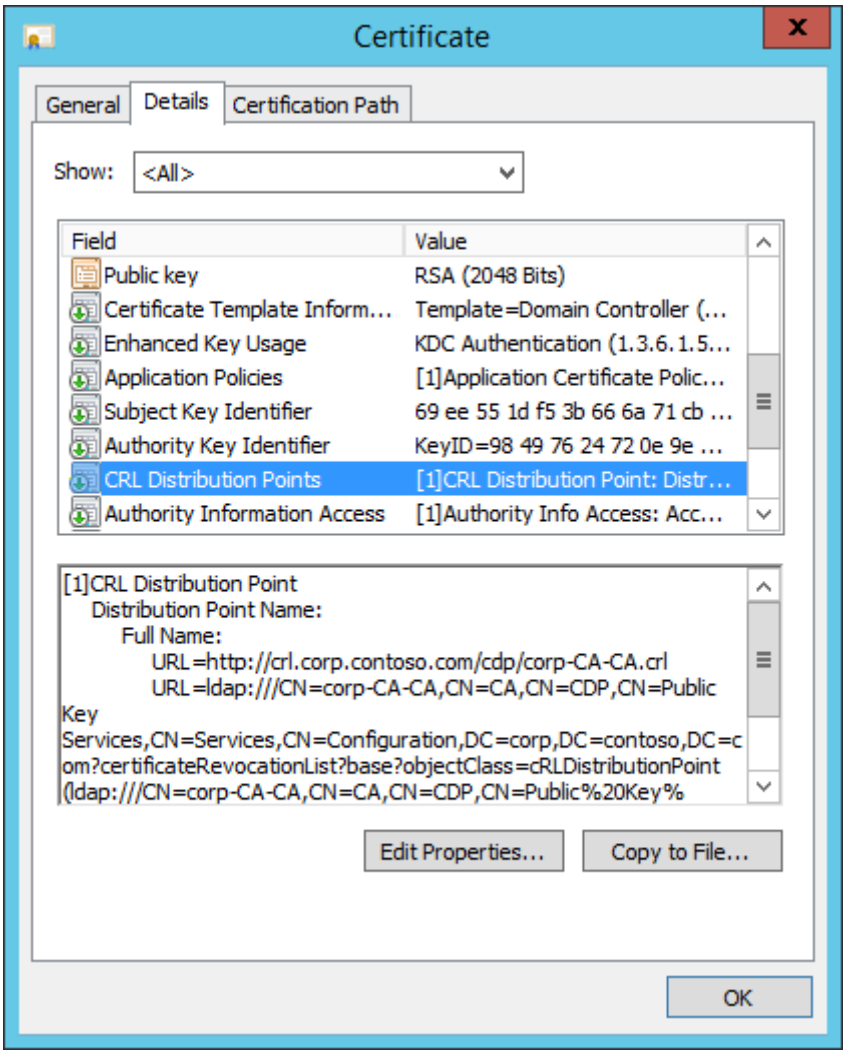
7. Repeat this procedure on all your domain controllers

> **Note**
>
> You can configure domain controllers to automatically enroll and renew their certificates. Automatic certificate enrollment helps prevent authentication outages due to expired certificates. Refer to the Windows Hello Deployment Guides to learn how to deploy automatic certificate enrollment for domain controllers.

> **Important**
>
> If you are not using automatic certificate enrollment, create a calendar reminder to alert you two months before the certificate expiration date. Send the reminder to multiple people in the organization to ensure more than one or two people know when these certificates expire.

Validate CDP in the new certificate

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, double-click the existing domain controller certificate includes **KDC Authentication** in the list of **Intended Purposes**
4. Select the **Details** tab. Scroll down the list until **CRL Distribution Points** is visible in the **Field** column of the list. Select **CRL Distribution Point**
5. Review the information below the list of fields to confirm the new URL for the CRL distribution point is present in the certificate. Select **OK**



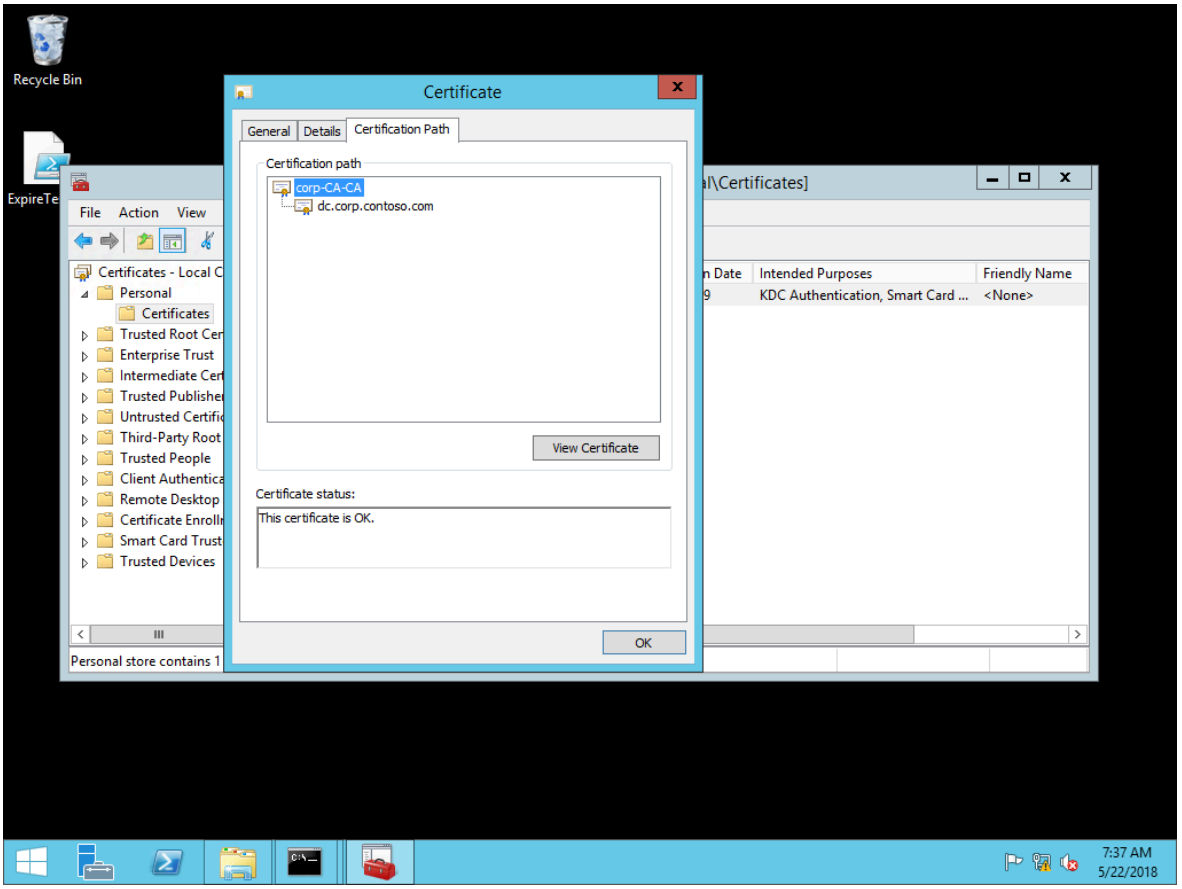
Deploy the root CA certificate to Microsoft Entra joined devices

The domain controllers have a certificate that includes the new CRL distribution point. Next, you need the enterprise root certificate so you can deploy it to Microsoft Entra joined devices. When you deploy the enterprise root certificates to a device, it ensures the device trusts any certificates issued by the certificate authority. Without the certificate, Microsoft Entra joined devices don't trust domain controller certificates and authentication fails.

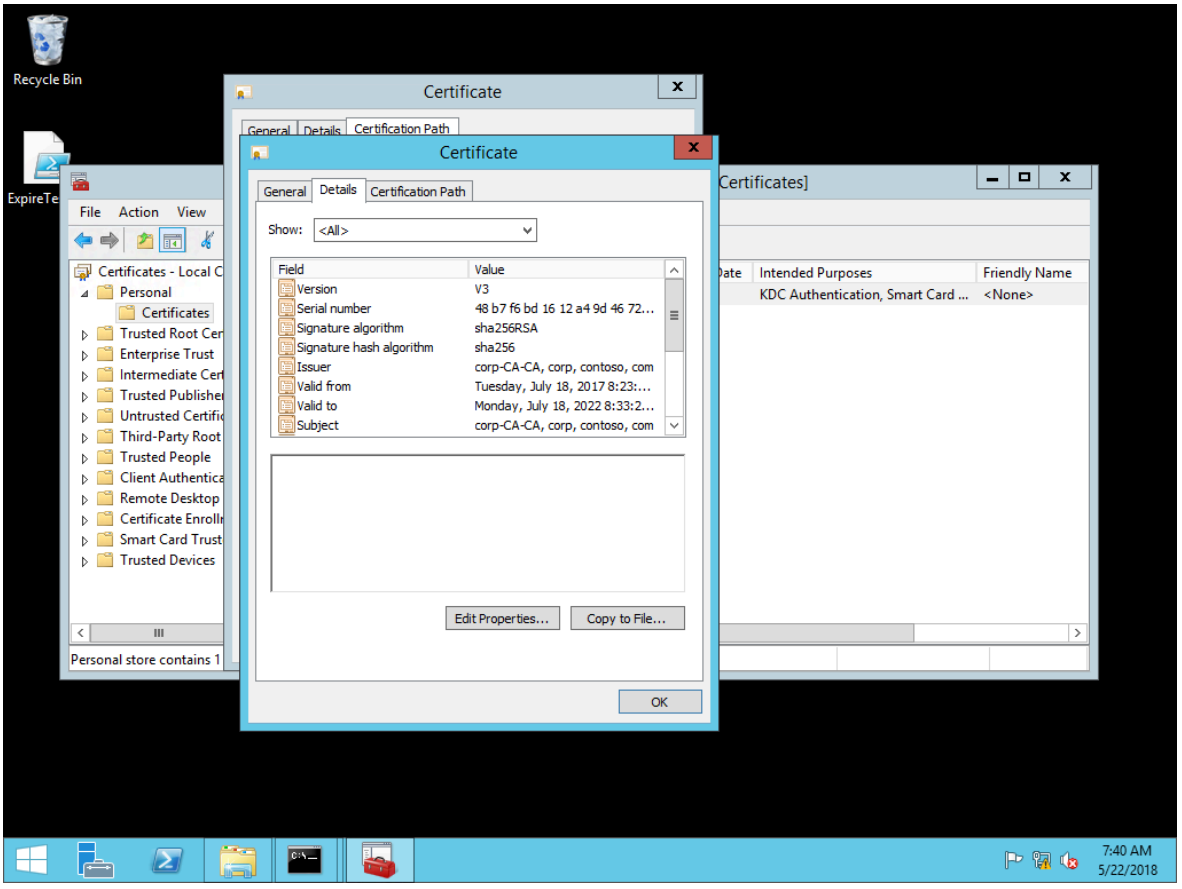
Export the enterprise root certificate

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, double-click the existing domain controller certificate includes **KDC Authentication** in the list of **Intended Purposes**

4. Select the **Certification Path** tab. In the **Certification path** view, select the topmost node and select **View Certificate**

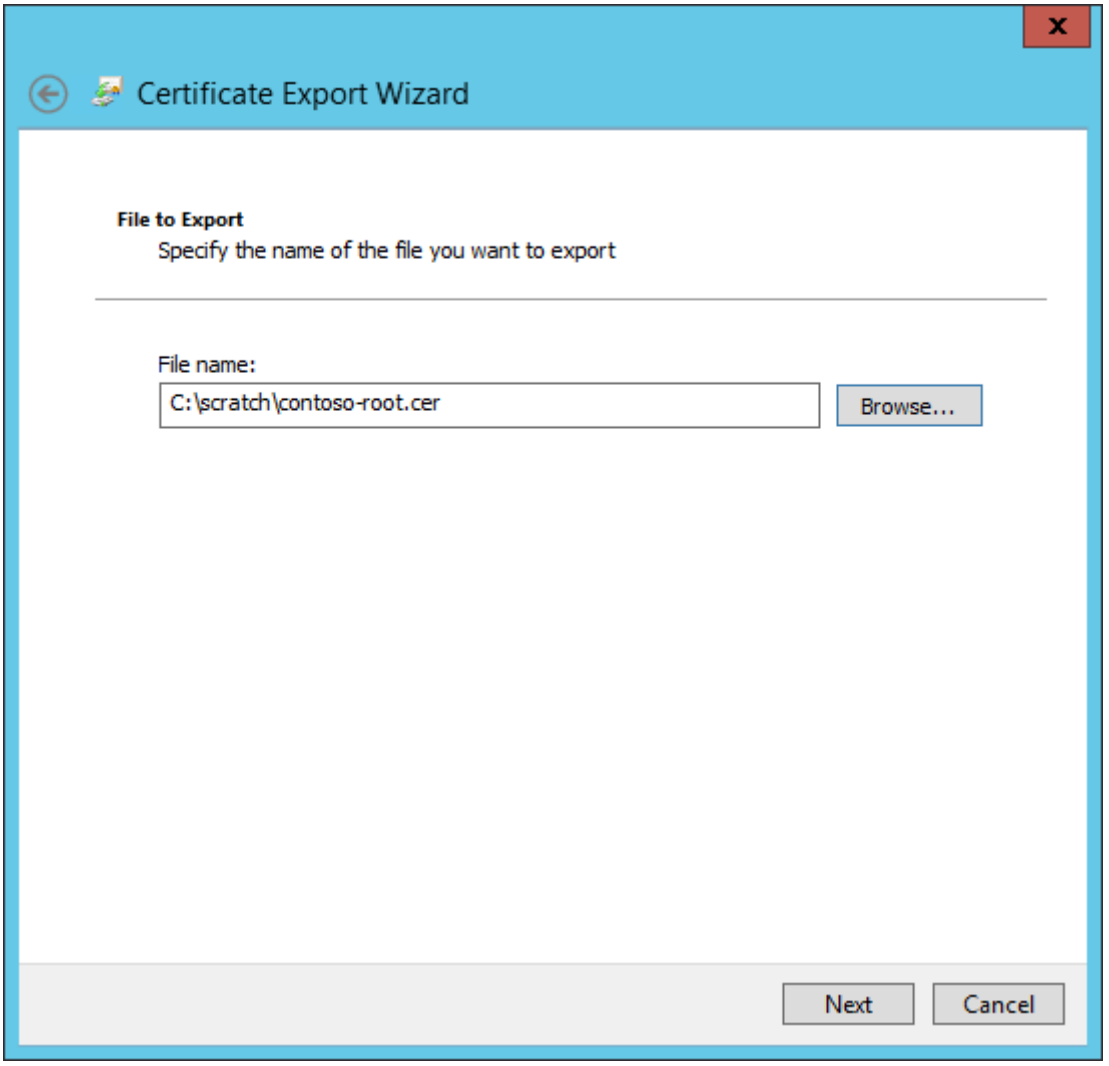


5. In the new **Certificate** dialog box, select the **Details** tab. Select **Copy to File**



6. In the **Certificate Export Wizard**, select **Next**
7. On the **Export File Format** page of the wizard, select **Next**

8. On the **File to Export** page in the wizard, type the name and location of the root certificate and select **Next**. Select **Finish** and then select **OK** to close the success dialog box



9. Select **OK** two times to return to the **Certificate Manager** for the local computer. Close the **Certificate Manager**

Deploy the certificate via Intune

To configure devices with Microsoft Intune, use a custom policy:

1. Go to the Microsoft Intune admin center
2. Select **Devices** > **Configuration profiles** > **Create profile**
3. Select **Platform** > **Windows 8.1 and later** and **Profile type** > **Trusted certificate**
4. Select **Create**
5. In **Configuration settings**, select the folder icon and browse for the enterprise root certificate file. Once the file is selected, select **Open** to upload it to Intune
6. Under **Destination store** dropdown, select **Computer certificate store - Root**
7. Select **Next**
8. Under **Assignment**, select a security group that contains as members the devices or users that you want to configure > **Next**

9. Review the policy configuration and select **Create**

If you plan on using certificates for on-premises single-sign on, perform the additional steps in Using Certificates for On-premises Single-sign On. Otherwise, you can sign in to a Microsoft Entra joined device with Windows Hello for Business and test SSO to an on-premises resource.

Feedback

Was this page helpful? Yes No

Provide product feedback

Hybrid certificate trust deployment guide

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** certificate trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

> **Important**
>
> Windows Hello for Business *cloud Kerberos trust* is the recommended deployment model when compared to the *key trust model*. It is also the recommended deployment model if you don't need to deploy certificates to the end users. For more information, see **cloud Kerberos trust deployment**.

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Public Key Infrastructure
- ✓ Authentication
- ✓ Device configuration
- ✓ Licensing for cloud services
- ✓ Prepare users to use Windows Hello

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure and validate the Public Key Infrastructure
- ✓ Configure Active Directory Federation Services

- ✓ Configure and enroll in Windows Hello for Business
- ✓ (optional) Configure single sign-on for Microsoft Entra joined devices

Federated authentication to Microsoft Entra ID

Windows Hello for Business hybrid certificate trust requires Active Directory to be federated with Microsoft Entra ID using AD FS. You must also configure the AD FS farm to support Microsoft Entra registered devices.

If you're new to AD FS and federation services:

- Review key AD FS concepts prior to deploying the AD FS farm
- Review the AD FS design guide to design and plan your federation service

Once you have your AD FS design ready, review deploying a federation server farm to configure AD FS in your environment

The AD FS farm used with Windows Hello for Business must be Windows Server 2016 with minimum update of KB4088889 (14393.2155).

Device registration and device write-back

Windows devices must be registered in Microsoft Entra ID. Devices can be registered in Microsoft Entra ID using either *Microsoft Entra join* or *Microsoft Entra hybrid join*.
For Microsoft Entra hybrid joined devices, review the guidance on the plan your Microsoft Entra hybrid join implementation page.

Refer to the Configure Microsoft Entra hybrid join for federated domains guide to learn more about using Microsoft Entra Connect Sync to configure Microsoft Entra device registration.
For a **manual configuration** of your AD FS farm to support device registration, review the Configure AD FS for Microsoft Entra device registration guide.

Hybrid certificate trust deployments require the *device write-back* feature. Authentication to AD FS needs both the user and the device to authenticate. Typically the users are synchronized, but not devices. This prevents AD FS from authenticating the device and results in Windows Hello for Business certificate enrollment failures. For this reason, Windows Hello for Business deployments need device write-back.

> ⓘ **Note**

Windows Hello for Business is tied between a user and a device. Both the user and device need to be synchronized between Microsoft Entra ID and Active Directory. Device write-back is used to update the `msDS-KeyCredentialLink` attribute on the computer object.

If you manually configured AD FS, or if you ran Microsoft Entra Connect Sync using *Custom Settings*, you must ensure to configure **device write-back** and **device authentication** in your AD FS farm. For more information, see Configure Device Write Back and Device Authentication.

Public Key Infrastructure

An enterprise public key infrastructure (PKI) is required as *trust anchor* for authentication. Domain controllers require a certificate for Windows clients to trust them.
The enterprise PKI and a certificate registration authority (CRA) are required to issue authentication certificates to users. Hybrid certificate trust deployment uses AD FS as a CRA.

During Windows Hello for Business provisioning, users receive a sign-in certificate through the CRA.

Next steps

Once the prerequisites are met, deploying Windows Hello for Business with a hybrid key trust model consists of the following steps:

- ✓ Configure and validate the PKI
- ✓ Configure AD FS
- ✓ Configure Windows Hello for Business settings
- ✓ Provision Windows Hello for Business on Windows clients
- ✓ Configure single sign-on (SSO) for Microsoft Entra joined devices

Next: configure and validate the Public Key Infrastructure >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure and validate the PKI in a hybrid certificate trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** certificate trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

Windows Hello for Business must have a Public Key Infrastructure (PKI) when using the *certificate trust* models. The domain controllers must have a certificate, which serves as a *root of trust* for clients. The certificate ensures that clients don't communicate with rogue domain controllers.

Hybrid certificate trust deployments issue users a sign-in certificate, enabling them to authenticate to Active Directory using Windows Hello for Business credentials. Additionally, hybrid certificate trust deployments issue certificates to registration authorities to provide defense-in-depth security when issuing user authentication certificates.

Deploy an enterprise certification authority

This guide assumes most enterprises have an existing public key infrastructure. Windows Hello for Business depends on an enterprise PKI running the Windows Server *Active Directory Certificate Services* role.
If you don't have an existing PKI, review Certification Authority Guidance to properly design your infrastructure. Then, consult the Test Lab Guide: Deploying an AD CS Two-Tier PKI Hierarchy for instructions on how to configure your PKI using the information from your design session.

Lab-based PKI

The following instructions may be used to deploy simple public key infrastructure that is suitable **for a lab environment**.

Sign in using *Enterprise Administrator* equivalent credentials on a Windows Server where you want the certification authority (CA) installed.

> **Note**
>
> Never install a certification authority on a domain controller in a production environment.

1. Open an elevated Windows PowerShell prompt
2. Use the following command to install the Active Directory Certificate Services role.

```PowerShell
Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
```

3. Use the following command to configure the CA using a basic certification authority configuration

```PowerShell
Install-AdcsCertificationAuthority
```

Configure the enterprise PKI

Configure domain controller certificates

Clients must trust the domain controllers, and the best way to enable the trust is to ensure that each domain controller has a *Kerberos Authentication* certificate. Installing a certificate on the domain controllers enables the Key Distribution Center (KDC) to prove its identity to other members of the domain. The certificates provide clients a root of trust external to the domain, namely the *enterprise certification authority*.

Domain controllers automatically request a *domain controller certificate* (if published) when they discover an enterprise CA is added to Active Directory. The certificates based on the *Domain Controller* and *Domain Controller Authentication* certificate templates don't include the *KDC Authentication* object identifier (OID), which was later added to the Kerberos RFC. Therefore, domain controllers need to request a certificate based on the *Kerberos Authentication* certificate template.

By default, the Active Directory CA provides and publishes the *Kerberos Authentication* certificate template. The cryptography configuration included in the template is based on older and less performant cryptography APIs. To ensure domain controllers request the proper certificate with the best available cryptography, use the *Kerberos*

Authentication certificate template as a *baseline* to create an updated domain controller certificate template.

> **ⓘ Important**
>
> The certificates issued to the domain controllers must meet the following requirements:
>
> - The *Certificate Revocation List (CRL) distribution point* extension must point to a valid CRL, or an *Authority Information Access (AIA)* extension that points to an Online Certificate Status Protocol (OCSP) responder
> - Optionally, the certificate *Subject* section could contain the directory path of the server object (the distinguished name)
> - The certificate *Key Usage* section must contain *Digital Signature* and *Key Encipherment*
> - Optionally, the certificate *Basic Constraints* section should contain: `[Subject Type=End Entity, Path Length Constraint=None]`
> - The certificate *extended key usage* section must contain Client Authentication (`1.3.6.1.5.5.7.3.2`), Server Authentication (`1.3.6.1.5.5.7.3.1`), and KDC Authentication (`1.3.6.1.5.2.3.5`)
> - The certificate *Subject Alternative Name* section must contain the Domain Name System (DNS) name
> - The certificate template must have an extension that has the value `DomainController`, encoded as a BMPstring. If you are using Windows Server Enterprise Certificate Authority, this extension is already included in the domain controller certificate template
> - The domain controller certificate must be installed in the local computer's certificate store

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Kerberos Authentication** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Domain Controller Authentication (Kerberos)* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **None** from the **Subject name format** list • Select **DNS name** from the **Include this information in alternate subject** list • Clear all other items
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

> **Note**
>
> Inclusion of the *KDC Authentication* OID in domain controller certificate is not required for Microsoft Entra hybrid joined devices. The OID is required for enabling authentication with Windows Hello for Business to on-premises resources by Microsoft Entra joined devices.

> **Important**
>
> For Microsoft Entra joined devices to authenticate to on-premises resources, ensure to:

- Install the root CA certificate in the device's trusted root certificate store. See how to deploy a trusted certificate profile via Intune
- Publish your certificate revocation list to a location that is available to Microsoft Entra joined devices, such as a web-based URL

Supersede existing domain controller certificates

The domain controllers may have an existing domain controller certificate. The Active Directory Certificate Services provides a default certificate template for domain controllers called *domain controller certificate*. Later releases of Windows Server provided a new certificate template called *domain controller authentication certificate*. These certificate templates were provided prior to the update of the Kerberos specification that stated Key Distribution Centers (KDCs) performing certificate authentication needed to include the *KDC Authentication* extension.

The *Kerberos Authentication* certificate template is the most current certificate template designated for domain controllers, and should be the one you deploy to all your domain controllers.
The *autoenrollment* feature allows you to replace the domain controller certificates. Use the following configuration to replace older domain controller certificates with new ones, using the *Kerberos Authentication* certificate template.

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the *Domain Controller Authentication (Kerberos)* (or the name of the certificate template you created in the previous section) template in the details pane and select **Properties**
4. Select the **Superseded Templates** tab. Select **Add**
5. From the **Add Superseded Template** dialog, select the *Domain Controller* certificate template and select **OK** > **Add**
6. From the **Add Superseded Template** dialog, select the *Domain Controller Authentication* certificate template and select **OK**
7. From the **Add Superseded Template** dialog, select the *Kerberos Authentication* certificate template and select **OK**
8. Add any other enterprise certificate templates that were previously configured for domain controllers to the **Superseded Templates** tab
9. Select **OK** and close the **Certificate Templates** console

The certificate template is configured to supersede all the certificate templates provided in the *superseded templates* list.
However, the certificate template and the superseding of certificate templates isn't active until the template is published to one or more certificate authorities.

> **Note**
>
> The domain controller's certificate must chain to a root in the NTAuth store. By default, the Active Directory Certificate Authority's root certificate is added to the NTAuth store. If you are using a non-Microsoft CA, this may not be done by default. If the domain controller certificate does not chain to a root in the NTAuth store, user authentication will fail. To see all certificates in the NTAuth store, use the following command:
>
> `Certutil -viewstore -enterprise NTAuth`

Configure an enrollment agent certificate template

A certificate registration authority (CRA) is a trusted authority that validates certificate request. Once it validates the request, it presents the request to the certification authority (CA) for issuance. The CA issues the certificate, returns it to the CRA, which returns the certificate to the requesting user. Windows Hello for Business certificate trust deployments use AD FS as the CRA.

The CRA enrolls for an *enrollment agent certificate*. Once the CRA verifies the certificate request, it signs the certificate request using its enrollment agent certificate and sends it to the CA. The Windows Hello for Business Authentication certificate template is configured to only issue certificates to certificate requests that have been signed with an enrollment agent certificate. The CA only issues a certificate for that template if the registration authority signs the certificate request.

> **Important**
>
> Follow the procedures below based on the AD FS service account used in your environment.

Create an enrollment agent certificate for Group Managed Service Accounts (GMSA)

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** and select **Manage**
3. In the **Certificate Template Console**, right-click on the **Exchange Enrollment Agent (Offline request)** template details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *WHFB Enrollment Agent* • Set the validity period to the desired value
Subject Name	Select **Supply in the request** **Note:** Group Managed Service Accounts (GMSA) don't support the *Build from this Active Directory information* option and will result in the AD FS server failing to enroll the enrollment agent certificate. You must configure the certificate template with *Supply in the request* to ensure that AD FS servers can perform the automatic enrollment and renewal of the enrollment agent certificate.
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**
Security	• Select **Add** • Select **Object Types** and select the **Service Accounts** check box • Select **OK** • Type `adfssvc` in the **Enter the object names to select** text box and select **OK** • Select the **adfssvc** from the **Group or users names** list. In the **Permissions for adfssvc** section:

Tab Name	Configurations
	◦ In the **Permissions for adfssvc** section, select the **Allow** check box for the **Enroll** permission ◦ Excluding the **adfssvc** user, clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other items in the **Group or users names** list • Select **OK**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Create an enrollment agent certificate for a standard service account

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console

2. Right-click **Certificate Templates** and select **Manage**

3. In the **Certificate Template Console**, right-click on the **Exchange Enrollment Agent (Offline request)** template details pane and select **Duplicate Template**

4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the **Certification Authority** list • Select **Windows 10 / Windows Server 2016** from the **Certificate Recipient** list
General	• Specify a **Template display name**, for example *WHFB Enrollment Agent* • Set the validity period to the desired value
Subject Name	• Select **Build from this Active Directory information** • Select **Fully distinguished name** from the **Subject name format** list

Tab Name	Configurations
	• Select the **User Principal Name (UPN)** check box under **Include this information in alternative subject name**
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

Security	• Select **Add**
	• Select **Object Types** and select the **Service Accounts** check box
	• Select **OK**
	• Type `adfssvc` in the **Enter the object names to select** text box and select **OK**
	• Select the **adfssvc** from the **Group or users names** list. In the **Permissions for adfssvc** section:
	  ◦ In the **Permissions for adfssvc** section, select the **Allow** check box for the **Enroll** permission
	  ◦ Excluding the **adfssvc** user, clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other items in the **Group or users names** list
	• Select **OK**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Configure a Windows Hello for Business authentication certificate template

During Windows Hello for Business provisioning, Windows clients request an authentication certificate from AD FS, which requests the authentication certificate on behalf of the user. This task configures the Windows Hello for Business authentication certificate template.

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console

2. Right-click **Certificate Templates** and select **Manage**

3. In the **Certificate Template Console**, right-click the **Smartcard Logon** template and select **Duplicate Template**

4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *WHFB Authentication* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **Fully distinguished name** from the **Subject name format** list • Select the **User Principal Name (UPN)** check box under **Include this information in alternative subject name**
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**
Extensions	Verify the **Application Policies** extension includes **Smart Card Logon**
Issuance Requirements	• Select the **This number of authorized signatures** check box. Type *1* in the text box • Select **Application policy** from the *Policy type required in signature* • Select **Certificate Request Agent** from in the *Application policy* list • Select the **Valid existing certificate** option
Request Handling	Select the **Renew with same key** check box
Security	• Select **Add** • Target an Active Directory security group that contains the users that you want to enroll in Windows Hello for Business. For example, if you have a group called *Window Hello for Business Users*, type it in the **Enter the object names to select** text box and select **OK** • Select the **Windows Hello for Business Users** from the **Group or users names** list. In the **Permissions for Windows Hello for Business Users** section: ◦ Select the **Allow** check box for the **Enroll** permission

Tab Name	Configurations
	◦ Excluding the group above (for example, *Window Hello for Business Users*), clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other entries in the **Group or users names** section if the check boxes aren't already cleared • Select **OK**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Mark the template as the Windows Hello Sign-in template

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials

Open an elevated command prompt end execute the following command

Windows Command Prompt

```
certutil.exe -dsTemplate WHFBAuthentication msPKI-Private-Key-Flag
+CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY
```

If the template was changed successfully, the output of the command will contain old and new values of the template parameters. The new value must contain the `CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY` parameter. Example:

Windows Command Prompt

```
CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=[yourdomain]:WHFBAuthentication

Old Value:
msPKI-Private-Key-Flag REG_DWORD = 5050080 (84213888)
CTPRIVATEKEY_FLAG_REQUIRE_SAME_KEY_RENEWAL -- 80 (128)
CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0
TEMPLATE_SERVER_VER_WINBLUE<<CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT -- 50000
(327680)
TEMPLATE_CLIENT_VER_WINBLUE<<CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT --
5000000 (83886080)

New Value:
msPKI-Private-Key-Flag REG_DWORD = 5250080 (86311040)
CTPRIVATEKEY_FLAG_REQUIRE_SAME_KEY_RENEWAL -- 80 (128)
CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0
TEMPLATE_SERVER_VER_WINBLUE<<CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT -- 50000
(327680)
```

```
CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY -- 200000 (2097152)
TEMPLATE_CLIENT_VER_WINBLUE<<CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT --
5000000 (83886080)
CertUtil: -dsTemplate command completed successfully."
```

> ⓘ **Note**
>
> If you gave your Windows Hello for Business Authentication certificate template a different name, then replace `WHFBAuthentication` in the above command with the name of your certificate template. It's important that you use the template name rather than the template display name. You can view the template name on the **General** tab of the certificate template using the Certificate Template management console (certtmpl.msc).

Unpublish Superseded Certificate Templates

The certification authority only issues certificates based on published certificate templates. For security, it's a good practice to unpublish certificate templates that the CA isn't configured to issue, including the pre-published templates from the role installation and any superseded templates.

The newly created *domain controller authentication* certificate template supersedes previous domain controller certificate templates. Therefore, you need to unpublish these certificate templates from all issuing certificate authorities.

Sign in to the CA or management workstation with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane > **Certificate Templates**
3. Right-click the *Domain Controller* certificate template and select **Delete**. Select **Yes** on the **Disable certificate templates** window
4. Repeat step 3 for the *Domain Controller Authentication* and *Kerberos Authentication* certificate templates

Publish the certificate templates to the CA

A certification authority can only issue certificates for certificate templates that are published to it. If you have more than one CA, and you want more CAs to issue certificates based on the certificate template, then you must publish the certificate template to them.

Sign in to the CA or management workstations with **Enterprise Admin** equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane
3. Select **Certificate Templates** in the navigation pane
4. Right-click the **Certificate Templates** node. Select **New** > **Certificate Template to issue**
5. In the **Enable Certificates Templates** window, select the *Domain Controller Authentication (Kerberos)*, *WHFB Enrollment Agent* and *WHFB Authentication* templates you created in the previous steps > select **OK**
6. Close the console

> ⓘ **Important**
>
> If you plan to deploy **Microsoft Entra joined** devices, and require single sign-on (SSO) to on-premises resources when signing in with Windows Hello for Business, follow the procedures to update your CA to include an http-based CRL distribution point.

Configure and deploy certificates to domain controllers

Configure automatic certificate enrollment for the domain controllers

Domain controllers automatically request a certificate from the *Domain controller certificate* template. However, domain controllers are unaware of newer certificate templates or superseded configurations on certificate templates. For domain controllers to automatically enroll and renew of certificates, configure a GPO for automatic certificate enrollment, and link it to the *Domain Controllers* OU.

1. Open the **Group Policy Management Console** (gpmc.msc)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type *Domain Controller Auto Certificate Enrollment* in the name box and select **OK**
5. Right-click the **Domain Controller Auto Certificate Enrollment** Group Policy object and select **Edit**

6. In the navigation pane, expand **Policies** under **Computer Configuration**
7. Expand **Windows Settings** > **Security Settings** > **Public Key Policies**
8. In the details pane, right-click **Certificate Services Client - Auto-Enrollment** and select **Properties**
9. Select **Enabled** from the **Configuration Model** list
10. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box
11. Select the **Update certificates that use certificate templates** check box
12. Select **OK**
13. Close the **Group Policy Management Editor**

Deploy the domain controller auto certificate enrollment GPO

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and expand the node with the Active Directory domain name. Right-click the **Domain Controllers** organizational unit and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select *Domain Controller Auto Certificate Enrollment* or the name of the domain controller certificate enrollment Group Policy object you previously created
4. Select **OK**

Validate the configuration

Windows Hello for Business is a distributed system, which on the surface appears complex and difficult. The key to a successful deployment is to validate phases of work prior to moving to the next phase.

Confirm your domain controllers enroll the correct certificates and not any superseded certificate templates. Check that each domain controller completed the certificate autoenrollment.

Use the event logs

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Using the Event Viewer, navigate to the **Application and Services** > **Microsoft** > **Windows** > **CertificateServices-Lifecycles-System** event log
2. Look for an event indicating a new certificate enrollment (autoenrollment):

    - The details of the event include the certificate template on which the certificate was issued
    - The name of the certificate template used to issue the certificate should match the certificate template name included in the event
    - The certificate thumbprint and EKUs for the certificate are also included in the event
    - The EKU needed for proper Windows Hello for Business authentication is Kerberos Authentication, in addition to other EKUs provide by the certificate template

Certificates superseded by your new domain controller certificate generate an *archive event* in the Event Log. The archive event contains the certificate template name and thumbprint of the certificate that was superseded by the new certificate.

Certificate Manager

You can use the Certificate Manager console to validate the domain controller has the properly enrolled certificate based on the correct certificate template with the proper EKUs. Use **certlm.msc** to view certificate in the local computers certificate stores. Expand the **Personal** store and view the certificates enrolled for the computer. Archived certificates don't appear in Certificate Manager.

Certutil.exe

You can use `certutil.exe` command to view enrolled certificates in the local computer. Certutil shows enrolled and archived certificates for the local computer. From an elevated command prompt, run the following command:

Windows Command Prompt

```
certutil.exe -q -store my
```

To view detailed information about each certificate in the store, and to validate automatic certificate enrollment enrolled the proper certificates, use the following command:

Windows Command Prompt

```
certutil.exe -q -v -store my
```

Troubleshooting

Windows triggers automatic certificate enrollment for the computer during boot, and when Group Policy updates. You can refresh Group Policy from an elevated command prompt using `gpupdate.exe /force`.

Alternatively, you can forcefully trigger automatic certificate enrollment using `certreq.exe -autoenroll -q` from an elevated command prompt.

Use the event logs to monitor certificate enrollment and archive. Review the configuration, such as publishing certificate templates to issuing certification authority and the *allow* auto enrollment permissions.

Section review and next steps

Before moving to the next section, ensure the following steps are complete:

- ✓ Configure domain controller certificates
- ✓ Supersede existing domain controller certificates
- ✓ Unpublish superseded certificate templates
- ✓ Configure an enrollment agent certificate template
- ✓ Configure an authentication certificate template
- ✓ Publish the certificate templates to the CA
- ✓ Deploy certificates to the domain controllers
- ✓ Validate the domain controllers configuration

Next: configure AD FS >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure Active Directory Federation Services in a hybrid certificate trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** certificate trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

The Windows Hello for Business certificate-based deployments use AD FS as the certificate registration authority (CRA). The CRA is responsible for issuing and revoking certificates to users. Once the registration authority verifies the certificate request, it signs the certificate request using its enrollment agent certificate and sends it to the certificate authority.
The CRA enrolls for an *enrollment agent certificate*, and the Windows Hello for Business *authentication certificate template* is configured to only issue certificates to requests signed with an enrollment agent certificate.

> **Note**
>
> In order for AD FS to verify user certificate requests for Windows Hello for Business, it needs to be able to access the `https://enterpriseregistration.windows.net` endpoint.

Configure the certificate registration authority

Sign-in the AD FS server with *domain administrator* equivalent credentials.

Open a **Windows PowerShell** prompt and type the following command:

PowerShell

```
Set-AdfsCertificateAuthority -EnrollmentAgent -
EnrollmentAgentCertificateTemplate WHFBEnrollmentAgent -
```

```
WindowsHelloCertificateTemplate WHFBAuthentication -
WindowsHelloCertificateProxyEnabled $true
```

> **Note**
>
> If you gave your Windows Hello for Business Enrollment Agent and Windows Hello for Business Authentication certificate templates different names, then replace *WHFBEnrollmentAgent* and *WHFBAuthentication* in the above command with the name of your certificate templates. It's important that you use the template name rather than the template display name. You can view the template name on the **General** tab of the certificate template by using the **Certificate Template** management console (certtmpl.msc). Or, you can view the template name by using the `Get-CATemplate` PowerShell cmdlet on a CA.

Enrollment agent certificate enrollment

AD FS performs its own certificate lifecycle management. Once the registration authority is configured with the proper certificate template, the AD FS server attempts to enroll the certificate on the first certificate request or when the service first starts.

Approximately 60 days prior to enrollment agent certificate's expiration, the AD FS service attempts to renew the certificate until it's successful. If the certificate fails to renew, and the certificate expires, the AD FS server requests a new enrollment agent certificate. You can view the AD FS event logs to determine the status of the enrollment agent certificate.

Group Memberships for the AD FS service account

The AD FS service account must be member of the security group targeted by the authentication certificate template autoenrollment (for example, *Window Hello for Business Users*). The security group provides the AD FS service with the permissions needed to enroll a Windows Hello for Business authentication certificate on behalf of the provisioning user.

> **Tip**
>
> The adfssvc account is the AD FS service account.

Sign-in a domain controller or management workstation with *Domain Admin* equivalent credentials.

1. Open **Active Directory Users and Computers**
2. Search for the security group targeted by the authentication certificate template autoenrollment (for example, *Window Hello for Business Users*)
3. Select the **Members** tab and select **Add**
4. In the **Enter the object names to select** text box, type **adfssvc** or substitute the name of the AD FS service account in your AD FS deployment > **OK**
5. Select **OK** to return to **Active Directory Users and Computers**
6. Restart the AD FS server

> **Note**
>
> For AD FS 2019 and later in a certificate trust model, a known PRT issue exists. You may encounter this error in AD FS Admin event logs: Received invalid Oauth request. The client 'NAME' is forbidden to access the resource with scope 'ugs'. For more information about the isse and its resolution, see Certificate trust provisioning with AD FS broken on windows server 2019.

Section review and next steps

Before moving to the next section, ensure the following steps are complete:

- ✓ Configure the certificate registration authority
- ✓ Update group memberships for the AD FS service account

Next: configure policy settings >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure and enroll in Windows Hello for Business in hybrid certificate trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** certificate trust
- **Join type:** Microsoft Entra join , Microsoft Entra hybrid join

---

Once the prerequisites are met, and the PKI and AD FS configurations are validated, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Configure Windows Hello for Business policy settings

There are two policy settings required to enable Windows Hello for Business in a certificate trust model:

- Use Windows Hello for Business
- Use certificate for on-premises authentication

Another optional, but recommended, policy setting is:

- Use a hardware security device

Use the following instructions to configure your devices using either Microsoft Intune or group policy (GPO).

GPO

You can configure the Use Windows Hello for Business policy setting in the computer or user node of a GPO:

- Deploying the computer node policy setting, results in all users that sign-in to the targeted devices to attempt a Windows Hello for Business enrollment
- Deploying the user node policy setting, results in only the targeted users to attempt a Windows Hello for Business enrollment

If both user and computer policy settings are deployed, the user policy setting has precedence.

> **Tip**
>
> Use the same *Windows Hello for Business Users* security group to assign **Certificate template permissions** to ensure the same members can enroll in the Windows Hello for Business authentication certificate.

Windows Hello for Business provisioning performs the initial enrollment of the Windows Hello for Business authentication certificate. This certificate expires based on the duration configured in the Windows Hello for Business *authentication certificate* template.

The process requires no user interaction, provided the user signs in using Windows Hello for Business. The certificate is renewed in the background before it expires.

To configure a device with group policy, use the Local Group Policy Editor. To configure multiple devices joined to Active Directory, create or edit a group policy object (GPO) and use the following settings:

Expand table

Group policy path	Group policy setting	Value
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business or **User Configuration\Administrative Templates\Windows Components\Windows Hello for Business**	Use Windows Hello for Business	**Enabled**
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business	Use certificate for on-premises authentication	**Enabled**

Group policy path	Group policy setting	Value
or **User Configuration\Administrative Templates\Windows Components\Windows Hello for Business**		
Computer Configuration\Windows Settings\Security Settings\Public Key Policies or **User Configuration\Windows Settings\Security Settings\Public Key Policies**	Certificate Services Client - Auto-Enrollment	- Select **Enabled** from the **Configuration Model** - Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** - Select **Update certificates that use certificate templates**
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business	Use a hardware security device	**Enabled**

> **Note**
>
> The enablement of the *Use a hardware security device* policy setting is optional, but recommended.

Group policies can be linked to domains or organizational units, filtered using security groups, or filtered using WMI filters.

> **Tip**
>
> The best way to deploy the Windows Hello for Business GPO is to use security group filtering. Only members of the targeted security group will provision Windows Hello for Business, enabling a phased rollout. This solution allows linking the GPO to the domain, ensuring the GPO is scoped to all security principals. The security group filtering ensures that only the members of the global group receive and apply the GPO, which results in the provisioning of Windows Hello for Business.

If you deploy Windows Hello for Business configuration using both Group Policy and Intune, Group Policy settings take precedence, and Intune settings are ignored. For more information about policy conflicts, see Policy conflicts from multiple policy sources

More policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after the user profile is loaded and before the user receives their desktop. For the provisioning process to begin, all prerequisite checks must pass.

You can determine the status of the prerequisite checks by viewing the **User Device Registration** admin log under **Applications and Services Logs** > **Microsoft** > **Windows**. This information is also available using the `dsregcmd.exe /status` command from a console. For more information, see dsregcmd.

User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**
3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use

their PIN to sign-in. The user may close the provisioning application and access their desktop

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fhybrid-cert-trust-enroll

After a successful key registration, Windows creates a certificate request using the same key pair to request a certificate. Windows sends the certificate request to the AD FS server for certificate enrollment.

The AD FS registration authority verifies the key used in the certificate request matches the key that was previously registered. On a successful match, the AD FS registration authority signs the certificate request using its enrollment agent certificate and sends it to the certificate authority.

> **Note**
>
> In order for AD FS to verify the key used in the certificate request, it needs to be able to access the `https://enterpriseregistration.windows.net` endpoint.

The CA validates that the certificate is signed by the registration authority. On successful validation, it issues a certificate based on the request and returns the certificate to the AD FS registration authority. The registration authority returns the certificate to Windows where it then installs the certificate in the current user's certificate store. Once this process completes, the Windows Hello for Business provisioning workflow informs the user that they can use their PIN to sign-in through the Action Center.

> **Note**
>
> Windows Server 2016 update KB4088889 (14393.2155) provides synchronous certificate enrollment during hybrid certificate trust provisioning. With this update, users don't need to wait for Microsoft Entra Connect to sync their public key on-premises. Users enroll their certificate during provisioning and can use the certificate for sign-in immediately after completing the provisioning. The update needs to be installed on the federation servers.

Sequence diagrams

To better understand the provisioning flows, review the following sequence diagrams based on the device join and authentication type:

- Provisioning for Microsoft Entra joined devices with managed authentication
- Provisioning for Microsoft Entra joined devices with federated authentication
- Provisioning in a hybrid certificate trust deployment model with federated authentication

To better understand the authentication flows, review the following sequence diagram:

- Microsoft Entra join authentication to Active Directory using a certificate
- Microsoft Entra hybrid join authentication using a certificate

---

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure single sign-on for Microsoft Entra joined devices

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid
- **Trust type:** key trust ,certificate trust
- **Join type:** Microsoft Entra join

Windows Hello for Business combined with Microsoft Entra joined devices makes it easy for users to securely access cloud-based resources using a strong, two-factor credential. As organizations transition resources to the cloud, some resources might remain on-premises, and Microsoft Entra joined devices might need to access them. With additional configurations to the hybrid deployment, you can provide single sign-on to on-premises resources for Microsoft Entra joined devices using Windows Hello for Business, using a key or a certificate.

> **Note**
>
> These steps are not needed when using the cloud Kerberos trust model.

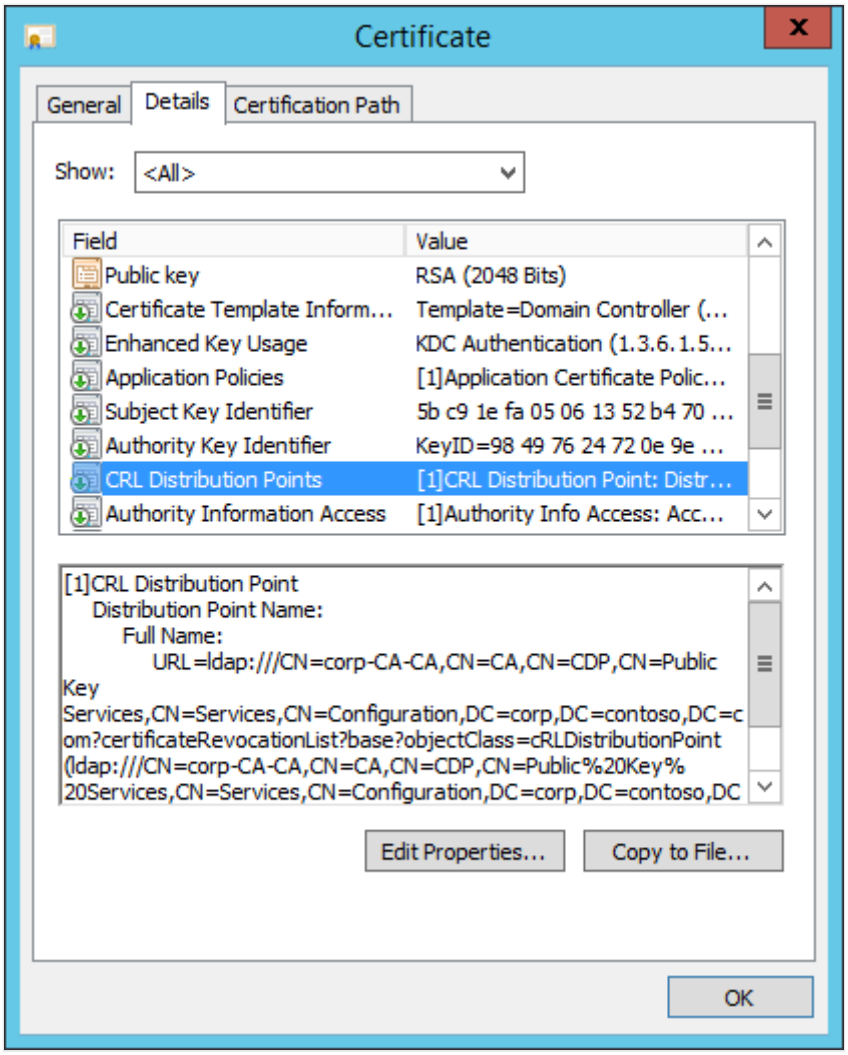
Prerequisites

Unlike Microsoft Entra hybrid joined devices, Microsoft Entra joined devices don't have a relationship with your Active Directory domain. This factor changes the way in which users authenticate to Active Directory. Validate the following configurations to ensure they support Microsoft Entra joined devices:

- ✓ Certificate Revocation List (CRL) Distribution Point
- ✓ Domain controller certificates
- ✓ Network infrastructure in place to reach the on-premises domain controllers. If the machines are external, you can use any VPN solution

CRL Distribution Point (CDP)

Certificates issued by a certificate authority can be revoked. When a certificate authority revokes a certificate, it writes information about the certificate into a *certificate revocation list* (CRL).
During certificate validation, Windows compares the current certificate with information in the CRL to determine if the certificate is valid.



In the screenshot, the CDP property of the domain controller certificate shows an LDAP path. Using Active Directory for domain joined devices provides a highly available CRL distribution point. However, Microsoft Entra joined devices can't read data from Active Directory, and certificate validation doesn't provide an opportunity to authenticate prior to reading the CRL. The authentication becomes a circular problem: the user is attempting to authenticate, but must read Active Directory to complete the authentication, but the user can't read Active Directory because they haven't authenticated.

To resolve this issue, the CRL distribution point must be a location accessible by Microsoft Entra joined devices that don't require authentication. The easiest solution is to publish the CRL distribution point on a web server that uses HTTP (not HTTPS).

If your CRL distribution point doesn't list an HTTP distribution point, then you need to reconfigure the issuing certificate authority to include an HTTP CRL distribution point, preferably first, in the list of distribution points.

> **Note**
>
> If your CA has published both the *Base* and the *Delta CRL*, make sure to publish the *Delta CRL* in the HTTP path. Include web server to fetch the *Delta CRL* by allowing **double escaping** in the (IIS) web server.

Domain controller certificates

Certificate authorities write CDP information in certificates as they're issued. If the distribution point changes, then previously issued certificates must be reissued for the certificate authority to include the new CDP. The domain controller certificate is one the critical components of Microsoft Entra joined devices authenticating to Active Directory.

Why does Windows need to validate the domain controller certificate?

Windows Hello for Business enforces the *strict KDC validation* security feature when authenticating from a Microsoft Entra joined device to a domain. This enforcement imposes more restrictive criteria that must be met by the Key Distribution Center (KDC). When authenticating using Windows Hello for Business on a Microsoft Entra joined device, the Windows client validates the reply from the domain controller by ensuring all of the following are met:

- The domain controller has the private key for the certificate provided
- The root CA that issued the domain controller's certificate is in the device's *Trusted Root Certificate Authorities*
- Use the *Kerberos Authentication certificate template* instead of any other older template
- The domain controller's certificate has the *KDC Authentication* extended key usage (EKU)
- The domain controller's certificate's subject alternate name has a DNS Name that matches the name of the domain
- The domain controller's certificate's signature hash algorithm is *sha256*
- The domain controller's certificate's public key is *RSA (2048 Bits)*

> **Important**

Authenticating from a Microsoft Entra hybrid joined device to a domain using Windows Hello for Business doesn't enforce that the domain controller certificate includes the *KDC Authentication* EKU. If you're adding Microsoft Entra joined devices to an existing domain environment, make sure to verify that your domain controller certificate has been updated to include the *KDC Authentication* EKU.

Configure a CRL distribution point for an issuing CA

Use this set of procedures to update the CA that issues domain controller certificates to include an http-based CRL distribution point.

Configure Internet Information Services to host CRL distribution point

You need to host your new certificate revocation list on a web server so Microsoft Entra joined devices can easily validate certificates without authentication. You can host these files on web servers many ways. The following steps are just one and may be useful for admins unfamiliar with adding a new CRL distribution point.

> **ⓘ Important**
>
> Do not configure the IIS server hosting your CRL distribution point to use https or a server authentication certificate. Clients should access the distribution point using http.

Install the web server

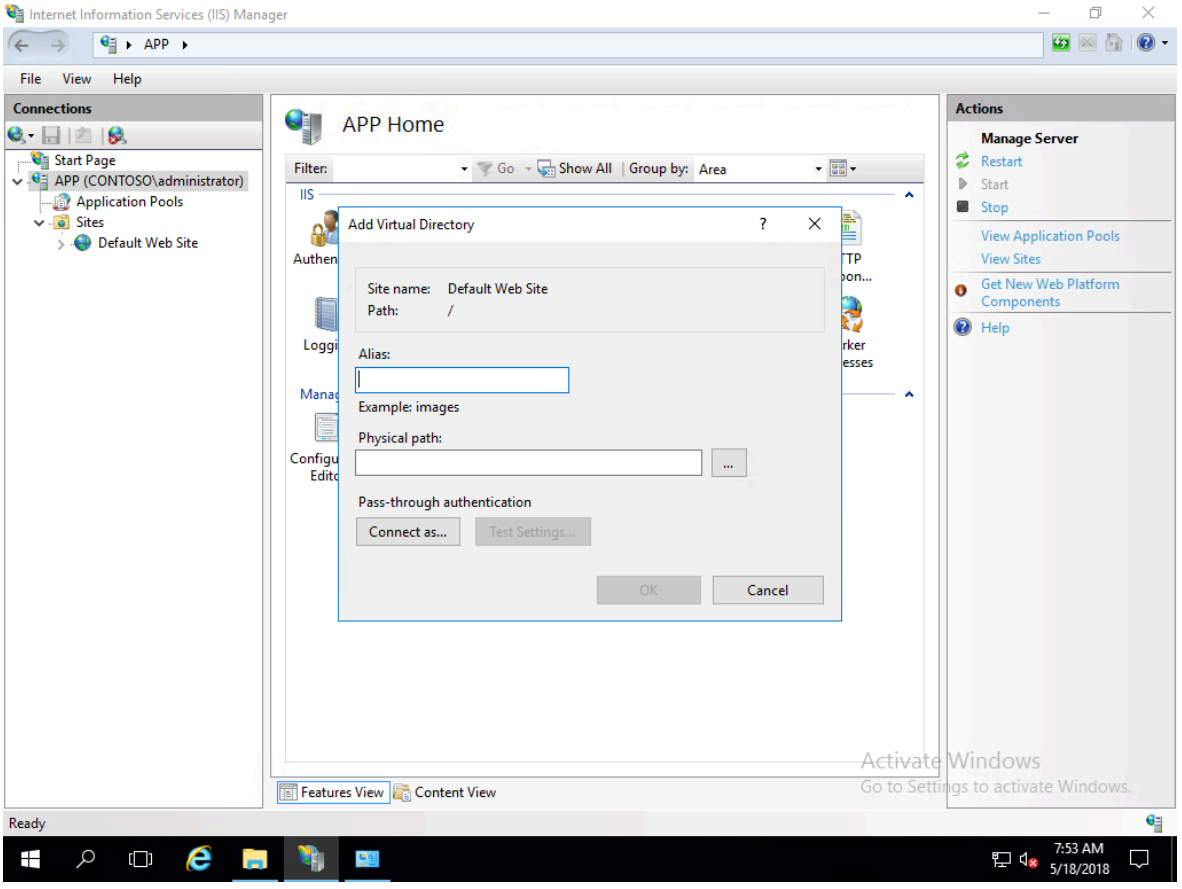
1. Sign-in to your server as a local administrator and start **Server Manager** if it didn't start during your sign in
2. Select the **Local Server** node in the navigation pane. Select **Manage** and select **Add Roles and Features**
3. In the **Add Role and Features Wizard**, select **Server Selection**. Verify the selected server is the local server. Select **Server Roles**. Select the check box next to **Web Server (IIS)**
4. Select **Next** through the remaining options in the wizard, accepting the defaults, and install the Web Server role

Configure the web server

1. From **Windows Administrative Tools**, Open **Internet Information Services (IIS) Manager**

2. Expand the navigation pane to show **Default Web Site**. Select and then right-click **Default Web site** and select **Add Virtual Directory...**

3. In the **Add Virtual Directory** dialog box, type **cdp** in **alias**. For physical path, type or browse for the physical file location where you'll host the certificate revocation list. For this example, the path `c:\cdp` is used. Select **OK**

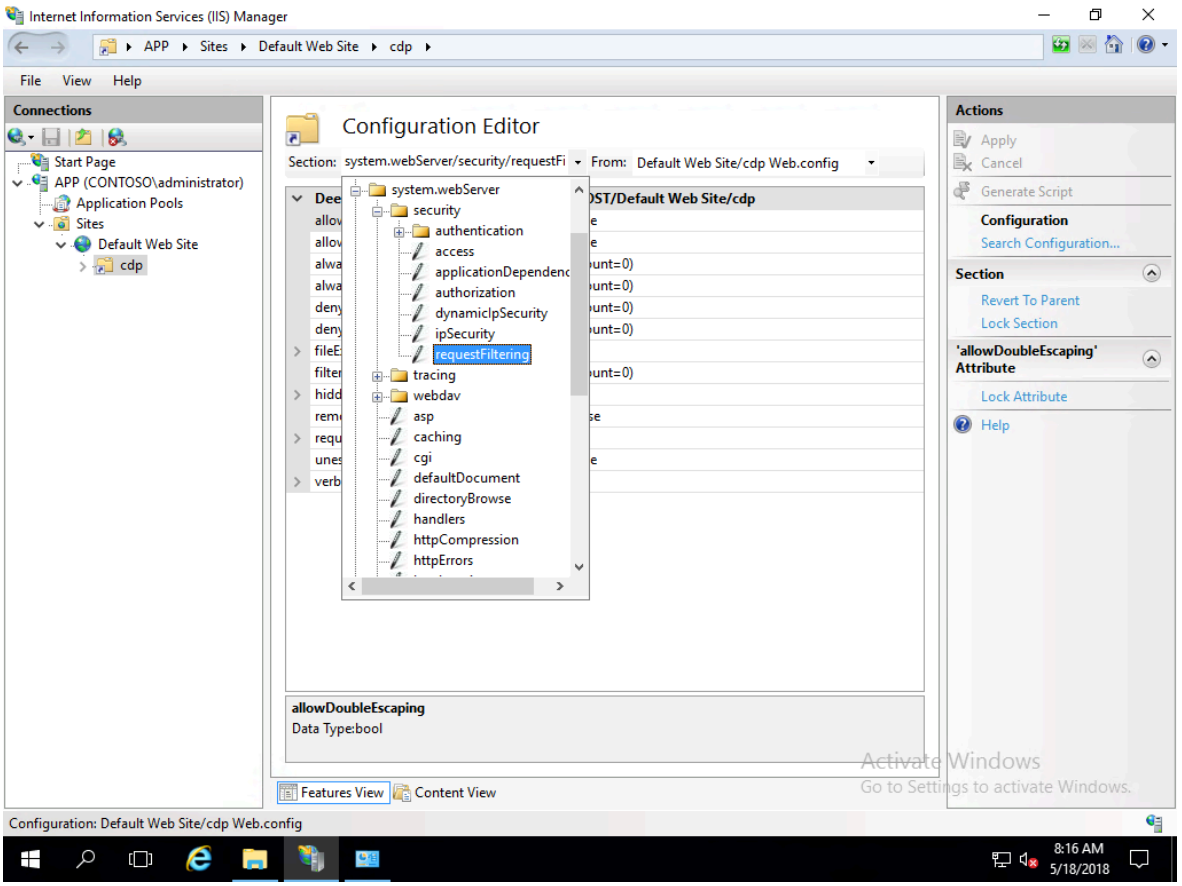


> ⓘ **Note**
>
> Make note of this path as you will use it later to configure share and file permissions.

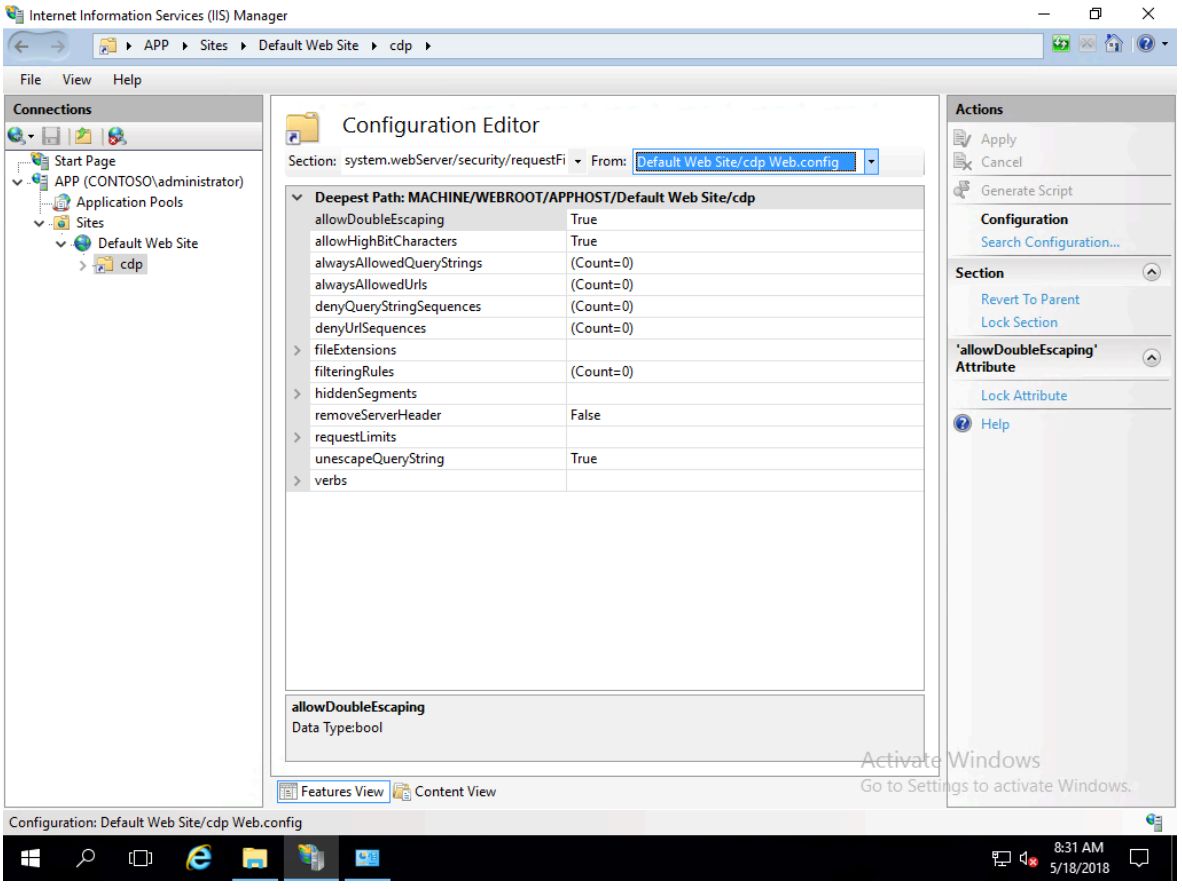
4. Select **CDP** under **Default Web Site** in the navigation pane. Open **Directory Browsing** in the content pane. Select **Enable** in the details pane

5. Select **CDP** under **Default Web Site** in the navigation pane. Open **Configuration Editor**

6. In the **Section** list, navigate to **system.webServer/security/requestFiltering**



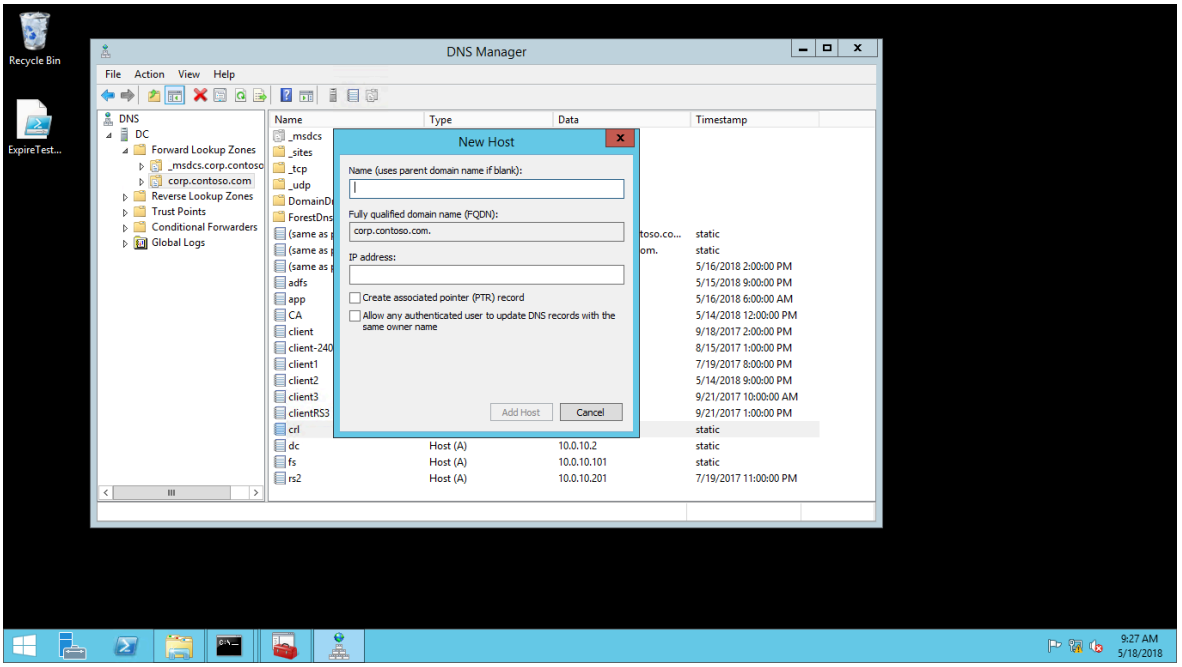
7. In the list of named value-pairs in the content pane, configure **allowDoubleEscaping** to **True**. Select **Apply** in the actions pane



8. Close **Internet Information Services (IIS) Manager**

Create a DNS resource record for the CRL distribution point URL

1. On your DNS server or from an administrative workstation, open **DNS Manager** from **Administrative Tools**
2. Expand **Forward Lookup Zones** to show the DNS zone for your domain. Right-click your domain name in the navigation pane and select **New Host (A or AAAA)...**
3. In the **New Host** dialog box, type **crl** in **Name**. Type the IP address of the web server you configured in **IP Address**. Select **Add Host**. Select **OK** to close the **DNS** dialog box. Select **Done**



4. Close the **DNS Manager**

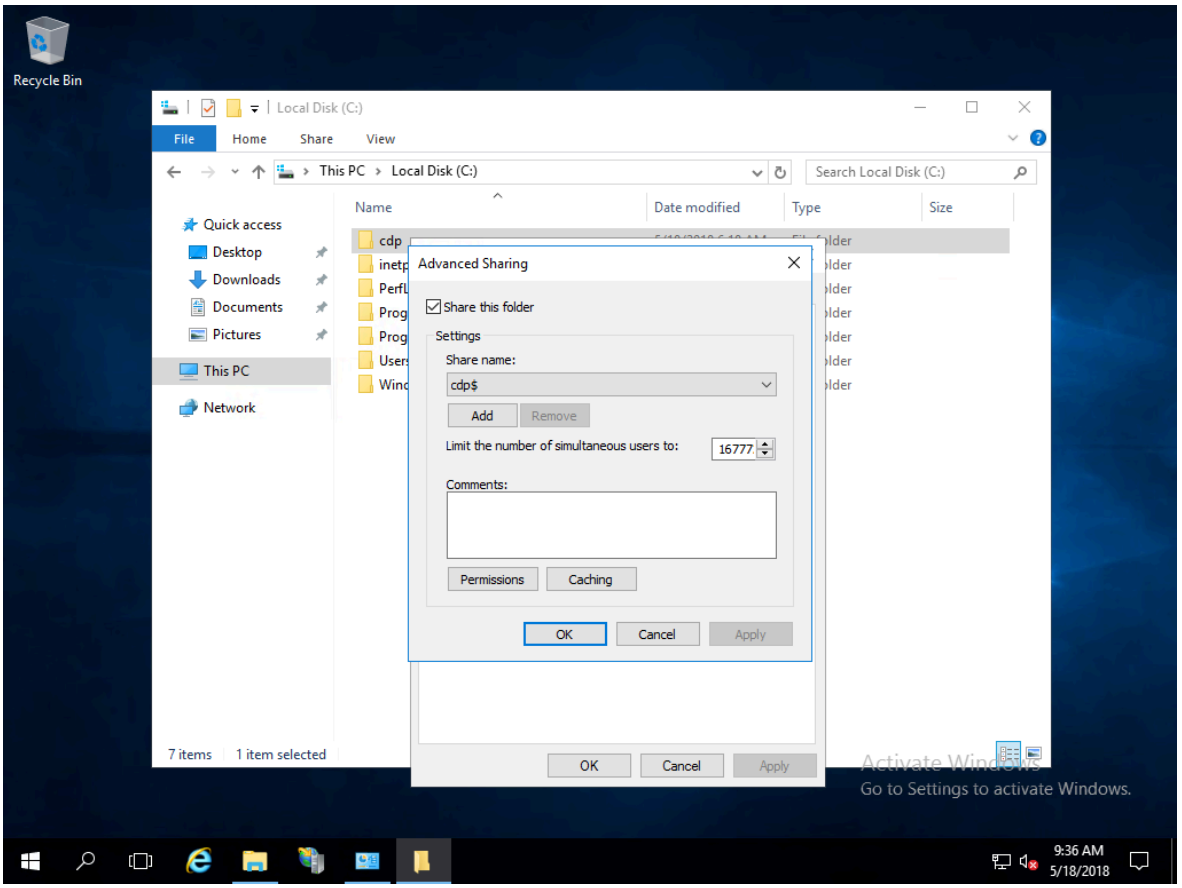
Prepare a file share to host the certificate revocation list

These procedures configure NTFS and share permissions on the web server to allow the certificate authority to automatically publish the certificate revocation list.

Configure the CDP file share

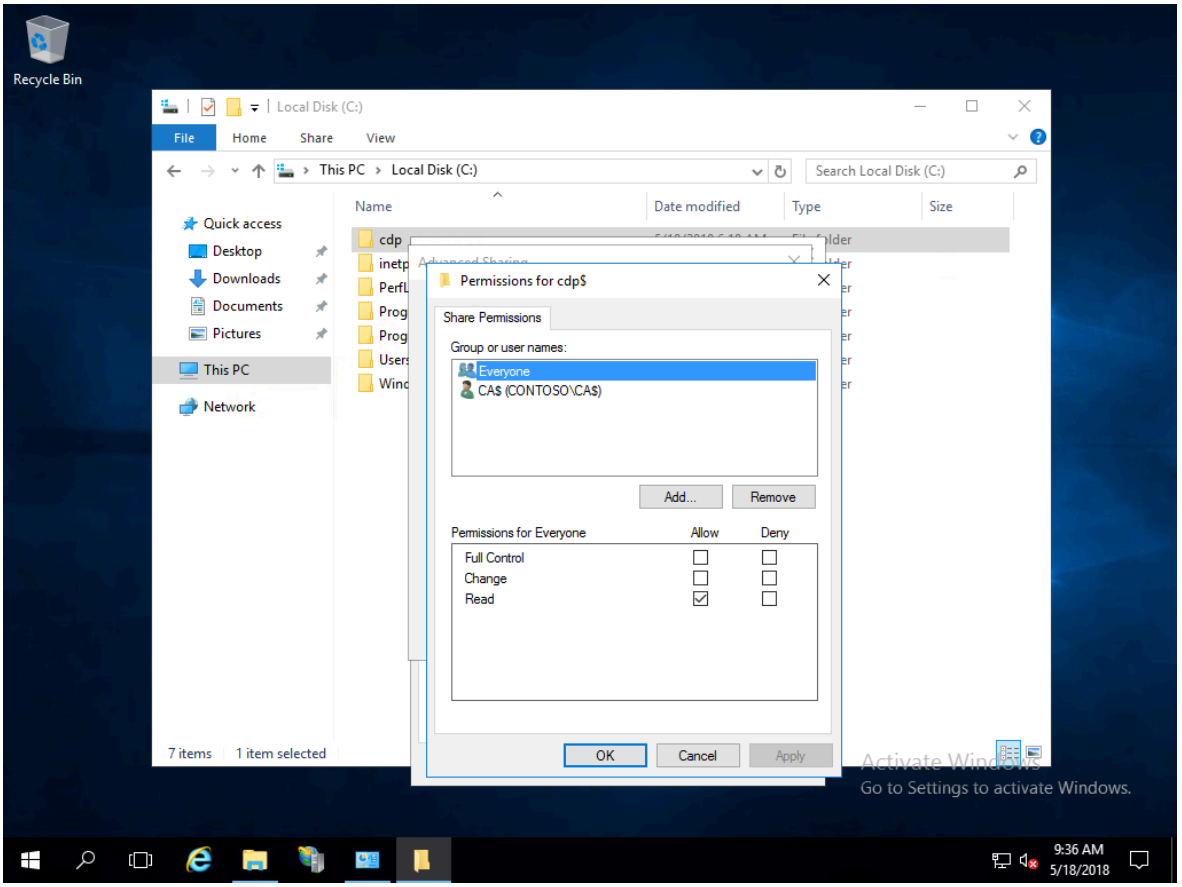
1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Sharing** tab. Select **Advanced Sharing**

3. Select **Share this folder**. Type **cdp$** in **Share name**. Select **Permissions**



4. In the **Permissions for cdp$** dialog box, select **Add**
5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Object Types**. In the **Object Types** dialog box, select **Computers**, and then select **OK**
6. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, in **Enter the object names to select**, type the name of the server running the certificate authority issuing the certificate revocation list, and then select **Check Names**. Select **OK**
7. In the **Permissions for cdp$** dialog box, select the certificate authority from the **Group or user names list**. In the **Permissions for** section, select **Allow** for **Full**

control. Select **OK**

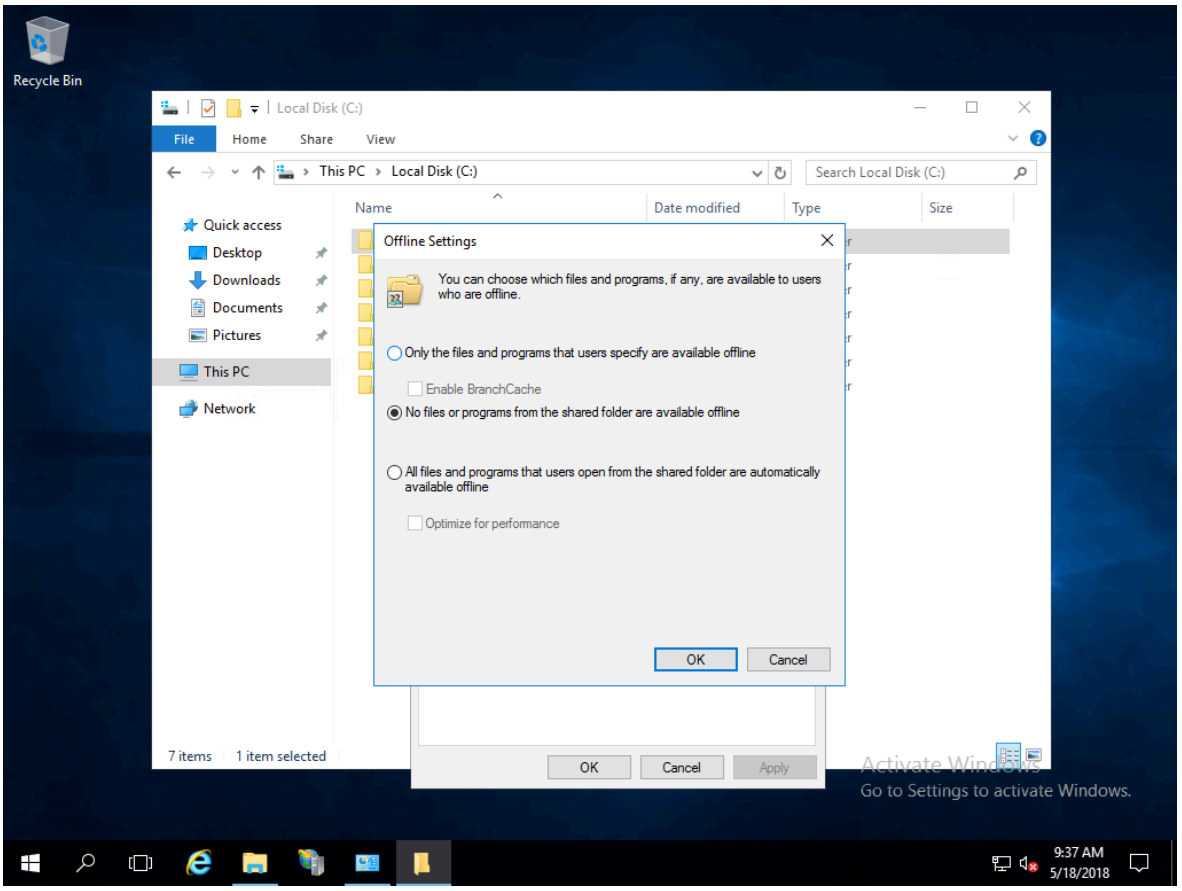


8. In the **Advanced Sharing** dialog box, select **OK**

> **Tip**
>
> Make sure that users can access `\\Server FQDN\sharename`.

Disable Caching

1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Sharing** tab. Select **Advanced Sharing**
3. Select **Caching**. Select **No files or programs from the shared folder are available offline**

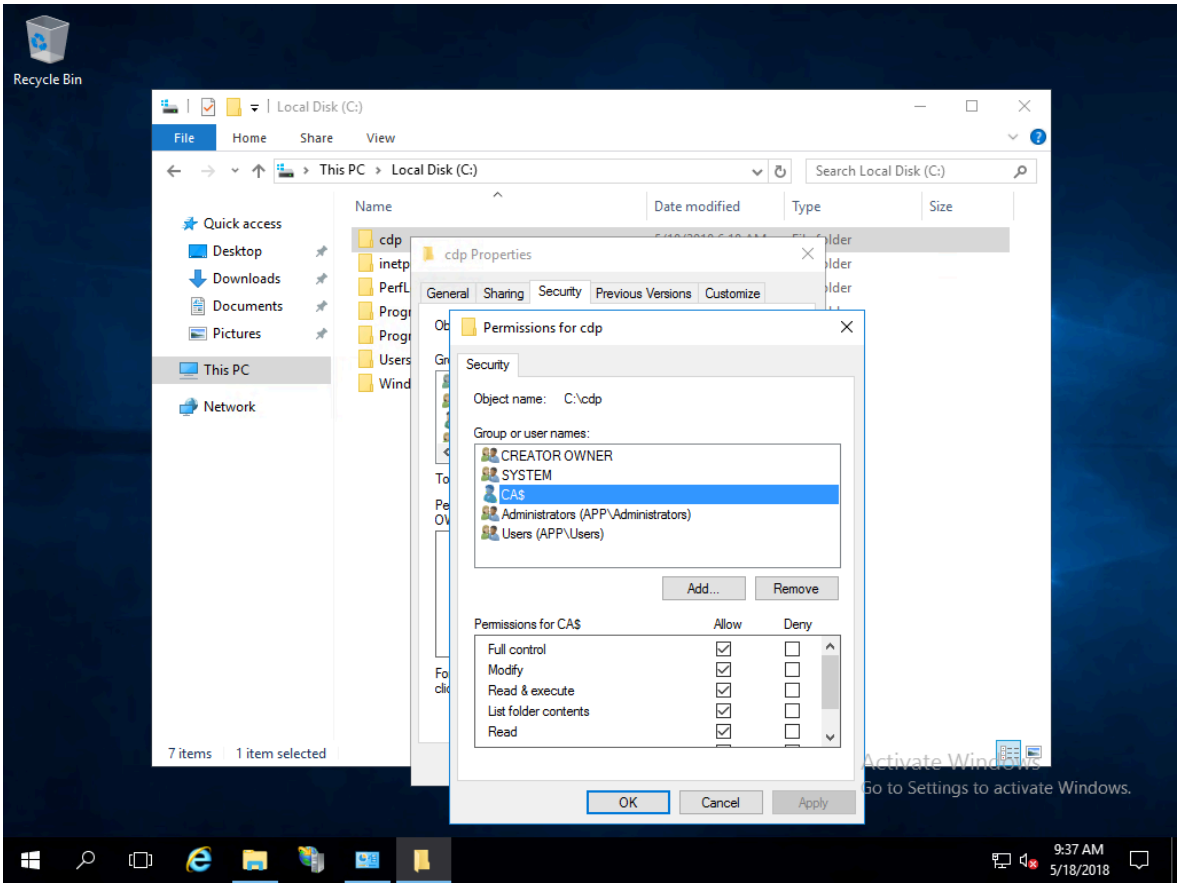


4. Select **OK**

Configure NTFS permission for the CDP folder

1. On the web server, open **Windows Explorer** and navigate to the **cdp** folder you created in step 3 of Configure the Web Server
2. Right-click the **cdp** folder and select **Properties**. Select the **Security** tab
3. On the **Security** tab, select Edit

4. In the **Permissions for cdp** dialog box, select **Add**



5. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, select **Object Types**. In the **Object Types** dialog box, select **Computers**. Select **OK**
6. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, in **Enter the object names to select**, type the name of the certificate authority, and then select **Check Names**. Select **OK**
7. In the **Permissions for cdp** dialog box, select the name of the certificate authority from the **Group or user names** list. In the **Permissions for** section, select **Allow** for **Full control**. Select **OK**
8. Select **Close** in the **cdp Properties** dialog box

Configure the new CDP and publishing location in the issuing CA

The web server is ready to host the CRL distribution point. Now, configure the issuing certificate authority to publish the CRL at the new location and to include the new CRL distribution point.

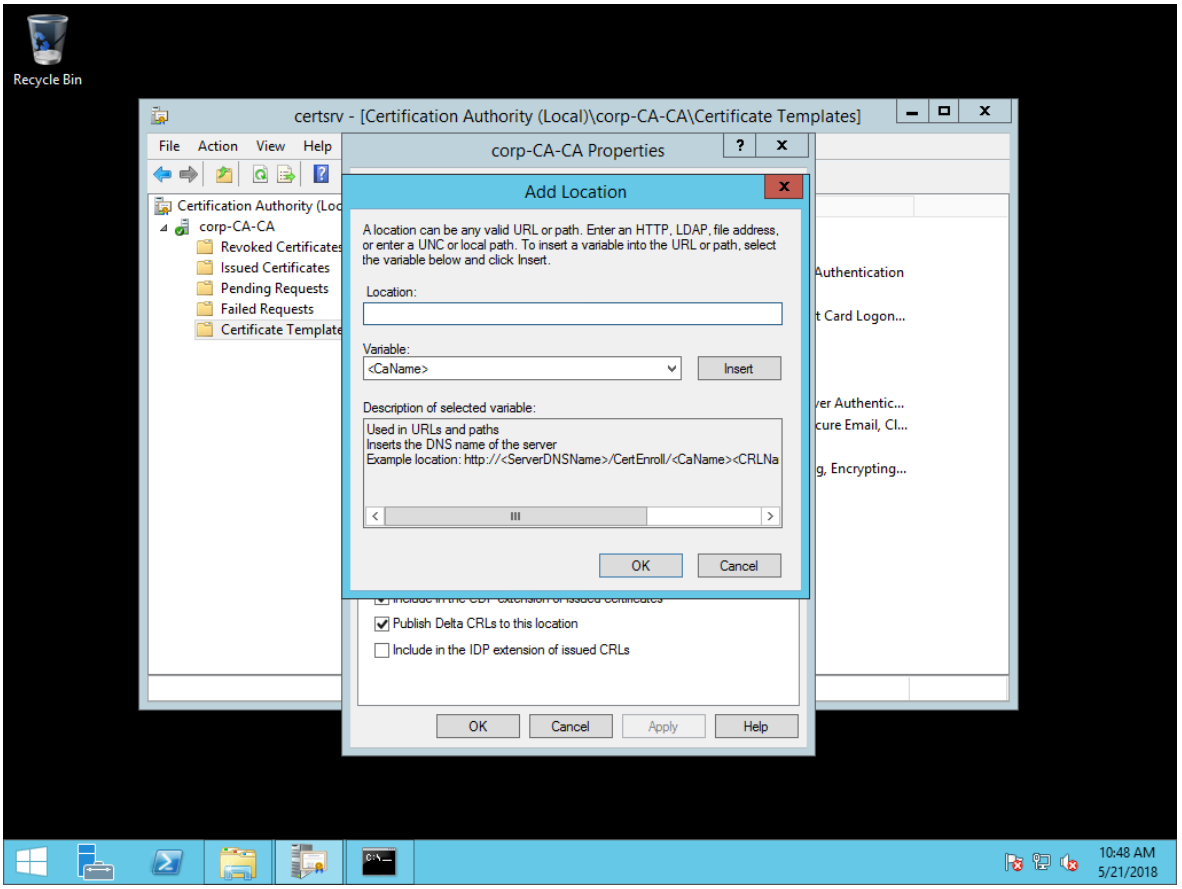
Configure the CRL distribution Point

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certification Authority** console from **Administrative Tools**

2. In the navigation pane, right-click the name of the certificate authority and select **Properties**

3. Select **Extensions**. On the **Extensions** tab, select **CRL Distribution Point (CDP)** from the **Select extension** list

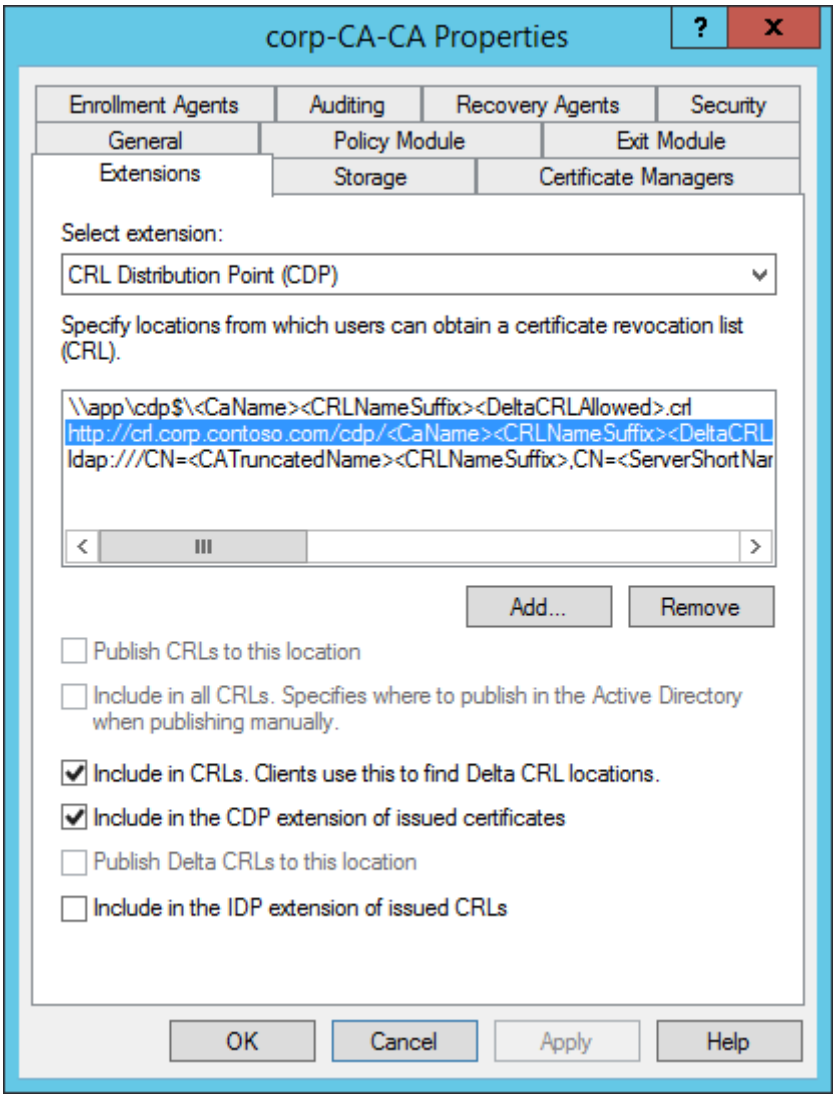
4. On the **Extensions** tab, select **Add**. Type **http://crl.[domainname]/cdp/** in **location**. For example, `<http://crl.corp.contoso.com/cdp/>` or `<http://crl.contoso.com/cdp/>` (don't forget the trailing forward slash)



5. Select <**CaName**> from the **Variable** list and select **Insert**. Select <**CRLNameSuffix**> from the **Variable** list and select **Insert**. Select <**DeltaCRLAllowed**> from the **Variable** list and select **Insert**.

6. Type **.crl** at the end of the text in **Location**. Select **OK**.

7. Select the CDP you just created.



8. Select **Include in CRLs. Clients use this to find Delta CRL locations**.

9. Select **Include in the CDP extension of issued certificates**.

10. Select **Apply** save your selections. Select **No** when ask to restart the service.

> ⓘ **Note**
>
> Optionally, you can remove unused CRL distribution points and publishing locations.

Configure the CRL publishing location

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certificate Authority** console from **Administrative Tools**

2. In the navigation pane, right-click the name of the certificate authority and select **Properties**

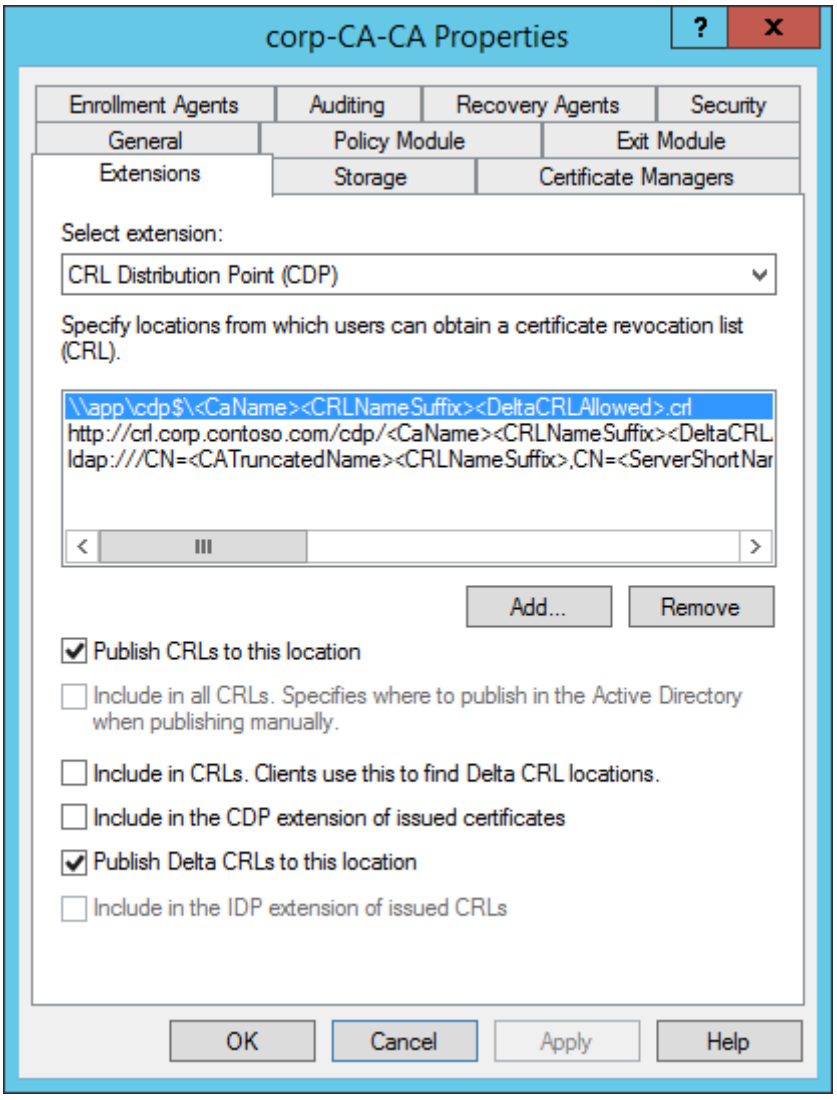
3. Select **Extensions**. On the **Extensions** tab, select **CRL Distribution Point (CDP)** from the **Select extension** list

4. On the **Extensions** tab, select **Add**. Type the computer and share name you create for your CRL distribution point in Configure the CDP file share. For example, **\\app\cdp$** (don't forget the trailing backwards slash)

5. Select **<CaName>** from the **Variable** list and select **Insert**. Select **<CRLNameSuffix>** from the **Variable** list and select **Insert**. Select **<DeltaCRLAllowed>** from the **Variable** list and select **Insert**

6. Type **.crl** at the end of the text in **Location**. Select **OK**

7. Select the CDP you just created.



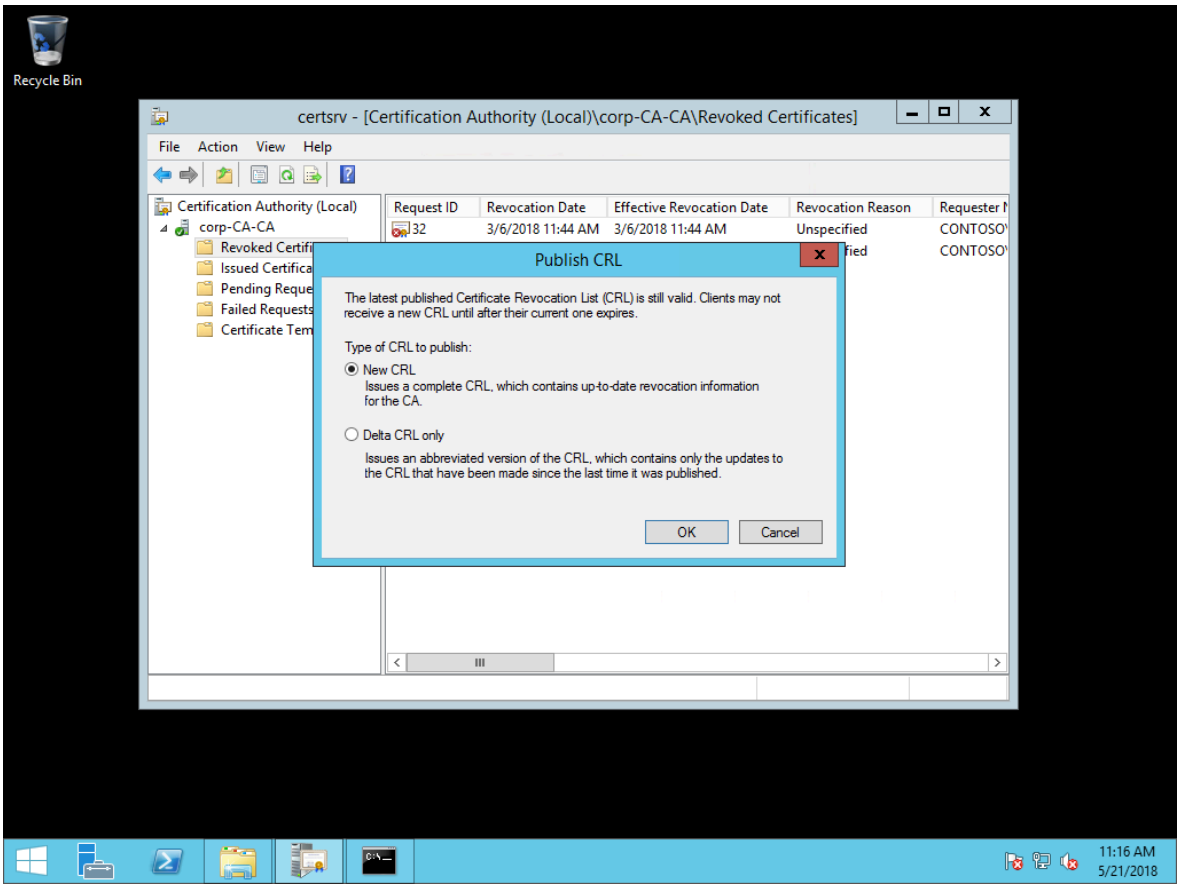
8. Select **Publish CRLs to this location**

9. Select **Publish Delta CRLs to this location**

10. Select **Apply** save your selections. Select **Yes** when ask to restart the service. Select **OK** to close the properties dialog box

Publish a new CRL

1. On the issuing certificate authority, sign-in as a local administrator. Start the **Certificate Authority** console from **Administrative Tools**.
2. In the navigation pane, right-click **Revoked Certificates**, hover over **All Tasks**, and select **Publish**



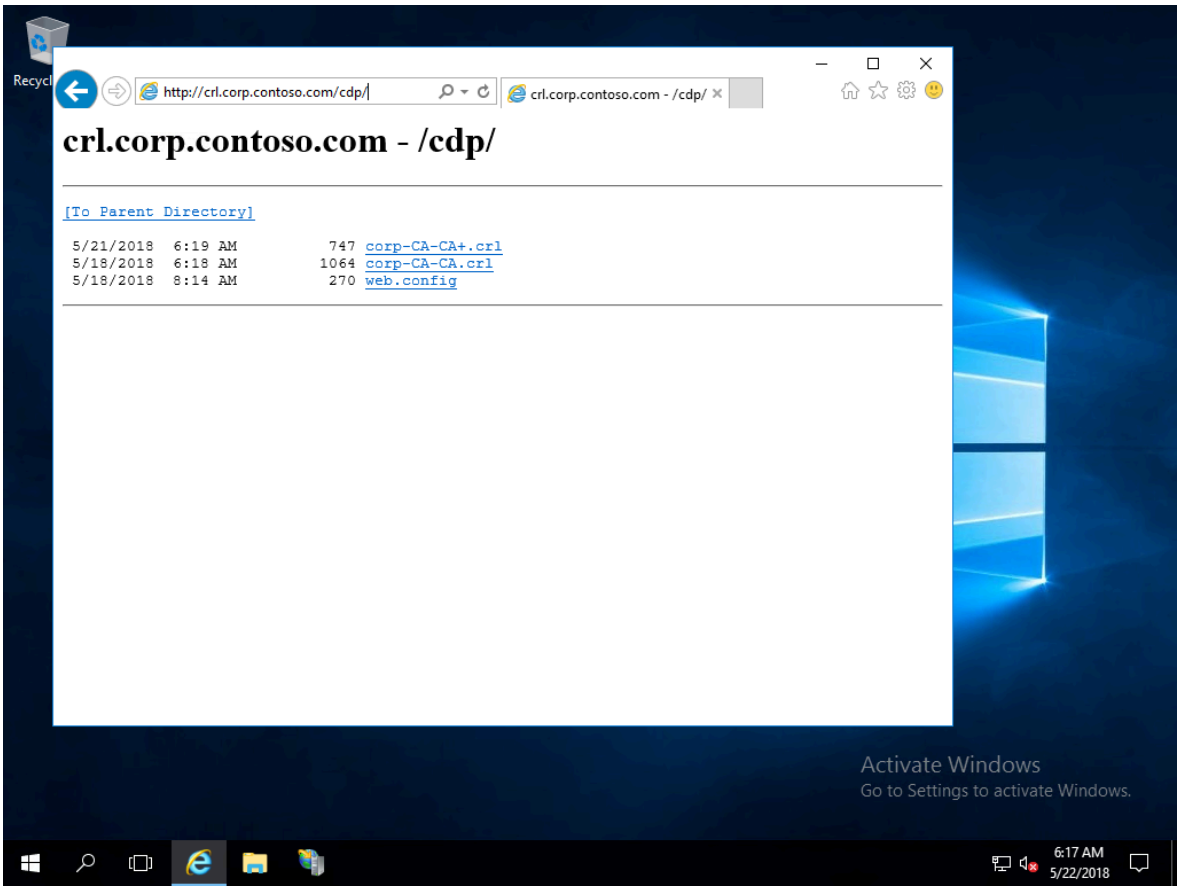
.

3. In the **Publish CRL** dialog box, select **New CRL** and select **OK**.

Validate CDP Publishing

Validate the new CRL distribution point is working.

1. Open a web browser. Navigate to `http://crl.[yourdomain].com/cdp`. You should see two files created from publishing the new CRL

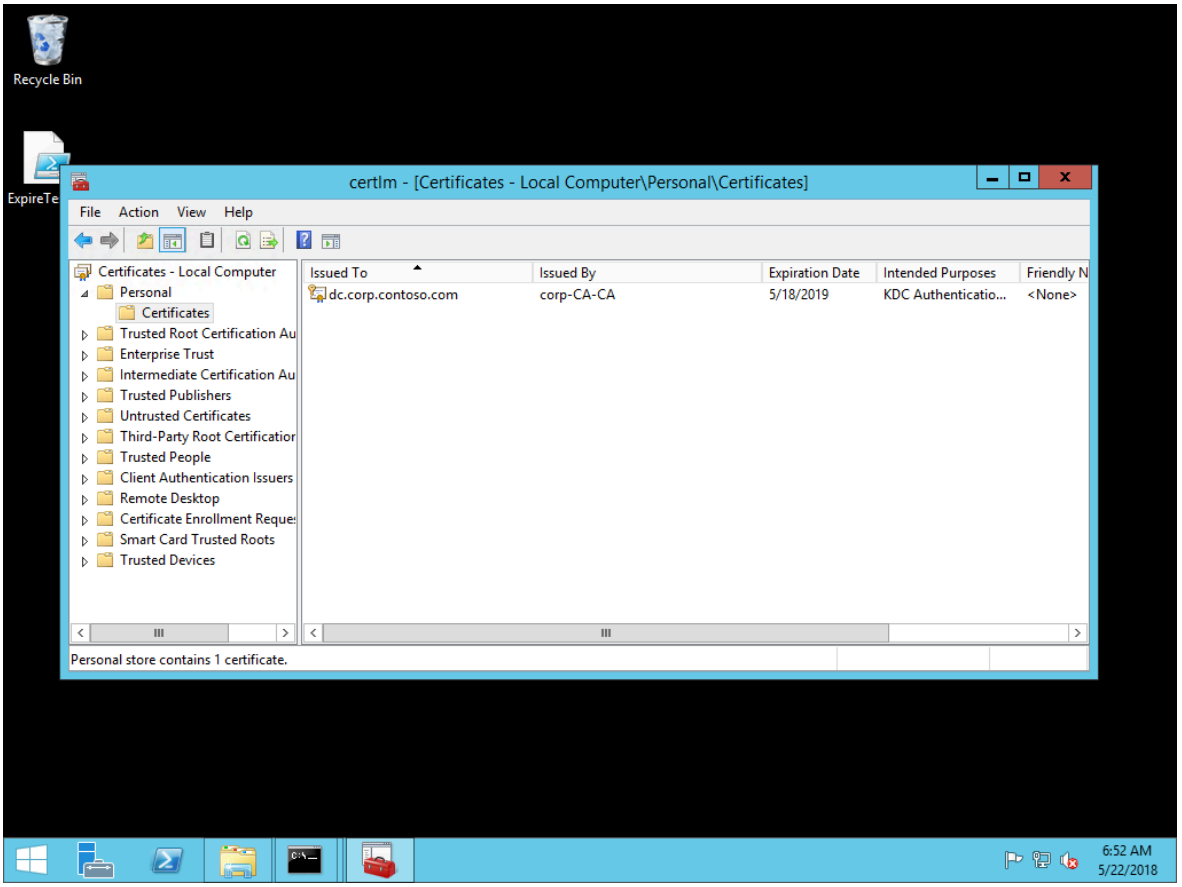


Reissue domain controller certificates

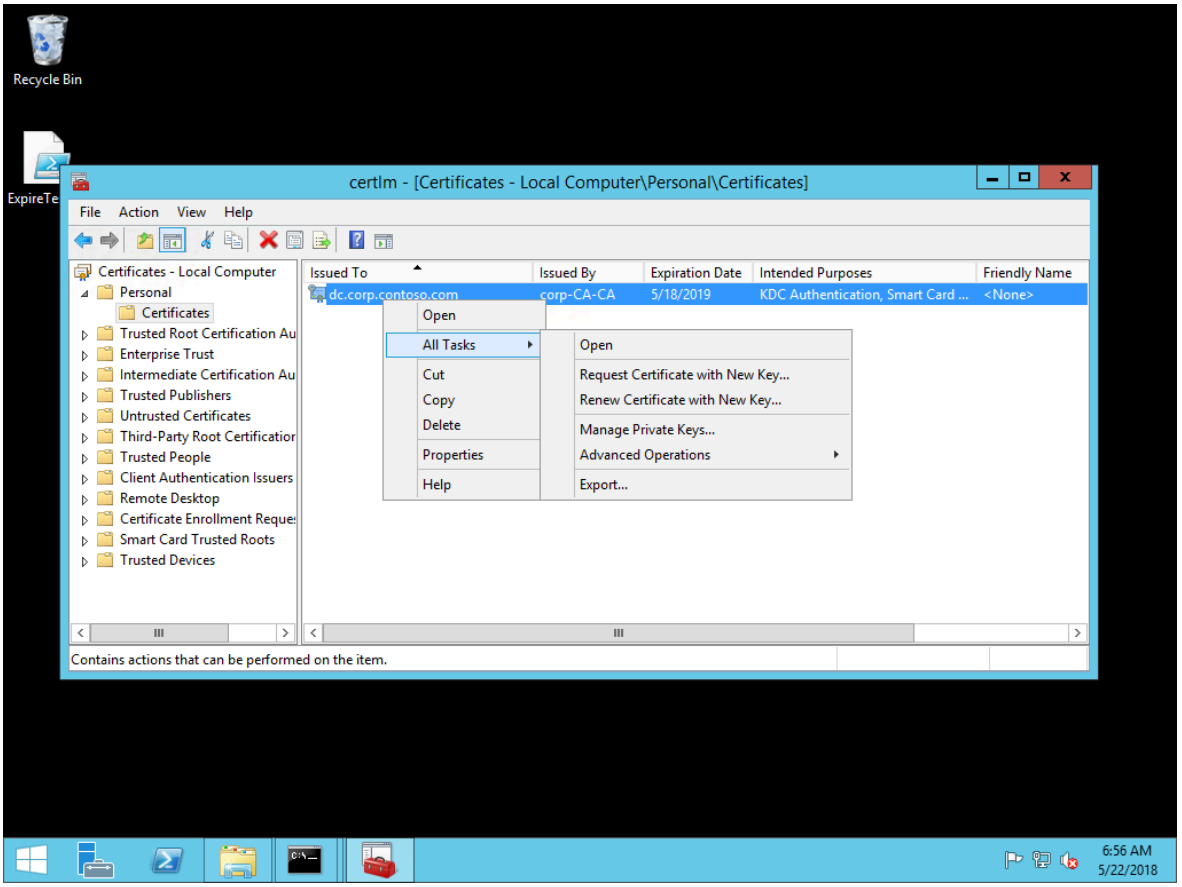
With the CA properly configured with a valid HTTP-based CRL distribution point, you need to reissue certificates to domain controllers as the old certificate doesn't have the updated CRL distribution point.

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, select the existing domain controller certificate that includes **KDC Authentication**

in the list of **Intended Purposes**



4. Right-click the selected certificate. Hover over **All Tasks** and then select **Renew Certificate with New Key...**. In the **Certificate Enrollment** wizard, select **Next**



5. In the **Request Certificates** page of the wizard, verify the selected certificate has the correct certificate template and ensure the status is available. Select **Enroll**
6. After the enrollment completes, select **Finish** to close the wizard

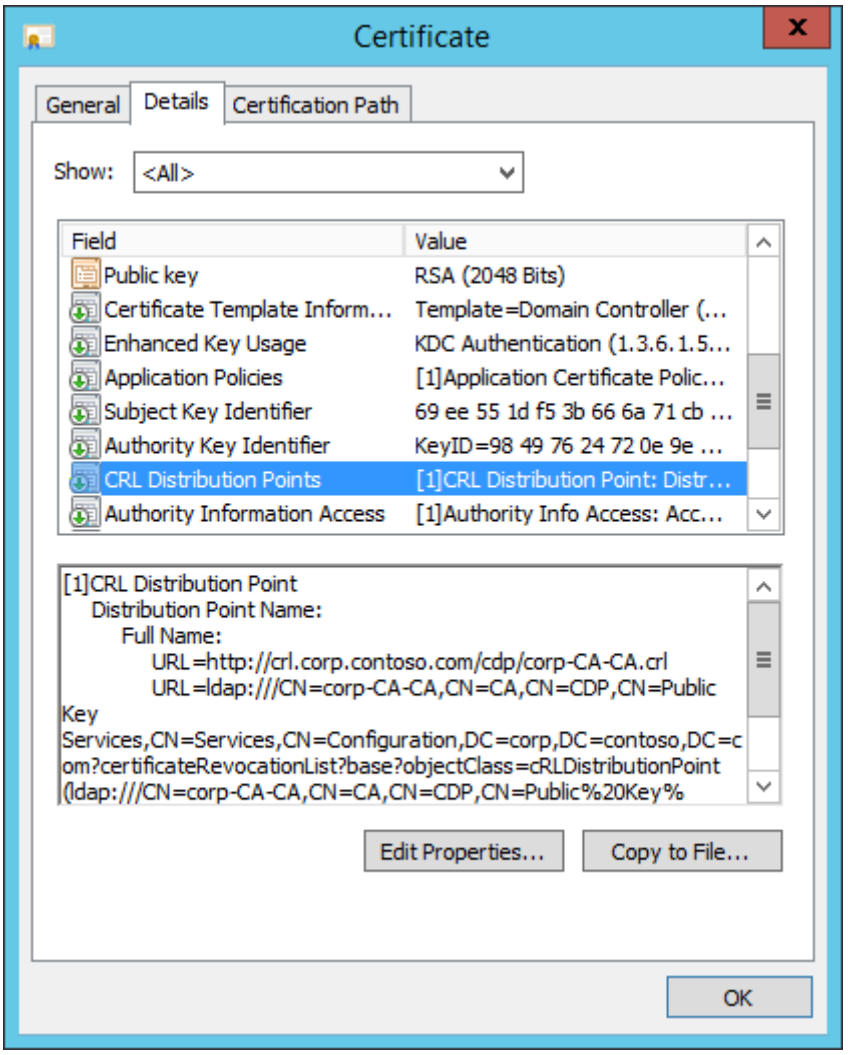
7. Repeat this procedure on all your domain controllers

> **Note**
>
> You can configure domain controllers to automatically enroll and renew their certificates. Automatic certificate enrollment helps prevent authentication outages due to expired certificates. Refer to the Windows Hello Deployment Guides to learn how to deploy automatic certificate enrollment for domain controllers.

> **Important**
>
> If you are not using automatic certificate enrollment, create a calendar reminder to alert you two months before the certificate expiration date. Send the reminder to multiple people in the organization to ensure more than one or two people know when these certificates expire.

Validate CDP in the new certificate

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, double-click the existing domain controller certificate includes **KDC Authentication** in the list of **Intended Purposes**
4. Select the **Details** tab. Scroll down the list until **CRL Distribution Points** is visible in the **Field** column of the list. Select **CRL Distribution Point**
5. Review the information below the list of fields to confirm the new URL for the CRL distribution point is present in the certificate. Select **OK**



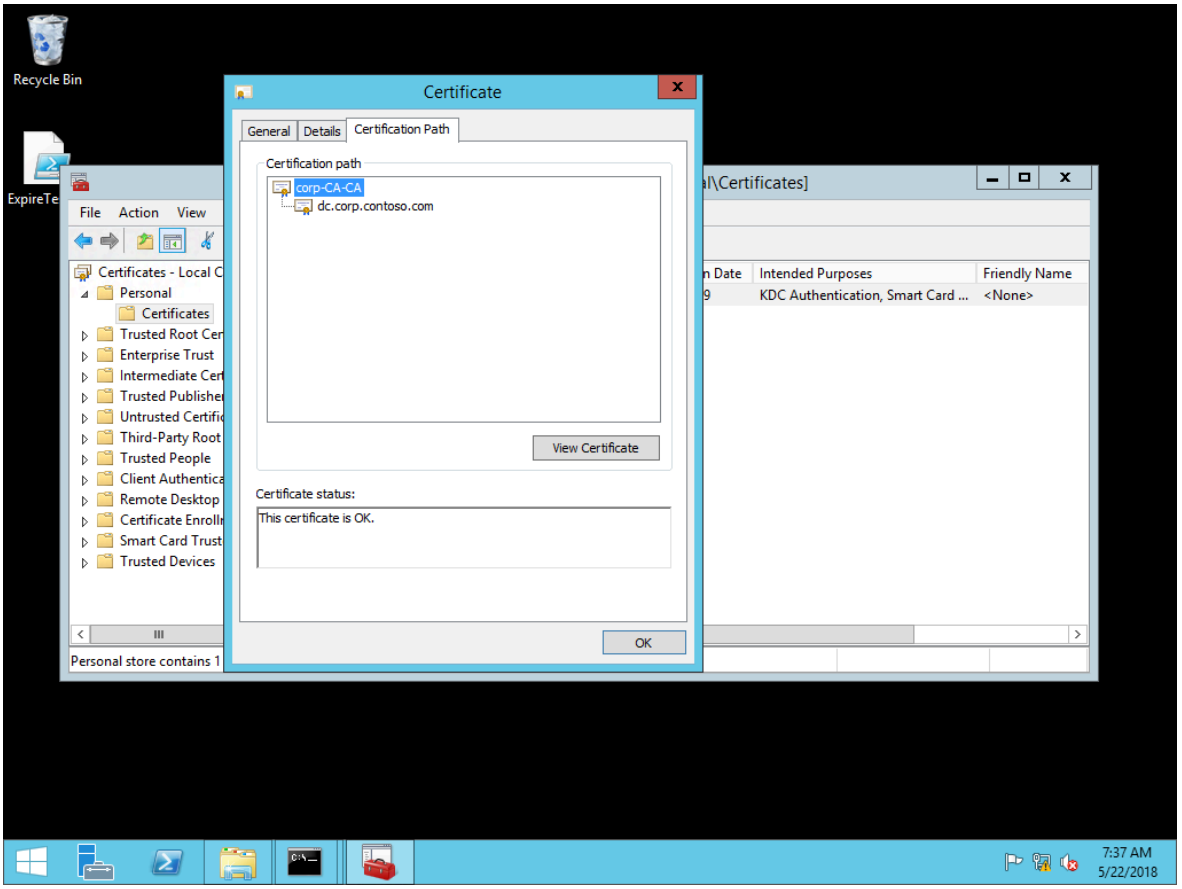
Deploy the root CA certificate to Microsoft Entra joined devices

The domain controllers have a certificate that includes the new CRL distribution point. Next, you need the enterprise root certificate so you can deploy it to Microsoft Entra joined devices. When you deploy the enterprise root certificates to a device, it ensures the device trusts any certificates issued by the certificate authority. Without the certificate, Microsoft Entra joined devices don't trust domain controller certificates and authentication fails.

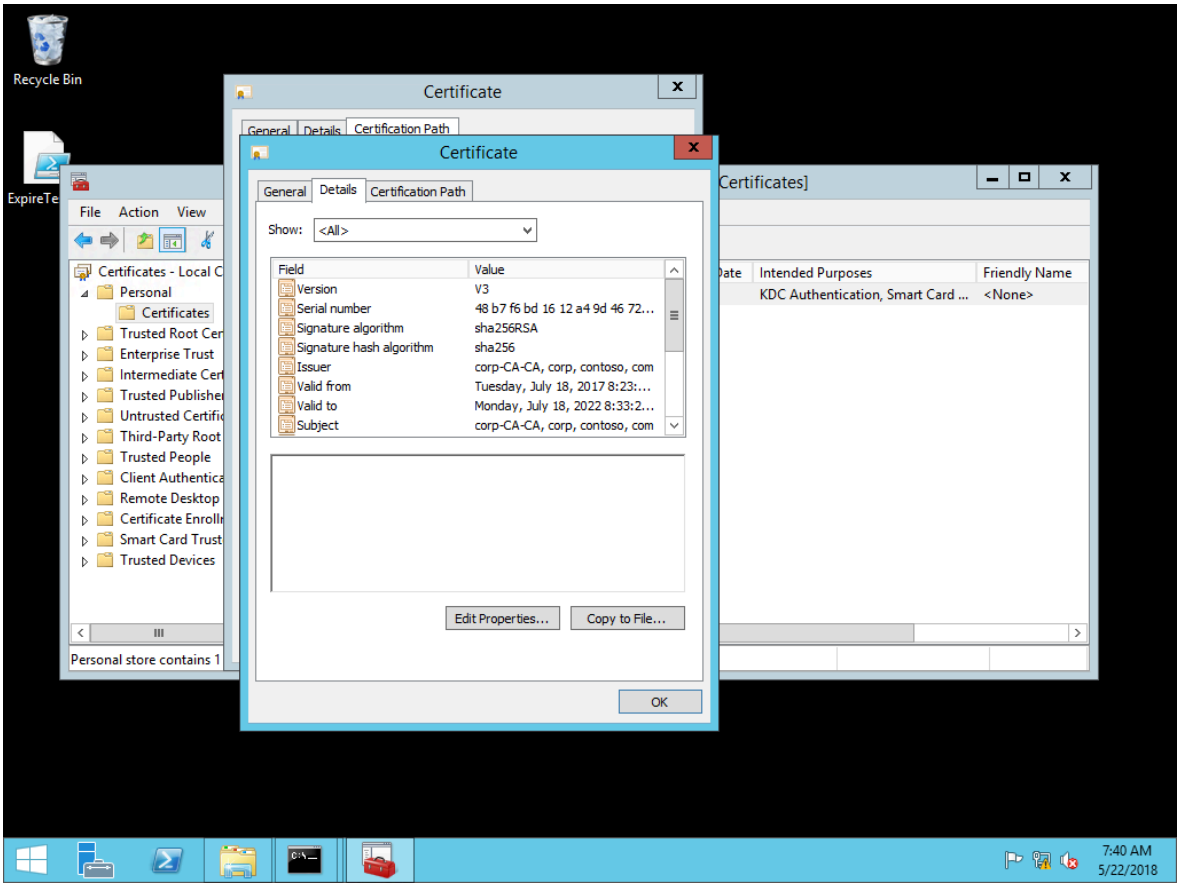
Export the enterprise root certificate

1. Sign-in a domain controller using administrative credentials
2. Open the **Run** dialog box. Type **certlm.msc** to open the **Certificate Manager** for the local computer
3. In the navigation pane, expand **Personal**. Select **Certificates**. In the details pane, double-click the existing domain controller certificate includes **KDC Authentication** in the list of **Intended Purposes**

4. Select the **Certification Path** tab. In the **Certification path** view, select the topmost node and select **View Certificate**

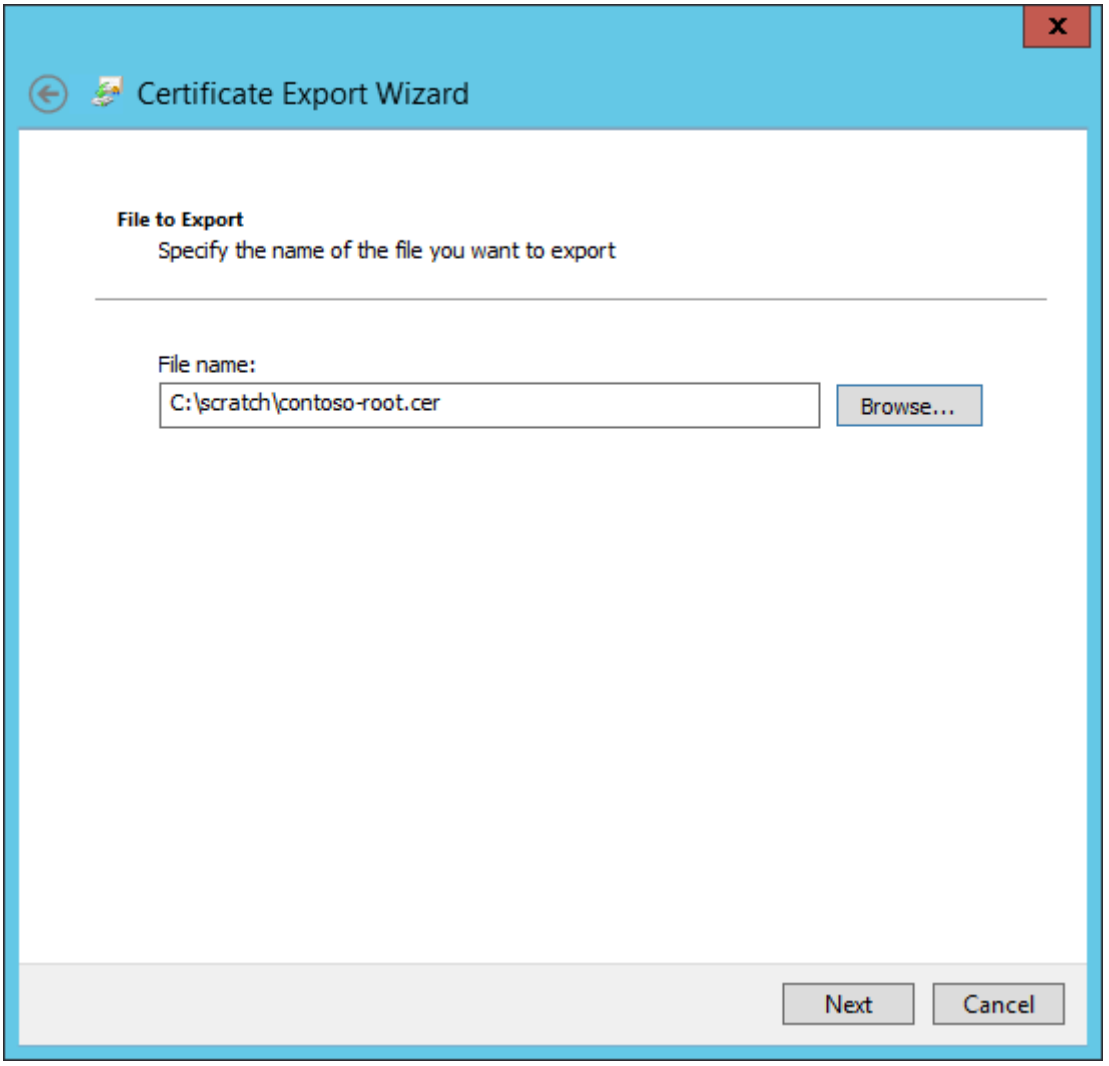


5. In the new **Certificate** dialog box, select the **Details** tab. Select **Copy to File**



6. In the **Certificate Export Wizard**, select **Next**
7. On the **Export File Format** page of the wizard, select **Next**

8. On the **File to Export** page in the wizard, type the name and location of the root certificate and select **Next**. Select **Finish** and then select **OK** to close the success dialog box



9. Select **OK** two times to return to the **Certificate Manager** for the local computer. Close the **Certificate Manager**

Deploy the certificate via Intune

To configure devices with Microsoft Intune, use a custom policy:

1. Go to the Microsoft Intune admin center
2. Select **Devices** > **Configuration profiles** > **Create profile**
3. Select **Platform** > **Windows 8.1 and later** and **Profile type** > **Trusted certificate**
4. Select **Create**
5. In **Configuration settings**, select the folder icon and browse for the enterprise root certificate file. Once the file is selected, select **Open** to upload it to Intune
6. Under **Destination store** dropdown, select **Computer certificate store - Root**
7. Select **Next**
8. Under **Assignment**, select a security group that contains as members the devices or users that you want to configure > **Next**

9. Review the policy configuration and select **Create**

If you plan on using certificates for on-premises single-sign on, perform the additional steps in Using Certificates for On-premises Single-sign On. Otherwise, you can sign in to a Microsoft Entra joined device with Windows Hello for Business and test SSO to an on-premises resource.

Feedback

Was this page helpful? Yes No

Provide product feedback

Using Certificates for Microsoft Entra joined on-premises single-sign on

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** hybrid ⓘ
- **Trust type:** certificate trust ⓘ
- **Join type:** Microsoft Entra join ⓘ

If you plan to use certificates for on-premises single-sign on, then follow these **additional** steps to configure the environment to enroll Windows Hello for Business certificates for Microsoft Entra joined devices.

> ⓘ **Important**
>
> Ensure you have performed the configurations in **Microsoft Entra joined devices for On-premises Single-Sign On** before you continue.

Steps you'll perform include:

- ✓ Prepare Microsoft Entra Connect
- ✓ Prepare the Network Device Enrollment Services Service Account
- ✓ Prepare Active Directory Certificate Services
- ✓ Install the Network Device Enrollment Services Role
- ✓ Configure Network Device Enrollment Services to work with Microsoft Intune
- ✓ Download, Install and Configure the Intune Certificate Connector
- ✓ Create and Assign a Simple Certificate Enrollment Protocol (SCEP) Certificate Profile

Requirements

You must install and configure additional infrastructure to provide Microsoft Entra joined devices with on-premises single-sign on.

- An existing Windows Server Enterprise Certificate Authority
- A domain joined Windows Server that hosts the Network Device Enrollment Services (NDES) role

High Availability

The NDES server role acts as a certificate registration authority (CRA). Certificate registration servers enroll certificates on behalf of the user. Users request certificates from the NDES service rather than directly from the issuing certificate authority.

The architecture of the NDES server prevents it from being clustered or load balanced for high availability. To provide high availability, you need to install more than one identically configured NDES servers, and use Microsoft Intune to load balance then (in round-robin fashion).

The Network Device Enrollment Service (NDES) server role can issue up to three unique certificate templates. The server role accomplishes this by mapping the purpose of the certificate request to a configured certificate template. The certificate request purpose has three options:

- Signature
- Encryption
- Signature and Encryption

If you need to deploy more than three types of certificates to the Microsoft Entra joined device, you need additional NDES servers. Alternatively, consider consolidating certificate templates to reduce the number of certificate templates.

Network Requirements

All communication occurs securely over port 443.

Prepare Microsoft Entra Connect

Successful authentication to on-premises resources using a certificate requires the certificate to provide a hint about the on-premises domain. The hint can be the user's Active Directory distinguished name as the subject of the certificate, or the hint can be the user's user principal name where the suffix matches the Active Directory domain name.

Most environments change the user principal name suffix to match the organization's external domain name (or vanity domain), which prevents the user principal name as a hint to locate a domain controller. Therefore, the certificate needs the user's on-premises distinguished name in the subject to properly locate a domain controller.

To include the on-premises distinguished name in the certificate's subject, Microsoft Entra Connect must replicate the Active Directory **distinguishedName** attribute to the Microsoft Entra ID **onPremisesDistinguishedName** attribute. Microsoft Entra Connect version 1.1.819 includes the proper synchronization rules needed for these attributes.

Verify Microsoft Entra Connect version

Sign-in to computer running Microsoft Entra Connect with access equivalent to *local administrator*.

1. Open **Synchronization Services** from the **Microsoft Entra Connect** folder.
2. In the **Synchronization Service Manager**, select **Help** and then select **About**.
3. If the version number isn't **1.1.819** or later, then upgrade Microsoft Entra Connect to the latest version.

Verify the onPremisesDistinguishedName attribute is synchronized

The easiest way to verify that the onPremisesDistingushedNamne attribute is synchronized is to use the Graph Explorer for Microsoft Graph.

1. Open a web browser and navigate to Graph Explorer.

2. Select **Sign in to Graph Explorer** and provide Microsoft Entra ID credentials.

   > ⓘ Note
   >
   > To successfully query the Graph API, adequate permissions must be granted

3. Select **Modify permissions (Preview)**. Scroll down and locate **User.Read.All** (or any other required permission) and select **Consent**. You'll now be prompted for delegated permissions consent

4. In the Graph Explorer URL, enter `https://graph.microsoft.com/v1.0/users/[userid]?$select=displayName,userPrincipalName,onPremisesDistinguishedName`, where **[userid]** is the user principal name of a user in Microsoft Entra ID. Select **Run query**.

   > ⓘ Note

Because the v1.0 endpoint of the Graph API only provides a limited set of parameters, we will use the $select Optional OData query parameter. For convenience, it is possible to switch the API version selector from **v1.0** to **beta** before performing the query. This will provide all available user information, but remember, **beta** endpoint queries should not be used in production scenarios.

Request

msgraph

```
GET https://graph.microsoft.com/v1.0/users/{id | userPrincipalName}?
$select=displayName,userPrincipalName,onPremisesDistinguishedName
```

5. In the returned results, review the JSON data for the **onPremisesDistinguishedName** attribute. Ensure the attribute has a value and that the value is accurate for the given user. If the **onPremisesDistinguishedName** attribute isn't synchronized the value will be **null**.

Response

HTTP

```
HTTP/1.1 200 OK
Content-type: application/json

{
    "@odata.context":
"https://graph.microsoft.com/v1.0/$metadata#users(displayName,userPrinc
ipalName,onPremisesDistinguishedName)/$entity",
    "displayName": "Nestor Wilke",
    "userPrincipalName": "NestorW@contoso.com",
    "onPremisesDistinguishedName" : "CN=Nestor
Wilke,OU=Operations,DC=contoso,DC=com"
}
```

Prepare the Network Device Enrollment Services (NDES) Service Account

Create the NDES Servers global security group

The deployment uses the **NDES Servers** security group to assign the NDES service the proper user right assignments.

Sign-in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. Open **Active Directory Users and Computers**.
2. Expand the domain node from the navigation pane.
3. Right-click the **Users** container. Hover over **New** and select **Group**.
4. Type **NDES Servers** in the **Group Name** text box.
5. Select **OK**.

Add the NDES server to the NDES Servers global security group

Sign-in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. Open **Active Directory Users and Computers**.
2. Expand the domain node from the navigation pane.
3. Select **Computers** from the navigation pane. Right-click the name of the NDES server that will host the NDES server role. Select **Add to a group**.
4. Type **NDES Servers** in **Enter the object names to select**. Select **OK**. Select **OK** on the **Active Directory Domain Services** success dialog.

> **Note**
>
> For high-availability, you should have more than one NDES server to service Windows Hello for Business certificate requests. You should add additional Windows Hello for Business NDES servers to this group to ensure they receive the proper configuration.

Create the NDES Service Account

The Network Device Enrollment Services (NDES) role runs under a service account. Typically, it's preferential to run services using a Group Managed Service Account (GMSA). While the NDES role can be configured to run using a GMSA, the Intune Certificate Connector wasn't designed nor tested using a GMSA and is considered an unsupported configuration. The deployment uses a normal services account.

Sign-in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. In the navigation pane, expand the node that has your domain name. Select **Users**.
2. Right-click the **Users** container. Hover over **New** and then select **User**. Type **NDESSvc** in **Full Name** and **User logon name**. Select **Next**.
3. Type a secure password in **Password**. Confirm the secure password in **Confirm Password**. Clear **User must change password at next logon**. Select **Next**.
4. Select **Finish**.

> **ⓘ Important**
>
> Configuring the service's account password to **Password never expires** may be more convenient, but it presents a security risk. Normal service account passwords should expire in accordance with the organizations user password expiration policy. Create a reminder to change the service account's password two weeks before it will expire. Share the reminder with others that are allowed to change the password to ensure the password is changed before it expires.

Create the NDES Service User Rights Group Policy object

The Group Policy object ensures the NDES Service account has the proper user right to assign all the NDES servers in the **NDES Servers** group. As you add new NDES servers to your environment and this group, the service account automatically receives the proper user rights through the Group Policy.

Sign-in a domain controller or management workstations with *Domain Admin* equivalent credentials.

1. Start the **Group Policy Management Console** (gpmc.msc)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane.
3. Right-click **Group Policy object** and select **New**.
4. Type **NDES Service Rights** in the name box and select **OK**.
5. In the content pane, right-click the **NDES Service Rights** Group Policy object and select **Edit**.
6. In the navigation pane, expand **Policies** under **Computer Configuration**.

7. Expand **Windows Settings** > **Security Settings** > **Local Policies**. Select **User Rights Assignments**.

8. In the content pane, double-click **Allow log on locally**. Select **Define these policy settings** and select **OK**. Select **Add User or Group...**. In the **Add User or Group** dialog box, select **Browse**. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **Administrators;Backup Operators;DOMAINNAME\NDESSvc;Users** where **DOMAINNAME** is the NetBios name of the domain (Example CONTOSO\NDESSvc) in **User and group names**. Select **OK** twice.

9. In the content pane, double-click **Log on as a batch job**. Select **Define these policy settings** and select **OK**. Select **Add User or Group...**. In the **Add User or Group** dialog box, select **Browse**. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **Administrators;Backup Operators;DOMAINNAME\NDESSvc;Performance Log Users** where **DOMAINNAME** is the NetBios name of the domain (Example CONTOSO\NDESSvc) in **User and group names**. Select **OK** twice.

10. In the content pane, double-click **Log on as a service**. Select **Define these policy settings** and select **OK**. Select **Add User or Group...**. In the **Add User or Group** dialog box, select **Browse**. In the **Select Users, Computers, Service Accounts, or Groups** dialog box, type **NT SERVICE\ALL SERVICES;DOMAINNAME\NDESSvc** where **DOMAINNAME** is the NetBios name of the domain (Example CONTOSO\NDESSvc) in **User and group names**. Select **OK** three times.

11. Close the **Group Policy Management Editor**.

Configure security for the NDES Service User Rights Group Policy object

The best way to deploy the **NDES Service User Rights** Group Policy object is to use security group filtering. This enables you to easily manage the computers that receive the Group Policy settings by adding them to a group.

Sign-in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. Start the **Group Policy Management Console** (gpmc.msc).
2. Expand the domain and select the **Group Policy Object** node in the navigation pane.
3. Double-click the **NDES Service User Rights** Group Policy object.

4. In the **Security Filtering** section of the content pane, select **Add**. Type **NDES Servers** or the name of the security group you previously created and select **OK**.
5. Select the **Delegation** tab. Select **Authenticated Users** and select **Advanced**.
6. In the **Group or User names** list, select **Authenticated Users**. In the **Permissions for Authenticated Users** list, clear the **Allow** check box for the **Apply Group Policy** permission. Select **OK**

Deploy the NDES Service User Rights Group Policy object

The application of the **NDES Service User Rights** Group Policy object uses security group filtering. This enables you to link the Group Policy object at the domain, ensuring the Group Policy object is within scope to all computers. However, the security group filtering ensures only computers included in the **NDES Servers** global security group receive and apply the Group Policy object, which results in providing the **NDESSvc** service account with the proper user rights.

Sign-in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and right-click the node that has your Active Directory domain name and select **Link an existing GPO**
3. In the **Select GPO** dialog box, select **NDES Service User Rights** or the name of the Group Policy object you previously created and select **OK**

> ⓘ **Important**
>
> Linking the **NDES Service User Rights** Group Policy object to the domain ensures the Group Policy object is in scope for all computers. However, not all computers will have the policy settings applied to them. Only computers that are members of the **NDES Servers** global security group receive the policy settings. All others computers ignore the Group Policy object.

Prepare Active Directory Certificate Authority

You must prepare the public key infrastructure and the issuing certificate authority to support issuing certificates using Microsoft Intune and the Network Devices Enrollment Services (NDES) server role. In this task, you'll

- Configure the certificate authority to let Intune provide validity periods
- Create an NDES-Intune Authentication Certificate template

- Create a Microsoft Entra joined Windows Hello for Business authentication certificate template
- Publish certificate templates

Configure the certificate authority to let Intune provide validity periods

When deploying certificates using Microsoft Intune, you have the option of providing the validity period in the SCEP certificate profile rather than relying on the validity period in the certificate template. If you need to issue the same certificate with different validity periods, it may be advantageous to use the SCEP profile, given the limited number of certificates a single NDES server can issue.

> ⓘ **Note**
>
> Skip this step if you do not want to enable Microsoft Intune to specify the validity period of the certificate. Without this configuration, the certificate request uses the validity period configured in the certificate template.

Sign-in to the issuing certificate authority with access equivalent to *local administrator*.

1. Open an elevated command prompt and type the following command:

   Windows Command Prompt

   ```
   certutil -setreg Policy\EditFlags +EDITF_ATTRIBUTEENDDATE
   ```

2. Restart the **Active Directory Certificate Services** service

Create an NDES-Intune authentication certificate template

NDES uses a server authentication certificate to authenticate the server endpoint, which encrypts the communication between it and the connecting client. The Intune Certificate Connector uses a client authentication certificate template to authenticate to the certificate registration point.

Sign-in to the issuing certificate authority or management workstations with *Domain Admin* equivalent credentials.

1. Open the **Certificate Authority** management console.

2. Right-click **Certificate Templates** and select **Manage**.
3. In the **Certificate Template Console**, right-click the **Computer** template in the details pane and select **Duplicate Template**.
4. On the **General** tab, type **NDES-Intune Authentication** in **Template display name**. Adjust the validity and renewal period to meet your enterprise's needs.

   > ⓘ **Note**
   >
   > If you use different template names, you'll need to remember and substitute these names in different portions of the lab.

5. On the **Subject** tab, select **Supply in the request**.
6. On the **Cryptography** tab, validate the **Minimum key size** is **2048**.
7. On the **Security** tab, select **Add**.
8. Select **Object Types**, then in the window that appears, choose **Computers** and select **OK**.
9. Type **NDES server** in the **Enter the object names to select** text box and select **OK**.
10. Select **NDES server** from the **Group or users names** list. In the **Permissions for** section, select the **Allow** check box for the **Enroll** permission. Clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other items in the **Group or users names** list if the check boxes aren't already cleared. Select **OK**.
11. Select on the **Apply** to save changes and close the console.

Create a Microsoft Entra joined Windows Hello for Business authentication certificate template

During Windows Hello for Business provisioning, Windows requests an authentication certificate from Microsoft Intune, which requests the authentication certificate on behalf of the user. This task configures the Windows Hello for Business authentication certificate template. You use the name of the certificate template when configuring the NDES Server.

Sign in a certificate authority or management workstations with *Domain Admin equivalent* credentials.

1. Open the **Certificate Authority** management console.

2. Right-click **Certificate Templates** and select **Manage**.
3. Right-click the **Smartcard Logon** template and choose **Duplicate Template**.
4. On the **Compatibility** tab, clear the **Show resulting changes** check box. Select **Windows Server 2012** or **Windows Server 2012 R2** from the **Certification Authority** list. Select **Windows Server 2012** or **Windows Server 2012 R2** from the **Certificate Recipient** list.
5. On the **General** tab, type **ENTRA JOINED WHFB Authentication** in **Template display name**. Adjust the validity and renewal period to meet your enterprise's needs.

   > ⓘ **Note**
   >
   > If you use different template names, you'll need to remember and substitute these names in different portions of the deployment

6. On the **Cryptography** tab, select **Key Storage Provider** from the **Provider Category** list. Select **RSA** from the **Algorithm name** list. Type **2048** in the **Minimum key size** text box. Select **SHA256** from the **Request hash** list.
7. On the **Extensions** tab, verify the **Application Policies** extension includes **Smart Card Logon**.
8. On the **Subject** tab, select **Supply in the request**.
9. On the **Request Handling** tab, select **Signature and encryption** from the **Purpose** list. Select the **Renew with same key** check box. Select **Enroll subject without requiring any user input**.
10. On the **Security** tab, select **Add**. Type **NDESSvc** in the **Enter the object names to select** text box and select **OK**.
11. Select **NDESSvc** from the **Group or users names** list. In the **Permissions for NDES Servers** section, select the **Allow** check box for **Read** and **Enroll**. Clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other entries in the **Group or users names** section if the check boxes aren't already cleared. Select **OK**.
12. Close the console.

Publish certificate templates

The certificate authority may only issue certificates for certificate templates that are published to that certificate authority. If you have more than one certificate authority and you want that certificate authority to issue certificates based on a specific certificate template, then you must publish the certificate template to all certificate authorities that are expected to issue the certificate.

> **ⓘ Important**
>
> Ensure you publish the **ENTRA JOINED WHFB Authentication** certificate templates to the certificate authority that Microsoft Intune uses by way of the NDES servers. The NDES configuration asks you to choose a certificate authority from which it requests certificates. You need to publish that certificate templates to that issuing certificate authority. The **NDES-Intune Authentication** certificate is directly enrolled and can be published to any certificate authority.

Sign in to the certificate authority or management workstations with an *enterprise admin* -equivalent credential.

1. Open the **Certificate Authority** management console
2. Expand the parent node from the navigation pane
3. Select **Certificate Templates** in the navigation pane
4. Right-click the **Certificate Templates** node. Select **New**, and select **Certificate Template** to issue
5. In the **Enable Certificates Templates** window, select the **NDES-Intune Authentication** and **ENTRA JOINED WHFB Authentication** templates you created in the previous steps. Select **OK** to publish the selected certificate templates to the certificate authority
6. Close the console

Install and Configure the NDES Role

This section includes the following articles:

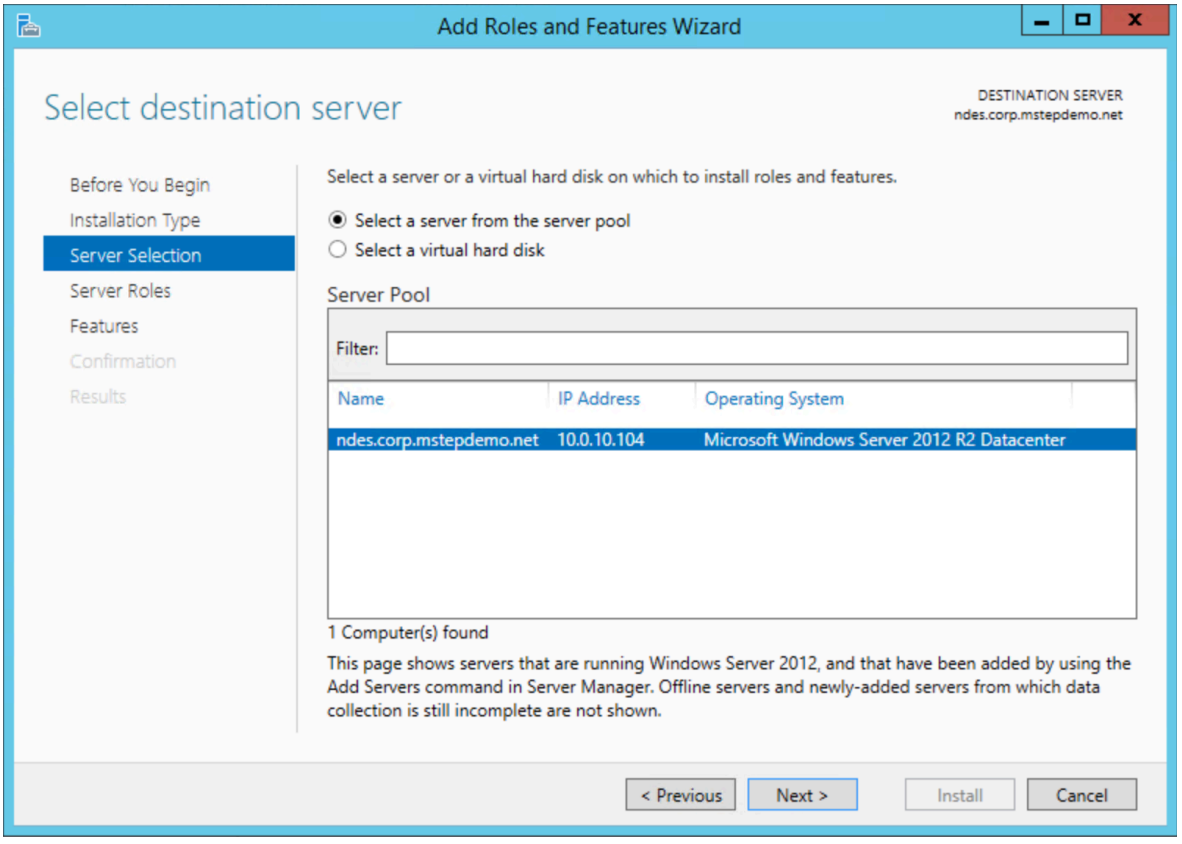
- Install the Network Device Enrollment Service Role
- Configure the NDES service account
- Configure the NDES role and Certificate Templates
- Create a Web Application Proxy for the Internal NDES URL
- Enroll for an NDES-Intune Authentication Certificate
- Configure the Web Server Certificate for NDES
- Verify the configuration

Install the Network Device Enrollment Services Role

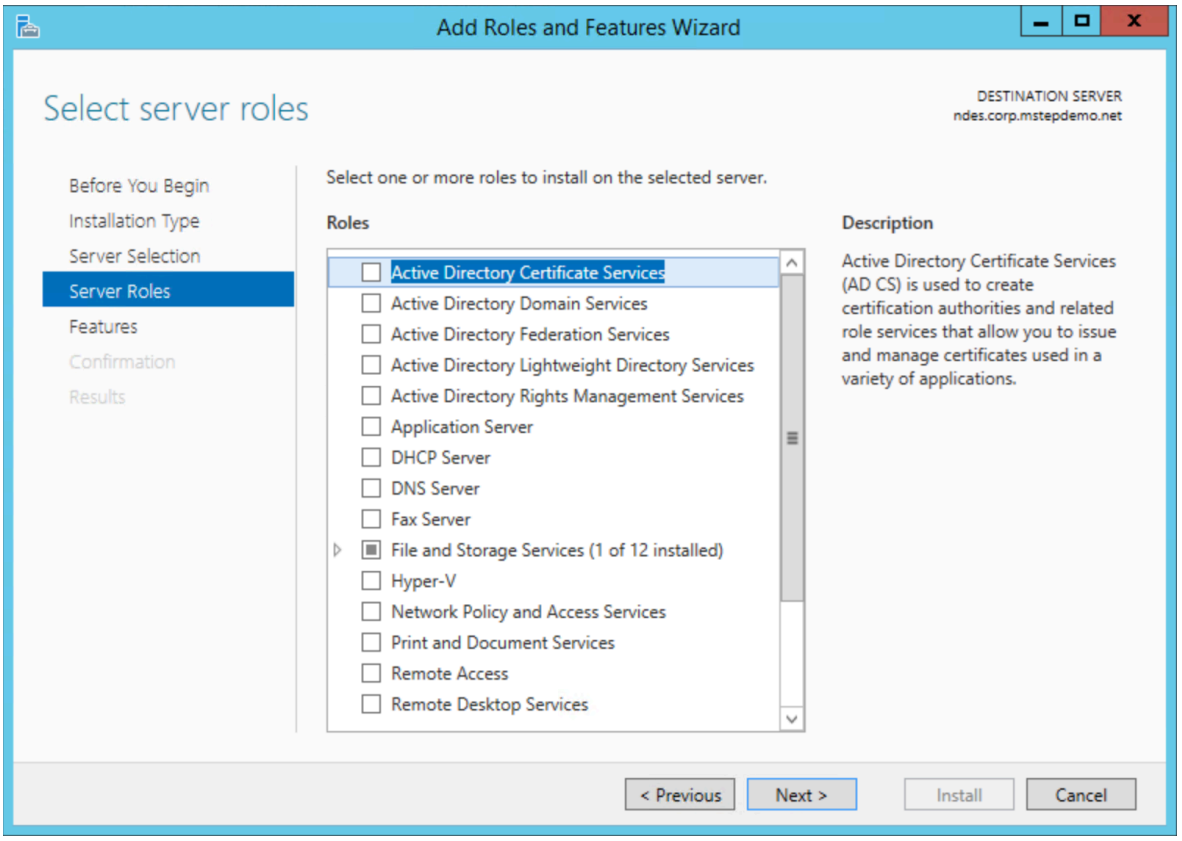
Install the Network Device Enrollment Service role on a computer other than the issuing certificate authority.

Sign-in to the certificate authority or management workstations with an *Enterprise Admin* equivalent credential.

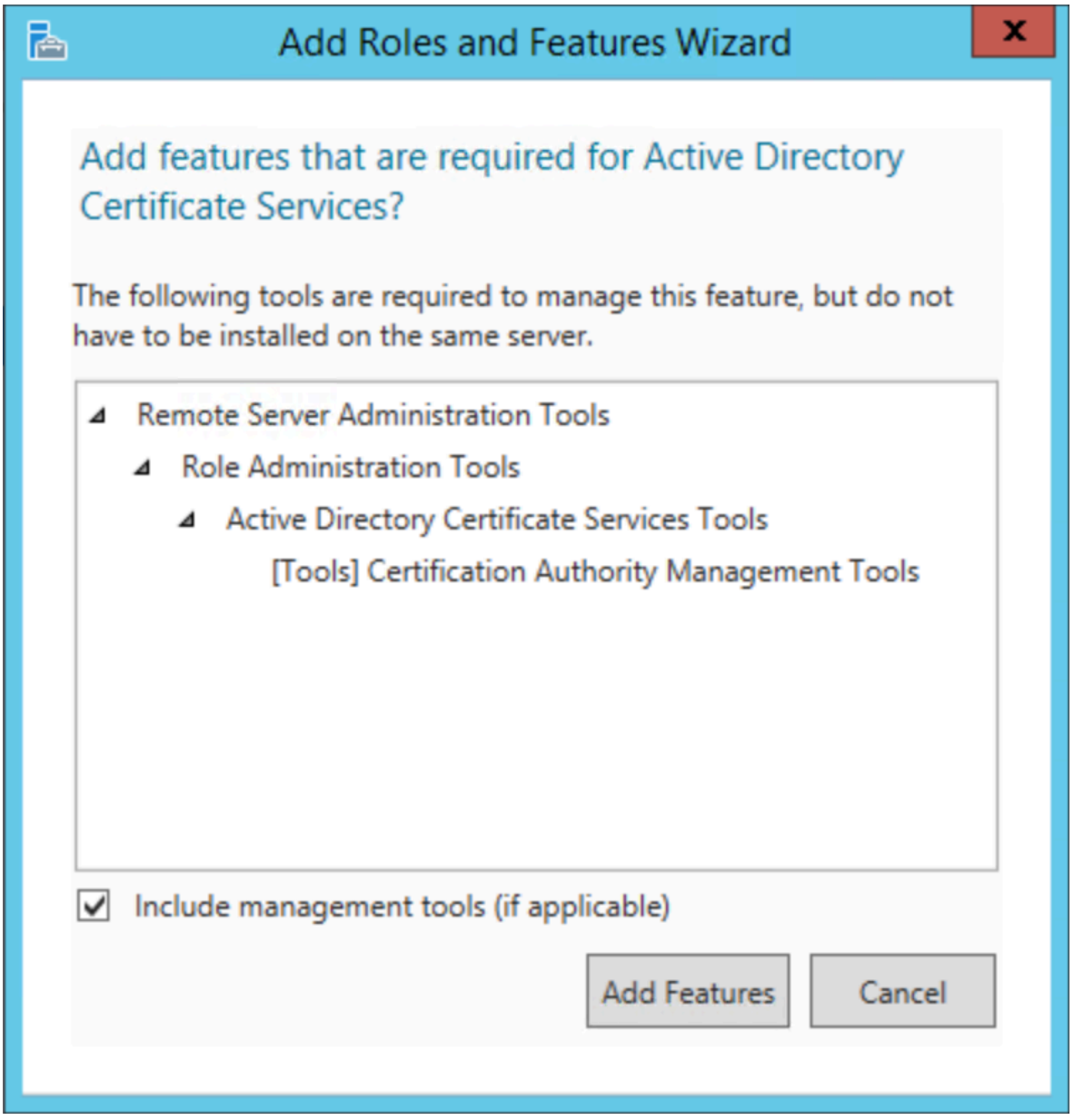
1. Open **Server Manager** on the NDES server
2. Select **Manage**. Select **Add Roles and Features**
3. In the **Add Roles and Features Wizard**, on the **Before you begin** page, select **Next**. Select **Role-based or feature-based installation** on the **Select installation type** page. Select **Next**. Select **Select a server from the server pool**. Select the local server from the **Server Pool** list. Select **Next**



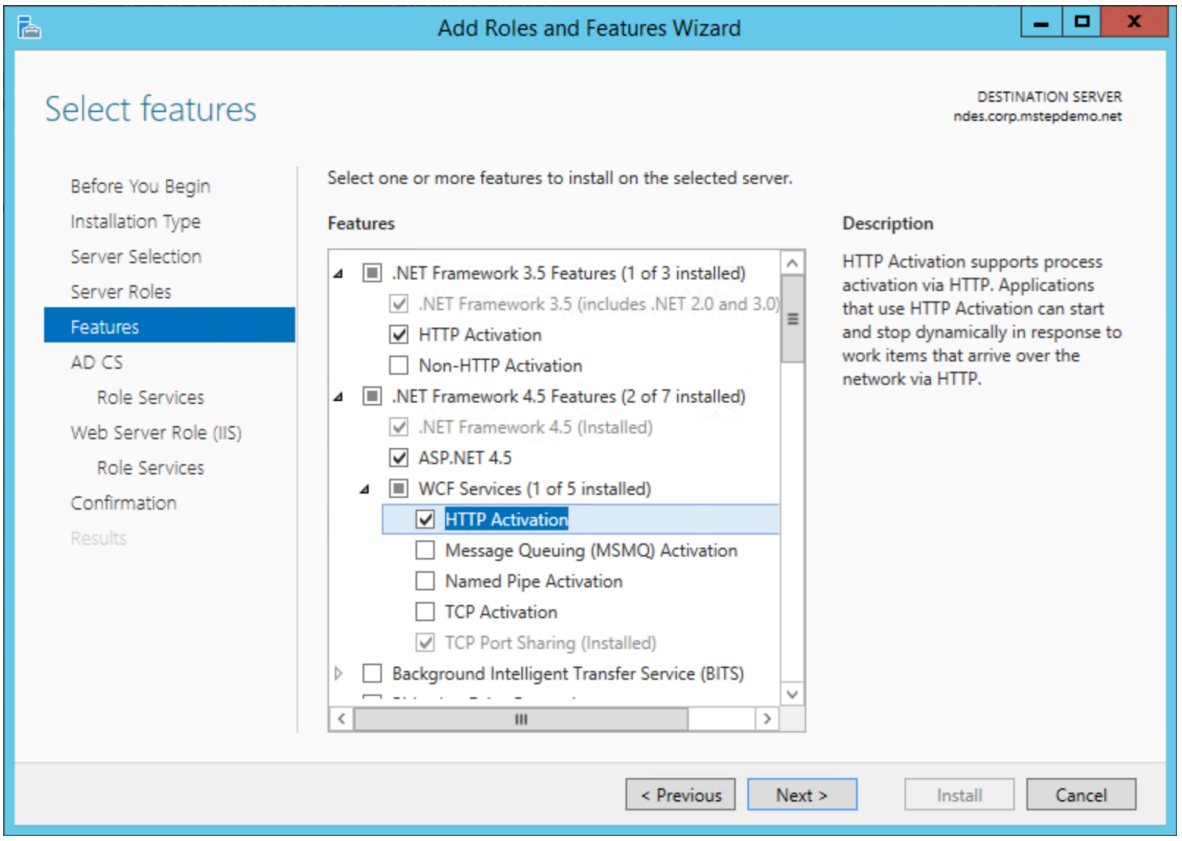
4. On the **Select server roles** page, select **Active Directory Certificate Services** from the **Roles** list



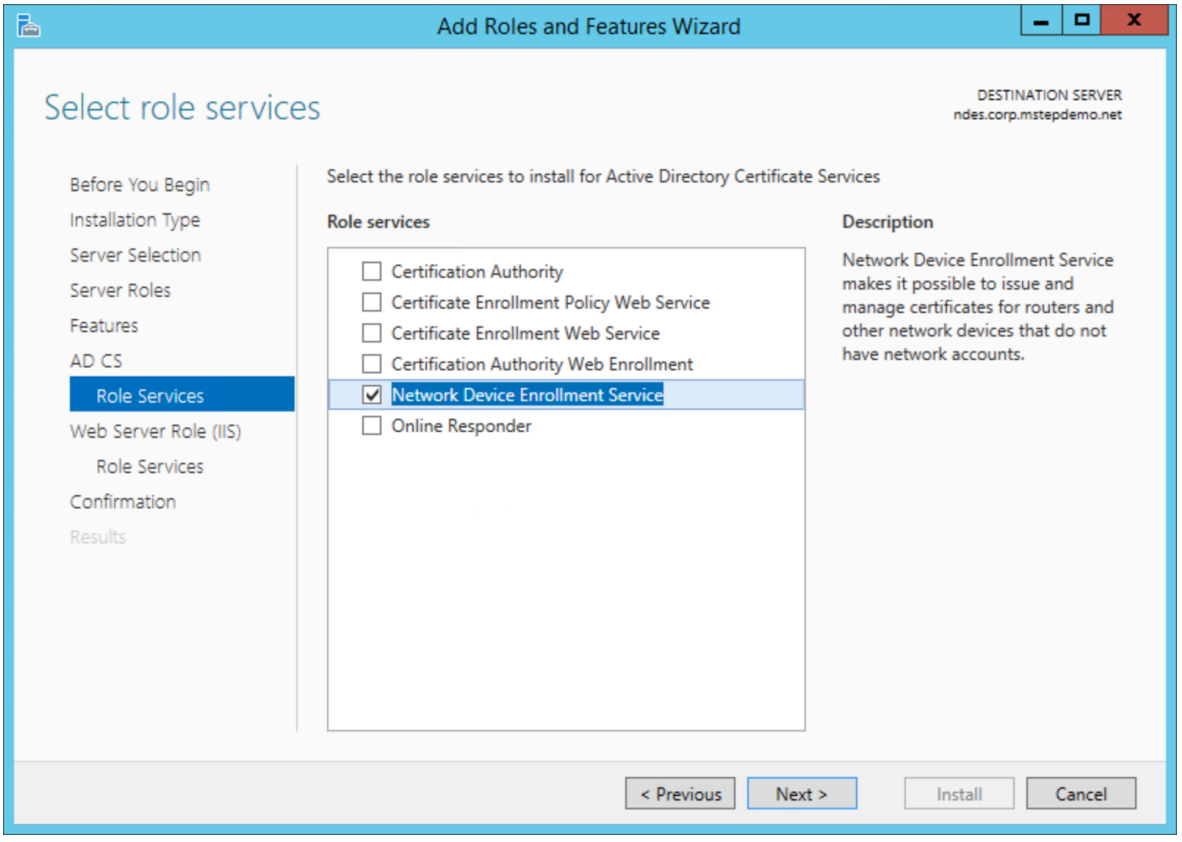
Select **Add Features** on the **Add Roles and Feature Wizard** dialog box. Select **Next**



5. On the **Features** page, expand **.NET Framework 3.5 Features**. Select **HTTP Activation**. Select **Add Features** on the **Add Roles and Feature Wizard** dialog box. Expand **.NET Framework 4.5 Features**. Expand **WCF Services**. Select **HTTP Activation**. Select **Add Features** on the **Add Roles and Feature Wizard** dialog box. Select **Next**



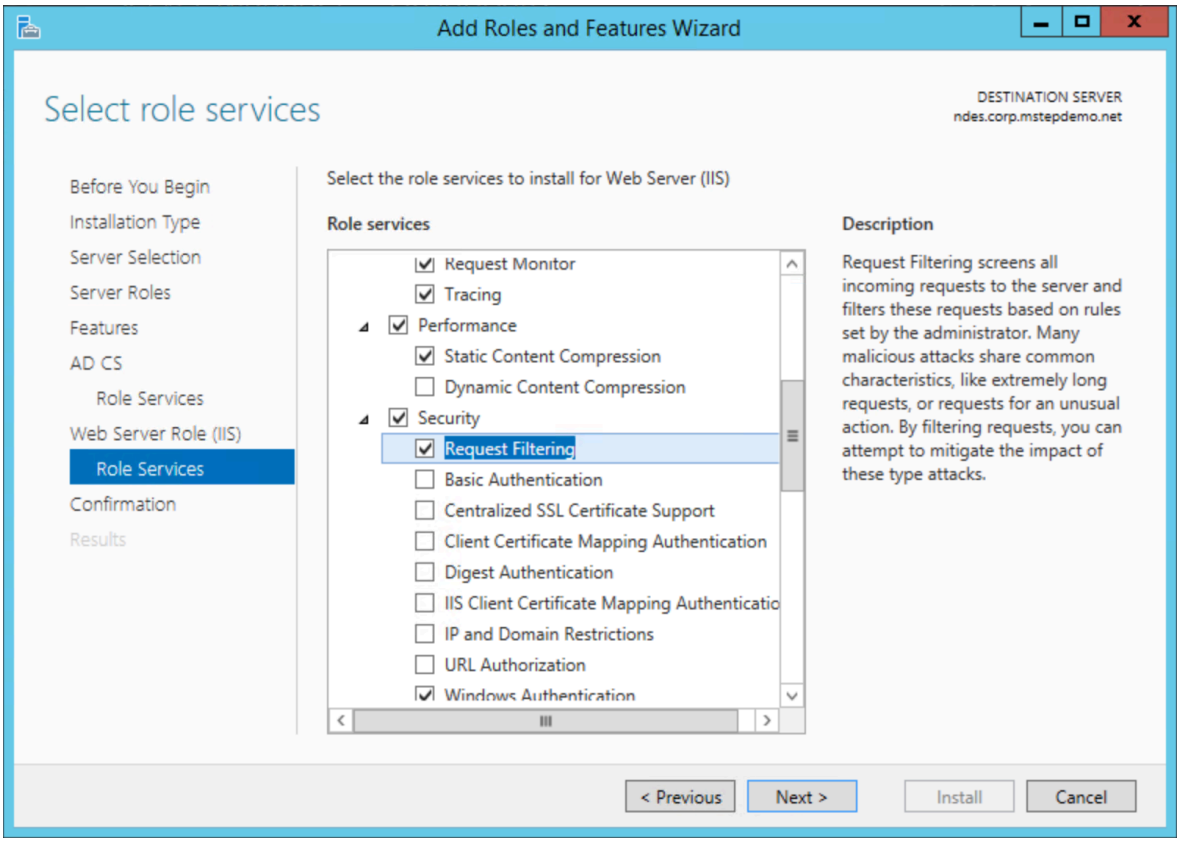
6. On the **Select role services** page, clear the **Certificate Authority** check box. Select the **Network Device Enrollment Service**. Select **Add Features** on the **Add Roles and Features Wizard** dialog box. Select **Next**



7. Select **Next** on the **Web Server Role (IIS)** page

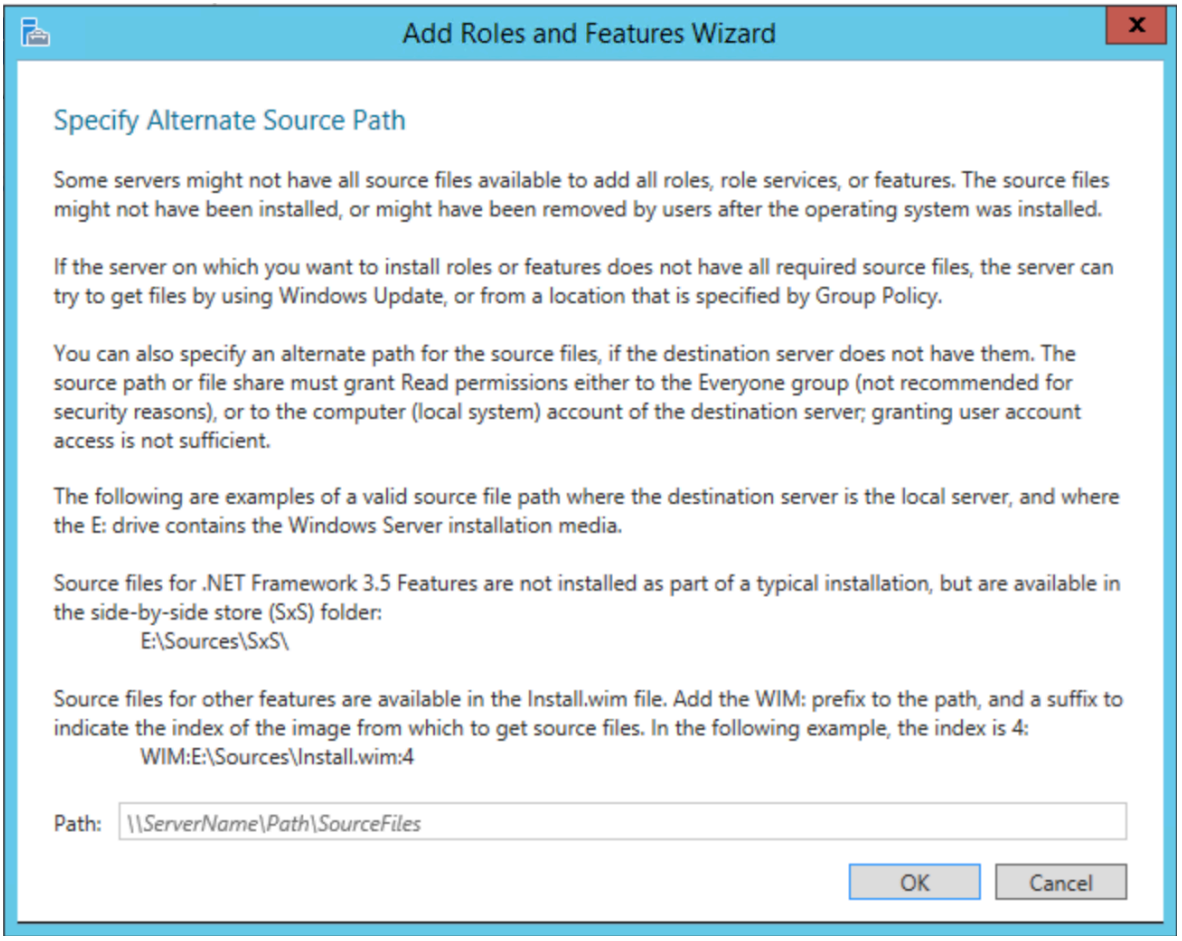
8. On the **Select role services** page for the Web Serve role, Select the following additional services if they aren't already selected and then select **Next**

    - **Web Server > Security > Request Filtering**
    - **Web Server > Application Development > ASP.NET 3.5**
    - **Web Server > Application Development > ASP.NET 4.5**
    - **Management Tools > IIS 6 Management Compatibility > IIS 6 Metabase Compatibility**
    - **Management Tools > IIS 6 Management Compatibility > IIS 6 WMI Compatibility**



9. Select **Install**. When the installation completes, continue with the next procedure. **Do not click Close**

> **ⓘ Important**
>
> .NET Framework 3.5 is not included in the typical installation. If the server is connected to the Internet, the installation attempts to get the files using Windows Update. If the server is not connected to the Internet, you need to **Specify an alternate source path** such as <driveLetter>:\Sources\SxS\



Configure the NDES service account

This task adds the NDES service account to the local IIS_USRS group. The task also configures the NDES service account for Kerberos authentication and delegation

Add the NDES service account to the IIS_USRS group

Sign-in the NDES server with access equivalent to *local administrator*.

1. Start the **Local Users and Groups** management console (`lusrmgr.msc`)
2. Select **Groups** from the navigation pane. Double-click the IIS_IUSRS group
3. In the **IIS_IUSRS Properties** dialog box, select **Add**. Type **NDESSvc** or the name of your NDES service account. Select **Check Names** to verify the name and then select **OK**. Select **OK** to close the properties dialog box
4. Close the management console.

Register a Service Principal Name on the NDES Service account

Sign-in the NDES server with access equivalent to *Domain Admins*.

1. Open an elevated command prompt

2. Type the following command to register the service principal name

```
Windows Command Prompt

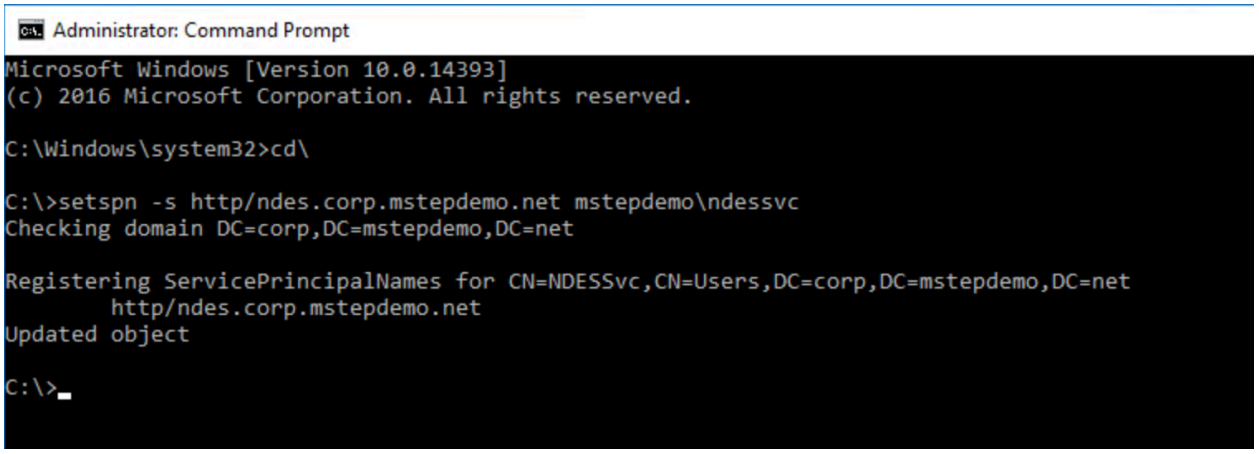
setspn -s http/[FqdnOfNdesServer] [DomainName\\NdesServiceAccount]
```

where **[FqdnOfNdesServer]** is the fully qualified domain name of the NDES server and **[DomainName\NdesServiceAccount]** is the domain name and NDES service account name separated by a backslash (\). An example of the command looks like the following:

```
Windows Command Prompt

setspn -s http/ndes.corp.contoso.com contoso\ndessvc
```

> **Note**
>
> If you use the same service account for multiple NDES Servers, repeat the following task for each NDES server under which the NDES service runs.

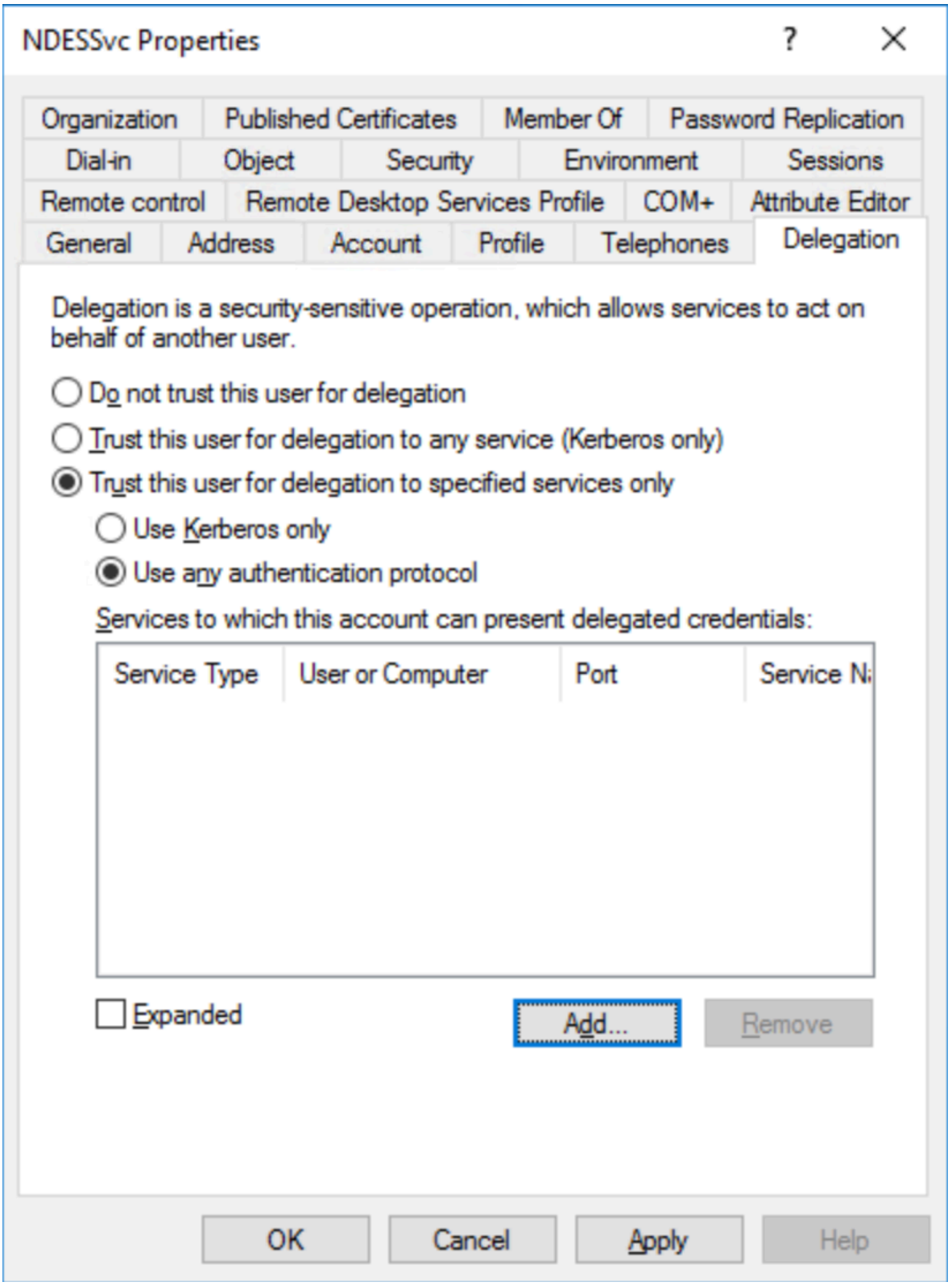


Configure the NDES Service account for delegation

The NDES service enrolls certificates on behalf of users. Therefore, you want to limit the actions it can perform on behalf of the user. You do this through delegation.

Sign-in a domain controller with a minimum access equivalent to *Domain Admins*.

1. Open **Active Directory Users and Computers**
2. Locate the NDES Service account (NDESSvc). Right-click and select **Properties**. Select the **Delegation** tab.

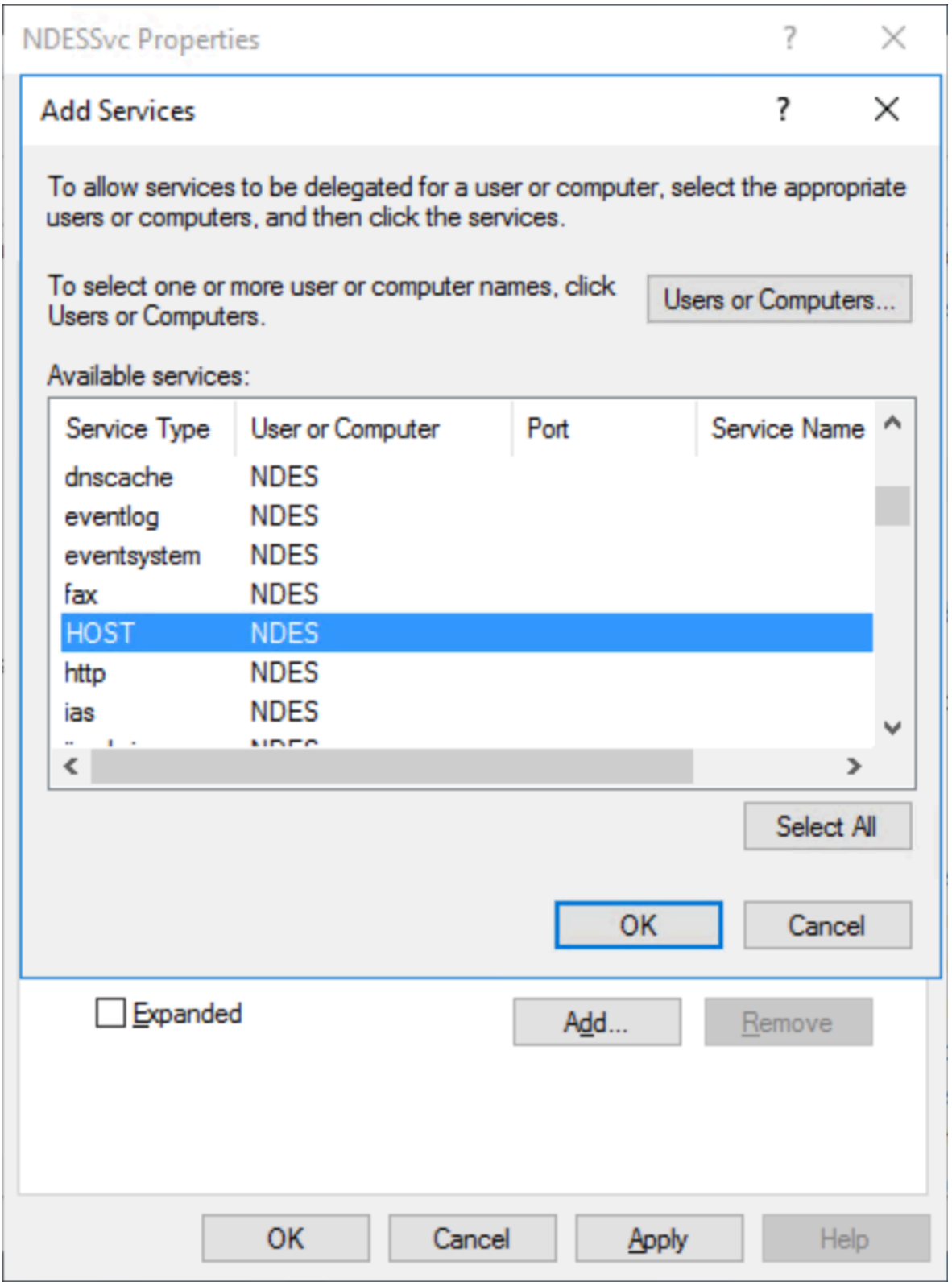


3. Select **Trust this user for delegation to specified services only**

4. Select **Use any authentication protocol**

5. Select **Add**

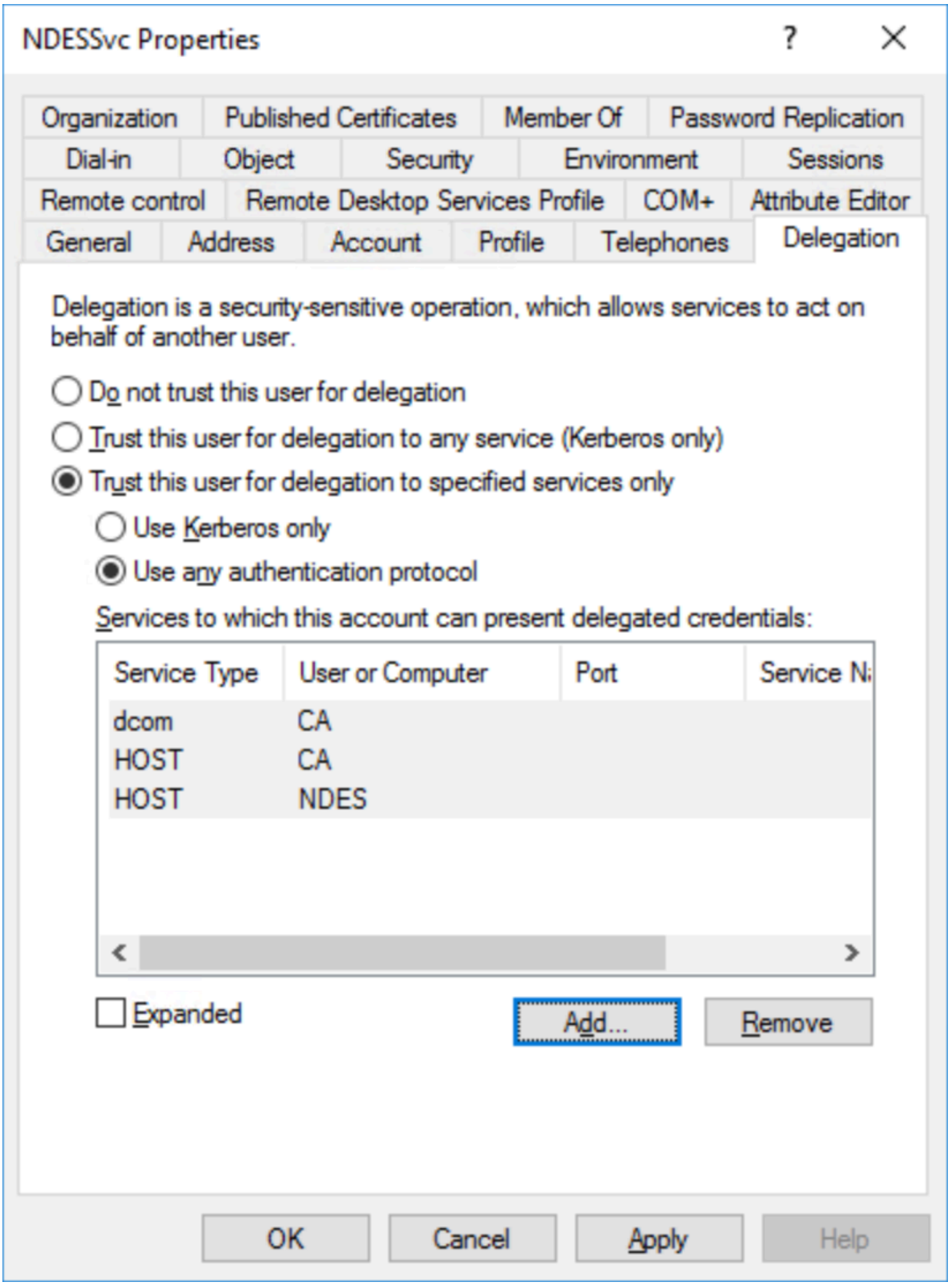
6. Select **Users or Computers...** Type the name of the *NDES Server* you use to issue Windows Hello for Business authentication certificates to Microsoft Entra joined devices. From the **Available services** list, select **HOST**. Select **OK**



7. Repeat steps 5 and 6 for each NDES server using this service account. Select **Add**

8. Select **Users or computers...** Type the name of the issuing certificate authority this NDES service account uses to issue Windows Hello for Business authentication certificates to Microsoft Entra joined devices. From the **Available services** list, select **dcom**. Hold the **CTRL** key and select **HOST**. Select **OK**

9. Repeat steps 8 and 9 for each issuing certificate authority from which one or more NDES servers request certificates



10. Select **OK**. Close **Active Directory Users and Computers**

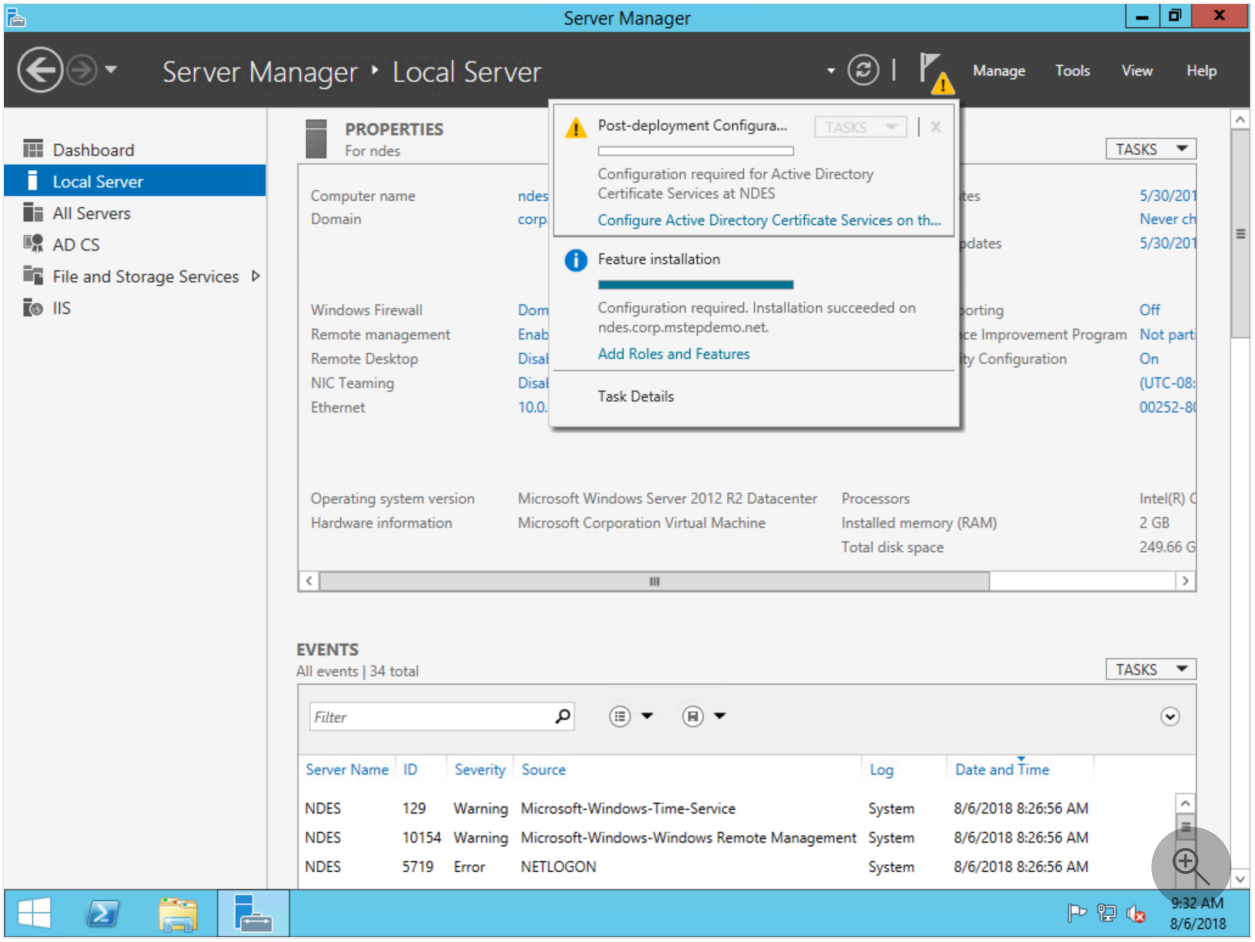
Configure the NDES Role and Certificate Templates

This task configures the NDES role and the certificate templates the NDES server issues.

Configure the NDES Role

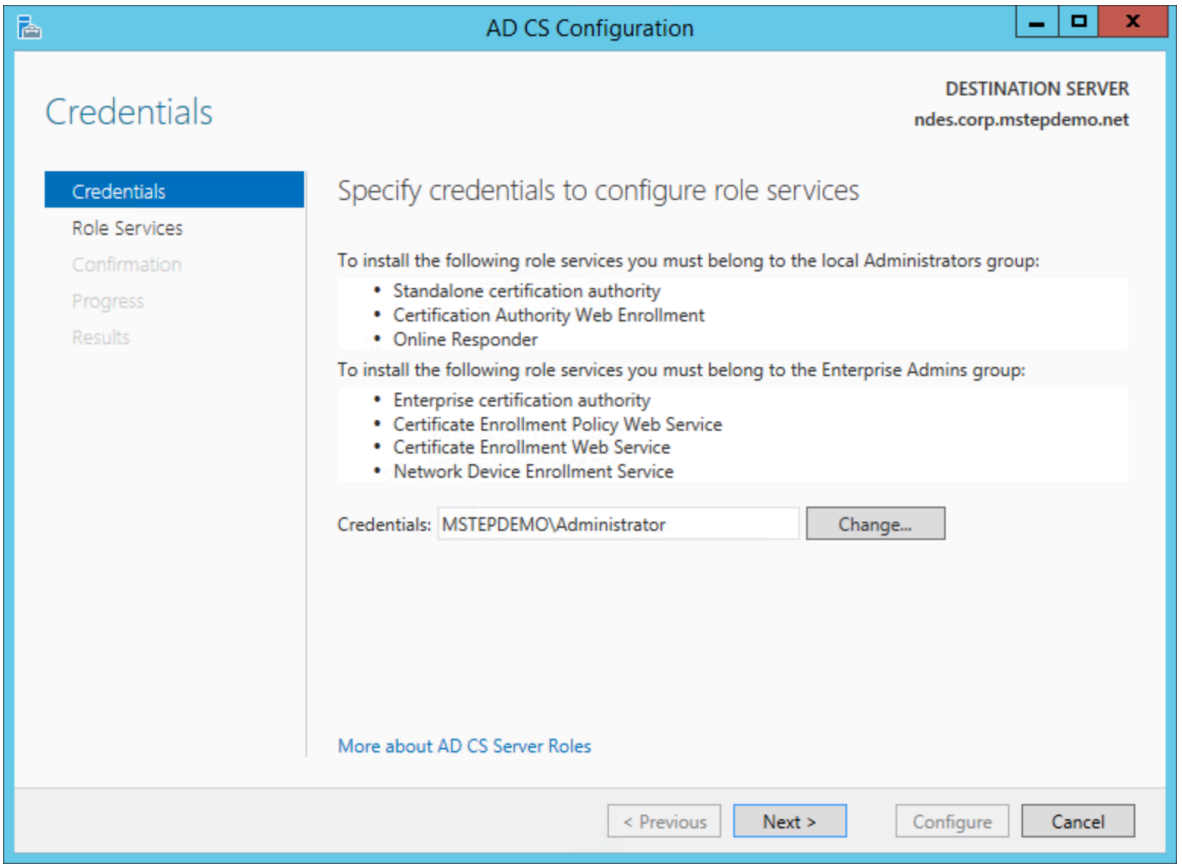
Sign-in to the certificate authority or management workstations with an *Enterprise Admin* equivalent credential.

> **Note**
>
> If you closed Server Manger from the last set of tasks, start Server Manager and click the action flag that shows a yellow exclamation point.

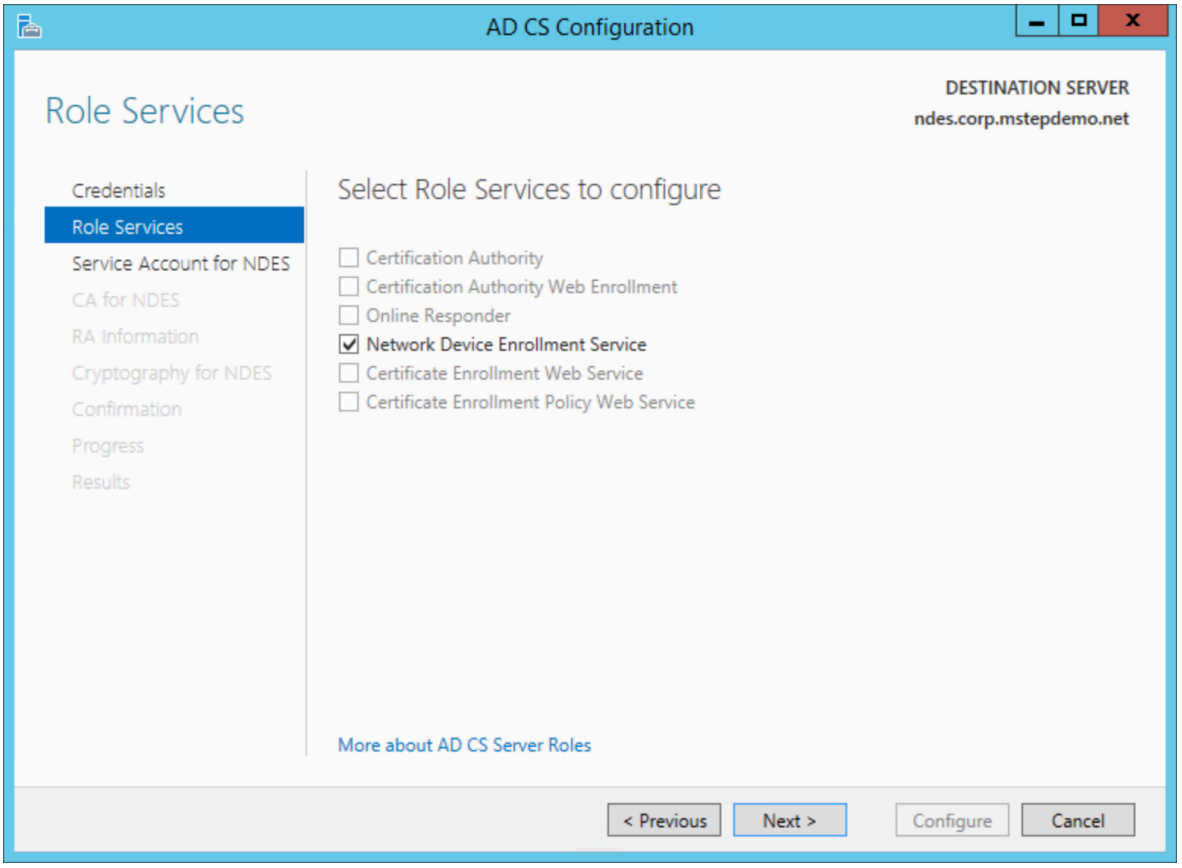


1. Select the **Configure Active Directory Certificate Services on the destination server** link

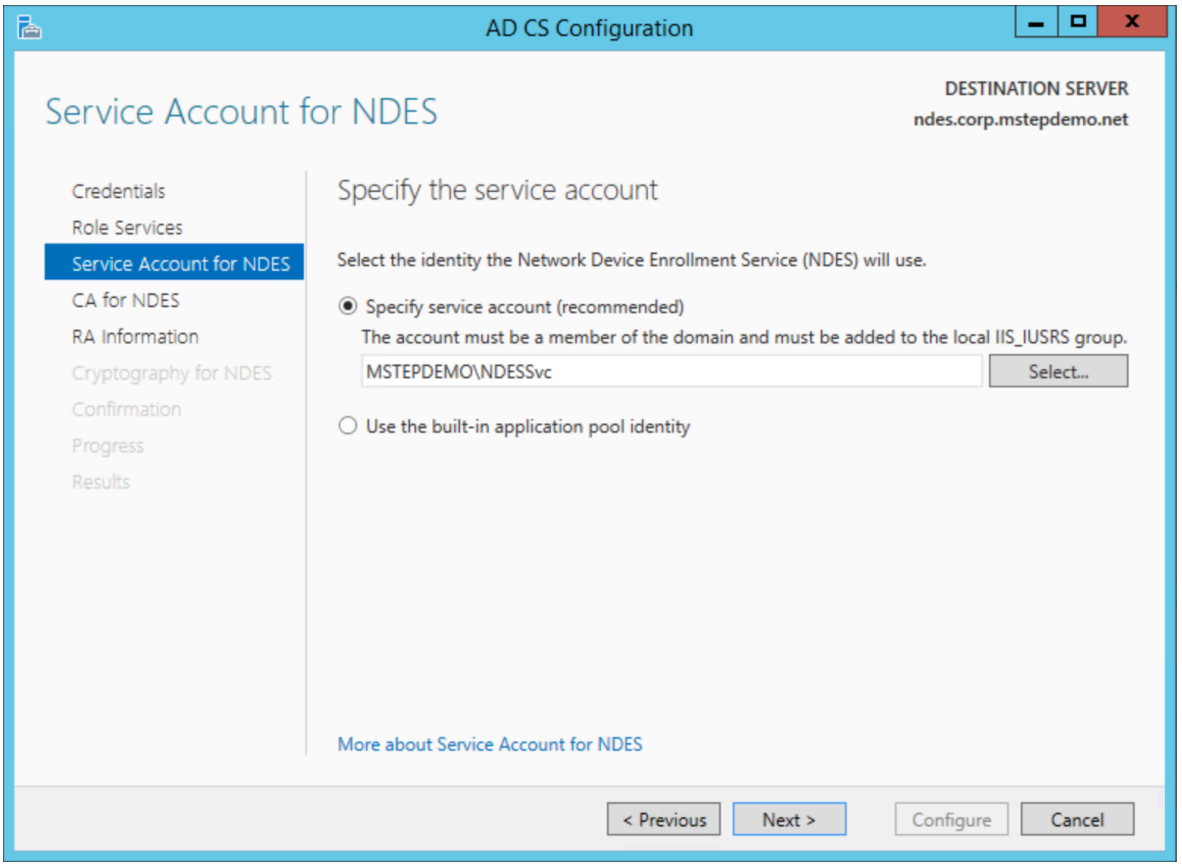
2. On the **Credentials** page, select **Next**.



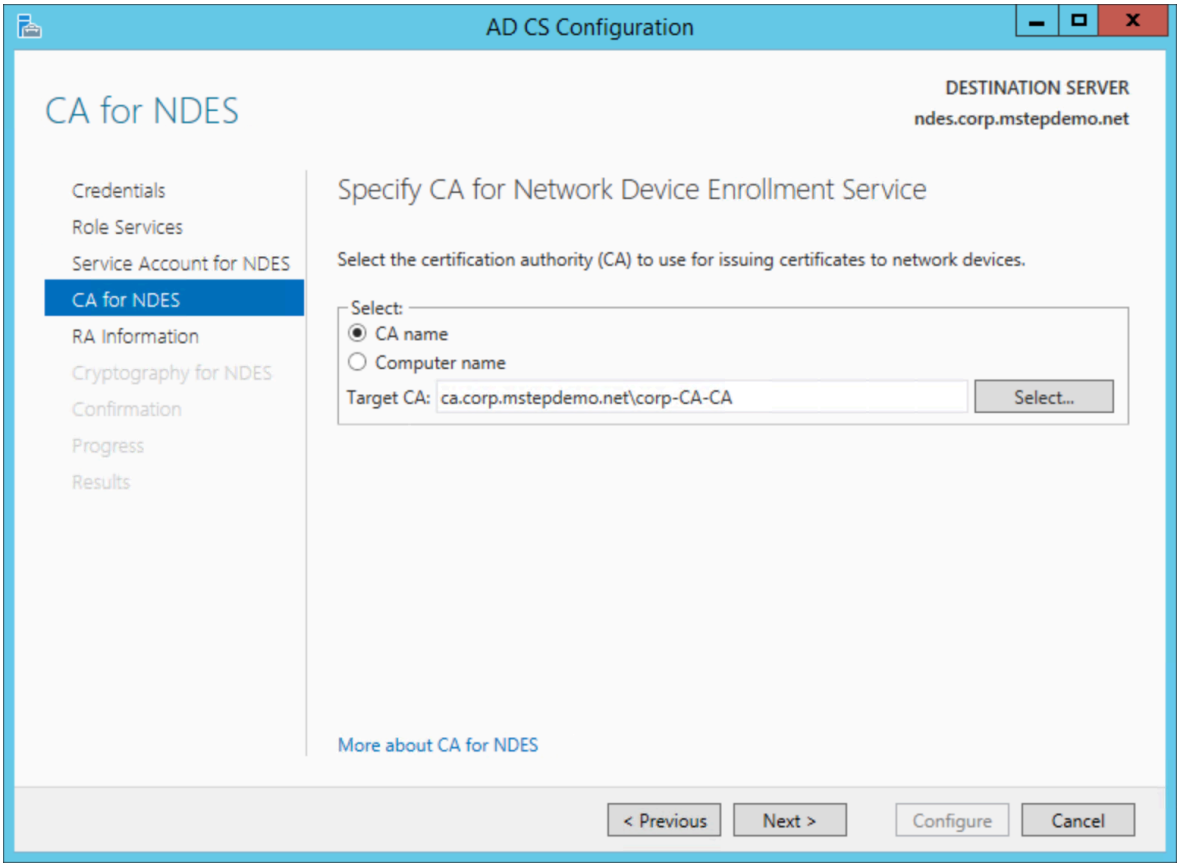
3. On the **Role Services** page, select **Network Device Enrollment Service** and then select **Next**



4. On the **Service Account for NDES** page, select **Specify service account (recommended)**. Select **Select...**. Type the user name and password for the NDES service account in the **Windows Security** dialog box. Select **Next**



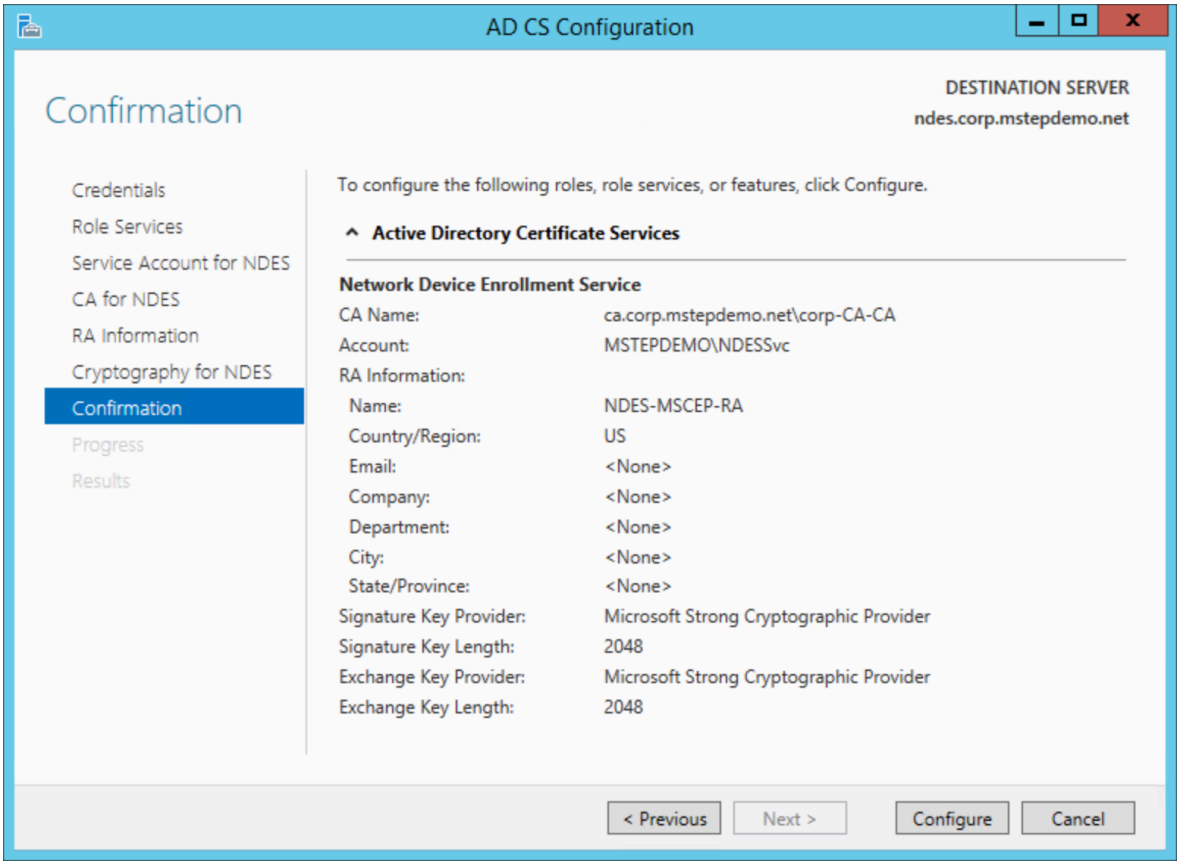
5. On the **CA for NDES** page, select **CA name**. Select **Select...**. Select the issuing certificate authority from which the NDES server requests certificates. Select **Next**



6. On the **RA Information**, select **Next**

7. On the **Cryptography for NDES** page, select **Next**

8. Review the **Confirmation** page. Select **Configure**



9. Select **Close** after the configuration completes

Configure Certificate Templates on NDES

A single NDES server can request a maximum of three certificate templates. The NDES server determines which certificate to issue based on the incoming certificate request that is assigned in the Microsoft Intune SCEP certificate profile. The Microsoft Intune SCEP certificate profile has three values.

- Digital Signature
- Key Encipherment
- Key Encipherment, Digital Signature

Each value maps to a registry value name in the NDES server. The NDES server translates an incoming SCEP provided value into the corresponding certificate template. The table below shows the SCEP profile values of the NDES certificate template registry value names.

Expand table

SCEP Profile Key usage	NDES Registry Value Name
Digital Signature	SignatureTemplate
Key Encipherment	EncryptionTemplate
Key Encipherment Digital Signature	GeneralPurposeTemplate

Ideally, you should match the certificate request with the registry value name to keep the configuration intuitive (encryption certificates use the encryption template, signature certificates use the signature template, etc.). A result of this intuitive design is the potential exponential growth in the NDES server. Imagine an organization that needs to issue nine unique signature certificates across their enterprise.

If the need arises, you can configure a signature certificate in the encryption registry value name or an encryption certificate in the signature registry value to maximize the use of your NDES infrastructure. This unintuitive design requires current and accurate documentation of the configuration to ensure the SCEP certificate profile is configured to enroll the correct certificate, regardless of the actual purpose. Each organization needs to balance ease of configuration and administration with additional NDES infrastructure and the management overhead that comes with it.

Sign-in to the NDES Server with *local administrator* equivalent credentials.

1. Open an elevated command prompt
2. Using the table above, decide which registry value name you'll use to request Windows Hello for Business authentication certificates for Microsoft Entra joined devices
3. Type the following command:

   Windows Command Prompt

   ```
   reg add HKLM\Software\Microsoft\Cryptography\MSCEP /v
   [registryValueName] /t REG_SZ /d [certificateTemplateName]
   ```

   where **registryValueName** is one of the three value names from the above table and where **certificateTemplateName** is the name of the certificate template you created for Windows Hello for Business Microsoft Entra joined devices. Example:

   Windows Command Prompt

   ```
   reg add HKLM\Software\Microsoft\Cryptography\MSCEP /v SignatureTemplate
   ```

```
/t REG_SZ /d ENTRAJOINEDWHFBAuthentication
```

4. Type **Y** when the command asks for permission to overwrite the existing value
5. Close the command prompt.

> **ⓘ Important**
>
> Use the **name** of the certificate template; not the **display name**. The certificate template name does not include spaces. You can view the certificate names by looking at the **General** tab of the certificate template's properties in the **Certificates Templates** management console (`certtmpl.msc`).

Create a Web Application Proxy for the internal NDES URL.

Certificate enrollment for Microsoft Entra joined devices occurs over the Internet. As a result, the internal NDES URLs must be accessible externally. You can do this easily and securely using Microsoft Entra application proxy. Microsoft Entra application proxy provides single sign-on and secure remote access for web applications hosted on-premises, such as Network Device Enrollment Services.

Ideally, you configure your Microsoft Intune SCEP certificate profile to use multiple external NDES URLs. This enables Microsoft Intune to round-robin load balance the certificate requests to identically configured NDES Servers (each NDES server can accommodate approximately 300 concurrent requests). Microsoft Intune sends these requests to Microsoft Entra Application Proxies.

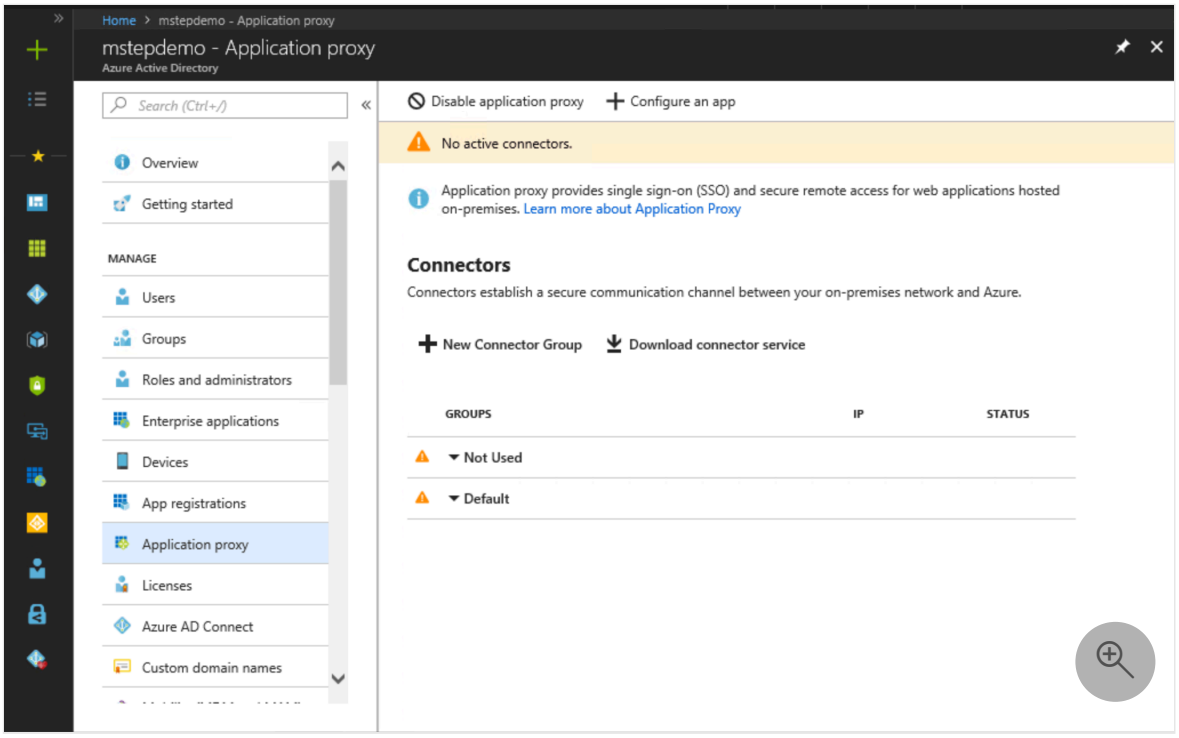
Microsoft Entra Application proxies are serviced by lightweight Application Proxy Connector agents. See What is Application Proxy for more details. These agents are installed on your on-premises, domain joined devices and make authenticated secure outbound connection to Azure, waiting to process requests from Microsoft Entra Application Proxies. You can create connector groups in Microsoft Entra ID to assign specific connectors to service specific applications.

Connector group automatically round-robin, load balance the Microsoft Entra application proxy requests to the connectors within the assigned connector group. This ensures Windows Hello for Business certificate requests have multiple dedicated Microsoft Entra application proxy connectors exclusively available to satisfy enrollment requests. Load balancing the NDES servers and connectors should ensure users enroll their Windows Hello for Business certificates in a timely manner.

Download and Install the Application Proxy Connector Agent

Sign-in a workstation with access equivalent to a *domain user*.

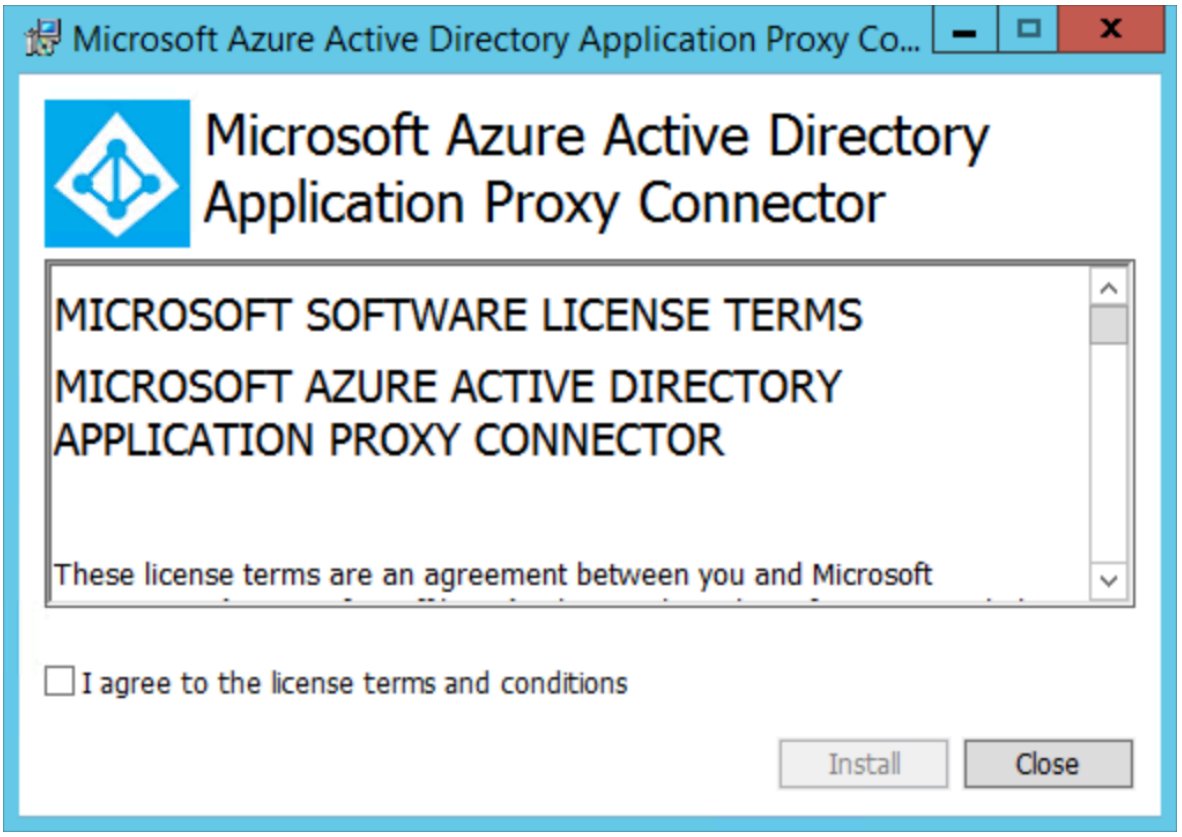
1. Access to the Microsoft Entra admin center as at least an Application Administrator
2. Select **All Services**. Type **Microsoft Entra ID** to filter the list of services. Under **SERVICES**, select **Microsoft Entra ID**
3. Under **MANAGE**, select **Application proxy**
4. Select **Download connector service**. Select **Accept terms & Download**. Save the file (AADApplicationProxyConnectorInstaller.exe) in a location accessible by others on the domain



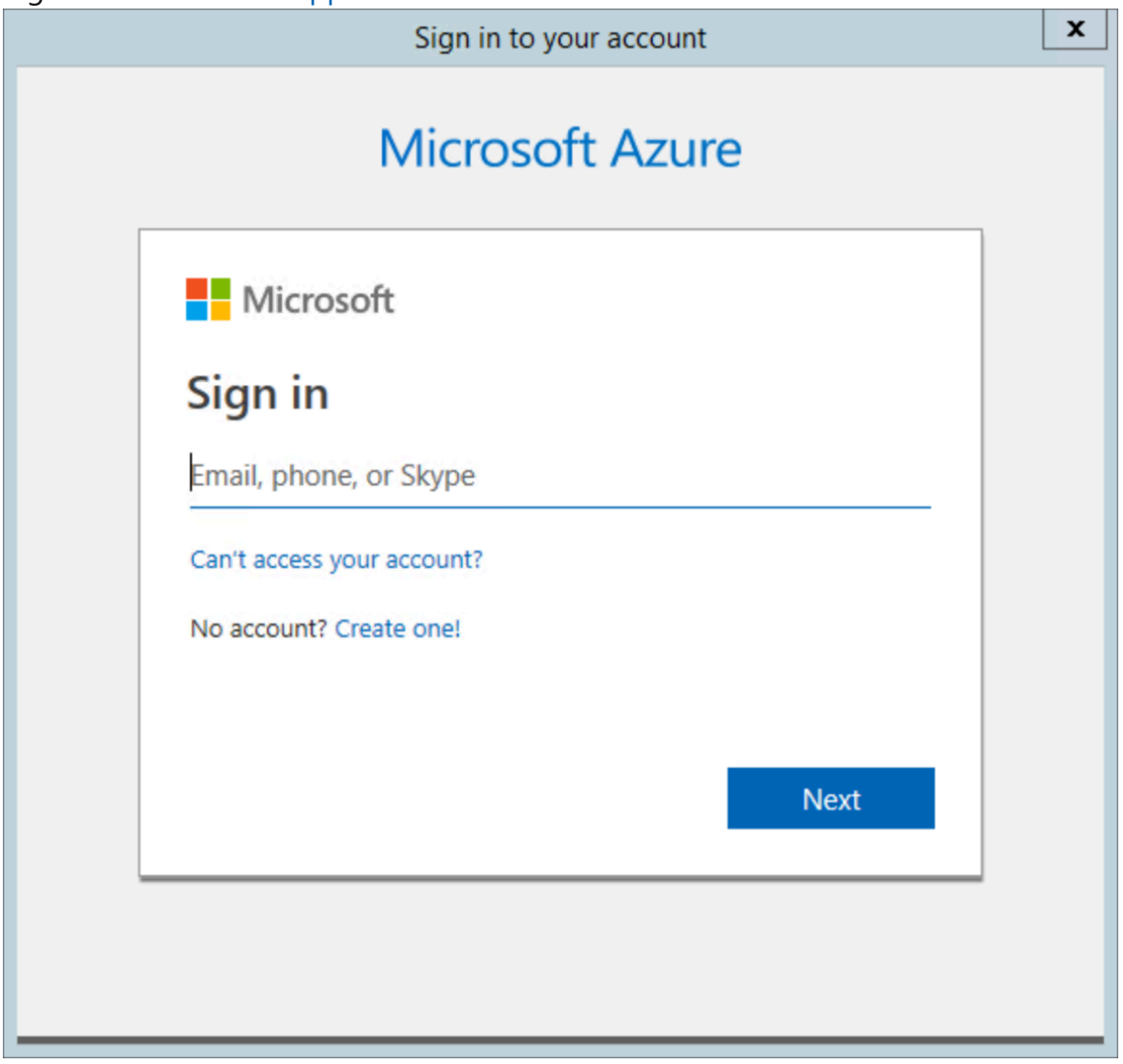
5. Sign-in the computer that will run the connector with access equivalent to a *domain user*

> **ⓘ Important**
>
> Install a minimum of two Microsoft Entra ID Proxy connectors for each NDES Application Proxy. Strategically locate Microsoft Entra application proxy connectors throughout your organization to ensure maximum availability. Remember, devices running the connector must be able to communicate with Azure and the on-premises NDES servers

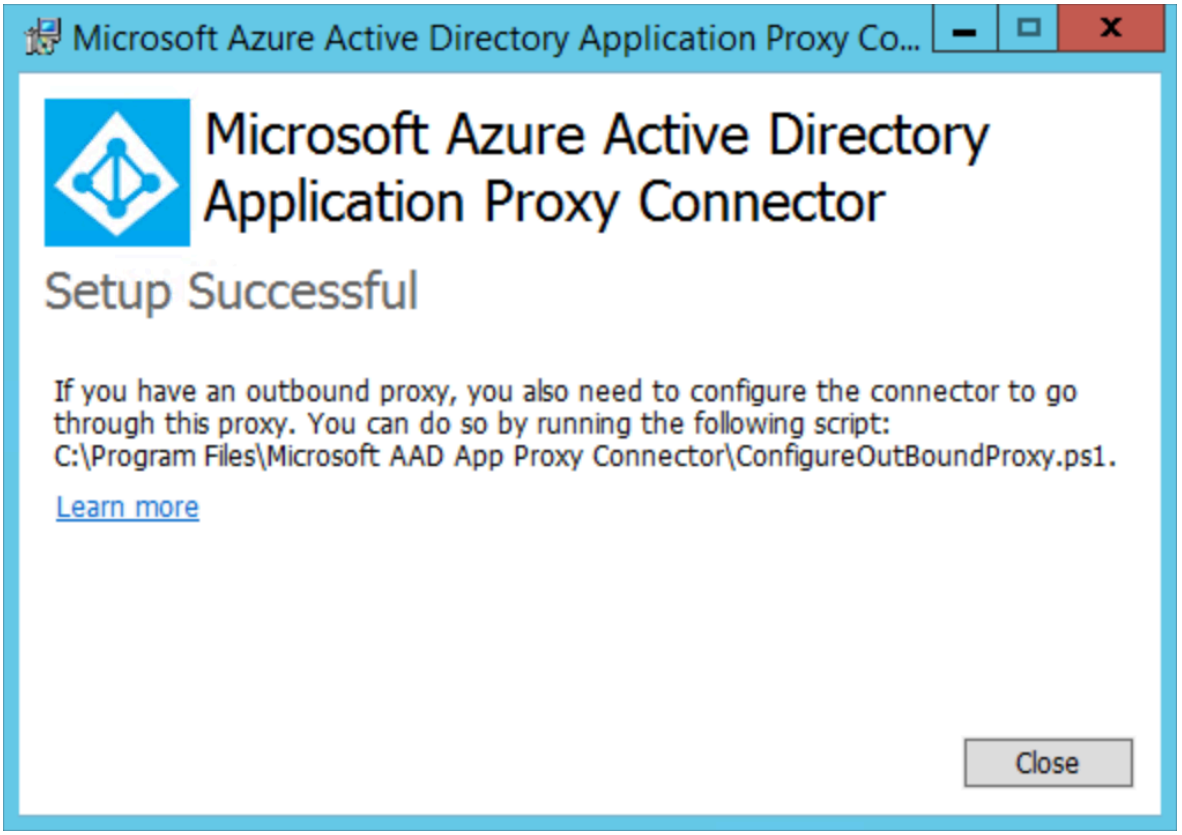
6. Start **AADApplicationProxyConnectorInstaller.exe**
7. Read the license terms and then select **I agree to the license terms and conditions**. Select **Install**



8. Sign-in as at least an Application Administrator



9. When the installation completes. Read the information regarding outbound proxy servers. Select **Close**



10. Repeat steps 5 - 10 for each device that will run the Microsoft Entra application proxy connector for Windows Hello for Business certificate deployments

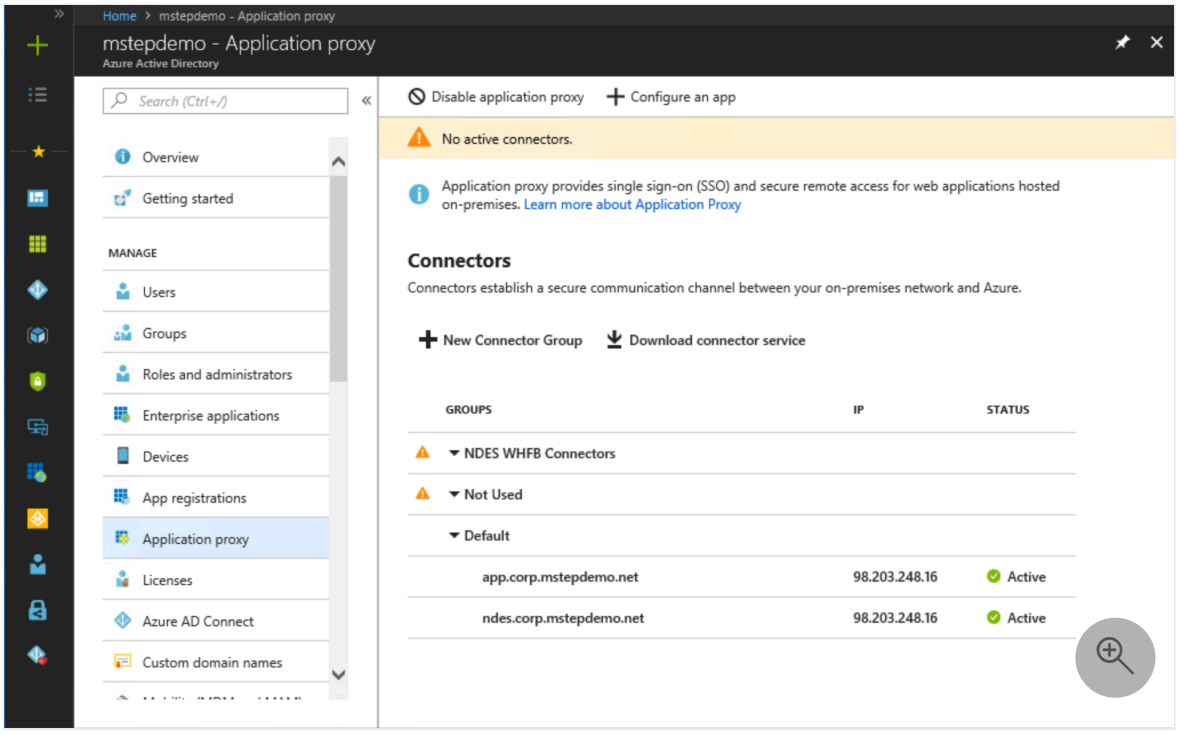
Create a Connector Group

Sign-in a workstation with access equivalent to a *domain user*.

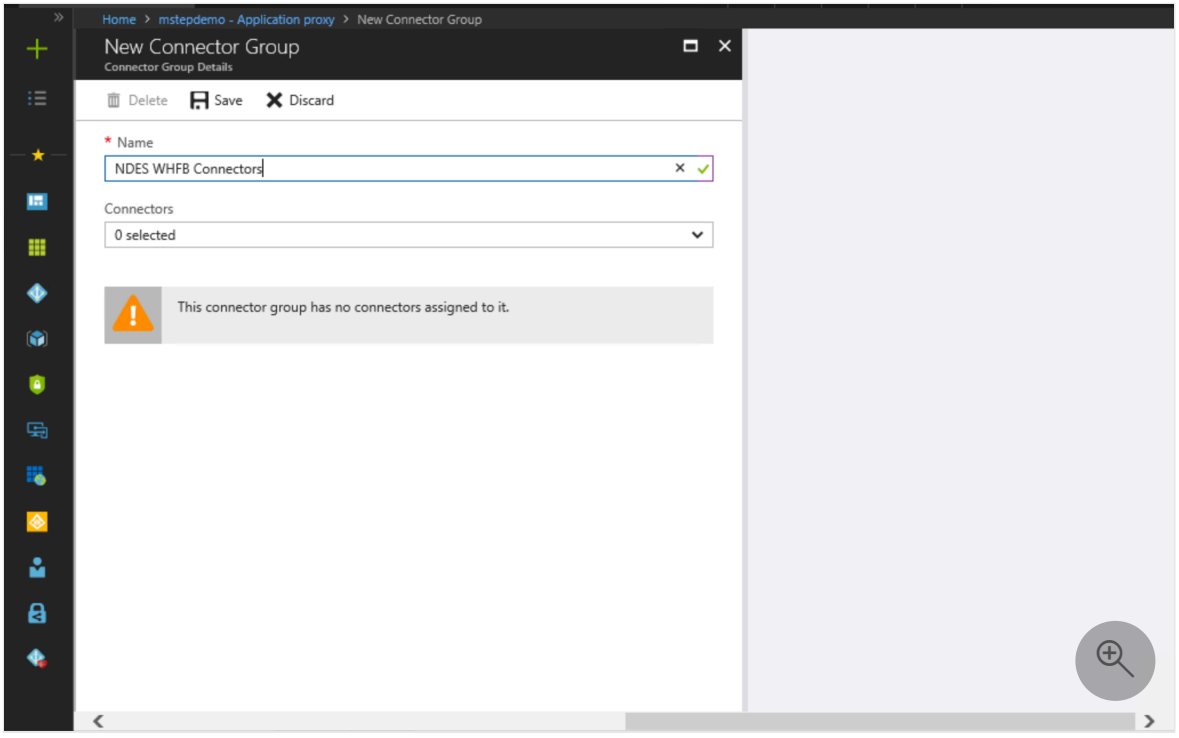
1. Access to the Microsoft Entra admin center as at least an Application Administrator

2. Select **All Services**. Type **Microsoft Entra ID** to filter the list of services. Under **SERVICES**, select **Microsoft Entra ID**

3. Under **MANAGE**, select **Application proxy**.



4. Select **New Connector Group**. Under **Name**, type **NDES WHFB Connectors**.



5. Select each connector agent in the **Connectors** list that will service Windows Hello for Business certificate enrollment requests

6. Select **Save**.

Create the Azure Application Proxy

Sign-in a workstation with access equivalent to a *domain user*.

1. Access to the Microsoft Entra admin center as at least an Application Administrator

2. Select **All Services**. Type **Microsoft Entra ID** to filter the list of services. Under **SERVICES**, select **Microsoft Entra ID**

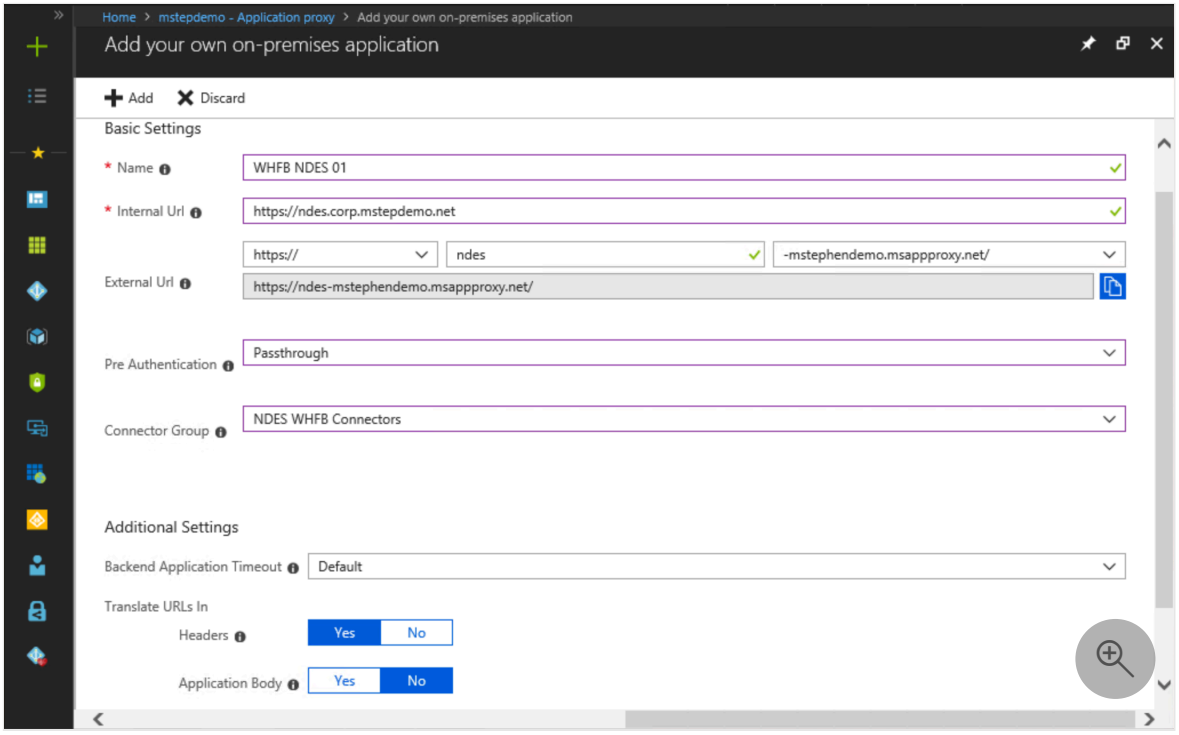
3. Under **MANAGE**, select **Application proxy**

4. Select **Configure an app**

5. Under **Basic Settings** next to **Name**, type **WHFB NDES 01**. Choose a name that correlates this Microsoft Entra application proxy setting with the on-premises NDES server. Each NDES server must have its own Microsoft Entra application proxy as two NDES servers can't share the same internal URL

6. Next to **Internal URL**, type the internal, fully qualified DNS name of the NDES server associated with this Microsoft Entra application proxy. For example, `https://ndes.corp.mstepdemo.net`. You need to match the primary host name (AD Computer Account name) of the NDES server, and prefix the URL with **https**

7. Under **Internal URL**, select **https://** from the first list. In the text box next to **https://**, type the hostname you want to use as your external hostname for the Microsoft Entra application proxy. In the list next to the hostname you typed, select a DNS suffix you want to use externally for the Microsoft Entra application proxy. It's recommended to use the default, -[tenantName].msapproxy.net where **[tenantName]** is your current Microsoft Entra tenant name (-mstephendemo.msappproxy.net).



8. Select **Passthrough** from the **Pre Authentication** list
9. Select **NDES WHFB Connectors** from the **Connector Group** list
10. Under **Additional Settings**, select **Default** from **Backend Application Timeout**. Under the **Translate URLs In** section, select **Yes** next to **Headers** and select **No** next to **Application Body**
11. Select **Add**
12. Sign-out of the Azure portal.

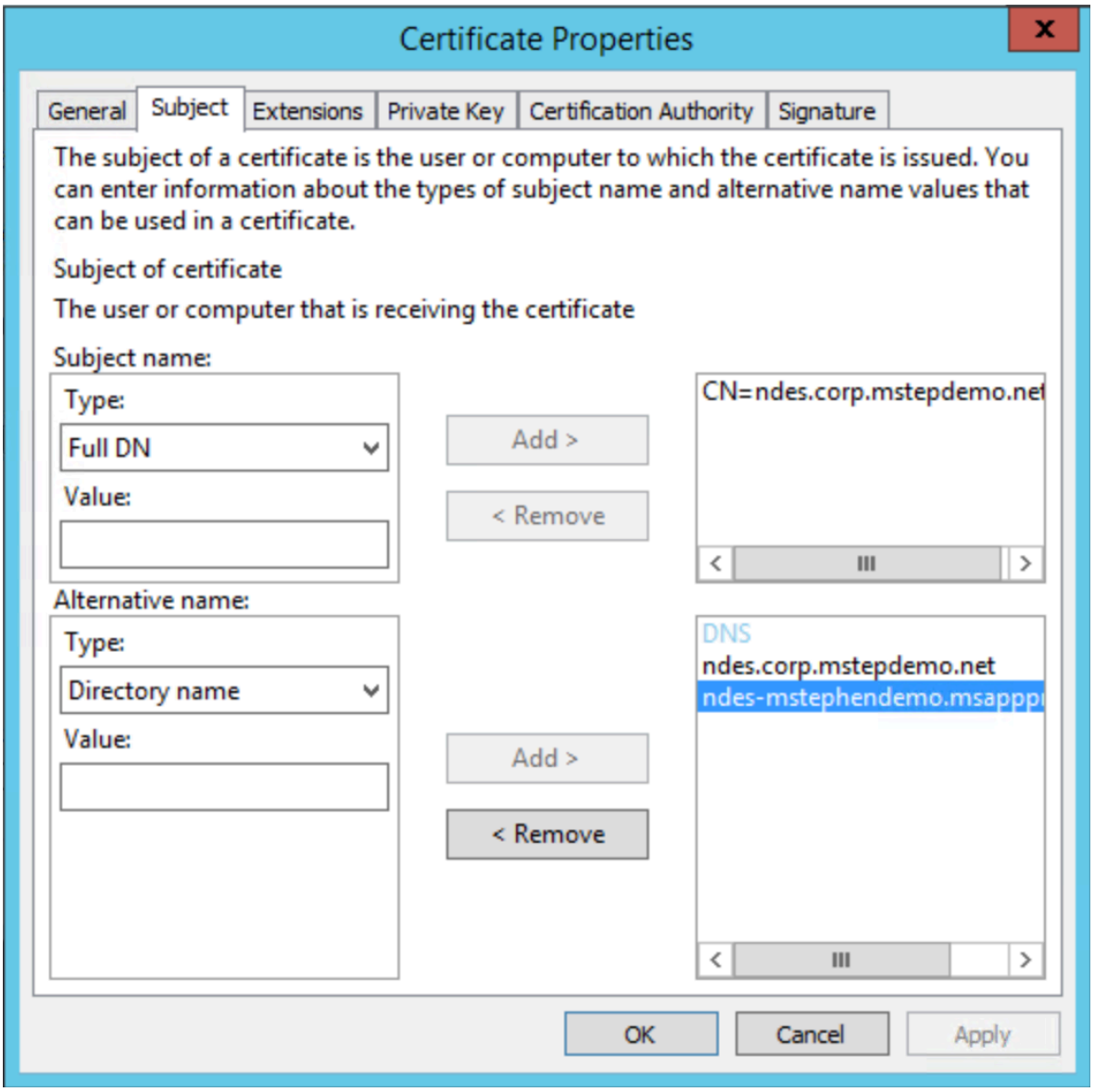
> **Important**
>
> Write down the internal and external URLs. You will need this information when you enroll the NDES-Intune Authentication certificate.

Enroll the NDES-Intune Authentication certificate

This task enrolls a client and server authentication certificate used by the Intune connector and the NDES server.

Sign-in the NDES server with access equivalent to *local administrators*.

1. Start the Local Computer **Certificate Manager** (certlm.msc)
2. Expand the **Personal** node in the navigation pane
3. Right-click **Personal**. Select **All Tasks** and **Request New Certificate**
4. Select **Next** on the **Before You Begin** page
5. Select **Next** on the **Select Certificate Enrollment Policy** page
6. On the **Request Certificates** page, Select the **NDES-Intune Authentication** check box
7. Select the **More information is required to enroll for this certificate. Click here to configure settings** link



8. Under **Subject name**, select **Common Name** from the **Type** list. Type the internal URL used in the previous task (without the https://, for example **ndes.corp.mstepdemo.net**) and then select **Add**

9. Under **Alternative name**, select **DNS** from the **Type** list. Type the internal URL used in the previous task (without the https://, for example **ndes.corp.mstepdemo.net**). Select **Add**. Type the external URL used in the previous task (without the https://, for example **ndes-mstephendemo.msappproxy.net**). Select **Add**. Select **OK** when finished

10. Select **Enroll**

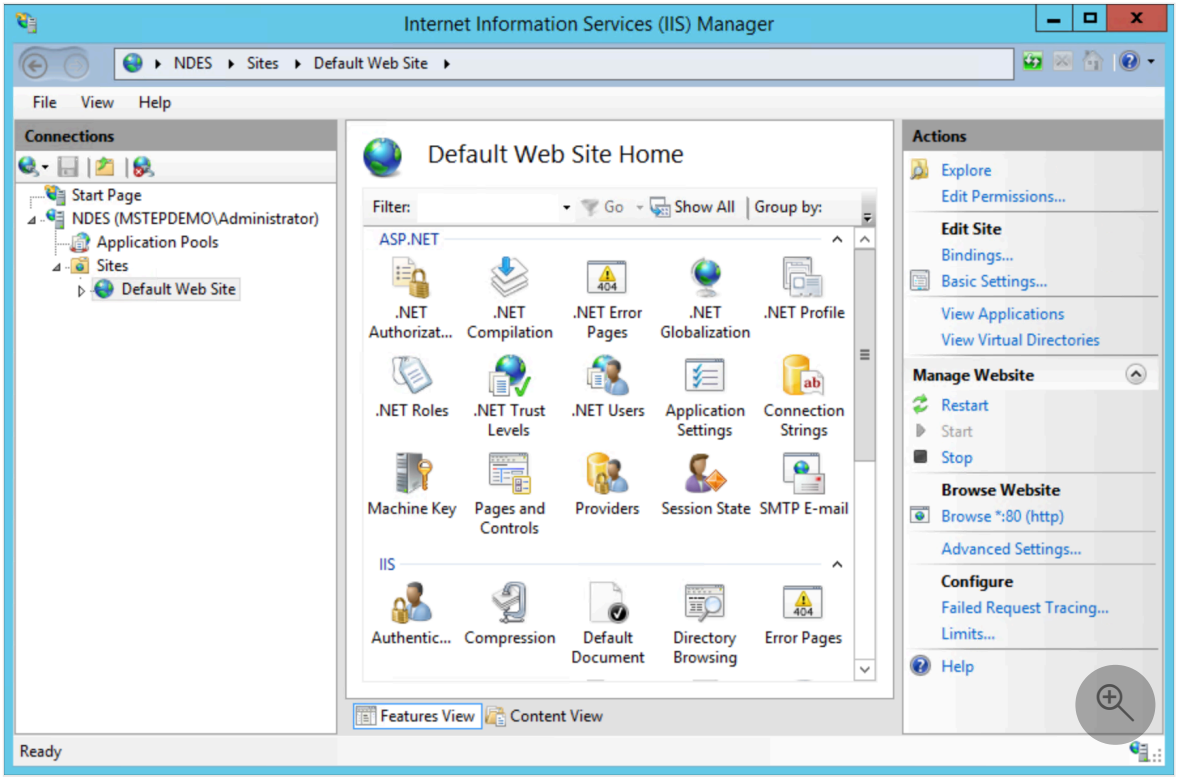
11. Repeat these steps for all NDES Servers used to request Windows Hello for Business authentication certificates for Microsoft Entra joined devices

Configure the Web Server Role

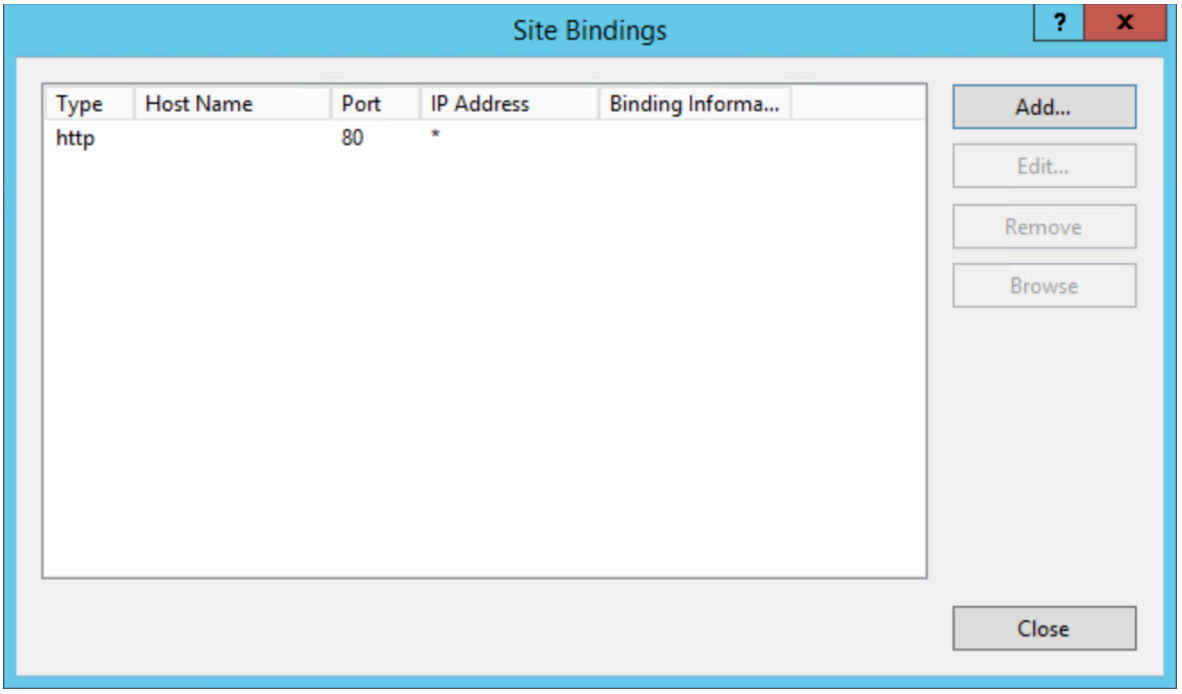
This task configures the Web Server role on the NDES server to use the server authentication certificate.

Sign-in the NDES server with access equivalent to *local administrator*.

1. Start **Internet Information Services (IIS) Manager** from **Administrative Tools**
2. Expand the node that has the name of the NDES server. Expand **Sites** and select **Default Web Site**

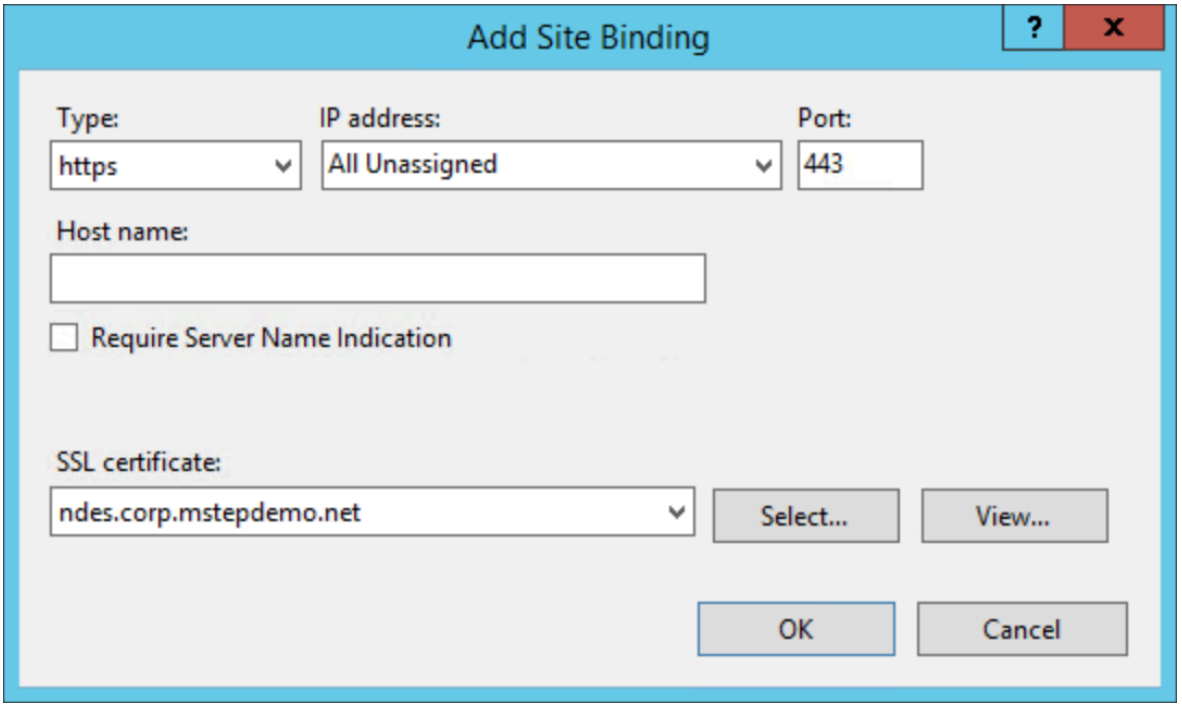


3. Select **Bindings...** under **Actions**. Select **Add**.



4. Select **https** from **Type**. Confirm the value for **Port** is **443**

5. Select the certificate you previously enrolled from the **SSL certificate** list. Select **OK**



6. Select **http** from the **Site Bindings** list. Select **Remove**

7. Select **Close** on the **Site Bindings** dialog box

8. Close **Internet Information Services (IIS) Manager**

Verify the configuration

This task confirms the TLS configuration for the NDES server.

Sign-in the NDES server with access equivalent to *local administrator*.

Disable Internet Explorer Enhanced Security Configuration

1. Open **Server Manager**. Select **Local Server** from the navigation pane
2. Select **On** next to **IE Enhanced Security Configuration** in the **Properties** section
3. In the **Internet Explorer Enhanced Security Configuration** dialog, under **Administrators**, select **Off**. Select **OK**
4. Close **Server Manager**

Test the NDES web server

1. Open **Internet Explorer**

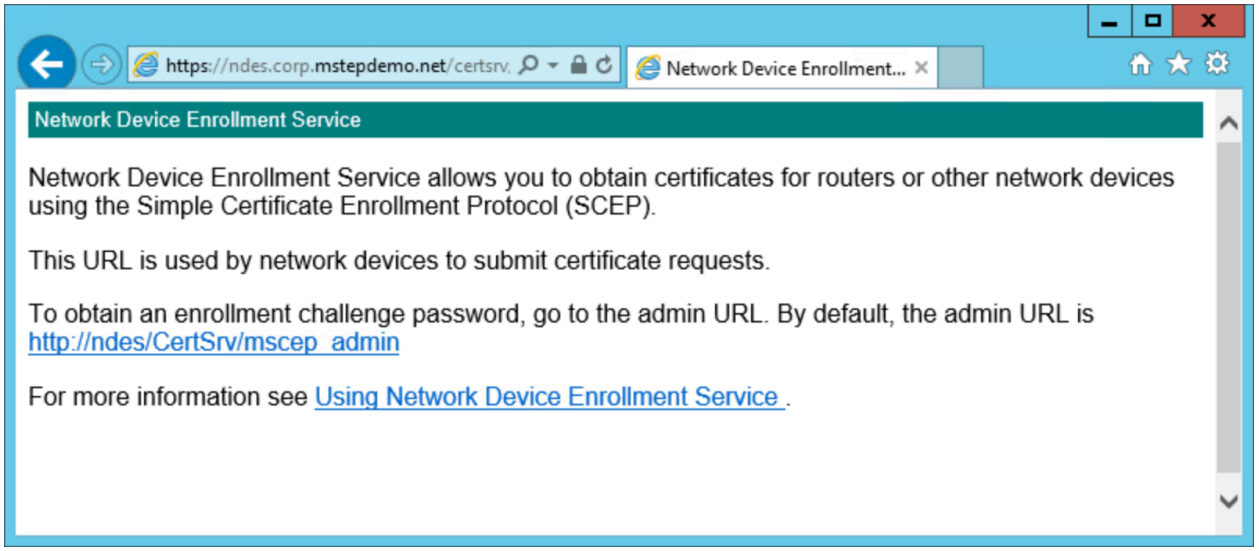
2. In the navigation bar, type

```
https
```

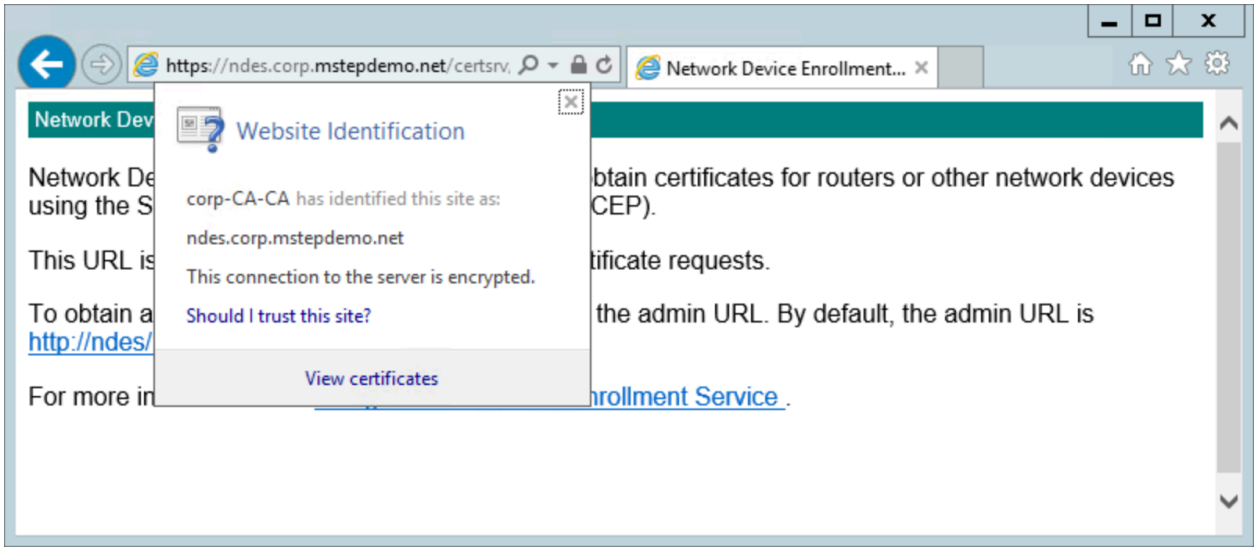
```
https://[fqdnHostName]/certsrv/mscep/mscep.dll
```

where **[fqdnHostName]** is the fully qualified internal DNS host name of the NDES server.

A web page similar to the following should appear in your web browser. If you don't see a similar page, or you get a **503 Service unavailable** message, ensure the NDES Service account has the proper user rights. You can also review the Application event log for events with the **NetworkDeviceEnrollmentService** source.



Confirm the web site uses the server authentication certificate.



Configure Network Device Enrollment Services to work with Microsoft Intune

You have successfully configured the Network Device Enrollment Services. You must now modify the configuration to work with the Intune Certificate Connector. In this task, you'll enable the NDES server and http.sys to handle long URLs.

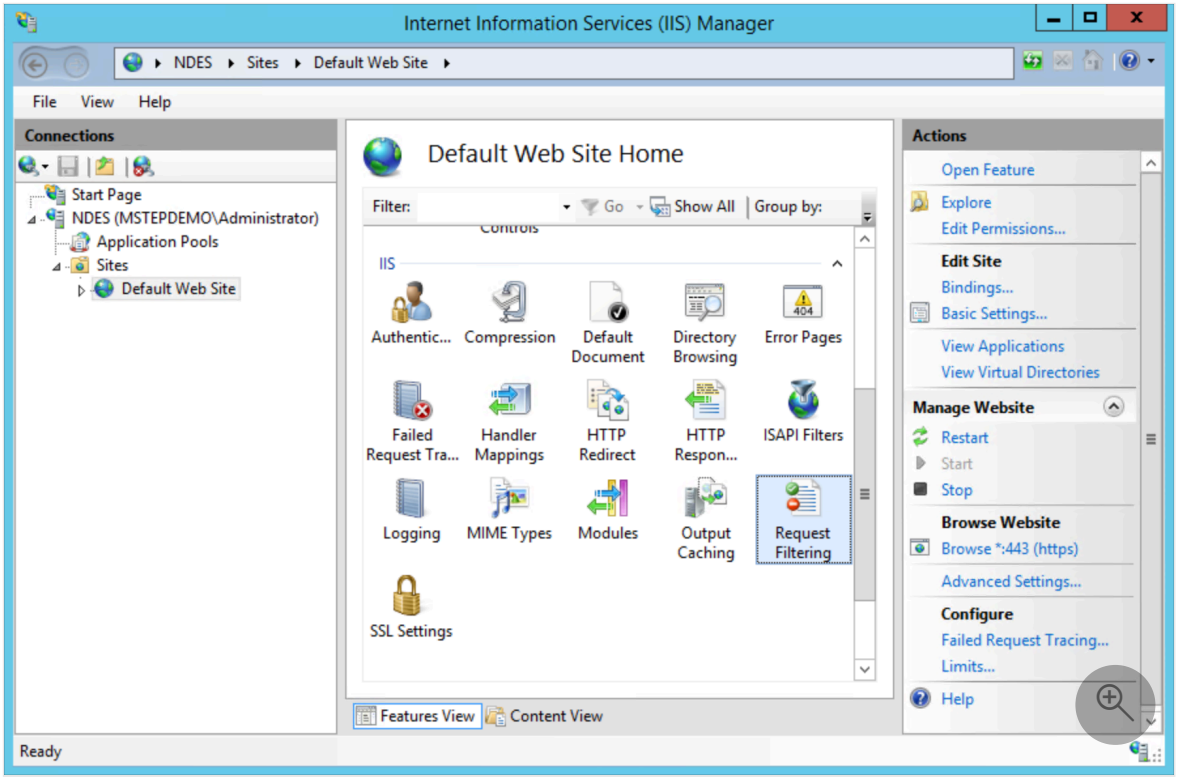
- Configure NDES to support long URLs

Configure NDES and HTTP to support long URLs

Sign-in the NDES server with access equivalent to *local administrator*.

Configure the Default Web Site

1. Start **Internet Information Services (IIS) Manager** from **Administrative Tools**
2. Expand the node that has the name of the NDES server. Expand **Sites** and select **Default Web Site**
3. In the content pane, double-click **Request Filtering**. Select **Edit Feature Settings...** in the action pane



4. Select **Allow unlisted file name extensions**
5. Select **Allow unlisted verbs**
6. Select **Allow high-bit characters**
7. Type **30000000** in **Maximum allowed content length (Bytes)**

8. Type **65534** in **Maximum URL length (Bytes)**
9. Type **65534** in **Maximum query string (Bytes)**
10. Select **OK**. Close **Internet Information Services (IIS) Manager**

Configure Parameters for HTTP.SYS

1. Open an elevated command prompt
2. Run the following commands:

```
Windows Command Prompt
```

```
reg add HKLM\SYSTEM\CurrentControlSet\Services\HTTP\Parameters /v
MaxFieldLength /t REG_DWORD /d 65534
reg add HKLM\SYSTEM\CurrentControlSet\Services\HTTP\Parameters /v
MaxRequestBytes /t REG_DWORD /d 65534
```

3. Restart the NDES server

Download, Install and Configure the Intune Certificate Connector

The Intune Certificate Connector application enables Microsoft Intune to enroll certificates using your on-premises PKI for users on devices managed by Microsoft Intune.

To learn how to download, install, and configure the Intune Certificate Connector, see Install the Certificate Connector for Microsoft Intune.

Configure the NDES Connector for certificate revocation (Optional)

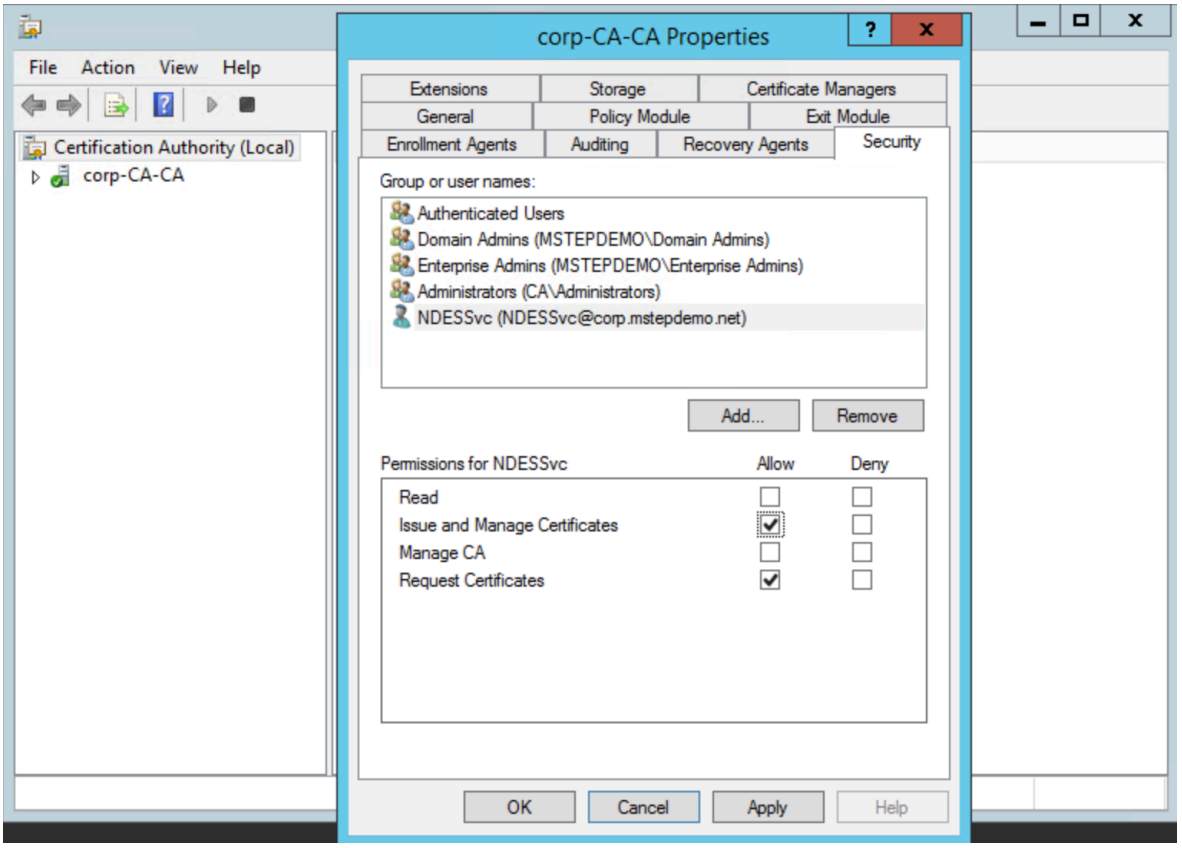
Optionally (not required), you can configure the Intune connector for certificate revocation when a device is wiped, unenrolled, or when the certificate profile falls out of scope for the targeted user (users are removed, deleted, or the profile is deleted). You need to select the **Certificate revocation** option during the connector configuration to enable automatic certificate revocation for certificates issued from a Microsoft Active Directory Certification Authority. Additionally, you need to enable the NDES Service account for revocation.

1. Sign in the certificate authority used by the NDES Connector with access equivalent to *domain administrator*

2. Start the **Certification Authority** management console

3. In the navigation pane, right-click the name of the certificate authority and select **Properties**

4. Select the **Security** tab, then select **Add**. In the **Enter the object names to select** box, enter **NDESSvc** (or the name you gave the NDES Service account). Select **Check Names**, then select **OK**. Select the NDES Service account from the **Group or user names** list. Select **Allow** for the **Issue and Manage Certificates** permission. Select **OK**



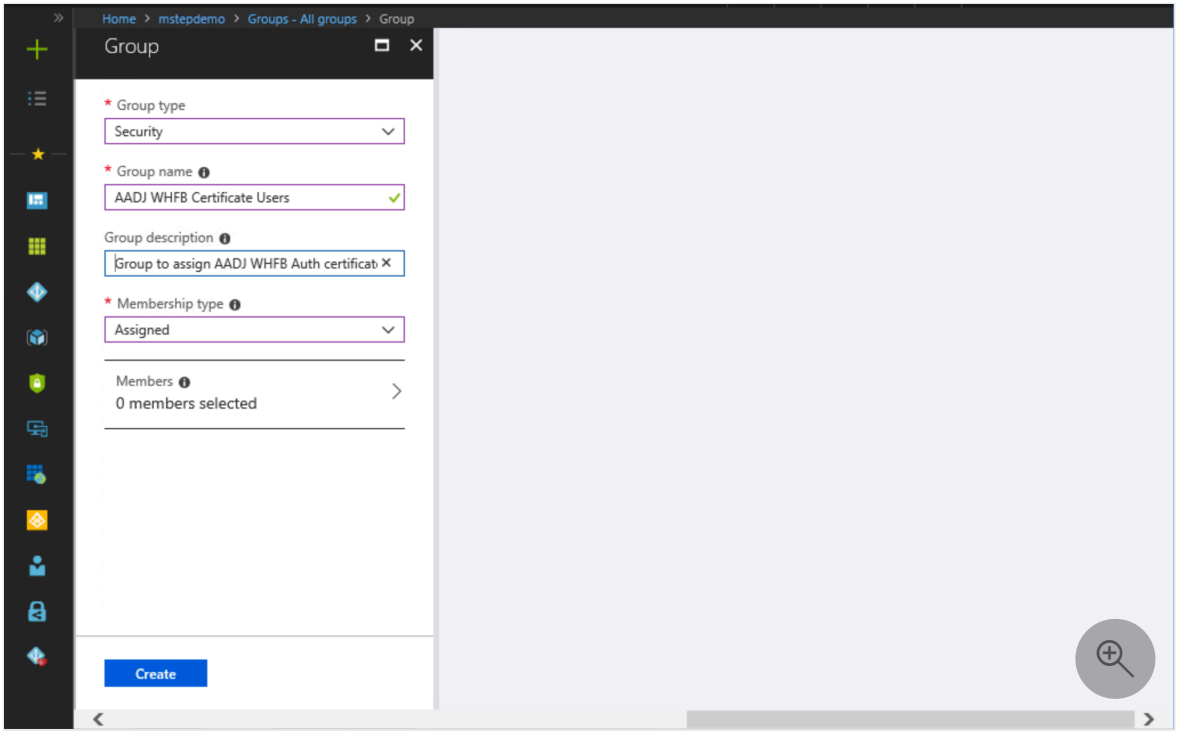
5. Close the **Certification Authority**

Create and Assign a Simple Certificate Enrollment Protocol (SCEP) Certificate Profile

Create an ENTRA JOINED WHFB Certificate Users Group

Sign-in a workstation with access equivalent to a *domain user*.

1. Access to the Microsoft Entra admin center as at least an Application Administrator
2. Select **All Services**. Type **Microsoft Entra ID** to filter the list of services. Under **SERVICES**, select **Microsoft Entra ID**
3. Select **Groups**. Select **New group**
4. Select **Security** from the **Group type** list
5. Under **Group Name**, type the name of the group. For example, **ENTRA JOINED WHFB Certificate Users**
6. Provide a **Group description**, if applicable
7. Select **Assigned** from the **Membership type** list



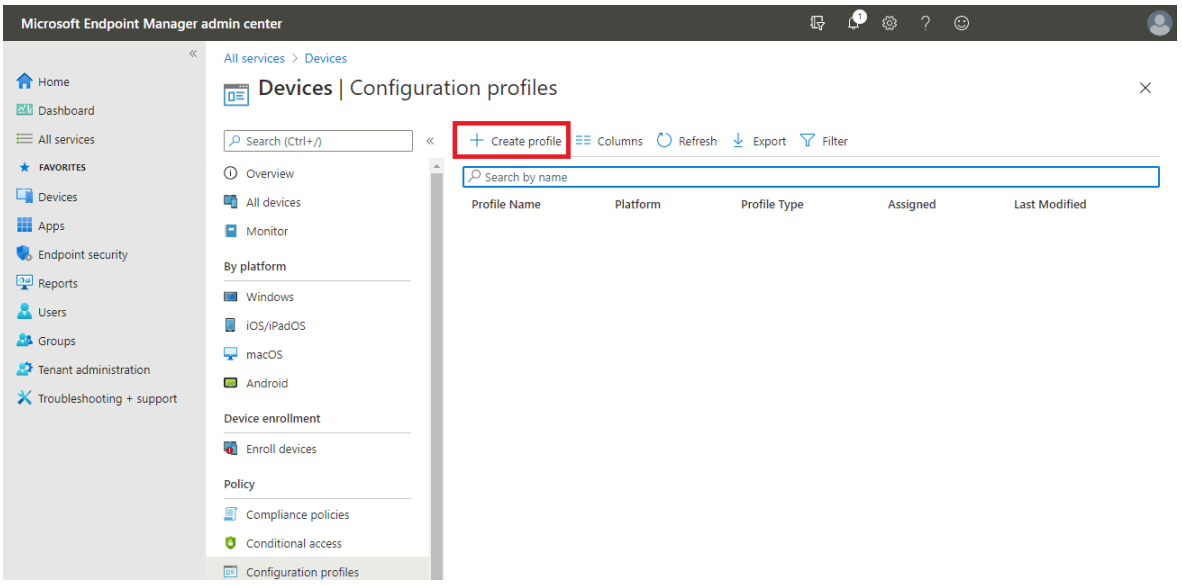
8. Select **Members**. Use the **Select members** pane to add members to this group. When finished, select **Select**
9. Select **Create**.

Create a SCEP Certificate Profile

Sign-in a workstation with access equivalent to a *domain user*.

1. Sign in to the Microsoft Intune admin center
2. Select **Devices**, and then select **Configuration Profiles**

3. Select **Create Profile**.



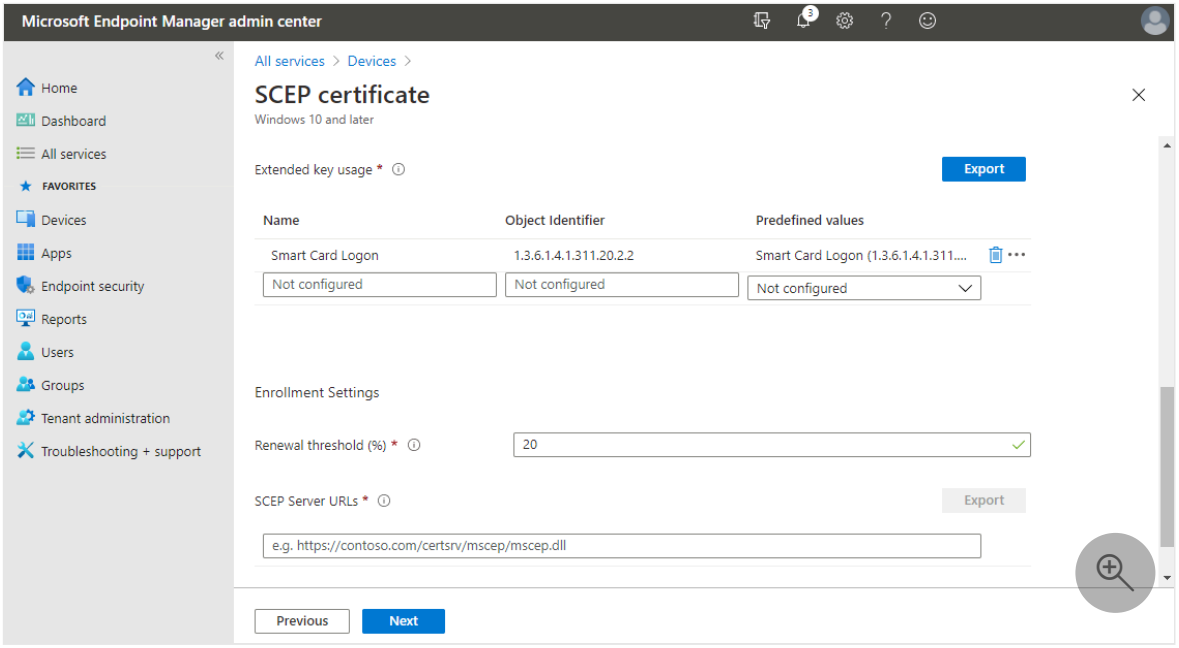
4. Select **Windows 10 and later** from the **Platform** list
5. Choose **SCEP certificate** from the **Profile** list, and select **Create**
6. The **SCEP Certificate** wizard should open. Next to **Name**, type **WHFB Certificate Enrollment**
7. Next to **Description**, provide a description meaningful for your environment, then select **Next**
8. Select **User** as a certificate type
9. Configure **Certificate validity period** to match your organization.

> **ⓘ Important**
>
> Remember that you need to configure your certificate authority to allow Microsoft Intune to configure certificate validity

10. Select **Enroll to Windows Hello for Business, otherwise fail (Windows 10 and later)** from the **Key storage provider (KSP)** list
11. Next to **Subject name format**, type **CN={{OnPrem_Distinguished_Name}}** to make the on-premises distinguished name the subject of the issued certificate

> **ⓘ Note**
>
> If the distinguished name contains special characters like a plus sign ("+"), comma (","), semicolon (";"), or equal sign ("="), the bracketed name must be enclosed in quotation marks: `CN="{{OnPrem_Distinguished_Name}}"`.
>
> If the length of the distinguished name is more than 64 characters, the name length enforcement on the Certification Authority must be disabled.

12. Specify **User Principal Name (UPN)** as a **Subject Alternative Name** parameter. Set its value as {{UserPrincipalName}}
13. Refer to the "Configure Certificate Templates on NDES" task for how you configured the **ENTRA JOINED WHFB Authentication** certificate template in the registry. Select the appropriate combination of key usages from the **Key Usages** list that map to the configured NDES template in the registry. In this example, the **ENTRA JOINED WHFB Authentication** certificate template was added to the **SignatureTemplate** registry value name. The **Key usage** that maps to that registry value name is **Digital Signature**
14. Select a previously configured **Trusted certificate** profile that matches the root certificate of the issuing certificate authority as a root certificate for the profile
15. Under **Extended key usage**, type **Smart Card Logon** under **Name**. Type **1.3.6.1.4.1.311.20.2.2** under **Object identifier**. Select **Add**
16. Type a percentage (without the percent sign) next to **Renewal Threshold** to determine when the certificate should attempt to renew. The recommended value is **20**

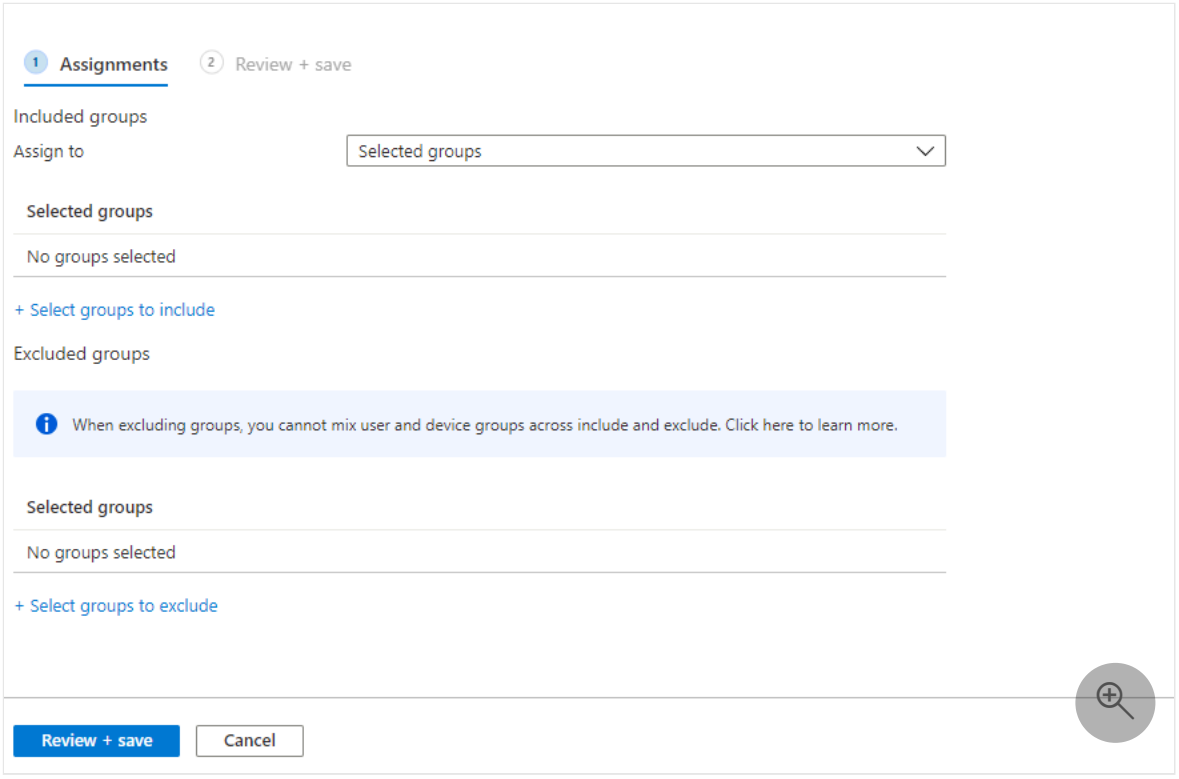


17. Under **SCEP Server URLs**, type the fully qualified external name of the Microsoft Entra application proxy you configured. Append to the name **/certsrv/mscep/mscep.dll**. For example, `https://ndes-mtephendemo.msappproxy.net/certsrv/mscep/mscep.dll`. Select **Add**. Repeat this step for each additional NDES Microsoft Entra application proxy you configured to issue Windows Hello for Business certificates. Microsoft Intune round-robin load balances requests among the URLs listed in the SCEP certificate profile
18. Select **Next**
19. Select **Next** several times to skip the **Scope tags**, **Assignments**, and **Applicability Rules** steps of the wizard and select **Create**

Assign Group to the WHFB Certificate Enrollment Certificate Profile

Sign-in a workstation with access equivalent to a *domain user*.

1. Sign-in to the Microsoft Intune admin center
2. Select **Devices**, and then select **Configuration Profiles**
3. Select **WHFB Certificate Enrollment**
4. Select **Properties**, and then select **Edit** next to the **Assignments** section
5. In the **Assignments** pane, select **Selected Groups** from the **Assign to** list. Select **Select groups to include**.



6. Select the **ENTRA JOINED WHFB Certificate Users** group. Select **Select**
7. Select **Review + Save**, and then **Save**

You have successfully completed the configuration. Add users that need to enroll a Windows Hello for Business authentication certificate to the **ENTRA JOINED WHFB Certificate Users** group. This group, combined with the device enrollment Windows Hello for Business configuration prompts the user to enroll for Windows Hello for Business and enroll a certificate that can be used to authentication to on-premises resources.

> **Note**
>
> The Passport for Work configuration service provider (CSP) which is used to manage Windows Hello for Business with Mobile Device Management (MDM) contains a policy called UseCertificateForOnPremAuth. This policy is not needed

when deploying certificates to Windows Hello for Business users through the instructions outlined in this document and should not be configured. Devices managed with MDM where UseCertificateForOnPremAuth is enabled will fail a prerequisite check for Windows Hello for Business provisioning. This failure will block users from setting up Windows Hello for Business if they don't already have it configured.

Section Review

- ✓ Requirements
- ✓ Prepare Microsoft Entra Connect
- ✓ Prepare the Network Device Enrollment Services (NDES) Service Account
- ✓ Prepare Active Directory Certificate Authority
- ✓ Install and Configure the NDES Role
- ✓ Configure Network Device Enrollment Services to work with Microsoft Intune
- ✓ Download, Install, and Configure the Intune Certificate Connector
- ✓ Create and Assign a Simple Certificate Enrollment Protocol (SCEP Certificate Profile)

Feedback

Was this page helpful? Yes No

Provide product feedback

On-premises key trust deployment guide

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises
- **Trust type:** key trust
- **Join type:** domain join

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Public Key Infrastructure
- ✓ Authentication
- ✓ Device configuration
- ✓ Licensing for cloud services
- ✓ Windows requirements
- ✓ Windows Server requirements
- ✓ Prepare users to use Windows Hello

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure and validate the Public Key Infrastructure
- ✓ Prepare and deploy AD FS with MFA
- ✓ Configure and enroll in Windows Hello for Business

Configure and validate the Public Key Infrastructure

Windows Hello for Business must have a Public Key Infrastructure (PKI) when using the *key trust* or *certificate trust* models. The domain controllers must have a certificate, which serves as a root of trust for clients. The certificate ensures that clients don't communicate with rogue domain controllers.

Deploy an enterprise certification authority

This guide assumes most enterprises have an existing public key infrastructure. Windows Hello for Business depends on an enterprise PKI running the Windows Server *Active Directory Certificate Services* role.
If you don't have an existing PKI, review Certification Authority Guidance to properly design your infrastructure. Then, consult the Test Lab Guide: Deploying an AD CS Two-Tier PKI Hierarchy for instructions on how to configure your PKI using the information from your design session.

Lab-based PKI

The following instructions may be used to deploy simple public key infrastructure that is suitable **for a lab environment**.

Sign in using *Enterprise Administrator* equivalent credentials on a Windows Server where you want the certification authority (CA) installed.

> **Note**
>
> Never install a certification authority on a domain controller in a production environment.

1. Open an elevated Windows PowerShell prompt
2. Use the following command to install the Active Directory Certificate Services role.

```
PowerShell

Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
```

3. Use the following command to configure the CA using a basic certification authority configuration

```
PowerShell

Install-AdcsCertificationAuthority
```

Configure the enterprise PKI

Configure domain controller certificates

Clients must trust the domain controllers, and the best way to enable the trust is to ensure that each domain controller has a *Kerberos Authentication* certificate. Installing a certificate on the domain controllers enables the Key Distribution Center (KDC) to prove its identity to other members of the domain. The certificates provide clients a root of trust external to the domain, namely the *enterprise certification authority*.

Domain controllers automatically request a *domain controller certificate* (if published) when they discover an enterprise CA is added to Active Directory. The certificates based on the *Domain Controller* and *Domain Controller Authentication* certificate templates don't include the *KDC Authentication* object identifier (OID), which was later added to the Kerberos RFC. Therefore, domain controllers need to request a certificate based on the *Kerberos Authentication* certificate template.

By default, the Active Directory CA provides and publishes the *Kerberos Authentication* certificate template. The cryptography configuration included in the template is based on older and less performant cryptography APIs. To ensure domain controllers request the proper certificate with the best available cryptography, use the *Kerberos Authentication* certificate template as a *baseline* to create an updated domain controller certificate template.

> **ⓘ Important**
>
> The certificates issued to the domain controllers must meet the following requirements:
>
> - The *Certificate Revocation List (CRL) distribution point* extension must point to a valid CRL, or an *Authority Information Access (AIA)* extension that points to an Online Certificate Status Protocol (OCSP) responder
> - Optionally, the certificate *Subject* section could contain the directory path of the server object (the distinguished name)
> - The certificate *Key Usage* section must contain *Digital Signature* and *Key Encipherment*
> - Optionally, the certificate *Basic Constraints* section should contain: `[Subject Type=End Entity, Path Length Constraint=None]`
> - The certificate *extended key usage* section must contain Client Authentication (`1.3.6.1.5.5.7.3.2`), Server Authentication (`1.3.6.1.5.5.7.3.1`), and KDC

Authentication (`1.3.6.1.5.2.3.5`)

- The certificate *Subject Alternative Name* section must contain the Domain Name System (DNS) name
- The certificate template must have an extension that has the value `DomainController`, encoded as a BMPstring. If you are using Windows Server Enterprise Certificate Authority, this extension is already included in the domain controller certificate template
- The domain controller certificate must be installed in the local computer's certificate store

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Kerberos Authentication** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Domain Controller Authentication (Kerberos)* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **None** from the **Subject name format** list • Select **DNS name** from the **Include this information in alternate subject** list • Clear all other items

Tab Name	Configurations
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Supersede existing domain controller certificates

The domain controllers may have an existing domain controller certificate. The Active Directory Certificate Services provides a default certificate template for domain controllers called *domain controller certificate*. Later releases of Windows Server provided a new certificate template called *domain controller authentication certificate*. These certificate templates were provided prior to the update of the Kerberos specification that stated Key Distribution Centers (KDCs) performing certificate authentication needed to include the *KDC Authentication* extension.

The *Kerberos Authentication* certificate template is the most current certificate template designated for domain controllers, and should be the one you deploy to all your domain controllers.
The *autoenrollment* feature allows you to replace the domain controller certificates. Use the following configuration to replace older domain controller certificates with new ones, using the *Kerberos Authentication* certificate template.

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the *Domain Controller Authentication (Kerberos)* (or the name of the certificate template you created in the previous section) template in the details pane and select **Properties**
4. Select the **Superseded Templates** tab. Select **Add**
5. From the **Add Superseded Template** dialog, select the *Domain Controller* certificate template and select **OK** > **Add**
6. From the **Add Superseded Template** dialog, select the *Domain Controller Authentication* certificate template and select **OK**

7. From the **Add Superseded Template** dialog, select the *Kerberos Authentication* certificate template and select **OK**
8. Add any other enterprise certificate templates that were previously configured for domain controllers to the **Superseded Templates** tab
9. Select **OK** and close the **Certificate Templates** console

The certificate template is configured to supersede all the certificate templates provided in the *superseded templates* list.
However, the certificate template and the superseding of certificate templates isn't active until the template is published to one or more certificate authorities.

> ⓘ **Note**
>
> The domain controller's certificate must chain to a root in the NTAuth store. By default, the Active Directory Certificate Authority's root certificate is added to the NTAuth store. If you are using a non-Microsoft CA, this may not be done by default. If the domain controller certificate does not chain to a root in the NTAuth store, user authentication will fail. To see all certificates in the NTAuth store, use the following command:
>
> `Certutil -viewstore -enterprise NTAuth`

Configure an internal web server certificate template

Windows clients communicate with AD FS via HTTPS. To meet this need, a *server authentication* certificate must be issued to all the nodes in the AD FS farm. On-premises deployments can use a *server authentication* certificate issued by the enterprise PKI. A *server authentication* certificate template must be configured, so the AD FS nodes can request a certificate.

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Web Server** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Internal Web Server* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Request Handling	Select **Allow private key to be exported**
Subject Name	Select **Supply in the request**
Security	Add **Domain Computers** with **Enroll** access
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Unpublish Superseded Certificate Templates

The certification authority only issues certificates based on published certificate templates. For security, it's a good practice to unpublish certificate templates that the CA isn't configured to issue, including the pre-published templates from the role installation and any superseded templates.

The newly created *domain controller authentication* certificate template supersedes previous domain controller certificate templates. Therefore, you need to unpublish these certificate templates from all issuing certificate authorities.

Sign in to the CA or management workstation with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console

2. Expand the parent node from the navigation pane > **Certificate Templates**
3. Right-click the *Domain Controller* certificate template and select **Delete**. Select **Yes** on the **Disable certificate templates** window
4. Repeat step 3 for the *Domain Controller Authentication* and *Kerberos Authentication* certificate templates

Publish certificate templates to the CA

A certification authority can only issue certificates for certificate templates that are published to it. If you have more than one CA, and you want more CAs to issue certificates based on the certificate template, then you must publish the certificate template to them.

Sign in to the CA or management workstations with **Enterprise Admin** equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane
3. Select **Certificate Templates** in the navigation pane
4. Right-click the **Certificate Templates** node. Select **New** > **Certificate Template** to issue
5. In the **Enable Certificates Templates** window, select the *Domain Controller Authentication (Kerberos)* and *Internal Web Server* templates you created in the previous steps. Select **OK** to publish the selected certificate templates to the certification authority
6. If you published the *Domain Controller Authentication (Kerberos)* certificate template, then unpublish the certificate templates you included in the superseded templates list

    - To unpublish a certificate template, right-click the certificate template you want to unpublish and select **Delete**. Select **Yes** to confirm the operation

7. Close the console

Configure and deploy certificates to domain controllers

Configure automatic certificate enrollment for the domain controllers

Domain controllers automatically request a certificate from the *Domain controller certificate* template. However, domain controllers are unaware of newer certificate templates or superseded configurations on certificate templates. For domain controllers to automatically enroll and renew of certificates, configure a GPO for automatic certificate enrollment, and link it to the *Domain Controllers* OU.

1. Open the **Group Policy Management Console** (gpmc.msc)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type *Domain Controller Auto Certificate Enrollment* in the name box and select **OK**
5. Right-click the **Domain Controller Auto Certificate Enrollment** Group Policy object and select **Edit**
6. In the navigation pane, expand **Policies** under **Computer Configuration**
7. Expand **Windows Settings** > **Security Settings** > **Public Key Policies**
8. In the details pane, right-click **Certificate Services Client - Auto-Enrollment** and select **Properties**
9. Select **Enabled** from the **Configuration Model** list
10. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box
11. Select the **Update certificates that use certificate templates** check box
12. Select **OK**
13. Close the **Group Policy Management Editor**

Deploy the domain controller auto certificate enrollment GPO

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and expand the node with the Active Directory domain name. Right-click the **Domain Controllers** organizational unit and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select *Domain Controller Auto Certificate Enrollment* or the name of the domain controller certificate enrollment Group Policy object you previously created
4. Select **OK**

Validate the configuration

Windows Hello for Business is a distributed system, which on the surface appears complex and difficult. The key to a successful deployment is to validate phases of work prior to moving to the next phase.

Confirm your domain controllers enroll the correct certificates and not any superseded certificate templates. Check that each domain controller completed the certificate autoenrollment.

Use the event logs

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Using the Event Viewer, navigate to the **Application and Services** > **Microsoft** > **Windows** > **CertificateServices-Lifecycles-System** event log
2. Look for an event indicating a new certificate enrollment (autoenrollment):

    - The details of the event include the certificate template on which the certificate was issued
    - The name of the certificate template used to issue the certificate should match the certificate template name included in the event
    - The certificate thumbprint and EKUs for the certificate are also included in the event
    - The EKU needed for proper Windows Hello for Business authentication is Kerberos Authentication, in addition to other EKUs provide by the certificate template

Certificates superseded by your new domain controller certificate generate an *archive event* in the Event Log. The archive event contains the certificate template name and thumbprint of the certificate that was superseded by the new certificate.

Certificate Manager

You can use the Certificate Manager console to validate the domain controller has the properly enrolled certificate based on the correct certificate template with the proper EKUs. Use **certlm.msc** to view certificate in the local computers certificate stores. Expand the **Personal** store and view the certificates enrolled for the computer. Archived certificates don't appear in Certificate Manager.

Certutil.exe

You can use `certutil.exe` command to view enrolled certificates in the local computer. Certutil shows enrolled and archived certificates for the local computer. From an elevated command prompt, run the following command:

Windows Command Prompt

```
certutil.exe -q -store my
```

To view detailed information about each certificate in the store, and to validate automatic certificate enrollment enrolled the proper certificates, use the following command:

Windows Command Prompt

```
certutil.exe -q -v -store my
```

Troubleshooting

Windows triggers automatic certificate enrollment for the computer during boot, and when Group Policy updates. You can refresh Group Policy from an elevated command prompt using `gpupdate.exe /force`.

Alternatively, you can forcefully trigger automatic certificate enrollment using `certreq.exe -autoenroll -q` from an elevated command prompt.

Use the event logs to monitor certificate enrollment and archive. Review the configuration, such as publishing certificate templates to issuing certification authority and the *allow* auto enrollment permissions.

Section review and next steps

Before moving to the next section, ensure the following steps are complete:

- ✓ Configure domain controller and web server certificate templates
- ✓ Supersede existing domain controller certificates
- ✓ Unpublish superseded certificate templates
- ✓ Publish the certificate templates to the CA
- ✓ Deploy certificates to the domain controllers
- ✓ Validate the domain controllers configuration

Next: prepare and deploy AD FS >

Feedback

Was this page helpful? Yes No

Provide product feedback

Prepare and deploy Active Directory Federation Services - on-premises key trust

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises ⓘ
- **Trust type:** key trust ⓘ
- **Join type:** domain join ⓘ

Windows Hello for Business works exclusively with the Active Directory Federation Service (AD FS) role included with Windows Server. The on-premises key trust deployment model uses AD FS for *key registration* and *device registration*.

The following guidance describes the deployment of a new instance of AD FS using the Windows Information Database (WID) as the configuration database.
WID is ideal for environments with no more than **30 federation servers** and no more than **100 relying party trusts**. If your environment exceeds either of these factors, or needs to provide *SAML artifact resolution*, *token replay detection*, or needs AD FS to operate as a federated provider role, then the deployment requires the use of SQL as a configuration database.
To deploy AD FS using SQL as its configuration database, review the Deploying a Federation Server Farm checklist.

A new AD FS farm should have a minimum of two federation servers for proper load balancing, which can be accomplished with external networking peripherals, or with using the Network Load Balancing Role included in Windows Server.

Prepare the AD FS deployment by installing and **updating** two Windows Servers.

Enroll for a TLS server authentication certificate

Typically, a federation service is an edge facing role. However, the federation services and instance used with the on-premises deployment of Windows Hello for Business does not need Internet connectivity.

The AD FS role needs a *server authentication* certificate for the federation services, and you can use a certificate issued by your enterprise (internal) CA. The server authentication certificate should have the following names included in the certificate, if you are requesting an individual certificate for each node in the federation farm:

- **Subject Name**: the internal FQDN of the federation server
- **Subject Alternate Name**: the federation service name (e.g. *sts.corp.contoso.com*) or an appropriate wildcard entry (e.g. **.corp.contoso.com*)

The federation service name is set when the AD FS role is configured. You can choose any name, but that name must be different than the name of the server or host. For example, you can name the host server *adfs* and the federation service *sts*. In this example, the FQDN of the host is *adfs.corp.contoso.com* and the FQDN of the federation service is *sts.corp.contoso.com*.

You can also issue one certificate for all hosts in the farm. If you chose this option, leave the subject name *blank*, and include all the names in the subject alternate name when creating the certificate request. All names should include the FQDN of each host in the farm and the federation service name.

When creating a wildcard certificate, mark the private key as exportable, so that the same certificate can be deployed across each federation server and web application proxy within the AD FS farm. Note that the certificate must be trusted (chain to a trusted root CA). Once you have successfully requested and enrolled the server authentication certificate on one node, you can export the certificate and private key to a PFX file using the Certificate Manager console. You can then import the certificate on the remaining nodes in the AD FS farm.

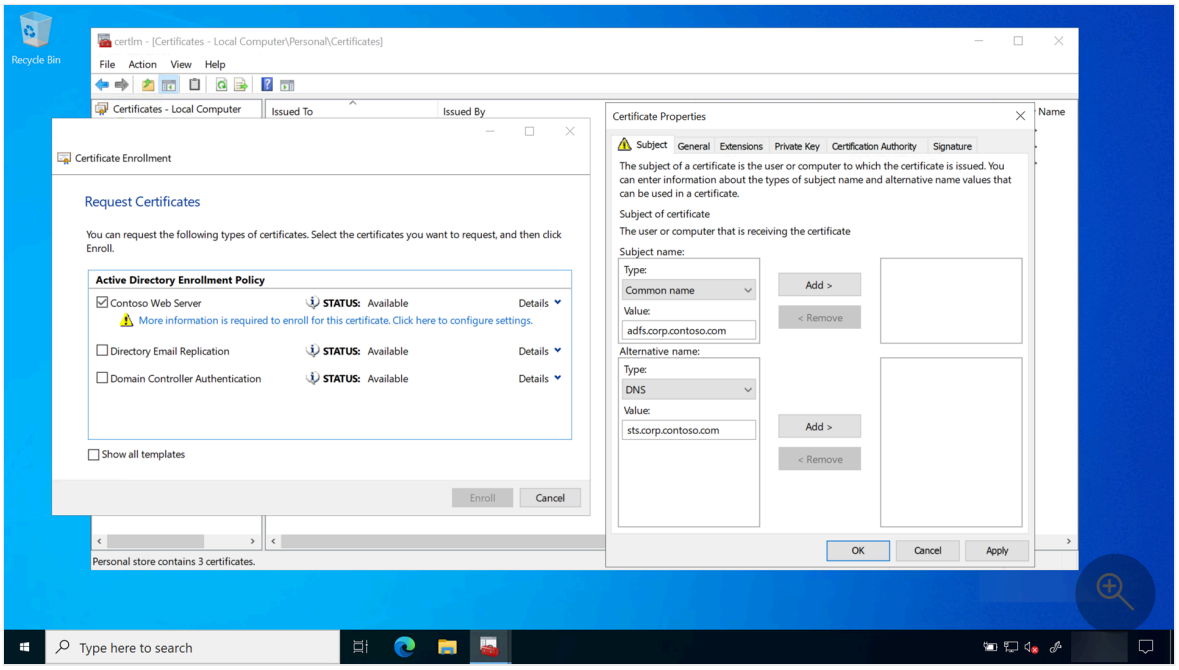
Be sure to enroll or import the certificate into the AD FS server's computer certificate store. Also, ensure all nodes in the farm have the proper TLS server authentication certificate.

AD FS authentication certificate enrollment

Sign-in the federation server with *domain administrator* equivalent credentials.

1. Start the Local Computer **Certificate Manager** (certlm.msc)
2. Expand the **Personal** node in the navigation pane
3. Right-click **Personal**. Select **All Tasks** > **Request New Certificate**
4. Select **Next** on the **Before You Begin** page
5. Select **Next** on the **Select Certificate Enrollment Policy** page
6. On the **Request Certificates** page, select the **Internal Web Server** check box

7. Select the ⚠ **More information is required to enroll for this certificate. Click here to configure settings** link



8. Under **Subject name**, select **Common Name** from the **Type** list. Type the FQDN of the computer hosting the AD FS role and then select **Add**
9. Under **Alternative name**, select **DNS** from the **Type** list. Type the FQDN of the name that you will use for your federation services (*sts.corp.contoso.com*). The name you use here MUST match the name you use when configuring the AD FS server role. Select **Add** and **OK** when finished
10. Select **Enroll**

A server authentication certificate should appear in the computer's personal certificate store.

Deploy the AD FS role

> ⓘ **Important**
>
> Finish the entire AD FS configuration on the first server in the farm before adding the second server to the AD FS farm. Once complete, the second server receives the configuration through the shared configuration database when it is added the AD FS farm.

Sign-in the federation server with *Enterprise Administrator* equivalent credentials.

1. Start **Server Manager**. Select **Local Server** in the navigation pane
2. Select **Manage** > **Add Roles and Features**
3. Select **Next** on the **Before you begin** page

4. On the **Select installation type** page, select **Role-based or feature-based installation** > **Next**
5. On the **Select destination server** page, choose **Select a server from the server pool**. Select the federation server from the **Server Pool** list and **Next**
6. On the **Select server roles** page, select **Active Directory Federation Services** and **Next**
7. Select **Next** on the **Select features** page
8. Select **Next** on the **Active Directory Federation Service** page
9. Select **Install** to start the role installation

Review to validate the AD FS deployment

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Confirm the AD FS farm uses the correct database configuration
- ✓ Confirm the AD FS farm has an adequate number of nodes and is properly load balanced for the anticipated load
- ✓ Confirm **all** AD FS servers in the farm have the latest updates installed
- ✓ Confirm all AD FS servers have a valid server authentication certificate

Device registration service account prerequisites

The use of Group Managed Service Accounts (GMSA) is the preferred way to deploy service accounts for services that support them. GMSAs have security advantages over normal user accounts because Windows handles password management. This means the password is long, complex, and changes periodically. AD FS supports GMSAs, and it should be configured using them for additional security.

GSMA uses the *Microsoft Key Distribution Service* that is located on the domain controllers. Before you can create a GSMA, you must first create a root key for the service. You can skip this if your environment already uses GSMA.

Create KDS Root Key

Sign-in a domain controller with *Enterprise Administrator* equivalent credentials.

Start an elevated PowerShell console and execute the following command:

```
PowerShell
```

```
Add-KdsRootKey -EffectiveTime (Get-Date).AddHours(-10)
```

Configure the Active Directory Federation Service Role

Use the following procedures to configure AD FS.

Sign-in to the federation server with *Domain Administrator* equivalent credentials. These procedures assume you are configuring the first federation server in a federation server farm.

1. Start **Server Manager**
2. Select the notification flag in the upper right corner and select **Configure the federation services on this server**
3. On the **Welcome** page, select **Create the first federation server farm** > **Next**
4. On the **Connect to Active Directory Domain Services** page, select **Next**
5. On the **Specify Service Properties** page, select the recently enrolled or imported certificate from the **SSL Certificate** list. The certificate is likely named after your federation service, such as *sts.corp.contoso.com*
6. Select the federation service name from the **Federation Service Name** list
7. Type the *Federation Service Display Name* in the text box. This is the name users see when signing in. Select **Next**
8. On the **Specify Service Account** page, select **Create a Group Managed Service Account**. In the **Account Name** box, type *adfssvc*
9. On the **Specify Configuration Database** page, select **Create a database on this server using Windows Internal Database** and select **Next**
10. On the **Review Options** page, select **Next**
11. On the **Pre-requisite Checks** page, select **Configure**
12. When the process completes, select **Close**

Add the AD FS service account to the *Key Admins* group

During Windows Hello for Business enrollment, the public key is registered in an attribute of the user object in Active Directory. To ensure that the AD FS service can add and remove keys are part of its normal workflow, it must be a member of the *Key Admins* global group.

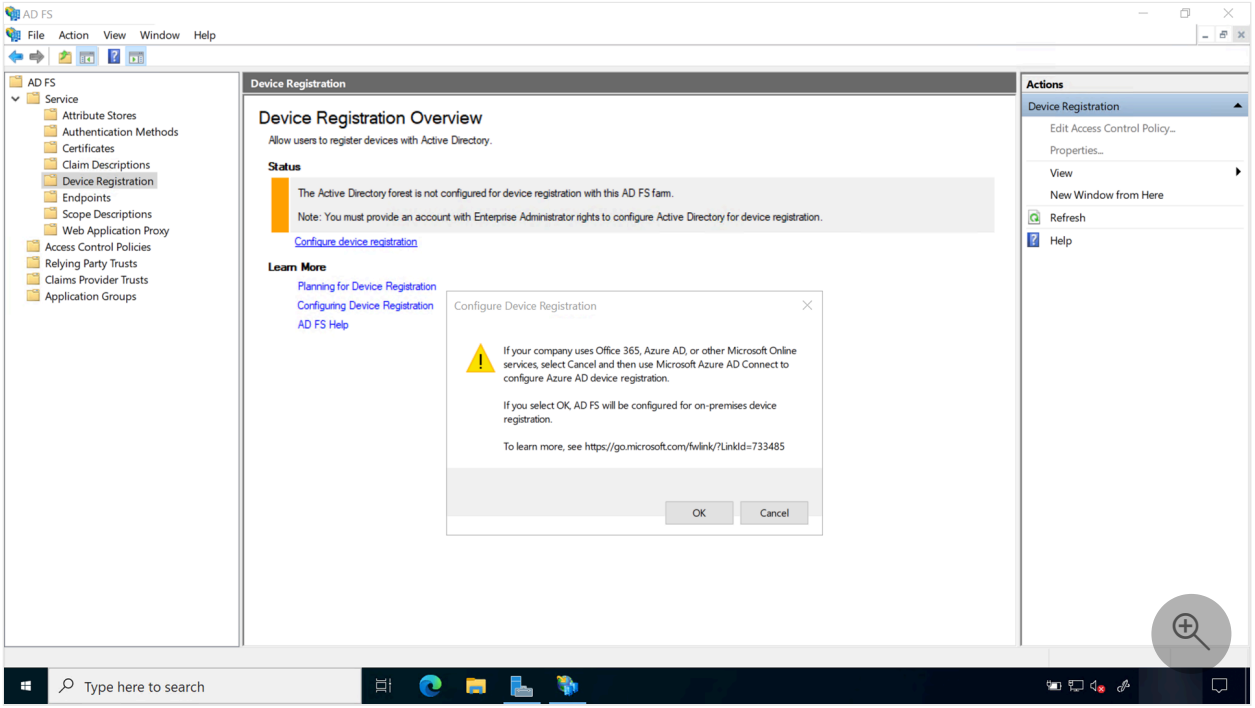
Sign-in to a domain controller or management workstation with *Domain Administrator* equivalent credentials.

1. Open **Active Directory Users and Computers**
2. Select the **Users** container in the navigation pane
3. Right-click **Key Admins** in the details pane and select **Properties**
4. Select the **Members** > **Add...**
5. In the **Enter the object names to select** text box, type *adfssvc*. Select **OK**
6. Select **OK** to return to **Active Directory Users and Computers**
7. Change to server hosting the AD FS role and restart it

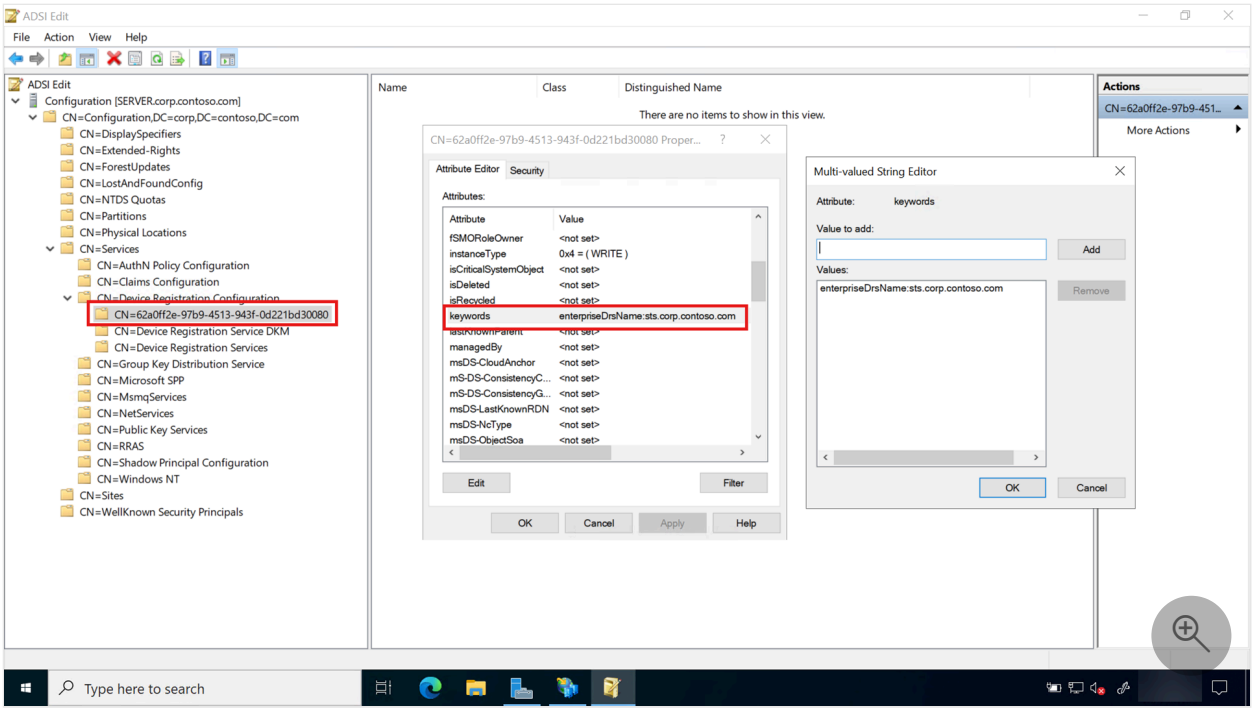
Configure the device registration service

Sign-in to the federation server with *Enterprise Administrator* equivalent credentials. These instructions assume you are configuring the first federation server in a federation server farm.

1. Open the **AD FS management** console
2. In the navigation pane, expand **Service**. Select **Device Registration**
3. In the details pane, select **Configure device registration**
4. In the **Configure Device Registration** dialog, Select **OK**



Triggering device registration from AD FS, creates the service connection point (SCP) in the Active Directory configuration partition. The SCP is used to store the device registration information that Windows clients will automatically discover.



Review to validate the AD FS and Active Directory configuration

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Record the information about the AD FS certificate, and set a renewal reminder at least six weeks before it expires. Relevant information includes: certificate serial number, thumbprint, common name, subject alternate name, name of the physical host server, the issued date, the expiration date, and issuing CA vendor (if a non-Microsoft certificate)
- ✓ Confirm you added the AD FS service account to the KeyAdmins group
- ✓ Confirm you enabled the Device Registration service

Additional federation servers

Organizations should deploy more than one federation server in their federation farm for high-availability. You should have a minimum of two federation services in your AD FS farm, however most organizations are likely to have more. This largely depends on the number of devices and users using the services provided by the AD FS farm.

Server authentication certificate

Each server you add to the AD FS farm must have a proper server authentication certificate. Refer to the Enroll for a TLS Server Authentication Certificate section of this

document to determine the requirements for your server authentication certificate. As previously stated, AD FS servers used exclusively for on-premises deployments of Windows Hello for Business can use enterprise server authentication certificates rather than server authentication certificates issued by public certificate authorities.

Install additional servers

Adding federation servers to the existing AD FS farm begins with ensuring the server are fully patched, to include Windows Server 2016 Update needed to support Windows Hello for Business deployments (https://aka.ms/whfbadfs1703). Next, install the Active Directory Federation Service role on the additional servers and then configure the server as an additional server in an existing farm.

Load balance AD FS

Many environments load balance using hardware devices. Environments without hardware load-balancing capabilities can take advantage the network load-balancing feature included in Windows Server to load balance the AD FS servers in the federation farm. Install the Windows Network Load Balancing feature on all nodes participating in the AD FS farm that should be load balanced.

Install Network Load Balancing Feature on AD FS Servers

Sign-in the federation server with *Enterprise Administrator* equivalent credentials.

1. Start **Server Manager**. Select **Local Server** in the navigation pane
2. Select **Manage** and then select **Add Roles and Features**
3. Select **Next** On the **Before you begin** page
4. On the **Select installation type** page, select **Role-based or feature-based installation** and select **Next**
5. On the **Select destination server** page, choose **Select a server from the server pool**. Select the federation server from the **Server Pool** list. Select **Next**
6. On the **Select server roles** page, select **Next**
7. Select **Network Load Balancing** on the **Select features** page
8. Select **Install** to start the feature installation

Configure Network Load Balancing for AD FS

Before you can load balance all the nodes in the AD FS farm, you must first create a new load balance cluster. Once you have created the cluster, then you can add new nodes to

that cluster.

Sign-in a node of the federation farm with *Administrator* equivalent credentials.

1. Open **Network Load Balancing Manager** from **Administrative Tools**
2. Right-click **Network Load Balancing Clusters**, and then select **New Cluster**
3. To connect to the host that is to be a part of the new cluster, in the **Host** text box, type the name of the host, and then select **Connect**
4. Select the interface that you want to use with the cluster, and then select **Next** (the interface hosts the virtual IP address and receives the client traffic to load balance)
5. In **Host Parameters**, select a value in **Priority (Unique host identifier)**. This parameter specifies a unique ID for each host. The host with the lowest numerical priority among the current members of the cluster handles all of the cluster's network traffic that is not covered by a port rule. Select **Next**
6. In **Cluster IP Addresses**, select **Add** and type the cluster IP address that is shared by every host in the cluster. NLB adds this IP address to the TCP/IP stack on the selected interface of all hosts that are chosen to be part of the cluster. Select **Next**
7. In **Cluster Parameters**, select values in **IP Address** and **Subnet mask** (for IPv6 addresses, a subnet mask value is not needed). Type the full Internet name that users will use to access this NLB cluster
8. In **Cluster operation mode**, select **Unicast** to specify that a unicast media access control (MAC) address should be used for cluster operations. In unicast mode, the MAC address of the cluster is assigned to the network adapter of the computer, and the built-in MAC address of the network adapter is not used. We recommend that you accept the unicast default settings. Select **Next**
9. In Port Rules, select Edit to modify the default port rules to use port 443

Additional AD FS Servers

1. To add more hosts to the cluster, right-click the new cluster, and then select **Add Host to Cluster**
2. Configure the host parameters (including host priority, dedicated IP addresses, and load weight) for the additional hosts by following the same instructions that you used to configure the initial host. Because you are adding hosts to an already configured cluster, all the cluster-wide parameters remain the same

Configure DNS for Device Registration

Sign-in the domain controller or administrative workstation with domain administrator equivalent credentials.
You'll need the *federation service* name to complete this task. You can view the

federation service name by selecting **Edit Federation Service Properties** from the **Action** pan of the **AD FS** management console, or by using `(Get-AdfsProperties).Hostname.` (PowerShell) on the AD FS server.

1. Open the **DNS Management** console
2. In the navigation pane, expand the domain controller name node and **Forward Lookup Zones**
3. In the navigation pane, select the node that has the name of your internal Active Directory domain name
4. In the navigation pane, right-click the domain name node and select **New Host (A or AAAA)**
5. In the **name** box, type the name of the federation service. In the **IP address** box, type the IP address of your federation server. Select **Add Host**
6. Right-click the `<domain_name>` node and select **New Alias (CNAME)**
7. In the **New Resource Record** dialog box, type `enterpriseregistration` in the **Alias** name box
8. In the **fully qualified domain name (FQDN)** of the target host box, type `federation_service_farm_name.<domain_name_fqdn`, and select OK
9. Close the DNS Management console

> ⓘ **Note**
>
> If your forest has multiple UPN suffixes, please make sure that `enterpriseregistration.<upnsuffix_fqdn>` is present for each suffix.

Configure the Intranet Zone to include the federation service

The Windows Hello provisioning presents web pages from the federation service. Configuring the intranet zone to include the federation service enables the user to authenticate to the federation service using integrated authentication. Without this setting, the connection to the federation service during Windows Hello provisioning prompts the user for authentication.

Create an Intranet Zone Group Policy

Sign-in the domain controller or administrative workstation with *Domain Admin* equivalent credentials:

1. Start the **Group Policy Management Console** (`gpmc.msc`)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type **Intranet Zone Settings** in the name box and select **OK**
5. In the content pane, right-click the **Intranet Zone Settings** Group Policy object and select **Edit**
6. In the navigation pane, expand **Policies** under **Computer Configuration**
7. Expand **Administrative Templates** > **Windows Component** > **Internet Explorer** > **Internet Control Panel** >**Security Page**. Open **Site to Zone Assignment List**
8. Select **Enable** > **Show**. In the **Value Name** column, type the url of the federation service beginning with https. In the **Value** column, type the number **1**. Select OK twice, then close the Group Policy Management Editor

Deploy the Intranet Zone Group Policy object

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and right-click the node that has your Active Directory domain name and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select **Intranet Zone Settings** or the name of the Windows Hello for Business Group Policy object you previously created and select **OK**

Validate and deploy multifactor authentication (MFA)

Windows Hello for Business requires users perform multifactor authentication (MFA) prior to enroll in the service. On-premises deployments can use, as MFA option:

- certificates

  > ⓘ **Note**
  >
  > When using this option, the certificates must be deployed to the users. For example, users can use their smart card or virtual smart card as a certificate authentication option.

- non-Microsoft authentication providers for AD FS
- custom authentication provider for AD FS

> **Important**
>
> As of July 1, 2019, Microsoft will no longer offer MFA Server for new deployments. New customers who would like to require multifactor authentication from their users should use cloud-based Microsoft Entra multifactor authentication. Existing customers who have activated MFA Server prior to July 1 will be able to download the latest version, future updates and generate activation credentials as usual.

For information on available non-Microsoft authentication methods see Configure Additional Authentication Methods for AD FS. For creating a custom authentication method see Build a Custom Authentication Method for AD FS in Windows Server

Follow the integration and deployment guide for the authentication provider you select to integrate and deploy it to AD FS. Make sure that the authentication provider is selected as a multifactor authentication option in the AD FS authentication policy. For information on configuring AD FS authentication policies see Configure Authentication Policies.

> **Tip**
>
> When you validate the AD FS configuration, verify if you need to update the configuration of user agent strings to support Windows Integrated Authentication (WIA). For more information, see **Change WIASupportedUserAgent settings**.

Review to validate the configuration

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Confirm all AD FS servers have a valid server authentication certificate. The subject of the certificate is the common name (FQDN) of the host or a wildcard name. The alternate name of the certificate contains a wildcard or the FQDN of the federation service
- ✓ Confirm the AD FS farm has an adequate number of nodes and is properly load balanced for the anticipated load
- ✓ Confirm you restarted the AD FS service
- ✓ Confirm you created a DNS A Record for the federation service and the IP address used is the load-balanced IP address
- ✓ Confirm you created and deployed the Intranet Zone settings to prevent double authentication to the federation server

✓ Confirm you have deployed a MFA solution for AD FS

Next: configure and enroll in Windows Hello for Business >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure and enroll in Windows Hello for Business in an on-premises key trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises ⓘ
- **Trust type:** key trust ⓘ
- **Join type:** domain join ⓘ

Once the prerequisites are met, and the PKI and AD FS configurations are validated, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Configure Windows Hello for Business policy settings

There's 1 policy setting required to enable Windows Hello for Business in a key trust model:

- Use Windows Hello for Business

Another optional, but recommended, policy setting is:

- Use a hardware security device

You can configure the Use Windows Hello for Business policy setting in the computer or user node of a GPO:

- Deploying the computer node policy setting, results in all users that sign-in to the targeted devices to attempt a Windows Hello for Business enrollment
- Deploying the user node policy setting, results in only the targeted users to attempt a Windows Hello for Business enrollment

If both user and computer policy settings are deployed, the user policy setting has precedence.

To configure a device with group policy, use the Local Group Policy Editor. To configure multiple devices joined to Active Directory, create or edit a group policy object (GPO) and use the following settings:

Expand table

Group policy path	Group policy setting	Value
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business or **User Configuration\Administrative Templates\Windows Components\Windows Hello for Business**	Use Windows Hello for Business	**Enabled**
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business	Use a hardware security device	**Enabled**

Group policies can be linked to domains or organizational units, filtered using security groups, or filtered using WMI filters.

> **Tip**
>
> The best way to deploy the Windows Hello for Business GPO is to use security group filtering. Only members of the targeted security group will provision Windows Hello for Business, enabling a phased rollout. This solution allows linking the GPO to the domain, ensuring the GPO is scoped to all security principals. The security group filtering ensures that only the members of the global group receive and apply the GPO, which results in the provisioning of Windows Hello for Business.

Additional policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after the user profile is loaded and before the user receives their desktop. For the provisioning process to begin, all prerequisite checks must pass.

You can determine the status of the prerequisite checks by viewing the **User Device Registration** admin log under **Applications and Services Logs** > **Microsoft** > **Windows**. This information is also available using the `dsregcmd.exe /status` command from a console. For more information, see dsregcmd.

User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**
3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use their PIN to sign-in. The user may close the provisioning application and access their desktop

The following video shows the Windows Hello for Business enrollment steps after signing in with a password, using a custom MFA adapter for AD FS.
https://learn-video.azurefd.net/vod/player?id=771165c0-e37f-4f9d-9e21-4f383cc6590d&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fon-premises-key-trust-enroll

Sequence diagram

To better understand the provisioning flows, review the following sequence diagram:

- Provisioning in an on-premises key trust deployment model

Feedback

Was this page helpful? Yes No

Provide product feedback

On-premises certificate trust deployment guide

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises
- **Trust type:** certificate trust
- **Join type:** domain join

Requirements

Before starting the deployment, review the requirements described in the Plan a Windows Hello for Business Deployment article.

Ensure that the following requirements are met before you begin:

- ✓ Public Key Infrastructure
- ✓ Authentication
- ✓ Device configuration
- ✓ Licensing for cloud services
- ✓ Windows requirements
- ✓ Windows Server requirements
- ✓ Prepare users to use Windows Hello

Deployment steps

Once the prerequisites are met, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure and validate the Public Key Infrastructure
- ✓ Prepare and deploy AD FS with MFA
- ✓ Configure and enroll in Windows Hello for Business

Configure and validate the Public Key Infrastructure

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises
- **Trust type:** certificate trust
- **Join type:** domain join

Windows Hello for Business must have a Public Key Infrastructure (PKI) when using the *key trust* or *certificate trust* models. The domain controllers must have a certificate, which serves as a root of trust for clients. The certificate ensures that clients don't communicate with rogue domain controllers. The certificate trust model extends certificate issuance to client computers. During Windows Hello for Business provisioning, the user receives a sign-in certificate.

Deploy an enterprise certification authority

This guide assumes most enterprises have an existing public key infrastructure. Windows Hello for Business depends on an enterprise PKI running the Windows Server *Active Directory Certificate Services* role.
If you don't have an existing PKI, review Certification Authority Guidance to properly design your infrastructure. Then, consult the Test Lab Guide: Deploying an AD CS Two-Tier PKI Hierarchy for instructions on how to configure your PKI using the information from your design session.

Lab-based PKI

The following instructions may be used to deploy simple public key infrastructure that is suitable **for a lab environment**.

Sign in using *Enterprise Administrator* equivalent credentials on a Windows Server where you want the certification authority (CA) installed.

> **Note**
>
> Never install a certification authority on a domain controller in a production environment.

1. Open an elevated Windows PowerShell prompt
2. Use the following command to install the Active Directory Certificate Services role.

PowerShell

```
Add-WindowsFeature Adcs-Cert-Authority -IncludeManagementTools
```

3. Use the following command to configure the CA using a basic certification authority configuration

PowerShell

```
Install-AdcsCertificationAuthority
```

Configure the enterprise PKI

Configure domain controller certificates

Clients must trust the domain controllers, and the best way to enable the trust is to ensure that each domain controller has a *Kerberos Authentication* certificate. Installing a certificate on the domain controllers enables the Key Distribution Center (KDC) to prove its identity to other members of the domain. The certificates provide clients a root of trust external to the domain, namely the *enterprise certification authority*.

Domain controllers automatically request a *domain controller certificate* (if published) when they discover an enterprise CA is added to Active Directory. The certificates based on the *Domain Controller* and *Domain Controller Authentication* certificate templates don't include the *KDC Authentication* object identifier (OID), which was later added to the Kerberos RFC. Therefore, domain controllers need to request a certificate based on the *Kerberos Authentication* certificate template.

By default, the Active Directory CA provides and publishes the *Kerberos Authentication* certificate template. The cryptography configuration included in the template is based on older and less performant cryptography APIs. To ensure domain controllers request the proper certificate with the best available cryptography, use the *Kerberos Authentication* certificate template as a *baseline* to create an updated domain controller certificate template.

> **ⓘ Important**
>
> The certificates issued to the domain controllers must meet the following requirements:

- The *Certificate Revocation List (CRL) distribution point* extension must point to a valid CRL, or an *Authority Information Access (AIA)* extension that points to an Online Certificate Status Protocol (OCSP) responder
- Optionally, the certificate *Subject* section could contain the directory path of the server object (the distinguished name)
- The certificate *Key Usage* section must contain *Digital Signature* and *Key Encipherment*
- Optionally, the certificate *Basic Constraints* section should contain: `[Subject Type=End Entity, Path Length Constraint=None]`
- The certificate *extended key usage* section must contain Client Authentication (`1.3.6.1.5.5.7.3.2`), Server Authentication (`1.3.6.1.5.5.7.3.1`), and KDC Authentication (`1.3.6.1.5.2.3.5`)
- The certificate *Subject Alternative Name* section must contain the Domain Name System (DNS) name
- The certificate template must have an extension that has the value `DomainController`, encoded as a **BMPstring**. If you are using Windows Server Enterprise Certificate Authority, this extension is already included in the domain controller certificate template
- The domain controller certificate must be installed in the local computer's certificate store

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Kerberos Authentication** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Domain Controller Authentication (Kerberos)* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **None** from the **Subject name format** list • Select **DNS name** from the **Include this information in alternate subject** list • Clear all other items
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Supersede existing domain controller certificates

The domain controllers may have an existing domain controller certificate. The Active Directory Certificate Services provides a default certificate template for domain controllers called *domain controller certificate*. Later releases of Windows Server provided a new certificate template called *domain controller authentication certificate*. These certificate templates were provided prior to the update of the Kerberos specification that stated Key Distribution Centers (KDCs) performing certificate authentication needed to include the *KDC Authentication* extension.

The *Kerberos Authentication* certificate template is the most current certificate template designated for domain controllers, and should be the one you deploy to all your domain controllers.
The *autoenrollment* feature allows you to replace the domain controller certificates. Use

the following configuration to replace older domain controller certificates with new ones, using the *Kerberos Authentication* certificate template.

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the *Domain Controller Authentication (Kerberos)* (or the name of the certificate template you created in the previous section) template in the details pane and select **Properties**
4. Select the **Superseded Templates** tab. Select **Add**
5. From the **Add Superseded Template** dialog, select the *Domain Controller* certificate template and select **OK** > **Add**
6. From the **Add Superseded Template** dialog, select the *Domain Controller Authentication* certificate template and select **OK**
7. From the **Add Superseded Template** dialog, select the *Kerberos Authentication* certificate template and select **OK**
8. Add any other enterprise certificate templates that were previously configured for domain controllers to the **Superseded Templates** tab
9. Select **OK** and close the **Certificate Templates** console

The certificate template is configured to supersede all the certificate templates provided in the *superseded templates* list.
However, the certificate template and the superseding of certificate templates isn't active until the template is published to one or more certificate authorities.

> ⓘ **Note**
>
> The domain controller's certificate must chain to a root in the NTAuth store. By default, the Active Directory Certificate Authority's root certificate is added to the NTAuth store. If you are using a non-Microsoft CA, this may not be done by default. If the domain controller certificate does not chain to a root in the NTAuth store, user authentication will fail. To see all certificates in the NTAuth store, use the following command:
>
> `Certutil -viewstore -enterprise NTAuth`

Configure an internal web server certificate template

Windows clients communicate with AD FS via HTTPS. To meet this need, a *server authentication* certificate must be issued to all the nodes in the AD FS farm. On-premises deployments can use a *server authentication* certificate issued by the enterprise PKI. A *server authentication* certificate template must be configured, so the AD FS nodes can request a certificate.

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** > **Manage**
3. In the **Certificate Template Console**, right-click the **Web Server** template in the details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *Internal Web Server* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Request Handling	Select **Allow private key to be exported**
Subject Name	Select **Supply in the request**
Security	Add **Domain Computers** with **Enroll** access
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Configure a Windows Hello for Business authentication certificate template

During Windows Hello for Business provisioning, Windows clients request an authentication certificate from AD FS, which requests the authentication certificate on behalf of the user. This task configures the Windows Hello for Business authentication certificate template.

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** and select **Manage**
3. In the **Certificate Template Console**, right-click the **Smartcard Logon** template and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list* • Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *WHFB Authentication* • Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces
Subject Name	• Select **Build from this Active Directory information** • Select **Fully distinguished name** from the **Subject name format** list • Select the **User Principal Name (UPN)** check box under **Include this information in alternative subject name**

Tab Name	Configurations
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**
Extensions	Verify the **Application Policies** extension includes **Smart Card Logon**
Issuance Requirements	• Select the **This number of authorized signatures** check box. Type *1* in the text box • Select **Application policy** from the *Policy type required in signature* • Select **Certificate Request Agent** from in the *Application policy* list • Select the **Valid existing certificate** option
Request Handling	Select the **Renew with same key** check box

Security	• Select **Add**
	• Target an Active Directory security group that contains the users that you want to enroll in Windows Hello for Business. For example, if you have a group called *Window Hello for Business Users*, type it in the **Enter the object names to select** text box and select **OK**
	• Select the **Windows Hello for Business Users** from the **Group or users names** list. In the **Permissions for Windows Hello for Business Users** section:
	  ◦ Select the **Allow** check box for the **Enroll** permission
	  ◦ Excluding the group above (for example, *Window Hello for Business Users*), clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other entries in the **Group or users names** section if the check boxes aren't already cleared
	• Select **OK**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Mark the template as the Windows Hello Sign-in template

Sign in to a CA or management workstations with *Enterprise Administrator* equivalent credentials

Open an elevated command prompt end execute the following command

```
Windows Command Prompt
```

```
certutil.exe -dsTemplate WHFBAuthentication msPKI-Private-Key-Flag
+CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY
```

If the template was changed successfully, the output of the command will contain old and new values of the template parameters. The new value must contain the `CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY` parameter. Example:

Windows Command Prompt

```
CN=Certificate Templates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=[yourdomain]:WHFBAuthentication

Old Value:
msPKI-Private-Key-Flag REG_DWORD = 5050080 (84213888)
CTPRIVATEKEY_FLAG_REQUIRE_SAME_KEY_RENEWAL -- 80 (128)
CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0
TEMPLATE_SERVER_VER_WINBLUE<<CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT -- 50000
(327680)
TEMPLATE_CLIENT_VER_WINBLUE<<CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT --
5000000 (83886080)

New Value:
msPKI-Private-Key-Flag REG_DWORD = 5250080 (86311040)
CTPRIVATEKEY_FLAG_REQUIRE_SAME_KEY_RENEWAL -- 80 (128)
CTPRIVATEKEY_FLAG_ATTEST_NONE -- 0
TEMPLATE_SERVER_VER_WINBLUE<<CTPRIVATEKEY_FLAG_SERVERVERSION_SHIFT -- 50000
(327680)
CTPRIVATEKEY_FLAG_HELLO_LOGON_KEY -- 200000 (2097152)
TEMPLATE_CLIENT_VER_WINBLUE<<CTPRIVATEKEY_FLAG_CLIENTVERSION_SHIFT --
5000000 (83886080)
CertUtil: -dsTemplate command completed successfully."
```

Note

If you gave your Windows Hello for Business Authentication certificate template a different name, then replace `WHFBAuthentication` in the above command with the name of your certificate template. It's important that you use the template name rather than the template display name. You can view the template name on the **General** tab of the certificate template using the Certificate Template management console (certtmpl.msc).

Unpublish Superseded Certificate Templates

The certification authority only issues certificates based on published certificate templates. For security, it's a good practice to unpublish certificate templates that the

CA isn't configured to issue, including the pre-published templates from the role installation and any superseded templates.

The newly created *domain controller authentication* certificate template supersedes previous domain controller certificate templates. Therefore, you need to unpublish these certificate templates from all issuing certificate authorities.

Sign in to the CA or management workstation with *Enterprise Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane > **Certificate Templates**
3. Right-click the *Domain Controller* certificate template and select **Delete**. Select **Yes** on the **Disable certificate templates** window
4. Repeat step 3 for the *Domain Controller Authentication* and *Kerberos Authentication* certificate templates

Publish certificate templates to the CA

A certification authority can only issue certificates for certificate templates that are published to it. If you have more than one CA, and you want more CAs to issue certificates based on the certificate template, then you must publish the certificate template to them.

Sign in to the CA or management workstations with **Enterprise Admin** equivalent credentials.

1. Open the **Certification Authority** management console
2. Expand the parent node from the navigation pane
3. Select **Certificate Templates** in the navigation pane
4. Right-click the **Certificate Templates** node. Select **New** > **Certificate Template** to issue
5. In the **Enable Certificates Templates** window, select the *Domain Controller Authentication (Kerberos)*, *Internal Web Server*, and *WHFB Authentication* templates you created in the previous steps. Select **OK** to publish the selected certificate templates to the certification authority
6. If you published the *Domain Controller Authentication (Kerberos)* certificate template, then unpublish the certificate templates you included in the superseded templates list

    - To unpublish a certificate template, right-click the certificate template you want to unpublish and select **Delete**. Select **Yes** to confirm the operation

7. Close the console

Configure and deploy certificates to domain controllers

Configure automatic certificate enrollment for the domain controllers

Domain controllers automatically request a certificate from the *Domain controller certificate* template. However, domain controllers are unaware of newer certificate templates or superseded configurations on certificate templates. For domain controllers to automatically enroll and renew of certificates, configure a GPO for automatic certificate enrollment, and link it to the *Domain Controllers* OU.

1. Open the **Group Policy Management Console** (gpmc.msc)
2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type *Domain Controller Auto Certificate Enrollment* in the name box and select **OK**
5. Right-click the **Domain Controller Auto Certificate Enrollment** Group Policy object and select **Edit**
6. In the navigation pane, expand **Policies** under **Computer Configuration**
7. Expand **Windows Settings** > **Security Settings** > **Public Key Policies**
8. In the details pane, right-click **Certificate Services Client - Auto-Enrollment** and select **Properties**
9. Select **Enabled** from the **Configuration Model** list
10. Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** check box
11. Select the **Update certificates that use certificate templates** check box
12. Select **OK**
13. Close the **Group Policy Management Editor**

Deploy the domain controller auto certificate enrollment GPO

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Start the **Group Policy Management Console** (gpmc.msc)

2. In the navigation pane, expand the domain and expand the node with the Active Directory domain name. Right-click the **Domain Controllers** organizational unit and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select *Domain Controller Auto Certificate Enrollment* or the name of the domain controller certificate enrollment Group Policy object you previously created
4. Select **OK**

Validate the configuration

Windows Hello for Business is a distributed system, which on the surface appears complex and difficult. The key to a successful deployment is to validate phases of work prior to moving to the next phase.

Confirm your domain controllers enroll the correct certificates and not any superseded certificate templates. Check that each domain controller completed the certificate autoenrollment.

Use the event logs

Sign in to domain controller or management workstations with *Domain Administrator* equivalent credentials.

1. Using the Event Viewer, navigate to the **Application and Services** > **Microsoft** > **Windows** > **CertificateServices-Lifecycles-System** event log
2. Look for an event indicating a new certificate enrollment (autoenrollment):

    - The details of the event include the certificate template on which the certificate was issued
    - The name of the certificate template used to issue the certificate should match the certificate template name included in the event
    - The certificate thumbprint and EKUs for the certificate are also included in the event
    - The EKU needed for proper Windows Hello for Business authentication is Kerberos Authentication, in addition to other EKUs provide by the certificate template

Certificates superseded by your new domain controller certificate generate an *archive event* in the Event Log. The archive event contains the certificate template name and thumbprint of the certificate that was superseded by the new certificate.

Certificate Manager

You can use the Certificate Manager console to validate the domain controller has the properly enrolled certificate based on the correct certificate template with the proper EKUs. Use **certlm.msc** to view certificate in the local computers certificate stores. Expand the **Personal** store and view the certificates enrolled for the computer. Archived certificates don't appear in Certificate Manager.

Certutil.exe

You can use `certutil.exe` command to view enrolled certificates in the local computer. Certutil shows enrolled and archived certificates for the local computer. From an elevated command prompt, run the following command:

Windows Command Prompt

```
certutil.exe -q -store my
```

To view detailed information about each certificate in the store, and to validate automatic certificate enrollment enrolled the proper certificates, use the following command:

Windows Command Prompt

```
certutil.exe -q -v -store my
```

Troubleshooting

Windows triggers automatic certificate enrollment for the computer during boot, and when Group Policy updates. You can refresh Group Policy from an elevated command prompt using `gpupdate.exe /force`.

Alternatively, you can forcefully trigger automatic certificate enrollment using `certreq.exe -autoenroll -q` from an elevated command prompt.

Use the event logs to monitor certificate enrollment and archive. Review the configuration, such as publishing certificate templates to issuing certification authority and the *allow* auto enrollment permissions.

Section review and next steps

Before moving to the next section, ensure the following steps are complete:

- ✓ Configure domain controller and web server certificate templates
- ✓ Supersede existing domain controller certificates
- ✓ Unpublish superseded certificate templates
- ✓ Publish the certificate templates to the CA
- ✓ Deploy certificates to the domain controllers
- ✓ Validate the domain controllers configuration

Next: prepare and deploy AD FS >

Feedback

Was this page helpful? Yes No

Provide product feedback

Prepare and deploy Active Directory Federation Services - on-premises certificate trust

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises
- **Trust type:** certificate trust
- **Join type:** domain join

Windows Hello for Business works exclusively with the Active Directory Federation Service (AD FS) role included with Windows Server. The on-premises certificate trust deployment model uses AD FS for *certificate enrollment* (CRA) and *device registration*.

The following guidance describes the deployment of a new instance of AD FS using the Windows Information Database (WID) as the configuration database.
WID is ideal for environments with no more than **30 federation servers** and no more than **100 relying party trusts**. If your environment exceeds either of these factors, or needs to provide *SAML artifact resolution*, *token replay detection*, or needs AD FS to operate as a federated provider role, then the deployment requires the use of SQL as a configuration database.
To deploy AD FS using SQL as its configuration database, review the Deploying a Federation Server Farm checklist.

A new AD FS farm should have a minimum of two federation servers for proper load balancing, which can be accomplished with external networking peripherals, or with using the Network Load Balancing Role included in Windows Server.

Prepare the AD FS deployment by installing and **updating** two Windows Servers.

Enroll for a TLS server authentication certificate

Typically, a federation service is an edge facing role. However, the federation services and instance used with the on-premises deployment of Windows Hello for Business does not need Internet connectivity.

The AD FS role needs a *server authentication* certificate for the federation services, and you can use a certificate issued by your enterprise (internal) CA. The server authentication certificate should have the following names included in the certificate, if you are requesting an individual certificate for each node in the federation farm:

- **Subject Name**: the internal FQDN of the federation server
- **Subject Alternate Name**: the federation service name (e.g. *sts.corp.contoso.com*) or an appropriate wildcard entry (e.g. **.corp.contoso.com*)

The federation service name is set when the AD FS role is configured. You can choose any name, but that name must be different than the name of the server or host. For example, you can name the host server *adfs* and the federation service *sts*. In this example, the FQDN of the host is *adfs.corp.contoso.com* and the FQDN of the federation service is *sts.corp.contoso.com*.

You can also issue one certificate for all hosts in the farm. If you chose this option, leave the subject name *blank*, and include all the names in the subject alternate name when creating the certificate request. All names should include the FQDN of each host in the farm and the federation service name.

When creating a wildcard certificate, mark the private key as exportable, so that the same certificate can be deployed across each federation server and web application proxy within the AD FS farm. Note that the certificate must be trusted (chain to a trusted root CA). Once you have successfully requested and enrolled the server authentication certificate on one node, you can export the certificate and private key to a PFX file using the Certificate Manager console. You can then import the certificate on the remaining nodes in the AD FS farm.

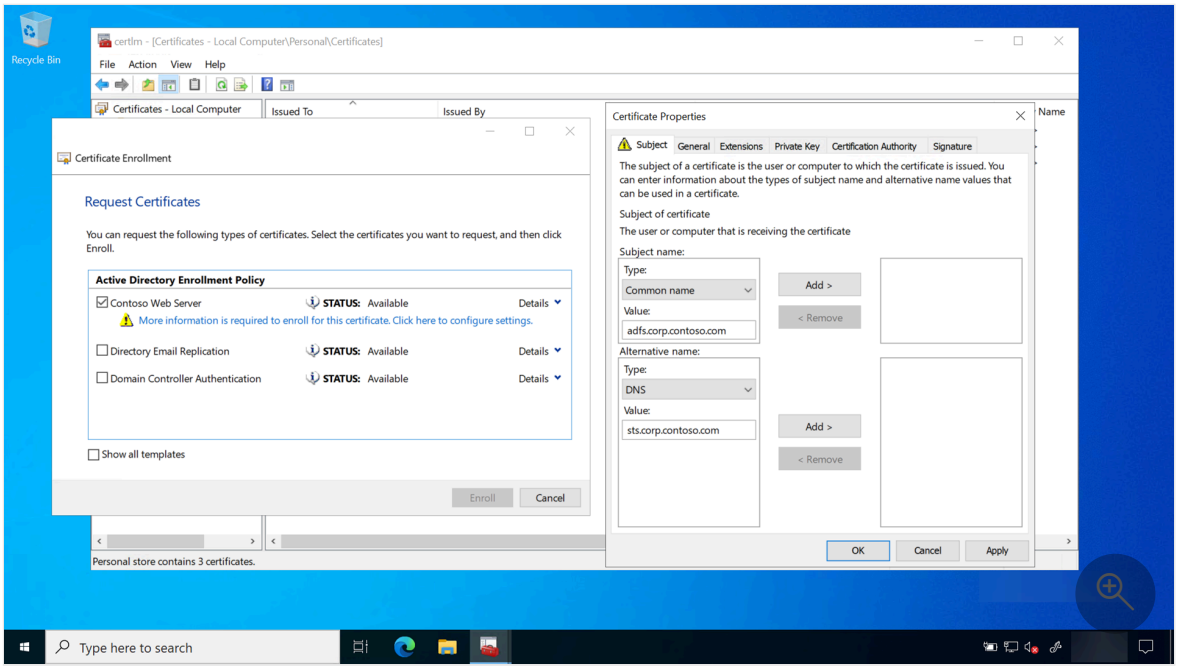
Be sure to enroll or import the certificate into the AD FS server's computer certificate store. Also, ensure all nodes in the farm have the proper TLS server authentication certificate.

AD FS authentication certificate enrollment

Sign-in the federation server with *domain administrator* equivalent credentials.

1. Start the Local Computer **Certificate Manager** (certlm.msc)
2. Expand the **Personal** node in the navigation pane
3. Right-click **Personal**. Select **All Tasks** > **Request New Certificate**
4. Select **Next** on the **Before You Begin** page
5. Select **Next** on the **Select Certificate Enrollment Policy** page
6. On the **Request Certificates** page, select the **Internal Web Server** check box

7. Select the ⚠ **More information is required to enroll for this certificate. Click here to configure settings** link



8. Under **Subject name**, select **Common Name** from the **Type** list. Type the FQDN of the computer hosting the AD FS role and then select **Add**
9. Under **Alternative name**, select **DNS** from the **Type** list. Type the FQDN of the name that you will use for your federation services (*sts.corp.contoso.com*). The name you use here MUST match the name you use when configuring the AD FS server role. Select **Add** and **OK** when finished
10. Select **Enroll**

A server authentication certificate should appear in the computer's personal certificate store.

Deploy the AD FS role

> ⓘ **Important**
>
> Finish the entire AD FS configuration on the first server in the farm before adding the second server to the AD FS farm. Once complete, the second server receives the configuration through the shared configuration database when it is added the AD FS farm.

Sign-in the federation server with *Enterprise Administrator* equivalent credentials.

1. Start **Server Manager**. Select **Local Server** in the navigation pane
2. Select **Manage** > **Add Roles and Features**
3. Select **Next** on the **Before you begin** page

4. On the **Select installation type** page, select **Role-based or feature-based installation** > **Next**
5. On the **Select destination server** page, choose **Select a server from the server pool**. Select the federation server from the **Server Pool** list and **Next**
6. On the **Select server roles** page, select **Active Directory Federation Services** and **Next**
7. Select **Next** on the **Select features** page
8. Select **Next** on the **Active Directory Federation Service** page
9. Select **Install** to start the role installation

Review to validate the AD FS deployment

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Confirm the AD FS farm uses the correct database configuration
- ✓ Confirm the AD FS farm has an adequate number of nodes and is properly load balanced for the anticipated load
- ✓ Confirm **all** AD FS servers in the farm have the latest updates installed
- ✓ Confirm all AD FS servers have a valid server authentication certificate

Device registration service account prerequisites

The use of Group Managed Service Accounts (GMSA) is the preferred way to deploy service accounts for services that support them. GMSAs have security advantages over normal user accounts because Windows handles password management. This means the password is long, complex, and changes periodically. AD FS supports GMSAs, and it should be configured using them for additional security.

GSMA uses the *Microsoft Key Distribution Service* that is located on the domain controllers. Before you can create a GSMA, you must first create a root key for the service. You can skip this if your environment already uses GSMA.

Create KDS Root Key

Sign-in a domain controller with *Enterprise Administrator* equivalent credentials.

Start an elevated PowerShell console and execute the following command:

```
PowerShell
```

```
Add-KdsRootKey -EffectiveTime (Get-Date).AddHours(-10)
```

Configure the Active Directory Federation Service Role

Use the following procedures to configure AD FS.

Sign-in to the federation server with *Domain Administrator* equivalent credentials. These procedures assume you are configuring the first federation server in a federation server farm.

1. Start **Server Manager**
2. Select the notification flag in the upper right corner and select **Configure the federation services on this server**
3. On the **Welcome** page, select **Create the first federation server farm** > **Next**
4. On the **Connect to Active Directory Domain Services** page, select **Next**
5. On the **Specify Service Properties** page, select the recently enrolled or imported certificate from the **SSL Certificate** list. The certificate is likely named after your federation service, such as *sts.corp.contoso.com*
6. Select the federation service name from the **Federation Service Name** list
7. Type the *Federation Service Display Name* in the text box. This is the name users see when signing in. Select **Next**
8. On the **Specify Service Account** page, select **Create a Group Managed Service Account**. In the **Account Name** box, type *adfssvc*
9. On the **Specify Configuration Database** page, select **Create a database on this server using Windows Internal Database** and select **Next**
10. On the **Review Options** page, select **Next**
11. On the **Pre-requisite Checks** page, select **Configure**
12. When the process completes, select **Close**

Add the AD FS service account to the *Key Admins* group

During Windows Hello for Business enrollment, the public key is registered in an attribute of the user object in Active Directory. To ensure that the AD FS service can add and remove keys are part of its normal workflow, it must be a member of the *Key Admins* global group.

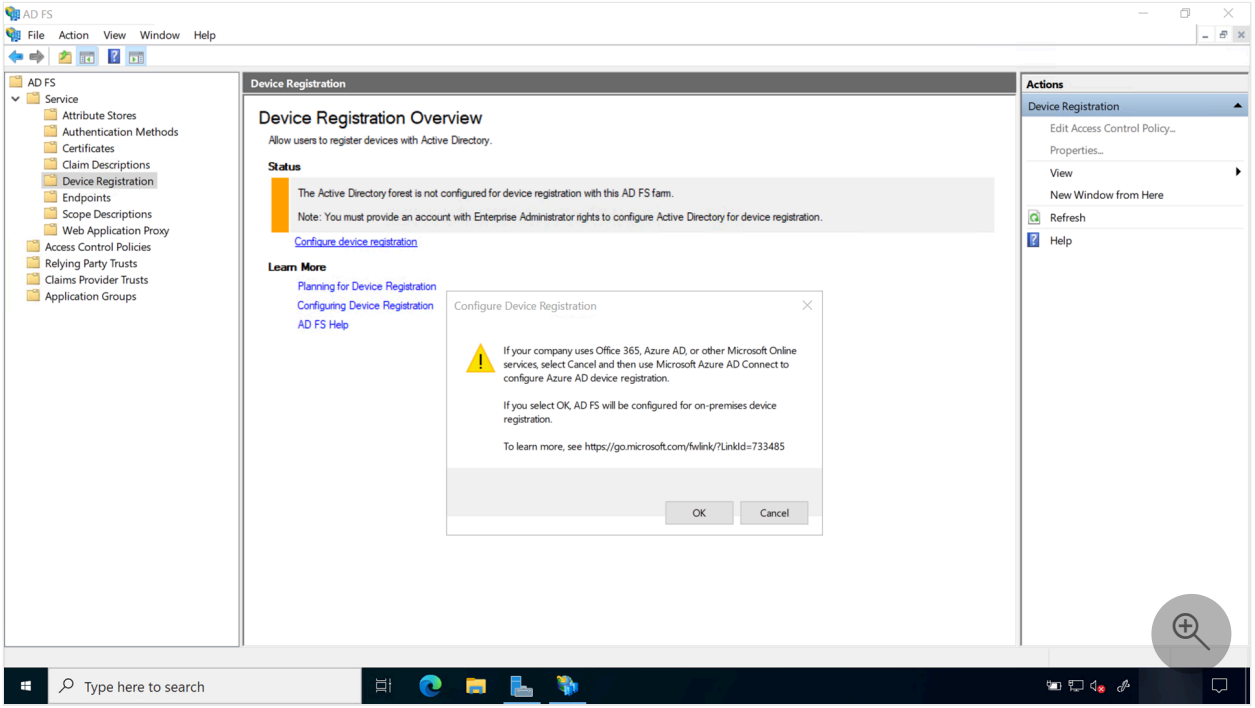
Sign-in to a domain controller or management workstation with *Domain Administrator* equivalent credentials.

1. Open **Active Directory Users and Computers**
2. Select the **Users** container in the navigation pane
3. Right-click **Key Admins** in the details pane and select **Properties**
4. Select the **Members** > **Add...**
5. In the **Enter the object names to select** text box, type *adfssvc*. Select **OK**
6. Select **OK** to return to **Active Directory Users and Computers**
7. Change to server hosting the AD FS role and restart it

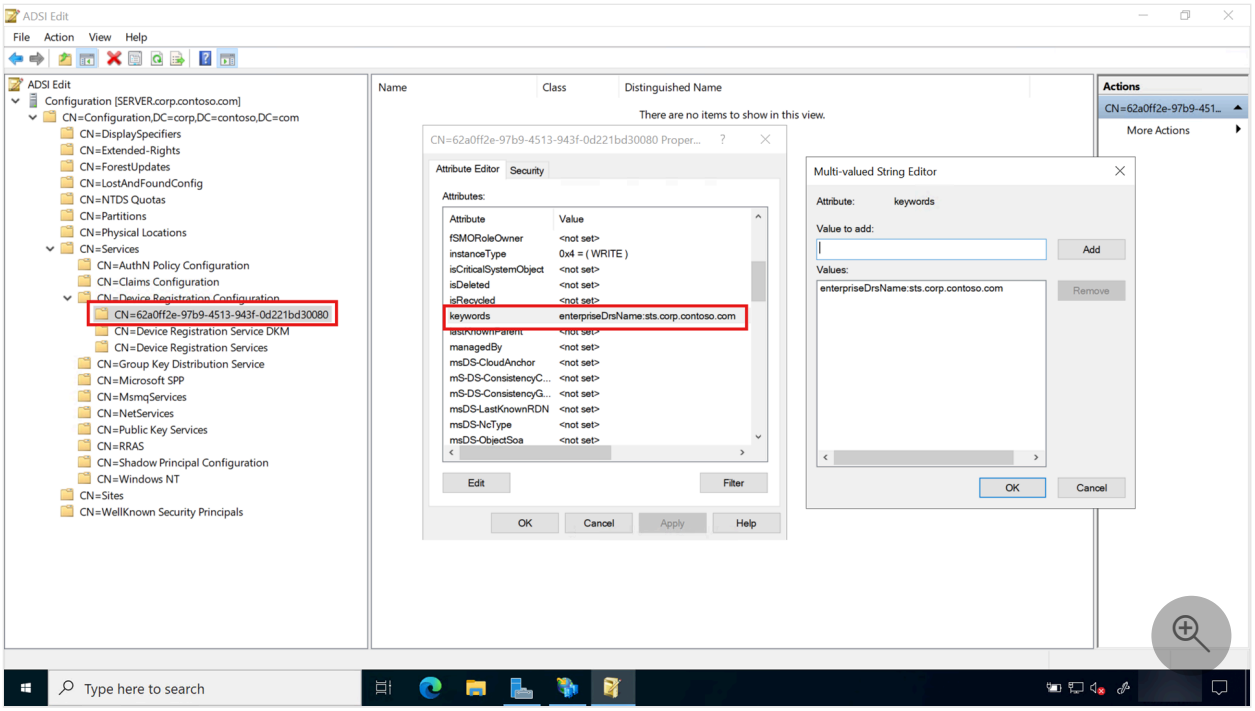
Configure the device registration service

Sign-in to the federation server with *Enterprise Administrator* equivalent credentials. These instructions assume you are configuring the first federation server in a federation server farm.

1. Open the **AD FS management** console
2. In the navigation pane, expand **Service**. Select **Device Registration**
3. In the details pane, select **Configure device registration**
4. In the **Configure Device Registration** dialog, Select **OK**



Triggering device registration from AD FS, creates the service connection point (SCP) in the Active Directory configuration partition. The SCP is used to store the device registration information that Windows clients will automatically discover.



> **Note**
>
> For AD FS 2019 and later in a certificate trust model, a known PRT issue exists. You may encounter this error in AD FS Admin event logs: Received invalid Oauth request. The client 'NAME' is forbidden to access the resource with scope 'ugs'. For more information about the isse and its resolution, see Certificate trust provisioning with AD FS broken on windows server 2019.

Review to validate the AD FS and Active Directory configuration

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Record the information about the AD FS certificate, and set a renewal reminder at least six weeks before it expires. Relevant information includes: certificate serial number, thumbprint, common name, subject alternate name, name of the physical host server, the issued date, the expiration date, and issuing CA vendor (if a non-Microsoft certificate)
- ✓ Confirm you added the AD FS service account to the KeyAdmins group
- ✓ Confirm you enabled the Device Registration service

Configure an enrollment agent certificate template

A certificate registration authority (CRA) is a trusted authority that validates certificate request. Once it validates the request, it presents the request to the certification authority (CA) for issuance. The CA issues the certificate, returns it to the CRA, which returns the certificate to the requesting user. Windows Hello for Business certificate trust deployments use AD FS as the CRA.

The CRA enrolls for an *enrollment agent certificate*. Once the CRA verifies the certificate request, it signs the certificate request using its enrollment agent certificate and sends it to the CA. The Windows Hello for Business Authentication certificate template is configured to only issue certificates to certificate requests that have been signed with an enrollment agent certificate. The CA only issues a certificate for that template if the registration authority signs the certificate request.

> **ⓘ Important**
>
> Follow the procedures below based on the AD FS service account used in your environment.

Create an enrollment agent certificate for Group Managed Service Accounts (GMSA)

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** and select **Manage**
3. In the **Certificate Template Console**, right-click on the **Exchange Enrollment Agent (Offline request)** template details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the *Certification Authority list*

Tab Name	Configurations
	• Select **Windows 10 / Windows Server 2016** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *WHFB Enrollment Agent* • Set the validity period to the desired value
Subject Name	Select **Supply in the request** **Note:** Group Managed Service Accounts (GMSA) don't support the *Build from this Active Directory information* option and will result in the AD FS server failing to enroll the enrollment agent certificate. You must configure the certificate template with *Supply in the request* to ensure that AD FS servers can perform the automatic enrollment and renewal of the enrollment agent certificate.
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

Security	• Select **Add**
	• Select **Object Types** and select the **Service Accounts** check box
	• Select **OK**
	• Type `adfssvc` in the **Enter the object names to select** text box and select **OK**
	• Select the **adfssvc** from the **Group or users names** list. In the **Permissions for adfssvc** section:
	  ◦ In the **Permissions for adfssvc** section, select the **Allow** check box for the **Enroll** permission
	  ◦ Excluding the **adfssvc** user, clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other items in the **Group or users names** list
	• Select **OK**

5. Select **OK** to finalize your changes and create the new template

6. Close the console

Create an enrollment agent certificate for a standard service account

Sign in to a CA or management workstations with *Domain Administrator* equivalent credentials.

1. Open the **Certification Authority** management console
2. Right-click **Certificate Templates** and select **Manage**
3. In the **Certificate Template Console**, right-click on the **Exchange Enrollment Agent (Offline request)** template details pane and select **Duplicate Template**
4. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2016** from the **Certification Authority** list • Select **Windows 10 / Windows Server 2016** from the **Certificate Recipient** list
General	• Specify a **Template display name**, for example *WHFB Enrollment Agent* • Set the validity period to the desired value
Subject Name	• Select **Build from this Active Directory information** • Select **Fully distinguished name** from the **Subject name format** list • Select the **User Principal Name (UPN)** check box under **Include this information in alternative subject name**
Cryptography	• Set the *Provider Category* to **Key Storage Provider** • Set the *Algorithm name* to **RSA** • Set the *minimum key size* to **2048** • Set the *Request hash* to **SHA256**

Security	• Select **Add**
	• Select **Object Types** and select the **Service Accounts** check box
	• Select **OK**
	• Type `adfssvc` in the **Enter the object names to select** text box and select **OK**
	• Select the **adfssvc** from the **Group or users names** list. In the **Permissions for adfssvc** section:
	  ◦ In the **Permissions for adfssvc** section, select the **Allow** check box for the **Enroll** permission
	  ◦ Excluding the **adfssvc** user, clear the **Allow** check box for the **Enroll** and **Autoenroll** permissions for all other items in the **Group**

Tab Name	Configurations
	or users names list • Select **OK**

5. Select **OK** to finalize your changes and create the new template
6. Close the console

Publish the certificate template to the CA

Sign in to the CA or management workstations with **Enterprise Admin** equivalent credentials.

1. Open the **Certification Authority** management console.
2. Expand the parent node from the navigation pane.
3. Select **Certificate Templates** in the navigation pane.
4. Right-click the **Certificate Templates** node. Select **New** > **Certificate Template** to issue.
5. In the **Enable Certificates Templates** window, select the *WHFB Enrollment Agent* template you created in the previous step. Select **OK** to publish the selected certificate templates to the certification authority.
6. If you published the *Domain Controller Authentication (Kerberos)* certificate template, then unpublish the certificate templates you included in the superseded templates list.
    - To unpublish a certificate template, right-click the certificate template you want to unpublish and select **Delete**. Select **Yes** to confirm the operation.
7. Close the console.

Configure the certificate registration authority

The Windows Hello for Business on-premises certificate-based deployment uses AD FS as the certificate registration authority (CRA). The registration authority is responsible for issuing certificates to users and devices. The registration authority is also responsible for revoking certificates when users or devices are removed from the environment.

Sign-in the AD FS server with *domain administrator* equivalent credentials.

Open a **Windows PowerShell** prompt and type the following command:

```
PowerShell
```

```
Set-AdfsCertificateAuthority -EnrollmentAgent -
EnrollmentAgentCertificateTemplate WHFBEnrollmentAgent -
WindowsHelloCertificateTemplate WHFBAuthentication
```

> **Note**
>
> If you gave your Windows Hello for Business Enrollment Agent and Windows Hello for Business Authentication certificate templates different names, then replace *WHFBEnrollmentAgent* and *WHFBAuthentication* in the above command with the name of your certificate templates. It's important that you use the template name rather than the template display name. You can view the template name on the **General** tab of the certificate template by using the **Certificate Template** management console (*certtmpl.msc*). Or, you can view the template name by using the `Get-CATemplate` PowerShell cmdlet on a CA.

Enrollment agent certificate lifecycle management

AD FS performs its own certificate lifecycle management. Once the registration authority is configured with the proper certificate template, the AD FS server attempts to enroll the certificate on the first certificate request or when the service first starts.

Approximately 60 days prior to enrollment agent certificate's expiration, the AD FS service attempts to renew the certificate until it is successful. If the certificate fails to renew, and the certificate expires, the AD FS server will request a new enrollment agent certificate. You can view the AD FS event logs to determine the status of the enrollment agent certificate.

Additional federation servers

Organizations should deploy more than one federation server in their federation farm for high-availability. You should have a minimum of two federation services in your AD FS farm, however most organizations are likely to have more. This largely depends on the number of devices and users using the services provided by the AD FS farm.

Server authentication certificate

Each server you add to the AD FS farm must have a proper server authentication certificate. Refer to the Enroll for a TLS Server Authentication Certificate section of this document to determine the requirements for your server authentication certificate. As

previously stated, AD FS servers used exclusively for on-premises deployments of Windows Hello for Business can use enterprise server authentication certificates rather than server authentication certificates issued by public certificate authorities.

Install additional servers

Adding federation servers to the existing AD FS farm begins with ensuring the server are fully patched, to include Windows Server 2016 Update needed to support Windows Hello for Business deployments (https://aka.ms/whfbadfs1703). Next, install the Active Directory Federation Service role on the additional servers and then configure the server as an additional server in an existing farm.

Load balance AD FS

Many environments load balance using hardware devices. Environments without hardware load-balancing capabilities can take advantage the network load-balancing feature included in Windows Server to load balance the AD FS servers in the federation farm. Install the Windows Network Load Balancing feature on all nodes participating in the AD FS farm that should be load balanced.

Install Network Load Balancing Feature on AD FS Servers

Sign-in the federation server with *Enterprise Administrator* equivalent credentials.

1. Start **Server Manager**. Select **Local Server** in the navigation pane
2. Select **Manage** and then select **Add Roles and Features**
3. Select **Next** On the **Before you begin** page
4. On the **Select installation type** page, select **Role-based or feature-based installation** and select **Next**
5. On the **Select destination server** page, choose **Select a server from the server pool**. Select the federation server from the **Server Pool** list. Select **Next**
6. On the **Select server roles** page, select **Next**
7. Select **Network Load Balancing** on the **Select features** page
8. Select **Install** to start the feature installation

Configure Network Load Balancing for AD FS

Before you can load balance all the nodes in the AD FS farm, you must first create a new load balance cluster. Once you have created the cluster, then you can add new nodes to that cluster.

Sign-in a node of the federation farm with *Administrator* equivalent credentials.

1. Open **Network Load Balancing Manager** from **Administrative Tools**
2. Right-click **Network Load Balancing Clusters**, and then select **New Cluster**
3. To connect to the host that is to be a part of the new cluster, in the **Host** text box, type the name of the host, and then select **Connect**
4. Select the interface that you want to use with the cluster, and then select **Next** (the interface hosts the virtual IP address and receives the client traffic to load balance)
5. In **Host Parameters**, select a value in **Priority (Unique host identifier)**. This parameter specifies a unique ID for each host. The host with the lowest numerical priority among the current members of the cluster handles all of the cluster's network traffic that is not covered by a port rule. Select **Next**
6. In **Cluster IP Addresses**, select **Add** and type the cluster IP address that is shared by every host in the cluster. NLB adds this IP address to the TCP/IP stack on the selected interface of all hosts that are chosen to be part of the cluster. Select **Next**
7. In **Cluster Parameters**, select values in **IP Address** and **Subnet mask** (for IPv6 addresses, a subnet mask value is not needed). Type the full Internet name that users will use to access this NLB cluster
8. In **Cluster operation mode**, select **Unicast** to specify that a unicast media access control (MAC) address should be used for cluster operations. In unicast mode, the MAC address of the cluster is assigned to the network adapter of the computer, and the built-in MAC address of the network adapter is not used. We recommend that you accept the unicast default settings. Select **Next**
9. In Port Rules, select Edit to modify the default port rules to use port 443

Additional AD FS Servers

1. To add more hosts to the cluster, right-click the new cluster, and then select **Add Host to Cluster**
2. Configure the host parameters (including host priority, dedicated IP addresses, and load weight) for the additional hosts by following the same instructions that you used to configure the initial host. Because you are adding hosts to an already configured cluster, all the cluster-wide parameters remain the same

Configure DNS for Device Registration

Sign-in the domain controller or administrative workstation with domain administrator equivalent credentials.
You'll need the *federation service* name to complete this task. You can view the federation service name by selecting **Edit Federation Service Properties** from the **Action**

pan of the **AD FS** management console, or by using `(Get-AdfsProperties).Hostname.` (PowerShell) on the AD FS server.

1. Open the **DNS Management** console
2. In the navigation pane, expand the domain controller name node and **Forward Lookup Zones**
3. In the navigation pane, select the node that has the name of your internal Active Directory domain name
4. In the navigation pane, right-click the domain name node and select **New Host (A or AAAA)**
5. In the **name** box, type the name of the federation service. In the **IP address** box, type the IP address of your federation server. Select **Add Host**
6. Right-click the `<domain_name>` node and select **New Alias (CNAME)**
7. In the **New Resource Record** dialog box, type `enterpriseregistration` in the **Alias** name box
8. In the **fully qualified domain name (FQDN)** of the target host box, type `federation_service_farm_name.<domain_name_fqdn`, and select OK
9. Close the DNS Management console

> ⓘ **Note**
>
> If your forest has multiple UPN suffixes, please make sure that `enterpriseregistration.<upnsuffix_fqdn>` is present for each suffix.

Configure the Intranet Zone to include the federation service

The Windows Hello provisioning presents web pages from the federation service. Configuring the intranet zone to include the federation service enables the user to authenticate to the federation service using integrated authentication. Without this setting, the connection to the federation service during Windows Hello provisioning prompts the user for authentication.

Create an Intranet Zone Group Policy

Sign-in the domain controller or administrative workstation with *Domain Admin* equivalent credentials:

1. Start the **Group Policy Management Console** (`gpmc.msc`)

2. Expand the domain and select the **Group Policy Object** node in the navigation pane
3. Right-click **Group Policy object** and select **New**
4. Type **Intranet Zone Settings** in the name box and select **OK**
5. In the content pane, right-click the **Intranet Zone Settings** Group Policy object and select **Edit**
6. In the navigation pane, expand **Policies** under **Computer Configuration**
7. Expand **Administrative Templates** > **Windows Component** > **Internet Explorer** > **Internet Control Panel** >**Security Page**. Open **Site to Zone Assignment List**
8. Select **Enable** > **Show**. In the **Value Name** column, type the url of the federation service beginning with https. In the **Value** column, type the number **1**. Select OK twice, then close the Group Policy Management Editor

Deploy the Intranet Zone Group Policy object

1. Start the **Group Policy Management Console** (gpmc.msc)
2. In the navigation pane, expand the domain and right-click the node that has your Active Directory domain name and select **Link an existing GPO…**
3. In the **Select GPO** dialog box, select **Intranet Zone Settings** or the name of the Windows Hello for Business Group Policy object you previously created and select **OK**

Event Logs

Use the event logs on the AD FS service to confirm the service account enrolled for an enrollment agent certificate. First, look for the AD FS event ID 443 that confirms certificate enrollment cycle has finished. Once confirmed the AD FS certificate enrollment cycle completed review the *CertificateLifecycle-User* event log. In this event log, look for event ID 1006, which indicates a new certificate was installed. Details of the event log should show:

- The account name under which the certificate was enrolled
- The action, which should read enroll
- The thumbprint of the certificate
- The certificate template used to issue the certificate

You cannot use the Certificate Manager to view enrolled certificates for group managed service accounts. Use the event log information to confirm the AD FS service account enrolled a certificate. Use `certutil.exe` to view the details of the certificate shown in the event log.

Group managed service accounts use user profiles to store user information, which included enrolled certificates. On the AD FS server, use a command prompt and navigate to `%systemdrive%\users\<adfsGMSA_name>\appdata\roaming\Microsoft\systemcertificates\my\certificates`.

Each file in this folder represents a certificate in the service account's Personal store (You may need to use `dir.exe /A` to view the files in the folder). Match the thumbprint of the certificate from the event log to one of the files in this folder. That file is the certificate. Use the `Certutil -q <certificateThumbprintFileName>` to view the basic information about the certificate.

For detailed information about the certificate, use `Certutil -q -v <certificateThumbprintFileName>`.

Validate and deploy multifactor authentication (MFA)

Windows Hello for Business requires users perform multifactor authentication (MFA) prior to enroll in the service. On-premises deployments can use, as MFA option:

- certificates

  > **Note**
  >
  > When using this option, the certificates must be deployed to the users. For example, users can use their smart card or virtual smart card as a certificate authentication option.

- non-Microsoft authentication providers for AD FS
- custom authentication provider for AD FS

> **Important**
>
> As of July 1, 2019, Microsoft will no longer offer MFA Server for new deployments. New customers who would like to require multifactor authentication from their users should use cloud-based Microsoft Entra multifactor authentication. Existing customers who have activated MFA Server prior to July 1 will be able to download the latest version, future updates and generate activation credentials as usual.

For information on available non-Microsoft authentication methods see Configure Additional Authentication Methods for AD FS. For creating a custom authentication

method see Build a Custom Authentication Method for AD FS in Windows Server

Follow the integration and deployment guide for the authentication provider you select to integrate and deploy it to AD FS. Make sure that the authentication provider is selected as a multifactor authentication option in the AD FS authentication policy. For information on configuring AD FS authentication policies see Configure Authentication Policies.

> **Tip**
>
> When you validate the AD FS configuration, verify if you need to update the configuration of user agent strings to support Windows Integrated Authentication (WIA). For more information, see **Change WIASupportedUserAgent settings**.

Review to validate the configuration

Before you continue with the deployment, validate your deployment progress by reviewing the following items:

- ✓ Configure an enrollment agent certificate template.
- ✓ Confirm only the AD FS service account has the allow enroll permission for the enrollment agent certificate template.
- ✓ Consider using an HSM to protect the enrollment agent certificate; however, understand the frequency and quantity of signature operations the enrollment agent server makes and understand the impact it has on overall performance.
- ✓ Confirm you properly configured the Windows Hello for Business authentication certificate template.
- ✓ Confirm all certificate templates were properly published to the appropriate issuing certificate authorities.
- ✓ Confirm the AD FS service account has the allow enroll permission for the Windows Hello Business authentication certificate template.
- ✓ Confirm the AD FS certificate registration authority is properly configured using the `Get-AdfsCertificateAuthority` Windows PowerShell cmdlet.
- ✓ Confirm you restarted the AD FS service.
- ✓ Confirm you properly configured load-balancing (hardware or software).
- ✓ Confirm you created a DNS A Record for the federation service and the IP address used is the load-balanced IP address.
- ✓ Confirm you created and deployed the Intranet Zone settings to prevent double authentication to the federation server.
- ✓ Confirm you have deployed a MFA solution for AD FS.

Next: configure and enroll in Windows Hello for Business >

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure and enroll in Windows Hello for Business in an on-premises certificate trust model

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises ⓘ
- **Trust type:** certificate trust ⓘ
- **Join type:** domain join ⓘ

Once the prerequisites are met, and the PKI and AD FS configurations are validated, deploying Windows Hello for Business consists of the following steps:

- ✓ Configure Windows Hello for Business policy settings
- ✓ Enroll in Windows Hello for Business

Configure Windows Hello for Business policy settings

There are 2 policy setting required to enable Windows Hello for Business in a certificate trust model:

- Use Windows Hello for Business
- Use certificate for on-premises authentication

Another optional, but recommended, policy setting is:

- Use a hardware security device

Follow the instructions below to configure your devices using either Microsoft Intune or group policy (GPO).

You can configure the Use Windows Hello for Business policy setting in the computer or user node of a GPO:

- Deploying the computer node policy setting, results in all users that sign-in to the targeted devices to attempt a Windows Hello for Business enrollment

- Deploying the user node policy setting, results in only the targeted users to attempt a Windows Hello for Business enrollment

If both user and computer policy settings are deployed, the user policy setting has precedence.

> **Tip**
>
> Use the same *Windows Hello for Business Users* security group to assign **Certificate template permissions** to ensure the same members can enroll in the Windows Hello for Business authentication certificate.

Enable automatic enrollment of certificates group policy setting

Windows Hello for Business provisioning performs the initial enrollment of the Windows Hello for Business authentication certificate. This certificate expires based on the duration configured in the Windows Hello for Business *authentication certificate* template.

The process requires no user interaction, provided the user signs in using Windows Hello for Business. The certificate is renewed in the background before it expires.

To configure a device with group policy, use the Local Group Policy Editor. To configure multiple devices joined to Active Directory, create or edit a group policy object (GPO) and use the following settings:

Expand table

Group policy path	Group policy setting	Value
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business or **User Configuration\Administrative Templates\Windows Components\Windows Hello for Business**	Use Windows Hello for Business	**Enabled**
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business or	Use certificate for on-premises authentication	**Enabled**

Group policy path	Group policy setting	Value
User Configuration\Administrative Templates\Windows Components\Windows Hello for Business		
Computer Configuration\Windows Settings\Security Settings\Public Key Policies or **User Configuration\Windows Settings\Security Settings\Public Key Policies**	Certificate Services Client - Auto-Enrollment	- Select **Enabled** from the **Configuration Model** - Select the **Renew expired certificates, update pending certificates, and remove revoked certificates** - Select **Update certificates that use certificate templates**
Computer Configuration\Administrative Templates\Windows Components\Windows Hello for Business	Use a hardware security device	**Enabled**

> **Note**
>
> The enablement of the *Use a hardware security device* policy setting is optional, but recommended.

Group policies can be linked to domains or organizational units, filtered using security groups, or filtered using WMI filters.

> **Tip**
>
> The best way to deploy the Windows Hello for Business GPO is to use security group filtering. Only members of the targeted security group will provision Windows Hello for Business, enabling a phased rollout. This solution allows linking the GPO to the domain, ensuring the GPO is scoped to all security principals. The security group filtering ensures that only the members of the global group receive and apply the GPO, which results in the provisioning of Windows Hello for Business.

Additional policy settings can be configured to control the behavior of Windows Hello for Business. For more information, see Windows Hello for Business policy settings.

Enroll in Windows Hello for Business

The Windows Hello for Business provisioning process begins immediately after the user profile is loaded and before the user receives their desktop. For the provisioning process to begin, all prerequisite checks must pass.

You can determine the status of the prerequisite checks by viewing the **User Device Registration** admin log under **Applications and Services Logs** > **Microsoft** > **Windows**. This information is also available using the `dsregcmd.exe /status` command from a console. For more information, see dsregcmd.

User experience

After a user signs in, the Windows Hello for Business enrollment process begins:

1. If the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**
3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device
5. The remainder of the provisioning includes Windows Hello for Business requesting an asymmetric key pair for the user, preferably from the TPM (or required if explicitly set through policy). Once the key pair is acquired, Windows communicates with the IdP to register the public key. When key registration completes, Windows Hello for Business provisioning informs the user they can use their PIN to sign-in. The user may close the provisioning application and access their desktop

After a successful key registration, Windows creates a certificate request using the same key pair to request a certificate. Windows sends the certificate request to the AD FS server for certificate enrollment.

The AD FS registration authority verifies the key used in the certificate request matches the key that was previously registered. On a successful match, the AD FS registration authority signs the certificate request using its enrollment agent certificate and sends it to the certificate authority.

The CA validates that the certificate is signed by the registration authority. On successful validation, it issues a certificate based on the request and returns the certificate to the AD FS registration authority. The registration authority returns the certificate to Windows where it then installs the certificate in the current user's certificate store.

The following video shows the Windows Hello for Business enrollment steps after signing in with a password, using a custom MFA adapter for AD FS.
https://learn-video.azurefd.net/vod/player?id=771165c0-e37f-4f9d-9e21-4f383cc6590d&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fon-premises-cert-trust-enroll

Sequence diagram

To better understand the provisioning flows, review the following sequence diagram:

- Provisioning in an on-premises certificate trust deployment model

---

Feedback

Was this page helpful? Yes No

Provide product feedback

Prepare users to provision and use Windows Hello for Business

Article • 02/25/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

This article provides guidance on how to prepare users to enroll and to use Windows Hello for Business. It also provides guidance on how to communicate the benefits of Windows Hello for Business to users.

Multi-factor authentication

The provisioning of Windows Hello requires users to authenticate with multi-factor (MFA). Ensure that you have a solution in place for users to use MFA during the process.

> **Tip**
>
> To facilitate user communication and to ensure a successful Windows Hello for Business deployment, you can find customizable material (email templates, posters, trainings, etc.) at Microsoft Entra templates.

Biometric gestures

Depending on the hardware, users might be prompted to register their fingerprint or face. Explain to users that for convenience, they should register their biometric gesture during the provisioning process. The biometric gesture can be used to unlock the device and to authenticate to resources that require Windows Hello for Business. Biometric gestures are valid only on the enrolled device and are not stored outside the device.

User experience

The next video shows the Windows Hello for Business enrollment experience after a user signs in with a password:

1. Since the device supports biometric authentication, the user is prompted to set up a biometric gesture. This gesture can be used to unlock the device and authenticate to resources that require Windows Hello for Business. The user can skip this step if they don't want to set up a biometric gesture
2. The user is prompted to use Windows Hello with the organization account. The user selects **OK**

3. The provisioning flow proceeds to the multi-factor authentication portion of the enrollment. Provisioning informs the user that it's actively attempting to contact the user through their configured form of MFA. The provisioning process doesn't proceed until authentication succeeds, fails or times out. A failed or timeout MFA results in an error and asks the user to retry
4. After a successful MFA, the provisioning flow asks the user to create and validate a PIN. This PIN must observe any PIN complexity policies configured on the device

https://learn-video.azurefd.net/vod/player?id=36dc8679-0fcc-4abf-868d-97ec8b749da7&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fprepare-users

After enrollment in Windows Hello, users should use their gesture (such as a PIN or fingerprint) for access to their devices and corporate resources. The unlock gesture is valid only on the enrolled device.

> **Important**
>
> Although the organization might require users to change their Active Directory or Microsoft Entra account password at regular intervals, changes to their passwords have no effect on Hello.

The next video shows the Windows Hello for Business enrollment experience as part of the out-of-box-experience (OOBE) process:

1. The user joins the device to Microsoft Entra ID and is prompted for MFA during the join process
2. The device is Managed by Microsoft Intune and applies Windows Hello for Business policy settings
3. After the user profile is loaded, but before the access to the desktop is granted, the user must enroll in Windows Hello

https://learn-video.azurefd.net/vod/player?id=44c16430-756f-490a-9fc1-80e2724fef8d&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fdeploy%2Fprepare-users

---

Feedback

Was this page helpful? Yes No

Provide product feedback

PIN reset

Applies to: ✅ Windows 11, ✅ Windows 10

This article describes how *Microsoft PIN reset service* enables your users to recover a forgotten Windows Hello for Business PIN, and how to configure it.

Overview

Windows Hello for Business provides the capability for users to reset forgotten PINs. There are two forms of PIN reset:

- *Destructive PIN reset*: The user's existing PIN and underlying credentials, including any keys or certificates added to their Windows Hello container, are deleted from the client and a new sign in key and PIN are provisioned. Destructive PIN reset is the default option, and doesn't require configuration.
- *Non-destructive PIN reset*: The user's Windows Hello for Business container and keys are preserved, but the user's PIN that they use to authorize key usage is changed. For nondestructive PIN reset, you must deploy the *Microsoft PIN reset service* and configure your clients' policy to enable the *PIN recovery* feature.

How nondestructive PIN reset works

Requirements:

- Hybrid or cloud-only Windows Hello for Business deployments
- Windows Enterprise, Education and Pro editions. There's no licensing requirement for this feature

When nondestructive PIN reset is enabled on a client, a *256-bit AES* key is generated locally. The key is added to a user's Windows Hello for Business container and keys as the *PIN reset protector*. This PIN reset protector is encrypted using a public key retrieved from the Microsoft PIN reset service and then stored on the client for later use during PIN reset. After a user initiates a PIN reset, completes authentication and multifactor authentication to Microsoft Entra ID, the encrypted PIN reset protector is sent to the Microsoft PIN reset service, decrypted, and returned to the client. The decrypted PIN reset protector is used to change the PIN used to authorize Windows Hello for Business keys, and it's then cleared from memory.

Using Group Policy, Microsoft Intune or a compatible MDM solution, you can configure Windows devices to securely use the Microsoft PIN reset service, which enables users to reset their forgotten PIN without requiring re-enrollment.

The following table compares destructive and nondestructive PIN reset:

Expand table

Category	Destructive PIN reset	Nondestructive PIN reset
Functionality	The user's existing PIN and underlying credentials, including any keys or certificates added to their Windows Hello container, are deleted from the client and a new sign in key and PIN are provisioned.	You must deploy the Microsoft PIN reset service and client policy to enable the PIN recovery feature. During a nondestructive PIN reset, the user's Windows Hello for Business container and keys are preserved, but the user's PIN that they use to authorize key usage is changed.
Microsoft Entra joined	Cert Trust, Key Trust, and cloud Kerberos trust	Cert Trust, Key Trust, and cloud Kerberos trust
Microsoft Entra hybrid joined	Cert Trust and cloud Kerberos trust for both settings and above the lock support destructive PIN reset. Key Trust doesn't support this option from above the lock screen. This is due to the sync delay between when a user provisions their Windows Hello for Business credential and being able to use it for sign-in. It does support from the settings page and the users must have a corporate network connectivity to the DC.	Cert Trust, Key Trust, and cloud Kerberos trust for both settings and above the lock support nondestructive PIN reset. No network connection is required for the DC.
On Premises	If AD FS is used for on premises deployments, users must have a corporate network connectivity to federation services.	The PIN reset service relies on Microsoft Entra identities, so it's only available for Microsoft Entra hybrid joined and Microsoft Entra joined devices.
Additional configuration required	Supported by default and doesn't require configuration	Deploy the Microsoft PIN reset service and client policy to enable the PIN recovery feature.
MSA/Enterprise	MSA and Enterprise	Enterprise only.

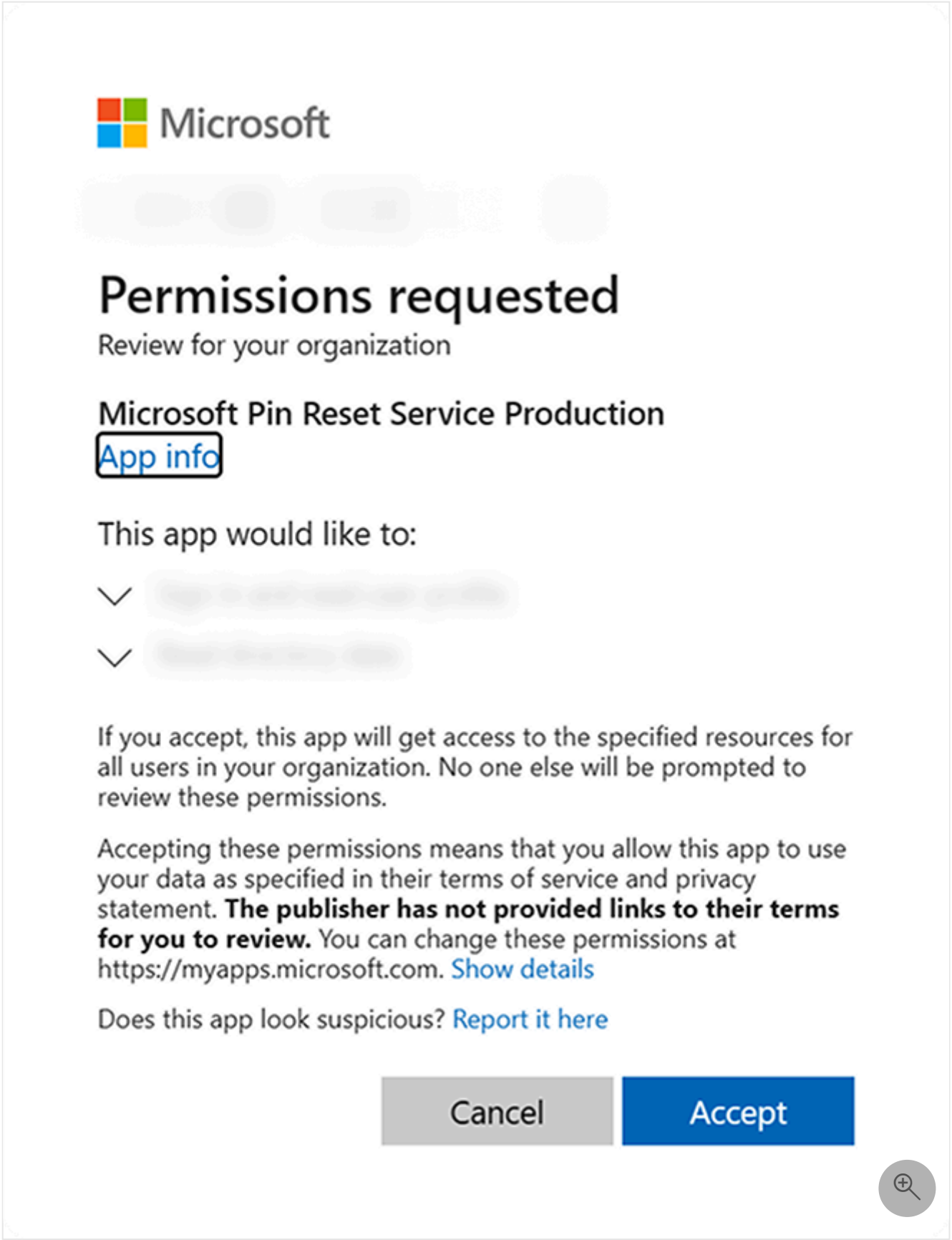
Enable the Microsoft PIN Reset Service in your Microsoft Entra tenant

Before you can use nondestructive PIN reset, you must register two applications in your Microsoft Entra tenant:

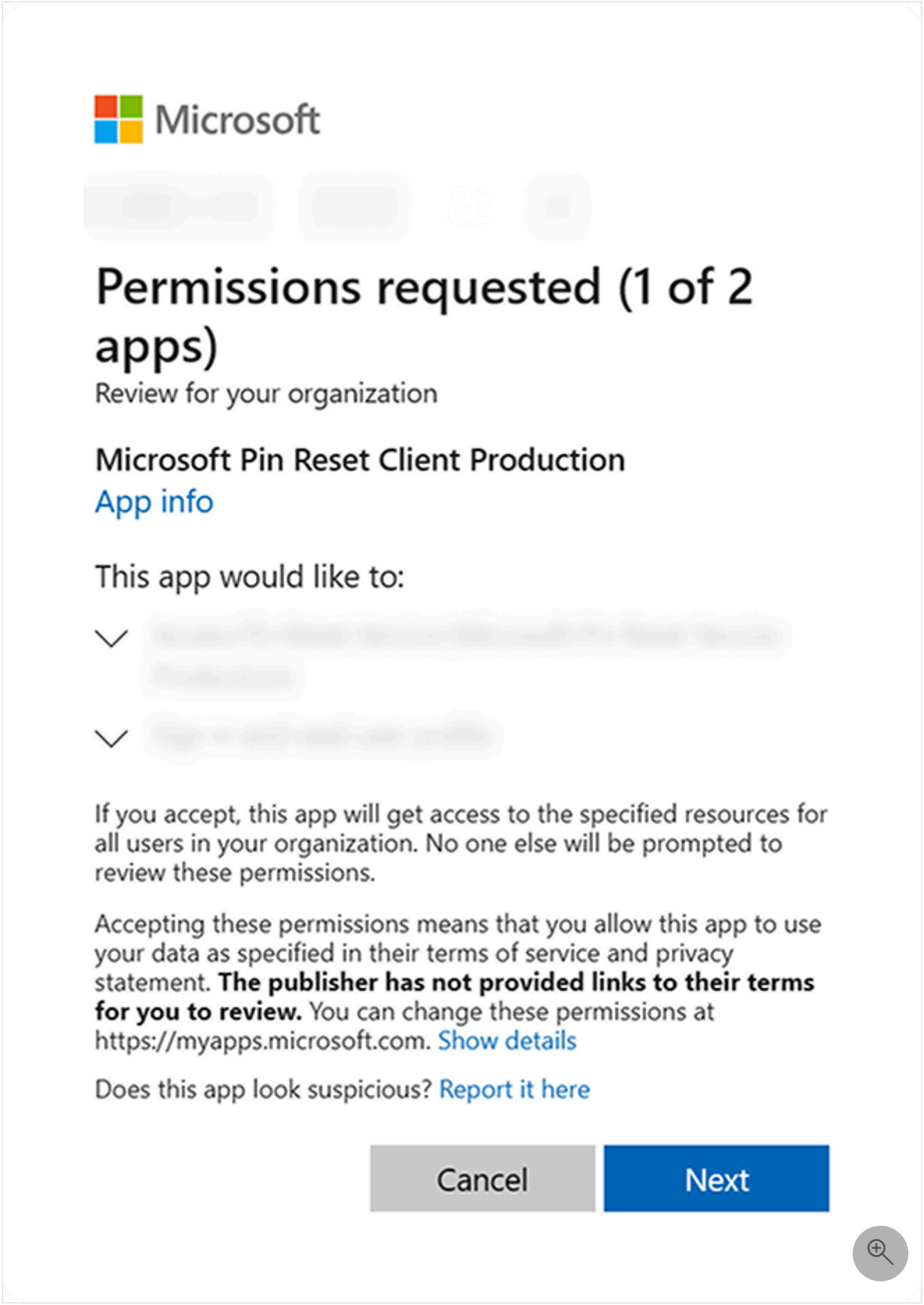
- Microsoft Pin Reset Service Production
- Microsoft Pin Reset Client Production

To register the applications, follow these steps:

1. Go to the Microsoft PIN Reset Service Production website, and sign in as at least an Application Administrator. Review the permissions requested by the *Microsoft Pin Reset Service Production* application and select **Accept** to give consent to the application to access your organization

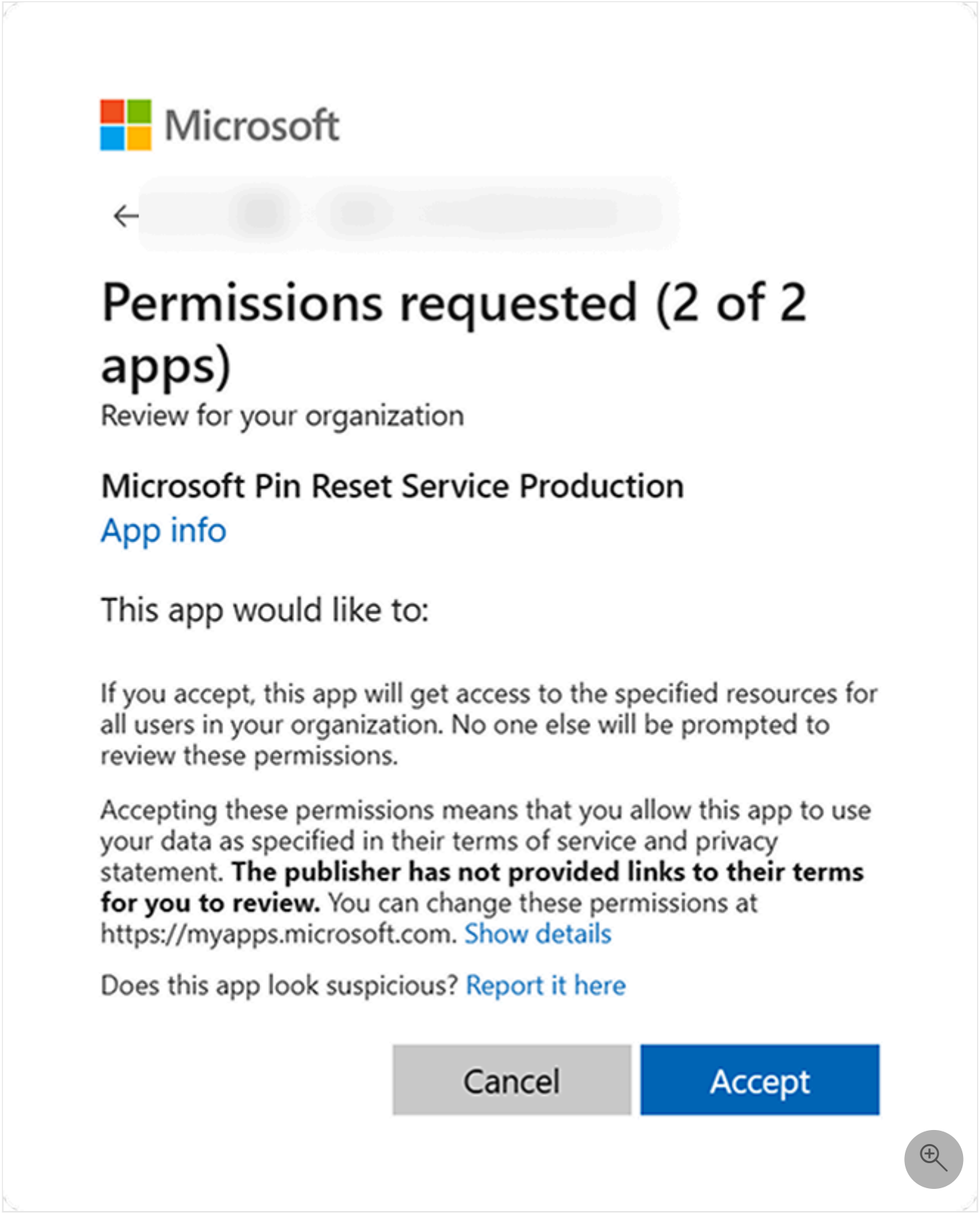


2. Go to the Microsoft PIN Reset Client Production website, and sign as at least an Application Administrator. Review the permissions requested by the *Microsoft Pin Reset Client Production* application, and select **Next**.



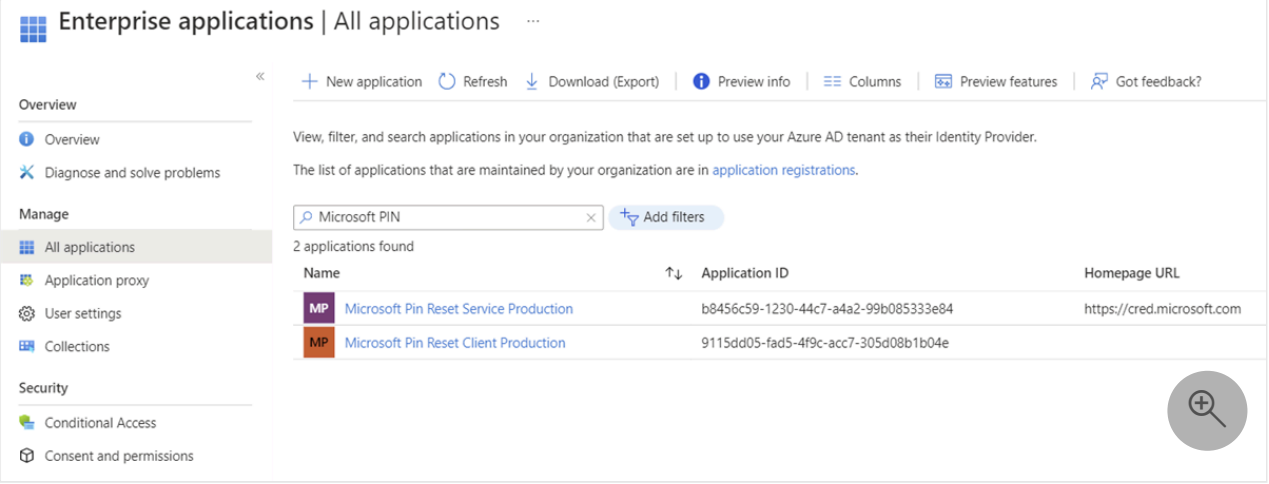
3. Review the permissions requested by the *Microsoft Pin Reset Service Production* application and select **Accept** to confirm consent to both applications to access your organization.

> **Note**
>
> After acceptance, the redirect page will show a blank page. This is a known behavior.



Confirm that the two PIN Reset service principals are registered in your tenant

1. Sign in to the Microsoft Entra Manager admin center
2. Select **Microsoft Entra ID** > **Applications** > **Enterprise applications**
3. Search by application name "Microsoft PIN" and verify that both **Microsoft Pin Reset Service Production** and **Microsoft Pin Reset Client Production** are in the list



Enable PIN recovery on the clients

To enable PIN recovery on the clients, you can use:

- Microsoft Intune/MDM
- Group policy

The following instructions provide details how to configure your devices. Select the option that best suits your needs.

Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Windows Hello For Business	Enable Pin Recovery	True

Assign the policy to a group that contains as members the devices or users that you want to configure.

> **Note**
>
> You can also configure PIN recovery from the **Endpoint security** blade:
>
> 1. Sign in to the Microsoft Intune admin center
> 2. Select **Endpoint security** > **Account protection** > **Create Policy**

Alternatively, you can configure devices using a custom policy with the PassportForWork CSP.

Expand table

OMA-URI	Data type	Value
`./Vendor/MSFT/Policy/PassportForWork/`*TenantId*`/Policies/EnablePinRecovery`	Boolean	True

> **Note**
>
> You must replace `TenantId` with the identifier of your Microsoft Entra tenant. To look up your Tenant ID, see How to find your Microsoft Entra tenant ID or try the following, ensuring to sign-in with your organization's account::

msgraph

```
GET https://graph.microsoft.com/v1.0/organization?$select=id
```

Confirm that PIN Recovery policy is enforced on the devices

The *PIN reset* configuration can be viewed by running **dsregcmd /status** from the command line. This state can be found under the output in the user state section as the **CanReset** line item. If **CanReset** reports as DestructiveOnly, then only destructive PIN reset is enabled. If **CanReset** reports DestructiveAndNonDestructive, then nondestructive PIN reset is enabled.

Sample User state Output for Destructive PIN Reset

Windows Command Prompt

```
+----------------------------------------------------------------------+
| User State                                                           |
+----------------------------------------------------------------------+
```

```
                    NgcSet : YES
                  NgcKeyId : {FA0DB076-A5D7-4844-82D8-50A2FB42EC7B}
                  CanReset : DestructiveOnly
            WorkplaceJoined : NO
              WamDefaultSet : YES
       WamDefaultAuthority : organizations
              WamDefaultId : https://login.microsoft.com
             WamDefaultGUID : { B16898C6-A148-4967-9171-64D755DA8520 } (AzureAd)

+----------------------------------------------------------------------+
```

Sample User state Output for Non-Destructive PIN Reset

Windows Command Prompt

```
+----------------------------------------------------------------------+
| User State                                                           |
+----------------------------------------------------------------------+

                    NgcSet : YES
                  NgcKeyId : {FA0DB076-A5D7-4844-82D8-50A2FB42EC7B}
                  CanReset : DestructiveAndNonDestructive
            WorkplaceJoined : NO
              WamDefaultSet : YES
       WamDefaultAuthority : organizations
              WamDefaultId : https://login.microsoft.com
             WamDefaultGUID : { B16898C6-A148-4967-9171-64D755DA8520 } (AzureAd)

+----------------------------------------------------------------------+
```

Configure allowed URLs for federated identity providers on Microsoft Entra joined devices

Applies to: Microsoft Entra joined devices

PIN reset on Microsoft Entra joined devices uses a flow called *web sign-in* to authenticate users in the lock screen. Web sign-in only allows navigation to specific domains. If web sign-in attempts to navigate to a domain that isn't allowed, it displays a page with the error message: *We can't open that page right now*.
If you have a federated environment and authentication is handled using AD FS or a non-Microsoft identity provider, then you must configure your devices with a policy to allow a list of domains that can be reached during PIN reset flows. When set, it ensures that authentication pages from that identity provider can be used during Microsoft Entra joined PIN reset.

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Authentication	Configure Web Sign In Allowed Urls	Provide a semicolon delimited list of domains needed for authentication during the PIN reset scenario. An example value would be **signin.contoso.com;portal.contoso.com**

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the Policy CSP.

Expand table

Setting
• OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebSignInAllowedUrls` • Data type: String • Value: Provide a semicolon delimited list of domains needed for authentication during the PIN reset scenario. An example value would be **signin.contoso.com;portal.contoso.com**

> **Note**
>
> For Azure Government, there is a known issue with PIN reset on Microsoft Entra joined devices failing. When the user attempts to launch PIN reset, the PIN reset UI shows an error page that says, "*We can't open that page right now*". The ConfigureWebSignInAllowedUrls policy can be used to work around this issue. If you are experiencing this problem and you are using Azure US Government cloud, set **login.microsoftonline.us** as the value for the ConfigureWebSignInAllowedUrls policy.

User experience

Destructive and nondestructive PIN reset scenarios use the same steps for initiating a PIN reset. If users have forgotten their PINs, but have an alternate sign-in method, they can navigate to Sign-in options in *Settings* and initiate a PIN reset from the PIN options. If users don't have an alternate way to sign into their devices, PIN reset can also be initiated from the Windows lock screen with the *PIN credential provider*. Users must authenticate and complete multifactor authentication to reset their PIN. After PIN reset is complete, users can sign in using their new PIN.

> **Important**

For Microsoft Entra hybrid joined devices, users must have corporate network connectivity to domain controllers to complete destructive PIN reset. If AD FS is being used for certificate trust or for on-premises only deployments, users must also have corporate network connectivity to federation services to reset their PIN.

Reset PIN from Settings

1. Sign-in to Windows 10 using an alternate credential
2. Open **Settings** > **Accounts** > **Sign-in options**
3. Select **PIN (Windows Hello)** > **I forgot my PIN** and follow the instructions

Reset PIN from the lock screen

For Microsoft Entra joined devices:

1. If the PIN credential provider isn't selected, expand the **Sign-in options** link, and select the PIN pad icon
2. Select **I forgot my PIN** from the PIN credential provider
3. Select an authentication option from the list of presented options. This list is based on the different authentication methods enabled in your tenant (like Password, PIN, Security key)
4. Follow the instructions provided by the provisioning process
5. When finished, unlock your desktop using your newly created PIN

https://learn-video.azurefd.net/vod/player?id=310f7665-6276-4ad8-b76e-429073c10972&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fpin-reset

For Microsoft Entra hybrid joined devices:

1. If the PIN credential provider isn't selected, expand the **Sign-in options** link, and select the PIN pad icon
2. Select **I forgot my PIN** from the PIN credential provider
3. Enter your password and press enter
4. Follow the instructions provided by the provisioning process
5. When finished, unlock your desktop using your newly created PIN

> **Note**
>
> Key trust on Microsoft Entra hybrid joined devices doesn't support destructive PIN reset from above the Lock Screen. This is due to the sync delay between when a user provisions their Windows Hello for Business credential and being able to use it for sign-in. For this

deployment model, you must deploy non-destructive PIN reset for above lock PIN reset to work.

You may find that PIN reset from Settings only works post sign in. Also, the lock screen PIN reset function doesn't work if you have any matching limitation of self-service password reset from the lock screen. For more information, see Enable Microsoft Entra self-service password reset at the Windows sign-in screen.

Last updated on 03/29/2026

Dual enrollment

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

This article describes Windows Hello for Business functionalities or scenarios that apply to:

- **Deployment type:** on-premises ⓘ , hybrid ⓘ
- **Trust type:** certificate trust ⓘ
- **Join type:** domain join ⓘ , Microsoft Entra hybrid join ⓘ

> **ⓘ Important**
>
> Dual enrollment does not replace or provide the same security as Privileged Access Workstations feature. Microsoft encourages organizations to use the Privileged Access Workstations for their privileged credential users. Organizations can consider Windows Hello for Business dual enrollment in situations where the Privileged Access feature can't be used. To learn more, see **Privileged Access Workstations**.

Dual enrollment enables administrators to perform elevated, administrative functions by enrolling both their nonprivileged and privileged credentials on their device.

By design, Windows doesn't enumerate all Windows Hello for Business users from within a user's session. Using the group policy setting, **Allow enumeration of emulated smart card for all users**, you can configure a device to enumerate all enrolled Windows Hello for Business credentials on selected devices.

With this setting, administrative users can sign in to Windows using their nonprivileged Windows Hello credentials for normal work flow such as email, but can launch Microsoft Management Consoles (MMCs), Remote Desktop Services clients, and other applications by selecting **Run as different user** or **Run as administrator**, selecting the privileged user account, and providing their PIN. Administrators can also take advantage of this feature with command-line applications by using `runas.exe` combined with the `/smartcard` argument. This enables administrators to perform their day-to-day operations without needing to sign in and out, or use fast user switching when alternating between privileged and nonprivileged workloads.

> **ⓘ Important**

You must configure a Windows computer for Windows Hello for Business dual enrollment before either user (privileged or non-privileged) provisions Windows Hello for Business. Dual enrollment is a special setting that is configured on the Windows Hello container during creation.

Configure Windows Hello for Business dual enrollment

Here are the steps to enable dual enrollment:

- Configure Active Directory to support Domain Administrator enrollment
- Configure dual enrollment using Group Policy

Configure Active Directory to support Domain Administrator enrollment

The designed Windows Hello for Business configuration gives the `Key Admins` group read and write permissions to the `msDS-KeyCredentialsLink` attribute. You provided these permissions at root of the domain and use object inheritance to ensure the permissions apply to all users in the domain regardless of their location within the domain hierarchy.

Active Directory Domain Services uses `AdminSDHolder` to secure privileged users and groups from unintentional modification by comparing and replacing the security on privileged users and groups to match those defined on the AdminSDHolder object on an hourly cycle. For Windows Hello for Business, your domain administrator account might receive the permissions but they disappear from the user object unless you give the `AdminSDHolder` read and write permissions to the `msDS-KeyCredential` attribute.

Sign in to a domain controller or management workstation with access equivalent to *domain administrator*.

1. Type the following command to add the **allow** read and write property permissions for msDS-KeyCredentialLink attribute for the `Key Admins` group on the `AdminSDHolder` object.

   Windows Command Prompt

   ```
   dsacls "CN=AdminSDHolder,CN=System,DC=domain,DC=com" /g "
   [domainName\keyAdminGroup]":RPWP;msDS-KeyCredentialLink
   ```

where `DC=domain,DC=com` is the LDAP path of your Active Directory domain and `domainName\keyAdminGroup` is the NetBIOS name of your domain and the name of the group you use to give access to keys based on your deployment. For example:

Windows Command Prompt

```
dsacls "CN=AdminSDHolder,CN=System,DC=corp,DC=mstepdemo,DC=net" /g
"mstepdemo\Key Admins":RPWP;msDS-KeyCredentialLink
```

2. To trigger security descriptor propagation, open `ldp.exe`.
3. Select **Connection** and select **Connect...** Next to **Server**, type the name of the domain controller that holds the PDC role for the domain. Next to **Port**, type **389** and select **OK**.
4. Select **Connection** and select **Bind...** Select **OK** to bind as the currently signed-in user.
5. Select **Browser** and select **Modify**. Leave the **DN** text box blank. Next to **Attribute**, type **RunProtectAdminGroupsTask**. Next to **Values**, type `1`. Select **Enter** to add this to the **Entry List**.
6. Select **Run** to start the task.
7. Close LDP.

Configure dual enrollment with group policy

You configure Windows to support dual enrollment using the computer configuration portion of a Group Policy object:

1. Using the Group Policy Management Console (GPMC), create a new domain-based Group Policy object and link it to an organizational Unit that contains Active Directory computer objects used by privileged users.
2. Edit the Group Policy object from step 1.
3. Enable the **Allow enumeration of emulated smart cards for all users** policy setting located under **Computer Configuration->Administrative Templates->Windows Components->Windows Hello for Business**
4. Close the Group Policy Management Editor to save the Group Policy object. Close the GPMC.
5. Restart computers targeted by this Group Policy object.

The computer is ready for dual enrollment. Sign in as the privileged user first and enroll for Windows Hello for Business. Once completed, sign out and sign in as the nonprivileged user and enroll for Windows Hello for Business. You can now use your privileged credential to perform privileged tasks without using your password and without needing to switch users.

Feedback

Was this page helpful? Yes No

Provide product feedback

Dynamic lock

Article • 12/02/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

Dynamic lock is a feature that automatically locks a Windows device when a Bluetooth paired phone signal falls below the maximum Received Signal Strength Indicator (RSSI) value. The feature makes it more difficult for someone to gain access to your device if you step away from your PC and forget to lock it.

> **ⓘ Important**
>
> The dynamic lock feature only locks the device if the Bluetooth signal falls **and** the system is idle. If the system isn't idle (for example, an intruder gets access *before* the Bluetooth signal falls below the limit), the device won't lock. Therefore, the dynamic lock feature is an additional barrier. It doesn't replace the need for the user to lock the computer. It only reduces the probability of someone gaining access if the user forgets to lock it.

You can configure Windows devices to use the **dynamic lock** using a Group Policy Object (GPO).

1. Using the Group Policy Management Console (GPMC), scope a domain-based Group Policy to computer accounts in Active Directory.
2. Edit the Group Policy object from Step 1.
3. Enable the **Configure dynamic lock factors** policy setting located under **Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**.
4. Close the Group Policy Management Editor to save the Group Policy object.

   The Group Policy Editor, when the policy is enabled, creates a default signal rule policy with the following value:

   XML

   ```
   <rule schemaVersion="1.0">
   <signal type="bluetooth" scenario="Dynamic Lock" classOfDevice="512"
rssiMin="-10" rssiMaxDelta="-10"/>
   </rule>
   ```

> **ⓘ Important**

Microsoft recommends using the default values for this policy settings. Measurements are relative based on the varying conditions of each environment. Therefore, the same values may produce different results. Test policy settings in each environment prior to broadly deploying the setting.

For this policy setting, the `type` and `scenario` attribute values are static and can't change. The `classofDevice` is configurable but Phone is the only currently supported configuration. The attribute defaults to Phone and uses the values from the following table:

Expand table

Description	Value
Miscellaneous	0
Computer	256
Phone	512
LAN/Network Access Point	768
Audio/Video	1024
Peripheral	1280
Imaging	1536
Wearable	1792
Toy	2048
Health	2304
Uncategorized	7936

The `rssiMin` attribute value signal indicates the strength needed for the device to be considered *in-range*. The default value of `-10` enables a user to move about an average size office or cubicle without triggering Windows to lock the device. The `rssiMaxDelta` has a default value of `-10`, which instruct Windows to lock the device once the signal strength weakens by more than measurement of 10.

RSSI measurements are relative and lower as the bluetooth signals between the two paired devices reduces. Therefore a measurement of 0 is stronger than -10, which is stronger than -60, which is an indicator the devices are moving further apart from each other.

Configure Dynamic lock with Microsoft Intune

To configure Dynamic lock using Microsoft Intune, follow these steps:

1. Open the Microsoft Intune admin center and navigate to Devices > Windows > Configuration policies.
2. Create a new policy:
    - Platform: Windows 10 and later
    - Profile type: Templates - Custom
    - Select Create
3. Configure the profile:
    - Name: Provide a name for the profile.
    - Description: (Optional) Add a description.
4. Add OMA-URI settings:
    - Enable Dynamic lock:
        - Name: Enable Dynamic lock
        - Description: (Optional) This setting enables Dynamic lock
        - OMA-URI: ./Device/Vendor/MSFT/PassportForWork/DynamicLock/DynamicLock
        - Data type: Boolean
        - Value: True
    - Define the Dynamic lock signal rule:
        - Name: Dynamic lock Signal Rule
        - Description: (Optional) This setting configures Dynamic lock values
        - OMA-URI: ./Device/Vendor/MSFT/PassportForWork/DynamicLock/Plugins
        - Data type: String
        - Value: `<rule schemaVersion="1.0"><signal type="bluetooth" scenario="Dynamic Lock" classOfDevice="512" rssiMin="-10" rssiMaxDelta="-10"/></rule>`
5. Assign the profile to the appropriate groups.

Feedback

Was this page helpful? Yes No

Provide product feedback

Multi-factor unlock

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

Windows Hello for Business supports the use of a single credential (PIN and biometrics) for unlocking a device. Therefore, if any of those credentials are compromised (shoulder surfed), an attacker could gain access to the system.

Windows Hello for Business can be configured with *multi-factor unlock*, by extending Windows Hello with trusted signals. Administrators can configure devices to request a combination of factors and trusted signals to unlock them.

Multi-factor unlock is ideal for organizations that:

- Have expressed that PINs alone don't meet their security needs
- Want to prevent Information Workers from sharing credentials
- Want their organizations to comply with regulatory two-factor authentication policy
- Want to retain the familiar Windows sign-in user experience and not settle for a custom solution

How it works

First unlock factor credential provider and **Second unlock credential provider** are responsible for the bulk of the configuration. Each of these components contains a globally unique identifier (GUID) that represents a different Windows credential provider. With the policy setting enabled, users unlock the device using at least one credential provider from each category before Windows allows the user to proceed to their desktop.

The policy setting has three components:

- First unlock factor credential provider
- Second unlock factor credential provider
- Signal rules for device unlock

Configure unlock factors

> ⊗ **Caution**

When the **DontDisplayLastUserName** security policy is enabled, it is known to interfere with the ability to use multi factor unlock.

The **First unlock factor credential providers** and **Second unlock factor credential providers** portion of the policy setting each contain a comma separated list of credential providers.

Supported credential providers include:

Expand table

Credential Provider	GUID
PIN	`{D6886603-9D2F-4EB2-B667-1971041FA96B}`
Fingerprint	`{BEC09223-B018-416D-A0AC-523971B639F5}`
Facial Recognition	`{8AF662BF-65A0-4D0A-A540-A338A999D36F}`
Trusted Signal (Phone proximity, Network location)	`{27FBDB57-B613-4AF2-9D7E-4FA7A66C21AD}`

Note

Multifactor unlock does not support non-Microsoft credential providers or credential providers not listed in the above table.

The default credential providers for the **First unlock factor credential provider** include:

- PIN
- Fingerprint
- Facial Recognition

The default credential providers for the **Second unlock factor credential provider** include:

- Trusted Signal
- PIN

Configure a comma separated list of credential provider GUIDs you want to use as first and second unlock factors. While a credential provider can appear in both lists, a credential supported by that provider can only satisfy one of the unlock factors. Listed credential providers don't need to be in any specific order.

For example, if you include the PIN and fingerprint credential providers in both first and second factor lists, a user can use their fingerprint or PIN as the first unlock factor. Whichever factor you use to satisfy the first unlock factor can't be used to satisfy the second unlock factor. Each factor can therefore be used exactly once. The Trusted Signal provider can *only* be specified as part of the Second unlock factor credential provider list.

Configure Signal Rules for the Trusted Signal Credential Provider

The **Signal rules for device unlock** setting contains the rules the Trusted Signal credential provider uses to satisfy unlocking the device.

Rule element

You represent signal rules in XML. Each signal rule has a starting and ending `rule` element that contains the `schemaVersion` attribute and value. The current supported schema version is `1.0`.

Example

XML

```
<rule schemaVersion="1.0">
</rule>
```

Signal element

Each rule element has a `signal` element. All signal elements have a `type` element and `value`. The values supported are:

- bluetooth
- ipConfig
- wifi

Bluetooth

You define the bluetooth signal with more attributes in the signal element. The bluetooth configuration doesn't use any other elements. You can end the signal element

with short ending tag `/>`.

Expand table

Attribute	Value	Required
type	`bluetooth`	yes
scenario	`Authentication`	yes
classOfDevice	"*number*"	no
rssiMin	"*number*"	no
rssiMaxDelta	"*number*"	no

For example:

XML

```
<rule schemaVersion="1.0">
    <signal type="bluetooth" scenario="Authentication" classOfDevice="512"
rssiMin="-10" rssiMaxDelta="-10"/>
</rule>
```

The **classofDevice** attribute defaults to Phone and uses the values from the following table:

Expand table

Description	Value
Miscellaneous	0
Computer	256
Phone	512
LAN/Network Access Point	768
Audio/Video	1024
Peripheral	1280
Imaging	1536
Wearable	1792
Toy	2048

Description	Value
Health	2304
Uncategorized	7936

The **rssiMin** attribute value signal indicates the strength needed for the device to be considered "in-range". The default value of **-10** enables a user to move about an average size office or cubicle without triggering Windows to lock the device. The **rssiMaxDelta** has a default value of **-10**, which instruct Windows to lock the device once the signal strength weakens by more than measurement of 10.

RSSI measurements are relative, and lower as the bluetooth signals between the two paired devices reduces. A measurement of 0 is stronger than -10. A measurement of -10 is stronger than -60, and indicates that the devices are moving further apart from each other.

> **ⓘ Important**
>
> Microsoft recommends using the default values for this policy setting. Measurements are relative, based on the varying conditions of each environment. Therefore, the same values may produce different results. Test policy settings in each environment prior to broadly deploying the setting. Use the rssiMIN and rssiMaxDelta values from the XML file created by the Group Policy Management Editor or remove both attributes to use the default values.

IP Configuration

You define IP configuration signals using one or more ipConfiguration elements. Each element has a string value. IpConfiguration elements don't have attributes or nested elements.

IPv4Prefix

The IPv4 network prefix represented in Internet standard dotted-decimal notation. A network prefix that uses the Classless Inter-Domain Routing (CIDR) notation is required as part of the network string. A network port must not be present in the network string. A **signal** element may only contain one **ipv4Prefix** element. For example:

```
XML
```

```
<ipv4Prefix>192.168.100.0/24</ipv4Prefix>
```

The assigned IPv4 addresses in the range of 192.168.100.1 to 192.168.100.254 match this signal configuration.

IPv4Gateway

The IPv4 network gateway represented in Internet standard dotted-decimal notation. A network port or prefix must not be present in the network string. A **signal** element may only contain one **ipv4Gateway** element. For example:

XML

```
<ipv4Gateway>192.168.100.10</ipv4Gateway>
```

IPv4DhcpServer

The IPv4 DHCP server represented in Internet standard dotted-decimal notation. A network port or prefix must not be present in the network string. A **signal** element may only contain one **ipv4DhcpServer** element. For example:

XML

```
<ipv4DhcpServer>192.168.100.10</ipv4DhcpServer>
```

IPv4DnsServer

The IPv4 DNS server represented in Internet standard dotted-decimal notation. A network port or prefix must not be present in the network string.The **signal** element may contain one or more **ipv4DnsServer** elements.

Example:

XML

```
<ipv4DnsServer>192.168.100.10</ipv4DnsServer>
```

IPv6Prefix

The IPv6 network prefix represented in IPv6 network using Internet standard hexadecimal encoding. A network prefix in CIDR notation is required as part of the network string. A network port or scope ID must not be present in the network string. A **signal** element may only contain one **ipv6Prefix** element. For example:

```
XML

<ipv6Prefix>21DA:D3::/48</ipv6Prefix>
```

IPv6Gateway

The IPv6 network gateway represented in Internet standard hexadecimal encoding. An IPv6 scope ID may be present in the network string. A network port or prefix must not be present in the network string. A **signal** element may only contain one **ipv6Gateway** element. For example:

```
XML

<ipv6Gateway>21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A%2</ipv6Gateway>
```

IPv6DhcpServer

The IPv6 DNS server represented in Internet standard hexadecimal encoding. An IPv6 scope ID may be present in the network string. A network port or prefix must not be present in the network string. A **signal** element may only contain one **ipv6DhcpServer** element. For example:

```
XML

<ipv6DhcpServer>21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A%2</ipv6DhcpServer
```

IPv6DnsServer

The IPv6 DNS server represented in Internet standard hexadecimal encoding. An IPv6 scope ID may be present in the network string. A network port or prefix must not be present in the network string. The **signal** element may contain one or more **ipv6DnsServer** elements. For example:

```
XML

<ipv6DnsServer>21DA:00D3:0000:2F3B:02AA:00FF:FE28:9C5A%2</ipv6DnsServer>
```

dnsSuffix

The fully qualified domain name of your organization's internal DNS suffix where any part of the fully qualified domain name in this setting exists in the computer's primary DNS suffix. The **signal** element may contain one or more **dnsSuffix** elements. For example:

```
XML

<dnsSuffix>corp.contoso.com</dnsSuffix>
```

Wi-Fi

You define Wi-Fi signals using one or more wifi elements. Each element has a string value. Wifi elements don't have attributes or nested elements.

SSID

Contains the service set identifier (SSID) of a wireless network. The SSID is the name of the wireless network. The SSID element is required. For example:

```
XML

<ssid>corpnetwifi</ssid>
```

BSSID

Contains the basic service set identifier (BSSID) of a wireless access point. the BSSID is the mac address of the wireless access point. The BSSID element is optional. For example:

```
XML

<bssid>12-ab-34-ff-e5-46</bssid>
```

Security

Contains the type of security the client uses when connecting to the wireless network. The security element is required and must contain one of the following values:

Expand table

Value	Description
Open	The wireless network is an open network that doesn't require any authentication or encryption.
WEP	The wireless network is protected using Wired Equivalent Privacy.
WPA-Personal	The wireless network is protected using Wi-Fi Protected Access.
WPA-Enterprise	The wireless network is protected using Wi-Fi Protected Access-Enterprise.
WPA2-Personal	The wireless network is protected using Wi-Fi Protected Access 2, which typically uses a pre-shared key.
WPA2-Enterprise	The wireless network is protected using Wi-Fi Protected Access 2-Enterprise.

For example:

```
XML

<security>WPA2-Enterprise</security>
```

TrustedRootCA

Contains the thumbprint of the trusted root certificate of the wireless network. You can use any valid trusted root certificate. The value is represented as hexadecimal string, where each byte in the string is separated by a single space. The element is optional. For example:

```
XML

<trustedRootCA>a2 91 34 aa 22 3a a2 3a 4a 78 a2 aa 75 a2 34 2a 3a 11 4a aa</trustedRootCA>
```

Sig_quality

Contains numeric value ranging from 0 to 100 to represent the wireless network's signal strength needed to be considered a trusted signal.

For example:

```
XML
```

```
<sig_quality>80</sig_quality>
```

Sample Trusted Signal Configurations

> **ⓘ Important**
>
> These examples are wrapped for readability. Once properly formatted, the entire XML contents must be a single line.

Example 1

The following example configures an **IPConfig** signal type using **Ipv4Prefix**, **Ipv4DnsServer**, and **DnsSuffix** elements.

XML

```
<rule schemaVersion="1.0">
    <signal type="ipConfig">
        <ipv4Prefix>10.10.10.0/24</ipv4Prefix>
        <ipv4DnsServer>10.10.0.1</ipv4DnsServer>
        <ipv4DnsServer>10.10.0.2</ipv4DnsServer>
        <dnsSuffix>corp.contoso.com</dnsSuffix>
    </signal>
</rule>
```

Example 2

The following example configures an **IpConfig** signal type using a **dnsSuffix** element and a **bluetooth** signal for phones. The example implies that either the IpConfig **or** the Bluetooth rule must evaluate to true, for the resulting signal evaluation to be true.

> **ⓘ Note**
>
> Separate each rule element using a comma.

XML

```
<rule schemaVersion="1.0">
    <signal type="ipConfig">
        <dnsSuffix>corp.contoso.com</dnsSuffix>
    </signal>
```

```
</rule>,
<rule schemaVersion="1.0">
    <signal type="bluetooth" scenario="Authentication" classOfDevice="512"
rssiMin="-10" rssiMaxDelta="-10"/>
</rule>
```

Example 3

The following example configures the same as example 2 using compounding `and` elements. The example implies that the IpConfig **and** the Bluetooth rule must evaluate to true, for the resulting signal evaluation to be true.

XML

```
<rule schemaVersion="1.0">
<and>
  <signal type="ipConfig">
   <dnsSuffix>corp.microsoft.com</dnsSuffix>
  </signal>
  <signal type="bluetooth" scenario="Authentication" classOfDevice="512"
rssiMin="-10" rssiMaxDelta="-10"/>
</and>
</rule>
```

Example 4

The following example configures **Wi-Fi** as a trusted signal.

XML

```
<rule schemaVersion="1.0">
  <signal type="wifi">
    <ssid>contoso</ssid>
    <bssid>12-ab-34-ff-e5-46</bssid>
    <security>WPA2-Enterprise</security>
    <trustedRootCA>a2 91 34 aa 22 3a a2 3a 4a 78 a2 aa 75 a2 34 2a 3a 11 4a
aa</trustedRootCA>
    <sig_quality>80</sig_quality>
  </signal>
</rule>
```

Configure multi-factor unlock

To configure multi-factor unlock you can use:

- Microsoft Intune/CSP
- Group policy

> **Important**
>
> - PIN **must** be in at least one of the groups
> - Trusted signals **must** be combined with another credential provider
> - You can't use the same unlock factor to satisfy both categories. Therefore, if you include any credential provider in bothcategories, it means it can satisfy either category, but not both

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name
Administrative Templates > **Windows Hello for Business**	Device Unlock Plugins

1. Configure first and second unlock factors using the information in Configure Unlock Factors
2. If using trusted signals, configure the trusted signals used by the unlock factor using the information in Configure Signal Rules for the Trusted Signal Credential Provider

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the PassportForWork CSP.

Expand table

Setting
./Device/Vendor/MSFT/PassportForWork/DeviceUnlock

> **Important**
>
> You should remove all non-Microsoft credential providers to ensure users cannot unlock their devices if they do not have the required factors. The fall back options are to use passwords or smart cards (both of which could be disabled as needed).

User experience

Here's a brief video showing the user experience when multi-factor unlock is enabled:

1. The user first signs in with fingerprint + Bluetooth-paired phone
2. The user then signs in with fingerprint + PIN

https://learn-video.azurefd.net/vod/player?id=2bdf21db-30c9-4d8e-99ff-f3ae72c494fe&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Fmultifactor-unlock

Troubleshoot

Multi-factor unlock writes events to event log under **Application and Services Logs\Microsoft\Windows\HelloForBusiness** with the category name **Device Unlock**.

Events

Expand table

Event ID	Details
3520	Unlock attempt initiated
5520	Unlock policy not configured
6520	Warning event
7520	Error event
8520	Success event

Feedback

Was this page helpful? Yes No

Provide product feedback

Remote Desktop sign-in with Windows Hello for Business

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 10

You can use Windows Hello for Business to sign in to a remote desktop session, using the redirected smart card capabilities of the Remote Desktop Protocol (RDP). This is possible by deploying a certificate to the user's device, which is then used as the supplied credential when establishing the RDP connection to another Windows device.

This article describes two certificate deployment approaches, where authentication certificates are deployed to the Windows Hello for Business container:

- Using Microsoft Intune with SCEP or PKCS connectors
- Using an Active Directory Certificate Services (AD CS) enrollment policy

> **ⓘ Important**
>
> If you deploy the certificate using Microsoft Intune, and you have User Account Control configured to *Prompt for credentials on the secure desktop*, you won't be able to use the *run as* feature. In such scenario, when you try to execute an application with elevated privileges and choose the Windows Hello for Business credential, you'll receive the error message: **The username or password is incorrect**.

> **Tip**
>
> Consider using Remote Credential Guard instead of Windows Hello for Business for RDP sign-in. Remote Credential Guard provides single sign-on (SSO) to RDP sessions using Kerberos authentication, and doesn't require the deployment of certificates. For more information, see Remote Credential Guard.

How it works

Windows generates and stores cryptographic keys using a software component called a *key storage provider* (KSP):

- Software-based keys are created and stored using the *Microsoft Software Key Storage Provider*

- Smart card keys are created and stored using the *Microsoft Smart Card Key Storage Provider*
- Keys created and protected by Windows Hello for Business are created and stored using the *Microsoft Passport Key Storage Provider*

A certificate on a smart card starts with creating an asymmetric key pair using the Microsoft Smart Card KSP. Windows requests a certificate based on the key pair from your enterprises issuing certificate authority, which returns a certificate that is stored in the user's Personal certificate store. The private key remains on the smart card and the public key is stored with the certificate. Metadata on the certificate (and the key) stores the key storage provider used to create the key (remember the certificate contains the public key).

The same concept applies to Windows Hello for Business, except that the keys are created using the Microsoft Passport KSP. The user's private key remains protected by the device's security module (TPM) and the user's gesture (PIN/biometric). The certificate APIs hide the complexity. When an application uses a certificate, the certificate APIs locate the keys using the saved key storage provider. The key storage providers direct the certificate APIs on which provider they use to find the private key associated with the certificate. This is how Windows knows you have a smart card certificate without the smart card inserted, and prompts you to insert the smart card.

Windows Hello for Business emulates a smart card for application compatibility, and the Microsoft Passport KSP prompts the user for their biometric gesture or PIN.

> ⓘ **Note**
>
> Remote Desktop with biometric doesn't work with **Dual Enrollment** or scenarios where the user provides alternative credentials.

Requirements

Here's a list of requirements to enable RDP sign-in with Windows Hello for Business:

- ✓ A PKI infrastructure based on AD CS or third-party
- ✓ Windows Hello for Business deployed to the clients
- ✓ If you plan to support Microsoft Entra joined devices, the domain controllers must have a certificate, which serves as a *root of trust* for the clients. The certificate ensures that clients don't communicate with rogue domain controllers

If you plan to deploy certificates using Microsoft Intune, here are more requirements:

- ✔ Ensure you have the infrastructure to support either SCEP or PKCS deployment
- ✔ Deploy the root CA certificate and any other intermediate certificate authority certificates to Microsoft Entra joined Devices using a Trusted root certificate policy

Create a certificate template

The process of creating a certificate template is applicable to scenarios where you use an on-premises Active Directory Certificate Services (AD CS) infrastructure.
You must first create a certificate template, and then deploy certificates based on that template to the Windows Hello for Business container.

The certificate template configuration is different depending on whether you deploy certificates using Microsoft Intune or an Active Directory enrollment policy. Select the option that best suits your needs.

Microsoft Intune

1. Sign in to your issuing certificate authority (CA) and open *Server Manager*
2. Select **Tools** > **Certification Authority**. The Certification Authority Microsoft Management Console (MMC) opens
3. In the MMC, expand the CA name and right-click **Certificate Templates** > **Manage**
4. The Certificate Templates console opens. All of the certificate templates are displayed in the details pane
5. Right-click the **Smartcard Logon** template and select **Duplicate Template**
6. Use the following table to configure the template:

Expand table

Tab Name	Configurations
Compatibility	• Clear the **Show resulting changes** check box • Select **Windows Server 2012 or Windows Server 2012 R2** from the *Certification Authority list* • Select **Windows Server 2012 or Windows Server 2012 R2** from the *Certification Recipient list*
General	• Specify a **Template display name**, for example *WHfB Certificate Authentication*

Tab Name	Configurations
	• Set the validity period to the desired value • Take note of the template name for later, which should be the same as the Template display name minus spaces (*WHfBCertificateAuthentication* in this example)
Extensions	Verify the **Application Policies** extension includes **Smart Card Logon**.
Subject Name	Select **Supply in the request**.
Request Handling	• Set the Purpose to **Signature and smartcard logon** and select **Yes** when prompted to change the certificate purpose • Select the **Renew with same key** check box • Select **Prompt the user during enrollment** **Note:** If you deploy certificates with a PKCS profile, select the option **Allow private key to be exported**
Cryptography	• Set the Provider Category to **Key Storage Provider** • Set the Algorithm name to **RSA** • Set the minimum key size to **2048** • Select **Requests must use one of the following providers** • Select **Microsoft Software Key Storage Provider** • Set the Request hash to **SHA256**
Security	Add the security principal used for SCEP or PKCS **Enroll** access

7. Select **OK** to finalize your changes and create the new template. Your new template should now appear in the list of Certificate Templates
8. Close the Certificate Templates console

Issue the certificate template

1. In the Certificate Authority console, right-click **Certificate Templates**, select **New** > **Certificate Template to Issue**
2. From the list of templates, select the template you previously created (**WHFB Certificate Authentication**) and select **OK**. It can take some time for the template to replicate to all servers and become available in this list
3. After the template replicates, in the MMC, right-click in the Certification Authority list, select **All Tasks** > **Stop Service**. Right-click the name of the CA again, select **All Tasks** > **Start Service**

Deploy certificates

The process of deploying certificates is different depending on whether you use Microsoft Intune or an Active Directory enrollment policy. Select the option that best suits your needs.

Microsoft Intune

This section describes how to configure a SCEP policy in Intune. Similar steps can be followed to configure a PKCS policy.

1. Go to the Microsoft Intune admin center
2. Select **Devices** > **Configuration profiles** > **Create profile**
3. Select **Platform** > **Windows 10 and later** and **Profile type** > **Templates** > **SCEP Certificate**
4. Select **Create**
5. In the *Basics* panel, provide a **Name** and, optionally, a **Description** > **Next**
6. In the *Configuration settings* panel, use the following table to configure the policy:

Expand table

Setting	Configurations
Certificate Type	User
Subject name format	`CN={{UserPrincipalName}}` **Note:** if there's a mismatch between the user UPN suffix and the Active Directory domain FQDN, use `CN={{OnPrem_Distinguished_Name}}` instead.
Subject alternative name	From the dropdown, select **User principal name (UPN)** with a value of `{{UserPrincipalName}}`
Certificate validity period	Configure a value of your choosing
Key storage provider (KSP)	**Enroll to Windows Hello for Business, otherwise fail**

Setting	Configurations
Key usage	**Digital Signature**
Key size (bits)	**2048**
For Hash algorithm	**SHA-2**
Root Certificate	Select **+Root Certificate** and select the trusted certificate profile created earlier for the Root CA Certificate
Extended key usage	• *Name:* **Smart Card Logon** • *Object Identifier:* `1.3.6.1.4.1.311.20.2.2` • *Predefined Values:* **Not configured** • *Name:* **Client Authentication** • *Object Identifier:* `1.3.6.1.5.5.7.3.2` • *Predefined Values:* **Client Authentication**
Renewal threshold (%)	Configure a value of your choosing
SCEP Server URLs	Provide the public endpoint(s) that you configured during the deployment of your SCEP infrastructure

7. Select **Next**
8. In the *Assignments* panel, assign the policy to a security group that contains as members the devices or users that you want to configure and select **Next**
9. In the *Applicability Rules* panel, configure issuance restrictions, if needed, and select **Next**
10. In the *Review + create* panel, review the policy configuration and select **Create**

For more information how to configure SCEP policies, see Configure SCEP certificate profiles in Intune. To configure PKCS policies, see Configure and use PKCS certificate with Intune.

> **⊗ Caution**
>
> If you deploy certificates via Intune and configure Windows Hello for Business via group policy, the devices will fail to obtain a certificate, logging the error code `0x82ab0011` in the `DeviceManagement-Enterprise-Diagnostic-Provider` log.

To avoid the error, configure Windows Hello for Business via Intune instead of group policy.

Use non-Microsoft certification authorities

If you're using a non-Microsoft PKI, the certificate templates published to the on-premises Active Directory may not be available. For guidance with integration of Intune/SCEP with non-Microsoft PKI deployments, refer to Use non-Microsoft certification authorities (CA) with SCEP in Microsoft Intune.

As an alternative to using SCEP, or if none of the previously covered solutions work in your environment, you can manually generate Certificate Signing Requests (CSR) for submission to your PKI. To assist with this approach, you can use the Generate-CertificateRequest PowerShell commandlet.

The `Generate-CertificateRequest` commandlet generates an `.inf` file for a pre-existing Windows Hello for Business key. The `.inf` can be used to generate a certificate request manually using `certreq.exe`. The commandlet also generates a `.req` file, which can be submitted to your PKI for a certificate.

Verify that the certificate is deployed

To verify that the certificate is correctly deployed to the Windows Hello for Business container, use the following command:

Windows Command Prompt

```
certutil -store -user my
```

The output lists keys and certificates stored in the user store. If a certificate issued from your CA is deployed to the Windows Hello for Business container, the output displays the certificate with a `Provider` value of `Microsoft Passport Key Storage Provider`.

For example:

Windows Command Prompt

```
C:\Users\amanda.brady>certutil -store -user my
my "Personal"
================ Certificate 0 ================
Serial Number: 110000001f4c4eccc46fc8f93a00000000001f
Issuer: CN=Contoso - Issuing CA, DC=CONTOSO, DC=COM
```

```
 NotBefore: 12/8/2023 6:16 AM
 NotAfter: 12/7/2024 6:16 AM
Subject: CN=amanda.brady@contoso.com
Non-root Certificate
Template:
1.3.6.1.4.1.311.21.8.2835349.12167323.7094945.1118853.678601.83.11484210.800
5739
Cert Hash(sha1): 63c6ce5fc512933179d3c0a5e94ecba98092f93d
Key Container = S-1-12-1-../../login.windows.net/../amanda.brady@contoso.com
Provider = Microsoft Passport Key Storage Provider
Private key is NOT exportable
Encryption test passed
```

User experience

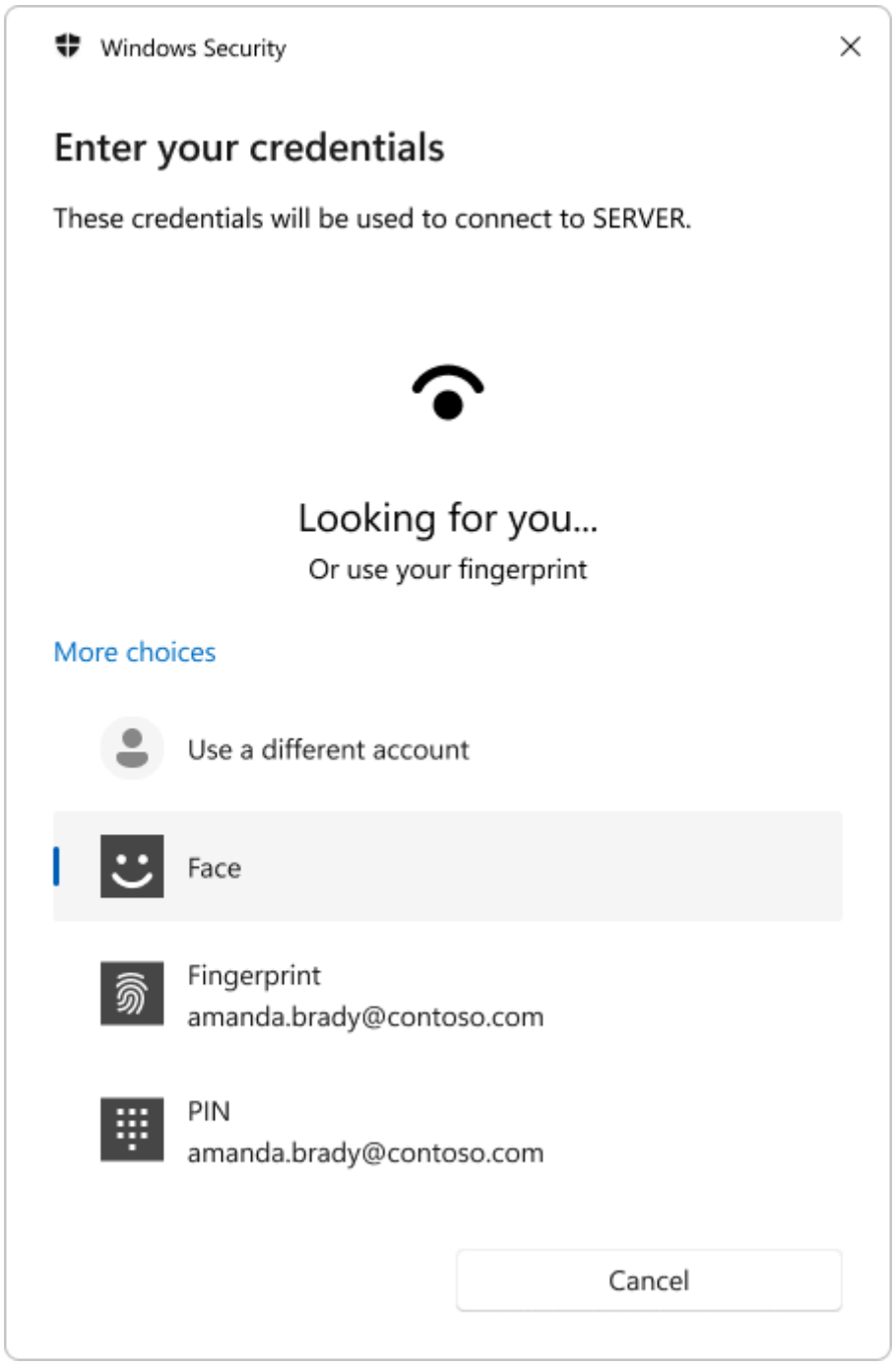
Once users obtain their certificate, they can RDP to any Windows devices in the same Active Directory forest as the users' Active Directory account by opening Remote Desktop Connection (`mstsc.exe`). When connecting to the remote host, they're prompted to use Windows Hello for Business to unlock the private key of the certificate.

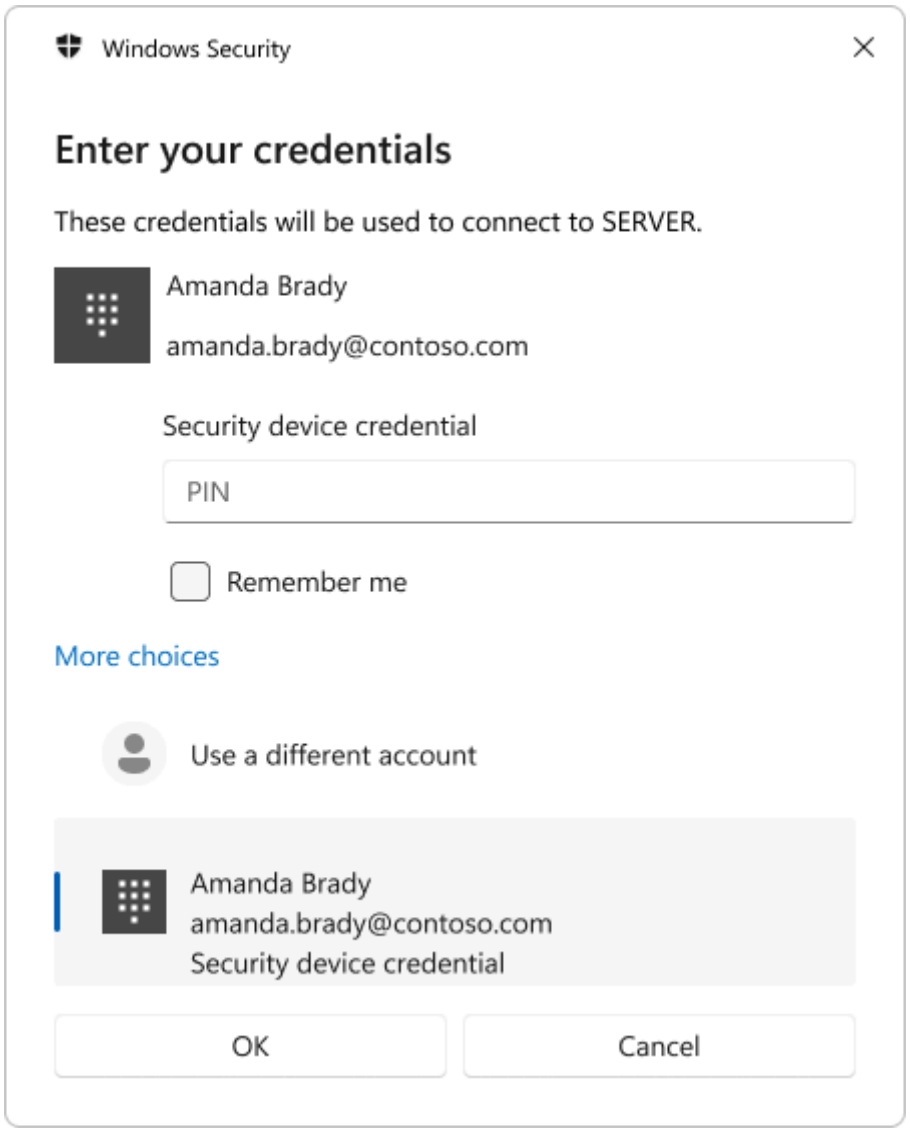
Microsoft Entra joined device

The user can authenticate using any available Windows Hello unlock gestures, including biometrics.

Microsoft Entra hybrid joined device

The credential prompt identifies the Windows Hello credential provider as *Security device credential*. The user must use the PIN credential provider to unlock.


Windows Security
Enter your credentials
These credentials will be used to connect to SERVER.
Looking for you...
Or use your fingerprint
More choices
Use a different account
Face
Fingerprint
amanda.brady@contoso.com
PIN
amanda.brady@contoso.com
Cancel



Here's a brief video showing the user experience from a Microsoft Entra joined device using fingerprint as unlock factor: https://learn-video.azurefd.net/vod/player?id=b6e1038d-98b5-48dc-8afb-65523d12cfaf&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fhello-for-business%2Frdp-sign-in

> **Note**
>
> The user must be authorized to connect to the remote server using the Remote Desktop protocol, for example by being a member of the Remote Desktop Users local group on the remote host.

Compatibility

While users appreciate the convenience of biometrics, and administrators value the security, you might experience compatibility issues with applications and Windows Hello

for Business certificates. In such scenarios, you can deploy a policy setting to revert to the previous behavior for the users needing it.

For more information, see Use Windows Hello for Business certificates as smart card certificate

Known issues

There's a known issue when attempting to perform TLS 1.3 client authentication with a Hello certificate via RDP. The authentication fails with the error: `ERR_SSL_CLIENT_AUTH_SIGNATURE_FAILED`. Microsoft is investigating possible solutions.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Hello for Business known deployment issues

Article • 11/22/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

The content of this article is to help troubleshoot known deployment issues for Windows Hello for Business.

PIN reset on Microsoft Entra join devices fails with *We can't open that page right now* error

PIN reset on Microsoft Entra joined devices uses a flow called *web sign-in* to authenticate the user above lock. Web sign in only allows navigation to specific domains. If web sign-in attempts to navigate to a domain that isn't allowed, it displays a page with the error message *We can't open that page right now*.

Identify PIN Reset allowed domains issue

The user can launch the PIN reset flow from the lock screen using the *I forgot my PIN* link in the PIN credential provider. Selecting the link launches a full screen UI for the PIN experience on Microsoft Entra join devices. Typically, the UI displays an authentication page, where the user authenticates using Microsoft Entra credentials and completes MFA.

In federated environments, authentication may be configured to route to AD FS or a non-Microsoft identity provider. If the PIN reset flow is launched and attempts to navigate to a federated identity provider server page, it fails and displays the *We can't open that page right now* error, if the domain for the server page isn't included in an allowlist.

If you're a customer of *Azure US Government* cloud, PIN reset also attempts to navigate to a domain that isn't included in the default allowlist. The result is the message *We can't open that page right now*.

Resolve PIN Reset allowed domains issue

To resolve the error, you can configure a list of allowed domains for PIN reset using the ConfigureWebSignInAllowedUrls policy. For information on how to configure the policy, see Configure allowed URLs for federated identity providers on Microsoft Entra joined devices.

Hybrid key trust sign in broken due to user public key deletion

Applies to:

- Hybrid key trust deployments
- Windows Server 2016, builds 14393.3930 to 14393.4048
- Windows Server 2019, builds 17763.1457 to 17763.1613

In Hybrid key trust deployments with domain controllers running certain builds of Windows Server 2016 and Windows Server 2019, the user's Windows Hello for Business key is deleted after they sign-in. Subsequent sign-ins will fail until the user's key is synced during the next Microsoft Entra Connect delta sync cycle.

Identify user public key deletion issue

After the user provisions a Windows Hello for Business credential in a hybrid key trust environment, the key must sync from Microsoft Entra ID to Active Directory during a Microsoft Entra Connect Sync cycle. The user's public key is written to the `msDS-KeyCredentialLink` attribute of the user object.

Before the user's Windows Hello for Business key syncs, sign-ins with Windows Hello for Business fail with the error message *That option is temporarily unavailable. For now, please use a different method to sign in.* After the key syncs successfully, the user can sign in and unlock with their PIN or enrolled biometrics.

In environments with the issue, after the first sign-in with Windows Hello for Business and provisioning is complete, the next sign-in attempt fails. In environments where domain controllers are running a mix of builds, some users may be impacted by the issue, and subsequent sign in attempts may be sent to different domain controllers. The result is intermittent sign-in failures.

After the initial sign-in attempt, the user's Windows Hello for Business public key is deleted from the `msDS-KeyCredentialLink attribute`. You can verify the deletion by querying a user's `msDS-KeyCredentialLink` attribute before and after sign-in. You can query the `msDS-KeyCredentialLink` in AD using Get-ADUser and specifying `msds-keycredentiallink` for the `-Properties` parameter.

Resolve user public key deletion issue

To resolve the issue, update Windows Server 2016 and 2019 domain controllers with the latest patches. For Windows Server 2016, the behavior is fixed in build *14393.4104* (KB4593226) and later. For Windows Server 2019, the behavior is fixed in build *17763.1637* (KB4592440).

Microsoft Entra joined device access to on-premises resources using key trust and non-Microsoft Certificate Authority (CA)

Applies to:

- Microsoft Entra joined key trust deployments
- Non-Microsoft certificate authority (CA) issuing domain controller certificates

Windows Hello for Business uses smart-card based authentication for many operations. This type of authentication has special guidelines when using a non-Microsoft CA for certificate issuance, some of which apply to the domain controllers. Not all Windows Hello for Business deployment types require these configurations. Accessing on-premises resources from a Microsoft Entra joined device does require special configuration when using a non-Microsoft CA to issue domain controller certificates.

For more information, read Guidelines for enabling smart card sign in with non-Microsoft certification authorities.

Identify on-premises resource access issues with non-Microsoft CAs

The issue can be identified using network traces or Kerberos logging from the client. In the network trace, the client fails to place a `TGS_REQ` request when a user attempts to access a resource. On the client, it can be observed in the Kerberos operation event log under `Application and Services/Microsoft/Windows/Security-Kerberos/Operational`. The logs are disabled by default. The failure event for this case includes the following information:

Console

```
Log Name:      Microsoft-Windows-Kerberos/Operational
Source:        Microsoft-Windows-Security-Kerberos
Event ID:      107
GUID:          {98e6cfcb-ee0a-41e0-a57b-622d4e1b30b1}
Task Category: None
Level:         Error
```

```
Keywords:
User:          SYSTEM
Description:

The Kerberos client received a KDC certificate that does not have a matched
domain name.
Expected Domain Name: ad.contoso.com
Error Code: 0xC000006D
```

Resolve on-premises resource access issue with non-Microsoft CAs

To resolve the issue, domain controller certificates must be updated so that the certificate subject contains the directory path of the server object (distinguished name). Example Subject: `CN=DC1,OU=Domain Controllers,DC=ad,DC=contoso,DC=com`

Alternatively, you can set the subject alternative name (SAN) of the domain controller certificate to contain the server object's fully qualified domain name and the NETBIOS name of the domain. Example Subject Alternative Name:

> `dns=dc1.ad.contoso.com`
>
> `dns=ad.contoso.com`
>
> `dns=ad`

Key trust authentication broken for Windows Server 2019

Applies to:

- Windows Server 2019
- Hybrid key trust deployments
- On-premises key trust deployments

Domain controllers running early versions of Windows Server 2019 have an issue that prevents key trust authentication from working properly. Networks traces report *KDC_ERR_CLIENT_NAME_MISMATCH*.

Identify Windows Server 2019 key trust authentication issue

On the client, authentication with Windows Hello for Business fails with the error message, *That option is temporarily unavailable. For now, please use a different method*

to sign in.

The error is presented on Microsoft Entra hybrid joined devices in key trust deployments after Windows Hello for Business is provisioned, but before a user's key is synced from Microsoft Entra ID to Active Directory. If a user's key isn't synced from Microsoft Entra ID and the `msDS-keycredentiallink` attribute on the user object in AD is populated for NGC, then it's possible that the error occurs.

Another indicator of the failure can be identified using network traces. If you capture network traces for a key trust sign-in event, the traces show Kerberos failing with the error *KDC_ERR_CLIENT_NAME_MISMATCH*.

Resolve Server 2019 key trust authentication issue

The issue is resolved in Windows Server 2019, build *17763.316* (KB4487044). Upgrade all Windows Server 2019 domain controllers to the build *17763.316* or newer to resolve the issue.

Certificate trust provisioning with AD FS broken on windows server 2019

Applies to:

- Windows Server 2019
- Hybrid certificate trust deployments
- On-premises certificate trust deployments

AD FS running on Windows Server 2019 fails to complete device authentication due to an invalid check of incoming scopes in the request. Device authentication to AD FS is a requirement for Windows Hello for Business to enroll a certificate using AD FS. The client blocks Windows Hello for Business provisioning until the authentication is successful.

Identify certificate trust with AD FS 2019 enrollment issue

The provisioning experience for Windows Hello for Business launches if the prerequisite checks are successful. The result of the provisioningAdmin checks is available in event logs under **Microsoft-Windows-User Device Registration**. If provisioning is blocked because device authentication doesn't succeed, event ID *362* is logged stating *User has successfully authenticated to the enterprise STS: No*.

Console

```
Log Name:      Microsoft-Windows-User Device Registration/Admin
Source:        Microsoft-Windows-User Device Registration
Date:          <Date and time>
Event ID:      362
Task Category: None
Level:         Warning
Keywords:
User:          <User SID>
Computer:      <Computer name>
Description:
Windows Hello for Business provisioning will not be launched.
Device is Azure Active Directory-joined ( AADJ or DJ++ ): Yes
User has logged on with Azure Active Directory credentials: Yes
Windows Hello for Business policy is enabled: Yes
Windows Hello for Business post-logon provisioning is enabled: Yes
Local computer meets Windows hello for business hardware requirements: Yes
User is not connected to the machine via Remote Desktop: Yes
User certificate for on premise auth policy is enabled: Yes
Enterprise user logon certificate enrollment endpoint is ready: Not Tested
Enterprise user logon certificate template is : No ( 1 : StateNoPolicy )
User has successfully authenticated to the enterprise STS: No
Certificate enrollment method: enrollment authority
See https://go.microsoft.com/fwlink/?linkid=832647 for more details.
```

If a device recently joined a domain, there may be a delay before the device authentication occurs. If the failing state of this prerequisite check persists, then it can indicate an issue with the AD FS configuration.

If the AD FS scope issue is present, event logs on the AD FS server indicate an authentication failure from the client. The error is logged in event logs under **AD FS/Admin** as event ID *1021* and the event specifies that the client is forbidden access to resource `http://schemas.microsoft.com/ws/2009/12/identityserver/selfscope` with scope `ugs`:

Console

```
Log Name:      AD FS/Admin
Source:        AD FS
Date:          <Date and time>
Event ID:      1021
Task Category: None
Level:         Error
Keywords:      AD FS
User:          <ADFS service Account>
Computer:      <Date and time>
Description:
Encountered error during OAuth token request.
Additional Data
```

```
Exception details:
Microsoft.IdentityServer.Web.Protocols.OAuth.Exceptions.OAuthUnauthorizedCli
entException: MSIS9368: Received invalid OAuth request. The client
'38aa3b87-a06d-4817-b275-7a316988d93b' is forbidden to access the resource
'http://schemas.microsoft.com/ws/2009/12/identityserver/selfscope' with
scope 'ugs'.
       at
Microsoft.IdentityServer.Web.Protocols.OAuth.OAuthProtocolContext.ValidateSc
opes(String scopeParameter, String clientId, String relyingPartyId)
       at
Microsoft.IdentityServer.Web.Protocols.OAuth.OAuthToken.OAuthJWTBearerReques
tContext.ValidateCore()
```

Resolve certificate trust with AD FS 2019 enrollment issue

This issue is fixed in Windows Server, version 1903 and later. For Windows Server 2019, the issue can be remediated by adding the ugs scope manually.

1. Launch AD FS management console. Browse to **Services** > **Scope Descriptions**.
2. Right select **Scope Descriptions** and select **Add Scope Description**.
3. Under name type *ugs*, and select **Apply** > **OK**.
4. Launch PowerShell as an administrator.
5. Get the ObjectIdentifier of the application permission with the ClientRoleIdentifier parameter equal to "38aa3b87-a06d-4817-b275-7a316988d93b":

   PowerShell

   ```
   (Get-AdfsApplicationPermission -ServerRoleIdentifiers
   'http://schemas.microsoft.com/ws/2009/12/identityserver/selfscope' | ?{
   $_.ClientRoleIdentifier -eq '38aa3b87-a06d-4817-b275-7a316988d93b'
   }).ObjectIdentifier
   ```

6. Execute the command `Set-AdfsApplicationPermission -TargetIdentifier <ObjectIdentifier from step 5> -AddScope 'ugs'`.
7. Restart the AD FS service.
8. On the client: Restart the client. The user should be prompted to provision Windows Hello for Business.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows Hello errors during PIN creation

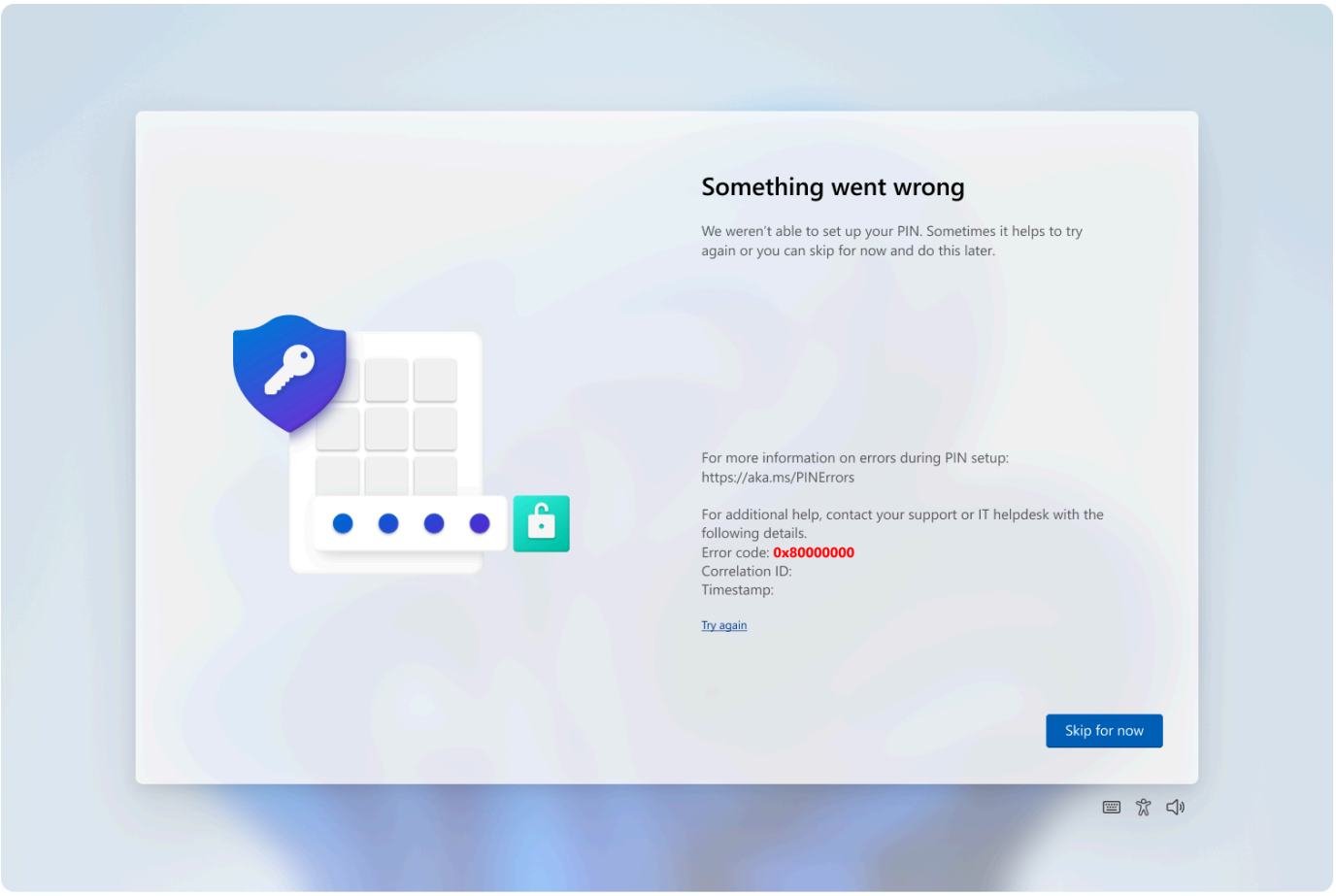
Applies to: ✅ Windows 11, ✅ Windows 10

When you set up Windows Hello in Windows client, you might get an error during the **Create a PIN** step. This article lists some of the error codes with recommendations for mitigating the problem.

> **Important**
>
> If you get an error code that isn't listed here, contact Microsoft Support.

Where is the error code?

The following image shows an example of an error during **Create a PIN**.



Error mitigations

When a user encounters an error when creating the work PIN, advise the user to try the following steps. Many errors can be mitigated by one of these steps.

1. Try to create the PIN again. Some errors are transient and resolve themselves
2. Sign out, sign in, and try to create the PIN again
3. Reboot the device and then try to create the PIN again
4. Unjoin the device from Microsoft Entra ID, rejoin, and then try to create the PIN again. To unjoin a device, go to **Settings** > **System** > **About** > **Disconnect from organization**

If the error occurs again, check the error code against the following table to see if there's another mitigation for that error. When no mitigation is listed in the table, contact Microsoft Support for assistance.

Expand table

Hex	Cause	Mitigation
0x80090005	NTE_BAD_DATA	Unjoin the device from Microsoft Entra ID and rejoin.
0x8009000F	The container or key already exists.	Unjoin the device from Microsoft Entra ID and rejoin.
0x80090011	The container or key wasn't found.	Unjoin the device from Microsoft Entra ID and rejoin.
0x80090029	TPM isn't set up.	Sign on with an administrator account. Select **Start**, type `tpm.msc`, and select **tpm.msc Microsoft Common Console Document**. In the **Actions** pane, select **Prepare the TPM**.
0x8009002A	NTE_NO_MEMORY	Close programs, which are taking up memory and try again.
0x80090031	NTE_AUTHENTICATION_IGNORED	Reboot the device. If the error occurs again after rebooting, reset the TPM or run Clear-TPM.
0x80090035	Policy requires TPM and the device doesn't have TPM.	Change the Windows Hello for Business policy to not require a TPM.
0x80090036	User canceled an interactive dialog.	User is asked to try again.
0x801C0003	User isn't authorized to enroll.	Check if the user has permission to perform the operation.
0x801C000E	Registration quota reached.	Unjoin some other device that is currently joined using the same account or increase the maximum number of devices per user.
0x801C000F	Operation successful, but the device requires a reboot.	Reboot the device.
0x801C0010	The AIK certificate isn't valid or trusted.	Sign out and then sign in again.
0x801C0011	The attestation statement of the transport key is invalid.	Sign out and then sign in again.
0x801C0012	Discovery request isn't in a valid format.	Sign out and then sign in again.
0x801C0015	The device is required to be joined to an Active Directory domain.	Join the device to an Active Directory domain.
0x801C0016	The federation provider configuration is empty	Go to http://clientconfig.microsoftonline-p.net/FPURL.xml and verify that the file isn't empty.
0x801C0017	The federation provider domain is empty	Go to http://clientconfig.microsoftonline-p.net/FPURL.xml and verify that the FPDOMAINNAME element isn't empty.
0x801C0018	The federation provider client configuration URL is empty	Go to http://clientconfig.microsoftonline-p.net/FPURL.xml and verify that the CLIENTCONFIG element contains a valid URL.
0x801C03E9	Server response message is invalid	Sign out and then sign in again.
0x801C03EA	Server failed to authorize user or device.	Check if the token is valid and user has permission to register Windows Hello for Business keys.
0x801C03EB	Server response http status isn't valid	Sign out and then sign in again.
0x801C03EC	Unhandled exception from server.	sign out and then sign in again.
0x801C03ED	Multi-factor authentication is required for a 'ProvisionKey' operation, but wasn't performed. -or-	Sign out and then sign in again. If that doesn't resolve the issue, unjoin the device from Microsoft Entra ID and rejoin. Allow user(s) to join to Microsoft Entra ID under Microsoft Entra Device settings.

Hex	Cause	Mitigation
	Token wasn't found in the Authorization header. -or- Failed to read one or more objects. -or- The request sent to the server was invalid. -or- User doesn't have permissions to join to Microsoft Entra ID.	
0x801C03EE	Attestation failed.	Sign out and then sign in again.
0x801C03EF	The AIK certificate is no longer valid.	Sign out and then sign in again.
0x801C03F2	Windows Hello key registration failed.	ERROR_BAD_DIRECTORY_REQUEST. Another object with the same value for property proxyAddresses already exists. To resolve the issue, refer to Duplicate Attributes Prevent Dirsync. Also, if no sync conflict exists, please verify that the "Mail/Email address" in Microsoft Entra ID and the Primary SMTP address are the same in the proxy address.
0x801C044D	Authorization token doesn't contain device ID.	Unjoin the device from Microsoft Entra ID and rejoin.
	Unable to obtain user token.	Sign out and then sign in again. Check network and credentials.
0x801C044E	Failed to receive user credentials input.	Sign out and then sign in again.
0x801C0451	User token switch account.	Delete the Web Account Manager token broker files located in `%LOCALAPPDATA%\Packages\Microsoft.AAD.BrokerPlugin_cw5n1h2txyewy\AC\TokenBroker\Accounts\*.*\` and reboot.
0xC00000BB	Your PIN or this option is temporarily unavailable.	The destination domain controller doesn't support the login method. Most often the KDC service doesn't have the proper certificate to support the login. Another common cause can be the client cannot verify the KDC certificate CRL. Use a different login method.

Errors with unknown mitigation

For errors listed in this table, contact Microsoft Support for assistance.

Expand table

Hex	Cause
0x80070057	Invalid parameter or argument is passed.
0X80072F0C	Unknown
0x80072F8F	A mismatch happens between the system's clock and the activation server's clock when attempting to activate Windows.
0x80090010	NTE_PERM
0x80090020	NTE_FAIL
0x80090027	Caller provided a wrong parameter. If non-Microsoft code receives this error, they must change their code.

Hex	Cause
0x8009002D	NTE_INTERNAL_ERROR
0x801C0001	ADRS server response isn't in a valid format.
0x801C0002	Server failed to authenticate the user.
0x801C0006	Unhandled exception from server.
0x801C000B	Redirection is needed and redirected location isn't a well known server.
0x801C000C	Discovery failed.
0x801C0013	Tenant ID isn't found in the token.
0x801C0014	User SID isn't found in the token.
0x801C0019	The federation provider client configuration is empty
0x801C001A	The DRS endpoint in the federation provider client configuration is empty.
0x801C001B	The device certificate isn't found.
0x801C03F0	There's no key registered for the user.
0x801C03F1	There's no UPN in the token.
0x801C044C	There's no core window for the current thread.
0x801c004D	DSREG_NO_DEFAULT_ACCOUNT: NGC provisioning is unable to find the default WAM account to use to request Microsoft Entra token for provisioning. Unable to enroll a device to use a PIN for login.
0xCAA30193	HTTP 403 Request Forbidden: it means request left the device, however either Server, proxy or firewall generated this response.

Last updated on 03/29/2026

Windows Hello for Business policy settings

Applies to: ✅ Windows 11, ✅ Windows 10

This reference article provides a comprehensive list of policy settings for Windows Hello for Business. The list of settings is sorted alphabetically and organized in four categories:

- **Feature settings**: used to enable Windows Hello for Business and configure basic options
- **PIN setting**: used to configure PIN authentication, like PIN complexity and recovery
- **Biometric setting**: used to configure biometric authentication
- **Smart card settings**: used to configure smart card authentication used in conjunction with Windows Hello for Business

For information about how to configure these settings, see Configure Windows Hello for Business.

Select one of the tabs to see the list of available settings:

Feature settings

Expand table

Setting Name	CSP	GPO
Configure device unlock factors	✅	✅
Configure dynamic lock factors	✅	✅
Use a hardware security device	✅	✅
Use certificate for on-premises authentication	✅	✅
Use cloud (Kerberos) trust for on-premises authentication	✅	✅
Use Windows Hello for Business	✅	✅

Configure device unlock factors

Configure a comma separated list of credential provider GUIDs, such as face and fingerprint provider GUIDs, to be used as the first and second unlock factors. If the trusted signal provider is specified as one of the unlock factors, you should also configure a comma separated list of signal rules in the form of xml for each signal type to be verified.

If you enable this policy setting, the user must use one factor from each list to successfully unlock. If you disable or don't configure this policy setting, users can continue to unlock with existing options.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/`DeviceUnlock
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**

For more information, see Multi-factor unlock.

Configure dynamic lock factors

Configure a comma separated list of signal rules in the form of xml for each signal type.

- If you enable this policy setting, the signal rules are evaluated to detect user absence and automatically lock the device
- If you disable or don't configure the setting, users can continue to lock with existing options

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/DynamicLock/`DynamicLock
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**

Use a hardware security device

A Trusted Platform Module (TPM) provides additional security benefits over software because data protected by it can't be used on other devices.

- If you enable this policy setting, Windows Hello for Business provisioning only occurs on devices with usable 1.2 or 2.0 TPMs. You can optionally exclude TPM revision 1.2 modules, which prevents Windows Hello for Business provisioning on those devices

> **Tip**

The TPM 1.2 specification only allows the use of RSA and the SHA-1 hashing algorithm. TPM 1.2 implementations vary in policy settings, which may result in support issues as lockout policies vary. It's recommended to exclude TPM 1.2 devices from Windows Hello for Business provisioning. -If you disable or don't configure this policy setting, the TPM is still preferred, but all devices can provision Windows Hello for Business using software if the TPM is nonfunctional or unavailable.

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/`RequireSecurityDevice `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/ExcludeSecurityDevices/`TPM12
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**

Use certificate for on-premises authentication

Use this policy setting to configure Windows Hello for Business to enroll a sign-in certificate used for on-premises authentication.

- If you enable this policy setting, Windows Hello for Business enrolls a sign-in certificate that is used for on-premises authentication
- If you disable or don't configure this policy setting, Windows Hello for Business will use a key or a Kerberos ticket (depending on other policy settings) for on-premises authentication

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/`UseCertificateForOnPremAuth
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business** **User Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**

Use cloud trust for on-premises authentication

Use this policy setting to configure Windows Hello for Business to use the cloud Kerberos trust model.

- If you enable this policy setting, Windows Hello for Business uses a Kerberos ticket retrieved from authenticating to Microsoft Entra ID for on-premises authentication
- If you disable or don't configure this policy setting, Windows Hello for Business uses a key or certificate (depending on other policy settings) for on-premises authentication

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/`UseCloudTrustForOnPremAuth
GPO	**Computer Configuration** > **Administrative Templates** > **Windows Components** > **Windows Hello for Business**

> **Note**
>
> Cloud Kerberos trust is incompatible with certificate trust. If the certificate trust policy setting is enabled, it takes precedence over this policy setting.

Use Windows Hello for Business

- If you enable this policy, the device provisions Windows Hello for Business using keys or certificates for all users
- If you disable this policy setting, the device doesn't provision Windows Hello for Business for any user
- If you don't configure this policy setting, users can provision Windows Hello for Business

Select the option *Don't start Windows Hello provisioning after sign-in* when you use a non-Microsoft solution to provision Windows Hello for Business:

- If you select *Don't start Windows Hello provisioning after sign-in*, Windows Hello for Business doesn't automatically start provisioning after the user has signed in
- If you don't select *Don't start Windows Hello provisioning after sign-in*, Windows Hello for Business automatically starts provisioning after the user has signed in

Expand table

	Path
CSP	`./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/`UsePassportForWork `./Device/Vendor/MSFT/PassportForWork/{TenantId}/Policies/`DisablePostLogonProvisioning
GPO	**Computer Configuration > Administrative Templates > Windows Components > Windows Hello for Business** **User Configuration > Administrative Templates > Windows Components > Windows Hello for Business**

Last updated on 03/29/2026

WebAuthn APIs for passwordless authentication on Windows

Applies to: ✅ Windows 11, ✅ Windows 10

Passwords can leave your customers vulnerable to data breaches and security attacks by malicious users.

Microsoft has long been a proponent of passwordless authentication, and has introduced the W3C/Fast IDentity Online 2 (FIDO2) Win32 WebAuthn platform APIs in Windows 10 (version 1903).

Starting in **Windows 11, version 22H2**, WebAuthn APIs support ECC algorithms and starting in **Windows 11 version 24H2** WebAuthn APIs support plugin passkey managers.

What does this mean?

By using WebAuthn APIs, developer partners and the developer community can use Windows Hello or FIDO2 Security Keys to implement passwordless multi-factor authentication for their applications on Windows devices.

Users of these apps or sites can use any browser that supports WebAuthn APIs for passwordless authentication. Users will have a familiar and consistent experience on Windows, no matter which browser they use.

Developers should use the WebAuthn APIs to support FIDO2 authentication keys in a consistent way for users. Additionally, developers can use all the transports that are available per FIDO2 specifications (USB, NFC, and BLE) while avoiding the interaction and management overhead.

> ⓘ **Note**
>
> When these APIs are in use, Windows 10 browsers or applications don't have direct access to the FIDO2 transports for FIDO-related messaging.

Passkey Managers

The WebAuthn Plugin APIs support management operations for passkeys (FIDO2 credentials) in native Win32 applications. Plugin passkey managers can create passkey, receive callbacks for credential deletion, manage credential metadata, and autofill credentials. The passkey manager

support enables Windows users to use their synced passkeys across browsers and apps on Windows with their chosen passkey managers. The support includes:

- Adding the plugin authenticator using the `WebAuthNPluginAddAuthenticator` API. This registers the provider as an authenticator and is viewable in Settings > Accounts > Passkeys > Advanced options.
- Handling credential creation requests from WebAuthn clients.
- Managing credential metadata with `WebAuthNPluginAuthenticatorAddCredentials` and `WebAuthNPluginAuthenticatorRemoveCredentials`.
- Performing user verification using Windows Hello with `WebAuthNPluginPerformUserVerification`.
- Updating plugin authenticator details with `WebAuthNPluginUpdateAuthenticatorDetails`.
- Removing the plugin authenticator with `WebAuthNPluginRemoveAuthenticator`.
- Responding to system-initiated operations such as `CancelOperation` and rendering the plugin status with a callback.

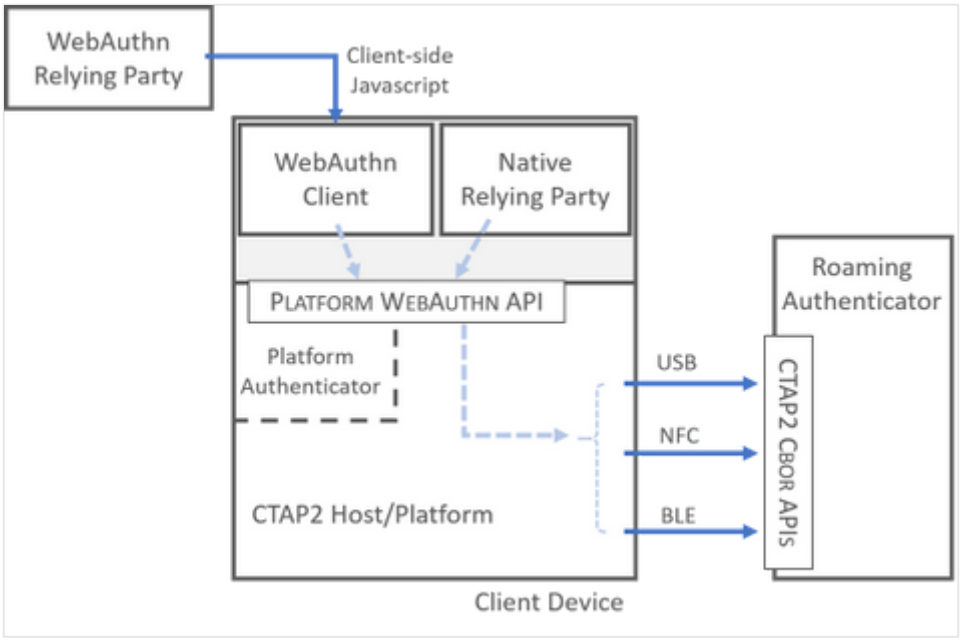
Cross Device Authentication

The Client to Authenticator Protocol 2 (CTAP2) and WebAuthn define an abstraction layer that creates an ecosystem for strongly authenticated credentials. In this ecosystem, any interoperable client (such as a native app or browser) that runs on a given client device uses a standardized method to interact with any interoperable authenticator. Interoperable authenticators include authenticators that are built into the client device (platform authenticators) and authenticators that connect to the client device by using USB, BLE, or NFC connections (roaming authenticators).

The authentication process starts when the user makes a specific user gesture that indicates consent for the operation. At the request of the client, the authenticator securely creates strong cryptographic keys and stores them locally.

After these client-specific keys are created, clients can request attestations for registration and authentication. The type of signature that the private key uses reflects the user gesture that was made.

The following diagram shows how CTAP and WebAuthn interact. The light blue dotted arrows represent interactions that depend on the specific implementation of the platform APIs.



Relationships of the components that participate in passwordless authentication

A combined WebAuthn/CTAP2 dance includes the following cast of characters:

- **Client device**. The *client device* is the hardware that hosts a given strong authentication. Laptops and phones are examples of client devices.
- **Relying parties and clients**. *Relying parties* are web or native applications that consume strong credentials. The relying parties run on client devices.
    - As a relying party, a native application can also act as a WebAuthn client to make direct WebAuthn calls.
    - As a relying party, a web application can't directly interact with the WebAuthn API. The relying party must broker the deal through the browser.

    > ⓘ **Note**
    >
    > The preceding diagram doesn't depict Single Sign-On (SSO) authentication. Be careful not to confuse FIDO relying parties with federated relying parties.

- **WebAuthn API**. The *WebAuthn API* enables clients to make requests to authenticators. The client can request the authenticator to create a key, provide an assertion about a key, report capabilities, manage a PIN, and so on.
- **CTAP2 platform/host**. The *platform* (also called the host in the CTAP2 spec) is the part of the client device that negotiates with authenticators. The platform is responsible for securely reporting the origin of the request and for calling the CTAP2 Concise Binary Object Representation (CBOR) APIs. If the platform isn't CTAP2-aware, the clients

themselves take on more of the burden. In this case, the components and interactions shown in the preceding diagram may differ.

- **Platform authenticator**. A *platform authenticator* usually resides on a client device. Examples of platform authenticators include fingerprint recognition technology that uses a built-in laptop fingerprint reader and facial recognition technology that uses a built-in smartphone camera. Cross-platform transport protocols such as USB, NFC or BLE can't access platform authenticators.

- **Roaming authenticator**. A *roaming authenticator* can connect to multiple client devices. Client devices must use a supported transport protocol to negotiate interactions. Examples of roaming authenticators include USB security keys, BLE-enabled smartphone applications, and NFC-enabled proximity cards. Roaming authenticators can support CTAP1, CTAP2, or both protocols.

Many relying parties and clients can interact with many authenticators on a single client device. A user can install multiple browsers that support WebAuthn, and might simultaneously have access to a built-in fingerprint reader, a plugged-in security key, and a BLE-enabled mobile application.

Interoperability

Before WebAuthn and CTAP2, there were U2F and CTAP1. U2F is the FIDO Alliance universal second-factor specification. There are many authenticators that speak CTAP1 and manage U2F credentials. WebAuthn was designed to be interoperable with CTAP1 Authenticators. A relying party that uses WebAuthn can still use U2F credentials if the relying party doesn't require FIDO2-only functionality.

FIDO2 authenticators have already been implemented and WebAuthn relying parties might require the following optional features:

- Keys for multiple accounts (keys can be stored per relying party)
- Client PIN
- Location (the authenticator returns a location)
- Hash-based Message Authentication Code (HMAC)-secret (enables offline scenarios)

The following options might be useful in the future, but haven't been observed in the wild yet:

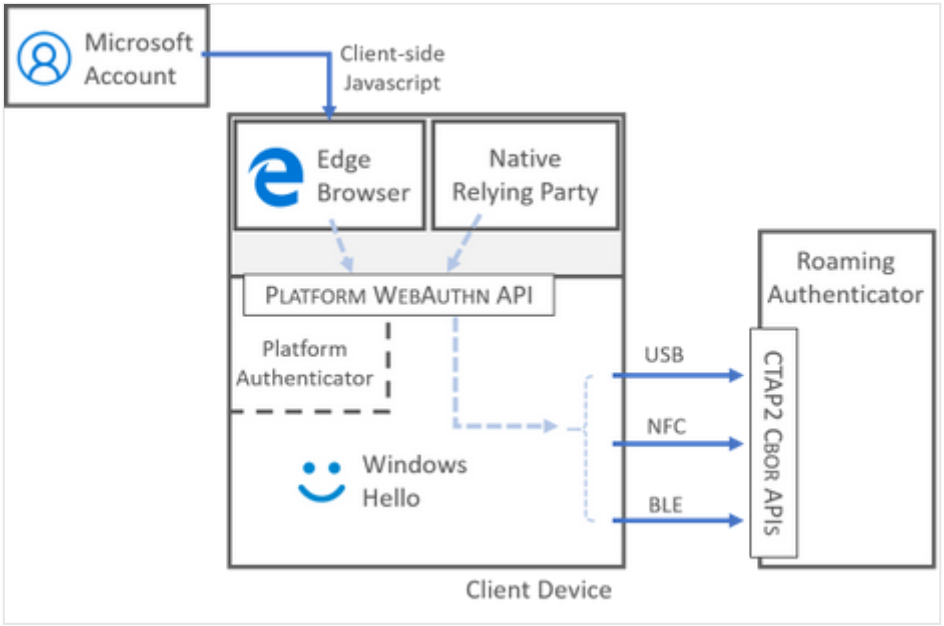
- Transactional approval
- User verification index (servers can determine whether biometric data that's stored locally has changed over time)
- User verification method (the authenticator returns the exact method)

- Biometric performance bounds (the relying party can specify acceptable false acceptance and false rejection rates)

Microsoft implementation

The Microsoft FIDO2 implementation has been years in the making. Software and services are implemented independently as standards-compliant entities. As of the Windows 10, version 1809 (October 2018) release, all Microsoft components use the latest WebAuthn Candidate Release. It's a stable release that's not expected to normatively change before the specification is finally ratified. Because Microsoft is among the first in the world to deploy FIDO2, some combinations of popular non-Microsoft components won't be interoperable yet.

Here's an approximate layout of where the Microsoft bits go:



Microsoft's implementation of WebAuthn and CTAP2 APIs

- **WebAuthn relying party: Microsoft Account**. If you aren't familiar with Microsoft Account, it's the sign-in service for Xbox, Outlook, and many other sites. The sign-in experience uses client-side JavaScript to trigger Microsoft Edge to talk to the WebAuthn APIs. Microsoft Account requires that authenticators have the following characteristics:
  - Keys are stored locally on the authenticator and not on a remote server
  - Offline scenarios work (enabled by using HMAC)
  - Users can put keys for multiple user accounts on the same authenticator
  - If it's necessary, authenticators can use a client PIN to unlock a TPM

> **ⓘ Important**

Because Microsoft Account requires features and extensions that are unique to FIDO2 CTAP2 authenticators, it doesn't accept CTAP1 (U2F) credentials.

- **WebAuthn client: Microsoft Edge**. Microsoft Edge can handle the user interface for the WebAuthn and CTAP2 features that this article describes. It also supports the AppID extension. Microsoft Edge can interact with both CTAP1 and CTAP2 authenticators. This scope for interaction means that it can create and use both U2F and FIDO2 credentials. However, Microsoft Edge doesn't speak the U2F protocol. Therefore, relying parties must use only the WebAuthn specification. Microsoft Edge on Android doesn't support WebAuthn.

  > **Note**
  >
  > For authoritative information about Microsoft Edge support for WebAuthn and CTAP, see Legacy Microsoft Edge developer documentation.

- **Platform: Windows 10, Windows 11**. Windows 10 and Windows 11 host the Win32 Platform WebAuthn APIs.

- **Roaming Authenticators**. You might notice that there's no *Microsoft* roaming authenticator. The reason is because there's already a strong ecosystem of products that specialize in strong authentication, and every customer (whether corporations or individuals) has different requirements for security, ease of use, distribution, and account recovery. For more information on the ever-growing list of FIDO2-certified authenticators, see FIDO Certified Products. The list includes built-in authenticators, roaming authenticators, and even chip manufacturers who have certified designs.

Developer references

The WebAuthn APIs are documented in the Microsoft/webauthn GitHub repo. To understand how FIDO2 authenticators work, review the following two specifications:

- Web Authentication: An API for accessing Public Key Credentials (available on the W3C site). This document is known as the WebAuthn spec.
- Client to Authenticator Protocol (CTAP). This document is available at the FIDO Alliance site, on which hardware and platform teams are working together to solve the problem of FIDO authentication.
- Passkey manager sample. This sample shows how to integrate a plugin passkey manager with the Windows platform.

Last updated on 12/18/2025

Windows Hello Enhanced Sign-in Security

Applies to: ✅ Windows 11, ✅ Windows 10

Summarize this article for me

Windows Hello enables biometrics or PIN authentication, eliminating the need for a password. Biometric authentication uses facial recognition or fingerprint to prove a user's identity in a way that's secure, personal, and convenient.

Enhanced Sign-in Security (ESS) provides an additional level of security to biometric data with the use of specialized hardware and software components. Virtualization Based Security (VBS) and Trusted Platform Module 2.0 are used to isolate and protect user's authentication data, and to secure the data communication channel.

How does Enhanced Sign-in Security protect biometric data

ESS and facial recognition

When ESS is enabled, the face algorithm is protected using VBS to isolate it from the rest of Windows. The hypervisor is used to specify and protect memory regions, so that they can only be accessed by processes running in VBS. The hypervisor allows the face camera to write to these memory regions providing an isolated pathway to deliver face data from the camera to the face matching algorithm.

Face templates are generated in VBS by the protected face algorithm. When not in use, face template data is encrypted using keys generated and only accessible to VBS, and then stored on disk.

ESS and fingerprint recognition

ESS is only supported on fingerprint sensors with match on sensor capabilities. This type of sensor includes a microprocessor and memory which can be used to isolate fingerprint matching and template storage using hardware.

Sensors that support ESS have a certificate embedded during manufacturing. The certificate can be validated by the Windows biometric components running in VBS, and is used to establish a secure session with the sensor. The sensor and Windows biometric components use the session to communicate enrollment operations and match results securely.

Credential operations

The Windows biometric components running in VBS establish a secure channel to the TPM using information shared with VBS by the TPM during boot. When a matching operation is a success, the biometric components in VBS use the secure channel to authorize the usage of Windows Hello keys for authenticating the user with their identity provider, applications, and services.

Enable Enhanced Sign-in Security

The enablement of ESS is dependent on specialized hardware, drivers, and firmware pre-installed on the system. Device manufacturers can choose to enable Enhanced Sign-in Security during device configuration in the factory.

Users can also set up ESS on their devices through Settings. Compatible sensor is required for ESS enrollment.

> **Note**
>
> All Copilot+ PCs have ESS enabled by default. Some Copilot+ PCs may not have built in ESS sensors. For more information, see **Copilot+ PC hardware requirements**.

System requirements

Compatible hardware and software components are required to enable Enhanced Sign-in Security:

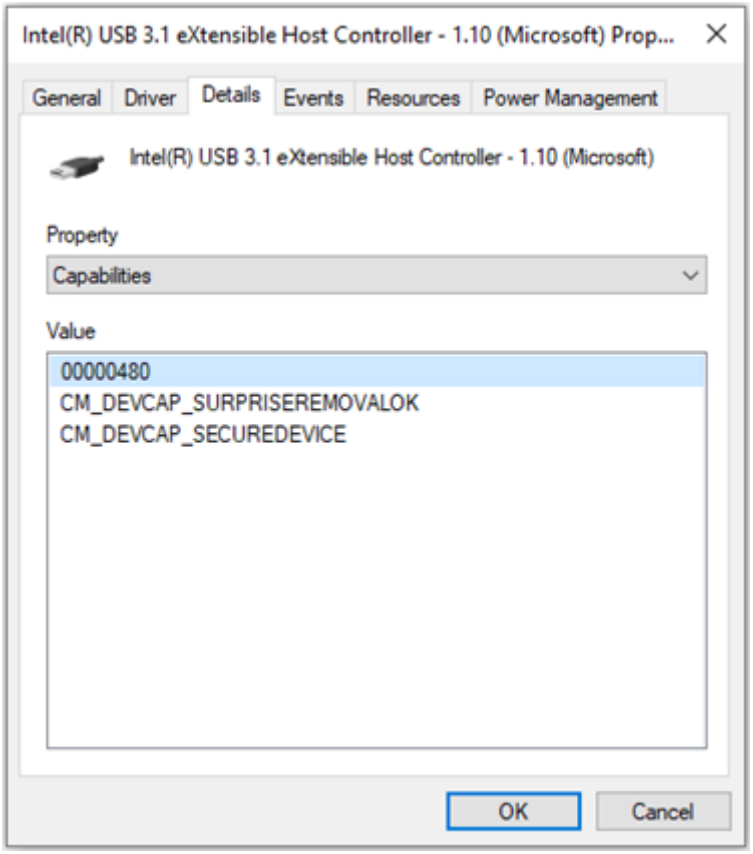
- Meet the requirements for Virtualization-Based Security (VBS), including Device Guard Enablement and Trusted Platform Module 2.0
- Biometric sensor hardware that supports ESS
- Biometric sensor drivers compatible with ESS
- Device firmware with a Secure Devices (SDEV) ACPI table configured by the device manufacturer for the included biometric hardware (ESS Face only)

Biometric sensor compatibility

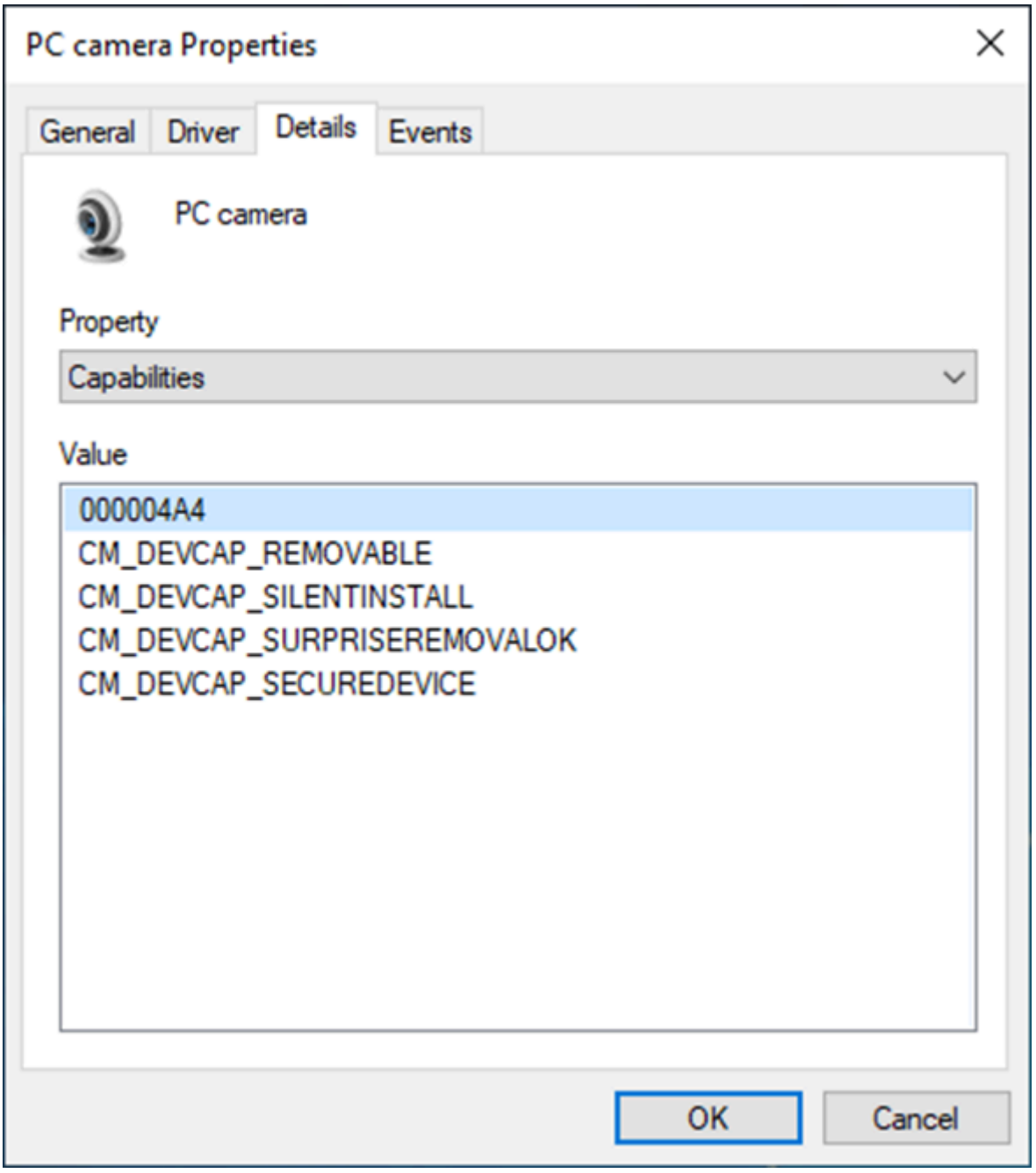
Face biometric sensor

ESS is designed to work with a select range of IR cameras, and it requires specific chipsets. Cameras that support ESS must have this feature built into their firmware, and using the standard Windows UVC camera driver that comes with the operating system is necessary.

To check if the camera module is ESS-capable, first go to Device Manager and expand the **Universal Serial Bus controllers** section. Right-click on the device that has **eXtensible Host Controller** in the name, and select the **Properties** option to view the device properties. If there are multiple entries for a host controller, check the properties section for all. Navigate to the **Details** tab of the driver and select **Capabilities** from the **Property** drop down menu. One of the devices should show the `CM_DEVCAP_SECUREDEVICE` capability.



Next, check the properties sections of the PC Cameras by going to the **Cameras** section in Device Manager. If there are multiple entries for PC Cameras, check the properties section for all. Navigate to the **Details** tab of the drivers and select **Capabilities** from the **Property** drop down menu. One of the PC camera devices should have the `CM_DEVCAP_SECUREDEVICE` capability.

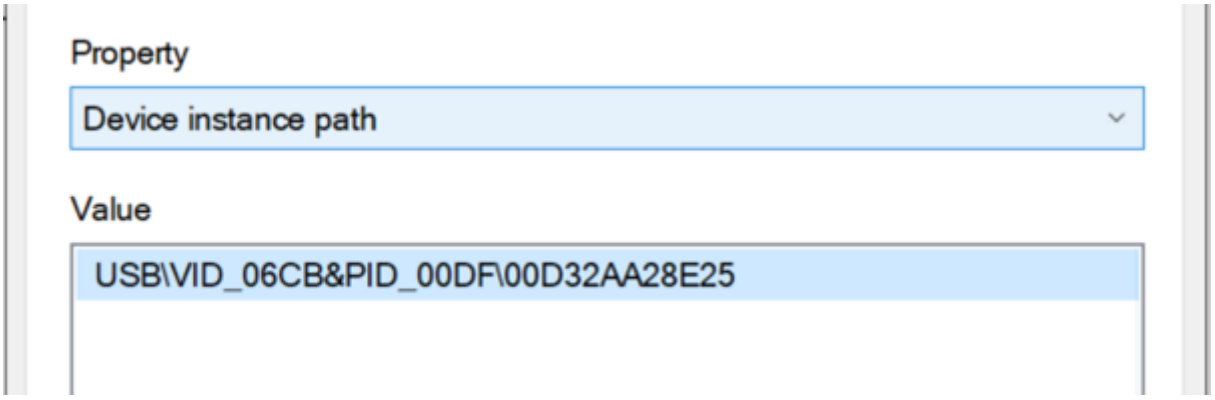


Fingerprint biometric sensor

ESS-capable fingerprint sensors must be match on chip:

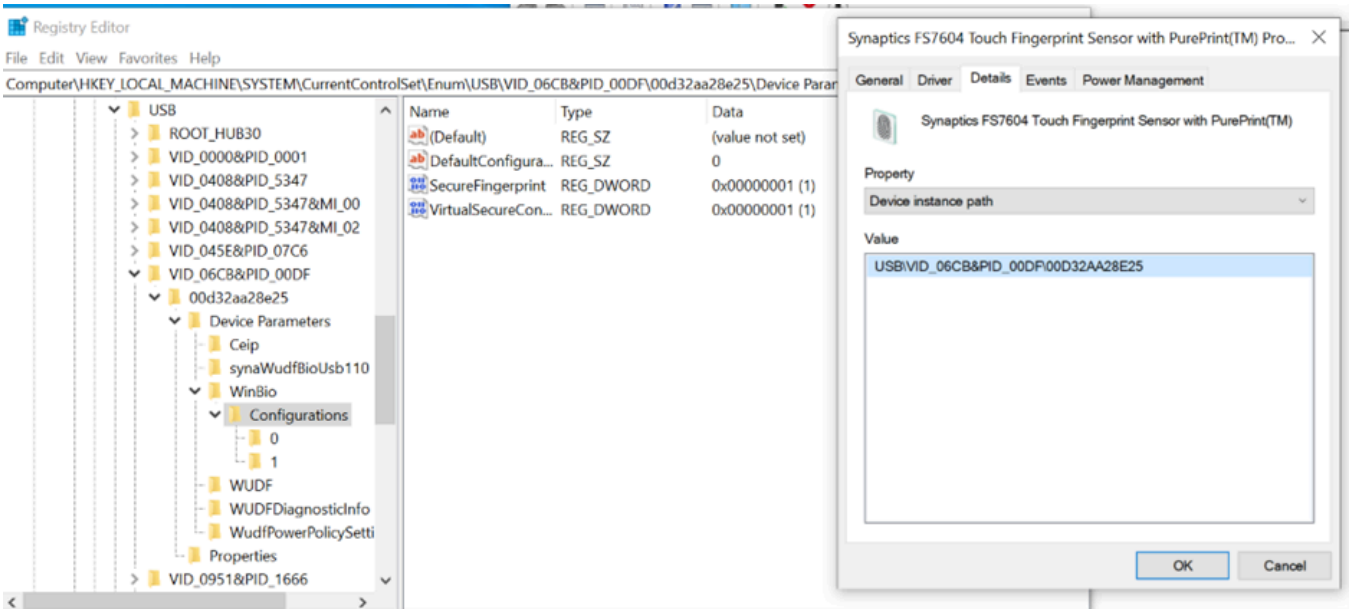
- The sensor must have a Microsoft-issued certificate burned into the device during manufacturing
- The device driver and firmware must support Enhanced Sign-in Security functionality

To check if a fingerprint module is ESS-capable, first go to Device Manager and expand the **Biometric devices** section. There should be an entry for a fingerprint sensor. Right-click the fingerprint reader entry and go to **Properties** > **Details**. Under the **Property** option, select **Device instance path**.



Open regedit.exe and navigate to `HKLM\SYSTEM\CurrentControlSet\Enum[DeviceInstancePath]\Device Parameters\WinBio\Configurations` where `DeviceInstancePath` is the path listed in Device Manager. Select **Configurations**. There should be a registry key listed named `SecureFingerprint` with a data value of `1`. If it doesn't exist, the device isn't secure-capable.

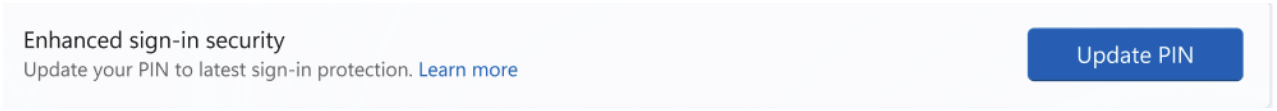
Configurations should also have two folders beneath it: one labeled `0` and one labeled `1`. If there's only one folder and not two, the device isn't secure-capable.



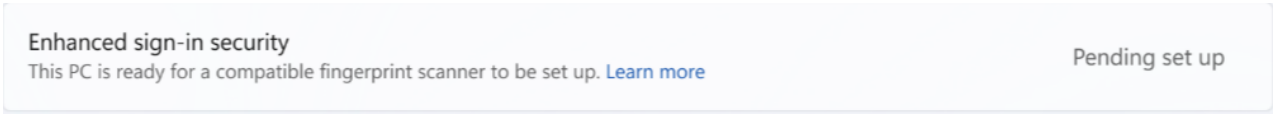
Pluggable/peripheral fingerprint scanner

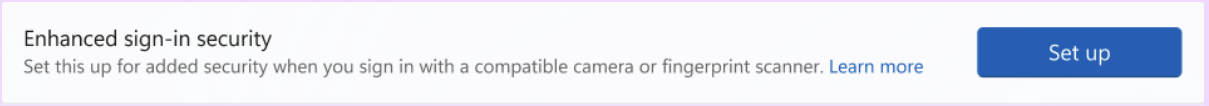
Peripheral ESS fingerprint scanner are supported on capable hardware and supported OS. When a supported ESS fingerprint scanner is introduced to a Windows machine running a supported build:

1. Navigate to Settings > Accounts > Sign-in options > Additional settings
2. If your PIN needs a refresh, you will first be prompted to “Update PIN” under the “Enhanced sign-in security” setting.



3. Otherwise, check if the device is in a good ESS state by confirming the "Enhanced sign-in security" setting says "Pending set up."



> ⓘ Note
>
> If your device isn't already in an ESS ready state, you'll be prompted to update your login security to ESS. On update, existing non-ESS enrollments and associated credentials, such as passkeys, will be removed to ensure a secure configuration. These can be reprovisioned after set up.
>
> 
>

4. Enroll your fingerprint using the ESS fingerprint scanner, and ESS will be enabled.

> ⓘ Note
>
> ESS is a system-level setting. On multi-user devices, the lowest Windows Hello security posture applies. Admins can upgrade the device to ESS, but all users will need to re-enroll upon their next login.

Verify if ESS is enabled

Windows Settings

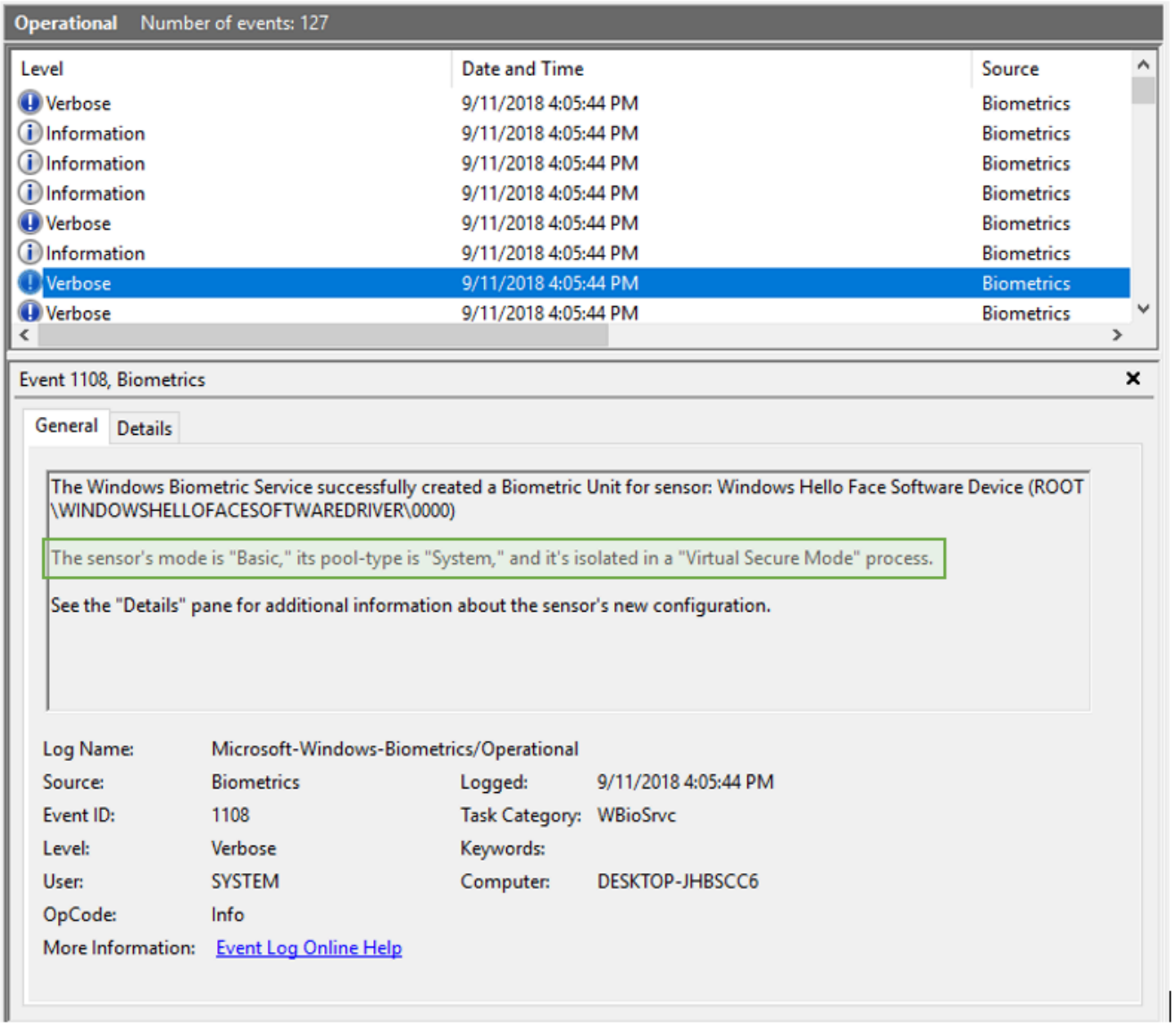
If ESS is enabled, the Sign-in options page of Windows Settings will have an Enhanced sign-in security entry that demonstrates as "On."

Event Viewer

The Windows biometric framework generates logs events when each sensor on a system is enumerated. These logs include information indicating whether a sensor is operating with Enhanced Sign-in Security enabled. Biometric event logs are found in Event Viewer under **Event**

Viewer > **Applications and Services Logs** > **Microsoft** > **Windows** > **Biometrics** > **Operational**.

If the biometric device is loaded properly by the Windows biometric framework, there's a log event ID `1108` for the corresponding sensor. If the device is operating with ESS enabled, the sensor is specified as *isolated* in a **Virtual Secure Mode** process. If the device isn't using ESS, it's specified as *isolated* in a **System** process.



In the event `1108`, cameras are described using *Windows Hello Face Software Device (ROOT\WINDOWSHELLOFACESOFTWAREDRIVER\0000)* and fingerprint devices are described using the device's specific module and device ID. For fingerprint devices, the device ID is listed in device Manager under **Biometric Devices** > **[Fingerprint Module]** > **Properties** > **Details** > **Device Instance Path**.

Application Compatibility

For devices with ESS-capable cameras, a Secure Devices (SDEV) table is required. When an SDEV table is implemented and VBS is turned on, the SDEV table is parsed by the Secure Kernel and restrictions are enforced on accessing Peripheral Component Interconnect (PCI) device configuration space. These restrictions are enacted to prevent malicious processes from manipulating the configuration space of secured devices specified in the SDEV table.

Applications that attempt to read/write the PCI configuration space, except by means explicitly supported by Windows, result in bug checks when the SDEV table is parsed and enforced.

All drivers and software included in the device image must be tested for compatibility, given these software restrictions. Software or drivers that are distributed to the system via Windows Update, the Microsoft Store, or other acceptable channels by the device manufacturer should also be checked for compatibility. Without this verification, there might be unexpected behavior on the system.

Unsupported scenarios

Non-ESS supported sensors

When ESS is enabled, only biometric sensors that support ESS work on the system. All noncapable sensors won't be enumerated by the Windows Biometric framework.

If there are any concerns around biometric modalities being blocked, contact the device manufacturer for support.

Pluggable/peripheral camera modules

ESS isn't supported for external camera modules. With ESS enabled, non-ESS biometric sensor operations are blocked. If you would like to use a non-ESS peripheral to sign in with Windows Hello, see Disable/Enable ESS

Wake on touch for fingerprint sensors

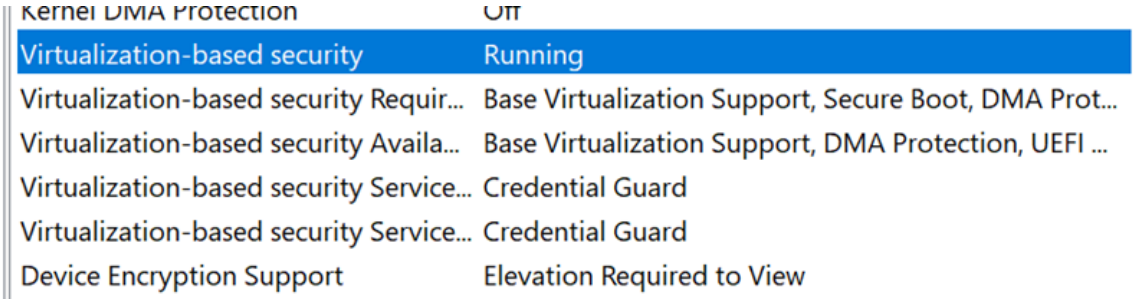
Wake on Touch (WoT) is the ability for the fingerprint sensor to wake the system and sign the user in without requiring the user to touch the sensor twice. Devices that support Modern Standby enable Wake on Touch sensor behavior.

Starting in Windows 11, version 22H2 with KB5027303, WoT is available for ESS devices.

Troubleshooting

Face/Fingerprint authentication isn't working

If biometric authentication isn't working, first check that VBS is running and that the secure component started. To check if VBS is running, open **System Information** > **System Summary**. There should be an entry for **Virtualization Based Security** listed as *Running*.



Also check that the biometric isolation trust-lets are running. These should be listed under **System Information** > **Software Environment** > **Running Tasks** as `bioiso.exe` and `ngciso.exe`. If either of these checks fail, the system might not meet the requirements for Enhanced Sign-in Security. Attempt to restart the biometric service using step #3.

1. In **Settings** > **Sign-in Options**, remove the nonfunctioning enrollment and re-enroll
   a. if the entry for Windows Hello Face/Fingerprint is unavailable with the condition *We couldn't find a fingerprint scanner compatible with Windows Hello Face*, or something similar, proceed to the next step
2. In device manager, the sensor should be listed under **Biometrics devices**. Reinstall the driver by right clicking the name of the device and select **Uninstall Device**. Restart the device, at which point Windows attempts to reinstall the driver. Check if authentication is working
3. To restart the biometric service, first remove PIN from the system by going to **Sign-in Options** and removing PIN. Open a command prompt as administrator and enter `net stop wbiosrvc && net start wbiosrvc`. Check if fingerprint authentication is working
4. If biometrics is still not working on the device, file a feedback item using Feedback Hub

To check that the secure connection succeeded, refer to the Verify if ESS is enabled section.

PIN isn't working

PIN can be reset in the lock screen under **Sign-in options**. To do so, remove PIN and add it again. This will prompt PIN reset, which should restore PIN functionality.

Disable/Enable ESS

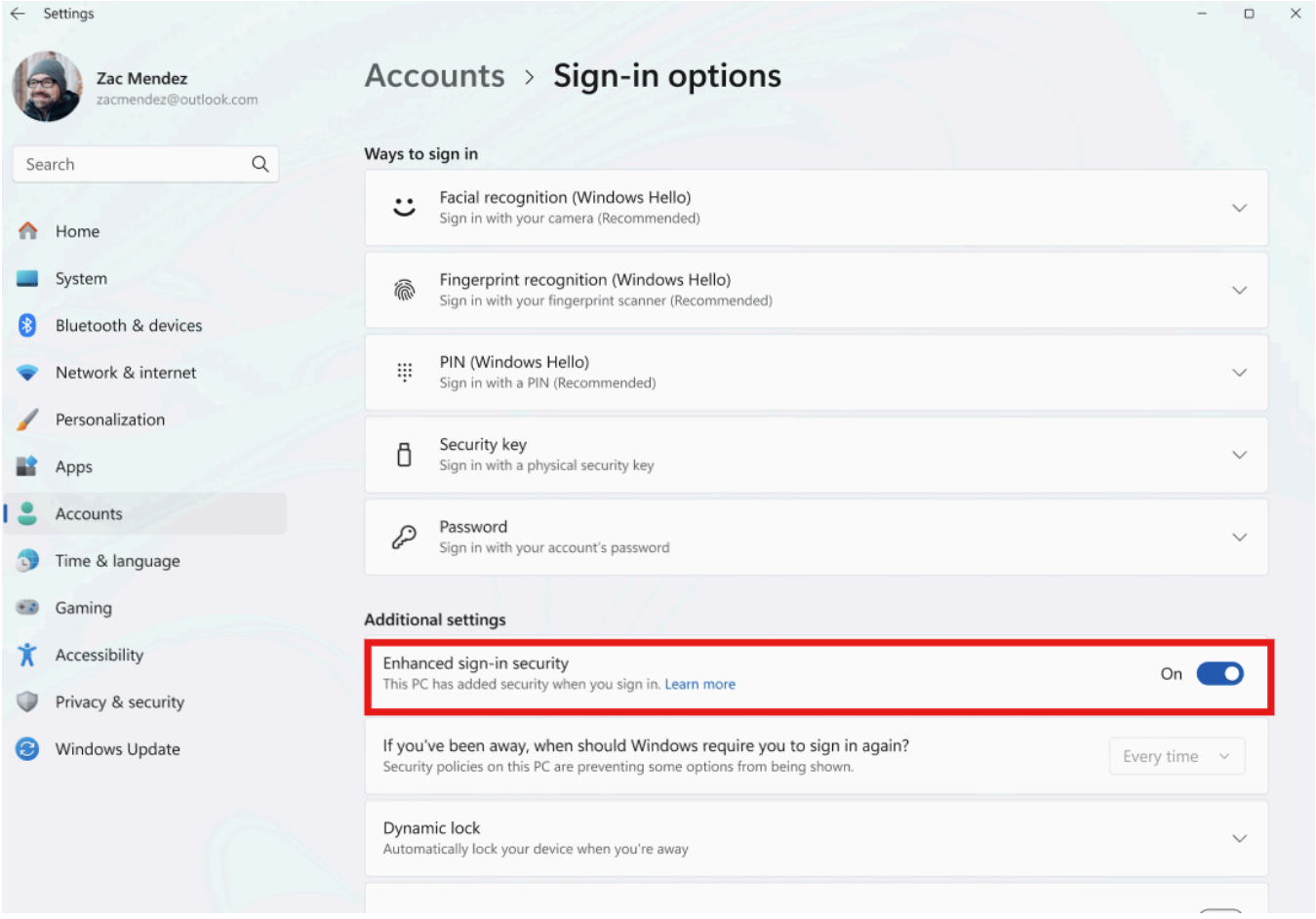
Starting in Windows 11, version 22H2 with KB5031455, users can temporarily turn off ESS if they would like to use an external peripheral to authenticate with Windows Hello on their device.

You can use the Settings app to disable ESS. Select **Start** > **Settings** > **Accounts** > **Sign-in options** or use the following shortcut:

Sign-in options

On devices running Windows 11, version 24H2 or newer, under **Additional settings** > **Enhanced sign-in security**, there's a toggle that allows you to enable or disable ESS if you have relevant sensors.

- When the toggle is **Off**, ESS is disabled and you can use non-ESS Windows Hello compatible peripherals to sign in.
- When the toggle is **On,** ESS is enable and you can't use non-ESS Windows Hello compatible peripherals to sign in. Remember, you can still use external peripherals within apps like Teams.
- If you have no toggle but see **Enhanced sign-in security** as **Pending set up**, the first sensor you enroll will determine your ESS state. For example, if you first enroll an ESS sensor, you will have ESS On. If you enroll a non-ESS sensor first, you will have ESS Off.



On devices running Windows 11, version 23H2, under **Additional settings** > **Sign in with an external camera or fingerprint reader**, there's a toggle that allows you to enable or disable ESS:

- When the toggle is **Off**, ESS is enabled and you can't use external peripherals to sign in. Remember, you can still use external peripherals within apps like Teams
- When the toggle is **On**, ESS is disabled and you can use Windows Hello compatible peripherals to sign in

---

Last updated on 02/19/2026

Common questions about Windows Hello for Business

Windows Hello for Business replaces password sign-in with strong authentication, using an asymmetric key pair. This Frequently Asked Questions (FAQ) article is intended to help you learn more about Windows Hello for Business.

Concepts

What's the difference between Windows Hello and Windows Hello for Business?

Windows Hello is an authentication technology that allows users to sign in to their Windows devices using biometric data, or a PIN, instead of a traditional password.

Windows Hello for Business is an extension of Windows Hello that provides enterprise-grade security and management capabilities, including device attestation, certificate-based authentication, and conditional access policies. Policy settings can be deployed to devices to ensure they're secure and compliant with organizational requirements.

Why a PIN is better than an online password

Three main reasons:

1. **A PIN is tied to a device**: one important difference between an online password and a Hello PIN is that the PIN is tied to the specific device on which it's set up. That PIN is useless to anyone without that specific hardware. Someone who obtains your online password can sign in to your account from anywhere, but if they obtain your PIN, they'd have to access your device too. The PIN can't be used anywhere except on that specific device. If you want to sign in on multiple devices, you have to set up Hello on each device
2. **A PIN is local to the device**: an online password is transmitted to the server. The password can be intercepted in transmission or obtained from a server. A PIN is local to the device, never transmitted anywhere, and it isn't stored on the server. When the PIN is created, it establishes a trusted relationship with the identity provider and creates an asymmetric key pair that is used for authentication. When you enter your PIN, you unlock the authentication key, which is used to sign the request that is sent to the authenticating

server. With Windows Hello for Business, the PIN is user-provided entropy used to load the private key in the Trusted Platform Module (TPM). The server doesn't have a copy of the PIN. For that matter, the Windows client doesn't have a copy of the current PIN either. The user must provide the entropy, the TPM-protected key, and the TPM that generated that key in order to successfully access the private key

3. **A PIN is backed by hardware**: the Hello PIN is backed by a Trusted Platform Module (TPM) chip, which is a secure crypto-processor that is designed to carry out cryptographic operations. The chip includes multiple physical security mechanisms to make it tamper resistant, and malicious software is unable to tamper with the security functions of the TPM. Windows doesn't link local passwords to TPM, therefore PINs are considered more secure than local passwords. User key material is generated and available within the TPM of the device. The TPM protects the key material from attackers who want to capture and reuse it. Since Hello uses asymmetric key pairs, users credentials can't be stolen in cases where the identity provider or websites the user accesses have been compromised. The TPM protects against various known and potential attacks, including PIN brute-force attacks. After too many incorrect guesses, the device is locked

The statement *A PIN is stronger than a password* is not directed at the strength of the entropy used by the PIN. It's about the difference between providing entropy versus continuing the use of a symmetric key (the password). The TPM has anti-hammering features that thwart brute-force PIN attacks (an attacker's continuous attempt to try all combination of PINs). Some organizations may worry about shoulder surfing. For those organizations, rather than increase the complexity of the PIN, implement the Multifactor Unlock feature.

What if someone steals the device?

To compromise a Windows Hello credential that TPM protects, an attacker must have access to the physical device. Then, the attacker must find a way to spoof the user's biometrics or guess the PIN. All these actions must be done before TPM anti-hammering protection locks the device.

Why do you need a PIN to use biometrics?

Windows Hello enables biometric sign-in with fingerprint, iris, or facial recognition. When you set up Windows Hello, you're asked to create a PIN after the biometric setup. The PIN enables you to sign in when you can't use your preferred biometric because of an injury or because the sensor is unavailable or not working properly. If you only had a biometric sign-in configured and, for any reason, were unable to use that method to sign in, you would have to sign in using

your account and password, which doesn't provide you with the same level of protection as Hello.

How are keys protected?

Anytime key material is generated, it must be protected against attack. The most robust way to do this is through specialized hardware. There's a long history of using hardware security modules (HSMs) to generate, store, and process keys for security-critical applications. Smart cards are a special type of HSM, as are devices that are compliant with the Trusted Computing Group TPM standard. Wherever possible, the Windows Hello for Business implementation takes advantage of onboard TPM hardware to generate and protect keys. Administrators can choose to allow key operations in software, but it's recommended the use of TPM hardware. The TPM protects against a variety of known and potential attacks, including PIN brute-force attacks. The TPM provides an additional layer of protection after an account lockout, too. When the TPM has locked the key material, the user will have to reset the PIN (which means the user will have to use MFA to reauthenticate to the IdP before the IdP allows re-registration). Resetting the PIN means that all keys and certificates encrypted with the old key material will be removed.

How does PIN caching work with Windows Hello for Business?

Windows Hello for Business provides a PIN caching user experience by using a ticketing system. Rather than caching a PIN, processes cache a ticket they can use to request private key operations. Microsoft Entra ID and Active Directory sign-in keys are cached under lock. This means the keys remain available for use without prompting, as long as the user is interactively signed-in. Microsoft Account sign-in keys are transactional keys, which means the user is always prompted when accessing the key.

Windows Hello for Business used as a smart card (smart card emulation that is enabled by default) provides the same user experience of default smart card PIN caching. Each process requesting a private key operation prompts the user for the PIN on first use. Subsequent private key operations won't prompt the user for the PIN.

The smart card emulation feature of Windows Hello for Business verifies the PIN and then discards the PIN in exchange for a ticket. The process doesn't receive the PIN, but rather the ticket that grants them private key operations. There isn't a policy setting to adjust the caching.

Where is Windows Hello biometrics data stored?

When you enroll in Windows Hello, a representation of your biometrics, called an enrollment profile, is created. The enrollment profile biometrics data is device specific, is stored locally on the device, and does not leave the device or roam with the user. Some external fingerprint sensors store biometric data on the fingerprint module itself rather than on Windows device. Even in this case, the biometrics data is stored locally on those modules, is device specific, doesn't roam, never leaves the module, and is never sent to Microsoft cloud or external server. For more details, see Windows Hello biometrics in the enterprise and Windows Hello face authentication.

What is the format used to store Windows Hello biometrics data on the device?

Windows Hello biometrics data is stored on the device as an encrypted template database. The data from the biometrics sensor (like face camera or fingerprint reader) creates a data representation—or graph—that is then encrypted before it's stored on the device. Each biometrics sensor on the device which is used by Windows Hello (face or fingerprint) will have its own biometric database file where template data is stored. Each biometrics database file is encrypted with unique, randomly generated key that is encrypted to the system using AES encryption producing an SHA256 hash.

Who has access on Windows Hello biometrics data?

Since Windows Hello biometrics data is stored in encrypted format, no user, or any process other than Windows Hello has access to it.

When is Windows Hello biometrics database file created? How is a user enrolled into Windows Hello face or fingerprint authentication?

Windows Hello biometrics template database file is created on the device only when a user is enrolled into Windows Hello biometrics-based authentication. An IT administrator may configure policy settings, but it's always a user's choice if they want to use biometrics or PIN. Users can check their current enrollment into Windows Hello biometrics by going to sign-in options on their device. Go to **Start** > **Settings** > **Accounts** > **Sign-in** options. If you don't see Windows Hello in Sign-in options, then it may not be available for your device or blocked by

admin via policy. Admins can request users to enroll into Windows Hello during Autopilot or during the initial setup of the device. Admins can disallow users to enroll into biometrics via Windows Hello for Business policy configurations. However, when allowed via policy configurations, enrollment into Windows Hello biometrics is always optional for users.

When is Windows Hello biometrics database file deleted? How can a user be unenrolled from Windows Hello face or fingerprint authentication?

To remove Windows Hello and any associated biometric identification data from the device, open **Start** > **Settings** > **Accounts** > **Sign-in options**. Select the Windows Hello biometrics authentication method you want to remove, and then select **Remove**. The action unenrolls from Windows Hello biometrics authentication and deletes the associated biometrics template database file. For more details, see Windows sign-in options and account protection (microsoft.com) .

Management and operations

Can I deploy and manage Windows Hello for Business by using Microsoft Configuration Manager?

Starting in Configuration Manager, version 2203, Windows Hello for Business deployments using Configuration Manager are no longer supported.

How do I delete a Windows Hello for Business container on a device?

You can delete the Windows Hello for Business container by executing the command `certutil.exe -deleteHelloContainer`.

What happens when a user forgets their PIN?

If the user can sign in with a password, they can reset their PIN by selecting the *I forgot my PIN* link in the Settings app or from the lock screen, by selecting the *I forgot my PIN* link on the PIN credential provider.

For on-premises deployments, devices must be connected to their on-premises network (domain controllers and/or certificate authority) to reset their PINs. Hybrid deployments can onboard their Microsoft Entra tenant to use the *Windows Hello for Business PIN reset service* to reset their PINs. Non-destructive PIN reset works without access to the corporate network. Destructive PIN reset requires access to the corporate network. For more details about destructive and non-destructive PIN reset, see PIN reset.

Does Windows Hello for Business prevent the use of simple PINs?

Yes. Our simple PIN algorithm looks for and disallows any PIN that has a constant delta from one digit to the next. The algorithm counts the number of steps required to reach the next digit, overflowing at 10 ('zero'). So, for example:

- The PIN 1111 has a constant delta of (0,0,0), so it isn't allowed
- The PIN 1234 has a constant delta of (1,1,1), so it isn't allowed
- The PIN 1357 has a constant delta of (2,2,2), so it isn't allowed
- The PIN 9630 has a constant delta of (7,7,7), so it isn't allowed
- The PIN 1593 has a constant delta of (4,4,4), so it isn't allowed
- The PIN 7036 has a constant delta of (3,3,3), so it isn't allowed
- The PIN 1231 doesn't have a constant delta (1,1,2), so it's allowed
- The PIN 1872 doesn't have a constant delta (7,9,5), so it's allowed

This check prevents repeating numbers, sequential numbers, and simple patterns. It always results in a list of 100 disallowed PINs (independent of the PIN length). This algorithm doesn't apply to alphanumeric PINs.

Which diagnostic data is collected when Windows Hello for Business is enabled?

To help Microsoft keep things working properly, to help detecting and preventing fraud, and to continue improving Windows Hello, diagnostic data about how people use Windows Hello is collected. For example:

- Data about whether people sign in with their face, iris, fingerprint, or PIN
- The number of times they use it
- Whether it works or not All this is valuable information that helps Microsoft building a better product. The data is pseudonymized, does not include biometric information, and

is encrypted before it is transmitted to Microsoft. You can choose to stop sending diagnostic data to Microsoft at any time. Learn more about diagnostic data in Windows.

Can I disable the PIN while using Windows Hello for Business?

No. The movement away from passwords is accomplished by gradually reducing the use of the password. In situations where you can't authenticate by using biometrics, you need a fallback mechanism that isn't a password. The PIN is the fallback mechanism. Disabling or hiding the PIN credential provider will disable the use of biometrics.

What happens when an unauthorized user gains possession of a device enrolled in Windows Hello for Business?

The unauthorized user won't be able to utilize any biometric options and will have the only option to enter a PIN.

If the user attempts to unlock the device by entering random PINs, after three unsuccessful attempts the credential provider will display the following message: **You've entered an incorrect PIN several times. To try again, enter A1B2C3 below**. Upon entering the challenge phrase *A1B2C3*, the user will be granted one more opportunity to enter the PIN. If unsuccessful, the provider will be disabled, leaving the user with the only option to reboot the device. Following the reboot, the aforementioned pattern repeats.

If unsuccessful attempts continue, the device will enter a lockout state, lasting for 1 minute after the first reboot, 2 minutes after the fourth reboot, and 10 minutes after the fifth reboot. The duration of each lockout increases accordingly. This behavior is a result of the TPM 2.0 anti-hammering feature. For more information about the TPM anti-hammering feature, see TPM 2.0 anti-hammering.

What happens if biometric authentication (facial recognition or fingerprint recognition) fails multiple times with Windows Hello for Business?

Windows includes anti-hammering protections for biometric sign in: after multiple unsuccessful attempts, a specific biometric method (for example, face or fingerprint) may become temporarily unavailable.

When a biometric method is unavailable, the user must sign in using another authentication method, such as a PIN. Once sign in using an alternative authentication method is completed, the biometric method becomes available again. Using an alternative authentication method to sign in is the only way to restore biometric availability. Restarting the device does not restore biometric availability.

This temporary unavailability of biometric authentication is implemented as a security measure to prevent repeated unauthorized authentication attempts. The number of failed attempts allowed before biometric authentication becomes unavailable cannot be changed by the user or organization at this time.

Design and planning

Can Windows Hello for Business work in air-gapped environments?

Yes. You can use the on-premises Windows Hello for Business deployment and combine it with a non-Microsoft MFA provider that doesn't require internet connectivity to achieve an air-gapped Windows Hello for Business deployment.

How many users can enroll for Windows Hello for Business on a single Windows device?

The maximum number of supported enrollments on a single device is 10. This lets 10 users each enroll their face and up to 10 fingerprints. For devices with more than 10 users, or for users that sign-in to many devices (for example, a support technician), it's recommended the use of FIDO2 security keys.

How many additional X.509 certificates can be imported or enrolled into an existing Windows Hello for Business enrollment?

The maximum number of certificates in a Windows Hello for Business container is 30. This limit includes the initial key and certificate created when the user enrolled for Windows Hello for Business.

I have extended Active Directory to Microsoft Entra ID. Can I use the on-premises deployment model?

No. If your organization is using Microsoft cloud services, then you must use a hybrid deployment model. On-premises deployments are exclusive to organizations who need more time before moving to the cloud and exclusively use Active Directory.

What attributes are synchronized by Microsoft Entra Connect with Windows Hello for Business?

Review Microsoft Entra Connect Sync: Attributes synchronized to Microsoft Entra ID for a list of attributes that sync based on scenarios. The base scenarios that include Windows Hello for Business are the Windows 10 scenario and the Device writeback scenario. Your environment may include other attributes.

Can I use non-Microsoft MFA providers with Windows Hello for Business?

Yes, if you're using federated hybrid deployment, you can use any non-Microsoft that provides an AD FS MFA adapter. A list of non-Microsoft MFA adapters can be found here.

Does Windows Hello for Business work with non-Microsoft federation servers?

Windows Hello for Business works with any non-Microsoft federation servers that support the protocols used during the provisioning experience.

Expand table

Protocol	Description
[MS-KPP]: Key Provisioning Protocol	Specifies the Key Provisioning Protocol, which defines a mechanism for a client to register a set of cryptographic keys on a user and device pair.
[MS-OAPX]: OAuth 2.0 Protocol Extensions	Specifies the OAuth 2.0 Protocol Extensions, which are used to extend the OAuth 2.0 Authorization Framework. These extensions enable authorization features such as resource specification, request identifiers, and log in hints.
[MS-OAPXBC]: OAuth 2.0 Protocol Extensions for Broker Clients	Specifies the OAuth 2.0 Protocol Extensions for Broker Clients, extensions to RFC6749 (the OAuth 2.0 Authorization Framework) that allow a broker client to obtain access tokens on behalf of calling clients.
[MS-OIDCE]: OpenID Connect 1.0 Protocol Extensions	Specifies the OpenID Connect 1.0 Protocol Extensions. These extensions define other claims to carry information about the user, including the user principal name, a locally unique identifier, a time for password expiration, and a URL for password change. These extensions also define more provider meta-data that enables the discovery of the issuer of access tokens and gives additional information about provider capabilities.

Can I enroll local Windows accounts in Windows Hello for Business?

Windows Hello for Business is not designed to work with local accounts.

What are the biometric requirements for Windows Hello for Business?

Read Windows Hello biometric requirements for more information.

Can I wear a mask to enroll or unlock using Windows Hello face authentication?

Wearing a mask to enroll is a security concern because other users wearing a similar mask may be able to unlock your device. Remove a mask if you're wearing one when you enroll or unlock with Windows Hello face authentication. If your working environment doesn't allow you to

remove a mask temporarily, consider un-enrolling from face authentication and only using PIN or fingerprint.

How does Windows Hello for Business work with Microsoft Entra registered devices?

A user will be prompted to set up a Windows Hello for Business key on a Microsoft Entra registered devices if the feature is enabled by policy. If the user has an existing Windows Hello container, the Windows Hello for Business key will be enrolled in that container and will be protected using existing gestures.

If a user has signed into their Microsoft Entra registered device with Windows Hello, their Windows Hello for Business key will be used to authenticate the user's work identity when they try to use Microsoft Entra resources. The Windows Hello for Business key meets Microsoft Entra multifactor authentication (MFA) requirements and reduces the number of MFA prompts users will see when accessing resources.

It's possible to Microsoft Entra register a domain joined device. If the domain joined device has a convenience PIN, sign in with the convenience PIN will no longer work. This configuration isn't supported by Windows Hello for Business.

For more information, see Microsoft Entra registered devices.

Does Windows Hello for Business work with non-Windows operating systems?

Windows Hello for Business is a feature of the Windows platform.

Does Windows Hello for Business work with Microsoft Entra Domain Services clients?

No, Microsoft Entra Domain Services is a separately managed environment in Azure, and hybrid device registration with cloud Microsoft Entra ID isn't available for it via Microsoft Entra Connect. Hence, Windows Hello for Business doesn't work with Microsoft Entra Domain Services.

Is Windows Hello for Business considered multifactor authentication?

Windows Hello for Business is two-factor authentication based on the observed authentication factors of: *something you have*, *something you know*, and *something that's part of you*. Windows Hello for Business incorporates two of these factors: something you have (the user's private key protected by the device's security module) and something you know (your PIN). With the proper hardware, you can enhance the user experience by introducing biometrics. By using biometrics, you can replace the "something you know" authentication factor with the "something that is part of you" factor, with the assurances that users can fall back to the "something you know factor".

> ⓘ **Note**
>
> The Windows Hello for Business key meets Microsoft Entra multifactor authentication (MFA) requirements and reduces the number of MFA prompts users will see when accessing resources. For more information, see **What is a Primary Refresh Token**.

Which is a better or more secure form of authentication, key or certificate?

Both types of authentication provide the same security; one is not more secure than the other. The trust models of your deployment determine how you authenticate to Active Directory. Both key trust and certificate trust use the same hardware-backed, two-factor credential. The difference between the two trust types is the issuance of end-entity certificates:

- The *key trust* model authenticates to Active Directory by using a raw key. Key trust doesn't require an enterprise-issued certificate, therefore you don't need to issue certificates to users (domain controller certificates are still needed)
- The *certificate trust* model authenticates to Active Directory by using a certificate. Therefore, you need to issue certificates to users. The certificate used in certificate trust uses the TPM-protected private key to request a certificate from your enterprise's issuing CA

What is convenience PIN?

Convenience PIN provides a simpler way to sign in to Windows than passwords, but it still uses a password for authentication. When the correct convenience PIN is provided to Windows, the password information is loaded from its cache and authenticates the user. Organizations using

convenience PINs should move to **Windows Hello for Business**. New Windows deployments should deploy Windows Hello for Business and not convenience PINs.

Can I use a convenience PIN with Microsoft Entra ID?

No. While it's possible to set a convenience PIN on Microsoft Entra joined and Microsoft Entra hybrid joined devices, convenience PIN isn't supported for authenticating Microsoft Entra user accounts (including synchronized identities). Convenience PIN is only supported for authenticating users to on-premises Active Directory and local account users. Organizations that want to authenticate to Microsoft Entra should deploy Windows Hello for Business, which provides users with an Entra credential that can be used to access Entra-protected resources. Organizations that do not use Windows Hello for Business can choose to deploy convenience PINs on their workstations, including Entra Joined or Entra Hybrid Joined workstations used by on-premises or synchronized user accounts, but convenience PIN will not be able to provide authentication or SSO to Entra. The convenience PIN may still be used for logging into the user's PC or for storing other credentials used by the organization, such as certificates or passkeys.

What about virtual smart cards?

Windows Hello for Business is the modern, two-factor authentication for Windows. Customers using virtual smart cards are strongly encouraged to move to Windows Hello for Business.

Why can't I use Windows Hello for Business when connecting remotely to a Windows device?

Windows Hello for Business is designed for local, interactive sign-in and protects credentials that are secured by the device's hardware. These credentials are only accessible on the device where the user is physically present.

While you can use Windows Hello for Business on your local device to sign in to a remote virtual machine and handle redirected authentication prompts, it is not supported to enroll into Windows Hello for Business or use Windows Hello for Business for authentication on the remote device itself

What URLs do I need to allow for a hybrid deployment?

For a list of required URLs, see Microsoft 365 Common and Office Online.

If your environment uses Microsoft Intune, see Network endpoints for Microsoft Intune.

Features

Can I use an external Windows Hello compatible camera when my computer has a built-in Windows Hello compatible camera?

Yes, you can use an external Windows Hello compatible camera if a device has an internal Windows Hello camera. When both cameras are present, the external camera is used for face authentication. For more information, see IT tools to support Windows 10, version 21H1. If ESS is enabled, see Windows Hello Enhanced Sign-in Security.

Can I use an external Windows Hello compatible camera or other Windows Hello compatible accessory when my laptop lid is closed or docked?

Some laptops and tablets with keyboards that close may not use an external Windows Hello compatible camera or other Windows Hello compatible accessory when the computer is docked with the lid closed. The issue has been addressed in Windows 11, version 22H2.

Can I use Windows Hello for Business credentials in private browser mode or "incognito" mode?

Windows Hello for Business credentials need access to device state, which is not available in private browser mode or incognito mode. Hence it can't be used in private browser or Incognito mode.

Can I use both a PIN and biometrics to unlock my device?

You can use *multifactor unlock* to require users to provide an extra factor to unlock their device. Authentication remains two-factor, but another factor is required before Windows allows the user to reach the desktop. To learn more, see Multifactor Unlock.

Cloud Kerberos trust

What is Windows Hello for Business cloud Kerberos trust?

Windows Hello for Business *cloud Kerberos trust* is a *trust model* that enables Windows Hello for Business deployment using the infrastructure introduced for supporting security key sign-in on Microsoft Entra hybrid joined devices and on-premises resource access on Microsoft Entra joined devices. Cloud Kerberos trust is the preferred deployment model if you do not need to support certificate authentication scenarios. For more information, see cloud Kerberos trust deployment.

Does Windows Hello for Business cloud Kerberos trust work in my on-premises environment?

This feature doesn't work in a pure on-premises AD domain services environment.

Does Windows Hello for Business cloud Kerberos trust work with RODC present in the hybrid environment?

Windows Hello for Business cloud Kerberos trust functions correctly when the client authenticates directly to a writable domain controller or to a Read-Only Domain Controller (RODC) that doesn't cache the user's credentials, in accordance with the Password Replication Policy. If the client attempts to authenticate to an RODC that can cache the user's credentials, cloud Kerberos trust authentication might fail. To mitigate this, deploy KDC certificates to all RODCs to support Windows Hello for Business key trust authentication, which is also required for those RODCs to support LDAP over SSL. This configuration ensures that authentication can seamlessly failover to Windows Hello for Business key trust authentication, thereby guaranteeing successful user authentication.

Do I need line of sight to a domain controller to use Windows Hello for Business cloud Kerberos trust?

Windows Hello for Business cloud Kerberos trust requires line of sight to a domain controller when:

- a user signs in for the first time or unlocks with Windows Hello for Business after provisioning
- attempting to access on-premises resources secured by Active Directory
- Hybrid Users who change the password locally on the device (via Alt+Ctrl+Del). They have to unlock with the PIN successfully with LOS to DC at least one time to cache the PIN login again.

Can I use RDP/VDI with Windows Hello for Business cloud Kerberos trust?

Windows Hello for Business cloud Kerberos trust can't be used as a supplied credential with RDP/VDI. Similar to key trust, cloud Kerberos trust can be used for RDP if a certificate is enrolled into Windows Hello for Business for this purpose. As an alternative, consider using Remote Credential Guard which doesn't require to deploy certificates.

Do all my domain controllers need to be fully patched as per the prerequisites for me to use Windows Hello for Business cloud Kerberos trust?

No, only the number necessary to handle the load from all cloud Kerberos trust devices.

Key trust

Why does authentication fail immediately after provisioning hybrid key trust?

In a hybrid deployment, a user's public key must sync from Microsoft Entra ID to Active Directory before it can be used to authenticate against a domain controller. This sync is handled by Microsoft Entra Connect and will occur during a normal sync cycle.

Can I use Windows Hello for Business key trust and RDP?

Remote Desktop Protocol (RDP) doesn't support using key-based authentication as supplied credentials. However, you can deploy certificates in the key trust model to enable RDP. For

more information, see Deploying certificates to key trust users to enable RDP. As an alternative, consider using Remote Credential Guard which doesn't require to deploy certificates.

How to enable passkeys (FIDO2) in Microsoft Entra ID

For enterprises that use passwords today, passkeys (FIDO2) provide a seamless way for workers to authenticate without entering a username or password. Passkeys (FIDO2) provide improved productivity for workers, and have better security.

This article lists requirements and steps to enable passkeys in your organization. After you complete these steps, users in your organization can then register and sign in to their Microsoft Entra account using a passkey stored on a FIDO2 security key, a native or third-party passkey provider, or in Microsoft Authenticator.

For more information about enabling passkeys in Microsoft Authenticator, see How to enable passkeys in Microsoft Authenticator.

For more information about passkey authentication, see Support for FIDO2 authentication with Microsoft Entra ID.

> **Note**
>
> Microsoft Entra ID currently supports:
>
> - synced passkeys
> - device-bound passkeys stored on FIDO2 security keys and in Microsoft Authenticator
>
> Passkeys (FIDO2) are available in all Microsoft Entra ID editions, including Microsoft Entra ID Free. No extra licenses are required. For more information, see Passkeys (FIDO2) authentication method in Microsoft Entra ID.

Get started

To enable passkeys for your organization, complete these steps in order:

1. **Enable passkey profiles** — Opt in to passkey profiles in the Microsoft Entra admin center. Your existing global settings transfer to a Default passkey profile.
2. **Create a passkey profile** — Define attestation, passkey type (device-bound or synced), and key restriction settings for your target groups.

3. **Apply a passkey profile to a targeted group** — Assign your passkey profile to specific user groups or all users.
4. **Enable synced passkeys** (optional) — If you want to allow synced passkeys, select **Synced** as a passkey type in your profile.
5. **Enforce passkey sign-in** (optional) — Create a Conditional Access authentication strength policy to require passkeys for sensitive resources.

Passkey profiles

What are passkey profiles?

Passkey profiles enable granular group-based configurations for passkey (FIDO2) authentication. Instead of a single tenant-wide setting, you can define specific requirements such as attestation, passkey type (device-bound or synced), or Authenticator Attestation GUID (AAGUID) restrictions. You can apply requirements in separate passkey profiles for different user groups, such as admins versus frontline staff.

> **Note**
>
> An Authentication Policy Administrator needs to configure a passkey profile to enable synced passkeys. For more information, see Enable synced passkeys.

A passkey profile is a named set of policy rules that governs how users in targeted groups can register and authenticate with passkeys (FIDO2). Profiles support advanced controls such as:

Expand table

Option	Configuration
Enforce attestation	Enabled, Disabled
Passkey types	Device-bound, Synced
Target specific authenticators	Allow or block specific authenticators by their AAGUID. For more information, see Authenticator Attestation GUID.

Examples of use cases for passkey profiles

> **Note**

If a passkey profile for both device-bound and synced passkeys targets Microsoft Authenticator, users need to run Microsoft Authenticator iOS version 6.8.37 or Android version 6.2507.4749.

Special consideration for high-privileged accounts

Expand table

Passkey profile	Target groups	Passkey types	Attestation enforcement	Key restrictions
All device-bound passkeys (attestation enforced)	IT admins Executives Engineering	Device-bound	Enabled	Disabled
All synced or device-bound passkeys	HR Sales	Device-bound, Synced	Disabled	Disabled

Targeted rollout of passkeys in Microsoft Authenticator

Expand table

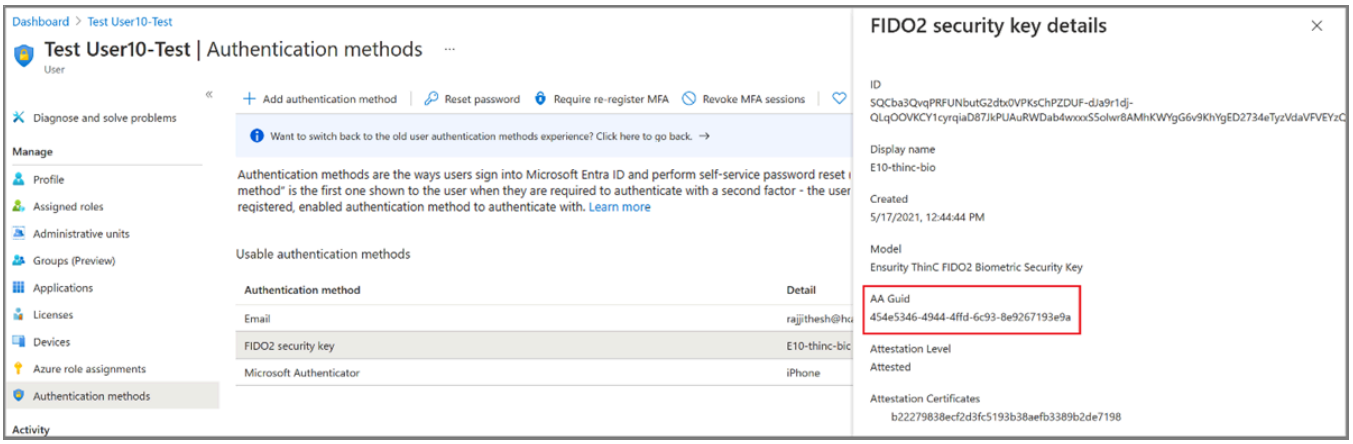
Passkey profile	Target groups	Passkey types	Attestation enforcement	Key restrictions
All device-bound passkeys (excluding Microsoft Authenticator)	All users	Device-bound	Enabled	Enabled - Behavior: Block - AAGUIDs: Microsoft Authenticator for iOS, Microsoft Authenticator for Android
Passkeys in Microsoft Authenticator	Pilot group 1 Pilot group 2	Device-bound	Enabled	Enabled - Behavior: Allow - AAGUIDs: Microsoft Authenticator for iOS, Microsoft Authenticator for Android

Passkey (FIDO2) Authenticator Attestation GUID (AAGUID)

The FIDO2 specification requires each passkey vendor to provide an Authenticator Attestation GUID (AAGUID) during registration. An AAGUID is a 128-bit identifier indicating the key type, such as the make and model. Passkey (FIDO2) providers on desktop and mobile devices are also expected to provide an AAGUID during registration.

> **Note**
>
> The vendor must ensure that the AAGUID is identical across all substantially identical security keys or passkey (FIDO2) providers made by that vendor, and different (with high probability) from the AAGUIDs of all other types of security keys or passkey (FIDO2) providers. To ensure this, the AAGUID for a given security key model or passkey (FIDO2) provider should be randomly generated. For more information, see Web Authentication: An API for accessing Public Key Credentials - Level 2 (w3.org).

You can work with your passkey vendor to determine the AAGUID of the passkey (FIDO2), or see FIDO2 security keys eligible for attestation with Microsoft Entra ID. If the passkey (FIDO2) is already registered, you can find the AAGUID by viewing the authentication method details of the passkey (FIDO2) for the user.



Passkey profile prerequisites

- Devices must support passkey (FIDO2) authentication. For Windows devices that are joined to Microsoft Entra ID, the best experience is on Windows 10 version 1903 or higher. Hybrid-joined devices must run Windows 10 version 2004 or higher.
- If a passkey profile for both device-bound and synced passkeys targets Microsoft Authenticator, users need to run Microsoft Authenticator iOS version 6.8.37 or Android version 6.2507.4749.
- Policy size limit:
  - The Authentication methods policy supports a size limit of 20 KB. You can't save more passkey profiles after the size limit is reached. To check the size, use the Get authenticationMethodsPolicy Microsoft Graph API to retrieve the JSON for the Authentication methods policy. Save the output as a .txt file, then right-click and select **Properties** to view the file size.
  - Reference sizes:

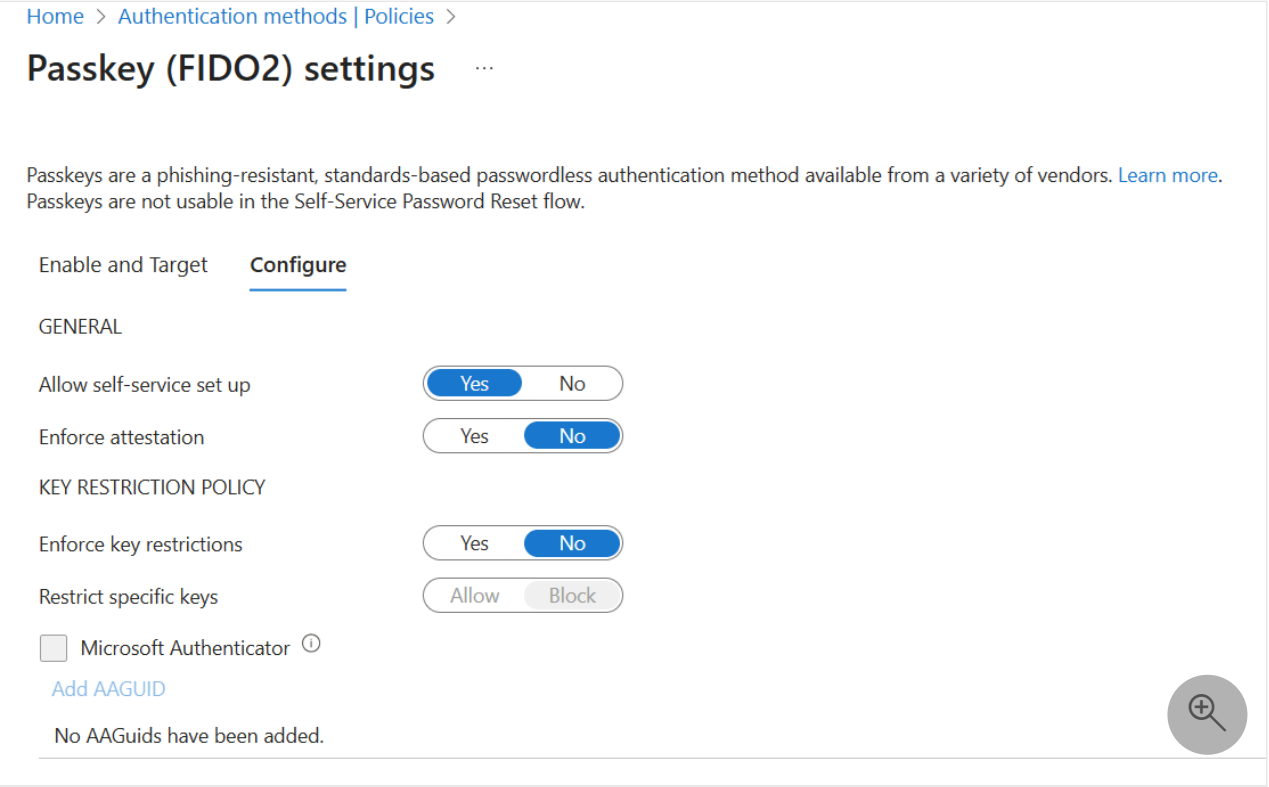
    - Base passkey policy without changes: 1.44 KB
    - Target with 1 applied passkey profile: 0.23 KB
    - Target with 5 applied passkey profiles: 0.4 KB
    - Passkey profile with no AAGUIDs: 0.4 KB
    - Passkey profile with 10 AAGUIDs: 0.3 KB
- Users must complete multifactor authentication (MFA) within the past five minutes before they can register a passkey (FIDO2).
- Users need an authenticator that supports Microsoft Entra ID's attestation requirements. For more information, see Microsoft Entra ID attestation for FIDO2 security key vendors.

Enable passkey profiles

> ⓘ **Note**
>
> When you enable passkey profiles, your global passkey (FIDO2) policy settings automatically transfer to a **Default passkey profile**. Up to three passkey profiles, including the **Default passkey profile**, are supported. Support for more passkey profiles is in development.

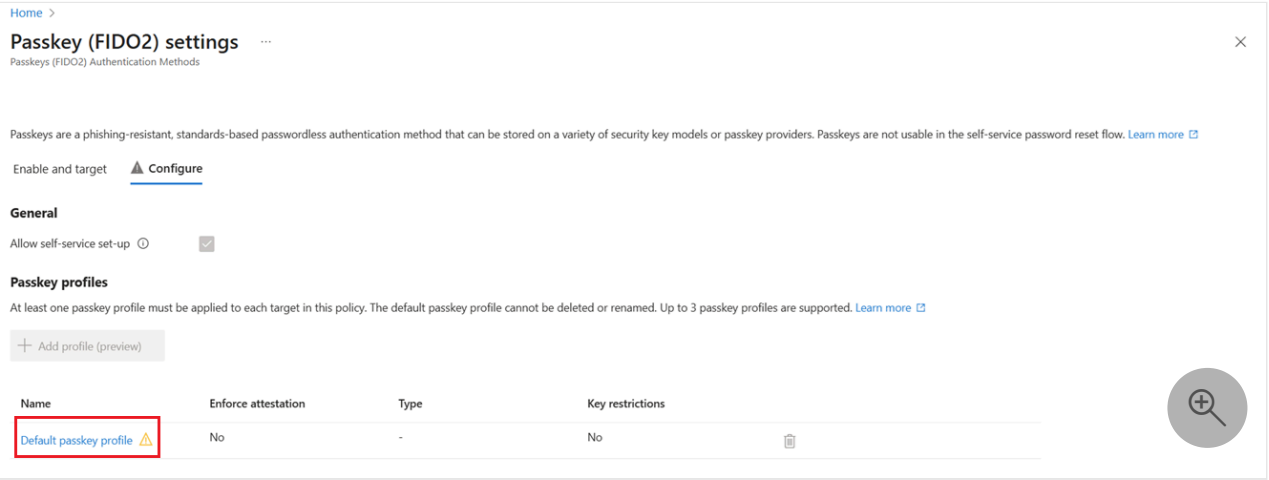
1. Sign in to the Microsoft Entra admin center as at least an Authentication Policy Administrator.
2. Browse to **Entra ID** > **Security** > **Authentication methods** > **Policies**.
3. Select **Passkey (FIDO2)**. Select the link in the banner text to opt-in to using passkeys profiles.

   > ⓘ **Note**
   >
   > After you opt in to enable passkey profiles, you can't opt out.



4. On the **Configure** tab, set **Allow self-service set up** to **Yes**. If set to **No**, users can't register a passkey by using Security info, even if passkeys (FIDO2) are enabled by the Authentication methods policy. This setting is a global policy; it's not on the profile level.

5. Select the **Default passkey profile**.



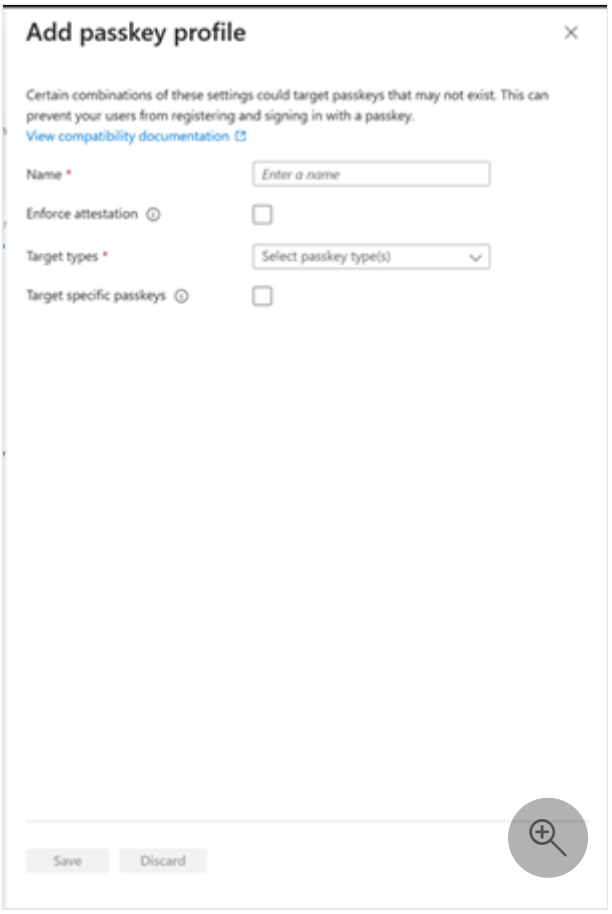
For **Passkey types**, select the types of passkeys that you want to allow.

6. Select **Save**.

Create a new passkey profile

1. On the **Configure** tab, select **+ Add passkey profile**.

2. Fill in the profile details.



> ⚠ **Warning**
>
> - If you set **Enforce attestation** to **Yes**, attestation is required at registration time. Microsoft Entra ID can verify the authenticator's make and model against trusted metadata. Attestation assures your organization that the passkey is genuine and comes from the stated vendor. If enforce attestation is set to **No**, Microsoft Entra ID can't guarantee any attribute about a passkey, including if it's synced or device-bound.
>
> - Attestation enforcement governs whether a passkey (FIDO2) is allowed only during registration. Users who register a passkey (FIDO2) without attestation aren't blocked from sign-in if **Enforce attestation** is set to **Yes** later.

For other vendor attestation requirements, see Microsoft Entra ID attestation for FIDO2 security key vendors.

Key Restriction Policy

- **Enforce key restrictions** should be set to **Yes** only if your organization wants to only allow or disallow certain security key models or passkey providers, which are identified by their AAGUID. You can work with your security key vendor to determine

the AAGUID of the passkey. If the passkey is already registered, you can find the AAGUID by viewing the authentication method details of the passkey for the user.

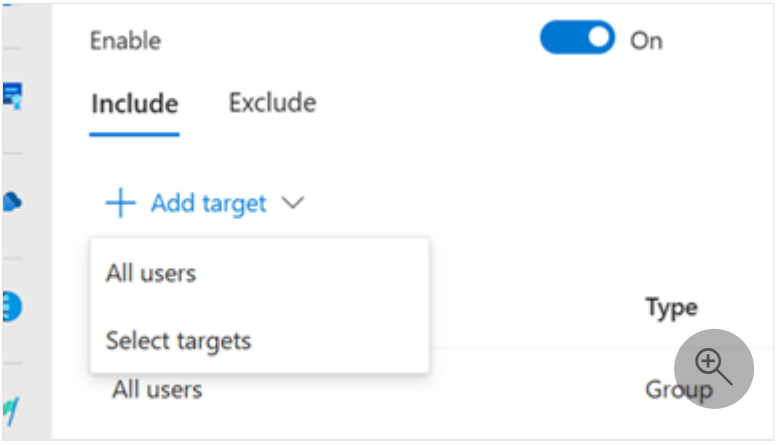
> ⚠ **Warning**
>
> - Key restrictions set the usability of specific models or providers for both registration and authentication. If you change key restrictions and remove an AAGUID that you previously allowed, users who previously registered an allowed method can no longer use it for sign-in.
> - Use AAGUID lists as a policy guide rather than a strict security control when **Enforce attestation** is set to **No**.

3. Select **Save**.

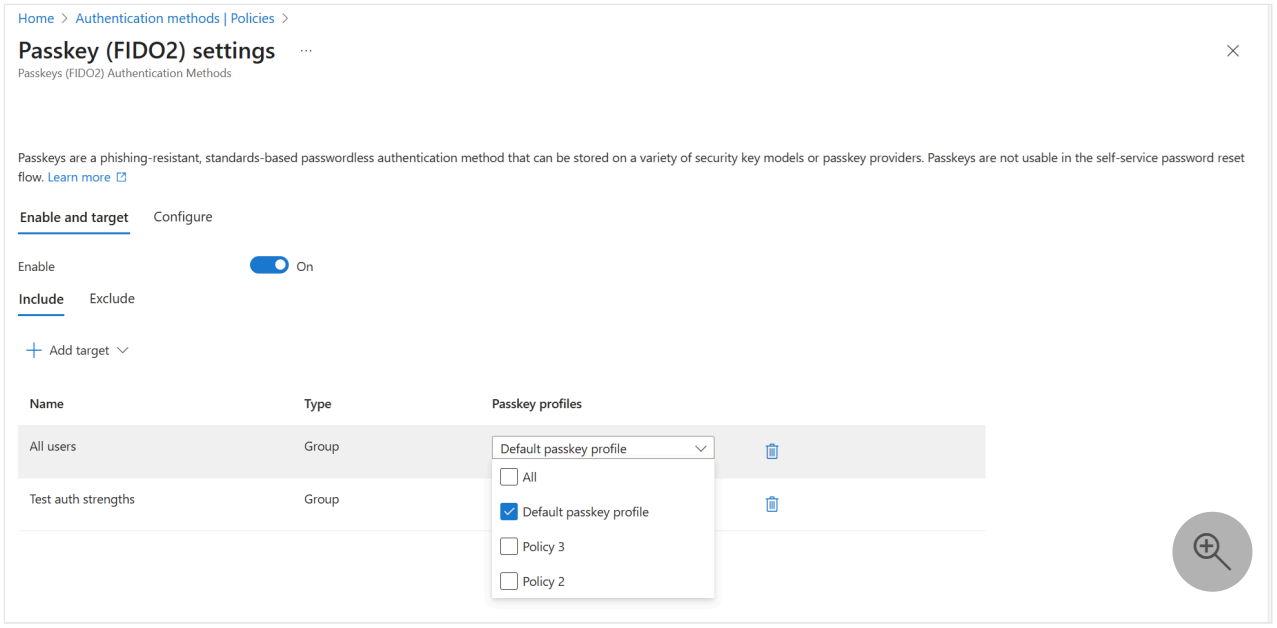
Apply a passkey profile to a targeted group

1. Select **Enable and Target**.

2. Select **Add target**, and then choose **All users** or **Select targets** to choose specific groups.



3. Select the passkey profiles that you want to assign to a specific target.

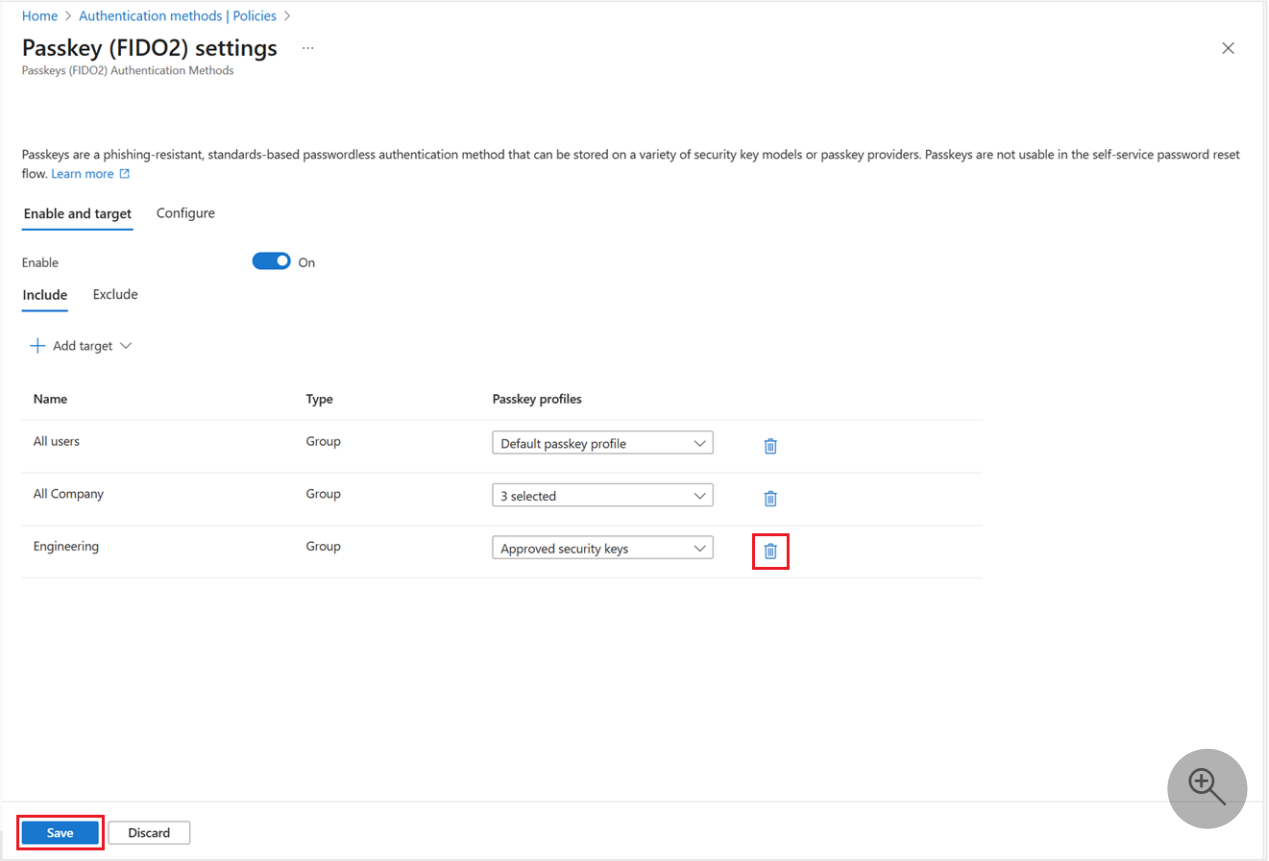


> **Note**
>
> A target group (for example, Engineering) can be scoped for multiple passkey profiles. When a user is scoped for multiple passkey profiles, registration and authentication with a passkey are allowed if the passkey fully satisfies the requirements of at least one of the scoped passkey profiles. There's no particular order to the check. If a user is a member of an excluded group in the **Passkeys (FIDO2)** authentication method policy, they're blocked from FIDO2 passkey registration or sign-in entirely, and this takes precedence over them being in any **Included** groups.

Delete a passkey profile

1. Select **Configure**.
2. Select the delete icon next to the passkey profile that you want to delete, and then select **Save**.

> **Note**
>
> You can delete a profile only if it's not assigned to a group of users in **Enable and target**. If the delete icon is unavailable, first remove any targets that are assigned that profile.



Synced passkeys (FIDO2)

What are synced versus device-bound passkeys?

Passkeys are FIDO2-based credentials that provide strong, phishing-resistant authentication. Microsoft Entra ID supports two main types of passkeys:

- Device-bound passkeys: The private key is created and stored on a single physical device and never leaves it. Examples:
  - Microsoft Authenticator (iOS)
  - Microsoft Authenticator (Android)
  - Security key
- Synced passkeys: The private key is created by the hardware security module (HSM) and encrypted on the local device. This encrypted key is then synced and stored in the cloud passkey provider. Other devices authenticated with the passkey provider may then use the passkey. This may differ depending on the provider. Synced passkeys do not support attestation. Examples:
  - Apple iCloud Keychain
  - Google Password Manager

> **Note**

Treat synced passkeys as phishing-resistant credentials but with the same security posture as other unattested authenticators.

Synced passkey requirements

- To enable synced passkeys, your organization must have passkey profiles enabled.
- An account with at least Authentication Policy Administrator permissions.
- The following table outlines the minimum device requirements for using synced passkeys. The columns represent the device platform where the user signs in.

Expand table

Passkey provider	Windows	macOS	iOS	Android
Apple Passwords (also called iCloud Keychain)	N/A	Natively built in. macOS 13+	Natively built in. iOS 16+	N/A
Google Password Manager	Built into Chrome	Built into Chrome	Built into Chrome. iOS 17+	Natively built in (excluding Samsung devices). Android 9+
Other passkey providers (such as 1Password, Bitwarden)	Check for a browser extension	Check for a browser extension	Check for an app. iOS 17+	Check for an app. Android 14+

Enable synced passkeys

1. Sign in to the Microsoft Entra admin center as at least an Authentication Policy Administrator.
2. Make sure passkey profiles are enabled.
3. Browse to **Entra ID** > **Security** > **Authentication methods** > **Policies**.
4. Select **Passkey (FIDO2)** > **Configure**.
5. Add a profile or edit an existing profile.
6. Under **Passkey type**, select **Synced**, and then save the profile.

> **Note**
>
> If you disable synced passkeys for a given passkey profile, targeted users can't sign in with a synced passkey even if they already registered one.

Delete a passkey (FIDO2)

To remove a passkey (FIDO2) associated with a user account, delete it from the user's authentication method.

1. Sign in to the Microsoft Entra admin center and search for the user whose passkey (FIDO2) needs to be removed.
2. Select **Authentication methods** > right-click **Passkey (device-bound)** and select **Delete**.

Enforce passkey (FIDO2) sign-in

To make users sign in with a passkey (FIDO2) when they access a sensitive resource, you can:

- Use a built-in phishing-resistant authentication strength

  Or

- Create a custom authentication strength

The following steps show how to create a custom authentication strength. It's a Conditional Access policy that allows passkey (FIDO2) sign-in for only a specific security key model or passkey (FIDO2) provider. For a list of FIDO2 providers, see FIDO2 security keys eligible for attestation with Microsoft Entra ID.

1. Sign in to the Microsoft Entra admin center as at least a Conditional Access Administrator.
2. Browse to **Entra ID** > **Authentication methods** > **Authentication strengths**.
3. Select **New authentication strength**.
4. Provide a **Name** for your new authentication strength.
5. Optionally provide a **Description**.
6. Select **Passkeys (FIDO2)**.
7. Optionally, if you want to restrict a specific AAGUID, select **Advanced options** > **Add AAGUID**. Enter the AAGUID, and select **Save**.
8. Choose **Next** and review the policy configuration.

Provision FIDO2 security keys using Microsoft Graph API (preview)

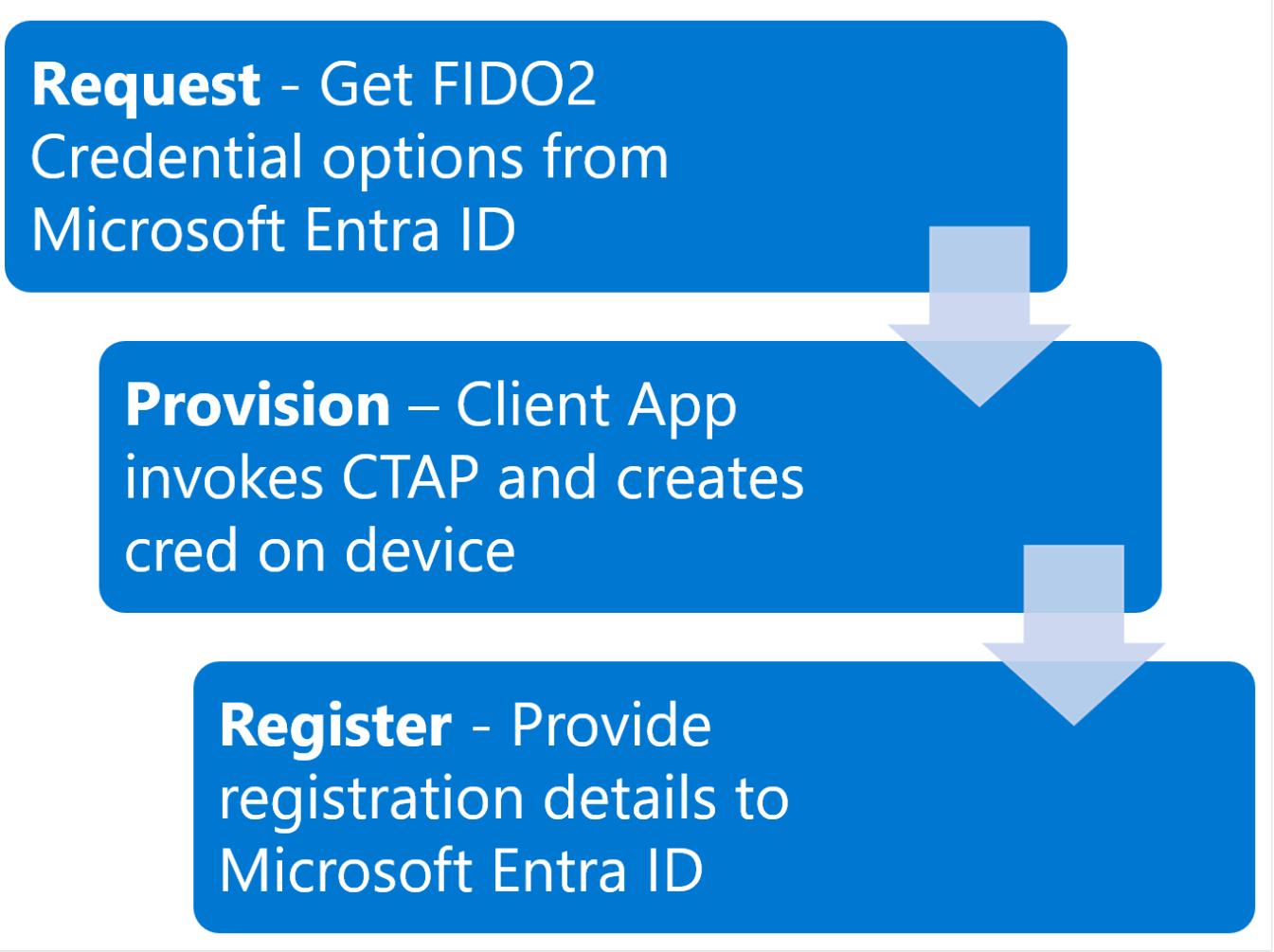
Currently in preview, administrators can use Microsoft Graph and custom clients to provision FIDO2 security keys on behalf of users. Provisioning requires the Authentication

Administrator role or a client application with UserAuthenticationMethod.ReadWrite.All permission. The provisioning improvements include:

- The ability to request WebAuthn **creation Options** from Microsoft Entra ID
- The ability to register the provisioned security key directly with Microsoft Entra ID

With these new APIs, organizations can build their own clients to provision passkey (FIDO2) credentials on security keys on behalf of a user. To simplify this process, three main steps are required.

1. **Request** creationOptions for a user: Microsoft Entra ID returns the necessary data for your client to provision a passkey (FIDO2) credential. This includes information such as user information, relying party ID, credential policy requirements, algorithms, registration challenge and more.
2. **Provision** the passkey (FIDO2) credential with the creation Options: Use the `creationOptions` and a client that supports the Client to Authenticator Protocol (CTAP) to provision the credential. During this step, you need to insert the security key and set a PIN.
3. **Register** the provisioned credential with Microsoft Entra ID: Use the formatted output from the provisioning process to provide Microsoft Entra ID the necessary data to register the passkey (FIDO2) credential for the targeted user.



Known issues

Security key provisioning

Administrator provisioning of security keys is in preview. See Microsoft Graph and custom clients to provision FIDO2 security keys on behalf of users.

Guest users

Registration of passkey (FIDO2) credentials isn't supported for internal or external guest users, including B2B collaboration users in the resource tenant.

UPN changes

If a user's UPN changes, you can no longer modify passkeys (FIDO2) to account for the change. If the user has a passkey (FIDO2), they need to sign in to Security info, delete the old passkey (FIDO2), and add a new one.

Related content

- Passkeys (FIDO2) authentication method in Microsoft Entra ID
- Support for FIDO2 authentication with Microsoft Entra ID
- How to enable passkeys in Microsoft Authenticator

Last updated on 03/20/2026

Windows passwordless experience

Article • 12/02/2024 • Applies to: ✅ Windows 11

Starting in Windows 11, version 22H2 with KB5030310, *Windows passwordless experience* is a security policy that promotes a user experience without passwords on Microsoft Entra joined devices.
When the policy is enabled, certain Windows authentication scenarios don't offer users the option to use a password, helping organizations and preparing users to gradually move away from passwords.

With Windows passwordless experience, users who sign in with Windows Hello or a FIDO2 security key:

- Can't use the password credential provider on the Windows lock screen
- Aren't prompted to use a password during in-session authentications (for example, UAC elevation, password manager in the browser, etc.)
- Don't have the option *Accounts* > *Change password* in the Settings app

> **Note**
>
> Users can reset their password using CTRL + ALT + DEL > **Manage your account**

Windows passwordless experience doesn't affect the initial sign-in experience and local accounts. It only applies to subsequent sign-ins for Microsoft Entra accounts. It also doesn't prevent a user from signing in with a password when using the *Other user* option in the lock screen.
The password credential provider is hidden only for the last signed in user who signed in Windows Hello or a FIDO2 security key. Windows passwordless experience isn't about preventing users from using passwords, rather to guide and educate them to not use passwords.

This article explains how to enable Windows passwordless experience and describes the user experiences.

> **Tip**
>
> Windows Hello for Business users can achieve passwordless sign-in from the first sign-in using the Web sign-in feature. For more information about Web sign-in, see

Web sign-in for Windows devices.

System requirements

Windows passwordless experience has the following requirements:

- Windows 11, version 22H2 with KB5030310 or later
- Microsoft Entra joined
- Windows Hello for Business credentials enrolled for the user, or a FIDO2 security key
- MDM-managed: Microsoft Intune or other MDM solution

> **Note**
>
> Microsoft Entra hybrid joined devices and Active Directory domain joined devices are currently out of scope.

Windows edition and licensing requirements

The following table lists the Windows editions that support Windows passwordless experience:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Windows passwordless experience license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Enable Windows passwordless experience with Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Authentication	Enable Passwordless Experience	Enabled

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the Policy CSP.

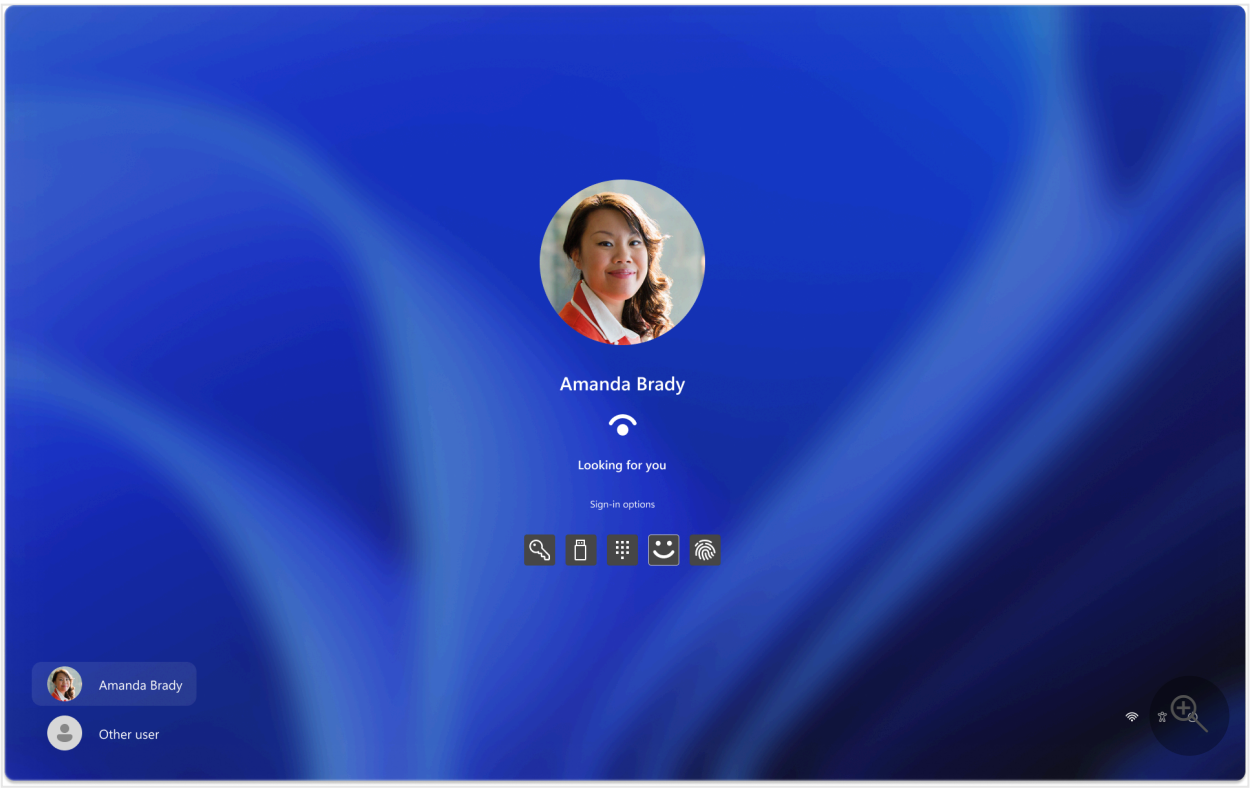
Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/Policy/Config/Authentication/EnablePasswordlessExperience` - **Data type:** int - **Value:** `1`

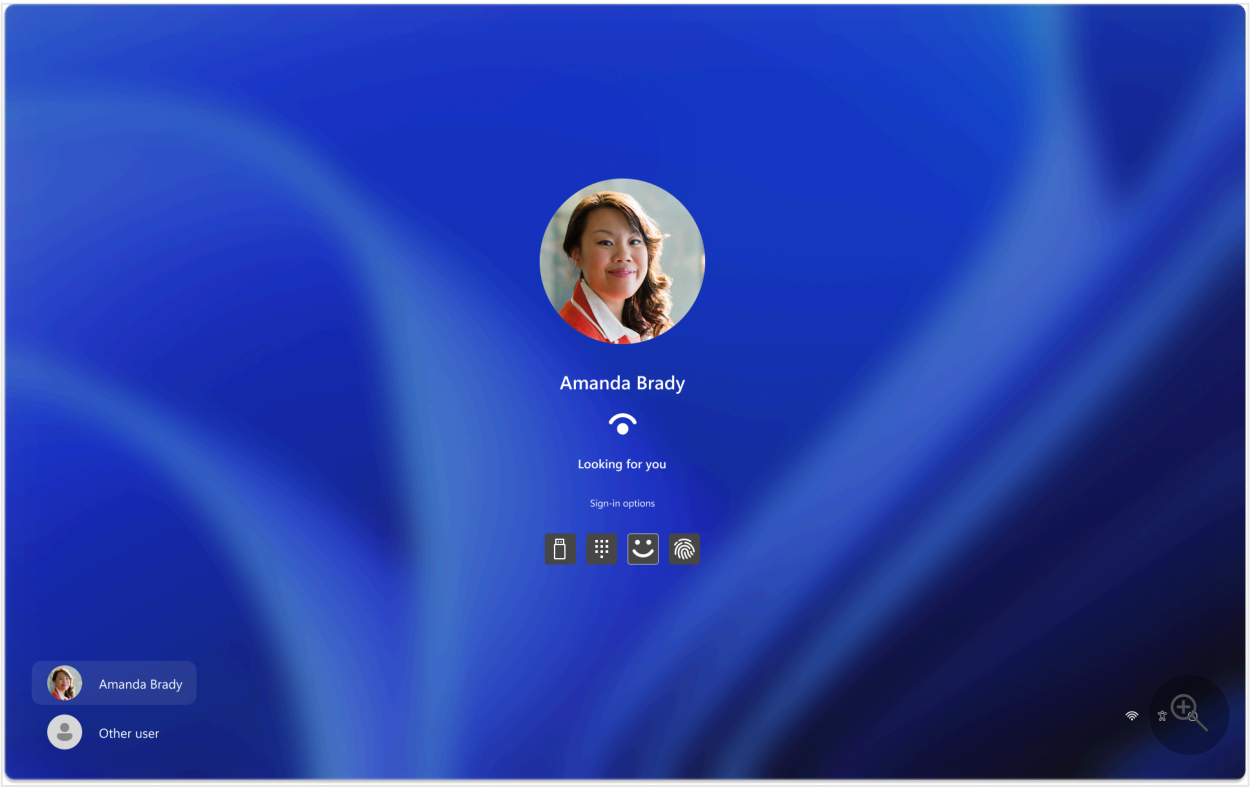
User experiences

Lock screen experience

Passwordless experience turned off: users can sign in using a password, as indicated by the presence of the password credential provider  in the Windows lock screen.



Passwordless experience turned on: the password credential provider is missing for the last user who signed in with strong credentials. A user can either sign in using a strong credential or opt to use the *Other user* option to sign in with a password.



In-session authentication experiences

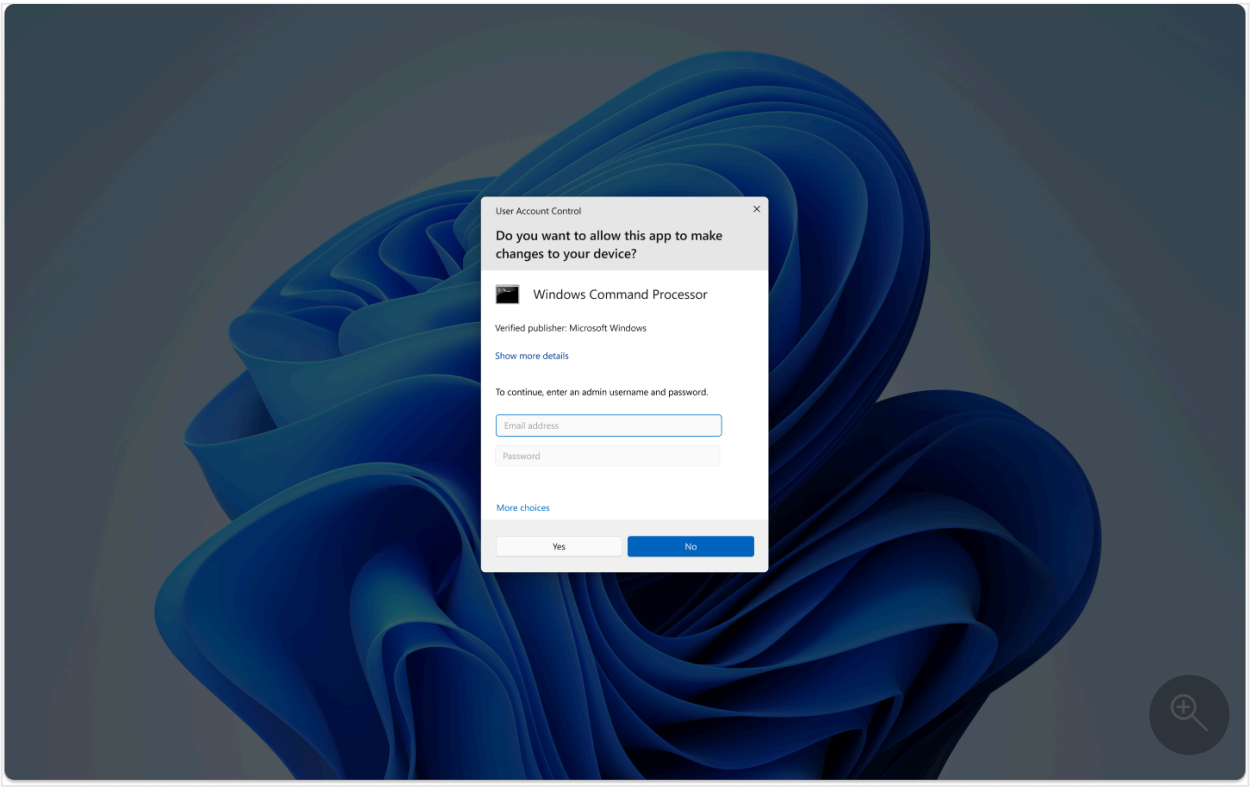
When Windows passwordless experience is enabled, users can't use the password credential provider for in-session authentication scenarios. In-session authentication scenarios include:

- Password Manager in a web browser
- Connecting to file shares or intranet sites
- User Account Control (UAC) elevation, except if a local user account is used for elevation

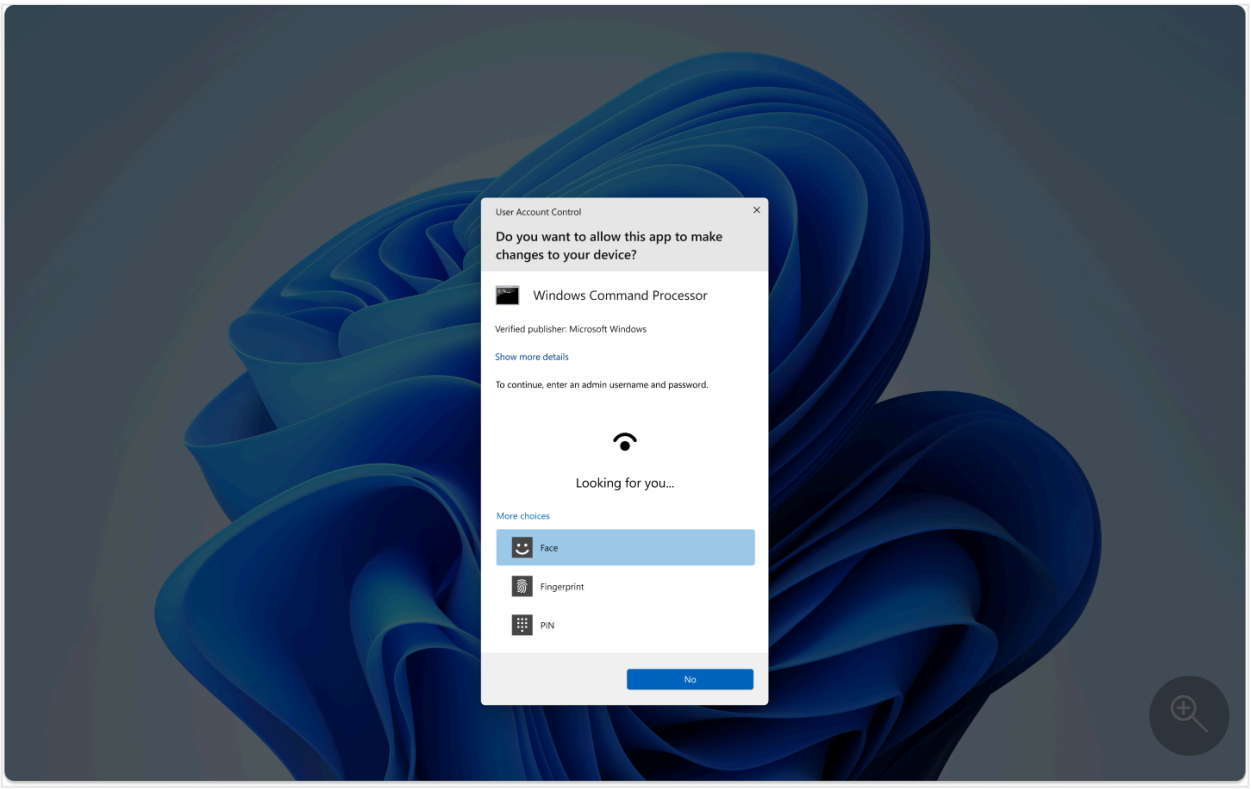
> **Note**
>
> RDP sign in defaults to the credential provider used during sign-in. However, a user can select the option *Use a different account* to sign in with a password.
>
> *Run as different user* is not impacted by Windows passwordless experience.

Example of UAC elevation experience:

Passwordless experience turned off: UAC elevation allows the user to authenticate using a password.



Passwordless experience turned on: UAC elevation doesn't allow the user to use the password credential provider for the currently logged on user. The user can authenticate using Windows Hello, a FIDO2 security key or a local user account, if available.



Recommendations

Here's a list of recommendations to consider before enabling Windows passwordless experience:

- If Windows Hello for Business is enabled, configure the PIN reset feature to allow users to reset their PIN from the lock screen. The PIN reset experience is improved starting in Windows 11, version 22H2 with KB5030310
- Don't configure the security policy *Interactive logon: Don't display last signed-in*, as it prevents Windows passwordless experience from working
- Don't disable the password credential provider using the *Exclude credential providers* policy. The key differences between the two policies are:
  - The Exclude credential providers policy disables passwords for *all accounts*, including local accounts. Windows passwordless experience only applies to Microsoft Entra accounts that sign in with Windows Hello or a FIDO2 security key. It also excludes *Other User* from the policy, so users have a backup sign in option
  - Exclude credential providers policy prevents the use of passwords for RDP and *Run as* authentication scenarios
- To facilitate helpdesk support operations, consider enabling the local administrator account or create a separate one, randomizing its password using the Windows Local Administrator Password Solution (LAPS)

Known issues

There's a known issue affecting the in-session authentication experience when using FIDO2 security keys, where security keys aren't always an available option. The product group is aware of this behavior and plans to improve this in the future.

Provide feedback

To provide feedback for Windows passwordless experience, open **Feedback Hub** and use the category **Security and Privacy** > **Passwordless experience**.

Feedback

Was this page helpful? Yes No

Provide product feedback

Support for passkeys in Windows

Applies to: ✅ Windows 11, ✅ Windows 10

Passkeys provide a more secure and convenient method to logging into websites and applications compared to passwords. Unlike passwords, which users must remember and type, passkeys are stored as secrets on a device and can use a device's unlock mechanism (such as biometrics or a PIN). Passkeys can be used without the need for other sign-in challenges, making the authentication process faster, secure, and more convenient.

You can use passkeys with any applications or websites that support them, to create and sign in with Windows Hello. Once a passkey is created and stored with Windows Hello, you can use your device's biometrics or PIN to sign in. Alternatively, you can use a companion device (phone or tablet) to sign in.

> **Note**
>
> Starting in Windows 11, version 22H2 with KB5030310, Windows provides a native experience for passkey management. However, passkeys can be used in all supported versions of Windows clients.

This article describes how to create and use passkeys on Windows devices.

How passkeys work

Microsoft has long been a founding member of the FIDO Alliance and has helped to define and use passkeys natively within a platform authenticator like Windows Hello. Passkeys utilize the FIDO industry security standard, which is adopted by all major platforms. Leading technology companies like Microsoft are backing passkeys as part of the FIDO Alliance, and numerous websites and apps are integrating support for passkeys.

The FIDO protocols rely on standard public/private key cryptography techniques to offer more secure authentication. When a user registers with an online service, their client device generates a new key pair. The private key is stored securely on the user's device, while the public key is registered with the service. To authenticate, the client device must prove that it possesses the private key by signing a challenge. The private keys can only be used after they're unlocked by the user using the Windows Hello unlock factor (biometrics or PIN).

FIDO protocols prioritize user privacy, as they're designed to prevent online services from sharing information or tracking users across different services. Additionally, any biometric

information used in the authentication process remains on the user's device and isn't transmitted across the network or to the service.

Passkeys compared to passwords

Passkeys have several advantages over passwords, including their ease of use and intuitive nature. Unlike passwords, passkeys are easy to create, don't need to be remembered, and don't need to be safeguarded. Additionally, passkeys are unique to each website or application, preventing their reuse. They're highly secure because they're only stored on the user's devices, with the service only storing public keys. Passkeys are designed to prevent attackers to guess or obtain them, which helps to make them resistant to phishing attempts where the attacker might try to trick the user into revealing the private key. Passkeys are enforced by the browsers or operating systems to only be used for the appropriate service, rather than relying on human verification. Finally, passkeys provide cross-device and cross-platform authentication, meaning that a passkey from one device can be used to sign in on another device.

Windows edition and licensing requirements

The following table lists the Windows editions that support passkeys:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Passkeys license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

User experience

Users can follow the steps in the following topics to create, use, and manage their passkeys on Windows:

- Create a passkey
- Use a passkey
- Manage passkeys

Passkeys in Bluetooth-restricted environments

For passkey cross-device authentication scenarios, both the Windows device and the mobile device must have Bluetooth enabled and connected to the Internet. This allows the user to authorize another device securely over Bluetooth without transferring or copying the passkey itself.

Some organizations restrict Bluetooth usage, which includes the use of passkeys. In such cases, organizations can allow passkeys by permitting Bluetooth pairing exclusively with passkey-enabled FIDO2 authenticators.

To limit the use of Bluetooth to only passkey use cases, use the Bluetooth Policy CSP and the DeviceInstallation Policy CSP.

Device configuration

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

To configure devices with Microsoft Intune, you can use a custom policy with these settings:

Expand table

Setting
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/Bluetooth/`AllowAdvertising • Data type: **Integer** • Value: `0` When set to `0`, the device doesn't send out advertisements.
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/Bluetooth/`AllowDiscoverableMode • Data type: **Integer** • Value: `0` When set to `0`, other devices can't detect the device.

Setting
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/Bluetooth/`AllowPrepairing • Data type: **Integer** • Value: `0` Prevents specific bundled Bluetooth peripherals from automatically pairing with the host device.
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/Bluetooth/`AllowPromptedProximalConnections • Data type: **Integer** • Value: `0` Prevents users from using Swift Pair and other proximity-based scenarios.
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/Bluetooth/`ServicesAllowedList • Data type: **String** • Value: `{0000FFFD-0000-1000-8000-00805F9B34FB};{0000FFF9-0000-1000-8000-00805F9B34FB}` Set a list of allowable Bluetooth services and profiles: - FIDO Alliance Universal Second Factor Authenticator service (`0000fffd-0000-1000-8000-00805f9b34fb`) - FIDO2 secure client-to-authenticator transport service (`0000FFF9-0000-1000-8000-00805F9B34FB`) For more information, see FIDO CTAP 2.1 standard specification and Bluetooth Assigned Numbers document.
• OMA-URI: `./Device/Vendor/MSFT/Policy/Config/DeviceInstallation/`PreventInstallationOfMatchingDeviceIDs • Data type: **String** • Value: `<enabled/><data id="DeviceInstall_IDs_Deny_Retroactive" value="true"/><data id="DeviceInstall_IDs_Deny_List" value="1BTH\MS_BTHPAN"/>` Disables the existing Bluetooth Personal Area Network (PAN) network adapter, preventing the installation of the Bluetooth Network Adapter that can be used for network connectivity or tethering.

> **Note**
>
> Once the settings are applied, if you try to pair a device via Bluetooth, it will initially pair and immediately disconnect. The Bluetooth device is blocked from loading and not available from Settings nor Device Manager.

Provide feedback

To provide feedback for passkeys, open **Feedback Hub** and use the category **Security and Privacy** > **Passkey**.

Last updated on 01/09/2026

Smart Card Technical Reference

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Smart Card Technical Reference describes the Windows smart card infrastructure for physical smart cards and how smart card-related components work in Windows. This document also contains information about tools that information technology (IT) developers and administrators can use to troubleshoot, debug, and deploy smart card-based strong authentication in the enterprise.

Audience

This document explains how the Windows smart card infrastructure works. To understand this information, you should have basic knowledge of public key infrastructure (PKI) and smart card concepts. This document is intended for:

- Enterprise IT developers, managers, and staff who are planning to deploy or are using smart cards in their organization.
- Smart card vendors who write smart card minidrivers or credential providers.

What are smart cards?

Smart cards are tamper-resistant portable storage devices that can enhance the security of tasks such as authenticating clients, signing code, securing e-mail, and signing in with a Windows domain account.

Smart cards provide:

- Tamper-resistant storage for protecting private keys and other forms of personal information
- Isolation of security-critical computations that involve authentication, digital signatures, and key exchange from other parts of the computer. These computations are performed on the smart card
- Portability of credentials and other private information between computers at work, home, or on the road

Smart cards can be used to sign in to domain accounts only, not local accounts. When you use a password to sign in interactively to a domain account, Windows uses the

Kerberos version 5 (v5) protocol for authentication. If you use a smart card, the operating system uses Kerberos v5 authentication with X.509 v3 certificates.

Virtual smart cards were introduced to alleviate the need for a physical smart card, the smart card reader, and the associated administration of that hardware.

> ⚠ **Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

In this technical reference

This reference contains the following topics:

- How Smart Card Sign-in Works in Windows
  - Smart Card Architecture
  - Certificate Requirements and Enumeration
  - Smart Card and Remote Desktop Services
  - Smart Cards for Windows Service
  - Certificate Propagation Service
  - Smart Card Removal Policy Service
- Smart Card Tools and Settings
  - Smart Cards Debugging Information
  - Smart Card Group Policy and Registry Settings
  - Smart Card Events

Feedback

Was this page helpful? Yes No

Provide product feedback

How Smart Card Sign-in Works in Windows

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for IT professional provides links to resources about the implementation of smart card technologies in the Windows operating system. It includes the following resources about the architecture, certificate management, and services that are related to smart card use:

- Smart Card Architecture: Learn about enabling communications with smart cards and smart card readers, which can be different according to the vendor that supplies them
- Certificate Requirements and Enumeration: Learn about requirements for smart card certificates based on the operating system, and about the operations that are performed by the operating system when a smart card is inserted into the computer
- Smart Card and Remote Desktop Services: Learn about using smart cards for remote desktop connections
- Smart Cards for Windows Service: Learn about how the Smart Cards for Windows service is implemented
- Certificate Propagation Service: Learn about how the certificate propagation service works when a smart card is inserted into a computer
- Smart Card Removal Policy Service: Learn about using Group Policy to control what happens when a user removes a smart card

Windows edition and licensing requirements

The following table lists the Windows editions that support Smart Cards for Windows Service:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Smart Cards for Windows Service license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Card Architecture

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional describes the system architecture that supports smart cards in the Windows operating system, including credential provider architecture and the smart card subsystem architecture.

Authentication is a process for verifying the identity of an object or person. When you authenticate an object, such as a smart card, the goal is to verify that the object is genuine. When you authenticate a person, the goal is to verify that you aren't dealing with an imposter.

In a networking context, authentication is the act of proving identity to a network application or resource. Typically, identity is proven by a cryptographic operation that uses a key only the user knows (such as with public key cryptography), or a shared key. The server side of the authentication exchange compares the signed data with a known cryptographic key to validate the authentication attempt. Storing the cryptographic keys in a secure central location makes the authentication process scalable and maintainable.

For smart cards, Windows supports a provider architecture that meets the secure authentication requirements and is extensible so that you can include custom credential providers. This topic includes information about:

- Credential provider architecture
- Smart card subsystem architecture

Credential provider architecture

The following table lists the components that are included in the interactive sign-in architecture:

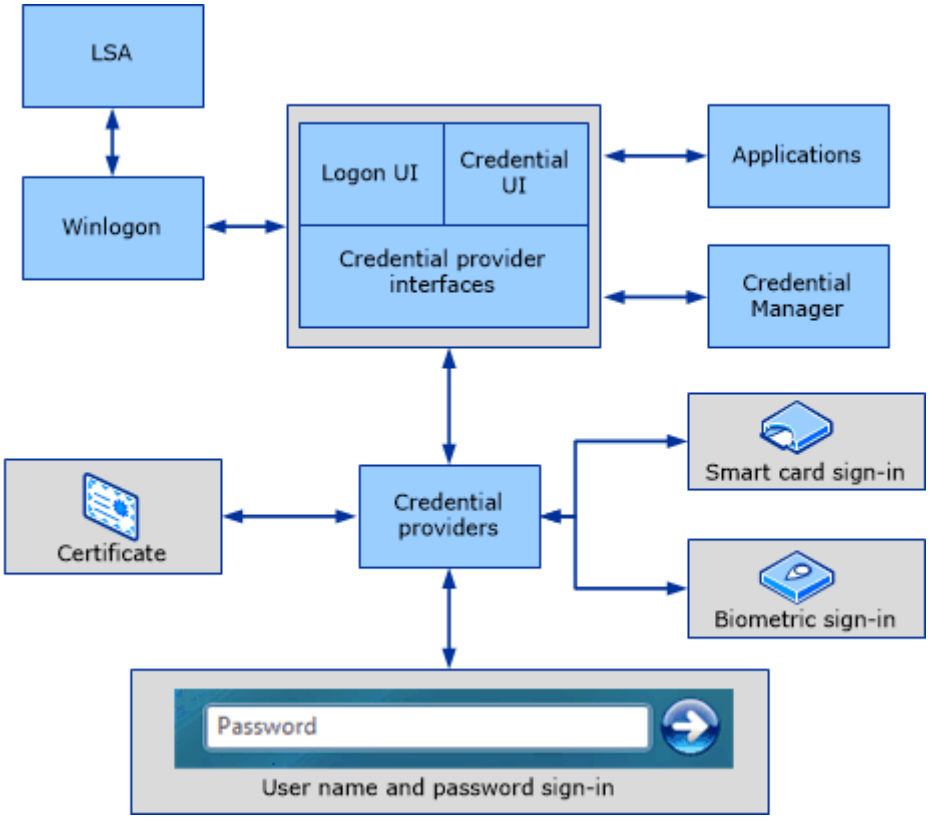
Expand table

Component	Description
Winlogon	Provides an interactive sign-in infrastructure.
Logon UI	Provides interactive UI rendering.
Credential providers	Describes credential information and serializing credentials.

Component	Description
(password and smart card)	
Local Security Authority (LSA)	Processes sign-in credentials.
Authentication packages	Includes NTLM and the Kerberos protocol. Communicates with server authentication packages to authenticate users.

Interactive sign-in in Windows begins when the user presses CTRL+ALT+DEL. The CTRL+ALT+DEL key combination is called a secure attention sequence (SAS). To keep other programs and processes from using it, Winlogon registers this sequence during the boot process.

After receiving the SAS, the UI then generates the sign-in tile from the information received from the registered credential providers. The following graphic shows the architecture for credential providers in the Windows operating system.



Typically, a user who signs in to a computer by using a local account or a domain account must enter a user name and password. These credentials are used to verify the user's identity. For smart card sign-in, a user's credentials are contained on the smart card's security chip. A smart card reader lets the computer interact with the security chip on the smart card. When users sign in with a smart card, they enter a personal identification number (PIN) instead of a user name and password.

Credential providers are in-process COM objects that run on the local system and are used to collect credentials. The Logon UI provides interactive UI rendering, Winlogon

provides interactive sign-in infrastructure, and credential providers work with both of these components to help gather and process credentials.

Winlogon instructs the Logon UI to display credential provider tiles after it receives an SAS event. The Logon UI queries each credential provider for the number of credentials it wants to enumerate. Credential providers have the option of specifying one of these tiles as the default. After all providers have enumerated their tiles, the Logon UI displays them to the user. The user interacts with a tile to supply the proper credentials. The Logon UI submits these credentials for authentication.

Combined with supporting hardware, credential providers can extend the Windows operating system to enable users to sign in by using biometrics (for example, fingerprint, retinal, or voice recognition), password, PIN, smart card certificate, or any custom authentication package. Enterprises and IT professionals can develop and deploy custom authentication mechanisms for all domain users, and they may explicitly require users to use this custom sign-in mechanism.

> ⓘ **Note**
>
> Credential providers are not enforcement mechanisms. They are used to gather and serialize credentials. The LSA and authentication packages enforce security.

Credential providers can be designed to support single sign-in (SSO). In this process, they authenticate users to a secure network access point (by using RADIUS and other technologies) for signing in to the computer. Credential providers are also designed to support application-specific credential gathering, and they can be used for authentication to network resources, joining computers to a domain, or to provide administrator consent for User Account Control (UAC).

Multiple credential providers can coexist on a computer.

Credential providers must be registered on a computer running Windows, and they're responsible for:

- Describing the credential information that is required for authentication
- Handling communication and logic with external authentication authorities
- Packaging credentials for interactive and network sign-in

> ⓘ **Note**
>
> The Credential Provider API does not render the UI. It describes what needs to be rendered.

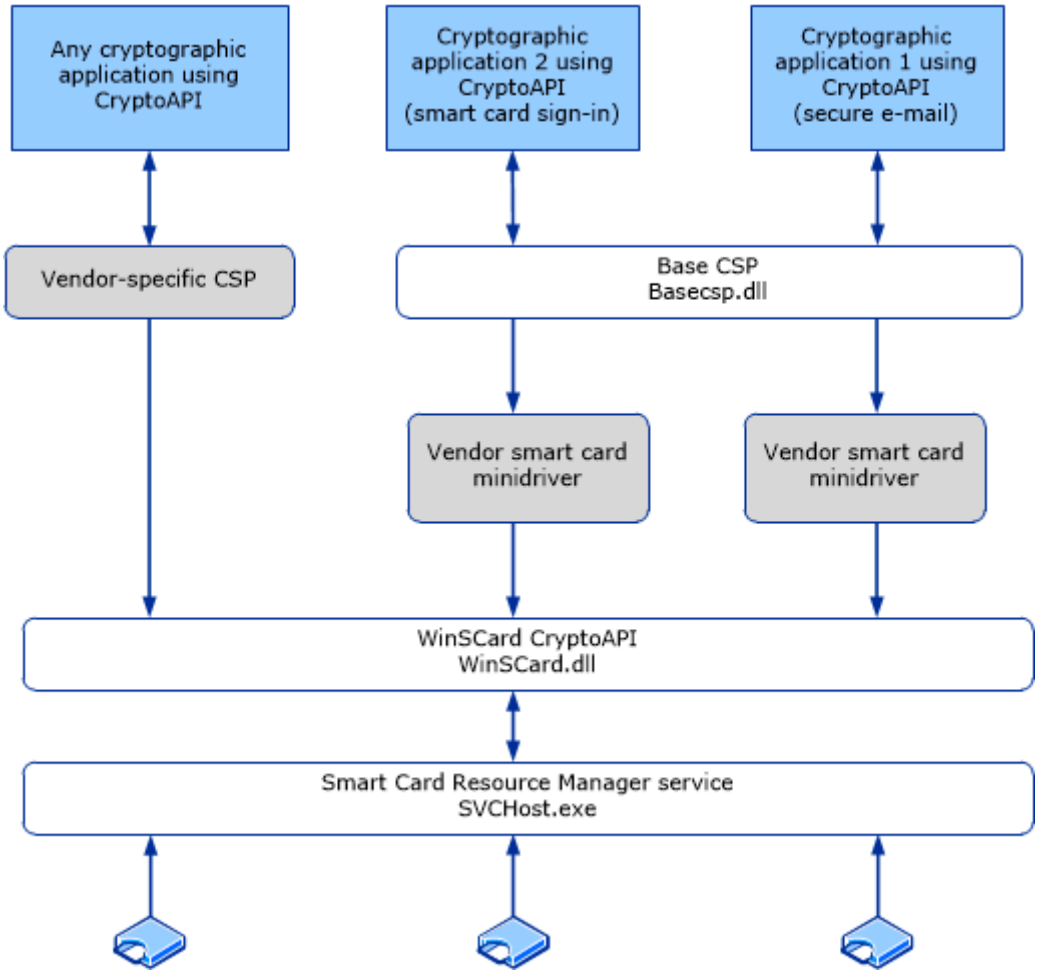
Only the password credential provider is available in safe mode.
The smart card credential provider is available in safe mode during networking.

Smart card subsystem architecture

Vendors provide smart cards and smart card readers, and in many cases the vendors are different for the smart card and the smart card reader. Drivers for smart card readers are written to the Personal Computer/Smart Card (PC/SC) standard. Each smart card must have a Cryptographic Service Provider (CSP) that uses the CryptoAPI interfaces to enable cryptographic operations, and the WinSCard APIs to enable communications with smart card hardware.

Base CSP and smart card minidriver architecture

The following graphic shows the relationship between the CryptoAPI, CSPs, the Smart Card Base Cryptographic Service Provider (Base CSP), and smart card minidrivers.



Caching with Base CSP and smart card KSP

Smart card architecture uses caching mechanisms to help streamlining operations and to improve a user's access to a PIN.

- Data caching: The data cache provides for a single process to minimize smart card I/O operations
- PIN caching: The PIN cache helps the user from having to reenter a PIN each time the smart card is unauthenticated

Data caching

Each CSP implements the current smart card data cache separately. The Base CSP implements a robust caching mechanism that allows a single process to minimize smart card I/O operations.

The existing global cache works as follows:

1. The application requests a cryptographic operation. For example, a user certificate is to be read from the smart card
2. The CSP checks its cache for the item
3. If the item isn't found in the cache, or if the item is cached but isn't up-to-date, the item is read from the smart card
4. After any item has been read from the smart card, it's added to the cache. Any existing out-of-date copy of that item is replaced

Three types of objects or data are cached by the CSP: pins (for more information, see PIN caching), certificates, and files. If any of the cached data changes, the corresponding object is read from the smart card in successive operations. For example, if a file is written to the smart card, the CSP cache becomes out-of-date for the files, and other processes read the smart card at least once to refresh their CSP cache.

The global data cache is hosted in the Smart Cards for Windows service. Windows includes two public smart card API calls, SCardWriteCache and SCardReadCache. These API calls make global data caching functionality available to applications. Every smart card that conforms to the smart card minidriver specification has a 16-byte card identifier. This value is used to uniquely identify cached data that pertains to a given smart card. The standard Windows GUID type is used. These APIs allow an application to add data to and read data from the global cache.

PIN caching

The PIN cache protects the user from entering a PIN every time the smart card is unauthenticated. After a smart card is authenticated, it will not differentiate among

host-side applications—any application can access private data on the smart card.

To mitigate this, the smart card enters an exclusive state when an application authenticates to the smart card. However, this means that other applications can't communicate with the smart card and will be blocked. Therefore, such exclusive connections are minimized. The issue is that a protocol (such as the Kerberos protocol) requires multiple signing operations. Therefore, the protocol requires exclusive access to the smart card over an extended period, or it requires multiple authentication operations. This is where the PIN cache is used to minimize exclusive use of the smart card without forcing the user to enter a PIN multiple times.

The following example illustrates how this works. In this scenario, there are two applications: Outlook and Internet Explorer. The applications use smart cards for different purposes.

1. The user starts Outlook and tries to send a signed e-mail. The private key is on the smart card
2. Outlook prompts the user for the smart card PIN. The user enters the correct PIN
3. E-mail data is sent to the smart card for the signature operation. The Outlook client formats the response and sends the e-mail
4. The user opens Internet Explorer and tries to access a protected site that requires Transport Layer Security (TLS) authentication for the client
5. Internet Explorer prompts the user for the smart card PIN. The user enters the correct PIN
6. The TLS-related private key operation occurs on the smart card, and the user is authenticated and signed in
7. The user returns to Outlook to send another signed e-mail. This time, the user isn't prompted for a PIN because the PIN is cached from the previous operation. Similarly, if the user uses Internet Explorer again for another operation, Internet Explorer won't prompt the user for a PIN

The Base CSP internally maintains a per-process cache of the PIN. The PIN is encrypted and stored in memory. The functions that are used to secure the PIN are RtlEncryptMemory, RtlDecryptMemory, and RtlSecureZeroMemory, which will empty buffers that contained the PIN.

Smart card selection

The following sections in this article describe how Windows uses the smart card architecture to select the correct smart card reader software, provider, and credentials for a successful smart card sign-in:

- Container specification levels
- Container operations
- Context flags
- Create a new container in silent context
- Smart card selection behavior
- Make a smart card reader match
- Make a smart card match
- Open an existing default container (no reader specified)
- Open an existing GUID-named container (no reader specified)
- Create a new container (no reader specified)
- Delete a container

Container specification levels

In response to a CryptAcquireContext call in CryptoAPI, the Base CSP tries to match the container that the caller specifies to a specific smart card and reader. The caller can provide a container name with varying levels of specificity, as shown in the following table, and sorted from most-specific to least-specific requests.

Similarly, in response to a NCryptOpenKey call in CNG, the smart card KSP tries to match the container the same way, and it takes the same container format, as shown in the following table.

> **Note**
>
> Before opening a key by using the smart card KSP, a call to NCryptOpenStorageProvider (`MS_SMART_CARD_KEY_STORAGE_PROVIDER`) must be made.

Expand table

Type	Name	Format
I	Reader Name and Container Name	`\.<Reader Name><Container Name>`
II	Reader Name and Container Name (NULL)	`\.<Reader Name>`
III	Container Name Only	`<Container Name>`
IV	Default Container (NULL) Only	NULL

The Base CSP and smart card KSP cache smart card handle information about the calling process and about the smart cards the process has accessed. When searching for a smart card container, the Base CSP or smart card KSP first checks its cache for the

process. If the cached handle is invalid or no match is found, the SCardUIDlg API is called to get the card handle.

Container operations

The following three container operations can be requested by using CryptAcquireContext:

1. Create a new container. (The CNG equivalent of CryptAcquireContext with dwFlags set to CRYPT_NEWKEYSET is NCryptCreatePersistedKey.)
2. Open an existing container. (The CNG equivalent of CryptAcquireContext to open the container is NCryptOpenKey.)
3. Delete a container. (The CNG equivalent of CryptAcquireContext with dwFlags set to CRYPT_DELETEKEYSET is NCryptDeleteKey.)

The heuristics that are used to associate a cryptographic handle with a particular smart card and reader are based on the container operation requested and the level of container specification used.

The following table shows the restrictions for the container creation operation.

Expand table

Specification	Restriction
No silent context	Key container creation must always be able to show UI, such as the PIN prompt.
No overwriting existing containers	If the specified container already exists on the chosen smart card, choose another smart card or cancel the operation.

Context flags

The following table shows the context flags used as restrictions for the container creation operation.

Expand table

Flag	Description
`CRYPT_SILENT`	No UI can be displayed during this operation.
`CRYPT_MACHINE_KEYSET`	No cached data should be used during this operation.

Flag	Description
`CRYPT_VERIFYCONTEXT`	Only public data can be accessed on the smart card.

In addition to container operations and container specifications, you must consider other user options, such as the CryptAcquireContext flags, during smart card selection.

> **Important**
>
> The CRYPT_SILENT flag cannot be used to create a new container.

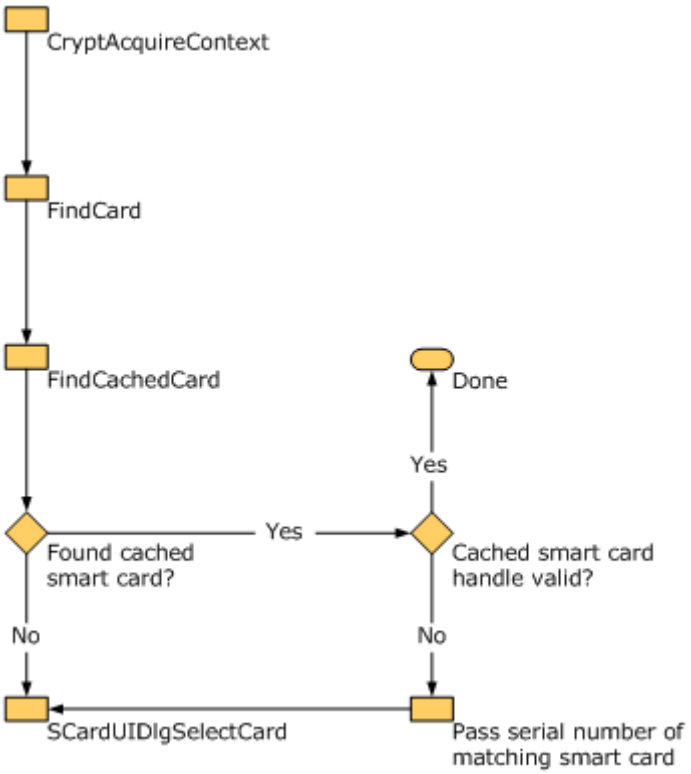
Create a new container in silent context

Applications can call the Base CSP with `CRYPT_DEFAULT_CONTAINER_OPTIONAL`, set the PIN in silent context, and then create a new container in silent context. This operation occurs as follows:

1. Call CryptAcquireContext by passing the smart card reader name in as a type II container specification level, and specifying the `CRYPT_DEFAULT_CONTAINER_OPTIONAL` flag
2. Call CryptSetProvParam by specifying `PP_KEYEXCHANGE_PIN` or `PP_SIGNATURE_PIN` and a null-terminated ASCII PIN.
3. Release the context acquired in Step 1
4. Call CryptAcquireContext with `CRYPT_NEWKEYSET`, and specify the type I container specification level
5. Call CryptGenKey to create the key

Smart card selection behavior

In some of the following scenarios, the user can be prompted to insert a smart card. If the user context is silent, this operation fails and no UI is displayed. Otherwise, in response to the UI, the user can insert a smart card or select **Cancel**. If the user cancels the operation, the operation fails. The flow chart shows the selection steps performed by the Windows operating system.



In general, smart card selection behavior is handled by the SCardUIDlgSelectCard API. The Base CSP interacts with this API by calling it directly. The Base CSP also sends callback functions that have the purpose of filtering and matching candidate smart cards. Callers of CryptAcquireContext provide smart card matching information. Internally, the Base CSP uses a combination of smart card serial numbers, reader names, and container names to find specific smart cards.

Each call to `SCardUI *` may result in additional information read from a candidate smart card. The Base CSP smart card selection callbacks cache this information.

Make a smart card reader match

For type I and type II container specification levels, the smart card selection process is less complex because only the smart card in the named reader can be considered a match. The process for matching a smart card with a smart card reader is:

1. Find the requested smart card reader. If it can't be found, the process fails (this requires a cache search by reader name)
2. If no smart card is in the reader, the user is prompted to insert a smart card. (this is only in nonsilent mode; if the call is made in silent mode, it fails)
3. For container specification level II only, the name of the default container on the chosen smart card is determined
4. To open an existing container or delete an existing container, find the specified container. If the specified container can't be found on this smart card, the user is prompted to insert a smart card

5. If the system attempts to create a new container, if the specified container already exists on this smart card, the process fails

Make a smart card match

For container specification levels III and IV, a broader method is used to match an appropriate smart card with a user context, because multiple cached smart cards might meet the criteria provided.

Open an existing default container (no reader specified)

> ⓘ **Note**
>
> This operation requires that you use the smart card with the Base CSP.

1. For each smart card that has been accessed by the Base CSP and the handle and container information are cached, the Base CSP looks for a valid default container. An operation is attempted on the cached SCARDHANDLE to verify its validity. If the smart card handle isn't valid, the Base CSP continues to search for a new smart card
2. If a matching smart card isn't found in the Base CSP cache, the Base CSP calls to the smart card subsystem. SCardUIDlgSelectCard() is used with an appropriate callback filter to find a matching smart card with a valid default container

Open an existing GUID-named container (no reader specified)

> ⓘ **Note**
>
> This operation requires that you use the smart card with the Base CSP.

1. For each smart card that is already registered with the Base CSP, search for the requested container. Attempt an operation on the cached SCARDHANDLE to verify its validity. If the smart card handle isn't valid, the smart card's serial number is passed to the `SCardUI *` API to continue searching for this specific smart card (rather than only a general match for the container name)
2. If a matching smart card isn't found in the Base CSP cache, a call is made to the smart card subsystem. `SCardUIDlgSelectCard()` is used with an appropriate callback filter to find a matching smart card with the requested container. Or, if a

smart card serial number resulted from the search in Step 1, the callback filter attempts to match the serial number, not the container name

Create a new container (no reader specified)

> ⓘ **Note**
>
> This operation requires that you use the smart card with the Base CSP.

If the PIN isn't cached, no CRYPT_SILENT is allowed for the container creation because the user must be prompted for a PIN, at a minimum.

For other operations, the caller may be able to acquire a *verify* context against the default container `CRYPT_DEFAULT_CONTAINER_OPTIONAL` and then make a call with CryptSetProvParam to cache the user PIN for subsequent operations.

1. For each smart card already known by the CSP, refresh the stored SCARDHANDLE and make the following checks:
   a. If the smart card has been removed, continue the search
   b. If the smart card is present, but it already has the named container, continue the search
   c. If the smart card is available, but a call to CardQueryFreeSpace indicates that the smart card has insufficient storage for an additional key container, continue the search
   d. Otherwise, use the first available smart card that meets the above criteria for the container creation
2. If a matching smart card isn't found in the CSP cache, make a call to the smart card subsystem. The callback that is used to filter enumerated smart cards verifies that a candidate smart card doesn't already have the named container, and that CardQueryFreeSpace indicates the smart card has sufficient space for an additional container. If no suitable smart card is found, the user is prompted to insert a smart card

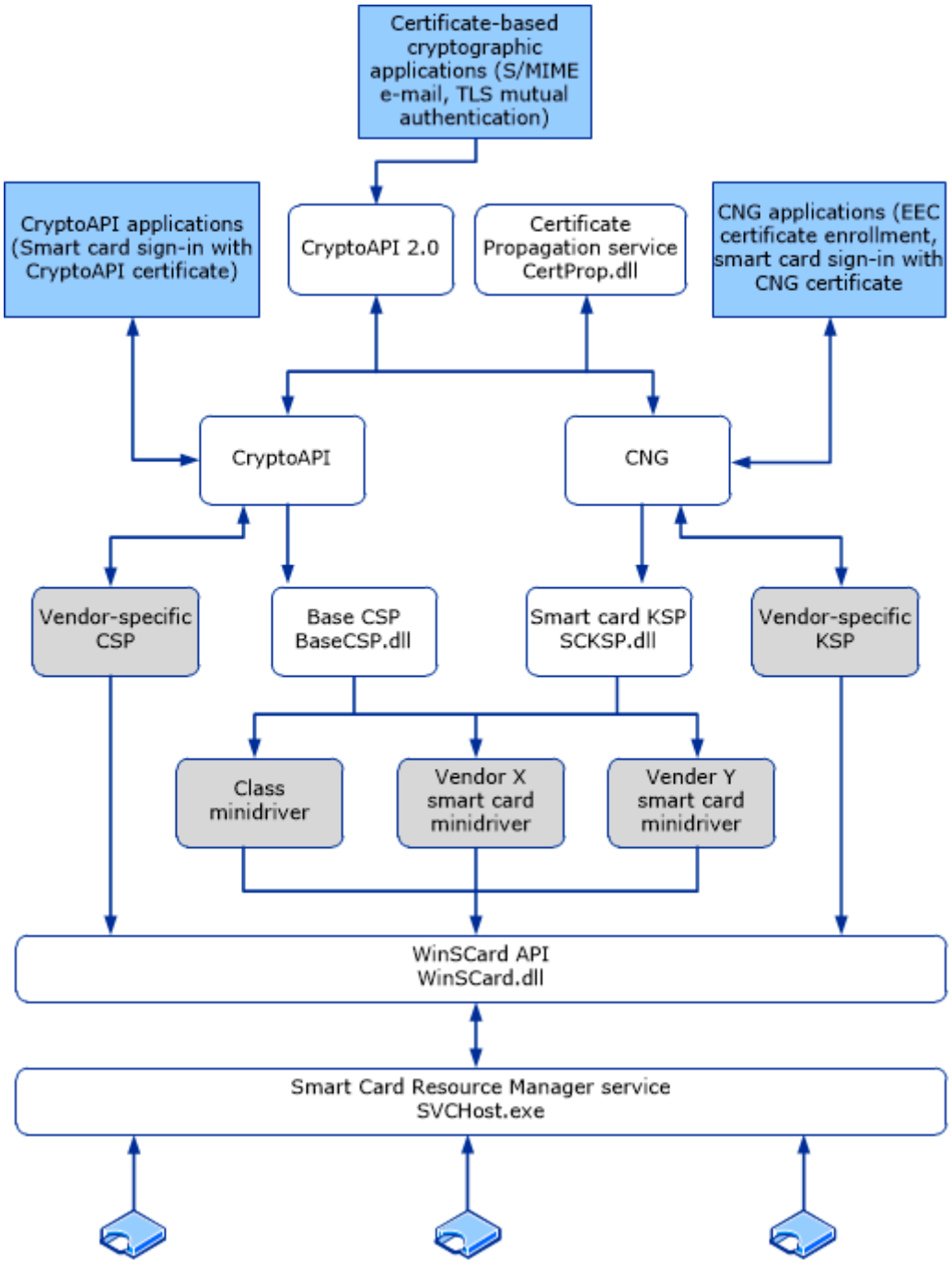
Delete a container

1. If the specified container name is NULL, the default container is deleted. Deleting the default container causes a new default container to be selected arbitrarily. For this reason, this operation isn't recommended
2. For each smart card already known by the CSP, refresh the stored SCARDHANDLE and make the following checks:

   a. If the smart card doesn't have the named container, continue the search
   b. If the smart card has the named container, but the smart card handle is no longer valid, store the serial number of the matching smart card and pass it to SCardUI
3. If a matching smart card isn't found in the CSP cache, make a call to the smart card subsystem. The callback that is used to filter enumerated smart cards should verify that a candidate smart card has the named container. If a serial number was provided as a result of the previous cache search, the callback should filter enumerated smart cards on serial number rather than on container matches. If the context is non-silent and no suitable smart card is found, display UI that prompts the user to insert a smart card

Base CSP and KSP-based architecture in Windows

The following diagram shows the Cryptography architecture that is used by the Windows operating system.



Base CSP and smart card KSP properties in Windows

> ⓘ Note
>
> The API definitions are located in WinCrypt.h and WinSCard.h.

Expand table

Property	Description
`PP_USER_CERTSTORE`	- Used to return an `HCERTSTORE` that contains all user certificates on the smart card - Read-only (used only by `CryptGetProvParam`)

Property	Description
	- Caller responsible for closing the certificate store - Certificate encoded using `PKCS_7_ASN_ENCODING` or `X509_ASN_ENCODING` - CSP should set `KEY_PROV_INFO` on certificates - Certificate store should be assumed to be an in-memory store - Certificates should have a valid `CRYPT_KEY_PROV_INFO` as a property
`PP_ROOT_CERTSTORE`	- Read and Write (used by `CryptGetProvParam` and `CryptSetProvParam`) - Used to write a collection of root certificates to the smart card or return `HCERTSTORE`, which contains root certificates from the smart card - Used primarily for joining a domain by using a smart card - Caller responsible for closing the certificate store
`PP_SMARTCARD_READER`	- Read-only (used only by `CryptGetProvParam`) - Returns the smart card reader name as an ANSI string that is used to construct a fully qualified container name (that is, a smart card reader plus a container)
`PP_SMARTCARD_GUID`	- Return smart card GUID (also known as a serial number), which should be unique for each smart card - Used by the certificate propagation service to track the source of a root certificate
`PP_UI_PROMPT`	- Used to set the search string for the `SCardUIDlgSelectCard` card insertion dialog box - Persistent for the entire process when it's set - Write-only (used only by `CryptSetProvParam`)

Implications for CSPs in Windows

Cryptographic Service Providers (CSPs), including custom smart card CSPs, continue to be supported but this approach isn't recommended. Using the existing Base CSP and smart card KSP with the smart card minidriver model for smart cards provides significant benefits in terms of performance, and PIN and data caching. One minidriver can be configured to work under CryptoAPI and CNG layers. This provides benefits from enhanced cryptographic support, including elliptic curve cryptography and AES.

If a smart card is registered by a CSP and a smart card minidriver, the one that was installed most recently will be used to communicate with the smart card.

Write a smart card minidriver, CSP, or KSP

CSPs and KSPs are meant to be written only if specific functionality isn't available in the current smart card minidriver architecture. For example, the smart card minidriver architecture supports hardware security modules, so a minidriver could be written for a

hardware security module, and a CSP or KSP may not be required unless it's needed to support algorithms that aren't implemented in the Base CSP or smart card KSP.

For more information about how to write a smart card minidriver, CSP, or KSP, see Smart Card Minidrivers.

Feedback

Was this page helpful? Yes No

Provide product feedback

Certificate Requirements and Enumeration

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional and smart card developers describes how certificates are managed and used for smart card sign-in.

When a smart card is inserted, the following steps are performed.

> **Note**
>
> Unless otherwise mentioned, all operations are performed silently (CRYPT_SILENT is passed to CryptAcquireContext).

1. The smart card resource manager database searches for the smart card's cryptographic service provider (CSP).
2. A qualified container name is constructed by using the smart card reader name, and it's passed to the CSP. The format is `\\.<Reader name>\`
3. CryptAcquireContext is called to retrieve a context to the default container. If a failure occurs, the smart card is unusable for smart card sign-in.
4. The name of the container is retrieved by using the PP_CONTAINER parameter with CryptGetProvParam.
5. Using the context acquired in Step 3, the CSP is queried for the PP_USER_CERTSTORE parameter. For more information, see Smart Card Architecture. If the operation is successful, the name of a certificate store is returned, and the program flow skips to Step 8.
6. If the operation in Step 5 fails, the default container context from Step 3 is queried for the AT_KEYEXCHANGE key.
7. The certificate is then queried from the key context by using KP_CERTIFICATE. The certificate is added to an in-memory certificate store.
8. For each certificate in the certificate store from Step 5 or Step 7, the following checks are performed:

a. The certificate must be valid, based on the computer system clock (not expired or valid with a future date)
b. The certificate must not be in the AT_SIGNATURE part of a container
c. The certificate must have a valid user principal name (UPN)
d. The certificate must have the digital signature key usage
e. The certificate must have the smart card logon EKU

   Any certificate that meets these requirements is displayed to the user with the certificate's UPN (or e-mail address or subject, depending on the presence of the certificate extensions)

9. The process then chooses a certificate, and the PIN is entered
10. LogonUI.exe packages the information and sends it to Lsass.exe to process the sign-in attempt
11. If successful, `LogonUI.exe` closes. This causes the context acquired in Step 3 to be released

Smart card sign-in flow in Windows

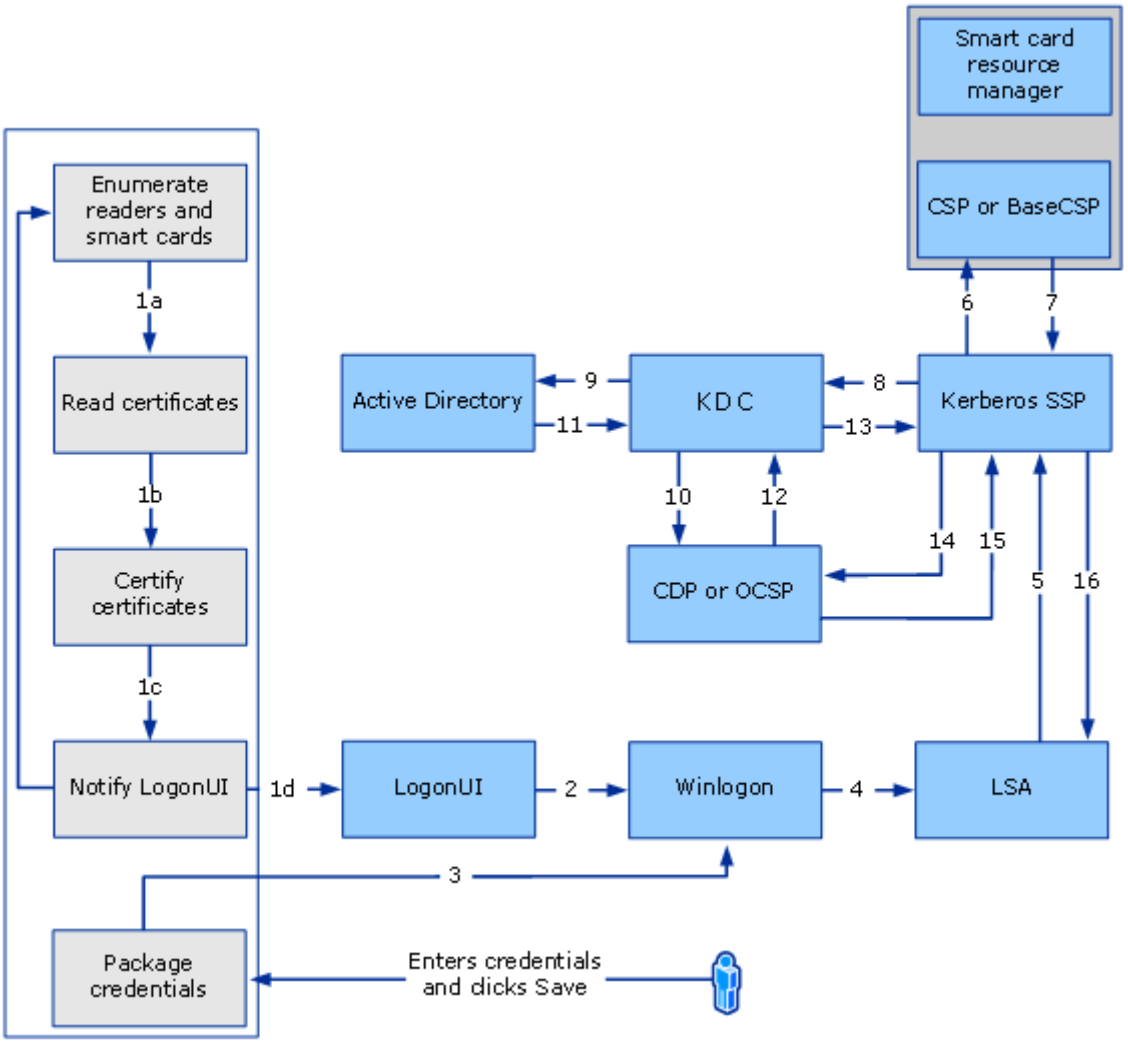
Most issues during authentication occur because of session behavior changes. When changes occur, the Local Security Authority (LSA) doesn't reacquire the session context; it relies instead on the Cryptographic Service Provider to handle the session change.

Client certificates that don't contain a UPN in the `subjectAltName` (SAN) field of the certificate can be enabled for sign-in, which supports a wider variety of certificates and supports multiple sign-in certificates on the same card.

Support for multiple certificates on the same card is enabled by default. New certificate types must be enabled through Group Policy.

If you enable the **Allow signature keys valid for Logon** credential provider policy, any certificates that are available on the smart card with a signature-only key are listed on the sign-in screen. This allows users to select their sign-in experience. If the policy is disabled or not configured, smart card signature-key-based certificates aren't listed on the sign-in screen.

The following diagram illustrates how smart card sign-in works in the supported versions of Windows.



Smart card sign-in flow

Following are the steps that are performed during a smart card sign-in:

1. Winlogon requests the sign-in UI credential information

2. Asynchronously, smart card resource manager starts, and the smart card credential provider does the following:
   a. Gets credential information (a list of known credentials, or if no credentials exist, the smart card reader information that Windows detected)
   b. Gets a list of smart card readers (by using the WinSCard API) and the list of smart cards inserted in each of them
   c. Enumerates each card to verify that a sign-in certificate that is controlled by Group Policy is present. If the certificate is present, the smart card credential provider copies it into a temporary, secure cache on the computer or terminal

> ⓘ **Note**

> Smartcard cache entries are created for certificates with a subject name or with a subject key identifier. If the certificate has a subject name, it is stored with an index that is based on the subject name and certificate issuer. If another certificate with the same subject name and certificate issuer is used, it will replace the existing cached entry. A change in this behavior, allows for the condition when the certificate does not have a subject name, the cache is created with an index that is based on the subject key identifier and certificate issuer. If another certificate has the same the subject key identifier and certificate issuer, the cache entry is replaced. When certificates have neither a subject name nor subject key identifier, a cached entry is not created.

   a. Notifies the sign-in UI that it has new credentials

3. The sign-in UI requests the new credentials from the smart card credential provider. As a response, the smart card credential provider provides each sign-in certificate to the sign-in UI, and corresponding sign-in tiles are displayed. The user selects a smart card-based sign-in certificate tile, and Windows displays a PIN dialog box

4. The user enters the PIN, and then presses ENTER. The smart card credential provider encrypts the PIN

5. The credential provider that resides in the LogonUI system collects the PIN. As part of packaging credentials in the smart card credential provider, the data is packaged in a KERB_CERTIFICATE_LOGON structure. The main contents of the KERB_CERTIFICATE_LOGON structure are the smart card PIN, CSP data (such as reader name and container name), user name, and domain name. User name is required if the sign-in domain isn't in the same forest because it enables a certificate to be mapped to multiple user accounts

6. The credential provider wraps the data (such as the encrypted PIN, container name, reader name, and card key specification) and sends it back to LogonUI

7. Winlogon presents the data from LogonUI to the LSA with the user information in LSALogonUser

8. LSA calls the Kerberos authentication package (Kerberos SSP) to create a Kerberos authentication service request (KRB_AS_REQ), which containing a preauthenticator (as specified in RFC 4556: Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)).

   If the authentication is performed by using a certificate that uses a digital signature, the preauthentication data consists of the user's public certificate and

the certificate that is digitally signed with the corresponding private key.

If the authentication is performed by using a certificate that uses key encipherment, the preauthentication data consists of the user's public certificate and the certificate that is encrypted with the corresponding private key.

9. To sign the request digitally (as per RFC 4556), a call is made to the corresponding CSP for a private key operation. Because the private key in this case is stored in a smart card, the smart card subsystem is called, and the necessary operation is completed. The result is sent back to the Kerberos security support provider (SSP).

10. The Kerberos SSP sends an authentication request for a ticket-granting-ticket (TGT) (per RFC 4556) to the Key Distribution Center (KDC) service that runs on a domain controller.

11. The KDC finds the user's account object in Active Directory Domain Services (AD DS), as detailed in Client certificate requirements and mappings, and uses the user's certificate to verify the signature.

12. The KDC validates the user's certificate (time, path, and revocation status) to ensure that the certificate is from a trusted source. The KDC uses CryptoAPI to build a certification path from the user's certificate to a root certification authority (CA) certificate that resides in the root store on the domain controller. The KDC then uses CryptoAPI to verify the digital signature on the signed authenticator that was included in the preauthentication data fields. The domain controller verifies the signature and uses the public key from the user's certificate to prove that the request originated from the owner of the private key that corresponds to the public key. The KDC also verifies that the issuer is trusted and appears in the NTAUTH certificate store.

13. The KDC service retrieves user account information from AD DS. The KDC constructs a TGT, which is based on the user account information that it retrieves from AD DS. The TGT's authorization data fields include the user's security identifier (SID), the SIDs for universal and global domain groups to which the user belongs, and (in a multidomain environment) the SIDs for any universal groups of which the user is a member.

14. The domain controller returns the TGT to the client as part of the KRB_AS_REP response.

> **Note**
>
> The KRB_AS_REP packet consists of:

- Privilege attribute certificate (PAC)
- User's SID
- SIDs of any groups of which the user is a member
- A request for ticket-granting service (TGS)
- Preauthentication data

TGT is encrypted with the master key of the KDC, and the session key is encrypted with a temporary key. This temporary key is derived based on RFC 4556. Using CryptoAPI, the temporary key is decrypted. As part of the decryption process, if the private key is on a smart card, a call is made to the smart card subsystem by using the specified CSP to extract the certificate corresponding to the user's public key. (Programmatic calls for the certificate include CryptAcquireContext, CryptSetProvParam with the PIN, CryptgetUserKey, and CryptGetKeyParam.) After the temporary key is obtained, the Kerberos SSP decrypts the session key.

15. The client validates the reply from the KDC (time, path, and revocation status). It first verifies the KDC's signature by the construction of a certification path from the KDC's certificate to a trusted root CA, and then it uses the KDC's public key to verify the reply signature.

16. Now that a TGT has been obtained, the client obtains a service ticket, which is used to sign in to the local computer.

17. With success, LSA stores the tickets and returns a success message to LSALogonUser. After this success message is issued, user profile for the device is selected and set, Group Policy refresh is instantiated, and other actions are performed.

18. After the user profile is loaded, the Certification Propagation Service (CertPropSvc) detects this event, reads the certificates from the smart card (including the root certificates), and then populates them into the user's certificate store (MYSTORE)

19. CSP to smart card resource manager communication happens on the LRPC Channel.

20. On successful authentication, certificates are propagated to the user's store asynchronously by the Certificate Propagation Service (CertPropSvc).

21. When the card is removed, certificates in the temporary secure cache store are removed. The Certificates are no longer available for sign-in, but they remain in the user's certificate store.

> **Note**
>
> A SID is created for each user or group at the time a user account or a group account is created within the local security accounts database or within AD DS. The SID never changes, even if the user or group account is renamed.

For more information about the Kerberos protocol, see Microsoft Kerberos.

By default, the KDC verifies that the client's certificate contains the smart card client authentication EKU szOID_KP_SMARTCARD_LOGON. However, if enabled, the **Allow certificates with no extended key usage certificate attribute** Group Policy setting allows the KDC to not require the SC-LOGON EKU. SC-LOGON EKU isn't required for account mappings that are based on the public key.

KDC certificate

Active Directory Certificate Services provides three kinds of certificate templates:

- Domain controller
- Domain controller authentication
- Kerberos authentication

Depending on the configuration of the domain controller, one of these types of certificates is sent as a part of the AS_REP packet.

Client certificate requirements and mappings

Certificate requirements are listed by versions of the Windows operating system. Certificate mapping describes how information from the certificate is mapped to the user account.

Certificate requirements

Expand table

Component	Requirements
CRL distribution point location	Not required
Key usage	Digital signature

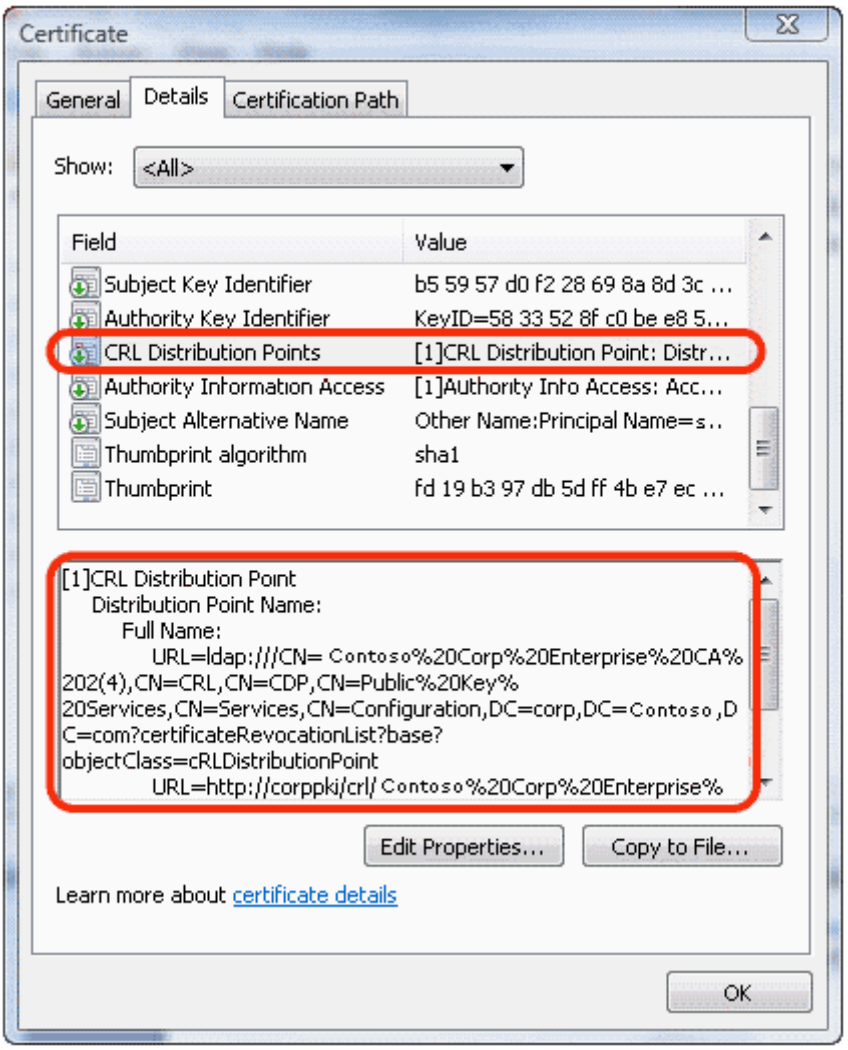
Component	Requirements
Basic constraints	Not required
extended key usage (EKU)	The smart card sign-in object identifier isn't required. **Note** If an EKU is present, it must contain the smart card sign-in EKU. Certificates with no EKU can be used for sign-in.
Subject alternative name	E-mail ID isn't required for smart card sign-in.
Subject	Not required
Key exchange (AT_KEYEXCHANGE field)	Not required for smart card sign-in certificates if a Group Policy setting is enabled. (By default, Group Policy settings aren't enabled.)
CRL	Not required
UPN	Not required
Notes	You can enable any certificate to be visible for the smart card credential provider.

Client certificate mappings

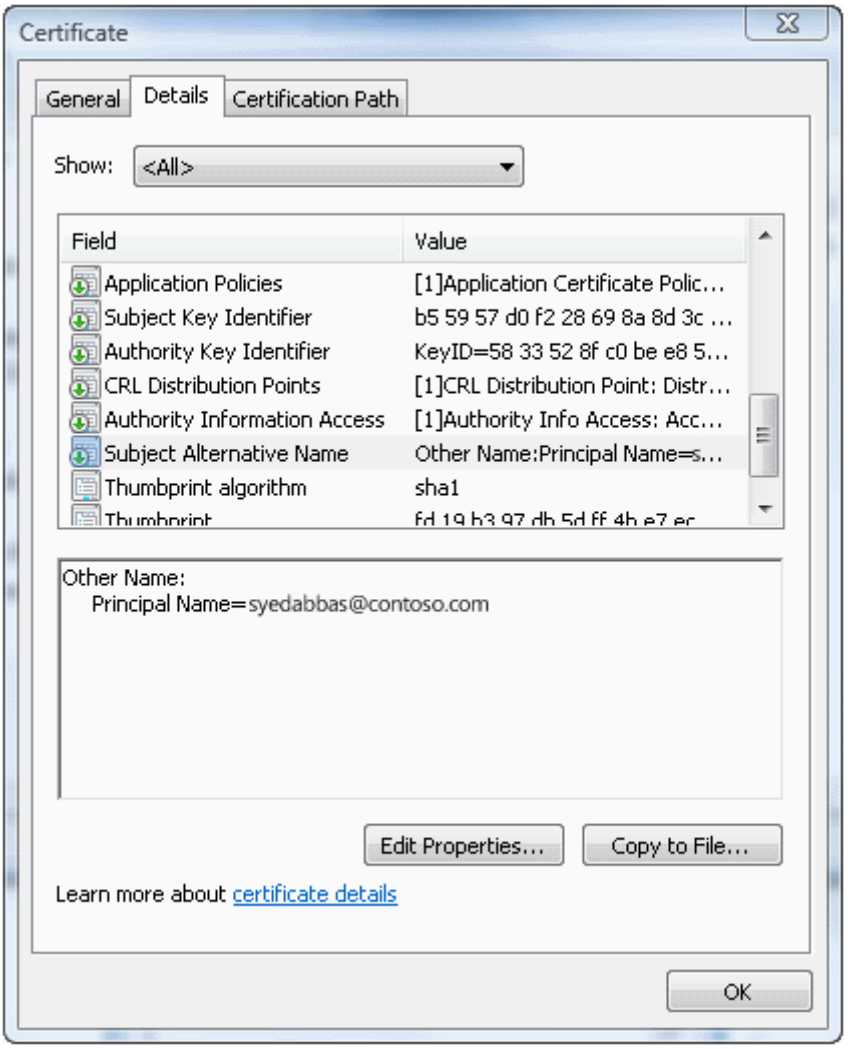
Certificate mapping is based on the UPN that is contained in the subjectAltName (SAN) field of the certificate. Client certificates that don't contain information in the SAN field are also supported.

SSL/TLS can map certificates that don't have SAN, and the mapping is done by using the AltSecID attributes on client accounts. The X509 AltSecID, which is used by SSL/TLS client authentication is of the form "X509: `<Issuer Name>` `<Subject Name`. The `<Issuer Name>` and `<Subject Name>` are taken from the client certificate, with '\r' and '\n' replaced with ','.

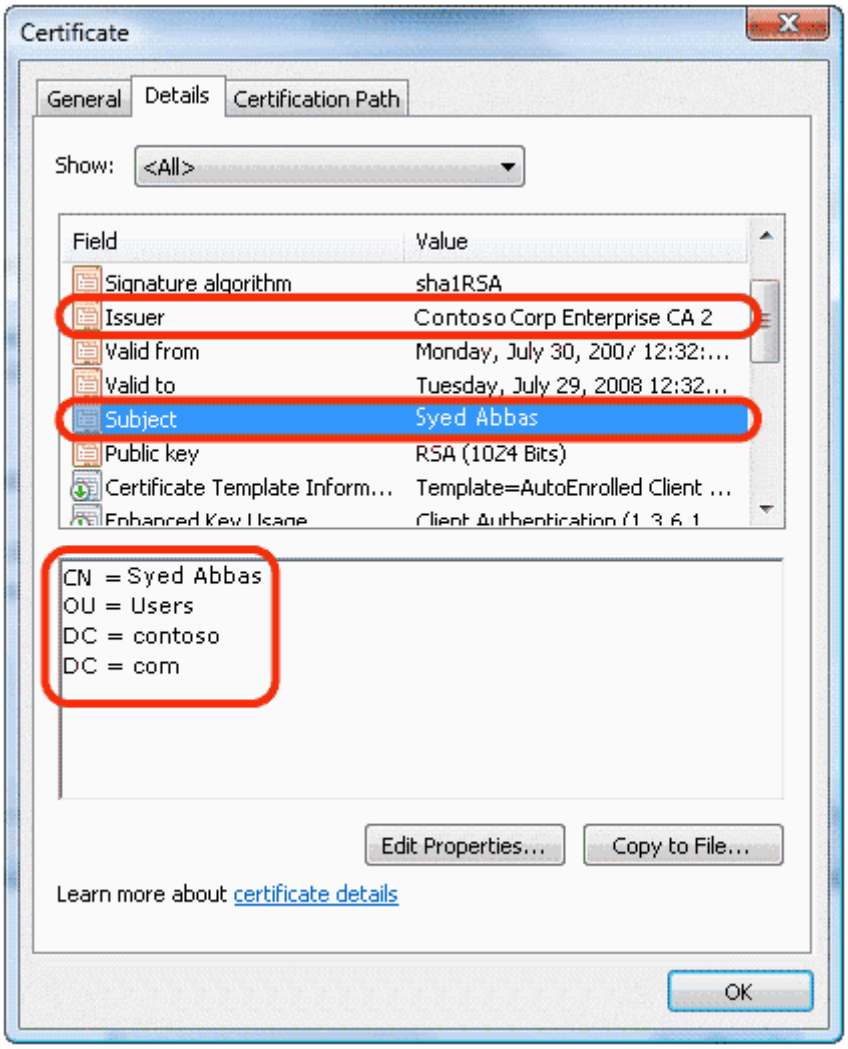
Certificate revocation list distribution points



UPN in Subject Alternative Name field

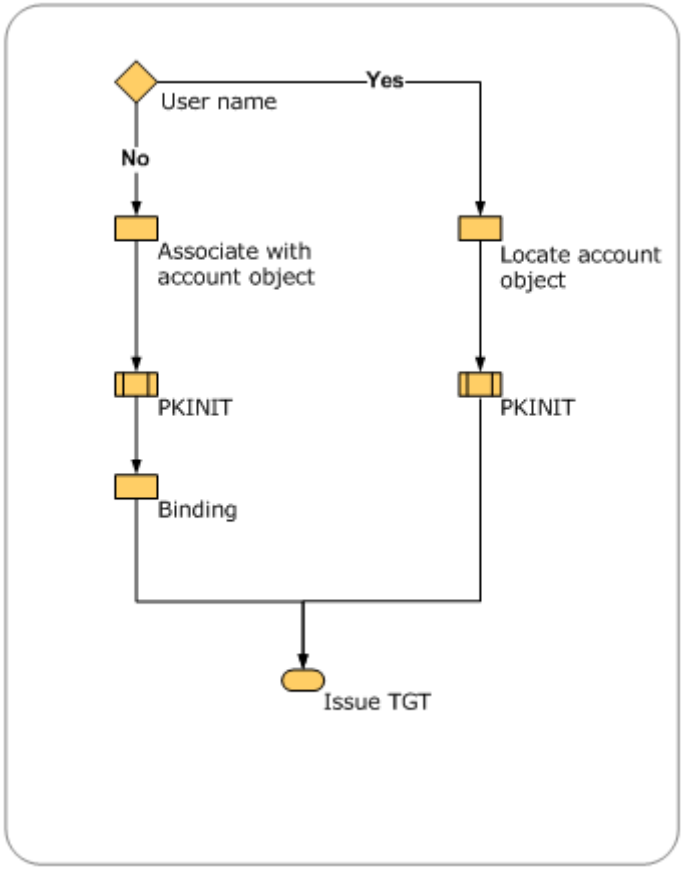


Subject and Issuer fields



This account mapping is supported by the KDC in addition to six other mapping methods. The following figure demonstrates a flow of user account mapping logic that is used by the KDC.

High-level flow of certificate processing for sign-in



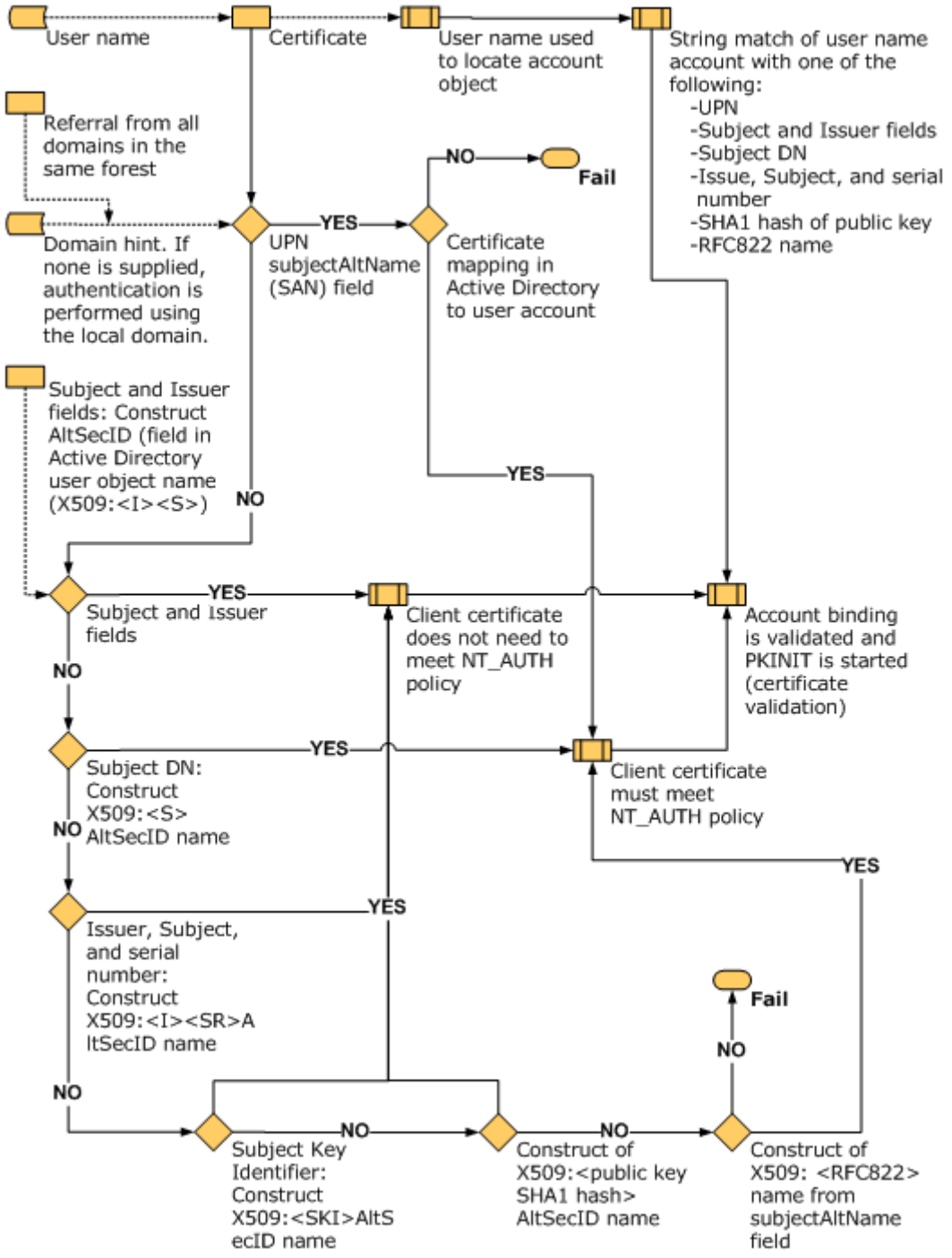
The certificate object is parsed to look for content to perform user account mapping.

- When a user name is provided with the certificate, the user name is used to locate the account object. This operation is the fastest, because string matching occurs
- When only the certificate object is provided, multiple operations are performed to locate the user name to map the user name to an account object
- When no domain information is available for authentication, the local domain is used by default. If any other domain is to be used for lookup, a domain name hint should be provided to perform the mapping and binding

Mapping based on generic attributes isn't possible because there's no generic API to retrieve attributes from a certificate. Currently, the first method that locates an account successfully stops the search. But a configuration error occurs if two methods map the same certificate to different user accounts when the client doesn't supply the client name through the mapping hints.

The following figure illustrates the process of mapping user accounts for sign-in in the directory by viewing various entries in the certificate.

Certificate processing logic



NT_AUTH policy is best described in the CERT_CHAIN_POLICY_NT_AUTH parameter section of the CertVerifyCertificateChainPolicy function. For more information, see CertVerifyCertificateChainPolicy.

Smart card sign-in for a single user with one certificate into multiple accounts

A single user certificate can be mapped to multiple accounts. For example, a user might be able to sign in to a user account and also to sign in as a domain administrator. The mapping is done by using the constructed AltSecID based on attributes from client

accounts. For information about how this mapping is evaluated, see Client certificate requirements and mappings.

> ⓘ **Note**
>
> Because each account has a different user name, we recommend that you enable the **Allow user name hint** Group Policy setting (**X509HintsNeeded** registry key) to provide the optional fields that allow users to enter their user names and domain information to sign in.

Based on the information that is available in the certificate, the sign-in conditions are:

1. If no UPN is present in the certificate:
   a. Sign-in can occur in the local forest or in another forest if a single user with one certificate needs to sign in to different accounts
   b. A hint must be supplied if mapping isn't unique (for example, if multiple users are mapped to the same certificate)
2. If a UPN is present in the certificate:
   a. The certificate can't be mapped to multiple users in the same forest
   b. The certificate can be mapped to multiple users in different forests. For a user to sign in to other forests, an X509 hint must be supplied to the user

Smart card sign-in for multiple users into a single account

A group of users might sign in to a single account (for example, an administrator account). For that account, user certificates are mapped so that they're enabled for sign-in.

Several distinct certificates can be mapped to a single account. For this to work properly, the certificate can't have UPNs.

For example, if Certificate1 has CN=CNName1, Certificate2 has CN=User1, and Certificate3 has CN=User2, the AltSecID of these certificates can be mapped to a single account by using the Active Directory Users and Computers name mapping.

Smart card sign-in across forests

For account mapping to work across forests, particularly in cases where there isn't enough information available on the certificate, the user might enter a hint in the form

of a user name, such as *domain\user*, or a fully qualified UPN such as `user@contoso.com`.

> **Note**
>
> For the hint field to appear during smart card sign-in, the **Allow user name hint** Group Policy setting (**X509HintsNeeded** registry key) must be enabled on the client.

OCSP support for PKINIT

Online Certificate Status Protocol (OCSP), which is defined in RFC 2560, enables applications to obtain timely information about the revocation status of a certificate. Because OCSP responses are small and well bound, constrained clients might want to use OCSP to check the validity of the certificates for Kerberos on the KDC, to avoid transmission of large CRLs, and to save bandwidth on constrained networks. For information about CRL registry keys, see Smart Card Group Policy and Registry Settings.

The KDCs in Windows attempt to get OCSP responses and use them when available. This behavior can't be disabled. CryptoAPI for OCSP caches OCSP responses and the status of the responses. The KDC supports only OCSP responses for the signer certificate.

Windows client computers attempt to request the OCSP responses and use them in the reply when they're available. This behavior can't be disabled.

Smart card root certificate requirements for use with domain sign-in

For sign-in to work in a smart card-based domain, the smart card certificate must meet the following conditions:

- The KDC root certificate on the smart card must have an HTTP CRL distribution point listed in its certificate
- The smart card sign-in certificate must have the HTTP CRL distribution point listed in its certificate
- The CRL distribution point must have a valid CRL published and a delta CRL, if applicable, even if the CRL distribution point is empty
- The smart card certificate must contain one of the following:
  - A subject field that contains the DNS domain name in the distinguished name. If it doesn't, resolution to an appropriate domain fails, so Remote Desktop

Services and the domain sign-in with the smart card fail

- A UPN where the domain name resolves to the actual domain. For example, if the domain name is `Engineering.Corp.Contoso`, the UPN is `username@engineering.corp.contoso.com`. If any part of the domain name is omitted, the Kerberos client can't find the appropriate domain

To allow smart card sign-in to a domain in these versions, do the following:

1. Enable HTTP CRL distribution points on the CA
2. Restart the CA
3. Reissue the KDC certificate
4. Issue or reissue the smart card sign-in certificate
5. Propagate the updated root certificate to the smart card that you want to use for the domain sign-in

The workaround is to enable the **Allow user name hint** Group Policy setting (**X509HintsNeeded** registry key), which allows the user to supply a hint in the credentials user interface for domain sign-in.

If the client computer isn't joined to the domain or if it's joined to a different domain, the client computer can resolve the server domain only by looking at the distinguished name on the certificate, not the UPN. For this scenario to work, the certificate requires a full subject, including `DC=<DomainControllerName>`, for domain name resolution.

To deploy root certificates on a smart card for the currently joined domain, you can use the following command:

Windows Command Prompt

```
certutil.exe -scroots update
```

For more information about this option for the command-line tool, see -SCRoots.

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Card and Remote Desktop Services

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional describes the behavior of Remote Desktop Services when you implement smart card sign-in.

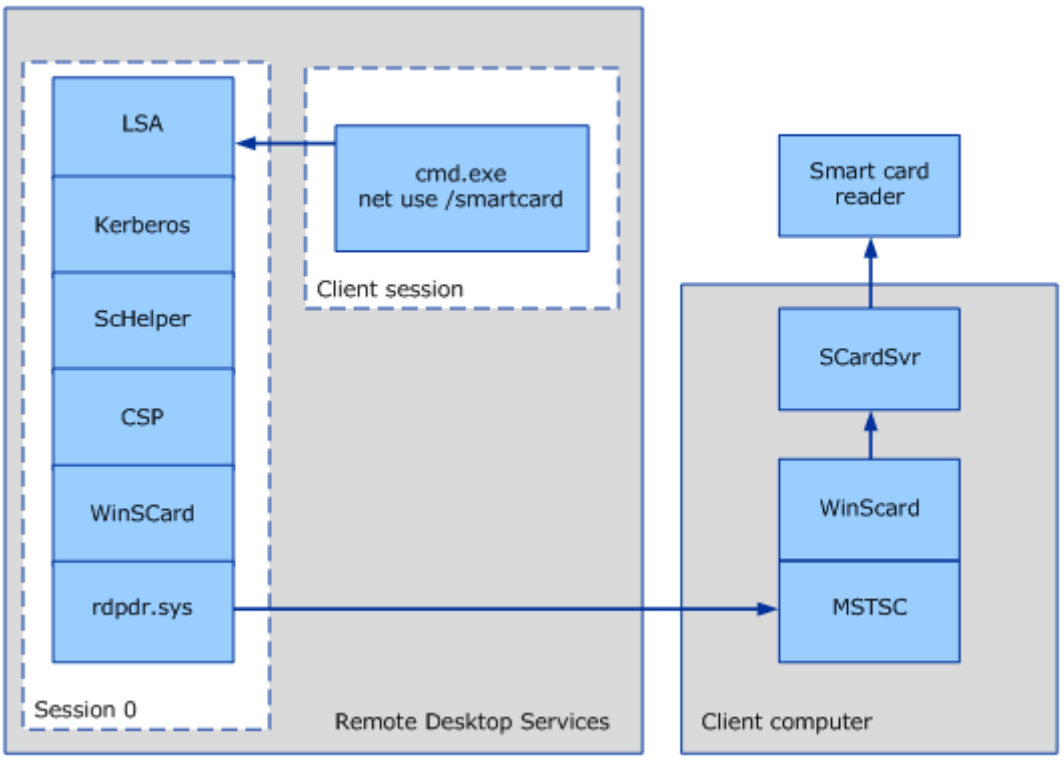
Smart card redirection logic and *WinSCard API* are combined to support multiple redirected sessions into a single process.

Smart card support is required to enable many Remote Desktop Services scenarios. These include:

- Using Fast User Switching or Remote Desktop Services. A user isn't able to establish a redirected smart card-based remote desktop connection. That is, the connect attempt isn't successful in Fast User Switching or from a Remote Desktop Services session
- Enabling *Encrypting File System* (EFS) to locate the user's smart card reader from the *Local Security Authority* (LSA) process in Fast User Switching or in a Remote Desktop Services session. If EFS isn't able to locate the smart card reader or certificate, EFS can't decrypt user files

Remote Desktop Services redirection

In a Remote Desktop scenario, a user is using a remote server for running services, and the smart card is local to the computer that the user is using. In a smart card sign-in scenario, the smart card service on the remote server redirects to the smart card reader that is connected to the local computer where the user is trying to sign in.



Remote Desktop redirection

Notes about the redirection model:

1. This scenario is a remote sign-in session on a computer with Remote Desktop Services. In the remote session (labeled as *Client session*), the user runs `net use /smartcard`
2. Arrows represent the flow of the PIN after the user types the PIN at the command prompt until it reaches the user's smart card in a smart card reader that is connected to the Remote Desktop Connection (RDC) client computer
3. The authentication is performed by the LSA in session 0
4. The CryptoAPI processing is performed in the LSA (`lsass.exe`). This is possible because RDP redirector (`rdpdr.sys`) allows per-session, rather than per-process, context
5. The ScHelper library is a CryptoAPI wrapper that is specific to the Kerberos protocol
6. The redirection decision is made on a per smart card context basis, based on the session of the thread that performs the `SCardEstablishContext` call

RD Session Host server single sign-in experience

As a part of the Common Criteria compliance, the RDC client must be configurable to use Credential Manager to acquire and save the user's password or smart card PIN. Common Criteria compliance requires that applications not have direct access to the user's password or PIN.

Common Criteria compliance requires specifically that the password or PIN never leave the LSA unencrypted. A distributed scenario should allow the password or PIN to travel between one trusted LSA and another, and it can't be unencrypted during transit.

When smart card-enabled single sign-in (SSO) is used for Remote Desktop Services sessions, users still need to sign in for every new Remote Desktop Services session. However, the user isn't prompted for a PIN more than once to establish a Remote Desktop Services session. For example, after the user double-clicks a Microsoft Word document icon that resides on a remote computer, the user is prompted to enter a PIN. This PIN is sent by using a secure channel that the credential SSP has established. The PIN is routed back to the RDC client over the secure channel and sent to Winlogon. The user doesn't receive any additional prompts for the PIN, unless the PIN is incorrect or there are smart card-related failures.

Remote Desktop Services and smart card sign-in

Remote Desktop Services enables users to sign in with a smart card by entering a PIN on the RDC client computer and sending it to the RD Session Host server in a manner similar to authentication that is based on user name and password.

In addition, group policy settings that are specific to Remote Desktop Services need to be enabled for smart card-based sign-in.

To enable smart card sign-in to a Remote Desktop Session Host (RD Session Host) server, the Key Distribution Center (KDC) certificate must be present on the RDC client computer. If the computer isn't in the same domain or workgroup, the following command can be used to deploy the certificate:

Windows Command Prompt

```
certutil.exe -dspublish NTAuthCA "DSCDPContainer"
```

The `DSCDPContainer` Common Name (CN) is usually the name of the certification authority.

Example:

Windows Command Prompt

```
certutil -dspublish NTAuthCA <CertFile> "CN=NTAuthCertificates,CN=Public Key
Services,CN=Services,CN=Configuration,DC=engineering,DC=contoso,DC=com"
```

For information about this option for the command-line tool, see -dsPublish.

Remote Desktop Services and smart card sign-in across domains

To enable remote access to resources in an enterprise, the root certificate for the domain must be provisioned on the smart card. From a computer that is joined to a domain, run the following command at the command line:

Windows Command Prompt

```
certutil.exe -scroots update
```

For information about this option for the command-line tool, see -SCRoots.

For Remote Desktop Services across domains, the KDC certificate of the RD Session Host server must also be present in the client computer's NTAUTH store. To add the store, run the following command at the command line:

Windows Command Prompt

```
certutil -addstore -enterprise NTAUTH <CertFile>
```

Where *CertFile* is the root certificate of the KDC certificate issuer.

For information about this option for the command-line tool, see -addstore.

> **Note**
>
> To sign in with a smart card from a computer that is not joined to a domain, the smart card must contain the root certification of the domain controller. A public key infrastructure (PKI) secure channel cannot be established without the root certification of the domain controller.

Sign-in to Remote Desktop Services across a domain works only if the UPN in the certificate uses the following form: `<ClientName>@<DomainDNSName>`.

The UPN in the certificate must include a domain that can be resolved. Otherwise, the Kerberos protocol can't determine which domain to contact. You can resolve this issue by enabling GPO X509 domain hints. For more information about this setting, see Smart Card Group Policy and Registry Settings.

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Cards for Windows Service

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional and smart card developers describes how the Smart Cards for Windows service (formerly called Smart Card Resource Manager) manages readers and application interactions.

The Smart Cards for Windows service provides the basic infrastructure for all other smart card components as it manages smart card readers and application interactions on the computer. It is fully compliant with the specifications set by the PC/SC Workgroup. For information about these specifications, see the PC/SC Workgroup Specifications website.

The Smart Cards for Windows service runs in the context of a local service, and it is implemented as a shared service of the services host (svchost) process. The Smart Cards for Windows service, Scardsvr, has the following service description:

PowerShell

```
<serviceData
    dependOnService="PlugPlay"
    description="@%SystemRoot%\System32\SCardSvr.dll,-5"
    displayName="@%SystemRoot%\System32\SCardSvr.dll,-1"
    errorControl="normal"
    group="SmartCardGroup"
    imagePath="%SystemRoot%\system32\svchost.exe -k
LocalServiceAndNoImpersonation"
    name="SCardSvr"
    objectName="NT AUTHORITY\LocalService"
    requiredPrivileges="SeCreateGlobalPrivilege,SeChangeNotifyPrivilege"
    sidType="unrestricted"
    start="demand"
    type="win32ShareProcess"
    >
  <failureActions resetPeriod="900">
      <actions>
         <action
             delay="120000"
             type="restartService"
         />
         <action
             delay="300000"
             type="restartService"
         />
         <action
              delay="0"
```

```
                type="none"
            />
        </actions>
    </failureActions>
    <securityDescriptor name="ServiceXSecurity"/>
</serviceData>

    <registryKeys buildFilter="">
        <registryKey
keyName="HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SCardSvr\Param
eters">
        <registryValue
            name="ServiceDll"
            value="%SystemRoot%\System32\SCardSvr.dll"
            valueType="REG_EXPAND_SZ"
            />
        <registryValue
            name="ServiceMain"
            value="CalaisMain"
            valueType="REG_SZ"
            />
        <registryValue
            name="ServiceDllUnloadOnStop"
            value="1"
            valueType="REG_DWORD"
            />
        </registryKey>
    </registryKeys>
```

> **Note**
>
> For winscard.dll to be invoked as the proper class installer, the INF file for a smart card reader must specify the following for **Class** and **ClassGUID**:
>
> `Class=SmartCardReader` `ClassGuid={50DD5230-BA8A-11D1-BF5D-0000F805F530}`

By default, the service is configured for manual mode. Creators of smart card reader drivers must configure their INFs so that they start the service automatically and winscard.dll files call a predefined entry point to start the service during installation. The entry point is defined as part of the **SmartCardReader** class, and it is not called directly. If a device advertises itself as part of this class, the entry point is automatically invoked to start the service when the device is inserted. Using this method ensures that the service is enabled when it is needed, but it is also disabled for users who do not use smart cards.

When the service is started, it performs several functions:

1. It registers itself for service notifications

2. It registers itself for Plug and Play (PnP) notifications related to device removal and additions
3. It initializes its data cache and a global event that signals that the service has started

> **Note**
>
> For smart card implementations, consider sending all communications in Windows operating systems with smart card readers through the Smart Cards for Windows service. This provides an interface to track, select, and communicate with all drivers that declare themselves members of the smart card reader device group.

The Smart Cards for Windows service categorizes each smart card reader slot as a unique reader, and each slot is also managed separately, regardless of the device's physical characteristics. The Smart Cards for Windows service handles the following high-level actions:

- Device introduction
- Reader initialization
- Notifying clients of new readers
- Serializing access to readers
- Smart card access
- Tunneling of reader-specific commands

See also

How Smart Card Sign-in Works in Windows

Feedback

Was this page helpful? Yes No

Provide product feedback

Certificate propagation service

Article • 10/29/2024 •

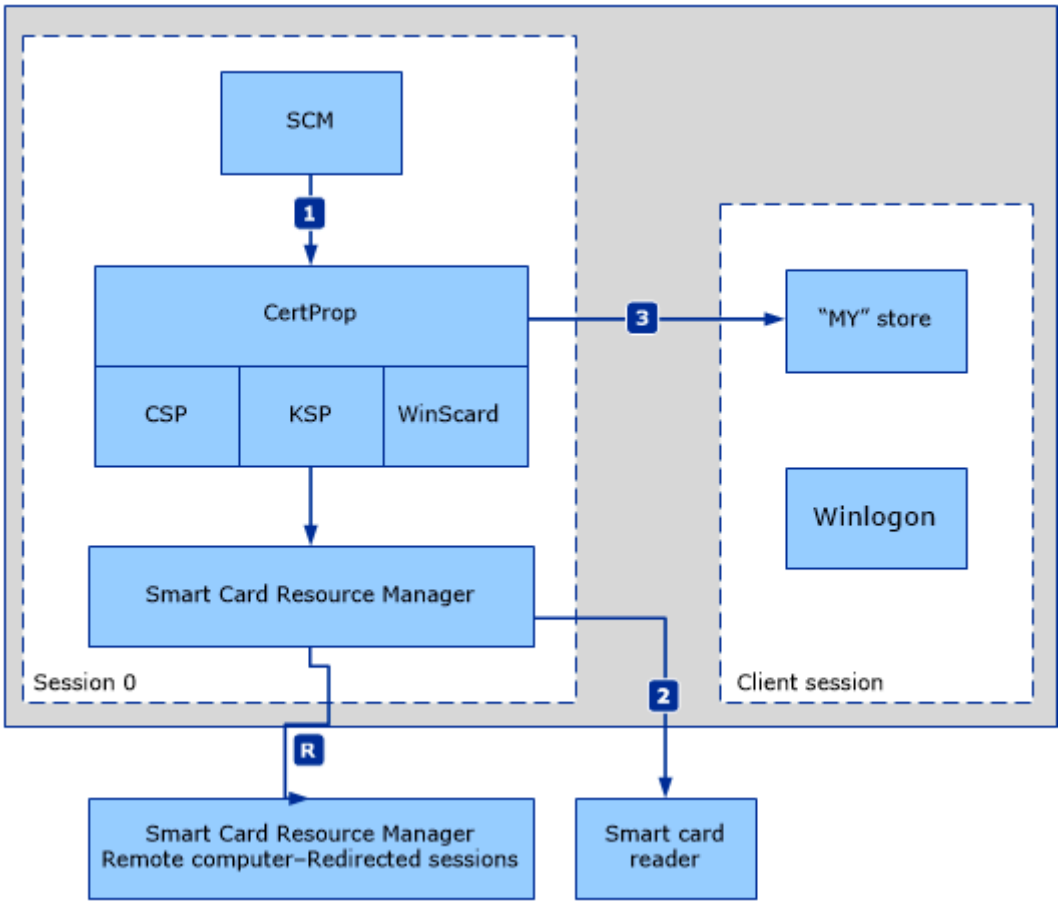
Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The certificate propagation service (CertPropSvc) is a Windows service that activates when a user inserts a smart card in a reader that is attached to the device. The action causes the certificates to be read from the smart card. The certificates are then added to the user's Personal store. Certificate propagation service actions are controlled by using Group Policy. For more information, see Smart Card Group Policy and Registry Settings.

> **Note**
>
> The certificate propagation service must be running for smart card Plug and Play to work.

The following figure shows the flow of the certificate propagation service. The action begins when a signed-in user inserts a smart card.

1. The arrow labeled **1** indicates that the Service Control Manager (SCM) notifies the certificate propagation service (CertPropSvc) when a user signs in, and CertPropSvc begins to monitor the smart cards in the user session
2. The arrow labeled **R** represents the possibility of a remote session and the use of smart card redirection
3. The arrow labeled **2** indicates the certification to the reader
4. The arrow labeled **3** indicates the access to the certificate store during the client session



5. A signed-in user inserts a smart card
6. CertPropSvc is notified that a smart card was inserted
7. CertPropSvc reads all certificates from all inserted smart cards. The certificates are written to the user's personal certificate store

> **Note**
>
> The certificate propagation service is started as a Remote Desktop Services dependency.

Properties of the certificate propagation service include:

- `CERT_STORE_ADD_REPLACE_EXISTING_INHERIT_PROPERTIES` adds certificates to a user's Personal store
- If the certificate has the `CERT_ENROLLMENT_PROP_ID` property (as defined by `wincrypt.h`), it filters empty requests and places them in the current user's request store, but it doesn't propagate them to the user's Personal store
- The service doesn't propagate any computer certificates to a user's Personal store or propagate user certificates to a computer store
- The service propagates certificates according to Group Policy options that are set, which might include:

- **Turn on certificate propagation from the smart card** specifies whether a user's certificate should be propagated
- **Turn on root certificate propagation from smart card** specifies whether root certificates should be propagated
- **Configure root certificate cleanup** specifies how root certificates are removed

Root certificate propagation service

Root certificate propagation is responsible for the following smart card deployment scenarios when public key infrastructure (PKI) trust hasn't yet been established:

- Joining the domain
- Accessing a network remotely

In both cases, the computer isn't joined to a domain, and therefore, trust isn't being managed by group policy. However, the objective is to authenticate to a remote server, such as the domain controller. Root certificate propagation provides the ability to use the smart card to include the missing trust chain.

When the smart card is inserted, the certificate propagation service propagates any root certificates on the card to the trusted smart card root computer certificate stores. This process establishes a trust relationship with the enterprise resources. You might also use a subsequent cleanup action when the user's smart card is removed from the reader, or when the user signs out. This is configurable with group policy. For more information, see Smart Card Group Policy and Registry Settings.

For more information about root certificate requirements, see Smart card root certificate requirements for use with domain sign-in.

See also

How Smart Card Sign-in Works in Windows

Feedback

Was this page helpful? Yes No

Provide product feedback

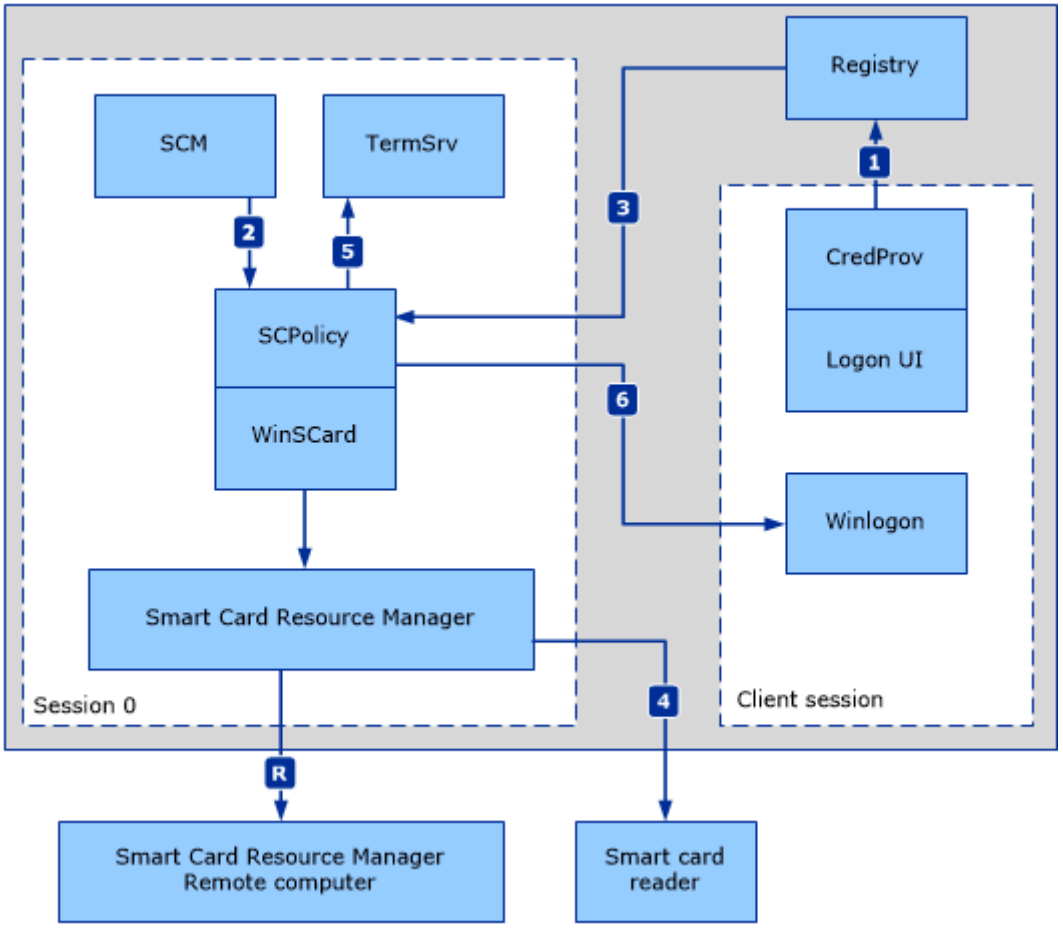
Smart Card Removal Policy Service

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the role of the removal policy service (`ScPolicySvc`) in smart card implementations.

The smart card removal policy service is applicable when a user signs in with a smart card and then removes that smart card from the reader. The action that is performed when the smart card is removed is controlled by group policy settings. For more information, see Smart Card Group Policy and Registry Settings.



The numbers in the diagram represent the following actions:

1. `Winlogon` isn't directly involved in monitoring for smart card removal events. The sequence of steps that are involved when a smart card is removed begins with the smart card credential provider in the sign-in UI process. When a user successfully signs in with a smart card, the smart card credential provider captures the reader name. This information is then stored in the registry with the session identifier where the sign-in was initiated

2. The smart card resource manager service notifies the smart card removal policy service that a sign-in occurred
3. `ScPolicySvc` retrieves the smart card information that the smart card credential provider stored in the registry. This call is redirected if the user is in a remote session. If the smart card is removed, `ScPolicySvc` is notified
4. `ScPolicySvc` calls Remote Desktop Services to take the appropriate action if the request is to sign out the user or to disconnect the user's session, which might result in data loss. If the setting is configured to lock the computer when the smart card is removed, `ScPolicySvc` sends a message to Winlogon to lock the computer.

See also

How Smart Card Sign-in Works in Windows

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Card Tools and Settings

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional and smart card developer links to information about smart card debugging, settings, and events.

This section of the Smart Card Technical Reference contains information about the following:

- Smart Cards Debugging Information: Learn about tools and services in supported versions of Windows to help identify certificate issues
- Smart Card Group Policy and Registry Settings: Learn about smart card-related Group Policy settings and registry keys that can be set on a per-computer basis, including how to edit and apply Group Policy settings to local or domain computers
- Smart Card Events: Learn about events that can be used to manage smart cards in an organization, including how to monitor installation, use, and errors

See also

Smart Card Technical Reference

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Card Troubleshooting

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article explains tools and services that smart card developers can use to help identify certificate issues with the smart card deployment.

Debugging and tracing smart card issues requires a variety of tools and approaches. The following sections provide guidance about tools and approaches you can use.

- Certutil
- Debugging and tracing using Windows software trace preprocessor (WPP)
- Kerberos protocol, Key Distribution Center (KDC), and NTLM debugging and tracing
- Smart Card service
- Smart card readers
- CryptoAPI 2.0 Diagnostics

Certutil

For a complete description of Certutil including examples that show how to use it, see Certutil [W2012].

List certificates available on the smart card

To list certificates that are available on the smart card, type `certutil.exe -scinfo`.

> ⓘ **Note**
>
> Entering a PIN is not required for this operation. You can press ESC if you are prompted for a PIN.

Delete certificates on the smart card

Each certificate is enclosed in a container. When you delete a certificate on the smart card, you're deleting the container for the certificate.

To find the container value, type `certutil.exe -scinfo`.

To delete a container, type `certutil.exe -delkey -csp "Microsoft Base Smart Card Crypto Provider" "<ContainerValue>"`.

Debugging and tracing using WPP

WPP simplifies tracing the operation of the trace provider. It provides a mechanism for the trace provider to log real-time binary messages. Logged messages can be converted to a human-readable trace of the operation. For more information, see Diagnostics with WPP - The NDIS blog.

Enable the trace

Using WPP, use one of the following commands to enable tracing:

Windows Command Prompt

```
tracelog.exe -kd -rt -start <FriendlyName> -guid <GUID> -f .
<LogFileName*>.etl -flags <flags> -ft 1
logman.exe start <FriendlyName> -ets -p {<GUID>} -<Flags> -ft 1 -rt -o .
<LogFileName><em>.etl -mode 0x00080000</em>
```

You can use the parameters in the following table.

Expand table

Friendly name	GUID	Flags
`scardsvr`	13038e47-ffec-425d-bc69-5707708075fe	0xffff
`winscard`	3fce7c5f-fb3b-4bce-a9d8-55cc0ce1cf01	0xffff
`basecsp`	133a980d-035d-4e2d-b250-94577ad8fced	0x7
`scksp`	133a980d-035d-4e2d-b250-94577ad8fced	0x7
`msclmd`	fb36caf4-582b-4604-8841-9263574c4f2c	0x7
`credprov`	dba0e0e0-505a-4ab6-aa3f-22f6f743b480	0xffff
`certprop`	30eae751-411f-414c-988b-a8bfa8913f49	0xffff
`scfilter`	eed7f3c9-62ba-400e-a001-658869df9a91	0xffff
`wudfusbccid`	a3c09ba3-2f62-4be5-a50f-8278a646ac9d	0xffff

Examples

To enable tracing for the SCardSvr service:

Windows Command Prompt

```
tracelog.exe -kd -rt -start scardsvr -guid \#13038e47-ffec-425d-bc69-
5707708075fe -f .\scardsvr.etl -flags 0xffff -ft 1
logman.exe start scardsvr -ets -p {13038e47-ffec-425d-bc69-5707708075fe}
0xffff -ft 1 -rt -o .\scardsvr.etl -mode 0x00080000
```

To enable tracing for `scfilter.sys`:

Windows Command Prompt

```
tracelog.exe -kd -rt -start scfilter -guid \#eed7f3c9-62ba-400e-a001-
658869df9a91 -f .\scfilter.etl -flags 0xffff -ft 1
```

Stop the trace

Using WPP, use one of the following commands to stop the tracing:

Windows Command Prompt

```
tracelog.exe -stop <*FriendlyName*>
logman.exe -stop <*FriendlyName*> -ets
```

For example, to stop a trace:

Windows Command Prompt

```
tracelog.exe -stop scardsvr
logman.exe -stop scardsvr -ets
```

Kerberos protocol, KDC, and NTLM debugging and tracing

You can use these resources to troubleshoot these protocols and the KDC:

- Kerberos and LDAP Troubleshooting Tips
- Windows Driver Kit (WDK) and Debugging Tools for Windows (WinDbg). You can use the trace log tool in this SDK to debug Kerberos authentication failures.

To begin tracing, you can use `Tracelog`. Different components use different control GUIDs as explained in these examples. For more information, see Tracelog

NTLM

To enable tracing for NTLM authentication, run the following command on the command line:

Windows Command Prompt

```
tracelog.exe -kd -rt -start ntlm -guid \#5BBB6C18-AA45-49b1-A15F-
085F7ED0AA90 -f .\ntlm.etl -flags 0x15003 -ft 1
```

To stop tracing for NTLM authentication, run this command:

Windows Command Prompt

```
tracelog -stop ntlm
```

Kerberos authentication

To enable tracing for Kerberos authentication, run this command:

Windows Command Prompt

```
tracelog.exe -kd -rt -start kerb -guid \#6B510852-3583-4e2d-AFFE-
A67F9F223438 -f .\kerb.etl -flags 0x43 -ft 1
```

To stop tracing for Kerberos authentication, run this command:

Windows Command Prompt

```
tracelog.exe -stop kerb
```

KDC

To enable tracing for the KDC, run the following command on the command line:

Windows Command Prompt

```
tracelog.exe -kd -rt -start kdc -guid \#1BBA8B19-7F31-43c0-9643-6E911F79A06B
-f .\kdc.etl -flags 0x803 -ft 1
```

To stop tracing for the KDC, run the following command on the command line:

Windows Command Prompt

```
tracelog.exe -stop kdc
```

To stop tracing from a remote computer, run this command:

Windows Command Prompt

```
logman.exe -s <ComputerName>
```

> **Note**
>
> The default location for logman.exe is %systemroot%system32. Use the **-s** option to supply a computer name.

Configure tracing with the registry

You can also configure tracing by editing the Kerberos registry values shown in the following table.

Expand table

Element	Registry Key Setting
NTLM	HKEY_LOCAL_MACHINESYSTEM\CurrentControlSet\Control\Lsa\MSV1_0 Value name: NtLmInfoLevel Value type: DWORD Value data: c0015003
Kerberos	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos Value name: LogToFile Value type: DWORD Value data: 00000001 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters Value name: KerbDebugLevel Value type: DWORD Value data: c0000043 HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters Value name: LogToFile

Element	Registry Key Setting
	Value type: DWORD Value data: 00000001
KDC	HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Kdc Value name: KdcDebugLevel Value type: DWORD Value data: c0000803

If you used `Tracelog`, look for the following log file in your current directory: `kerb.etl/kdc.etl/ntlm.etl`.

If you used the registry key settings shown in the previous table, look for the trace log files in the following locations:

- NTLM: `%systemroot%\tracing\msv1_0`
- Kerberos: `%systemroot%\tracing\kerberos`
- KDC: `%systemroot%\tracing\kdcsvc`

To decode event trace files, you can use `Tracefmt` (tracefmt.exe). `Tracefmt` is a command-line tool that formats and displays trace messages from an event trace log file (.etl) or a real-time trace session. `Tracefmt` can display the messages in the Command Prompt window or save them in a text file. It is located in the \tools\tracing subdirectory of the Windows Driver Kit (WDK). For more information, see Tracefmt.

Smart Card service

The smart card resource manager service runs in the context of a local service. It's implemented as a shared service of the services host (svchost) process.

To check if Smart Card service is running:

1. Press CTRL+ALT+DEL, and then select **Start Task Manager**
2. In the **Windows Task Manager** dialog box, select the **Services** tab
3. Select the **Name** column to sort the list alphabetically, and then type **s**
4. In the **Name** column, look for **SCardSvr**, and then look under the **Status** column to see if the service is running or stopped

To restart Smart Card service:

1. Run as administrator at the command prompt
2. If the **User Account Control** dialog box appears, confirm that the action it displays is what you want, and then select **Yes**

3. At the command prompt, type `net stop SCardSvr`
4. At the command prompt, type `net start SCardSvr`

You can use the following command at the command prompt to check whether the service is running: `sc queryex scardsvr`.

The following code sample is an example output from this command:

Console

```
SERVICE_NAME: scardsvr
    TYPE        : 20 WIN32_SHARE_PROCESS
    STATE       : 4 RUNNING
                  (STOPPABLE, NOT_PAUSABLE, ACCEPTS_SHUTDOWN)
    WIN32_EXIT_CODE  : 0 (0x0)
    SERVICE_EXIT_CODE : 0 (0x0)
    CHECKPOINT     : 0x0
    WAIT_HINT     : 0x0
    PID        : 1320
    FLAGS       :
C:\>
```

Smart card readers

As with any device connected to a computer, Device Manager can be used to view properties and begin the debug process.

To check if smart card reader is working:

1. Navigate to **Computer**
2. Right-click **Computer**, and then select **Properties**
3. Under **Tasks**, select **Device Manager**
4. In Device Manager, expand **Smart card readers**, select the name of the smart card reader you want to check, and then select **Properties**

> ⓘ **Note**
>
> If the smart card reader is not listed in Device Manager, in the **Action** menu, select **Scan for hardware changes**.

CryptoAPI 2.0 Diagnostics

CryptoAPI 2.0 Diagnostics is available in Windows versions that support CryptoAPI 2.0 and can help you troubleshoot public key infrastructure (PKI) issues.

CryptoAPI 2.0 Diagnostics logs events in the Windows event log. The logs contain detailed information about certificate chain validation, certificate store operations, and signature verification. This information makes it easier to identify the causes of issues and reduces the time required for diagnosis.

For more information about CryptoAPI 2.0 Diagnostics, see Troubleshooting an Enterprise PKI.

See also

Smart Card Technical Reference

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart Card Group Policy and Registry Settings

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article for IT professionals and smart card developers describes the Group Policy settings, registry key settings, local security policy settings, and credential delegation policy settings that are available for configuring smart cards.

The following sections and tables list the smart card-related Group Policy settings and registry keys that can be set on a per-computer basis. If you use domain Group Policy Objects (GPOs), you can edit and apply Group Policy settings to local or domain computers.

- Primary Group Policy settings for smart cards
  - Allow certificates with no extended key usage certificate attribute
  - Allow ECC certificates to be used for logon and authentication
  - Allow Integrated Unblock screen to be displayed at the time of logon
  - Allow signature keys valid for Logon
  - Allow time invalid certificates
  - Allow user name hint
  - Configure root certificate clean up
  - Display string when smart card is blocked
  - Filter duplicate logon certificates
  - Force the reading of all certificates from the smart card
  - Notify user of successful smart card driver installation
  - Prevent plaintext PINs from being returned by Credential Manager
  - Reverse the subject name stored in a certificate when displaying
  - Turn on certificate propagation from smart card
  - Turn on root certificate propagation from smart card
  - Turn on Smart Card Plug and Play service
- Base CSP and Smart Card KSP registry keys
- CRL checking registry keys
- Additional smart card Group Policy settings and registry keys

Primary Group Policy settings for smart cards

The following smart card Group Policy settings are in Computer Configuration\Administrative Templates\Windows Components\Smart Card.

The registry keys are in the following locations:

- **HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\ScPnP\EnableScPnP**
- **HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\SmartCardCredentialProvider**
- **HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\CertProp**

> ⓘ Note

Smart card reader registry information is in **HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\Calais\Readers**.
Smart card registry information is in **HKEY_LOCAL_MACHINE\Software\Microsoft\Cryptography\Calais\SmartCards**.

The following table lists the default values for these GPO settings. Variations are documented under the policy descriptions in this article.

Expand table

Server type or GPO	Default value
Default Domain Policy	Not configured
Default Domain Controller Policy	Not configured
Stand-Alone Server Default Settings	Not configured
Domain Controller Effective Default Settings	Disabled
Member Server Effective Default Settings	Disabled
Client Computer Effective Default Settings	Disabled

Allow certificates with no extended key usage certificate attribute

You can use this policy setting to allow certificates without an extended key usage (EKU) set to be used for sign-in.

> **Note**
>
> extended key usage certificate attribute is also known as extended key usage.
>
> In versions of Windows before Windows Vista, smart card certificates that are used to sign in require an EKU extension with a smart card logon object identifier. This policy setting can be used to modify that restriction.

When this policy setting is turned on, certificates with the following attributes can also be used to sign in with a smart card:

- Certificates with no EKU
- Certificates with an All Purpose EKU
- Certificates with a Client Authentication EKU

When this policy setting isn't turned on, only certificates that contain the smart card logon object identifier can be used to sign in with a smart card.

Expand table

Item	Description
Registry key	AllowCertificatesWithNoEKU

Item	Description
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Allow ECC certificates to be used for logon and authentication

You can use this policy setting to control whether elliptic curve cryptography (ECC) certificates on a smart card can be used to sign in to a domain.

When this setting is turned on, ECC certificates on a smart card can be used to sign in to a domain.

When this setting isn't turned on, ECC certificates on a smart card can't be used to sign in to a domain.

Expand table

Item	Description
Registry key	`EnumerateECCCerts`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None
Notes and resources	This policy setting only affects a user's ability to sign in to a domain. ECC certificates on a smart card that are used for other applications, such as document signing, aren't affected by this policy setting. If you use an ECDSA key to sign in, you must also have an associated ECDH key to permit sign in when you're not connected to the network.

Allow Integrated Unblock screen to be displayed at the time of logon

You can use this policy setting to determine whether the integrated unblock feature is available in the sign-in user interface (UI). The feature was introduced as a standard feature in the Credential Security Support Provider in Windows Vista.

When this setting is turned on, the integrated unblock feature is available.

When this setting isn't turned on, the feature is not available.

Expand table

Item	Description
Registry key	`AllowIntegratedUnblock`
Default values	No changes per operating system versions Disabled and not configured are equivalent

Item	Description
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None
Notes and resources	To use the integrated unblock feature, the smart card must support it. Check with the hardware manufacturer to verify that the smart card supports this feature. You can create a custom message that the user sees when the smart card is blocked by configuring the policy setting Display string when smart card is blocked.

Allow signature keys valid for Logon

You can use this policy setting to allow signature key-based certificates to be enumerated and available for sign-in.

When this setting is turned on, any certificates that are available on the smart card with a signature-only key are listed on the sign-in screen.

When this setting isn't turned on, certificates available on the smart card with a signature-only key aren't listed on the sign-in screen.

Expand table

Item	Description
Registry key	**AllowSignatureOnlyKeys**
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Allow time invalid certificates

You can use this policy setting to permit certificates that are expired or not yet valid to be displayed for sign-in.

> **Note**
>
> Before Windows Vista, certificates were required to contain a valid time and to not expire. For a certificate to be used, it must be accepted by the domain controller. This policy setting only controls which certificates are displayed on the client computer.

When this setting is turned on, certificates are listed on the sign-in screen whether they have an invalid time, or their time validity has expired.

When this policy setting isn't turned on, certificates that are expired or not yet valid aren't listed on the sign-in screen.

Expand table

Item	Description
Registry key	`AllowTimeInvalidCertificates`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Allow user name hint

You can use this policy setting to determine whether an optional field appears during sign-in and provides a subsequent elevation process where users can enter their username or username and domain, which associates a certificate with the user.

When this policy setting is turned on, users see an optional field where they can enter their username or username and domain.

When this policy setting isn't turned on, users don't see this optional field.

Expand table

Item	Description
Registry key	`X509HintsNeeded`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Configure root certificate clean-up

You can use this policy setting to manage the cleanup behavior of root certificates. Certificates are verified by using a trust chain, and the trust anchor for the digital certificate is the Root Certification Authority (CA). A CA can issue multiple certificates with the root certificate as the top certificate of the tree structure. A private key is used to sign other certificates. This creates an inherited trustworthiness for all certificates immediately under the root certificate.

When this policy setting is turned on, you can set the following cleanup options:

- **No cleanup**. When the user signs out or removes the smart card, the root certificates used during their session persist on the computer.
- **Clean up certificates on smart card removal**. When the smart card is removed, the root certificates are removed.
- **Clean up certificates on log off**. When the user signs out of Windows, the root certificates are removed.

When this policy setting isn't turned on, root certificates are automatically removed when the user signs out of Windows.

Expand table

Item	Description
Registry key	`RootCertificateCleanupOption`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Display string when smart card is blocked

You can use this policy setting to change the default message that a user sees if their smart card is blocked.

When this policy setting is turned on, you can create and manage the displayed message that the user sees when a smart card is blocked.

When this policy setting isn't turned on (and the integrated unblock feature is also enabled), the user sees the system's default message when the smart card is blocked.

Expand table

Item	Description
Registry key	`IntegratedUnblockPromptString`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: This policy setting is only effective when the Allow Integrated Unblock screen to be displayed at the time of logon policy is enabled.

Filter duplicate logon certificates

You can use this policy setting to configure which valid sign-in certificates are displayed.

> **Note**
>
> During the certificate renewal period, a user's smart card can have multiple valid sign-in certificates issued from the same certificate template, which can cause confusion about which certificate to select. This behavior can occur when a certificate is renewed and the old certificate has not expired yet.
>
> If two certificates are issued from the same template with the same major version and they are for the same user (this is determined by their UPN), they are determined to be the same.

When this policy setting is turned on, filtering occurs so that the user can select from only the most current valid certificates.

If this policy setting isn't turned on, all the certificates are displayed to the user.

This policy setting is applied to the computer after the Allow time invalid certificates policy setting is applied.

Expand table

Item	Description
Registry key	`FilterDuplicateCerts`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None
Notes and resources	If there are two or more of the same certificates on a smart card and this policy setting is enabled, the certificate with the most distant expiration time is displayed.

Force the reading of all certificates from the smart card

You can use this policy setting to manage how Windows reads all certificates from the smart card for sign-in. During sign-in, Windows reads only the default certificate from the smart card unless it supports retrieval of all certificates in a single call. This policy setting forces Windows to read all the certificates from the smart card.

When this policy setting is turned on, Windows attempts to read all certificates from the smart card, regardless of the CSP feature set.

When this policy isn't turned on, Windows attempts to read only the default certificate from smart cards that don't support retrieval of all certificates in a single call. Certificates other than the default aren't available for sign-in.

Expand table

Item	Description
Registry key	`ForceReadingAllCertificates`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None **Important**: Enabling this policy setting can adversely impact performance during the sign-in process in certain situations.
Notes and resources	Contact the smart card vendor to determine if your smart card and associated CSP support the required behavior.

Notify user of successful smart card driver installation

You can use this policy setting to control whether the user sees a confirmation message when a smart card device driver is installed.

When this policy setting is turned on, the user sees a confirmation message when a smart card device driver is installed.

When this setting isn't turned on, the user doesn't see a smart card device driver installation message.

Expand table

--	--
Registry key	`ScPnPNotification`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None
Notes and resources	This policy setting applies only to smart card drivers that have passed the Windows Hardware Quality Labs (WHQL) testing process.

Prevent plaintext PINs from being returned by Credential Manager

You can use this policy setting to prevent Credential Manager from returning plaintext PINs.

> **Note**
>
> Credential Manager is controlled by the user on the local computer, and it stores credentials from supported browsers and Windows applications. Credentials are saved in special encrypted folders on the computer under the user's profile.

When this policy setting is turned on, Credential Manager doesn't return a plaintext PIN.

When this setting isn't turned on, Credential Manager can return plaintext PINs.

Expand table

Item	Description
Registry key	`DisallowPlaintextPin`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Item	Description
Notes and resources	If this policy setting is enabled, some smart cards might not work in computers running Windows. Consult the smart card manufacturer to determine whether this policy setting should be enabled.

Reverse the subject name stored in a certificate when displaying

You can use this policy setting to control the way the subject name appears during sign-in.

> **Note**
>
> To help users distinguish one certificate from another, the user principal name (UPN) and the common name are displayed by default. For example, when this setting is enabled, if the certificate subject is *CN=User1, OU=Users, DN=example, DN=com* and the UPN is *user1@example.com*, *User1* is displayed with *user1@example.com*. If the UPN is not present, the entire subject name is displayed. This setting controls the appearance of that subject name, and it might need to be adjusted for your organization.

When this policy setting is turned on, the subject name during sign-in appears reversed from the way that it's stored in the certificate.

When this policy setting isn't turned on, the subject name appears the same as it's stored in the certificate.

Expand table

Item	Description
Registry key	`ReverseSubject`
Default values	No changes per operating system versions Disabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None

Turn on certificate propagation from smart card

You can use this policy setting to manage the certificate propagation that occurs when a smart card is inserted.

> **Note**
>
> The certificate propagation service applies when a signed-in user inserts a smart card in a reader that is attached to the computer. This action causes the certificate to be read from the smart card. The certificates are then added to the user's Personal store.

When this policy setting is turned on, certificate propagation occurs when the user inserts the smart card.

When this policy setting is turned off, certificate propagation doesn't occur, and the certificates aren't available to applications, like Outlook.

Expand table

Item	Description
Registry key	`CertPropEnabled`
Default values	No changes per operating system versions Enabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: This policy setting must be enabled to allow the Turn on root certificate propagation from smart card setting to work when it is enabled.

Turn on root certificate propagation from smart card

You can use this policy setting to manage the root certificate propagation that occurs when a smart card is inserted.

> **Note**
>
> The certificate propagation service applies when a signed-in user inserts a smart card in a reader that is attached to the computer. This action causes the certificate to be read from the smart card. The certificates are then added to the user's Personal store.

When this policy setting is turned on, root certificate propagation occurs when the user inserts the smart card.

When this policy setting isn't turned on, root certificate propagation doesn't occur when the user inserts the smart card.

Expand table

Item	Description
Registry key	`EnableRootCertificate Propagation`
Default values	No changes per operating system versions Enabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: For this policy setting to work, the Turn on certificate propagation from smart card policy setting must also be enabled.
Notes and resources	

Turn on Smart Card Plug and Play service

You can use this policy setting to control whether Smart Card Plug and Play is enabled.

> **Note**

Your users can use smart cards from vendors who have published their drivers through Windows Update without needing special middleware. These drivers will be downloaded in the same way as drivers for other devices in Windows. If an appropriate driver isn't available from Windows Update, a PIV-compliant mini driver that's included with any of the supported versions of Windows is used for these cards.

When this policy setting is turned on, the system attempts to install a smart card device driver the first time a smart card is inserted in a smart card reader.

When this policy setting isn't turned on, a device driver isn't installed when a smart card is inserted in a smart card reader.

Expand table

Item	Description
Registry key	`EnableScPnP`
Default values	No changes per operating system versions Enabled and not configured are equivalent
Policy management	Restart requirement: None Sign off requirement: None Policy conflicts: None
Notes and resources	This policy setting applies only to smart card drivers that have passed the Windows Hardware Quality Labs (WHQL) testing process.

Base CSP and Smart Card KSP registry keys

The following registry keys can be configured for the base cryptography service provider (CSP) and the smart card key storage provider (KSP). The following tables list the keys. All keys use the DWORD type.

The registry keys for the Base CSP are in the registry in `HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Cryptography\Defaults\Provider\Microsoft Base Smart Card Crypto Provider`.

The registry keys for the smart card KSP are in `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Cryptography\Providers\Microsoft Smart Card Key Storage Provider`.

Registry keys for the base CSP and smart card KSP

Expand table

Registry Key	Description
AllowPrivateExchangeKeyImport	A non-zero value allows RSA exchange (for example, encryption) private keys to be imported for use in key archival scenarios. Default value: 00000000
AllowPrivateSignatureKeyImport	A non-zero value allows RSA signature private keys to be imported for use in key archival scenarios.

Registry Key	Description
	Default value: 00000000
DefaultPrivateKeyLenBits	Defines the default length for private keys, if desired. Default value: 00000400 Default key generation parameter: 1024-bit keys
RequireOnCardPrivateKeyGen	This key sets the flag that requires on-card private key generation (default). If this value is set, a key generated on a host can be imported into the smart card. This is used for smart cards that don't support on-card key generation or where key escrow is required. Default value: 00000000
TransactionTimeoutMilliseconds	Default timeout values allow you to specify whether transactions that take an excessive amount of time will fail. Default value: 000005dc The default timeout for holding transactions to the smart card is 1.5 seconds.

Additional registry keys for the smart card KSP:

Expand table

Registry Key	Description
AllowPrivateECDHEKeyImport	This value allows Ephemeral Elliptic Curve Diffie-Hellman (ECDHE) private keys to be imported for use in key archival scenarios. Default value: 00000000
AllowPrivateECDSAKeyImport	This value allows Elliptic Curve Digital Signature Algorithm (ECDSA) private keys to be imported for use in key archival scenarios. Default value: 00000000

CRL checking registry keys

The following table lists the keys and the corresponding values to turn off certificate revocation list (CRL) checking at the Key Distribution Center (KDC) or client. To manage CRL checking, you must configure settings for both the KDC and the client.

Expand table

Registry Key	Details
`HKEY_LOCAL_MACHINE\SYSTEM\CCS\Services\Kdc\UseCachedCRLOnlyAndIgnoreRevocationUnknownErrors`	Type = DWORD Value = 1
`HKEY_LOCAL_MACHINE\SYSTEM\CCS\Control\LSA\Kerberos\Parameters\UseCachedCRLOnlyAndIgnoreRevocationUnknownErrors`	Type = DWORD Value = 1

Additional smart card Group Policy settings and registry keys

In a smart card deployment, additional Group Policy settings can be used to enhance ease-of-use or security. Two of these policy settings that can complement a smart card deployment are:

- Turning off delegation for computers
- Interactive logon: Do not require CTRL+ALT+DEL (not recommended)

The following smart card-related Group Policy settings are in **Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options**.

Local security policy settings

Expand table

Group Policy setting and registry key	Default	Description
Interactive logon: Require smart card **scforceoption**	Disabled	This security policy setting requires users to sign in to a computer by using a smart card. **Enabled** Users can sign in to the computer only by using a smart card. **Disabled** Users can sign in to the computer by using any method. NOTE: the Windows LAPS-managed local account is exempted from this policy when Enabled.
Interactive logon: Smart card removal behavior **scremoveoption**	This policy setting isn't defined, which means that the system treats it as **No Action**.	This setting determines what happens when the smart card for a signed-in user is removed from the smart card reader. The options are: **No Action** **Lock Workstation**: The workstation is locked when the smart card is removed, so users can leave the area, take their smart card with them, and still maintain a protected session. **Force Logoff**: The user is automatically signed out when the smart card is removed. **Disconnect if a Remote Desktop Services session**: Removal of the smart card disconnects the session without signing out the user. The user can reinsert the smart card and resume the session later, or at another computer that's equipped with a smart card reader, without having to sign in again. If the session is local, this policy setting functions identically to the **Lock Workstation** option.

From the Local Security Policy Editor (secpol.msc), you can edit and apply system policies to manage credential delegation for local or domain computers.

The following smart card-related Group Policy settings are in **Computer Configuration\Administrative Templates\System\Credentials Delegation**.

Registry keys are in `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Lsa\Credssp\PolicyDefaults`.

> ⓘ Note

In the following table, fresh credentials are those that you are prompted for when running an application.

Credential delegation policy settings

Expand table

Group Policy setting and registry key	Default	Description
Allow Delegating Fresh Credentials **AllowFreshCredentials**	Not configured	This policy setting applies: When server authentication was achieved through a trusted X509 certificate or Kerberos protocol. To applications that use the CredSSP component (for example, Remote Desktop Services). **Enabled**: You can specify the servers where the user's fresh credentials can be delegated. **Not configured**: After proper mutual authentication, delegation of fresh credentials is permitted to Remote Desktop Services running on any computer. **Disabled**: Delegation of fresh credentials to any computer isn't permitted. **Note**: This policy setting can be set to one or more service principal names (SPNs). The SPN represents the target server where the user credentials can be delegated. A single wildcard character is permitted when specifying the SPN, for example: Use *TERMSRV/** for Remote Desktop Session Host (RD Session Host) running on any computer. Use *TERMSRV/host.humanresources.fabrikam.com* for RD Session Host running on the host.humanresources.fabrikam.com computer. Use *TERMSRV/*.humanresources.fabrikam.com* for RD Session Host running on all computers in .humanresources.fabrikam.com
Allow Delegating Fresh Credentials with NTLM-only Server Authentication **AllowFreshCredentialsWhenNTLMOnly**	Not configured	This policy setting applies: When server authentication was achieved by using NTLM. To applications that use the CredSSP component (for example, Remote Desktop). **Enabled**: You can specify the servers where the user's fresh credentials can be delegated. **Not configured**: After proper mutual authentication, delegation of fresh credentials is permitted to RD Session Host running on any computer (TERMSRV/*). **Disabled**: Delegation of fresh credentials isn't permitted to any computer. **Note**: This policy setting can be set to one or more SPNs. The SPN represents the target server where the user credentials can be delegated. A single wildcard character (*) is permitted when specifying the SPN. See the **Allow Delegating Fresh Credentials** policy setting description for examples.

Group Policy setting and registry key	Default	Description
Deny Delegating Fresh Credentials **DenyFreshCredentials**	Not configured	This policy setting applies to applications that use the CredSSP component (for example, Remote Desktop). **Enabled**: You can specify the servers where the user's fresh credentials can't be delegated. **Disabled** or **Not configured**: A server is not specified. **Note**: This policy setting can be set to one or more SPNs. The SPN represents the target server where the user credentials can't be delegated. A single wildcard character (*) is permitted when specifying the SPN. For examples, see the "Allow delegating fresh credentials" policy setting.

If you're using Remote Desktop Services with smart card logon, you can't delegate default and saved credentials. The registry keys in the following table, which are at `HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Lsa\Credssp\PolicyDefaults`, and the corresponding Group Policy settings are ignored.

Expand table

Registry Key	Corresponding Group Policy setting
AllowDefaultCredentials	Allow Delegating Default Credentials
AllowDefaultCredentialsWhenNTLMOnly	Allow Delegating Default Credentials with NTLM-only Server Authentication
AllowSavedCredentials	Allow Delegating Saved Credentials
AllowSavedCredentialsWhenNTLMOnly	Allow Delegating Saved Credentials with NTLM-only Server Authentication

See also

Smart Card Technical Reference

Feedback

Was this page helpful? Yes No

Provide product feedback

Smart card events

Article • 10/29/2024 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the events related to smart card deployment and development.

Many events can be used to monitor smart card activities on a device, including installation, use, and errors. The next sections describe the events and information that you can use to manage smart cards in an organization.

Smart card reader name

The Smart Card Resource Manager doesn't use the device name from *Device Manager* to describe a smart card reader. Instead, the name is constructed from three device attributes that are queried directly from the smart card reader driver.

The following three attributes are used to construct the smart card reader name:

- Vendor name
- Interface device type
- Device unit

The smart card reader device name is constructed in the form `<VendorName><Type><DeviceUnit>`. For example *Contoso Smart Card Reader 0* is constructed from the following information:

- Vendor name: *Contoso*
- Interface device type: *Smart Card Reader*
- Device unit: *0*

Smart card warning events

> **Note**
>
> *IOCTL* in the following table refers to input and output control.

Expand table

Event ID	Warning Message	Description
620	Smart Card Resource Manager was unable to cancel IOCTL %3 for reader '%2': %1. The reader may no longer be responding. If this error persists, your smart card or reader may not be functioning correctly. %n%nCommand Header: %4	This occurs if the Resource Manager attempts to cancel a command to the smart card reader when the smart card service is shutting down or after a smart card is removed from the smart card reader and the command couldn't be canceled. This can leave the smart card reader in an unusable state until it's removed from the computer or the computer is restarted. %1 = Windows error code %2 = Smart card reader name %3 = IOCTL being canceled %4 = First 4 bytes of the command that was sent to the smart card
619	Smart Card Reader '%2' hasn't responded to IOCTL %3 in %1 seconds. If this error persists, your smart card or reader may not be functioning correctly. %n%nCommand Header: %4	This occurs when a reader hasn't responded to an IOCTL after an unusually long period of time. Currently, this error is sent after a reader doesn't respond for 150 seconds. This can leave the smart card reader in an unusable state until it's removed from the computer or the computer is restarted. %1 = Number of seconds the IOCTL has been waiting %2 = Smart card reader name %3 = IOCTL sent %4 = First 4 bytes of the command that was sent to the smart card

Smart card error events

Expand table

Event ID	Error Message	Description
202	Failed to initialize Server Application	An error occurred, and the service can't initialize properly. Restarting the computer may resolve the issue.
203	Server Control has no memory for reader reference object.	Internal, unrecoverable error that indicates a failure in the smart card service. The most

Event ID	Error Message	Description
		common cause is limited computer resources. Restarting the computer may resolve the issue.
204	Server Control failed to create shutdown event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
205	Reader object has duplicate name: %1	There are two smart card readers that have the same name. Remove the smart card reader that is causing this error message. %1 = Name of the smart card reader that is duplicated
206	Failed to create global reader change event.	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue.
401	Reader shutdown exception from eject smart card command	A smart card reader couldn't eject a smart card while the smart card reader was shutting down.
406	Reader object can't Identify Device	A smart card reader didn't properly respond to a request for information about the device, which is required for constructing the smart card reader name. The smart card reader won't be recognized by the service until it's removed from the computer and reinserted or until the computer is restarted.
502	Initialization of Service Status Critical Section failed	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue.
504	Resource Manager can't create shutdown event flag: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
506	Smart Card Resource Manager failed to register service: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code

Event ID	Error Message	Description
506	Smart Card Resource Manager received unexpected exception from PnP event %1	An attempt to add a Plug and Play reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
507	No memory available for Service Status Critical Section	There isn't enough system memory available. This prevents the service from managing the status. Restarting the computer may resolve the issue.
508	Smart Card Resource Manager received unexpected exception from PnP event %1	An attempt to add a Plug and Play reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
509	Smart Card Resource Manager received unexpected exception from PnP event %1	An attempt to add a Plug and Play reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
510	Smart Card Resource Manager received NULL handle from PnP event %1	An attempt to add a Plug and Play smart card reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
511	Smart Card Resource Manager received unexpected exception from PnP event %1	An attempt to add a Plug and Play reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
512	Smart Card Resource Manager received NULL handle from PnP event %1	An attempt to add a Plug and Play smart card reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name

Event ID	Error Message	Description
513	Smart Card Resource Manager received unexpected exception from PnP event %1	An attempt to add a Plug and Play reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
514	Smart Card Resource Manager failed to add reader %2: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code %2 = Smart card reader name
515	Smart Card Resource Manager failed to declare state: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The smart card service may not operate properly. Restarting the service or computer may resolve this issue. %1 = Windows error code
516	Smart Card Resource Manager Failed to declare shutdown: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The smart card service may not be able to stop. Restarting the computer may resolve this issue. %1 = Windows error code
517	Smart Card Resource Manager received unexpected exception attempting to add reader %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Smart card reader name
521	Smart Card Resource Manager received NULL handle from PnP event %1	An attempt to add a Plug and Play smart card reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name
523	Smart Card Resource Manager received NULL handle from PnP event %1	An attempt to add a Plug and Play smart card reader failed. The device may already be in use or may be defective. To resolve this error message, try to add the device again or restart the computer. %1 = The affected handle name

Event ID	Error Message	Description
602	WDM Reader driver initialization can't open reader device: %1	The service can't open a communication channel with the smart card reader. You can't use the smart card reader until the issue is resolved. %1 = Windows error code
603	WDM Reader driver initialization has no memory available to control device %1	There isn't enough system memory available. This prevents the service from managing the smart card reader that was added. Restarting the computer may resolve the issue. %1 = Name of affected reader
604	Server control can't set reader removal event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
605	Reader object failed to create overlapped event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
606	Reader object failed to create removal event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
607	Reader object failed to start monitor thread: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
608	Reader monitor failed to create power down timer: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code
609	Reader monitor failed to create overlapped event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code

Event ID	Error Message	Description
610	Smart Card Reader '%2' rejected IOCTL %3: %1 If this error persists, your smart card or reader may not be functioning correctly.%n%nCommand Header: %4	The reader can't successfully transmit the indicated IOCTL to the smart card. This can indicate hardware failure, but this error can also occur if a smart card or smart card reader is removed from the system while an operation is in progress. %1 = Windows error code %2 = Name of the smart card reader %3 = IOCTL that was sent %4 = First 4 bytes of the command sent to the smart card These events are caused by legacy functionality in the smart card stack. It can be ignored if there's no noticeable failure in the smart card usage scenarios. You might also see this error if your eSIM is recognized as a smartcard controller.
611	Smart Card Reader initialization failed	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve this issue.
612	Reader insertion monitor error retry threshold reached: %1	This occurs when a smart card reader fails several times to respond properly to the IOCTL, which indicates whether a smart card is present in the reader. The smart card reader is marked as defective, and it isn't recognized by the service until it's removed from the computer and reinserted or until the computer is restarted. %1 = Windows error code
615	Reader removal monitor error retry threshold reached: %1	This occurs when a smart card reader fails several times to respond properly to the IOCTL, which indicates whether a smart card is present in the reader. The smart card reader is marked as defective, and it isn't recognized by the service until it's removed from the computer and reinserted or until the computer is restarted. %1 = Windows error code
616	Reader monitor '%2' received uncaught error code: %1	This occurs when a smart card reader fails several times to respond properly to the IOCTL, which indicates whether a smart card is present in the reader. The smart card reader is marked

Event ID	Error Message	Description
		as defective, and it isn't recognized by the service until it's removed from the computer and reinserted or until the computer is restarted. %1 = Windows error code %2 = Reader name
617	Reader monitor '%1' exception -- exiting thread	An unknown error occurred while monitoring a smart card reader for smart card insertions and removals. The smart card reader is marked as defective, and it isn't recognized by the service until it's removed from the computer and reinserted or until the computer is restarted. %1 = Smart card reader name
618	Smart Card Resource Manager encountered an unrecoverable internal error.	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue.
621	Server Control failed to access start event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code These events are caused by legacy functionality in the smart card stack. It can be ignored if there's no noticeable failure in the smart card usage scenarios.
622	Server Control failed to access stop event: %1	Internal, unrecoverable error that indicates a failure in the smart card service. The most common cause is limited computer resources. Restarting the computer may resolve the issue. %1 = Windows error code

Smart card Plug and Play events

Expand table

Event ID	Event type	Event Message	Description
1000	Error	Couldn't get device ID for smart card in reader %1.	Smart card Plug and Play couldn't obtain the device ID for the smart card. This

Event ID	Event type	Event Message	Description
		The return code is %2.	information is required to determine the correct driver. The smart card may be defective. %1 = Smart card reader name %2 = Windows error code
1001	Information	Software successfully installed for smart card in reader %1. The smart card name is %2.	Smart card Plug and Play successfully installed a minidriver for the inserted card. %1 = Smart card reader name %2 = Name of new smart card device

See also

Smart Card Technical Reference

Feedback

Was this page helpful? Yes No

Provide product feedback

Virtual Smart Card Overview

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> ⚠ **Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

This article provides an overview of the virtual smart card technology.

Feature description

Virtual smart card technology offers comparable security benefits to physical smart cards by using two-factor authentication. Virtual smart cards emulate the functionality of physical smart cards, but they use the Trusted Platform Module (TPM) chip that is available on devices. Virtual smart cards don't require the use of a separate physical smart card and reader. You create virtual smart cards in the TPM, where the keys used for authentication are stored in cryptographically-secured hardware.

By utilizing TPM devices that provide the same cryptographic capabilities as physical smart cards, virtual smart cards accomplish the three key properties that are desired for smart cards: non-exportability, isolated cryptography, and anti-hammering.

Practical applications

Virtual smart cards are functionally similar to physical smart cards, appearing in Windows as smart cards that are always-inserted. Virtual smart cards can be used for authentication to external resources, protection of data by encryption, and integrity through signing. You can deploy virtual smart cards by using in-house methods or a purchased solution, and they can be a replacement for other methods of strong authentication in a corporate setting of any scale.

Authentication use cases

Two-factor authentication–based remote access

After a user has a fully functional TPM virtual smart card, provisioned with a sign-in certificate, the certificate is used to gain authenticated access to corporate resources. When the proper certificate is provisioned to the virtual card, the user need only provide the PIN for the virtual smart card, as if it was a physical smart card, to sign in to the domain.

In practice, this is as easy as entering a password to access the system. Technically, it's far more secure. Using the virtual smart card to access the system proves to the domain that the user who is requesting authentication has possession of the personal computer upon which the card has been provisioned and knows the virtual smart card PIN. Because this request couldn't have possibly originated from a system other than the system certified by the domain for this user's access, and the user couldn't have initiated the request without knowing the PIN, a strong two-factor authentication is established.

Client authentication

Virtual smart cards can also be used for client authentication by using TLS/SSL or a similar technology. Similar to domain access with a virtual smart card, an authentication certificate can be provisioned for the virtual smart card, provided to a remote service, as requested in the client authentication process. This adheres to the principles of two-factor authentication because the certificate is only accessible from the computer that hosts the virtual smart card, and the user is required to enter the PIN for initial access to the card.

Virtual smart card redirection for remote desktop connections

The concept of two-factor authentication associated with virtual smart cards relies on the proximity of users to the devices that they use to access domain. When you connect to a device that is hosting virtual smart cards, you can't use the virtual smart cards located on the remote device during the remote session. However, you can access the virtual smart cards on the connecting device (which is under your physical control), which are loaded onto the remote device. You can use the virtual smart cards as if they were installed by using the remote devices' TPM, extending your privileges to the remote device, while maintaining the principles of two-factor authentication.

Confidentiality use cases

S/MIME email encryption

Physical smart cards are designed to hold private keys. You can use the private keys for email encryption and decryption. The same functionality exists in virtual smart cards. By using S/MIME with a user's public key to encrypt email, the sender of an email is assured that only the person with the corresponding private key can decrypt the email. This assurance is a result of the non-exportability of the private key. It never exists within reach of malicious software, and it remains protected by the TPM—even during decryption.

BitLocker for data volumes

BitLocker Drive Encryption technology makes use of symmetric-key encryption to protect the content of a user's hard drive. BitLocker ensures that if the physical ownership of a hard drive is compromised, an adversary won't be able to read data off the drive. The key used to encrypt the drive can be stored in a virtual smart card, which necessitates knowledge of the virtual smart card PIN to access the drive, and possession of the device that is hosting the TPM virtual smart card. If the drive is obtained without access to the TPM that hosts the virtual smart card, any brute force attack will be difficult.

You can use BitLocker to encrypt portable drives, storing keys in virtual smart cards. In this scenario, unlike using BitLocker with a physical smart card, the encrypted drive can be used only when it's connected to device for the virtual smart card that is used to encrypt the drive, because the BitLocker key is only accessible from the device. This method can be useful to ensure the security of backup drives and personal storage uses outside the main hard drive, too.

Data integrity use case

Signing data

To verify authorship of data, a user can sign it by using a private key stored in the virtual smart card. Digital signatures confirm the integrity and origin of the data.

- Storing the key in an operating system that is accessible, malicious users could access it and use it to modify already signed data or to spoof the key owner's identity
- Storing the key in a virtual smart card, means that you can only use it to sign data on the host device. You can't export the key to other systems (intentionally or unintentionally, such as with malware theft), making digital signatures more secure than other methods for private key storage

Hardware requirements

To use the virtual smart card technology, TPM 1.2 is the minimum required for devices running a supported operating system.

Feedback

Was this page helpful? Yes No

Provide product feedback

Understand and Evaluate Virtual Smart Cards

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **⚠ Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

This article describes the virtual smart card technology and how it can fit into your authentication design.

Virtual smart card technology uses cryptographic keys that are stored on computers that have the Trusted Platform Module (TPM) installed. Virtual smart cards offer comparable security benefits to conventional smart cards by using two-factor authentication. The technology also offers more convenience for users and has a lower cost to deploy. By utilizing TPM devices that provide the same cryptographic capabilities as conventional smart cards, virtual smart cards accomplish the three key properties that are desired for smart cards: nonexportability, isolated cryptography, and anti-hammering.

Virtual smart cards are functionally similar to physical smart cards. They appear as always-inserted smart cards, and they can be used for authentication to external resources, protection of data by secure encryption, and integrity through reliable signing. Since TPM-enabled hardware is readily available and virtual smart cards can be deployed using existing certificate enrollment methods, virtual smart cards can become a replacement for other methods of strong authentication in a corporate setting of any scale.

This topic contains the following sections:

- Comparing virtual smart cards with physical smart cards: compares properties, functional aspects, security, and cost.
- Authentication design options: describes how passwords, smart cards, and virtual smart cards can be used to reach authentication goals in your organization.

Comparing virtual smart cards with physical smart cards

Virtual smart cards function much like physical smart cards, but they differ in that they protect private keys by using the TPM of the computer instead of smart card media.

A virtual smart card appears to applications as a conventional smart card. Private keys in the virtual smart card are protected, not by isolation of physical memory, but by the cryptographic capabilities of the TPM. All sensitive information is encrypted by using the TPM and then stored on the hard drive in its encrypted form.

All cryptographic operations occur in the secure, isolated environment of the TPM, and the unencrypted private keys are never used outside this environment. So like physical smart cards, virtual smart cards remain secure from any malware on the host. Additionally, if the hard drive is compromised in some way, a malicious user won't be able to access keys that are stored in the virtual smart card because they're securely encrypted by using the TPM. Keys can also be protected by BitLocker Drive Encryption.

Virtual smart cards maintain the three key properties of physical smart cards:

- **Non-exportability**: Because all private information on the virtual smart card is encrypted by using the TPM on the host computer, it can't be used on a different computer with a different TPM. Additionally, TPMs are designed to be tamper-resistant and nonexportable, so a malicious user can't reverse engineer an identical TPM or install the same TPM on a different computer. For more information, see Evaluate Virtual Smart Card Security
- **Isolated cryptography**: TPMs provide the same properties of isolated cryptography that are offered by physical smart cards, and this is utilized by virtual smart cards. Unencrypted copies of private keys are loaded only within the TPM and never into memory that is accessible by the operating system. All cryptographic operations with these private keys occur inside the TPM
- **Anti-hammering**: If a user enters a PIN incorrectly, the virtual smart card responds by using the anti-hammering logic of the TPM, which rejects further attempts for a period of time instead of blocking the card. This is also known as lockout. For more information, see Evaluate Virtual Smart Card Security

The following subsections compare the functionality, security, and cost of virtual smart cards and physical smart cards.

Functionality

The virtual smart card system that was designed by Microsoft closely mimics the functionality of conventional smart cards. The most striking difference to the end user is that the virtual smart card is essentially a smart card that is always inserted into the computer. There's no method to export the user's virtual smart card for use on other computers, which adds to the security of virtual smart cards. If a user requires access to network resources on multiple computers, multiple virtual smart cards can be issued for that user. Additionally, a computer that is shared among multiple users can host multiple virtual smart cards for different users.

The basic user experience for a virtual smart card is as simple as using a password to access a network. Because the smart card is loaded by default, the user must enter the PIN that is tied to the card to gain access. Users are no longer required to carry cards and readers or to take physical action to use the card.

Additionally, although the anti-hammering functionality of the virtual smart card is equally secure to that of a physical smart card, virtual smart card users are never required to contact an administrator to unblock the card. Instead, they wait a period of time (depending on the TPM specifications) before they reattempt to enter the PIN. Alternatively, the administrator can reset the lockout by providing owner authentication data to the TPM on the host computer.

Security

Physical smart cards and virtual smart cards offer comparable levels of security. They both implement two-factor authentication for using network resources. However, they differ in certain aspects, including physical security and the practicality of an attack. Due to their compact and portable design, conventional smart cards are most frequently kept close to their intended user. They offer little opportunity for acquisition by a potential adversary, so any sort of interaction with the card is difficult without committing some variety of theft.

TPM virtual smart cards, however, reside on a user's computer that may frequently be left unattended, which provides an opportunity for a malicious user to hammer the TPM. Although virtual smart cards are fully protected from hammering (as are physical smart cards), this accessibility makes the logistics of an attack simpler. Additionally, the anti-hammering behavior of a TPM smart card differs in that it only presents a time delay in response to repeated PIN failures, as opposed to fully blocking the user.

However, there are several advantages provided by virtual smart cards to mitigate these slight security deficits. Most importantly, a virtual smart card is much less likely to be lost. Virtual smart cards are integrated into computers and devices that the user already owns for other purposes and has incentive to keep safe. If the computer or device that

hosts the virtual smart card is lost or stolen, a user notices its loss quicker than the loss of a physical smart card. When a computer or device is identified as lost, the user can notify the administrator of the system, who can revoke the certificate that is associated with the virtual smart card on that device. This precludes any future unauthorized access on that computer or device if the PIN for the virtual smart card is compromised.

Cost

If a company wants to deploy physical smart cards, they need to purchase smart cards and smart card readers for all employees. Although relatively inexpensive options can be found, options that ensure the three key properties of smart card security (most notably, nonexportability) are more expensive. If employees have computers with a built-in TPM, virtual smart cards can be deployed with no additional material costs. These computers and devices are relatively common in the market.

The maintenance cost of virtual smart cards is less than that for physical smart cards, which are easily lost, stolen, or broken from normal wear. TPM virtual smart cards are only lost or broken if the host computer or device is lost or broken, which in most cases is much less frequently.

Comparison summary

Expand table

Physical Smart Cards	TPM virtual smart cards
Protects private keys by using the built-in cryptographic functionality of the card.	Protects private keys by using the cryptographic functionality of the TPM.
Stores private keys in isolated non-volatile memory on the card, which means that access to private keys is only from the card, and access is never allowed to the operating system.	Stores encrypted private keys on the hard drive. The encryption ensures that these keys can only be decrypted and used in the TPM, not in the accessible memory of the operating system.
Guarantees nonexportability through the card manufacturer, which includes isolating private information from operating system access.	Guarantees nonexportability through the TPM manufacturer, which includes the inability of an adversary to replicate or remove the TPM.
Performs and isolates cryptographic operations within the built-in capabilities of the card.	Performs and isolates cryptographic operations in the TPM of the user's computer or device.

Physical Smart Cards	TPM virtual smart cards
Provides anti-hammering through the card. After some failed PIN entry attempts, the card blocks further access until administrative action is taken.	Provides anti-hammering through the TPM. Successive failed attempts increase the device lockout time (the time the user has to wait before trying again). This can be reset by an administrator.
Requires that users carry their smart card and smart card reader with them to access network resources.	Allows users to access their TPM-enabled computers or devices, and potentially access the network, without other equipment.
Enables credential portability by inserting the smart card into smart card readers that are attached to other computers.	Prevents exporting credentials from a given computer or device. However, virtual smart cards can be issued for the same user on multiple computers or devices by using additional certificates.
Enables multiple users to access network resources through the same computer by inserting their personal smart cards.	Enables multiple users to access network resources through the same computer or device by issuing a virtual smart card for each user on that computer or device.
Requires the user to carry the card, making it more difficult for an attacker to access the device and launch a hammering attempt.	Stores virtual smart card on the user's computer, which might be left unattended and allow a greater risk window for hammering attempts.
Provides a generally single-purpose device that is carried explicitly for the purpose of authentication. The smart card can be easily misplaced or forgotten.	Installs the virtual smart card on a device that has other purposes for the user, so the user has greater incentive to be responsible for the computer or device.
Alerts users that their card is lost or stolen only when they need to sign in and notice it's missing.	Installs the virtual smart card on a device that the user likely needs for other purposes, so users will notice its loss much more quickly. This reduces the associated risk window.
Requires companies to invest in smart cards and smart card readers for all employees.	Requires that companies ensure all employees have TPM-enabled computers, which are relatively common.
Enables using a smart card removal policy to affect system behavior when the smart card is removed. For example, the policy can dictate if the user's sign-in session is locked or terminated when the user removes the card.	Eliminates the necessity for a smart card removal policy because a TPM virtual smart card is always present and can't be removed from the computer.

Authentication design options

The following section presents several commonly used options and their respective strengths and weaknesses, which organizations can consider for authentication.

Passwords

A password is a secret string of characters that is tied to the identification credentials for a user's account. This establishes the user's identity. Although passwords are the most commonly used form of authentication, they're also the weakest. In a system where passwords are used as the sole method of user authentication, only individuals who know their passwords are considered valid users.

Password authentication places a great deal of responsibility on the user. Passwords must be sufficiently complex so they can't be easily guessed, but they must be simple enough to be committed to memory and not stored in a physical location. Even if this balance is successfully achieved, a wide variety of attacks exist (such as brute force attacks, eavesdropping, and social engineering tactics) where a malicious user can acquire a user's password and impersonate that person's identity. A user often won't realize that the password is compromised, which makes it's easy for a malicious user to maintain access to a system if a valid password has been obtained.

One-time passwords

A one-time password (OTP) is similar to a traditional password, but it's more secure in that it can be used only once to authenticate a user. The method for determining each new password varies by implementation. Assuming a secure deployment of each new password, OTPs have several advantages over the classic password model of authentication. Most importantly, if a given OTP token is intercepted in transmission between the user and the system, the interceptor can't use it for any future transactions. Similarly, if a malicious user obtains a valid user's OTP, the interceptor will have limited access to the system (only one session).

Smart cards

Smart cards are physical authentication devices, which improve on the concept of a password by requiring that users actually have their smart card device with them to access the system, in addition to knowing the PIN that provides access to the smart card. Smart cards have three key properties that help maintain their security:

- **Non-exportability**: Information stored on the card, such as the user's private keys, can't be extracted from one device and used in another medium

- **Isolated cryptography**: Any cryptographic operations that are related to the card (such as secure encryption and decryption of data) occur in a cryptographic processor on the card, so malicious software on the host computer can't observe the transactions
- **Anti-hammering**: To prevent access to the card by a brute-force attack, a set number of consecutive unsuccessful PIN entry attempts blocks the card until administrative action is taken

Smart cards provide greatly enhanced security over passwords alone, because it's much more difficult for a malicious user to gain and maintain access to a system. Most importantly, access to a smart card system requires that users have a valid card and that they know the PIN that provides access to that card. It's difficult for a thief to acquire the card and the PIN.

Additional security is achieved by the singular nature of the card because only one copy of the card exists, only one individual can use the sign-in credentials, and users will quickly notice if the card has been lost or stolen. This greatly reduces the risk window of credential theft when compared to using a password alone.

The additional security comes with added material and support costs. Traditional smart cards are expensive to purchase (cards and card readers must be supplied to employees), and users can misplace or lose them.

Virtual smart cards

Virtual smart cards emulate the functionality of traditional smart cards. Instead of requiring the purchase of additional hardware, virtual smart cards utilize technology that users already own and are more likely to always have with them. Theoretically, any device that can provide the three key properties of smart cards (nonexportability, isolated cryptography, and anti-hammering) can be commissioned as a virtual smart card. The virtual smart card platform is limited to the use of the Trusted Platform Module (TPM) chip, which is on most modern devices.

Virtual smart cards that utilize a TPM provide the three main security principles of traditional smart cards: nonexportability, isolated cryptography, and anti-hammering. Virtual smart cards are less expensive to implement and more convenient for users. Since many corporate computers already have a built-in TPM, there's no cost associated with purchasing new hardware. The user's possession of a computer or device is equivalent to the possession of a smart card, and a user's identity can't be assumed from any other computer or device without administrative provisioning of further credentials. Thus, two-factor authentication is achieved because the user must have a

computer that is set up with a virtual smart card and know the PIN to use the virtual smart card.

Feedback

Was this page helpful? Yes No

Provide product feedback

Get Started with Virtual Smart Cards: Walkthrough Guide

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **⚠ Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

This topic for the IT professional describes how to set up a basic test environment for using TPM virtual smart cards.

Virtual smart cards are a technology from Microsoft that offers comparable security benefits in two-factor authentication to physical smart cards. They also offer more convenience for users and lower cost for organizations to deploy. By utilizing Trusted Platform Module (TPM) devices that provide the same cryptographic capabilities as physical smart cards, virtual smart cards accomplish the three key properties that are desired by smart cards: nonexportability, isolated cryptography, and anti-hammering.

This step-by-step walkthrough shows you how to set up a basic test environment for using TPM virtual smart cards. After you complete this walkthrough, you'll have a functional virtual smart card installed on the Windows computer.

You should be able to complete this walkthrough in less than one hour, excluding installing software and setting up the test domain.

Walkthrough steps

- Prerequisites
- Step 1: Create the certificate template
- Step 2: Create the TPM virtual smart card
- Step 3: Enroll for the certificate on the TPM Virtual Smart Card

> **ⓘ Important**

This basic configuration is for test purposes only. It is not intended for use in a production environment.

Prerequisites

You'll need:

- A computer running Windows 10 with an installed and fully functional TPM (version 1.2 or version 2.0)
- A test domain to which the computer listed above can be joined
- Access to a server in that domain with a fully installed and running certification authority (CA)

Step 1: Create the certificate template

On your domain server, you need to create a template for the certificate that you request for the virtual smart card.

To create the certificate template

1. On your server, open the Microsoft Management Console (MMC). One way to do this is to type **mmc.exe** from the **Start** menu, right-click **mmc.exe**, and select **Run as administrator**
2. Select **File** > **Add/Remove Snap-in**
3. In the available snap-ins list, select **Certificate Templates**, and then select **Add**
4. Certificate Templates is now located under **Console Root** in the MMC. Double-click it to view all the available certificate templates
5. Right-click the **Smartcard Logon** template, and select **Duplicate Template**
6. On the **Compatibility** tab, under **Certification Authority**, review the selection, and change it if needed
7. On the **General** tab:
   a. Specify a name, such as **TPM Virtual Smart Card Logon**
   b. Set the validity period to the desired value
8. On the **Request Handling** tab:
   a. Set the **Purpose** to **Signature and smartcard logon**
   b. Select **Prompt the user during enrollment**
9. On the **Cryptography** tab:
   a. Set the minimum key size to 2048

b. Select **Requests must use one of the following providers**, and then select **Microsoft Base Smart Card Crypto Provider**

10. On the **Security** tab, add the security group that you want to give **Enroll** access to. For example, if you want to give access to all users, select the **Authenticated users** group, and then select **Enroll** permissions for them
11. Select **OK** to finalize your changes and create the new template. Your new template should now appear in the list of Certificate Templates
12. Select **File**, then select **Add/Remove Snap-in** to add the Certification Authority snap-in to your MMC console. When asked which computer you want to manage, select the computer on which the CA is located, probably **Local Computer**
13. In the left pane of the MMC, expand **Certification Authority (Local)**, and then expand your CA within the Certification Authority list
14. Right-click **Certificate Templates**, select **New**, and then select **Certificate Template to Issue**
15. From the list, select the new template that you created (**TPM Virtual Smart Card Logon**), and then select **OK**

> ⓘ **Note**
>
> It can take some time for your template to replicate to all servers and become available in this list.

1. After the template replicates, in the MMC, right-click in the Certification Authority list, select **All Tasks**, and then select **Stop Service**. Then, right-click the name of the CA again, select **All Tasks**, and then select **Start Service**.

Step 2: Create the TPM virtual smart card

In this step, you create the virtual smart card on the client computer by using the command-line tool, Tpmvscmgr.exe.

To create the TPM virtual smart card

1. On a domain-joined computer, open a Command Prompt window with Administrative credentials.
2. At the command prompt, type the following, and then press ENTER:

```
tpmvscmgr.exe create /name TestVSC /pin default /adminkey random /generate
```

This creates a virtual smart card with the name **TestVSC**, omit the unlock key, and generate the file system on the card. The PIN is set to the default, 12345678.

1. Wait several seconds for the process to finish. Upon completion, Tpmvscmgr.exe provides you with the device instance ID for the TPM Virtual Smart Card. Store this ID for later reference because you need it to manage or remove the virtual smart card. To be prompted for a PIN, instead of **/pin default** you can type **/pin prompt**.

For more information about the Tpmvscmgr command-line tool, see Use Virtual Smart Cards and Tpmvscmgr.

Step 3: Enroll for the certificate on the TPM Virtual Smart Card

The virtual smart card must be provisioned with a sign-in certificate for it to be fully functional.

To enroll the certificate

1. Open the Certificates console by typing **certmgr.msc** on the **Start** menu
2. Right-click **Personal**, select **All Tasks**, and then select **Request New Certificate**
3. Follow the prompts and when offered a list of templates, select the **TPM Virtual Smart Card Logon** check box (or whatever you named the template in Step 1)
4. If prompted for a device, select the Microsoft virtual smart card that corresponds to the one you created in the previous section. It displays as **Identity Device (Microsoft Profile)**
5. Enter the PIN that was established when you created the TPM virtual smart card, and then select **OK**
6. Wait for the enrollment to finish, and then select **Finish**

The virtual smart card can now be used as an alternative credential to sign in to your domain. To verify that your virtual smart card configuration and certificate enrollment were successful, sign out of your current session, and then sign in. When you sign in, you'll see the icon for the new TPM virtual smart card on the Secure Desktop (sign in) screen or you are automatically directed to the TPM smart card sign-in dialog box. Select the icon, enter your PIN (if necessary), and then select **OK**. You should be signed in to your domain account.

See also

- Understanding and Evaluating Virtual Smart Cards
- Use Virtual Smart Cards
- Deploy Virtual Smart Cards

Feedback

Was this page helpful? Yes No

Provide product feedback

Use Virtual Smart Cards

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> ⚠ **Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

Learn about the requirements for virtual smart cards, how to use and manage them.

Requirements, restrictions, and limitations

Expand table

Area	Requirements and details
Supported Trusted Platform Module (TPM)	Any TPM that adheres to the TPM main specifications for version 1.2 or version 2.0 (as set by the Trusted Computing Group) is supported for use as a virtual smart card. For more information, see the TPM Main Specification.
Supported virtual smart cards per computer	Ten smart cards can be connected to a computer or device at one time. This includes physical and virtual smart cards combined. **Note** You can create more than one virtual smart card; however, after creating more than four virtual smart cards, you may start to notice performance degradation. Because all smart cards appear as if they're always inserted, if more than one person shares a computer or device, each person can see all the virtual smart cards that are created on that computer or device. If the user knows the PIN values for all the virtual smart cards, the user will also be able to use them.
Supported number of certificates on a virtual smart card	A single TPM virtual smart card can contain 30 distinct certificates with the corresponding private keys. Users can continue to renew certificates on the card until the total number of certificates on a card exceeds 90. The reason that the total number of certificates is different from the total number of private keys is that sometimes the renewal can be done with the same private key—in which case a new private key isn't generated.

Area	Requirements and details
PIN, PIN Unlock Key (PUK), and Administrative key requirements	The PIN and the PUK must be a minimum of eight characters that can include numerals, alphabetic characters, and special characters. The Administrative key must be entered as 48 hexadecimal characters. It's a 3-key triple DES with ISO/IEC 9797 padding method 2 in CBC chaining mode.

Using Tpmvscmgr.exe

To create and delete TPM virtual smart cards for end users, the Tpmvscmgr command-line tool is included as a command-line tool with the operating system. You can use the **Create** and **Delete** parameters to manage virtual smart cards on local or remote computers. For information about using this tool, see Tpmvscmgr.

Create and delete virtual smart cards programmatically

Virtual smart cards can also be created and deleted by using APIs. For more information, see the following classes and interfaces:

- TpmVirtualSmartCardManager
- RemoteTpmVirtualSmartCardManager
- ITpmVirtualSmartCardManager
- ITPMVirtualSmartCardManagerStatusCallBack

You can use APIs in the `Windows.Device.SmartCards` namespace to build Microsoft Store apps to manage the full lifecycle of virtual smart cards. For information about how to build an app to do this, see Strong Authentication: Building Apps That Leverage Virtual Smart Cards in Enterprise, BYOD, and Consumer Environments.

The following table describes the features that can be developed in a Microsoft Store app:

Expand table

Feature	Physical Smart Card	Virtual Smart Card
Query and monitor smart card readers	Yes	Yes
List available smart cards in a reader, and retrieve the card name and card ID	Yes	Yes

Feature	Physical Smart Card	Virtual Smart Card
Verify if the administrative key of a card is correct	Yes	Yes
Provision (or reformat) a card with a given card ID	Yes	Yes
Change the PIN by entering the old PIN and specifying a new PIN	Yes	Yes
Change the administrative key, reset the PIN, or unblock the smart card by using a challenge/response method	Yes	Yes
Create a virtual smart card	Not applicable	Yes
Delete a virtual smart card	Not applicable	Yes
Set PIN policies	No	Yes

For more information about these Windows APIs, see:

- Windows.Devices.SmartCards namespace (Windows)
- Windows.Security.Cryptography.Certificates namespace (Windows)

Distinguishing TPM-based virtual smart cards from physical smart cards

To help users visually distinguish a Trusted Platform Module (TPM)-based virtual smart card from physical smart cards, the virtual smart card has a different icon. The virtual smart card icon  is displayed during sign-in, and on other screens that require the user to enter the PIN for a virtual smart card.

A TPM-based virtual smart card is labeled *Security Device* in the user interface.

Changing the PIN

The PIN for a virtual smart card can be changed by following these steps:

- Sign in with the old PIN or password
- Press `Ctrl` + `Alt` + `Del` and select **Change a password**
- Select **Sign-in Options**
- Select the virtual smart card icon
- Enter and confirm the new PIN

Resolving issues

TPM not provisioned

For a TPM-based virtual smart card to function properly, a provisioned TPM must be available on the computer:

- If the TPM is disabled in the BIOS, or it isn't provisioned with full ownership and the storage root key, the TPM virtual smart card creation fails
- If the TPM is initialized after creating a virtual smart card, the card will no longer function, and it must be re-created
- If the operating system is reinstalled, prior TPM virtual smart cards are no longer available and need to be re-created
- If the operating system is upgraded, prior TPM virtual smart cards are available to use in the upgraded operating system

TPM in lockout state

Sometimes, due to frequent incorrect PIN attempts from a user, the TPM may enter the lockout state. To resume using the TPM virtual smart card, it's necessary to reset the lockout on the TPM by using the owner's password or to wait for the lockout to expire. Unblocking the user PIN doesn't reset the lockout in the TPM. When the TPM is in lockout, the TPM virtual smart card appears as if it's blocked. When the TPM enters the lockout state because the user entered an incorrect PIN too many times, it may be necessary to reset the user PIN by using the virtual smart card management tools, such as Tpmvscmgr command-line tool.

See also

For information about authentication, confidentiality, and data integrity use cases, see Virtual Smart Card Overview.

Feedback

Was this page helpful? Yes No

Provide product feedback

Deploy Virtual Smart Cards

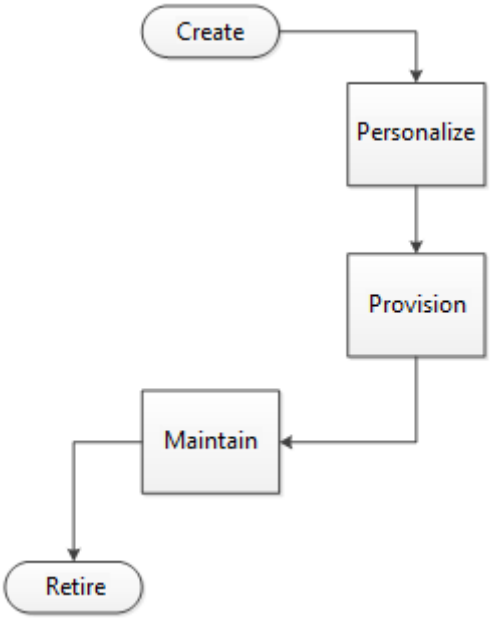
Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **⚠ Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

This article discusses the factors to consider when you deploy a virtual smart card authentication solution.

Traditional identity devices, such as physical smart cards, follow a predictable lifecycle in any deployment, as shown in the following diagram.



A device manufacturer creates physical devices, and then an organization purchase and deploy them. The device passes through the personalization stage, where its unique properties are set. In smart cards, these properties are the *administrator key*, *Personal Identification Number (PIN)*, *PIN Unlock Key (PUK)*, and its physical appearance. During the device provisioning phase, the required certificates are installed, such as a sign-in certificate. After you provision the device, it's ready for use. You'll maintain the device, for example you may replace cards when they're lost or stolen, or reset PINs when users

forget them. Finally, you'll retire devices when they exceed their intended lifetime or when employees leave the company.

This topic contains information about the following phases in a virtual smart card lifecycle:

- Create and personalize virtual smart cards
- Provision virtual smart cards
- Maintain virtual smart cards

Create and personalize virtual smart cards

A corporation purchases the devices to deploy then. The device passes through the personalization stage, where its unique properties are set. In smart cards, these properties are the administrator key, Personal Identification Number (PIN), PIN Unlock Key (PUK), and its physical appearance. The security that is provided for a TPM virtual smart card is fully provisioned in the host TPM.

Trusted Platform Module readiness

The TPM Provisioning Wizard, which is launched from the **TPM Management Console**, takes the user through all the steps to prepare the TPM for use.

When you create virtual smart cards, consider the following actions in the TPM:

- **Enable and Activate**: TPMs are built into many devices. In some cases, the TPM must be enabled and activated through the BIOS
- **Take ownership**: When you provision the TPM, you set an owner password for managing the TPM in the future, and you establish the *storage root key*. To provide anti-hammering protection for virtual smart cards, the user or a domain administrator must be able to reset the TPM owner password. For corporate use of TPM virtual smart cards, the domain administrator should restrict access to the TPM owner password by storing it in Active Directory, and not in the local registry. When TPM ownership is set, you must clear and reinitialize the TPM
- **Manage**: You can manage ownership of a virtual smart card by changing the owner password, and you can manage anti-hammering logic by resetting the lockout time

A TPM might operate in reduced functionality mode, which may occur if the operating system can't determine if the owner password is available to the user. During reduce functionality mode, you can use the TPM to create a virtual smart card, but it's

preferable to bring the TPM to a fully ready state so that any unexpected circumstances won't leave the user blocked from using the device.

Those smart card deployment management tools that require a status check of a TPM before attempting to create a TPM virtual smart card can do so using the TPM WMI interface.

Depending on the setup of the device designated for installing TPM virtual smart cards, it may be necessary to provision the TPM before continuing with the virtual smart card deployment. For more information about provisioning, see Use Virtual Smart Cards.

For more information about managing TPMs by using built-in tools, see Trusted Platform Module Services Group Policy Settings.

Creation

A TPM virtual smart card simulates a physical smart card, using the TPM to provide the same functionality as physical smart card hardware.
A virtual smart card appears within the operating system as a physical smart card that is always inserted. Windows presents a *virtual smart card reader* and a *virtual smart card* to applications using the same interface as physical smart cards. The messages to and from the virtual smart card are translated to TPM commands, ensuring the integrity of the virtual smart card through the three properties of smart card security:

- **Non-exportability**: Because all private information on the virtual smart card is encrypted by using the TPM on the host computer, it can't be used on a different computer with a different TPM. Additionally, TPMs are designed to be tamper-resistant and non-exportable, so a malicious user can't reverse engineer an identical TPM or install the same TPM on a different computer. For more information, see Evaluate Virtual Smart Card Security.
- **Isolated cryptography**: TPMs provide the same properties of isolated cryptography that is offered by physical smart cards, which is utilized by virtual smart cards. Unencrypted copies of private keys are loaded only within the TPM and never into memory that is accessible by the operating system. All cryptographic operations with these private keys occur inside the TPM.
- **Anti-hammering**: If a user enters a PIN incorrectly, the virtual smart card responds by using the anti-hammering logic of the TPM, which rejects further attempts for some time instead of blocking the card. This is also known as lockout. For more information, see Blocked virtual smart card and Evaluate Virtual Smart Card Security.

There are several options for creating virtual smart cards, depending on the size of the deployment and budget of the organization. The lowest cost option is using `tpmvscmgr.exe` to create cards individually on users' computers. Alternatively, a virtual smart card management solution can be purchased to more easily accomplish virtual smart card creation on a larger scale and aid in further phases of deployment. Virtual smart cards can be created on computers that are to be provisioned for an employee or on those that are already in an employee's possession. In either approach, there should be some central control over personalization and provisioning. If a computer is intended for use by multiple employees, multiple virtual smart cards can be created on a computer.

For information about the TPM Virtual Smart Card command-line tool, see Tpmvscmgr.

Personalization

During virtual smart card personalization, the values for the administrator key, PIN, and PUK are assigned. As with a physical card, knowing the administrator key is important for resetting the PIN or for deleting the card in the future. (If you set a PUK, you can't use the administrator key to reset the PIN.)

Because the administrator key is critical to the security of the card, it's important to consider the deployment environment and decide on the proper administrator key setting strategy. Options for these strategies include:

- **Uniform**: Administrator keys for all the virtual smart cards deployed in the organization are the same. Although using the same key makes the maintenance infrastructure easy (only one key needs to be stored), it's highly insecure. This strategy might be sufficient for small organizations, but if the administrator key is compromised, all virtual smart cards that use the key must be reissued
- **Random, not stored**: Administrator keys are assigned randomly for all virtual smart cards, and they aren't recorded. This is a valid option if the deployment administrators don't require the ability to reset PINs, and instead prefer to delete and reissue virtual smart cards. This is a viable strategy if the administrator prefers to set PUK values for the virtual smart cards and then use this value to reset PINs, if necessary
- **Random, stored**: you assign the administrator keys randomly, storing them in a central location. Each card's security is independent of the others. This is a secure strategy on a large scale, unless the administrator key database is compromised
- **Deterministic**: Administrator keys are the result of some function or known information. For example, the user ID could be used to randomly generate data that can be further processed through a symmetric encryption algorithm by using

a secret. This administrator key can be similarly regenerated when needed, and it doesn't need to be stored. The security of this method relies on the security of the secret used.

Although the PUK and the administrator key methodologies provide unlocking and resetting functionality, they do so in different ways. The PUK is a PIN that is entered on the computer to enable a user PIN reset.

The administrator key methodology takes a challenge-response approach. The card provides a set of random data after users verify their identity to the deployment administrator. The administrator then encrypts the data with the administrator key and gives the encrypted data back to the user. If the encrypted data matches that produced by the card during verification, the card will allow PIN reset. Because the administrator key is never accessible by anyone other than the deployment administrator, it can't be intercepted or recorded by any other party (including employees). This provides significant security benefits beyond using a PUK, an important consideration during the personalization process.

TPM virtual smart cards can be personalized on an individual basis when they're created with the Tpmvscmgr command-line tool. Or organizations can purchase a management solution that can incorporate personalization into an automated routine. Another advantage of such a solution is the automated creation of administrator keys. Tpmvscmgr.exe allows users to create their own administrator keys, which can be detrimental to the security of the virtual smart cards.

Provision virtual smart cards

Provisioning is the process of loading specific credentials onto a TPM virtual smart card. These credentials consist of certificates that are created to give users access to a specific service, such as domain sign-in. A maximum of 30 certificates is allowed on each virtual smart card. As with physical smart cards, several decisions must be made regarding the provisioning strategy, based on the environment of the deployment and the desired level of security.

A high-assurance level of secure provisioning requires absolute certainty about the identity of the individual who is receiving the certificate. Therefore, one method of high-assurance provisioning is utilizing previously provisioned strong credentials, such as a physical smart card, to validate identity during provisioning. In-person proofing at enrollment stations is another option, because an individual can easily and securely prove his or her identity with a passport or driver's license, although this can become infeasible on a larger scale. To achieve a similar level of assurance, a large organization can implement an "enroll-on-behalf-of" strategy, in which employees are enrolled with

their credentials by a superior who can personally verify their identities. This creates a chain of trust that ensures individuals are checked in person against their proposed identities, but without the administrative strain of provisioning all virtual smart cards from a single central enrollment station.

For deployments in which a high-assurance level isn't a primary concern, you can use self-service solutions. These can include using an online portal to obtain credentials or simply enrolling for certificates by using Certificate Manager, depending on the deployment. Consider that virtual smart card authentication is only as strong as the method of provisioning. For example, if weak domain credentials (such as a password alone) are used to request the authentication certificate, virtual smart card authentication will be equivalent to using only the password, and the benefits of two-factor authentication are lost.

For information about using Certificate Manager to configure virtual smart cards, see Get Started with Virtual Smart Cards: Walkthrough Guide.

High-assurance and self-service solutions approach virtual smart card provisioning by assuming that the user's computer has been issued prior to the virtual smart card deployment, but this isn't always the case. If virtual smart cards are being deployed with new computers, they can be created, personalized, and provisioned on the computer before the user has contact with that computer.

In this situation, provisioning becomes relatively simple, but identity checks must be put in place to ensure that the recipient of the computer is the individual who was expected during provisioning. This can be accomplished by requiring the employee to set the initial PIN under the supervision of the deployment administrator or manager.

When you're provisioning your computers, you should also consider the longevity of credentials that are supplied for virtual smart cards. This choice must be based on the risk threshold of the organization. Although longer lived credentials are more convenient, they're also more likely to become compromised during their lifetime. To decide on the appropriate lifetime for credentials, the deployment strategy must take into account the vulnerability of their cryptography (how long it could take to crack the credentials), and the likelihood of attack.

For compromised virtual smart cards, administrators should be able to revoke the associated credentials, like they would with a lost or stolen laptop. Revoking credentials requires a record of which credentials match which user and device, but the functionality doesn't natively exist in Windows. Deployment administrators might want to consider add-on solutions to maintain a record.

Virtual smart cards on consumer devices used for corporate access

There are techniques that allow employees to provision virtual smart cards and enroll for certificates that can be used to authenticate the users. This is useful when employees attempt to access corporate resources from devices that aren't joined to the corporate domain. Those devices can be further defined to not allow users to download and run applications from sources other than the Microsoft Store.

You can use APIs to build Microsoft Store apps that you can use to manage the full lifecycle of virtual smart cards. For more information, see Create and delete virtual smart cards programmatically.

TPM ownerAuth in the registry

When a device or computer isn't joined to a domain, the TPM ownerAuth is stored in the registry under HKEY_LOCAL_MACHINE. This exposes some threats. Most of the threat vectors are protected by BitLocker, but threats that aren't protected include:

- A malicious user possesses a device that has an active local sign-in session before the device locks. The malicious user could attempt a brute-force attack on the virtual smart card PIN, and then access the corporate secrets
- A malicious user possesses a device that has an active virtual private network (VPN) session. The device is then compromised

The proposed mitigation for the previous scenarios is to use Exchange ActiveSync (EAS) policies to reduce the automatic lockout time from five minutes to 30 seconds of inactivity. You can set policies for automatic lockout while provisioning virtual smart cards. If an organization wants more security, they can also configure a setting to remove the ownerAuth from the local device.

For configuration information about the TPM ownerAuth registry key, see the Group Policy setting **Configure the level of TPM owner authorization information** available to the operating system.

For information about EAS policies, see Exchange ActiveSync Policy Engine Overview.

Managed and unmanaged cards

The following table describes the important differences between managed and unmanaged virtual smart cards that exist on consumer devices:

Expand table

Operation	Managed and unmanaged cards	Unmanaged cards
Reset PIN when the user forgets the PIN	Yes	No. Delete and recreate the card.
Allow user to change the PIN	Yes	No. Delete and recreate the card.

Managed cards

A managed virtual smart card can be serviced by the IT administrator or another person in that designated role. It allows the IT administrator to have influence or complete control over specific aspects of the virtual smart card from its creation to deletion. To manage these cards, a virtual smart card deployment management tool is often required.

Managed card creation

A user can create blank virtual smart card by using the *Tpmvscmgr* command-line tool, which is a built-in tool executed with administrative credentials through an elevated command prompt. The virtual smart card must be created with well-known parameters (such as default values), and it should be left unformatted (specifically, the **/generate** option shouldn't be specified).

The following command creates a virtual smart card that can later be managed by a smart card management tool launched from another computer (as explained in the next section):

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey DEFAULT /PIN PROMPT
```

Alternatively, instead of using a default administrator key, a user can enter an administrator key at the command line:

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey PROMPT /PIN PROMPT
```

In either case, the card management system needs to be aware of the initial administrator key. The requirement is so that the card management system can take ownership of the virtual smart card and change the administrator key to a value that is

only accessible through the card management tool operated by the IT administrator. For example, when you use the default, the administrator key is set to:

```
102030405060708010203040506070801020304050607080
```

For information about using this command-line tool, see Tpmvscmgr.

Managed card management

After the virtual smart card is created, the user needs to open a remote desktop connection to an enrollment station, for example, in a computer that is joined to the domain. Virtual smart cards that are associated with a client computer are available for use in the remote desktop connection. The user can open a card management tool inside the remote session that can take ownership of the card and provision it for use by the user. This requires that a user is allowed to establish a remote desktop connection from a non-domain-joined computer to a domain-joined computer. This might require a specific network configuration, such as through IPsec policies.

When users need to reset or change a PIN, they need to use the remote desktop connection to complete these operations. They can use the built-in tools for PIN unlock and PIN change or the smart card management tool.

Certificate management for managed cards

Similar to physical smart cards, virtual smart cards require certificate enrollment.

Certificate issuance

Users can enroll for certificates from within a remote desktop session that is established to provision the card. This process can also be managed by the smart card management tool that the user runs through the remote desktop connection. This model works for deployments that require the user to sign a request for enrollment by using a physical smart card. The driver for the physical smart card doesn't need to be installed on the client computer if it's installed on the remote computer. This is made possible by smart card redirection functionality, which ensures that smart cards that are connected to the client computer are available for use during a remote session.

Alternatively, without establishing a remote desktop connection, users can enroll for certificates from the Certificate Management console (certmgr.msc) on a client computer. Users can also create a request and submit it to a server from within a custom certificate enrollment application (for example, a registration authority) that has controlled access to the certification authority (CA). This requires specific enterprise

configuration and deployments for Certificate Enrollment Policies (CEP) and Certificate Enrollment Services (CES).

Certificate lifecycle management

You can renew certificates through remote desktop connections, certificate enrollment policies, or certificate enrollment services. Renewal requirements could be different from the initial issuance requirements, based on the renewal policy.

Certificate revocation requires careful planning. When information about the certificate to be revoked is reliably available, the specific certificate can be easily revoked. When information about the certificate to be revoked isn't easy to determine, all certificates that are issued to the user under the policy that was used to issue the certificate might need to be revoked. For example, this could occur if an employee reports a lost or compromised device, and information that associates the device with a certificate isn't available.

Unmanaged cards

Unmanaged virtual smart cards aren't serviceable by an IT administrator. Unmanaged cards might be suitable if an organization doesn't have an elaborate smart card deployment management tool and using remote desktop connections to manage the card isn't desirable. Because unmanaged cards aren't serviceable by the IT administrator, when a user needs help with a virtual smart card (for example, resetting or unlocking a PIN), the only option available to the user is to delete the card and create it again. This results in loss of the user's credentials and he or she must re-enroll.

Unmanaged card creation

A user can create a virtual smart card by using the Tpmvscmgr command-line tool, which is run with administrative credentials through an elevated command prompt. The following command creates an unmanaged card that can be used to enroll certificates:

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey RANDOM /PIN PROMPT /generate
```

This command creates a card with a randomized administrator key. The key is automatically discarded after the creation of the card. If users forget or want to change their PIN, they need to delete the card and create it again. To delete the card, a user can run the following command:

```
tpmvscmgr.exe destroy /instance <instance ID>
```

where `<instance ID>` is the value that is printed on the screen when the user creates the card. Specifically, for the first card created, the instance ID is `ROOT\SMARTCARDREADER\0000`.

Certificate management for unmanaged cards

Depending on the security requirements that are unique to an organization, users can initially enroll for certificates from the certificate management console (certmgr.msc) or from within custom certificate enrollment applications. The latter method can create a request and submit it to a server that has access to the Certification Authority. This requires specific organizational configurations and deployments for certificate enrollment policies and certificate enrollment services. Windows has built-in tools, specifically Certreq.exe and Certutil.exe, which can be used by scripts to perform the enrollment from the command line.

Requesting the certificate by providing domain credentials only

The simplest way for users to request certificates is to provide their domain credentials through a script that can perform the enrollment through built-in components you have in place for certificate requests.

Alternatively, an application (such as a line-of-business app) can be installed on the computer to perform enrollment by generating a request on the client. The request is submitted to an HTTP server, which can forward it to a registration authority.

Another option is to have the user access an enrollment portal that is available through Internet Explorer. The webpage can use the scripting APIs to perform certificate enrollment.

Signing the request with another certificate

You can provide users with a short-term certificate through a Personal Information Exchange (.pfx) file. You can generate the .pfx file by initiating a request from a domain-joined computer. You can enforce other policy constraints on the .pfx file to assert the identity of the user.

The user can import the certificate into the **MY** store (which is the user's certificate store). And your organization can present the user with a script that can be used to sign the request for the short-term certificate and to request a virtual smart card.

For deployments that require users to use a physical smart card to sign the certificate request, you can use the procedure:

1. Users initiate a request on a domain-joined computer
2. Users complete the request by using a physical smart card to sign the request
3. Users download the request to the virtual smart card on their client computer

Using one-time password for enrollment

Another option to ensure that users are strongly authenticated before virtual smart card certificates are issued, is to send a user a one-time password through SMS, email, or phone. The user then types the one-time password during the certificate enrollment from an application or a script on a desktop that invokes built-in command-line tools.

Certificate lifecycle management: certificate renewal can be done from the same tools that are used for the initial certificate enrollment. Certificate enrollment policies and certificate enrollment services can also be used to perform automatic renewal.

Certificate revocation requires careful planning. When information about the certificate to be revoked is reliably available, the specific certificate can be easily revoked. When information about the certificate to be revoked isn't easy to determine, all certificates issued to the user under the policy that was used to issue the certificate might need to be revoked. For example, if an employee reports a lost or compromised device, and information that associates the device with a certificate isn't available.

Maintain virtual smart cards

Maintenance is a significant portion of the virtual smart card lifecycle and one of the most important considerations from a management perspective. After virtual smart cards are created, personalized, and provisioned, they can be used for convenient two-factor authentication. Deployment administrators must be aware of several common administrative scenarios, which can be approached by using a purchased virtual smart card solution or on a case-by-case basis with in-house methods.

Renewal: Renewing virtual smart card credentials is a regular task that is necessary to preserve the security of a virtual smart card deployment. Renewal is the result of a signed request from a user who specifies the key pair desired for the new credentials. Depending on user's choice or deployment specification, the user can request credentials with the same key pair as previously used, or choose a newly generated key pair.

When renewing with a previously used key, no extra steps are required because a strong certificate with this key was issued during the initial provisioning. However, when the user requests a new key pair, you must take the same steps that were used during provisioning to assure the strength of the credentials. Renewal with new keys should occur periodically to counter sophisticated long-term attempts by malicious users to infiltrate the system. When new keys are assigned, you must ensure that the new keys are being used by the expected individuals on the same virtual smart cards.

Reset PINs: Resetting virtual smart card PINs is also a frequent necessity, because employees forget their PINs. There are two ways to accomplish this, depending on choices made earlier in the deployment: Use a PUK (if the PUK is set), or use a challenge-response approach with the administration key. Before resetting the PIN, the user's identity must be verified by using some means other than the card—most likely the verification method that you used during initial provisioning (for example, in-person proofing). This is necessary in user-error scenarios when users forget their PINs. However, you should never reset a PIN if it has been compromised because the level of vulnerability after the PIN is exposed is difficult to identify. The entire card should be reissued.

Lockout reset: A frequent precursor to resetting a PIN is the necessity of resetting the TPM lockout time because the TPM anti-hammering logic will be engaged with multiple PIN entry failures for a virtual smart card. This is currently device specific.

Retire cards: The final aspect of virtual smart card management is retiring cards when they're no longer needed. When an employee leaves the company, it's desirable to revoke domain access. Revoking sign-in credentials from the certification authority (CA) accomplishes this goal.

The card should be reissued if the same computer is used by other employees without reinstalling the operating system. Reusing the former card can allow the former employee to change the PIN after leaving the organization, and then hijack certificates that belong to the new user to obtain unauthorized domain access. However, if the employee takes the virtual smart card-enabled computer, it's only necessary to revoke the certificates that are stored on the virtual smart card.

Emergency preparedness

Card reissuance

The most common scenario in an organization is reissuing virtual smart cards, which can be necessary if the operating system is reinstalled or if the virtual smart card is compromised in some manner. Reissuance is essentially the recreation of the card,

which involves establishing a new PIN and administrator key and provisioning a new set of associated certificates. This is an immediate necessity when a card is compromised, for example, if the virtual smart card-protected computer is exposed to an adversary who might have access to the correct PIN. Reissuance is the most secure response to an unknown exposure of a card's privacy. Additionally, reissuance is necessary after an operating system is reinstalled because the virtual smart card device profile is removed with all other user data when the operating system is reinstalled.

Blocked virtual smart card

The anti-hammering behavior of a TPM virtual smart card is different from that of a physical smart card. A physical smart card blocks itself after the user enters the wrong PIN a few times. A TPM virtual smart card enters a timed delay after the user enters the wrong PIN a few times. If the TPM is in the timed-delay mode, when the user attempts to use the TPM virtual smart card, the user is notified that the card is blocked. Furthermore, if you enable the integrated unlock functionality, the user can see the user interface to unlock the virtual smart card and change the PIN. Unlocking the virtual smart card doesn't reset the TPM lockout. The user needs to perform an extra step to reset the TPM lockout or wait for the timed delay to expire.

For more information about setting the Allow Integrated Unblock policy, see Allow Integrated Unblock screen to be displayed at the time of logon.

Feedback

Was this page helpful? Yes No

Provide product feedback

Evaluate Virtual Smart Card Security

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

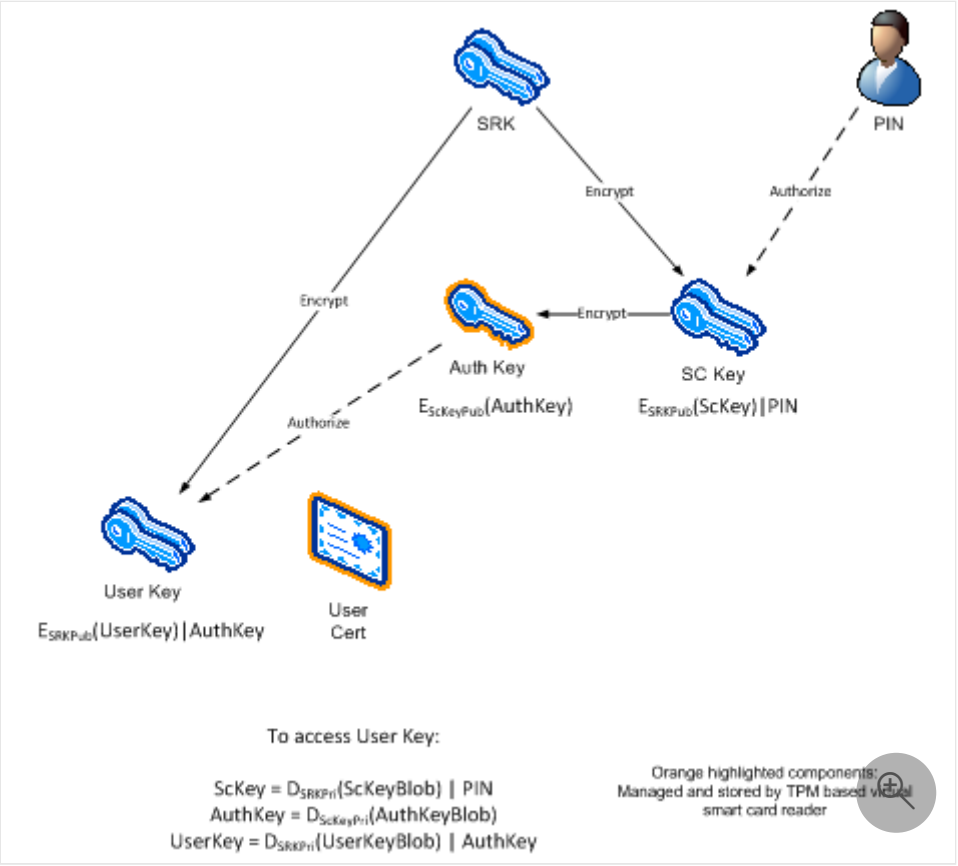
> ⚠ **Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

In this article, you'll learn about security characteristics and considerations when deploying TPM virtual smart cards.

Virtual smart card non-exportability details

A crucial aspect of TPM virtual smart cards is their ability to securely store and use secret data. Specifically, that the secured data is non-exportable.
Data can be accessed and used within the virtual smart card system, but it's meaningless outside of its intended environment. In TPM virtual smart cards, security is ensured with a secure key hierarchy, which includes several chains of encryption. The chain originates with the TPM storage root key, which is generated and stored within the TPM and never exposed outside the chip. The TPM key hierarchy is designed to allow encryption of user data with the storage root key, but it authorizes decryption with the user PIN so that changing the PIN doesn't require re-encryption of the data.

The following diagram illustrates the secure key hierarchy and the process of accessing the user key.



The following keys are stored on the hard disk:

- User key
- Smart card key, which is encrypted by the storage root key
- Authorization key for the user key decryption, which is encrypted by the public portion of the smart card key

When the user enters a PIN, the use of the decrypted smart card key is authorized with this PIN. If this authorization succeeds, the decrypted smart card key is used to decrypt the auth key. The auth key is then provided to the TPM to authorize the decryption and use of the user's key that is stored on the virtual smart card.

The auth key is the only sensitive data used as plaintext outside the TPM, but its presence in memory is protected by Microsoft Data Protection API (DPAPI), such that before being stored in any way, it's encrypted. All data other than the auth key is processed only as plaintext within the TPM, which is isolated from external access.

Virtual smart card anti-hammering details

The anti-hammering functionality of virtual smart cards relies on the anti-hammering functionality of the TPM that is enabling the virtual smart card. However, the TPM version 1.2 and subsequent specifications (as designed by the Trusted Computing Group) provide flexible guidelines for responding to hammering. The spec requires only

that the TPM implement protection against trial-and-error attacks on the user PIN, PUK, and challenge/response mechanism.

The Trusted Computing Group specifies that if the response to attacks involves suspending proper function of the TPM for some period of time, or until administrative action is taken, the TPM must prevent running the authorized TPM commands. The TPM can prevent running any TPM commands until the termination of the attack response. Beyond using a time delay or requiring administrative action, a TPM could also force a reboot when an attack is detected. The Trusted Computing Group allows manufacturers a level of creativity in their choice of implementation. The methodology used by TPM manufacturers determines the anti-hammering response of TPM virtual smart cards. Some typical aspects of protection from attacks include:

1. Allow only a limited number of wrong PIN attempts before enabling a lockout that enforces a time delay before any further commands are accepted by the TPM.

   > **Note**
   >
   > If the user enters the wrong PIN five consecutive times for a virtual smart card (which works in conjunction with the TPM), the card is blocked. When the card is blocked, it must be unblocked by using the administrative key or the PUK.

2. Increase the time delay exponentially as the user enters the wrong PIN so that an excessive number of wrong PIN attempts quickly trigger long delays in accepting commands.

3. Have a failure leakage mechanism to allow the TPM to reset the timed delays over a period of time. This is useful in cases where a valid user has entered the wrong PIN occasionally, for example, due to complexity of the PIN.

For example, it will take 14 years to guess an eight character PIN for a TPM that implements the following protection:

1. Number of wrong PINs allowed before entering lockout (threshold): 9
2. Time the TPM is in lockout after the threshold is reached: 10 seconds
3. Timed delay doubles for each wrong PIN after the threshold is reached

---

Feedback

Was this page helpful? Yes No

Provide product feedback

Tpmvscmgr

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **⚠ Warning**
>
> Windows Hello for Business and FIDO2 security keys are modern, two-factor authentication methods for Windows. Customers using virtual smart cards are encouraged to move to Windows Hello for Business or FIDO2. For new Windows installations, we recommend Windows Hello for Business or FIDO2 security keys.

The Tpmvscmgr command-line tool allows users with Administrative credentials to create and delete TPM virtual smart cards on a computer. For examples of how this command can be used, see Examples.

Syntax

```
Tpmvscmgr create [/quiet] /name <name> /AdminKey {DEFAULT | PROMPT | RANDOM} [/PIN {DEFAULT | PROMPT}] [/PUK {DEFAULT | PROMPT}] [/generate] [/machine <machine name>] [/pinpolicy [policy options]] [/attestation {AIK_AND_CERT | AIK_ONLY}] [/?]
```

```
Tpmvscmgr destroy [/quiet] [/instance <device instance ID>] [/machine <machine name>] [/?]
```

Parameters for Create command

The Create command sets up new virtual smart cards on the user's system. It returns the instance ID of the newly created card for later reference if deletion is required. The instance ID is in the format `ROOT\SMARTCARDREADER\000n` where n starts from 0 and is increased by 1 each time you create a new virtual smart card.

Expand table

Parameter	Description
/name	Required. Indicates the name of the new virtual smart card.
/AdminKey	Indicates the desired administrator key that can be used to reset the PIN of the card if the user forgets the PIN.

Parameter	Description
	DEFAULT Specifies the default value of 010203040506070801020304050607080102030405060708. **PROMPT** Prompts the user to enter a value for the administrator key. **RANDOM** Results in a random setting for the administrator key for a card that is not returned to the user. This creates a card that might not be manageable by using smart card management tools. When generated with RANDOM, the administrator key is set as 48 hexadecimal characters.
/PIN	Indicates desired user PIN value. **DEFAULT** Specifies the default PIN of 12345678. **PROMPT** Prompts the user to enter a PIN at the command line. The PIN must be a minimum of eight characters, and it can contain numerals, characters, and special characters.
/PUK	Indicates the desired PIN Unlock Key (PUK) value. The PUK value must be a minimum of eight characters, and it can contain numerals, characters, and special characters. If the parameter is omitted, the card is created without a PUK. **DEFAULT** Specifies the default PUK of 12345678. **PROMPT** Prompts the user to enter a PUK at the command line.
/generate	Generates the files in storage that are necessary for the virtual smart card to function. If the /generate parameter is omitted, it's equivalent to creating a card without this file system. A card without a file system can be managed only by a smart card management system such as Microsoft Configuration Manager.
/machine	Allows you to specify the name of a remote computer on which the virtual smart card can be created. This can be used in a domain environment only, and it relies on DCOM. For the command to succeed in creating a virtual smart card on a different computer, the user running this command must be a member in the local administrators group on the remote computer.
/pinpolicy	If **/pin prompt** is used, **/pinpolicy** allows you to specify the following PIN policy options: **minlen** <minimum PIN length> If not specified, defaults to 8. The lower bound is 4. **maxlen** <maximum PIN length> If not specified, defaults to 127. The upper bound is 127. **uppercase** Can be **ALLOWED**, **DISALLOWED**, or **REQUIRED.** Default is **ALLOWED.** **lowercase** Can be **ALLOWED**, **DISALLOWED**, or **REQUIRED.** Default is **ALLOWED.** **digits** Can be **ALLOWED**, **DISALLOWED**, or **REQUIRED.** Default is **ALLOWED.** **specialchars** Can be **ALLOWED**, **DISALLOWED**, or **REQUIRED.** Default is **ALLOWED.** When using **/pinpolicy**, PIN characters must be printable ASCII characters.
/attestation	Configures attestation (subject only). This attestation uses an Attestation Identity Key (AIK) certificate as a trust anchor to vouch that the virtual smart card keys and certificates are truly hardware bound. The attestation methods are: **AIK_AND_CERT** Creates an AIK and obtains an AIK certificate from the Microsoft cloud certification authority (CA). This requires the device to have a TPM with an EK

Parameter	Description
	certificate. If this option is specified and there's no network connectivity, it's possible that creation of the virtual smart card will fail. **AIK_ONLY** Creates an AIK but doesn't obtain an AIK certificate.
/?	Displays Help for this command.

Parameters for Destroy command

The Destroy command securely deletes a virtual smart card from a computer.

> **⚠ Warning**
>
> When a virtual smart card is deleted, it cannot be recovered.

Expand table

Parameter	Description
/instance	Specifies the instance ID of the virtual smart card to be removed. The instanceID was generated as output by Tpmvscmgr.exe when the card was created. The **/instance** parameter is a required field for the Destroy command.
/machine	Allows you to specify the name of a remote computer on which the virtual smart card will be deleted. This can be used in a domain environment only, and it relies on DCOM. For the command to succeed in deleting a virtual smart card on a different computer, the user running this command must be a member in the local administrators group on the remote computer.
/?	Displays Help for this command.

Remarks

Membership in the Administrators group (or equivalent) on the target computer is the minimum required to run all the parameters of this command.

For alphanumeric inputs, the full 127 character ASCII set is allowed.

Examples

The following command shows how to create a virtual smart card that can be later managed by a smart card management tool launched from another computer.

Console

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey DEFAULT
/PIN PROMPT
```

Alternatively, instead of using a default administrator key, you can create an administrator key at the command line. The following command shows how to create an administrator key.

Console

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey PROMPT
/PIN PROMPT
```

The following command will create the unmanaged virtual smart card that can be used to enroll certificates.

Console

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /AdminKey RANDOM
/PIN PROMPT /generate
```

The preceding command will create a virtual smart card with a randomized administrator key. The key is automatically discarded after the card is created. This means that if the user forgets the PIN or wants to the change the PIN, the user needs to delete the card and create it again. To delete the card, the user can run the following command.

Console

```
tpmvscmgr.exe destroy /instance <instance ID>
```

where <instance ID> is the value printed on the screen when the user created the card. Specifically, for the first card created, the instance ID is ROOT\SMARTCARDREADER\0000.

The following command will create a TPM virtual smart card with the default value for the administrator key and a specified PIN policy and attestation method:

Console

```
tpmvscmgr.exe create /name "VirtualSmartCardForCorpAccess" /PIN PROMPT
/pinpolicy minlen 4 maxlen 8 /AdminKey DEFAULT /attestation AIK_AND_CERT
/generate
```

Feedback

Was this page helpful? Yes No

Provide product feedback

Enterprise certificate pinning overview

Article • 12/02/2024 • Applies to: ✅ Windows 11, ✅ Windows 10

Enterprise certificate pinning is a Windows feature for remembering (pinning), a root issuing certificate authority, or end-entity certificate, to a domain name.
The feature helps to reduce man-in-the-middle attacks by protecting internal domain names from chaining to unwanted or fraudulently issued certificates.

> ⓘ **Note**
>
> External domain names, where the certificate issued to these domains is issued by a public certificate authority, are not ideal for enterprise certificate pinning.

Windows Certificate APIs (*CertVerifyCertificateChainPolicy* and *WinVerifyTrust*) are updated to check if the site's chain that authenticates servers matches a restricted set of certificates.
The restrictions are encapsulated in a *Pin Rules Certificate Trust List (CTL)* that is configured and deployed to Windows devices.
Any site certificates that trigger a name mismatch causes Windows to write an event to the *CAPI2 event log*, and prevents the user from browsing the web site.

> ⓘ **Note**
>
> Enterprise Certificate Pinning feature triggering doesn't cause clients other than Microsoft Edge to block the connection.

Deployment

To deploy enterprise certificate pinning, you need to:

- Create a well-formatted certificate pinning rule XML file
- Create a pin rules certificate trust list file from the XML file
- Apply the pin rules certificate trust list file to a reference administrative computer
- Deploy the registry configuration on the reference computer via group policy

Create a pin rules XML file

The XML-based pin rules file consists of a sequence of PinRule elements. Each PinRule element contains a sequence of one or more Site elements and a sequence of zero or more Certificate elements.

```
XML
```

```
<PinRules ListIdentifier="PinRulesExample" Duration="P28D">

  <PinRule Name="AllCertificateAttributes" Error="None" Log="true">
    <Certificate File="Single.cer"/>
    <Certificate File="Multiple.p7b"/>
    <Certificate File="Multiple.sst"/>
    <Certificate Directory="Multiple"/>
    <Certificate Base64="MIIBy … QFzuM"/>
    <Certificate File="WillExpire.cer" EndDate="2015-05-12T00:00:00Z"/>
    <Site Domain="xyz.com"/>
  </PinRule>

  <PinRule Name="MultipleSites" Log="false">
    <Certificate File="Root.cer"/>
    <Site Domain="xyz.com"/>
    <Site Domain=".xyz.com"/>
    <Site Domain="*.abc.xyz.com" AllSubdomains="true"/>
    <Site Domain="WillNormalize.com"/>
  </PinRule>

</PinRules>
```

PinRules element

The PinRules element can have the following attributes. For help with formatting Pin Rules, see Represent a date in XML or Represent a duration in XML.

Expand table

Attribute	Description	Required
Duration or **NextUpdate**	Specifies when the Pin Rules expires. Either is required. **NextUpdate** takes precedence if both are specified. **Duration**, represented as an XML TimeSpan data type, doesn't allow years and months. You represent the **NextUpdate** attribute as an XML DateTime data type in UTC.	**Required?** Yes. At least one is required.
LogDuration or **LogEndDate**	Configures auditing only to extend beyond the expiration of enforcing the Pin Rules. **LogEndDate**, represented as an XML DateTime data type in UTC, takes precedence if both are specified. You represent **LogDuration** as an XML TimeSpan data type, which doesn't allow years and months. If `none of the attributes are specified, auditing expiration uses **Duration** or **NextUpdate** attributes.	No.
ListIdentifier	Provides a friendly name for the list of pin rules. Windows doesn't use this attribute for certificate pinning enforcement; however, it's included when the pin rules are converted to a certificate trust list (CTL).	No.

PinRule element

The **PinRule** element can have the following attributes.

Expand table

Attribute	Description	Required
Name	Uniquely identifies the **PinRule**. Windows uses the attribute to identify the element for a parsing error or for verbose output. The attribute isn't included in the generated certificate trust list (CTL).	Yes.
Error	Describes the action Windows performs when it encounters a PIN mismatch. You can choose from the following string values: - **Revoked** - Windows reports the certificate protecting the site as if it was revoked. This typically prevents the user from accessing the site. - **InvalidName** - Windows reports the certificate protecting the site as if the name on the certificate doesn't match the name of the site. This typically results in prompting the user before accessing the site. - **None** - The default value. No error is returned. You can use the setting to audit the pin rules without introducing any user friction.	No.
Log	A Boolean value represents a string that equals **true** or **false**. By default, logging is enabled (**true**).	No.

Certificate element

The **Certificate** element can have the following attributes.

Expand table

Attribute	Description	Required
File	Path to a file containing one or more certificates. Where the certificate(s) can be encoded as: - single certificate - p7b - sst These files can also be Base64 formatted. All **Site** elements included in the same **PinRule** element can match any of these certificates.	Yes (File, Directory, or Base64 must be present).
Directory	Path to a directory containing one or more of the above certificate files. Skips any files not containing any certificates.	Yes (File, Directory, or Base64 must be present).
Base64	Base64 encoded certificate(s). Where the certificate(s) can be encoded as: - single certificate - p7b - sst This allows the certificates to be included in the XML file without a file directory dependency. Note: You can use **certutil -encode** to convert a .cer file into base64. You can	Yes (File, Directory, or Base64 must be present).

Attribute	Description	Required
	then use Notepad to copy and paste the base64 encoded certificate into the pin rule.	
EndDate	Enables you to configure an expiration date for when the certificate is no longer valid in the pin rule. If you are in the process of switching to a new root or CA, you can set the **EndDate** to allow matching of this element's certificates. If the current time is past the **EndDate**, when creating the certificate trust list (CTL) the parser outputs a warning message and excludes the certificate(s) from the Pin Rule in the generated CTL. For help with formatting Pin Rules, see Represent a date in XML.	No.

Site element

The **Site** element can have the following attributes.

Expand table

Attribute	Description	Required
Domain	Contains the DNS name to be matched for this pin rule. When you create the certificate trust list, the parser normalizes the input name string value as follows: - If the DNS name has a leading "*", it's removed. - Non-ASCII DNS name is converted to ASCII Puny Code. - Upper case ASCII characters are converted to lower case. If the normalized name has a leading ".", then wildcard left-hand label matching is enabled. For example, ".xyz.com" would match "abc.xyz.com".	Yes.
AllSubdomains	By default, wildcard left-hand label matching is restricted to a single left-hand label. This attribute can be set to "true" to enable wildcard matching of all of the left-hand labels. For example, setting this attribute would also match "123.abc.xyz.com" for the ".xyz.com" domain value.	No.

Create a pin rules certificate trust list

The *Certutil.exe* command includes the *generatePinRulesCTL* argument. The argument parses the XML file and generates the encoded certificate trust list (CTL) that you add to your reference Windows device and then deploy. The syntax is:

Windows Command Prompt

```
CertUtil [Options] -generatePinRulesCTL XMLFile CTLFile [SSTFile]
  Generate Pin Rules CTL
    XMLFile -- input XML file to be parsed.
    CTLFile -- output CTL file to be generated.
    SSTFile -- optional .sst file to be created.
         The .sst file contains all of the certificates
         used for pinning.
```

```
Options:
  -f                -- Force overwrite
  -v                -- Verbose operation
```

- The same certificate(s) can occur in multiple **PinRule** elements
- The same domain can occur in multiple **PinRule** elements
- Certutil coalesces these in the resultant pin rules certificate trust list
- Certutil.exe doesn't strictly enforce the XML schema definition

Certutil performs the following to enable other tools to add/consume their own specific elements and attributes:

- Skips elements before and after the **PinRules** element
- Skips any element not matching **Certificate** or **Site** within the **PinRules** element
- Skips any attributes not matching the above names for each element type

Use the *certutil* command with the *generatePinRulesCTL* argument along with your XML file that contains your certificate pinning rules. Lastly, provide the name of an output file that will include your certificate pinning rules in the form of a certificate trust list.

Windows Command Prompt

```
certutil -generatePinRulesCTL certPinRules.xml pinrules.stl
```

Apply certificate pinning rules to a reference computer

Now that your certificate pinning rules are in the certificate trust list format, you need to apply the settings to a reference computer as a prerequisite to deploying the setting to your enterprise. To simplify the deployment configuration, it's best to apply your certificate pinning rules to a computer that has the Group Policy Management Console (GPMC) included in the Remote Server Administration Tools (RSAT).

Use *certutil.exe* to apply your certificate pinning rules to your reference computer using the *setreg* argument.
The *setreg* argument takes a secondary argument that determines the location of where certutil writes the certificate pining rules.
The secondary argument is *chain\PinRules*.
The last argument you provide is the name of file that contains your certificate pinning rules in certificate trust list format (`.stl`).
You pass the name of the file as the last argument. You must prefix the file name with the `@` symbol as in the following example:

Windows Command Prompt

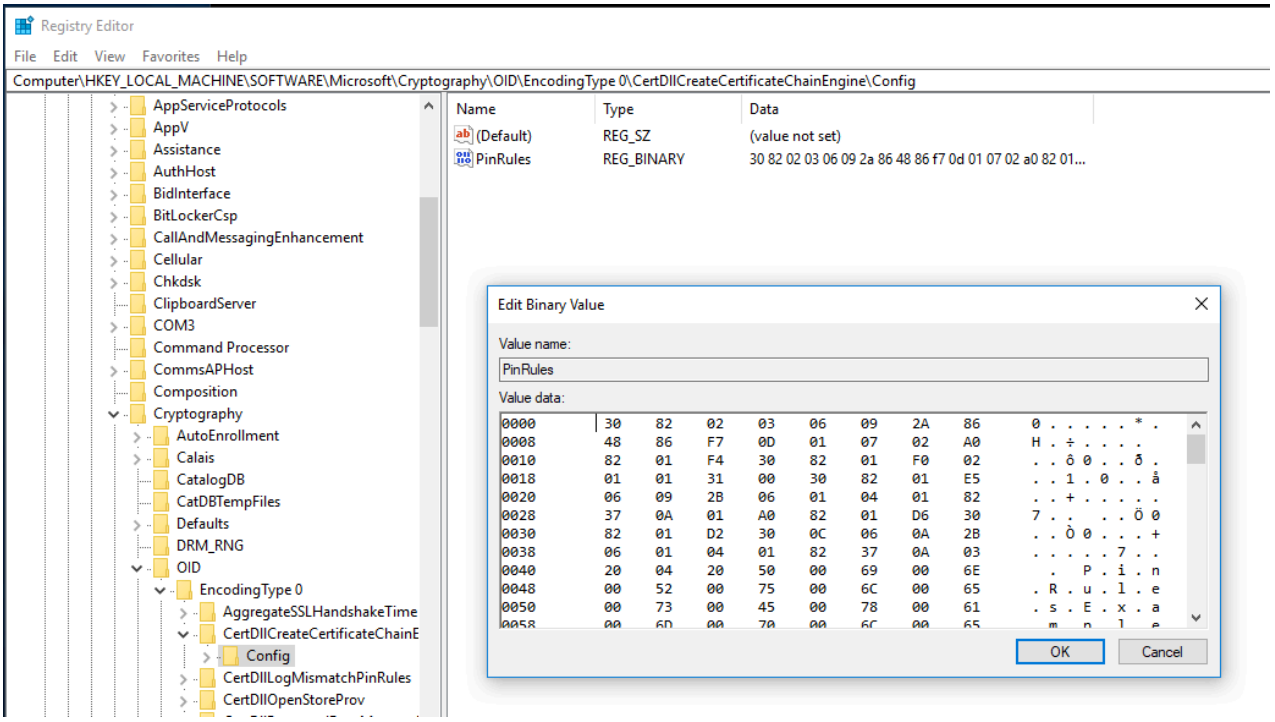
```
Certutil -setreg chain\PinRules @pinrules.stl
```

> **Note**
>
> You must execute the command from an elevated command prompt.

Certutil writes the binary information to the following registration location:

Expand table

Name	Value
Key	HKLM\SOFTWARE\Microsoft\Cryptography\OID\EncodingType0\CertDllCreateCertificateChainEngine\Config
Name	PinRules
Value	Binary contents from the certificate pin rules certificate trust list file
Data type	REG_BINARY



Deploy enterprise pin rule settings using group policy

From the XML file, you've created a certificate pinning trust list file. Then, you've applied the content of the file to your reference device from which you can run the Group Policy Management Console.

The next step consists of configuring a group policy object that includes the applied certificate pin rule settings, and deploy it in your environment.

Sign-in to the reference computer using domain administrator equivalent credentials.

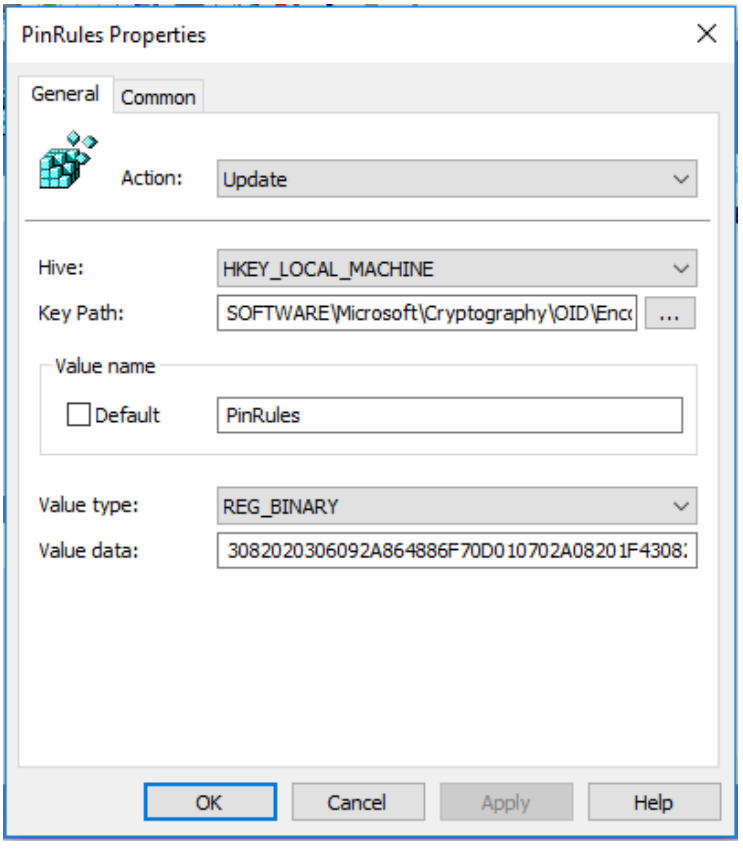
1. Start the **Group Policy Management Console** (gpmc.msc)

2. In the navigation pane, expand the forest node and then expand the domain node
3. Expand the node that contains your Active Directory's domain name
4. Select the **Group Policy objects** node. Right-click the **Group Policy objects** node and select **New**
5. In the **New GPO** dialog box, type *Enterprise Certificate Pinning Rules* in the **Name** text box and select **OK**
6. In the content pane, right-click the **Enterprise Certificate Pinning Rules** Group Policy object and select **Edit**
7. In the **Group Policy Management Editor**, in the navigation pane, expand the **Preferences** node under **Computer Configuration**. Expand **Windows Settings**
8. Right-click the **Registry** node and select **New**
9. In the **New Registry Properties** dialog box, select **Update** from the **Action** list. Select **HKEY_LOCAL_MACHINE** from the **Hive** list
10. For the **Key Path**, select **...** to launch the **Registry Item Browser**. Navigate to the following registry key and select the **PinRules** registry value name:

`HKLM\SOFTWARE\Microsoft\Cryptography\OID\EncodingType0\CertDllCreateCertificateChainEngine\Config`

Select **Select** to close the **Registry Item Browser**

1. The **Key Path** should contain the selected registry key. The **Value name** configuration should contain the registry value name ***PinRules***. **Value type** should read ***REG_BINARY*** and **Value data** should contain a long series of numbers from 0-9 and letters ranging from A-F (hexadecimal). Select **OK** to save your settings and close the dialog box



1. Close the **Group Policy Management Editor** to save your settings

2. Link the **Enterprise Certificate Pinning Rules** GPO to the OU containing the devices that you want to configure

Additional pin rules logging

To help constructing certificate pinning rules, you can configure the **PinRulesLogDir** setting under the certificate chain configuration registry key to include a parent directory to log pin rules.

Expand table

Name	Value
Key	HKLM\SOFTWARE\Microsoft\Cryptography\OID\EncodingType0\CertDllCreateCertificateChainEngine\Config
Name	PinRulesLogDir
Value	The Parent directory where Windows should write the additional pin rule logs
Data type	REG_SZ

Permission for the pin rule log folder

The folder in which Windows writes the additional pin rule logs must have permissions so that all users and applications have full access. You can run the following commands from an elevated command prompt to achieve the proper permissions.

Windows Command Prompt

```
set PinRulesLogDir=c:\PinRulesLog
mkdir %PinRulesLogDir%
icacls %PinRulesLogDir% /grant *S-1-15-2-1:(OI)(CI)(F)
icacls %PinRulesLogDir% /grant *S-1-1-0:(OI)(CI)(F)
icacls %PinRulesLogDir% /grant *S-1-5-12:(OI)(CI)(F)
icacls %PinRulesLogDir% /inheritance:e /setintegritylevel (OI)(CI)L
```

When an application verifies a TLS/SSL certificate chain that contains a server name matching a DNS name in the server certificate, Windows writes a .p7b file consisting of all the certificates in the server's chain to one of three child folders:

- `AdminPinRules`: Matched a site in the enterprise certificate pinning rules
- `AutoUpdatePinRules`: Matched a site in the certificate pinning rules managed by Microsoft
- `NoPinRules`: Didn't match any site in the certificate pin rules

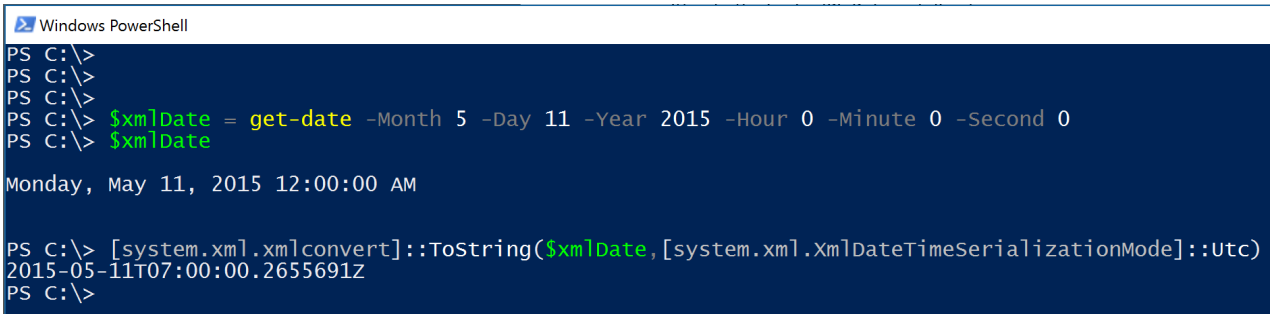
The output file name consists of the leading eight ASCII hex digits of the root's SHA1 thumbprint followed by the server name. For example:

- `D4DE20D0_xsi.outlook.com.p7b`
- `DE28F4A4_www.yammer.com.p7b`

If there's either an enterprise certificate pin rule or a Microsoft certificate pin rule mismatch, then Windows writes the .p7b file to the **MismatchPinRules** child folder. If the pin rules have expired, then Windows writes the .p7b to the **ExpiredPinRules** child folder.

Represent a date in XML

Many attributes within the pin rules xml file are dates.
These dates must be properly formatted and represented in UTC.
You can use Windows PowerShell to format these dates.
You can then copy and paste the output of the cmdlet into the XML file.



For simplicity, you can truncate decimal point (.) and the numbers after it. However, be certain to append the uppercase "Z" to the end of the XML date string.

Windows Command Prompt

```
2015-05-11T07:00:00.2655691Z
2015-05-11T07:00:00Z
```

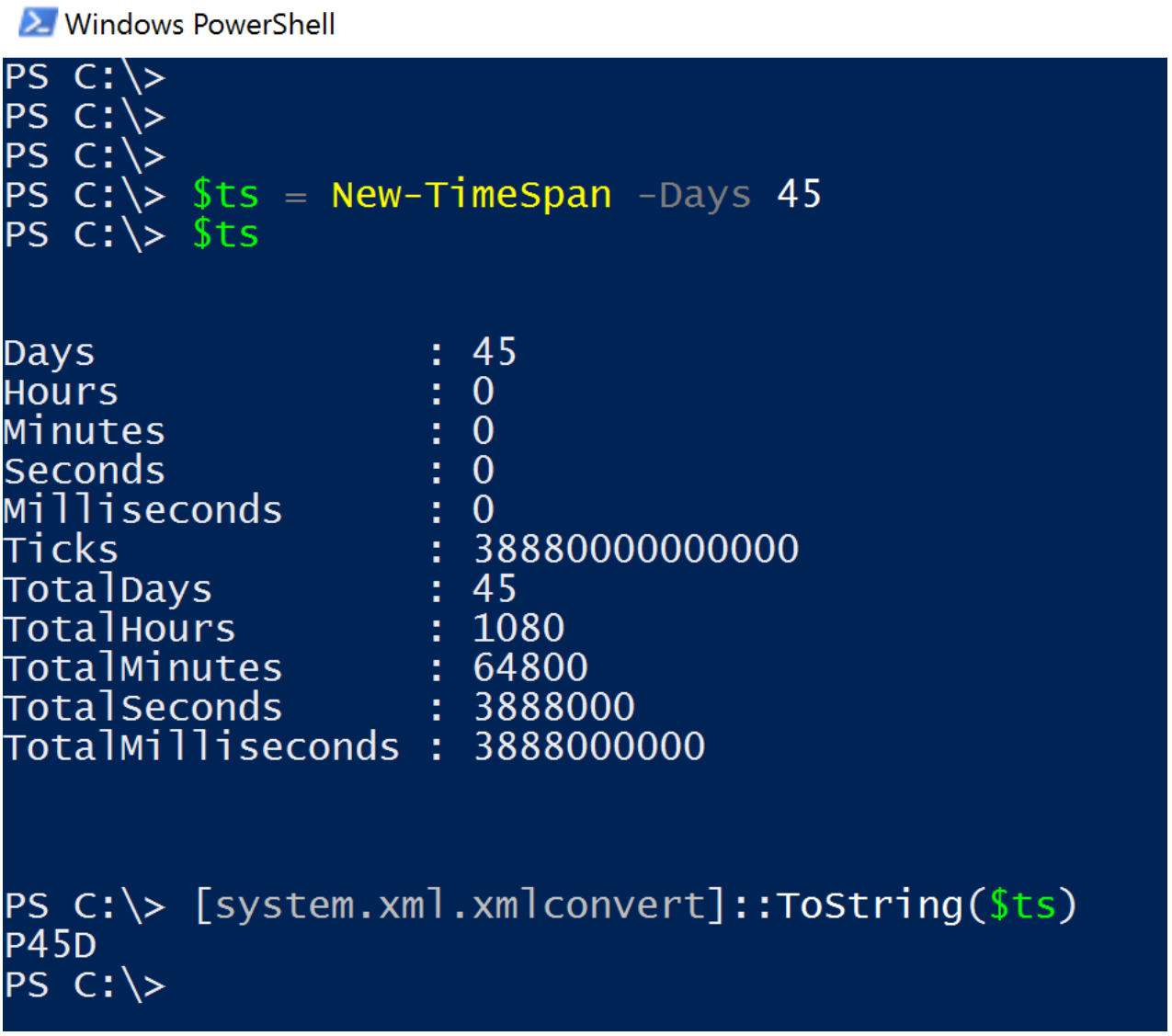
Convert an XML date

You can also use Windows PowerShell to validate and convert an XML date into a human readable date to validate it's the correct date.



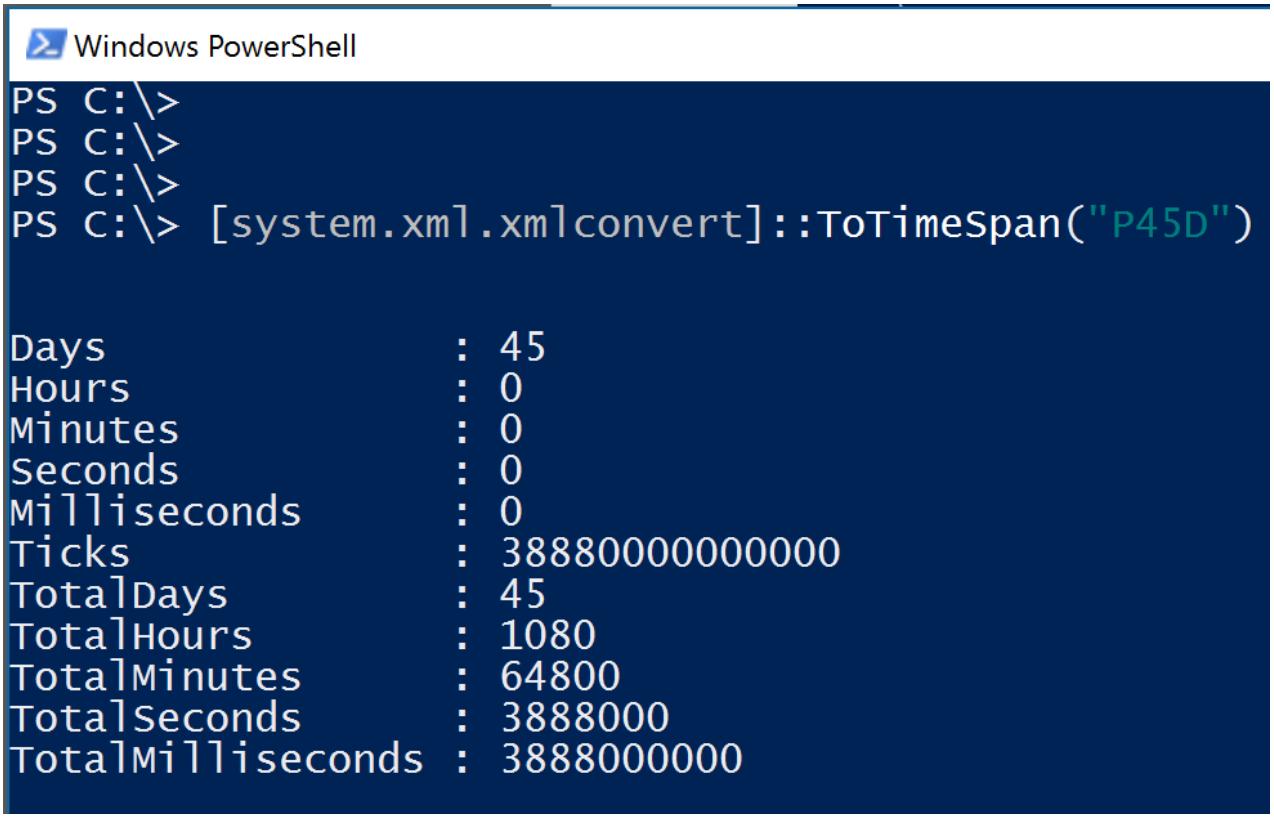
Represent a duration in XML

Some elements may be configured to use a duration rather than a date. You must represent the duration as an XML timespan data type. You can use Windows PowerShell to properly format and validate durations (timespans) and copy and paste them into your XML file.



Convert an XML duration

You can convert an XML formatted timespan into a timespan variable that you can read.



Certificate trust list XML schema definition (XSD)

XML

```
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="PinRules">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="PinRule" maxOccurs="unbounded" minOccurs="1">
          <xs:complexType>
            <xs:sequence>
              <xs:element name="Certificate" maxOccurs="unbounded" minOccurs="0">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:dateTime" name="EndDate"
use="optional"/>
                      <xs:attribute type="xs:string" name="File" use="optional"/>
                      <xs:attribute type="xs:string" name="Directory"
use="optional"/>
                      <xs:attribute type="xs:base64Binary" name="Base64"
use="optional"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
              <xs:element name="Site" maxOccurs="unbounded" minOccurs="1">
                <xs:complexType>
                  <xs:simpleContent>
                    <xs:extension base="xs:string">
                      <xs:attribute type="xs:string" name="Domain"/>
                      <xs:attribute type="xs:boolean" name="AllSubdomains"
```

```
use="optional" default="false"/>
                    </xs:extension>
                  </xs:simpleContent>
                </xs:complexType>
              </xs:element>
            </xs:sequence>
            <xs:attribute type="xs:string" name="Name"/>
            <xs:attribute name="Error" use="optional" default="None">
              <xs:simpleType>
                <xs:restriction base="xs:string">
                  <xs:enumeration value ="Revoked"/>
                  <xs:enumeration value ="InvalidName"/>
                  <xs:enumeration value ="None"/>
                </xs:restriction>
              </xs:simpleType>
            </xs:attribute>
            <xs:attribute type="xs:boolean" name="Log" use="optional"
default="true"/>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
      <xs:attribute type="xs:duration" name="Duration" use="optional"/>
      <xs:attribute type="xs:duration" name="LogDuration" use="optional"/>
      <xs:attribute type="xs:dateTime" name="NextUpdate" use="optional"/>
      <xs:attribute type="xs:dateTime" name="LogEndDate" use="optional"/>
      <xs:attribute type="xs:string" name="ListIdentifier" use="optional"/>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

Feedback

Was this page helpful? Yes No

Provide product feedback

Web sign-in for Windows

Applies to: ✅ Windows 11

Starting in Windows 11, version 22H2 with KB5030310, you can enable a web-based sign-in experience on Microsoft Entra joined devices. This feature is called *Web sign-in*, and it unlocks new sign-in options and capabilities.

Web sign-in is a *credential provider*, and it was initially introduced in Windows 10 with support for Temporary Access Pass (TAP) only. With the release of Windows 11, the supported scenarios and capabilities of Web sign-in are expanded.
For example, you can sign in with the Microsoft Authenticator app or with a SAML-P federated identity.

This article describes how to configure Web sign-in and the supported key scenarios.

System requirements

Here are the prerequisites for using Web sign-in:

- Windows 11, version 22H2 with 5030310, or later
- Microsoft Entra joined
- Internet connectivity, as the authentication is done over the Internet

> **Important**
>
> Web sign-in is not supported for Microsoft Entra hybrid joined or domain joined devices.

Windows edition and licensing requirements

The following table lists the Windows editions that support Web sign-in:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Web sign-in license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Configure web sign-in

To use web sign-in, your devices must be configured with different policies. Review the following instructions to configure your devices using either Microsoft Intune or a provisioning package (PPKG).

> **Note**
>
> Web sign-in uses a system-managed local account called *WsiAccount*. The account is created automatically when you enable Web sign-in, and it's not displayed in the user selection list. Every time a user uses the Web sign-in credential provider, the *WsiAccount* account is enabled. After the user signs in, the account is disabled.

Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Authentication	Enable Web Sign In	Enabled
Authentication	Configure Web Sign In Allowed Urls	This setting is optional, and it contains a list of domains required for sign in, for example: - `idp.example.com` - `example.com`
Authentication	Configure Webcam Access Domain Names	This setting is optional, and it should be configured if you need to use the webcam during the sign-in process. Specify the list of domains that are allowed to use the webcam during the sign-in process, for example: `example.com`

Assign the policy to a group that contains as members the devices or users that you want to configure.

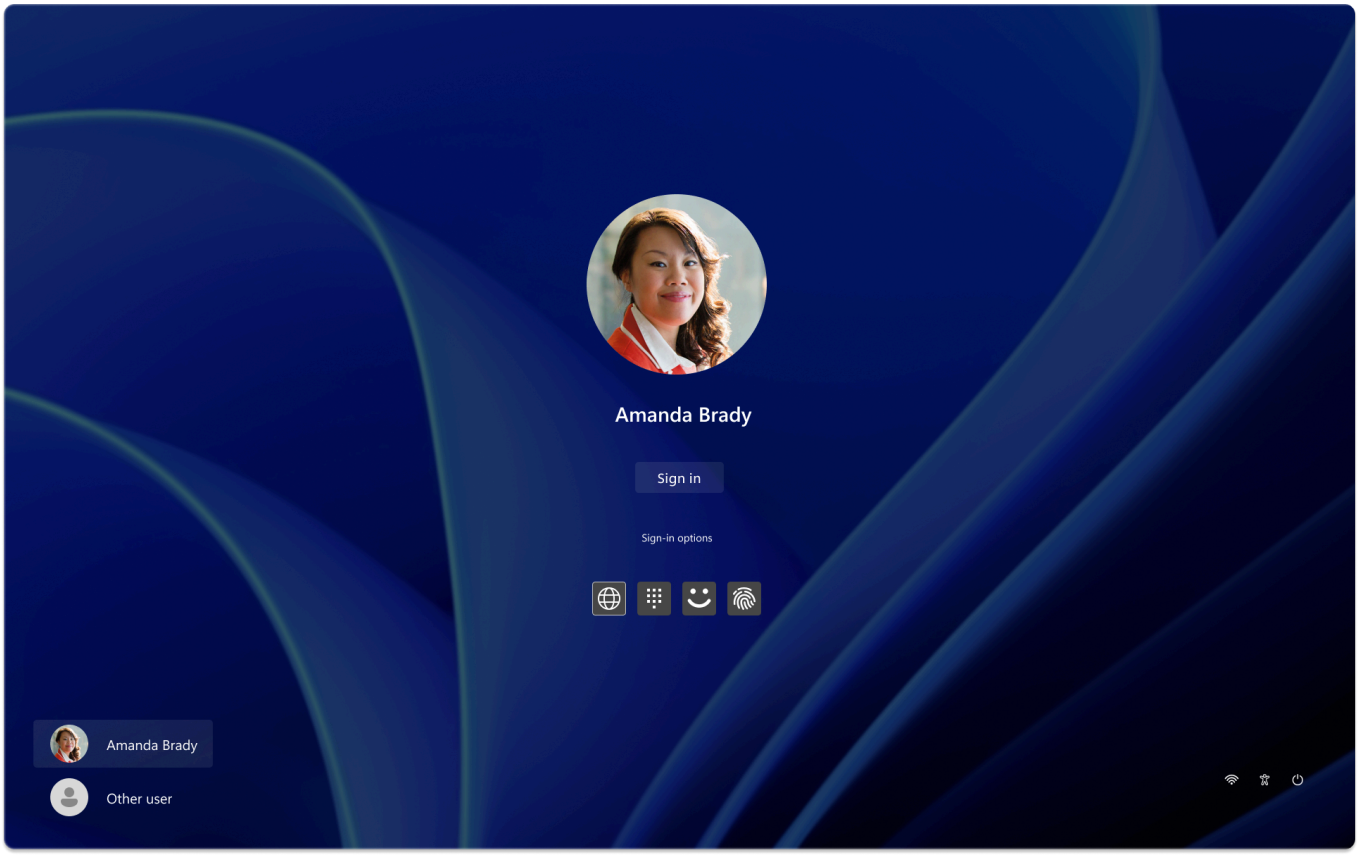
Alternatively, you can configure devices using a custom policy with the following settings:

Expand table

OMA-URI	More information
`./Vendor/MSFT/Policy/Config/Authentication/EnableWebSignIn`	EnableWebSignIn
`./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebSignInAllowedUrls`	ConfigureWebSignInAllowedUrls
`./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebCamAccessDomainNames`	ConfigureWebcamAccessDomainNames

User experiences

Once the devices are configured, a new sign-in experience becomes available, as indicated by the presence of the Web sign-in credential provider in the Windows lock screen.



Here's a list of key scenarios supported by Web sign-in, and a brief animation showing the user experience. Select the thumbnail to start the animation.

Passwordless sign-in

Users can sign in to Windows passwordless, even before enrolling in Windows Hello for Business. For example, by using the Microsoft Authenticator app as a sign-in method.

https://learn-video.azurefd.net/vod/player?id=974e445a-b78a-4555-86db-919473907535&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fweb-sign-in%2F

> **Tip**
>
> When used in conjuction with *Windows Hello for Business passwordless*, you can hide the password credential provider from the lock screen as well as in-session authentication scenarios. This enables a truly passwordless Windows experience.

To learn more:

- Enable passwordless sign-in with Microsoft Authenticator
- Passwordless authentication options for Microsoft Entra ID
- Windows passwordless experience

Windows Hello for Business PIN reset

The Windows Hello PIN reset flow is seamless and more robust than in previous versions.

https://learn-video.azurefd.net/vod/player?id=310f7665-6276-4ad8-b76e-429073c10972&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fweb-sign-in%2F

For more information, see PIN reset.

Temporary Access Pass (TAP)

A Temporary Access Pass (TAP) is a time-limited passcode granted by an administrator to a user. Users can sign in with a TAP using the Web sign-in credential provider. For example:

- to onboard Windows Hello for Business or a FIDO2 security key
- if lost or forgotten FIDO2 security key and unknown password

https://learn-video.azurefd.net/vod/player?id=8d80bef4-96a8-4467-8e67-e0637bdabcd8&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fweb-sign-in%2F

For more information, see Use a Temporary Access Pass.

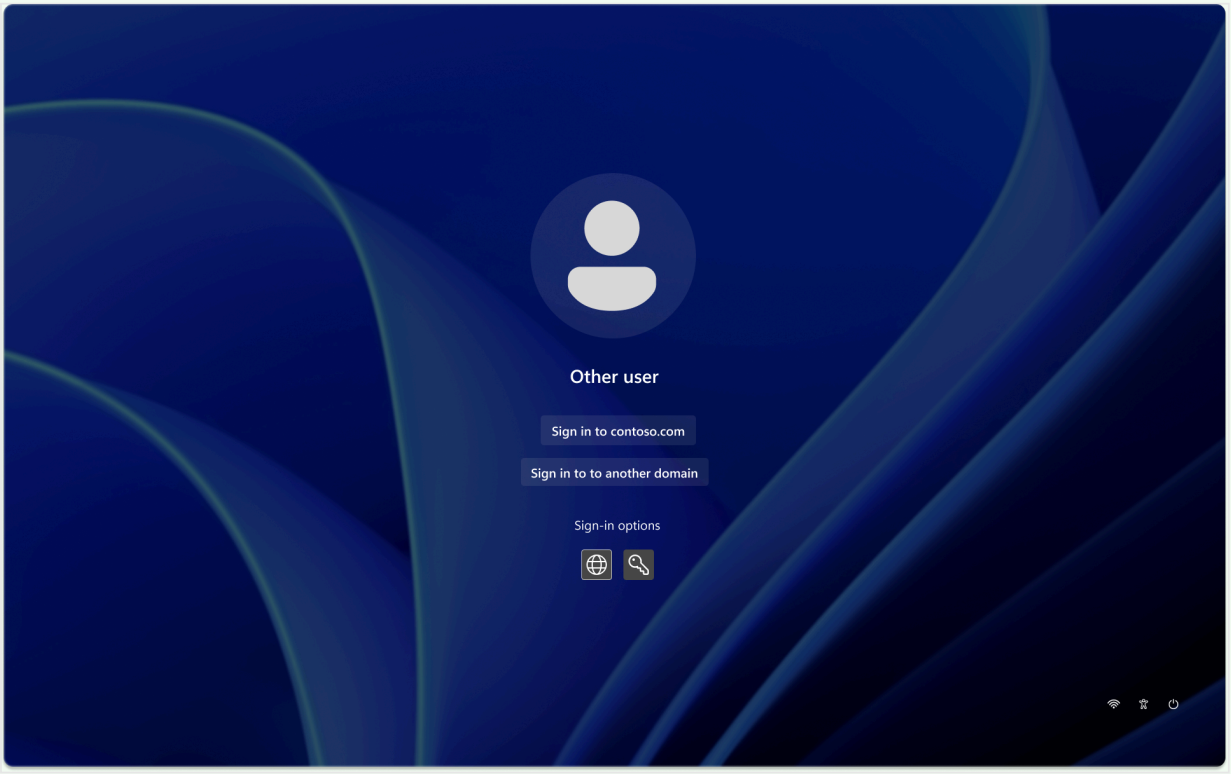
Federated authentication

If the Microsoft Entra tenant is federated with a non-Microsoft SAML-P identity provider (IdP), federated users can sign using the Web sign-in credential provider.

https://learn-video.azurefd.net/vod/player?id=88ad0efb-9031-428c-a3cf-612c47810ecf&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fweb-sign-in%2F

> **Tip**
>
> To improve the user experience for federated identities:
>
> - Enable Windows Hello for Business. Once the user signs in, the user can enroll in Windows Hello for Business and then use it to sign in to the device
> - Configure the *preferred Microsoft Entra tenant name* feature, which allows users to select the domain name during the sign-in process. The users are then automatically redirected to the

identity provider sign-in page



For more information about preferred tenant name, see Authentication CSP - PreferredAadTenantDomainName.

Important considerations

Here's a list of important considerations to keep in mind when configuring or using Web sign-in:

- Cached credentials aren't supported with Web sign-in. If the device is offline, the user can't use the Web sign-in credential provider to sign in
- After sign out, the user isn't displayed in the user selection list
- Once enabled, the Web sign-in credential provider is the default credential provider for new users signing in to the device. To change the default credential provider, you can use the DefaultCredentialProvider ADMX-backed policy
- The user can exit the Web sign-in flow by pressing `Ctrl` + `Alt` + `Delete` to get back to the Windows lock screen

Known issues

- If you attempt to sign in while the device is offline, you get the following message: *It doesn't look like you're connected to the Internet. Check your connection and try again*. Selecting the *Back to sign-in* option doesn't bring you back to the lock screen. As a workaround, you can press `Ctrl` + `Alt` + `Delete` to get back to the lock screen.

Provide feedback

To provide feedback for web sign-in, open **Feedback Hub** and use the category **Security and Privacy** > **Passwordless experience**.

Last updated on 03/29/2026

Configure federated sign-in for Windows devices

Article • 01/27/2025 • Applies to: ✅ Windows 11, ✅ Windows 11 SE

Starting in Windows 11 SE, version 22H2 and Windows 11 Pro Edu/Education, version 22H2 with KB5022913, you can enable your users to sign-in using a federated identity provider (IdP) via a web sign-in experience. Signing in with a federated identity can be a great way to simplify the sign-in process for your users: instead of having to remember a username and password defined in Microsoft Entra ID, they can sign-in using their existing credentials from the IdP. For example, students and educators can use QR code badges to sign-in.

Benefits of federated sign-in

A federated sign-in experience enables students to sign-in in less time, and with less friction. With fewer credentials to remember and a simplified sign-in process, students are more engaged and focused on learning.

There are two Windows features that enable a federated sign-in experience:

- *Federated sign-in*, which is designed for 1:1 student devices. For an optimal experience, you should not enable federated sign-in on shared devices
- *Web sign-in*, which provides a similar experience to *Federated sign-in*, and can be used for shared devices

> **Important**
>
> *Federated sign-in* and *Web sign-in* require different configurations, which are explained in this document.

Prerequisites

To enable a federated sign-in experience, the following prerequisites must be met:

1. A Microsoft Entra tenant, with one or multiple domains federated to a third-party IdP. For more information, see What is federation with Microsoft Entra ID? and Use a SAML 2.0 IdP for Single Sign On

> **Note**
>
> If your organization uses a third-party federation solution, you can configure single sign-on to Microsoft Entra ID if the solution is compatible with Microsoft Entra ID. For questions regarding compatibility, contact your identity provider. If you're an IdP, and would like to validate your solution for interoperability, refer to these **guidelines**.

- For a step-by-step guide on how to configure **Google Workspace** as an identity provider for Microsoft Entra ID, see Configure federation between Google Workspace and Microsoft Entra ID
- For a step-by-step guide on how to configure **Clever** as an identity provider for Microsoft Entra ID, see Setup guide for Badges into Windows and Microsoft Entra ID

2. Individual IdP accounts created: each user requires an account defined in the third-party IdP platform

3. Individual Microsoft Entra accounts created: each user requires a matching account defined in Microsoft Entra ID. These accounts are commonly created through automated solutions, for example:

   - School Data Sync (SDS)
   - Microsoft Entra Connect Sync for environment with on-premises AD DS
   - PowerShell scripts that call the Microsoft Graph API
   - provisioning tools offered by the IdP

   For more information about identity matching, see Identity matching in Microsoft Entra ID.

4. Licenses assigned to the Microsoft Entra user accounts. It's recommended to assign licenses to a dynamic group: when new users are provisioned in Microsoft Entra ID, the licenses are automatically assigned. For more information, see Assign licenses to users by group membership in Microsoft Entra ID

5. Enable Federated sign-in or Web sign-in on the Windows devices, depending if the devices are shared or assigned to a single student

To use Federated sign-in or Web sign-in, the devices must have Internet access. These features don't work without it, as the authentication is done over the Internet.

> **Important**

WS-Fed is the only supported federated protocol to join a device to Microsoft Entra ID. If you have a SAML 2.0 IdP, it's recommended to complete the Microsoft Entra join process using one of the following methods:

- Provisioning packages (PPKG)
- Windows Autopilot self-deploying mode

Windows edition and licensing requirements

The following table lists the Windows editions that support Federated sign-in:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
No	No	Yes	Yes

Federated sign-in license entitlements are granted by the following licenses:

Expand table

Windows Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	No	No	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Federated sign-in is supported on the following Windows editions and versions:

- Windows 11 SE, version 22H2 and later
- Windows 11 Pro Edu/Education, version 22H2 with KB5022913

Web sign-in is supported starting in Windows 11 SE/Pro Edu/Education, version 22H2 with KB5026446.

Configure a federated sign-in experience

You can configure a federated sign-in experience for student assigned (1:1) devices or student shared devices:

- When federated sign-in is configured for **student assigned (1:1) devices**, you use a Windows feature called *Federated sign-in*. The first user who signs in to the device

with a federated identity becomes the *primary user*. The primary user is always displayed in the bottom left corner of the sign-in screen
- When federated sign-in is configured for **student shared devices**, you use a Windows feature called *Web sign-in*. With Web sign-in there's no primary user, and the sign-in screen displays, by default, the last user who signed in to the device

The configuration is different for each scenario, and is described in the following sections.

Configure Federated sign-in for student assigned (1:1) devices

Review the following instructions to configure your devices using either Microsoft Intune or a provisioning package (PPKG).

Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Education	Is Education Environment	Enabled
Federated Authentication	Enable Web Sign In For Primary User	Enabled
Authentication	Configure Web Sign In Allowed Urls	Enter the list of domains, with each URL in a separate row. For example: - `samlidp.clever.com` - `clever.com` - `mobile-redirector.clever.com`
Authentication	Configure Webcam Access Domain Names	This setting is optional, and it should be configured if you need to use the webcam during the sign-in process. Specify the list of domains that are allowed to use the webcam during the sign-in process, separated by a semicolon. For example: `clever.com`

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the following settings:

Expand table

Setting
OMA-URI: `./Vendor/MSFT/Policy/Config/Education/IsEducationEnvironment` **Data type**: int **Value**: `1`
OMA-URI: `./Vendor/MSFT/Policy/Config/FederatedAuthentication/EnableWebSignInForPrimaryUser` **Data type**: int **Value**: `1`
OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebSignInAllowedUrls` **Data type**: String **Value**: Semicolon separated list of domains, for example: `samlidp.clever.com;clever.com;mobile-redirector.clever.com`
OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebCamAccessDomainNames`** **Data type**: String **Value**: This setting is optional, and it should be configured if you need to use the webcam during the sign-in process. Specify the list of domains that are allowed to use the webcam during the sign-in process, separated by a semicolon. For example: `clever.com`

Configure Web sign-in for student shared devices

Review the following instructions to configure your shared devices using either Microsoft Intune or a provisioning package (PPKG).

Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Education	Is Education Environment	Enabled

Category	Setting name	Value
SharedPC	Enable Shared PC Mode With OneDrive Sync	True
Authentication	Enable Web Sign In	Enabled
Authentication	Configure Web Sign In Allowed Urls	Enter the list of domains, with each URL in a separate row. For example: - `samlidp.clever.com` - `clever.com` - `mobile-redirector.clever.com`
Authentication	Configure Webcam Access Domain Names	This setting is optional, and it should be configured if you need to use the webcam during the sign-in process. Specify the list of domains that are allowed to use the webcam during the sign-in process, separated by a semicolon. For example: `clever.com`

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the following settings:

Expand table

Setting
OMA-URI: `./Vendor/MSFT/Policy/Config/Education/IsEducationEnvironment` **Data type**: int **Value**: `1`
OMA-URI: `./Vendor/MSFT/SharedPC/EnableSharedPCModeWithOneDriveSync` **Data type**: Boolean **Value**: True
OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/EnableWebSignIn` **Data type**: Integer **Value**: `1`
OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebSignInAllowedUrls` **Data type**: String **Value**: Semicolon separated list of domains, for example: `samlidp.clever.com;clever.com;mobile-redirector.clever.com`

Setting
OMA-URI: `./Vendor/MSFT/Policy/Config/Authentication/ConfigureWebCamAccessDomainNames` **Data type**: String **Value**: This setting is optional, and it should be configured if you need to use the webcam during the sign-in process. Specify the list of domains that are allowed to use the webcam during the sign-in process, separated by a semicolon. For example: `clever.com`

How to use federated sign-in

Once the devices are configured, a new sign-in experience becomes available.

As users enter their username, they're redirected to the identity provider sign-in page. Once the Idp authenticates the users, they're signed-in. In the following animation, you can observe how the first sign-in process works for a student assigned (1:1) device:



> **ⓘ Important**
>
> For student assigned (1:1) devices, once the policy is enabled, the first user who sign-in to the device will also set the disambiguation page to the identity provider domain on the device. This means that the device will be defaulting to that IdP. The user can exit the Federated sign-in flow by pressing `Ctrl`+`Alt`+`Delete` to get back to the standard Windows sign-in screen. The behavior is different for student shared devices, where the disambiguation page is always shown, unless preferred Microsoft Entra tenant name is configured.

Important considerations

Known issues affecting student assigned (1:1) devices

Federated sign-in for student assigned (1:1) devices doesn't work with the following settings enabled:

- **EnableSharedPCMode** or **EnableSharedPCModeWithOneDriveSync**, which are part of the SharedPC CSP
- **Interactive logon: do not display last signed in**, which is a security policy part of the Policy CSP
- **Take a Test** in kiosk mode, since it uses the security policy above

Known issues affecting student shared devices

The following issues are known to affect student shared devices:

- Non-federated users can't sign-in to the devices, including local accounts
- **Take a Test** in kiosk mode, since it uses a local guest account to sign in

Account management

For student shared devices, it's recommended to configure the account management policies to automatically delete the user profiles after a certain period of inactivity or disk levels. For more information, see Set up a shared or guest Windows device.

Preferred Microsoft Entra tenant name

To improve the user experience, you can configure the *preferred Microsoft Entra tenant name* feature.
When using preferred Microsoft Entra tenant name, the users bypass the disambiguation page and are redirected to the identity provider sign-in page. This configuration can be especially useful for student shared devices, where the disambiguation page is always shown.

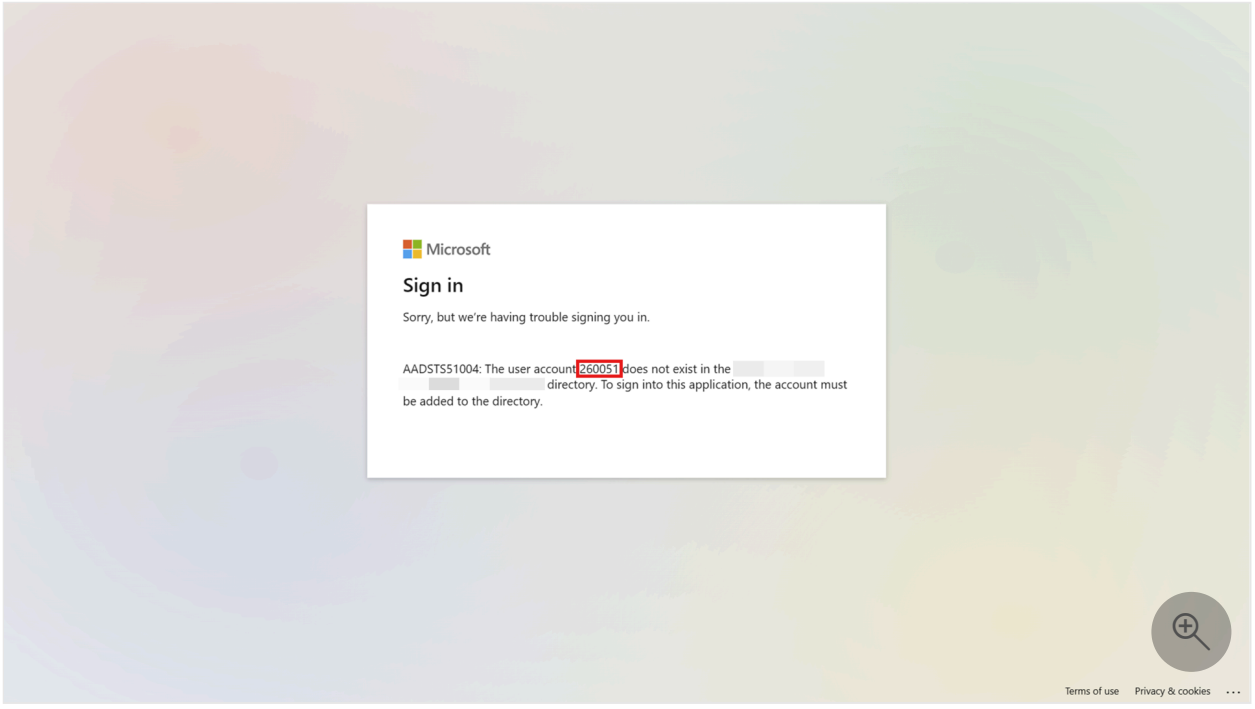
For more information about preferred tenant name, see Authentication CSP - PreferredAadTenantDomainName.

Identity matching in Microsoft Entra ID

When a Microsoft Entra user is federated, the user's identity from the IdP must match an existing user object in Microsoft Entra ID. After the token sent by the IdP is validated, Microsoft Entra ID searches for a matching user object in the tenant by using an attribute called *ImmutableId*.

> **Note**
>
> The ImmutableId is a string value that **must be unique** for each user in the tenant, and it shouldn't change over time. For example, the ImmutableId could be the student ID or SIS ID. The ImmutableId value should be based on the federation setup and configuration with your IdP, so confirm with your IdP before setting it.

If the matching object is found, the user is signed-in. Otherwise, the user is presented with an error message. The following picture shows that a user with the ImmutableId *260051* can't be found:



> **Important**
>
> The ImmutableId matching is case-sensitive.

The ImmutableId is typically configured when the user is created in Microsoft Entra ID, but it can also be updated later.
In a scenario where a user is federated and you want to change the ImmutableId, you must:

1. Convert the federated user to a cloud-only user (update the UPN to a non-federated domain)
2. Update the ImmutableId
3. Convert the user back to a federated user

Here's a PowerShell example to update the ImmutableId for a federated user:

PowerShell

```
Set-ExecutionPolicy -ExecutionPolicy RemoteSigned -Scope CurrentUser -Force
Install-Module Microsoft.Graph -Scope CurrentUser
Import-Module Microsoft.Graph
Connect-MgGraph -Scopes 'User.Read.All', 'User.ReadWrite.All'

#1. Convert the user from federated to cloud-only
Update-MgUser -UserId alton@example.com -UserPrincipalName
alton@example.onmicrosoft.com

#2. Convert the user back to federated, while setting the immutableId
Update-MgUser -UserId alton@example.onmicrosoft.com -UserPrincipalName
alton@example.com -OnPremisesImmutableId '260051'
```

Troubleshooting

- The user can exit the federated sign-in flow by pressing `Ctrl`+`Alt`+`Delete` to get back to the standard Windows sign-in screen
- Select the *Other User* button, and the standard username/password credentials are available to log into the device

---

Feedback

Was this page helpful? Yes No

Provide product feedback

Configure added LSA protection

Article • 03/26/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article explains how to configure added protection for the Local Security Authority (LSA) process to prevent code injection that can compromise credentials.

The LSA, which includes the Local Security Authority Server Service (LSASS) process, validates users for local and remote sign-ins and enforces local security policies. On Windows 8.1 and later, added protection for the LSA is provided to prevent nonprotected processes from reading memory and injecting code. This feature provides added security for the credentials that LSA stores and manages. You can achieve further protection when you use Unified Extensible Firmware Interface (UEFI) lock and Secure Boot. When these settings are enabled, disabling the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa** registry key has no effect.

Protected process requirements for plug-ins or drivers

For an LSA plug-in or driver to successfully load as a protected process, it must meet the criteria in the following two sections.

Signature verification

Protected mode requires any plug-in that's loaded into the LSA to be digitally signed with a Microsoft signature. Any plug-ins that are unsigned or aren't signed with a Microsoft signature fail to load in LSA. Examples of plug-ins are smart card drivers, cryptographic plug-ins, and password filters.

- LSA plug-ins that are drivers, such as smart card drivers, need to be signed by using Windows Hardware Quality Labs (WHQL) certification. For more information, see WHQL release signature.
- LSA plug-ins that don't have a WHQL certification process must be signed by using the file signing service for LSA.

Adherence to Microsoft Security Development Lifecycle (SDL) process guidance

- All plug-ins must conform to the applicable SDL process guidance. For more information, see Microsoft Security Development Lifecycle (SDL) – Process Guidance.
- Even if the plug-ins are properly signed with a Microsoft signature, noncompliance with the SDL process can result in failure to load a plug-in.

Recommended practices

Use the following list to thoroughly test enabling LSA protection before you broadly deploy the feature:

- Identify all the LSA plug-ins and drivers that your organization uses. Include non-Microsoft drivers or plug-ins, such as smart card drivers and cryptographic plug-ins, and any internally developed software that's used to enforce password filters or password change notifications.
- Ensure that all the LSA plug-ins are digitally signed with a Microsoft certificate so they don't fail to load under LSA protection.
- Ensure that all the correctly signed plug-ins can successfully load into LSA and that they perform as expected.
- Use the audit logs to identify any LSA plug-ins and drivers that fail to run as a protected process.

Limitations of enabling LSA protection

If added LSA protection is enabled, you can't debug a custom LSA plug-in. You can't attach a debugger to LSASS when it's a protected process. In general, there's no supported way to debug a running protected process.

Audit for LSA plug-ins and drivers that don't load as a protected process

Before you enable LSA protection, use audit mode to identify LSA plug-ins and drivers that fail to load in LSA protected mode. While in audit mode, the system generates event logs that identify all the plug-ins and drivers that fail to load under LSA if LSA protection is enabled. The messages are logged without actually blocking the plug-ins or drivers.

The events described in this section are recorded in Event Viewer in the **Operational** log under **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity**. These events can help you identify LSA plug-ins and drivers that fail to load due to signing reasons. To manage these events, you can use the **wevtutil** command-line tool. For information about this tool, see Wevtutil.

> **Important**
>
> Audit events aren't generated if Smart App Control is enabled on a device. To check or change the status of Smart App Control, open the Windows Security application and go to the **App & browser control** page. Select **Smart App Control settings** to check whether Smart App Control is enabled. If you want to audit added LSA protection, change the configuration to **Off**.

> **Note**

Audit mode for added LSA protection is enabled by default on devices running Windows 11 version 22H2 and later. If your device is running this build or later, no other actions are needed to audit added LSA protection.

Enable audit mode for LSASS.exe on a single computer

1. Open the Registry Editor, or enter **RegEdit.exe** in the **Run** dialog, and then go to the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe** registry key.
2. Open the **AuditLevel** value. Set its data type to **dword** and its data value to **00000008**.
3. Restart the computer.

After you take these steps, look for events that have the following IDs: 3065 and 3066. To check for these events, open Event Viewer, and then expand **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity** > **Operational**.

- **Event 3065** occurs when a code integrity check determines that a process, usually LSASS.exe, attempts to load a driver that doesn't meet the security requirements for shared sections. However, due to the system policy currently set, the image is allowed to load.
- **Event 3066** occurs when a code integrity check determines that a process, usually LSASS.exe, attempts to load a driver that doesn't meet the Microsoft signing level requirements. However, due to the system policy currently set, the image is allowed to load.

If a plug-in or driver contains shared sections, event 3066 is logged with event 3065. Removing the shared sections should prevent both events from occurring unless the plug-in doesn't meet the Microsoft signing level requirements.

Important

These operational events aren't generated when a kernel debugger is attached and enabled on a system.

Enable audit mode for LSASS.exe on multiple computers

To enable audit mode for multiple computers in a domain, you can use the registry client-side extension for Group Policy to deploy the LSASS.exe audit-level registry value. You need to modify the **HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe** registry key.

1. Open the Group Policy Management Console by entering **gpmc.msc** in the **Run** dialog or selecting **Group Policy Management Console** from the **Start** menu.
2. Create a new Group Policy Object (GPO) that's linked at the domain level or linked to the organizational unit that contains your computer accounts. Or select a GPO that's already deployed.
3. Right-click the GPO, and then select **Edit** to open the Group Policy Management Editor.

4. Expand **Computer Configuration** > **Preferences** > **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then select **Registry Item**. The **New Registry Properties** dialog appears.
6. In the **New Registry Properties** dialog, select or enter the following values:

    - For **Hive**, select **HKEY_LOCAL_MACHINE.**
    - For **Key Path**, select **SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\LSASS.exe**.
    - For **Value name**, enter **AuditLevel**.
    - For **Value type**, select **REG_DWORD**.
    - For **Value data**, enter **00000008**.

7. Select **OK**.

> ⓘ **Note**
>
> For the GPO to take effect, the GPO change must be replicated to all domain controllers in the domain.

To opt in for added LSA protection on multiple computers, you can use the registry client-side extension for Group Policy to modify **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa**. For instructions, see Enable and configure added LSA credentials protection later in this article.

Identify plug-ins and drivers that LSASS.exe fails to load

When LSA protection is enabled, the system generates event logs that identify all the plug-ins and drivers that fail to load under LSA. After you opt in to added LSA protection, you can use the event log to identify LSA plug-ins and drivers that fail to load in LSA protection mode.

Check for the following events in Event Viewer by expanding **Applications and Services Logs** > **Microsoft** > **Windows** > **CodeIntegrity** > **Operational**:

- **Event 3033** occurs when a code integrity check determines that a process, usually LSASS.exe, attempts to load a driver that doesn't meet the Microsoft signing level requirements.
- **Event 3063** occurs when a code integrity check determines that a process, usually LSASS.exe, attempts to load a driver that doesn't meet the security requirements for shared sections.

Shared sections typically result when programming techniques allow instance data to interact with other processes that use the same security context. Shared sections can create security vulnerabilities.

Enable and configure added LSA credentials protection

You can configure added LSA protection for devices running Windows 8.1 or later, or Windows Server 2012 R2 or later, by using the procedures in this section.

Devices that use Secure Boot and UEFI

When you enable LSA protection on x86-based or x64-based devices that use Secure Boot or UEFI, you can store a UEFI variable in the UEFI firmware by using a registry key or policy. When enabled with UEFI lock, LSASS runs as a protected process, and this setting is stored in a UEFI variable in the firmware.

When the setting is stored in the firmware, the UEFI variable can't be deleted or changed to configure added LSA protection by modifying the registry or by policy. The UEFI variable must be reset by using the instructions in Remove the LSA protection UEFI variable.

When enabled without a UEFI lock, LSASS runs as a protected process, and this setting isn't stored in a UEFI variable. This setting is applied by default on devices with a new installation of Windows 11 version 22H2 or later.

On x86-based or x64-based devices that don't support UEFI or where Secure Boot is disabled, you can't store the configuration for LSA protection in the firmware. These devices rely solely on the presence of the registry key. In this scenario, it's possible to disable LSA protection by using remote access to the device. Disablement of LSA protection doesn't take effect until the device reboots.

Automatic enablement

For client devices running Windows 11 version 22H2 and later, added LSA protection is enabled by default if the following criteria are met:

- The device is a new installation of Windows 11 version 22H2 or later, not upgraded from a previous release.
- The device is enterprise joined (Active Directory domain joined, Microsoft Entra domain joined, or hybrid Microsoft Entra domain joined).
- The device is capable of hypervisor-protected code integrity (HVCI).

Automatic enablement of added LSA protection on Windows 11 version 22H2 and later doesn't set a UEFI variable for the feature. If you want to set a UEFI variable, you can use a registry configuration or policy.

Enable LSA protection on a single computer

You can enable LSA protection on a single computer by using the registry or by using Local Group Policy.

Enable by using the registry

1. Open the Registry Editor, or enter **RegEdit.exe** in the **Run** dialog, and then go to the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa** registry key.
2. Open the **RunAsPPL** value, and edit its data:

    - To configure the feature with a UEFI variable, use a type of **dword** and a data value of **00000001**.
    - To configure the feature without a UEFI variable, use a type of **dword** and a data value of **00000002**. This value is only enforced on Windows 11 build 22H2 and later.

3. Restart the computer.

Enable by using Local Group Policy on Windows 11 version 22H2 and later

1. Open the Local Group Policy Editor by entering **gpedit.msc** in the **Run** dialog.
2. Expand **Computer Configuration** > **Administrative Templates** > **System** > **Local Security Authority**.
3. Open the **Configures LSASS to run as a protected process** policy.
4. Set the policy to **Enabled**.
5. Under **Options**, select one of the following options:

    - To configure the feature with a UEFI variable, select **Enabled with UEFI Lock**.
    - To configure the feature without a UEFI variable, select **Enabled without UEFI Lock**.

6. Select **OK**.
7. Restart the computer.

Enable LSA protection by using Group Policy

1. Open the Group Policy Management Console by entering **gpmc.msc** in the **Run** dialog or selecting **Group Policy Management Console** from the **Start** menu.
2. Create a new GPO that's linked at the domain level or linked to the organizational unit that contains your computer accounts. Or select a GPO that's already deployed.
3. Right-click the GPO, and then select **Edit** to open the Group Policy Management Editor.
4. Expand **Computer Configuration** > **Preferences** > **Windows Settings**.
5. Right-click **Registry**, point to **New**, and then select **Registry Item**. The **New Registry Properties** dialog appears.
6. In the **New Registry Properties** dialog, select or enter the following values:

    - For **Hive**, select **HKEY_LOCAL_MACHINE.**
    - For **Key Path**, select **SYSTEM\CurrentControlSet\Control\Lsa**.
    - For **Value name**, enter **RunAsPPL**.
    - For **Value type**, select **REG_DWORD**.
    - For **Value data**, enter one of the following values:
        - To enable LSA protection with a UEFI variable, enter **00000001**.

    - To enable LSA protection without a UEFI variable, enter **00000002**. This setting is only enforced on Windows 11 version 22H2 and later.
7. Select **OK**.

Enable LSA protection by creating a custom device configuration profile

For devices running Windows 11 version 22H2 and later, you can take the steps in the following sections to enable and configure LSA protection. This procedure uses the Microsoft Intune admin center to create a custom device configuration profile.

Create a profile

1. In the Intune admin center, go to **Devices** > **Windows** > **Configuration profiles**, and then select **Create profile**.
2. On the **Create a profile** screen, select the following options:

    - Under **Platform**, select **Windows 10 and later**.
    - Under **Profile type**, select **Templates**, and then select **Custom**.

3. Select **Create**.
4. On the **Basics** screen, enter a name and optional description for the profile, and then select **Next**.

Add initial configuration settings

1. On the **Configuration settings** screen, select **Add**.
2. On the **Add row** screen, enter the following information:

    - For **Name**, enter a name for the Open Mobile Alliance – Uniform Resource (OMA-URI) setting.
    - For **OMA-URI**, enter **./Device/Vendor/MSFT/Policy/Config/LocalSecurityAuthority/ConfigureLsaProtectedProcess**.
    - For **Data type**, select **Integer**.
    - For **Value**, enter one of the following values:
        - To configure LSASS to run as a protected process with UEFI lock, enter **1**.
        - To configure LSASS to run as a protected process without UEFI lock, enter **2**.

3. Select **Save**, and then select **Next**.

Finish configuring the profile

1. On the **Assignments** page, configure the assignments, and then select **Next**.
2. On the **Applicability Rules** page, configure any applicability rules, and then select **Next**.
3. On the **Review + create** page, verify the configuration, and then select **Create**.

4. Restart the computer.

For more information about this policy configuration service provider (CSP), see LocalSecurityAuthority - ConfigureLsaProtectedProcess.

Disable LSA protection

You can disable LSA protection by using the registry or by using Local Group Policy. If the device is using Secure Boot and you set the LSA protection UEFI variable in the firmware, you can use a tool to remove the UEFI variable.

Disable by using the registry

1. Open the Registry Editor, or enter **RegEdit.exe** in the **Run** dialog, and then go to the **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa** registry key.
2. Open the **RunAsPPL** value, and set its data value to **00000000**. Or delete the **RunAsPPL** value.
3. If the protected processes light (PPL) feature was enabled with a UEFI variable, use the Local Security Authority Protected Process Opt-out tool to remove the UEFI variable.
4. Restart the computer.

Disable by using local policy on Windows 11 version 22H2 and later

1. Open the Local Group Policy Editor by entering **gpedit.msc** in the **Run** dialog.
2. Expand **Computer Configuration** > **Administrative Templates** > **System** > **Local Security Authority**.
3. Open the **Configures LSASS to run as a protected process** policy.
4. Set the policy to **Enabled**.
5. Under **Options**, select **Disabled**.
6. Select **OK**.
7. Restart the computer.

> **Note**
>
> If you set this policy to **Not Configured** and the policy was previously enabled, the prior setting doesn't get cleaned up and continues to be enforced. You must set the policy to **Disabled** under the **Options** dropdown to disable the feature.

Remove the LSA protection UEFI variable

You can use the Local Security Authority (LSA) Protected Process Opt-out tool from the Microsoft Download Center to delete the UEFI variable if the device is using Secure Boot.

> **Note**
>
> The Download Center offers two files named *LsaPplConfig.efi*. The smaller file is for x86-based systems and the larger file is for x64-based systems.

For more information about managing Secure Boot, see UEFI Firmware.

> **Caution**
>
> When Secure Boot is turned off, all the Secure Boot and UEFI-related configurations are reset. You should turn off Secure Boot only when all other means to disable LSA protection fail.

Verify LSA protection

To determine whether LSA starts in protected mode when Windows starts, take the following steps:

1. Open Event Viewer.
2. Expand **Windows Logs** > **System**.
3. Look for the following **WinInit** event: **12: LSASS.exe was started as a protected process with level: 4**.

LSA and Credential Guard

LSA protection is a security feature that defends sensitive information like credentials from theft by blocking untrusted LSA code injection and process memory dumping. LSA protection runs in the background by isolating the LSA process in a container and preventing other processes, like malicious actors or apps, from accessing the feature. This isolation makes LSA protection a vital security feature, which is why it's enabled by default in Windows 11.

Starting in Windows 10, Credential Guard also helps prevent credential theft attacks by protecting NTLM password hashes, Kerberos ticket-granting tickets (TGTs), and credentials stored by applications as domain credentials. Kerberos, NTLM, and Credential Manager isolate secrets by using virtualization-based security (VBS).

When Credential Guard is enabled, the LSA process communicates with a component called the isolated LSA process, or LSAIso.exe, that stores and protects secrets. Data stored by the isolated LSA process is protected by using VBS and isn't accessible to the rest of the operating system. LSA uses remote procedure calls to communicate with the isolated LSA process.

Starting in Windows 11 version 22H2, VBS and Credential Guard are enabled by default on all devices that meet the system requirements. Credential Guard is supported on 64-bit Secure Boot devices only. LSA protection and Credential Guard are complementary, and systems that support Credential Guard or enable it by default can also enable and benefit from LSA protection. For more information about Credential Guard, see Credential Guard overview.

More resources

- Credentials protection and management
- Partner Center for Windows Hardware

Feedback

Was this page helpful? Yes No

Credential Guard overview

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Credential Guard prevents credential theft attacks by protecting NTLM password hashes, Kerberos Ticket Granting Tickets (TGTs), and credentials stored by applications as domain credentials.

Credential Guard uses Virtualization-based security (VBS) to isolate secrets so that only privileged system software can access them. Unauthorized access to these secrets can lead to credential theft attacks like *pass the hash* and *pass the ticket*.

When enabled, Credential Guard provides the following benefits:

- **Hardware security**: NTLM, Kerberos, and Credential Manager take advantage of platform security features, including Secure Boot and virtualization, to protect credentials
- **Virtualization-based security**: NTLM, Kerberos derived credentials, and other secrets run in a protected environment that is isolated from the running operating system
- **Protection against advanced persistent threats**: when credentials are protected using VBS, the credential theft attack techniques and tools used in many targeted attacks are blocked. Malware running in the operating system with administrative privileges can't extract secrets that are protected by VBS

> ⓘ **Note**
>
> While Credential Guard is a powerful mitigation, persistent threat attacks will likely shift to new attack techniques, and you should also incorporate other security strategies and architectures.

Default enablement

Starting in **Windows 11, 22H2** and **Windows Server 2025**, VBS and Credential Guard are enabled by default on devices that meet the requirements.

The default enablement is **without UEFI Lock**, thus allowing administrators to disable Credential Guard remotely if needed.

When Credential Guard is enabled, VBS is automatically enabled too.

> ⓘ **Note**

If Credential Guard is explicitly **disabled** *before* a device is updated to Windows 11, version 22H2 / Windows Server 2025 or later, default enablement does not overwrite the existing settings. That device will continue to have Credential Guard disabled even after updating to a version of Windows that enables Credential Guard by default.

Default enablement on Windows

Devices running Windows 11, 22H2 or later have Credential Guard enabled by default if they:

- Meet the license requirements
- Meet the hardware and software requirements
- Aren't explicitly configured to disable Credential Guard

> **Note**
>
> Devices running Windows 11 Pro/Pro Edu 22H2 or later may have Virtualization-based Security (VBS) and/or Credential Guard automatically enabled if they meet the other requirements for default enablement, and have previously run Credential Guard. For example if Credential Guard was enabled on an Enterprise device that later downgraded to Pro.
>
> > To determine whether the Pro device is in this state, check if the following registry key exists: `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0\IsolatedCredentialsRootSecret`. If you wish to disable Credential Guard, see **configure Credential Guard**.

Default enablement on Windows Server

Devices running Windows Server 2025 or later have Credential Guard enabled by default if they:

- Meet the license requirements
- Meet the hardware and software requirements
- Aren't explicitly configured to disable Credential Guard
- Are joined to a domain
- Aren't a domain controller

> **Important**
>
> For information about known issues related to default enablement, see Credential Guard: known issues.

System requirements

For Credential Guard to provide protection, the device must meet certain hardware, firmware, and software requirements.

Devices that exceed the minimum hardware and firmware qualifications receive additional protections and are more hardened against certain threats.

Hardware and software requirements

Credential Guard requires the features:

- Virtualization-based security (VBS)

  > **Note**
  >
  > VBS has different requirements to enable it on different hardware platforms. For more information, see Virtualization-based Security requirements

- Secure Boot

While not required, the following features are recommended to provide additional protections:

- Trusted Platform Module (TPM), as it provides binding to hardware. TPM versions 1.2 and 2.0 are supported, either discrete or firmware
- UEFI lock, as it prevents attackers from disabling Credential Guard with a registry key change

For detailed information on protections for improved security that are associated with hardware and firmware options, see additional security qualifications.

Credential Guard in virtual machines

Credential Guard can protect secrets in Hyper-V virtual machines, just as it would on a physical machine. When Credential Guard is enabled on a VM, secrets are protected from attacks *inside*

the VM. Credential Guard doesn't provide protection from privileged system attacks originating from the host.

The requirements to run Credential Guard in Hyper-V virtual machines are:

- The Hyper-V host must have an IOMMU
- The Hyper-V virtual machine must be generation 2

> **Note**
>
> Credential Guard is not supported on Hyper-V or Azure generation 1 VMs. Credential Guard is available on generation 2 VMs only.

Windows edition and licensing requirements

The following table lists the Windows editions that support Credential Guard:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
No	Yes	No	Yes

Credential Guard license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
No	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Application requirements

When Credential Guard is enabled, certain authentication capabilities are blocked. Applications that require such capabilities break. We refer to these requirements as *application requirements*.

Applications should be tested before deployment to ensure compatibility with the reduced functionality.

> ⚠ **Warning**
>
> Enabling Credential Guard on domain controllers isn't recommended. Credential Guard doesn't provide any added security to domain controllers, and can cause application compatibility issues on domain controllers.
>
> Enabling Credential Guard on Exchange Server isn't supported and can lead to performance issues.

> ⓘ **Note**
>
> Credential Guard doesn't provide protections for the Active Directory database or the Security Accounts Manager (SAM). The credentials protected by Kerberos and NTLM when Credential Guard is enabled are also in the Active Directory database (on domain controllers) and the SAM (for local accounts).

Applications break if they require:

- Kerberos DES encryption support
- Kerberos unconstrained delegation
- Kerberos TGT extraction
- NTLMv1

Applications ask and expose credentials to risk if they require:

- Digest authentication
- Credential delegation
- MS-CHAPv2
- CredSSP

Applications might cause performance issues when they attempt to hook the isolated Credential Guard process `LSAIso.exe`.

Services or protocols that rely on Kerberos, such as file shares or remote desktop, continue to work and aren't affected by Credential Guard.

Next steps

- Learn how Credential Guard works
- Learn how to configure Credential Guard

- Review the advice and sample code for making your environment more secure and robust with Credential Guard in the Additional mitigations article
- Review considerations and known issues when using Credential Guard

---

Last updated on 03/29/2026

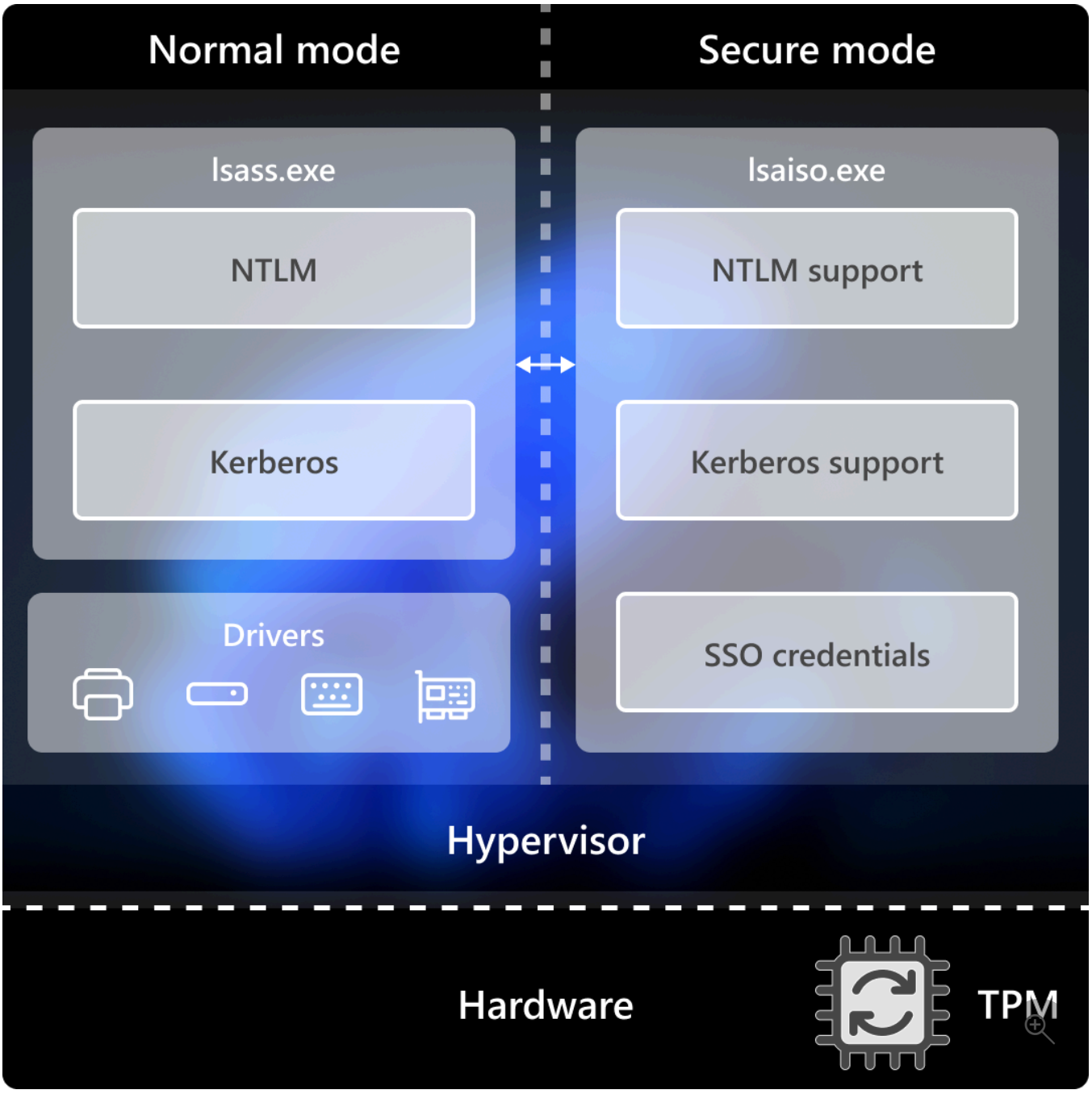
How Credential Guard works

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Kerberos, NTLM, and Credential Manager isolate secrets by using Virtualization-based security (VBS). Previous versions of Windows stored secrets in its process memory, in the Local Security Authority (LSA) process `lsass.exe`.

With Credential Guard enabled, the LSA process in the operating system talks to a component called the *isolated LSA process* that stores and protects those secrets, `LSAIso.exe`. Data stored by the isolated LSA process is protected using VBS and isn't accessible to the rest of the operating system. LSA uses remote procedure calls to communicate with the isolated LSA process.

For security reasons, the isolated LSA process doesn't host any device drivers. Instead, it only hosts a small subset of operating system binaries that are needed for security and nothing else. All the binaries are signed with a certificate that VBS trusts, and the signatures are validated before launching the file in the protected environment.



VSM and TPM Protections

Secrets protected by Credential Guard are protected in memory and isolated at runtime by the hypervisor using Virtual Secure Mode (VSM). On recent supported hardware with TPM 2.0, VSM data that is persisted will be protected by a key called the *VSM master key*, which is protected by device firmware protections. To learn more, see System Guard: How a hardware-based root of trust helps protect Windows. The VSM master key is protected by the TPM, ensuring that the key and secrets protected by Credential Guard can only be accessed in a trusted environment.

Credential Guard doesn't typically persist authentication data (NTLM hash and TGTs), as that data is lost between reboots and refreshed when the user signs into the system. This means that it isn't dependent on the VSM master key or the TPM to protect that data at reset.

> **Note**
>
> The VBS master key might not be protected by the TPM in any of the following environments:
>
> - If Secure Boot is disabled
> - If a TPM isn't available on the firmware

Credential Guard protection limits

Some ways to store credentials aren't protected by Credential Guard, including:

- When Credential Guard is enabled, NTLMv1, MS-CHAPv2, Digest, and CredSSP can't use the signed-in credentials. Thus, single sign-on doesn't work with these protocols. However, applications can prompt for credentials or use credentials stored in the Windows Vault, which aren't protected by Credential Guard with any of these protocols

  > **Caution**
  >
  > It's recommended that valuable credentials, such as the sign-in credentials, aren't used with NTLMv1, MS-CHAPv2, Digest, or CredSSP protocols.

- Software that manages credentials outside of Windows feature protection
- Local accounts and Microsoft Accounts
- Credential Guard doesn't protect the Active Directory database running on Windows Server domain controllers. It also doesn't protect credential input pipelines, such as Windows Server running Remote Desktop Gateway. If you're using a Windows Server OS as a client PC, it will get the same protection as it would when running a Windows client OS
- Key loggers
- Physical attacks
- Doesn't prevent an attacker with malware on the PC from using the privileges associated with any credential. We recommend using dedicated PCs for high value accounts, such as IT Pros and users with access to high value assets in your organization
- Non-Microsoft security packages
- Supplied credentials for NTLM authentication aren't protected. If a user is prompted for and enters credentials for NTLM authentication, these credentials are vulnerable to be read from LSASS memory. These same credentials are vulnerable to key loggers as well
- Kerberos service tickets aren't protected by Credential Guard, but the Kerberos Ticket Granting Ticket (TGT) is protected

- When Credential Guard is enabled, Kerberos doesn't allow *unconstrained Kerberos delegation* or *DES encryption*, not only for signed-in credentials, but also prompted or saved credentials
- When Credential Guard is enabled on a VM, it protects secrets from attacks inside the VM. However, it doesn't provide protection from privileged system attacks originating from the host
- Windows logon cached password verifiers (commonly called *cached credentials*) don't qualify as credentials because they can't be presented to another computer for authentication, and can only be used locally to verify credentials. They're stored in the registry on the local computer and provide validation for credentials when a domain-joined computer can't connect to AD DS during user logon. These *cached logons*, or more specifically, *cached domain account information*, can be managed using the security policy setting **Interactive logon: Number of previous logons to cache** if a domain controller isn't available

Next steps

- Learn how to configure Credential Guard
- Review the advice and sample code for making your environment more secure and robust with Credential Guard in the Additional mitigations article
- Review considerations and known issues when using Credential Guard

---

Last updated on 06/12/2025

Configure Credential Guard

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes how to configure Credential Guard using Microsoft Intune, Group Policy, or the registry.

Default enablement

Starting in Windows 11, 22H2 and Windows Server 2025, Credential Guard is enabled by default on devices which meet the requirements.

System administrators can explicitly enable or disable Credential Guard using one of the methods described in this article. Explicitly configured values overwrite the default enablement state after a reboot.

If a device has Credential Guard explicitly turned off before updating to a newer version of Windows where Credential Guard is enabled by default, it will remain disabled even after the update.

> ⓘ **Important**
>
> For information about known issues related to default enablement, see **Credential Guard: known issues**.

Enable Credential Guard

Credential Guard should be enabled before a device is joined to a domain or before a domain user signs in for the first time. If Credential Guard is enabled after domain join, the user and device secrets may already be compromised.

To enable Credential Guard, you can use:

- Microsoft Intune/MDM
- Group policy
- Registry

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

Configure Credential Guard with Intune

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Device Guard	Credential Guard	Select one of the options: - **Enabled with UEFI lock** - **Enabled without lock**

> **Important**
>
> If you want to be able to turn off Credential Guard remotely, choose the option **Enabled without lock**.

Assign the policy to a group that contains as members the devices or users that you want to configure.

> **Tip**
>
> You can also configure Credential Guard by using an *account protection* profile in endpoint security. For more information, see **Account protection policy settings for endpoint security in Microsoft Intune**.

Alternatively, you can configure devices using a custom policy with the DeviceGuard Policy CSP.

Expand table

Setting
Setting name: Turn On Virtualization Based Security **OMA-URI**: `./Device/Vendor/MSFT/Policy/Config/DeviceGuard/EnableVirtualizationBasedSecurity` **Data type**: int **Value**: `1`

Setting
Setting name: Credential Guard Configuration **OMA-URI**: `./Device/Vendor/MSFT/Policy/Config/DeviceGuard/LsaCfgFlags` **Data type**: int **Value**: **Enabled with UEFI lock**: `1` **Enabled without lock**: `2`

Once the policy is applied, restart the device.

Verify if Credential Guard is enabled

Checking Task Manager if `LsaIso.exe` is running isn't a recommended method for determining whether Credential Guard is running. Instead, use one of the following methods:

- System Information
- PowerShell
- Event Viewer

System Information

You can use *System Information* to determine whether Credential Guard is running on a device.

1. Select **Start**, type `msinfo32.exe`, and then select **System Information**
2. Select **System Summary**
3. Confirm that **Credential Guard** is shown next to **Virtualization-based Security Services Running**

PowerShell

You can use PowerShell to determine whether Credential Guard is running on a device. From an elevated PowerShell session, use the following command:

PowerShell

```
(Get-CimInstance -ClassName Win32_DeviceGuard -Namespace
root\Microsoft\Windows\DeviceGuard).SecurityServicesRunning
```

The command generates the following output:

- **0**: Credential Guard is disabled (not running)
- **1**: Credential Guard is enabled (running)

Event viewer

Perform regular reviews of the devices that have Credential Guard enabled, using security audit policies or WMI queries.
Open the Event Viewer (`eventvwr.exe`) and go to `Windows Logs\System` and filter the event sources for *WinInit*:

Event ID

Description

13 (Information)

logging
`Credential Guard (LsaIso.exe) was started and will protect LSA credentials.`

`14` (Information)

logging
`Credential Guard (LsaIso.exe) configuration: [**0x0** | **0x1** | **0x2**], **0**`

- The first variable: **0x1** or **0x2** means that Credential Guard is configured to run. **0x0** means that it's not configured to run.
- The second variable: **0** means that it's configured to run in protect mode. **1** means that it's configured to run in test mode. This variable should always be **0**.

`15` (Warning)

logging
`Credential Guard (LsaIso.exe) is configured but the secure kernel isn't running; continuing without Credential Guard.`

`16` (Warning)

logging
`Credential Guard (LsaIso.exe) failed to launch: [error code]`

`17`

logging

```
Error reading Credential Guard (LsaIso.exe) UEFI configuration: [error code]
```

Disable Credential Guard

There are different options to disable Credential Guard. The option you choose depends on how Credential Guard is configured:

- Credential Guard running in a virtual machine can be disabled by the host
- If Credential Guard is enabled **with UEFI Lock**, follow the procedure described in disable Credential Guard with UEFI Lock
- If Credential Guard is enabled **without UEFI Lock**, or as part of the default enablement update, use one of the following options to disable it:
  - Microsoft Intune/MDM
  - Group policy
  - Registry

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

Disable Credential Guard with Intune

If Credential Guard is enabled via Intune and without UEFI Lock, disabling the same policy setting disables Credential Guard.

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Device Guard	Credential Guard	**Disabled**

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the DeviceGuard Policy CSP.

Expand table

Setting
Setting name: Credential Guard Configuration **OMA-URI**: `./Device/Vendor/MSFT/Policy/Config/DeviceGuard/LsaCfgFlags` **Data type**: int **Value**: `0`

Once the policy is applied, restart the device.

Disable Credential Guard with UEFI lock

If Credential Guard is enabled with UEFI lock, follow this procedure since the settings are persisted in EFI (firmware) variables.

> **Note**
>
> This scenario requires physical presence at the machine to press a function key to accept the change.

1. Follow the steps in Disable Credential Guard
2. Delete the Credential Guard EFI variables by using bcdedit. From an elevated command prompt, type the following commands:

Windows Command Prompt

```
mountvol X: /s
copy %WINDIR%\System32\SecConfig.efi X:\EFI\Microsoft\Boot\SecConfig.efi /Y
bcdedit /create {0cb3b571-2f2e-4343-a879-d86a476d7215} /d "DebugTool"
/application osloader
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} path
"\EFI\Microsoft\Boot\SecConfig.efi"
bcdedit /set {bootmgr} bootsequence {0cb3b571-2f2e-4343-a879-d86a476d7215}
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} loadoptions DISABLE-LSA-
ISO
bcdedit /set {0cb3b571-2f2e-4343-a879-d86a476d7215} device partition=X:
mountvol X: /d
```

3. Restart the device. Before the OS boots, a prompt appears notifying that UEFI was modified, and asking for confirmation. The prompt must be confirmed for the changes to persist.

Disable Credential Guard for a virtual machine

From the host, you can disable Credential Guard for a virtual machine with the following command:

PowerShell

```powershell
Set-VMSecurity -VMName <VMName> -VirtualizationBasedSecurityOptOut $true
```

Next steps

- Review the advice and sample code for making your environment more secure and robust with Credential Guard in the Additional mitigations article
- Review considerations and known issues when using Credential Guard

---

Last updated on 03/29/2026

Additional mitigations

Article • 02/25/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Credential Guard offers mitigations against attacks on derived credentials, preventing the use of stolen credentials elsewhere. However, devices can still be vulnerable to certain attacks, even if the derived credentials are protected by Credential Guard. These attacks can include abusing privileges and use of derived credentials directly from a compromised device, re-using stolen credentials prior to the enablement of Credential Guard, and abuse of management tools and weak application configurations. Because of this, additional mitigation also must be deployed to make the domain environment more robust.

Additional security qualifications

All devices that meet baseline protections for hardware, firmware, and software can use Credential Guard.
Devices that meet more qualifications can provide added protections to further reduce the attack surface.

The following table list qualifications for improved security. We recommend meeting the additional qualifications to strengthen the level of security that Credential Guard can provide.

Expand table

Protection	Requirements	Security Benefits
Secure Boot configuration and management	- BIOS password or stronger authentication must be supported - In the BIOS configuration, BIOS authentication must be set - There must be support for protected BIOS option to configure list of permitted boot devices (for example, *Boot only from internal hard drive*) and boot device order, overriding `BOOTORDER` modification made by the operating system	- Prevent other operating systems from starting -Prevent changes to the BIOS settings
Hardware Rooted Trust Platform Secure Boot	- Boot Integrity (Platform Secure Boot) must be supported. See the Windows Hardware Compatibility Program requirements under System.Fundamentals.Firmware.CS.UEFISecureBoot.ConnectedStandby - Hardware Security Test Interface (HSTI) must be implemented. See Hardware Security Testability Specification	- Boot Integrity (Platform Secure Boot) from Power-On provides protections against physically present attackers, and defense-in-depth against malware. - HSTI provides

Protection	Requirements	Security Benefits
		security assurance for correctly secured silicon and platform
Firmware Update through Windows Update	- Firmware must support field updates through Windows Update and UEFI encapsulation update	Helps ensure that firmware updates are fast, secure, and reliable.
Securing Boot Configuration and Management	- Required BIOS capabilities: ability of OEM to add ISV, OEM, or Enterprise Certificate in Secure Boot DB at manufacturing time - Required configurations: Microsoft UEFI CA must be removed from Secure Boot DB. Support for 3rd-party UEFI modules is permitted but should use ISV-provided certificates or OEM certificate for the specific UEFI software	- Enterprises can choose to allow proprietary EFI drivers/applications to run - Removing Microsoft UEFI CA from Secure Boot DB provides full control to enterprises over software that runs before the operating system boots
VBS enablement of No-Execute (NX) protection for UEFI runtime services	- VBS enables NX protection on UEFI runtime service code and data memory regions. UEFI runtime service code must support read-only page protections, and UEFI runtime service data must not be executable. UEFI runtime service must meet the following requirements: - Implement UEFI 2.6 `EFI_MEMORY_ATTRIBUTES_TABLE`. All UEFI runtime service memory (code and data) must be described by this table - PE sections must be page-aligned in memory (not required for in non-volatile storage). - The Memory Attributes Table needs to correctly mark code and data as `RO/NX` for configuration by the OS - All entries must include attributes `EFI_MEMORY_RO`, `EFI_MEMORY_XP`, or both. - No entries may be left with neither of the above attributes, indicating memory that is both executable and writable. Memory must be either readable and executable or writable and non-executable (**SEE IMPORTANT INFORMATION AFTER THIS TABLE**)	- Vulnerabilities in UEFI runtime, if any, are blocked from compromising VBS (such as in functions like *UpdateCapsule* and *SetVariable*) - Reduces the attack surface to VBS from system firmware.
Firmware support for	- The Windows SMM Security Mitigations Table (WSMT) specification contains details of an ACPI table that was created for	- Protects against potential vulnerabilities in

Protection	Requirements	Security Benefits
SMM protection	use with Windows operating systems that support Windows virtualization-based features.	UEFI runtime services, if any, will be blocked from compromising VBS (such as in functions like UpdateCapsule and SetVariable) - Reduces the attack surface to VBS from system firmware - Blocks additional security attacks against SMM

> **ⓘ Important**
>
> Regarding **VBS enablement of NX protection for UEFI runtime services**:
>
> - It only applies to UEFI runtime service memory, and not UEFI boot service memory
> - The protection is applied by VBS on OS page tables
> - Don't use sections that are both writable and executable
> - Don't attempt to directly modify executable system memory
> - Don't use dynamic code

Restrict domain users to specific domain-joined devices

Credential theft attacks allow the attacker to steal secrets from one device and use them from another device. If a user can sign on to multiple devices then any device could be used to steal credentials. How do you ensure that users only sign on with devices that have Credential Guard enabled? By deploying authentication policies that restrict them to specific domain-joined devices that have been configured with Credential Guard. For the domain controller to know what device a user is signing on from, Kerberos armoring must be used.

Kerberos armoring

Kerberos armoring is part of RFC 6113. When a device supports Kerberos armoring, its TGT is used to protect the user's proof of possession which can mitigate offline dictionary attacks. Kerberos armoring also provides the additional benefit of signed KDC errors this mitigates tampering which can result in things such as downgrade attacks.

To enable Kerberos armoring for restricting domain users to specific domain-joined devices:

- Users need to be in domains that are running Windows Server 2012 R2 or higher
- All the domain controllers in these domains must be configured to support Kerberos armoring. Set the **KDC support for claims, compound authentication, and Kerberos armoring** Group Policy setting to either **Supported** or **Always provide claims**
- All the devices with Credential Guard that the users will be restricted to must be configured to support Kerberos armoring. Enable the **Kerberos client support for claims, compound authentication and Kerberos armoring** Group Policy settings under **Computer Configuration** > **Administrative Templates** > **System** > **Kerberos**.

Protect domain-joined device secrets

Since domain-joined devices also use shared secrets for authentication, attackers can steal those secrets as well. By deploying device certificates with Credential Guard, the private key can be protected. Then authentication policies can require that users sign on to devices that authenticate using those certificates. This prevents shared secrets stolen from the device to be used with stolen user credentials to sign on as the user.

Domain-joined device certificate authentication has the following requirements:

- Devices' accounts are in Windows Server 2012 domain functional level or higher.
- All domain controllers in those domains have KDC certificates that satisfy strict KDC validation certificate requirements:
  - KDC EKU present
  - DNS domain name matches the DNSName field of the SubjectAltName (SAN) extension
- Windows devices have the CA issuing the domain controller certificates in the enterprise store.
- A process is established to ensure the identity and trustworthiness of the device in a similar manner as you would establish the identity and trustworthiness of a user before issuing them a smartcard.

Deploy domain-joined device certificates

To guarantee that certificates with the required issuance policy are only installed on the devices these users must use, they must be deployed manually on each device. The same

security procedures used for issuing smart cards to users should be applied to device certificates.

For example, let's say you wanted to use the High Assurance policy only on these devices. Using a Windows Server Enterprise certificate authority, you would create a new template.

Create a new certificate template

1. From the Certificate Manager console, right-click **Certificate Templates** > **Manage**
2. Right-click **Workstation Authentication** > **Duplicate Template**
3. Right-click the new template, and then select **Properties**
4. On the **Extensions** tab, select **Application Policies** > **Edit**
5. Select **Client Authentication**, and then select **Remove**
6. Add the ID-PKInit-KPClientAuth EKU. Select **Add** > **New**, and then specify the following values:

    - Name: Kerberos Client Auth
    - Object Identifier: 1.3.6.1.5.2.3.4

7. On the **Extensions** tab, select **Issuance Policies** > **Edit**
8. Under **Issuance Policies**, select **High Assurance**
9. On the **Subject name** tab, clear the **DNS name** check box, and then select the **User Principal Name (UPN)** check box

Then on the devices that are running Credential Guard, enroll the devices using the certificate you created.

Enroll devices in a certificate

Run the following command:

```PowerShell
CertReq -EnrollCredGuardCert MachineAuthentication
```

> ⓘ **Note**
>
> You must restart the device after enrolling the machine authentication certificate.

How a certificate issuance policy can be used for access control

Beginning with the Windows Server 2008 R2 domain functional level, domain controllers support for authentication mechanism assurance provides a way to map certificate issuance policy OIDs to universal security groups. Windows Server 2012 domain controllers

with claim support can map them to claims. To learn more about authentication mechanism assurance, see Authentication Mechanism Assurance for AD DS in Windows Server 2008 R2 Step-by-Step Guide on TechNet.

To see the issuance policies available

- The get-IssuancePolicy.ps1 shows all of the issuance policies that are available on the certificate authority.
  From a Windows PowerShell command prompt, run the following command:

```
PowerShell

.\get-IssuancePolicy.ps1 -LinkedToGroup:All
```

To link an issuance policy to a universal security group

- The set-IssuancePolicyToGroupLink.ps1 creates a Universal security group, creates an organizational unit, and links the issuance policy to that Universal security group.
  From a Windows PowerShell command prompt, run the following command:

```
PowerShell

.\set-IssuancePolicyToGroupLink.ps1 -IssuancePolicyName:"<name of issuance policy>" -groupOU:"<Name of OU to create>" -groupName:"<name of Universal security group to create>"
```

Restrict user sign-on

So we now have completed the following:

- Created a special certificate issuance policy to identify devices that meet the deployment criteria required for the user to be able to sign on
- Mapped that policy to a universal security group or claim
- Provided a way for domain controllers to get the device authorization data during user sign-on using Kerberos armoring. Now what is left to do is to configure the access check on the domain controllers. This is done using authentication policies

Authentication policies have the following requirements:

- User accounts are in a Windows Server 2012 domain functional level or higher domain

Create an authentication policy restricting users to the specific universal security group

1. Open Active Directory Administrative Center
2. Select **Authentication** > **New** > **Authentication Policy**
3. In the **Display name** box, enter a name for this authentication policy
4. Under the **Accounts** heading, select **Add**
5. In the **Select Users, Computers, or Service Accounts** dialog box, type the name of the user account you wish to restrict, and then select **OK**
6. Under the **User Sign On** heading, select the **Edit** button
7. Select **Add a condition**
8. In the **Edit Access Control Conditions** box, ensure that it reads **User** > **Group** > **Member of each** > **Value**, and then select **Add items**
9. In the **Select Users, Computers, or Service Accounts** dialog box, type the name of the universal security group that you created with the set-IssuancePolicyToGroupLink script, and then select **OK**
10. Select **OK** to close the **Edit Access Control Conditions** box
11. Select **OK** to create the authentication policy
12. Select Active Directory Administrative Center

> ⓘ **Note**
>
> When the authentication policy enforces policy restrictions, users will not be able to sign on using devices that do not have a certificate with the appropriate issuance policy deployed. This applies to both local and remote sign on scenarios. Therefore, it is strongly recommended to first only audit policy restrictions to ensure you don't have unexpected failures.

Discover authentication failures due to authentication policies

To make tracking authentication failures due to authentication policies easier, an operational log exists with just those events. To enable the logs on the domain controllers, in Event Viewer, navigate to **Applications and Services Logs\Microsoft\Windows\Authentication, right-click AuthenticationPolicyFailures-DomainController**, and then select **Enable Log**.

To learn more about authentication policy events, see Authentication Policies and Authentication Policy Silos.

Appendix: Scripts

Here's a list of scripts mentioned in this article.

Get the available issuance policies on the certificate authority

Save this script file as get-IssuancePolicy.ps1.

PowerShell

```
########################################
##     Parameters to be defined       ##
##     by the user                    ##
########################################
Param (
$Identity,
$LinkedToGroup
)
########################################
##     Strings definitions            ##
########################################
Data getIP_strings {
# culture="en-US"
ConvertFrom-StringData -stringdata @'
help1 = This command can be used to retrieve all available Issuance Policies
in a forest. The forest of the currently logged on user is targeted.
help2 = Usage:
help3 = The following parameter is mandatory:
help4 = -LinkedToGroup:<yes|no|all>
help5 = "yes" will return only Issuance Policies that are linked to groups.
Checks that the linked Issuance Policies are linked to valid groups.
help6 = "no" will return only Issuance Policies that are not currently linked
to any group.
help7 = "all" will return all Issuance Policies defined in the forest. Checks
that the linked Issuance policies are linked to valid groups.
help8 = The following parameter is optional:
help9 = -Identity:<Name, Distinguished Name or Display Name of the Issuance
Policy that you want to retrieve>. If you specify an identity, the option
specified in the "-LinkedToGroup" parameter is ignored.
help10 = Output: This script returns the Issuance Policy objects meeting the
criteria defined by the above parameters.
help11 = Examples:
errorIPNotFound = Error: no Issuance Policy could be found with Identity "{0}"
ErrorNotSecurity = Error: Issuance Policy "{0}" is linked to group "{1}" which
is not of type "Security".
ErrorNotUniversal = Error: Issuance Policy "{0}" is linked to group "{1}"
whose scope is not "Universal".
ErrorHasMembers = Error: Issuance Policy "{0}" is linked to group "{1}" which
has a non-empty membership. The group has the following members:
LinkedIPs = The following Issuance Policies are linked to groups:
displayName = displayName : {0}
Name = Name : {0}
dn = distinguishedName : {0}
        InfoName = Linked Group Name: {0}
        InfoDN = Linked Group DN: {0}
NonLinkedIPs = The following Issuance Policies are NOT linked to groups:
'@
```

```
}
##Import-LocalizedData getIP_strings
import-module ActiveDirectory
########################################
##           Help                     ##
########################################
function Display-Help {
    ""
    $getIP_strings.help1
    ""
$getIP_strings.help2
""
$getIP_strings.help3
"     " + $getIP_strings.help4
"             " + $getIP_strings.help5
    "             " + $getIP_strings.help6
    "             " + $getIP_strings.help7
""
$getIP_strings.help8
    "     " + $getIP_strings.help9
    ""
    $getIP_strings.help10
""
""
$getIP_strings.help11
    "     " + '$' + "myIPs = .\get-IssuancePolicy.ps1 -LinkedToGroup:All"
    "     " + '$' + "myLinkedIPs = .\get-IssuancePolicy.ps1 -
LinkedToGroup:yes"
    "     " + '$' + "myIP = .\get-IssuancePolicy.ps1 -Identity:""Medium
Assurance"""
""
}
$root = get-adrootdse
$domain = get-addomain -current loggedonuser
$configNCDN = [String]$root.configurationNamingContext
if ( !($Identity) -and !($LinkedToGroup) ) {
display-Help
break
}
if ($Identity) {
    $OIDs = get-adobject -Filter {(objectclass -eq "msPKI-Enterprise-Oid") -
and ((name -eq $Identity) -or (displayname -eq $Identity) -or
(distinguishedName -like $Identity)) } -searchBase $configNCDN -properties *
    if ($OIDs -eq $null) {
$errormsg = $getIP_strings.ErrorIPNotFound -f $Identity
write-host $errormsg -ForegroundColor Red
    }
    foreach ($OID in $OIDs) {
        if ($OID."msDS-OIDToGroupLink") {
# In case the Issuance Policy is linked to a group, it is good to check
whether there is any problem with the mapping.
            $groupDN = $OID."msDS-OIDToGroupLink"
            $group = get-adgroup -Identity $groupDN
    $groupName = $group.Name
# Analyze the group
            if ($group.groupCategory -ne "Security") {
```

```
$errormsg = $getIP_strings.ErrorNotSecurity -f $Identity, $groupName
                write-host $errormsg -ForegroundColor Red
            }
            if ($group.groupScope -ne "Universal") {
                $errormsg = $getIP_strings.ErrorNotUniversal -f $Identity,
$groupName
write-host $errormsg -ForegroundColor Red
            }
            $members = Get-ADGroupMember -Identity $group
            if ($members) {
                $errormsg = $getIP_strings.ErrorHasMembers -f $Identity,
$groupName
write-host $errormsg -ForegroundColor Red
                foreach ($member in $members) {
                    write-host "          "  $member -ForeGroundColor Red
                }
            }
        }
    }
    return $OIDs
    break
}
if (($LinkedToGroup -eq "yes") -or ($LinkedToGroup -eq "all")) {
    $LDAPFilter = "(&(objectClass=msPKI-Enterprise-Oid)(msDS-OIDToGroupLink=*)
(flags=2))"
    $LinkedOIDs = get-adobject -searchBase $configNCDN -LDAPFilter $LDAPFilter
-properties *
    write-host ""
    write-host "**************************************************"
    write-host $getIP_strings.LinkedIPs
    write-host "**************************************************"
    write-host ""
    if ($LinkedOIDs -ne $null){
      foreach ($OID in $LinkedOIDs) {
# Display basic information about the Issuance Policies
          ""
  $getIP_strings.displayName -f $OID.displayName
  $getIP_strings.Name -f $OID.Name
  $getIP_strings.dn -f $OID.distinguishedName
# Get the linked group.
          $groupDN = $OID."msDS-OIDToGroupLink"
          $group = get-adgroup -Identity $groupDN
          $getIP_strings.InfoName -f $group.Name
          $getIP_strings.InfoDN -f $groupDN
# Analyze the group
          $OIDName = $OID.displayName
    $groupName = $group.Name
          if ($group.groupCategory -ne "Security") {
          $errormsg = $getIP_strings.ErrorNotSecurity -f $OIDName, $groupName
          write-host $errormsg -ForegroundColor Red
          }
          if ($group.groupScope -ne "Universal") {
          $errormsg = $getIP_strings.ErrorNotUniversal -f $OIDName, $groupName
          write-host $errormsg -ForegroundColor Red
          }
          $members = Get-ADGroupMember -Identity $group
```

```
          if ($members) {
          $errormsg = $getIP_strings.ErrorHasMembers -f $OIDName, $groupName
          write-host $errormsg -ForegroundColor Red
              foreach ($member in $members) {
                  write-host "          "  $member -ForeGroundColor Red
              }
          }
          write-host ""
      }
    }else{
write-host "There are no issuance policies that are mapped to a group"
    }
    if ($LinkedToGroup -eq "yes") {
        return $LinkedOIDs
        break
    }
}
if (($LinkedToGroup -eq "no") -or ($LinkedToGroup -eq "all")) {
    $LDAPFilter = "(&(objectClass=msPKI-Enterprise-Oid)(!(msDS-
OIDToGroupLink=*))(flags=2))"
    $NonLinkedOIDs = get-adobject -searchBase $configNCDN -LDAPFilter
$LDAPFilter -properties *
    write-host ""
    write-host "************************************************************"
    write-host $getIP_strings.NonLinkedIPs
    write-host "************************************************************"
    write-host ""
    if ($NonLinkedOIDs -ne $null) {
      foreach ($OID in $NonLinkedOIDs) {
# Display basic information about the Issuance Policies
write-host ""
$getIP_strings.displayName -f $OID.displayName
$getIP_strings.Name -f $OID.Name
$getIP_strings.dn -f $OID.distinguishedName
write-host ""
      }
    }else{
write-host "There are no issuance policies which are not mapped to groups"
    }
    if ($LinkedToGroup -eq "no") {
        return $NonLinkedOIDs
        break
    }
}
```

> ⓘ Note
>
> If you're having trouble running this script, try replacing the single quote after the ConvertFrom-StringData parameter.

Link an issuance policy to a group

Save the script file as set-IssuancePolicyToGroupLink.ps1.

PowerShell

```
#######################################
##     Parameters to be defined      ##
##     by the user                   ##
#######################################
Param (
$IssuancePolicyName,
$groupOU,
$groupName
)
#######################################
##     Strings definitions           ##
#######################################
Data ErrorMsg {
# culture="en-US"
ConvertFrom-StringData -stringdata @'
help1 = This command can be used to set the link between a certificate
issuance policy and a universal security group.
help2 = Usage:
help3 = The following parameters are required:
help4 = -IssuancePolicyName:<name or display name of the issuance policy that
you want to link to a group>
help5 = -groupName:<name of the group you want to link the issuance policy
to>. If no name is specified, any existing link to a group is removed from the
Issuance Policy.
help6 = The following parameter is optional:
help7 = -groupOU:<Name of the Organizational Unit dedicated to the groups
which are linked to issuance policies>. If this parameter is not specified,
the group is looked for or created in the Users container.
help8 = Examples:
help9 = This command will link the issuance policy whose display name is "High
Assurance" to the group "HighAssuranceGroup" in the Organizational Unit
"OU_FOR_IPol_linked_groups". If the group or the Organizational Unit do not
exist, you will be prompted to create them.
help10 = This command will unlink the issuance policy whose name is
"402.164959C40F4A5C12C6302E31D5476062" from any group.
MultipleIPs = Error: Multiple Issuance Policies with name or display name "
{0}" were found in the subtree of "{1}"
NoIP = Error: no issuance policy with name or display name "{0}" could be
found in the subtree of "{1}".
IPFound = An Issuance Policy with name or display name "{0}" was successfully
found: {1}
MultipleOUs = Error: more than 1 Organizational Unit with name "{0}" could be
found in the subtree of "{1}".
confirmOUcreation = Warning: The Organizational Unit that you specified does
not exist. Do you want to create it?
OUCreationSuccess = Organizational Unit "{0}" successfully created.
OUcreationError = Error: Organizational Unit "{0}" could not be created.
OUFoundSuccess = Organizational Unit "{0}" was successfully found.
multipleGroups = Error: More than one group with name "{0}" was found in
Organizational Unit "{1}".
confirmGroupCreation = Warning: The group that you specified does not exist.
Do you want to create it?
```

```
groupCreationSuccess = Univeral Security group "{0}" successfully created.
groupCreationError = Error: Univeral Security group "{0}" could not be
created.
GroupFound = Group "{0}" was successfully found.
confirmLinkDeletion = Warning: The Issuance Policy "{0}" is currently linked
to group "{1}". Do you really want to remove the link?
UnlinkSuccess = Certificate issuance policy successfully unlinked from any
group.
UnlinkError = Removing the link failed.
UnlinkExit = Exiting without removing the link from the issuance policy to the
group.
IPNotLinked = The Certificate issuance policy is not currently linked to any
group. If you want to link it to a group, you should specify the -groupName
option when starting this script.
ErrorNotSecurity = Error: You cannot link issuance Policy "{0}" to group "{1}"
because this group is not of type "Security".
ErrorNotUniversal = Error: You cannot link issuance Policy "{0}" to group "
{1}" because the scope of this group is not "Universal".
ErrorHasMembers = Error: You cannot link issuance Policy "{0}" to group "{1}"
because it has a non-empty membership. The group has the following members:
ConfirmLinkReplacement = Warning: The Issuance Policy "{0}" is currently
linked to group "{1}". Do you really want to update the link to point to group
"{2}"?
LinkSuccess = The certificate issuance policy was successfully linked to the
specified group.
LinkError = The certificate issuance policy could not be linked to the
specified group.
ExitNoLinkReplacement = Exiting without setting the new link.
'@
}
# import-localizeddata ErrorMsg
function Display-Help {
""
write-host $ErrorMsg.help1
""
write-host $ErrorMsg.help2
""
write-host $ErrorMsg.help3
write-host "`t" $ErrorMsg.help4
write-host "`t" $ErrorMsg.help5
""
write-host $ErrorMsg.help6
write-host "`t" $ErrorMsg.help7
""
""
write-host $ErrorMsg.help8
""
write-host $ErrorMsg.help9
".\Set-IssuancePolicyToGroupMapping.ps1 -IssuancePolicyName ""High Assurance""
-groupOU ""OU_FOR_IPol_linked_groups"" -groupName ""HighAssuranceGroup"" "
""
write-host $ErrorMsg.help10
'.\Set-IssuancePolicyToGroupMapping.ps1 -IssuancePolicyName
"402.164959C40F4A5C12C6302E31D5476062" -groupName $null '
""
}
```

```
# Assumption:  The group to which the Issuance Policy is going
#              to be linked is (or is going to be created) in
#              the domain the user running this script is a member of.
import-module ActiveDirectory
$root = get-adrootdse
$domain = get-addomain -current loggedonuser
if ( !($IssuancePolicyName) ) {
display-Help
break
}
########################################
##     Find the OID object            ##
##     (aka Issuance Policy)          ##
########################################
$searchBase = [String]$root.configurationnamingcontext
$OID = get-adobject -searchBase $searchBase -Filter { ((displayname -eq
$IssuancePolicyName) -or (name -eq $IssuancePolicyName)) -and (objectClass -eq
"msPKI-Enterprise-Oid")} -properties *
if ($OID -eq $null) {
$tmp = $ErrorMsg.NoIP -f $IssuancePolicyName, $searchBase
write-host $tmp -ForeGroundColor Red
break;
}
elseif ($OID.GetType().IsArray) {
$tmp = $ErrorMsg.MultipleIPs -f $IssuancePolicyName, $searchBase
write-host $tmp -ForeGroundColor Red
break;
}
else {
$tmp = $ErrorMsg.IPFound -f $IssuancePolicyName, $OID.distinguishedName
write-host $tmp -ForeGroundColor Green
}
########################################
##  Find the container of the group   ##
########################################
if ($groupOU -eq $null) {
# default to the Users container
$groupContainer = $domain.UsersContainer
}
else {
$searchBase = [string]$domain.DistinguishedName
$groupContainer = get-adobject -searchBase $searchBase -Filter { (Name -eq
$groupOU) -and (objectClass -eq "organizationalUnit")}
if ($groupContainer.count -gt 1) {
$tmp = $ErrorMsg.MultipleOUs -f $groupOU, $searchBase
write-host $tmp -ForegroundColor Red
break;
}
elseif ($groupContainer -eq $null) {
$tmp = $ErrorMsg.confirmOUcreation
write-host $tmp " ( (y)es / (n)o )" -ForegroundColor Yellow -nonewline
$userChoice = read-host
if ( ($userChoice -eq "y") -or ($userChoice -eq "yes") ) {
new-adobject -Name $groupOU -displayName $groupOU -Type "organizationalUnit" -
ProtectedFromAccidentalDeletion $true -path $domain.distinguishedName
if ($?){
```

```
$tmp = $ErrorMsg.OUCreationSuccess -f $groupOU
write-host $tmp -ForegroundColor Green
}
else{
$tmp = $ErrorMsg.OUCreationError -f $groupOU
write-host $tmp -ForeGroundColor Red
break;
}
$groupContainer = get-adobject -searchBase $searchBase -Filter { (Name -eq
$groupOU) -and (objectClass -eq "organizationalUnit")}
}
else {
break;
}
}
else {
$tmp = $ErrorMsg.OUFoundSuccess -f $groupContainer.name
write-host $tmp -ForegroundColor Green
}
}
########################################
##  Find the group                ##
########################################
if (($groupName -ne $null) -and ($groupName -ne "")){
##$searchBase = [String]$groupContainer.DistinguishedName
$searchBase = $groupContainer
$group = get-adgroup -Filter { (Name -eq $groupName) -and (objectClass -eq
"group") } -searchBase $searchBase
if ($group -ne $null -and $group.gettype().isarray) {
$tmp = $ErrorMsg.multipleGroups -f $groupName, $searchBase
write-host $tmp -ForeGroundColor Red
break;
}
elseif ($group -eq $null) {
$tmp = $ErrorMsg.confirmGroupCreation
write-host $tmp " ( (y)es / (n)o )" -ForegroundColor Yellow -nonewline
$userChoice = read-host
if ( ($userChoice -eq "y") -or ($userChoice -eq "yes") ) {
new-adgroup -samAccountName $groupName -path $groupContainer.distinguishedName
-GroupScope "Universal" -GroupCategory "Security"
if ($?){
$tmp = $ErrorMsg.GroupCreationSuccess -f $groupName
write-host $tmp -ForegroundColor Green
}else{
$tmp = $ErrorMsg.groupCreationError -f $groupName
write-host $tmp -ForeGroundColor Red
break
}
$group = get-adgroup -Filter { (Name -eq $groupName) -and (objectClass -eq
"group") } -searchBase $searchBase
}
else {
break;
}
}
else {
```

```
$tmp = $ErrorMsg.GroupFound -f $group.Name
write-host $tmp -ForegroundColor Green
}
}
else {
#####
## If the group is not specified, we should remove the link if any exists
#####
if ($OID."msDS-OIDToGroupLink" -ne $null) {
$tmp = $ErrorMsg.confirmLinkDeletion -f $IssuancePolicyName, $OID."msDS-
OIDToGroupLink"
write-host $tmp " ( (y)es / (n)o )" -ForegroundColor Yellow -nonewline
$userChoice = read-host
if ( ($userChoice -eq "y") -or ($userChoice -eq "yes") ) {
set-adobject -Identity $OID -Clear "msDS-OIDToGroupLink"
if ($?) {
$tmp = $ErrorMsg.UnlinkSuccess
write-host $tmp -ForeGroundColor Green
}else{
$tmp = $ErrorMsg.UnlinkError
write-host $tmp -ForeGroundColor Red
}
}
else {
$tmp = $ErrorMsg.UnlinkExit
write-host $tmp
break
}
}
else {
$tmp = $ErrorMsg.IPNotLinked
write-host $tmp -ForeGroundColor Yellow
}
break;
}
########################################
##  Verify that the group is          ##
##  Universal, Security, and          ##
##  has no members                    ##
########################################
if ($group.GroupScope -ne "Universal") {
$tmp = $ErrorMsg.ErrorNotUniversal -f $IssuancePolicyName, $groupName
write-host $tmp -ForeGroundColor Red
break;
}
if ($group.GroupCategory -ne "Security") {
$tmp = $ErrorMsg.ErrorNotSecurity -f $IssuancePolicyName, $groupName
write-host $tmp -ForeGroundColor Red
break;
}
$members = Get-ADGroupMember -Identity $group
if ($members -ne $null) {
$tmp = $ErrorMsg.ErrorHasMembers -f $IssuancePolicyName, $groupName
write-host $tmp -ForeGroundColor Red
foreach ($member in $members) {write-host "   $member.name" -ForeGroundColor
Red}
```

```
break;
}
#######################################
##  We have verified everything. We  ##
##  can create the link from the     ##
##  Issuance Policy to the group.    ##
#######################################
if ($OID."msDS-OIDToGroupLink" -ne $null) {
$tmp = $ErrorMsg.ConfirmLinkReplacement -f $IssuancePolicyName, $OID."msDS-
OIDToGroupLink", $group.distinguishedName
write-host $tmp  "( (y)es / (n)o )" -ForegroundColor Yellow -nonewline
$userChoice = read-host
if ( ($userChoice -eq "y") -or ($userChoice -eq "yes") ) {
$tmp = @{'msDS-OIDToGroupLink'= $group.DistinguishedName}
set-adobject -Identity $OID -Replace $tmp
if ($?) {
$tmp = $Errormsg.LinkSuccess
write-host $tmp -Foreground Green
}else{
$tmp = $ErrorMsg.LinkError
write-host $tmp -Foreground Red
}
} else {
$tmp = $Errormsg.ExitNoLinkReplacement
write-host $tmp
break
}
}
else {
$tmp = @{'msDS-OIDToGroupLink'= $group.DistinguishedName}
set-adobject -Identity $OID -Add $tmp
if ($?) {
$tmp = $Errormsg.LinkSuccess
write-host $tmp -Foreground Green
}else{
$tmp = $ErrorMsg.LinkError
write-host $tmp -Foreground Red
}
}
```

> **Note**
>
> If you're having trouble running this script, try replacing the single quote after the ConvertFrom-StringData parameter.

Feedback

Was this page helpful? Yes No

Provide product feedback

Considerations and known issues when using Credential Guard

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Microsoft recommends that in addition to deploying Credential Guard, organizations move away from passwords to other authentication methods, such as Windows Hello for Business, FIDO 2 security keys, or smart cards.

Upgrade considerations

As Credential Guard evolves and enhances its security features, newer versions of Windows running Credential Guard might affect previously functional scenarios. For instance, Credential Guard could restrict the use of certain credentials or components to thwart malware exploiting vulnerabilities.

It's advisable to thoroughly test operational scenarios within an organization before updating devices that utilize Credential Guard.

Upgrades to Windows 11, version 22H2, and Windows Server 2025 have Credential Guard enabled by default unless explicitly disabled.

Wi-fi and VPN considerations

When Credential Guard is enabled, you can no longer use NTLM classic authentication (NTLMv1) for single-sign-on (SSO). You'll be forced to enter your credentials to use these protocols and can't save the credentials for future use.

If you're using WiFi and VPN endpoints that are based on MS-CHAPv2, they're subject to similar attacks as for NTLMv1.

For WiFi and VPN connections, it's recommended to move from MSCHAPv2-based connections (such as PEAP-MSCHAPv2 and EAP-MSCHAPv2), to certificate-based authentication (such as PEAP-TLS or EAP-TLS).

Delegation considerations

When Credential Guard is enabled, certain types of identity delegation are unusable, as their underlying authentication schemes are incompatible with Credential Guard or require supplied credentials.

When Credential Guard is enabled, Credential Security Support Provider ("CredSSP") is no longer able to use saved or SSO credentials, though cleartext credentials can still be supplied. CredSSP-based Delegation requires cleartext credentials to be supplied on the destination machine, and doesn't work with SSO once Credential Guard is enabled and blocks cleartext credential disclosure. Usage of CredSSP for delegation, and in general, isn't recommended due to the risk of credential theft.

Kerberos Unconstrained delegation and DES are blocked by Credential Guard. Unconstrained delegation isn't a recommended practice.

Instead Kerberos or Negotiate SSP are recommended for authentication generally, and for delegation, Kerberos Constrained Delegation and Resource-Based Kerberos Constrained Delegation are recommended. These methods provide greater credential security overall, and are also compatible with Credential Guard.

Non-Microsoft Security Support Providers considerations

Some non-Microsoft Security Support Providers (SSPs and APs) might not be compatible with Credential Guard because it doesn't allow non-Microsoft SSPs to ask for password hashes from LSA. However, SSPs and APs still get notified of the password when a user logs on and/or changes their password. Any use of undocumented APIs within custom SSPs and APs aren't supported.

It's recommended that custom implementations of SSPs/APs are tested with Credential Guard. SSPs and APs that depend on any undocumented or unsupported behaviors fail. For example, using the KerbQuerySupplementalCredentialsMessage API isn't supported. Replacing the NTLM or Kerberos SSPs with custom SSPs and APs.

For more information, see Restrictions around Registering and Installing a Security Package.

Saved Windows credentials considerations

Credential Manager allows you to store three types of credentials:

- Windows credentials
- Certificate-based credentials
- Generic credentials

Domain credentials that are stored in *Credential Manager* are protected with Credential Guard.

Generic credentials, such as user names and passwords that you use to sign in websites, aren't protected since the applications require your clear-text password. If the application doesn't need a copy of the password, they can save domain credentials as Windows credentials that are protected. Windows credentials are used to connect to other computers on a network.

The following considerations apply to the Credential Guard protections for Credential Manager:

- Windows credentials saved by the Remote Desktop client can't be sent to a remote host. Attempts to use saved Windows credentials fail, displaying the error message *Logon attempt failed*
- Applications that extract Windows credentials fail
- When credentials are backed up from a PC that has Credential Guard enabled, the Windows credentials can't be restored. If you need to back up your credentials, you must do so before you enable Credential Guard

TPM clearing considerations

Virtualization-based Security (VBS) uses the TPM to protect its key. When the TPM is cleared, the TPM protected key used to encrypt VBS secrets is lost.

> ⚠ **Warning**
>
> Clearing the TPM results in loss of protected data for all features that use VBS to protect data.
>
> When a TPM is cleared, **all** features that use VBS to protect data can no longer decrypt their protected data.

As a result, Credential Guard can no longer decrypt protected data. VBS creates a new TPM protected key for Credential Guard. Credential Guard uses the new key to protect new data. However, the previously protected data is lost forever.

> ⓘ **Note**
>
> Credential Guard obtains the key during initialization. The data loss will only impact persistent data and occur after the next system startup.

Windows credentials saved to Credential Manager

Since Credential Manager can't decrypt saved Windows Credentials, they're deleted. Applications should prompt for credentials that were previously saved. If saved again, then Windows credentials are protected Credential Guard.

Domain-joined device's automatically provisioned public key

Active Directory domain-joined devices automatically provision a bound public key, for more information about automatic public key provisioning, see Domain-joined Device Public Key Authentication.

Since Credential Guard can't decrypt the protected private key, Windows uses the domain-joined computer's password for authentication to the domain. Unless other policies are deployed, there shouldn't be a loss of functionality. If a device is configured to only use public key, then it can't authenticate with password until that policy is disabled. For more information on Configuring devices to only use public key, see Domain-joined Device Public Key Authentication.

Also if any access control checks including authentication policies require devices to have either the `KEY TRUST IDENTITY (S-1-18-4)` or `FRESH PUBLIC KEY IDENTITY (S-1-18-3)` well-known SIDs, then those access checks fail. For more information about authentication policies, see Authentication Policies and Authentication Policy Silos. For more information about well-known SIDs, see [MS-DTYP] Section 2.4.2.4 Well-known SID Structures.

Breaking DPAPI on domain-joined devices

On domain-joined devices, DPAPI can recover user keys using a domain controller from the user's domain. If a domain-joined device has no connectivity to a domain controller, then recovery isn't possible.

> ⓘ **Important**
>
> Best practice when clearing a TPM on a domain-joined device is to be on a network with connectivity to domain controllers. This ensures DPAPI functions and the user does not experience strange behavior.

Auto VPN configuration is protected with user DPAPI. User might not be able to use VPN to connect to domain controllers since the VPN configurations are lost. If you must clear the TPM on a domain-joined device without connectivity to domain controllers, then you should consider the following.

Domain user sign-in on a domain-joined device after clearing a TPM for as long as there's no connectivity to a domain controller:

Expand table

Credential Type	Behavior
Certificate (smart card or Windows Hello for Business)	All data protected with user DPAPI is unusable and user DPAPI doesn't work at all.
Password	If the user signed in with a certificate or password prior to clearing the TPM, then they can sign-in with password and user DPAPI is unaffected.

Once the device has connectivity to the domain controllers, DPAPI recovers the user's key and data protected prior to clearing the TPM can be decrypted.

Impact of DPAPI failures on Windows Information Protection

When data protected with user DPAPI is unusable, then the user loses access to all work data protected by Windows Information Protection. The impact includes: Outlook is unable to start and work protected documents can't be opened. If DPAPI is working, then newly created work data is protected and can be accessed.

Workaround: Users can resolve the problem by connecting their device to the domain and rebooting or using their Encrypting File System Data Recovery Agent certificate. For more information about Encrypting File System Data Recovery Agent certificate, see Create and verify an Encrypting File System (EFS) Data Recovery Agent (DRA) certificate.

Azure Virtual Machines lose access to the data protected by Credential Guard after deallocation

When an Azure Virtual Machine is deallocated, the underlying hardware is released, causing the keys protected by the TPM to become inaccessible. Consequently, any data protected by those keys also becomes inaccessible.

For more information, see States and billing status of Azure Virtual Machines.

Known issues

Credential Guard blocks certain authentication capabilities. Applications that require such capabilities won't function when Credential Guard is enabled.

This article describes known issues when Credential Guard is enabled.

Live migration with Hyper-V breaks when upgrading to Windows Server 2025

Devices that use CredSSP-based Delegation might no longer be able to use Live Migration with Hyper-V after upgrading to Windows Server 2025. Applications and services that rely on live migration (such as SCVMM) might also be affected. CredSSP-based delegation is the default for Windows Server 2022 and earlier for live migration.

Expand table

	Description
Affected devices	Any server with Credential Guard enabled might encounter this issue. Starting in Windows Server 2025, Credential Guard is enabled by default on all domain-joined servers that aren't domain controllers. Default enablement of Credential Guard can be preemptively blocked before upgrade.
Cause of the issue	Live Migration with Hyper-V, and applications and services that rely on it, are affected by the issue if one or both ends of a given connection try to use CredSSP with Credential Guard enabled. With Credential Guard enabled, CredSSP can only utilize supplied credentials, not saved or SSO credentials. If the source machine of a Live Migration uses CredSSP for delegation with Credential Guard enabled, the Live Migration fails. In most cases, Credential Guard's enablement state on the destination machine won't impact Live Migration. Live Migration also fails in cluster scenarios (for example, SCVMM), since any device might act as a source machine.
Resolution	Instead of CredSSP Delegation, Kerberos Constrained Delegation and Resource-Based Kerberos Constrained Delegation are recommended. These forms of delegation provide greater credential protections, in addition to being compatible with Credential Guard. Administrators of Hyper-V can configure these types of delegation manually or with the help of automated scripts.

Single sign-on for Network services breaks after upgrading to Windows 11, version 22H2 or Windows Server 2025

Devices that use 802.1x wireless or wired network, RDP, or VPN connections that rely on insecure protocols with password-based authentication are unable to use SSO to sign in and are forced to manually reauthenticate in every new Windows session when Credential Guard is running.

Expand table

	Description
Affected devices	Any device with Credential Guard enabled might encounter the issue. Starting in Windows 11, version 22H2, and Windows Server 2025, eligible devices that didn't disable Credential Guard, have it enabled by default. This affects all devices on Enterprise (E3 and E5) and Education licenses, and some Pro licenses, as long as they meet the minimum hardware requirements. All Windows Pro devices that previously ran Credential Guard on an eligible license and later downgraded to Pro, and which still meet the minimum hardware requirements, receive default enablement.
Cause of the issue	Applications and services are affected by the issue when they rely on insecure protocols that use password-based authentication. Such protocols are considered insecure because they can lead to password disclosure on the client or the server, and Credential Guard blocks them. Affected protocols include: - Kerberos unconstrained delegation (both SSO and supplied credentials are blocked) - Kerberos when PKINIT uses RSA encryption instead of Diffie-Hellman (both SSO and supplied credentials are blocked) - MS-CHAP (only SSO is blocked) - WDigest (only SSO is blocked) - NTLM v1 (only SSO is blocked) **Note**: Since only SSO is blocked for MS-CHAP, WDigest, and NTLM v1, these protocols can still be used by prompting the user to supply credentials.
Resolution	Microsoft recommends moving away from MSCHAPv2-based connections (for example, PEAP-MSCHAPv2 and EAP-MSCHAPv2), to certificate-based authentication (for example, PEAP-TLS or EAP-TLS). Credential Guard doesn't block certificate-based authentication. For a more immediate, but less secure fix, disable Credential Guard. Credential Guard doesn't have per-protocol or per-application policies, and it can either be turned on or off. If you disable Credential Guard, you leave stored domain credentials vulnerable to theft.

> **Tip**
>
> To prevent default enablement, configure your devices **to disable Credential Guard** before updating to a version which **received default enablement**. If the setting is not configured (which is the default state) and if the device is eligible, the device automatically enable Credential Guard after the update.
>
> If Credential Guard is explicitly disabled, the device won't automatically enable Credential Guard after the update.

> **Note**

To determine if a Windows Pro device receives default enablement when upgraded to **Windows 11, version 22H2** or **Windows Server 2025**, check if the registry key `IsolatedCredentialsRootSecret` is present in `Computer\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\MSV1_0`. If it's present, the device enables Credential Guard after the update.

Credential Guard can be disabled after upgrade by following the disablement instructions.

How to confirm the issue

MS-CHAP and NTLMv1 are relevant to the SSO breakage after the Windows 11, version 22H2 update. To confirm if Credential Guard is blocking MS-CHAP or NTLMv1, open the Event Viewer (`eventvwr.exe`) and go to `Application and Services Logs\Microsoft\Windows\NTLM\Operational`. Check the following logs:

Event ID (type)

Description

4013 (Warning)

logging

```
<string
id="NTLMv1BlockedByCredGuard"
value="Attempt to use NTLMv1 failed.
Target server: %1%nSupplied user: %2%nSupplied domain: %3%nPID
of client process: %4%nName
of client process: %5%nLUID
of client process: %6%nUser identity
of client process: %7%nDomain name
of user identity of client process: %8%nMechanism OID: 9%n%n
This device doesn't support NTLMv1.
/>
```

4014 (Error)

logging

```
<string
id="NTLMGetCredentialKeyBlockedByCredGuard"
value="Attempt to get credential key by call package blocked by Credential
Guard.%n%n
```

```
Calling Process Name: %1%nService Host Tag: %2"
/>
```

Issues with non-Microsoft applications

The following issue affects MSCHAPv2:

- Credential Guard doesn't work with MSCHAPv2 configurations, of which Cisco ISE is a common enterprise implementation.

The following issue affects the Java GSS API. See the following Oracle bug database article:

- JDK-8161921: Credential Guard doesn't allow sharing of TGT with Java

When Credential Guard is enabled on Windows, the Java GSS API doesn't authenticate. Credential Guard blocks specific application authentication capabilities and doesn't provide the TGT session key to applications, regardless of registry key settings. For more information, see Application requirements.

Vendor support

The following products and services don't support Credential Guard:

- Check Point Endpoint Security Client support for Microsoft Credential Guard and Hypervisor-Protected Code Integrity features
- *VMware Workstation and Device/Credential Guard aren't compatible* error in VMware Workstation on Windows 10 host (2146361)
- ThinkPad support for Hypervisor-Protected Code Integrity and Credential Guard in Microsoft Windows
- Windows devices with Credential Guard and Symantec Endpoint Protection 12.1

> **Important**
>
> This list isn't comprehensive. Check whether your product vendor, product version, or computer system supports Credential Guard on systems that run a specific version of Windows. Specific computer system models may be incompatible with Credential Guard.

Last updated on 04/22/2025

Remote Credential Guard

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Overview

Remote Credential Guard helps protecting credentials over a Remote Desktop (RDP) connection by redirecting Kerberos requests back to the device that's requesting the connection. If the target device is compromised, the credentials aren't exposed because both credential and credential derivatives are never passed over the network to the target device. Remote Credential Guard also provides single sign-on experiences for Remote Desktop sessions.

This article describes how to configure and use Remote Credential Guard.

> ⓘ **Important**
>
> For information on Remote Desktop connection scenarios involving helpdesk support, see Remote Desktop connections and helpdesk support scenarios in this article.

Compare Remote Credential Guard with other connection options

Using a Remote Desktop session without Remote Credential Guard has the following security implications:

- Credentials are sent to and stored on the remote host
- Credentials aren't protected from attackers on the remote host
- Attacker can use credentials after disconnection

The security benefits of Remote Credential Guard include:

- Credentials aren't sent to the remote host
- During the remote session, you can connect to other systems using SSO
- An attacker can act on behalf of the user only when the session is ongoing

The security benefits of Restricted Admin mode include:

- Credentials aren't sent to the remote host
- The Remote Desktop session connects to other resources as the remote host's identity
- An attacker can't act on behalf of the user and any attack is local to the server

Use the following table to compare different Remote Desktop connection security options:

Expand table

Feature	Remote Desktop	Remote Credential Guard	Restricted Admin mode
Single sign-on (SSO) to other systems as signed in user	✅	✅	❌
Multi-hop RDP	✅	✅	❌
Prevent use of user's identity during connection	❌	❌	✅
Prevent use of credentials after disconnection	❌	✅	✅
Prevent Pass-the-Hash (PtH)	❌	✅	✅
Supported authentication	Any negotiable protocol	Kerberos only	Any negotiable protocol
Credentials supported from the remote desktop client device	- Signed on credentials - Supplied credentials - Saved credentials	- Signed on credentials - Supplied credentials	- Signed on credentials - Supplied credentials - Saved credentials
RDP access granted with	Membership of **Remote Desktop Users** group on remote host	Membership of **Remote Desktop Users** group on remote host	Membership of **Administrators** group on remote host

Remote Credential Guard requirements

To use Remote Credential Guard, the remote host and the client must meet the following requirements.

The remote host:

- Must allow the user to access via Remote Desktop connections
- Must allow delegation of nonexportable credentials to the client device

The client device:

- Must be running the Remote Desktop Windows application. The Remote Desktop Universal Windows Platform (UWP) application doesn't support Remote Credential Guard
- Must use Kerberos authentication to connect to the remote host. If the client can't connect to a domain controller, then RDP attempts to fall back to NTLM. Remote Credential Guard doesn't allow NTLM fallback because it would expose credentials to risk

Windows edition and licensing requirements

The following table lists the Windows editions that support Remote Credential Guard:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Remote Credential Guard license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Enable delegation of nonexportable credentials on the remote hosts

This policy is required on the remote hosts to support Remote Credential Guard and Restricted Admin mode. It allows the remote host to delegate nonexportable credentials to the client device.
If you disable or don't configure this setting, Restricted Admin and Remote Credential Guard mode aren't supported. Users must pass their credentials to the host, exposing them to the risk of credential theft from attackers on the remote host.

To enable delegation of nonexportable credentials on the remote hosts, you can use:

- Microsoft Intune/MDM
- Group policy
- Registry

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Administrative Templates > System > Credentials Delegation	Remote host allows delegation of nonexportable credentials	Enabled

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the Policy CSP.

Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/Policy/Config/CredentialsDelegation/RemoteHostAllowsDelegationOfNonExportableCredentials` - **Data type:** string - **Value:** `<enabled/>`

Configure delegation of credentials on the clients

To enable Remote Credential Guard on the clients, you can configure a policy that prevents the delegation of credentials to the remote hosts.

> **Tip**
>
> If you don't want to configure your clients to enforce Remote Credential Guard, you can use the following command to use Remote Credential Guard for a specific RDP session:
>
> **Windows Command Prompt**
>
> ```
> mstsc.exe /remoteGuard
> ```
>
> If the server hosts the RDS Host role, then the command works only if the user is an administrator of the remote host.

The policy can have different values, depending on the level of security you want to enforce:

- **Disabled**: *Restricted Admin* and *Remote Credential Guard* mode aren't enforced and the Remote Desktop Client can delegate credentials to remote devices
- **Require Restricted Admin**: the Remote Desktop Client must use Restricted Admin to connect to remote hosts
- **Require Remote Credential Guard**: Remote Desktop Client must use Remote Credential Guard to connect to remote hosts
- **Restrict credential delegation**: Remote Desktop Client must use Restricted Admin or Remote Credential Guard to connect to remote hosts. In this configuration, Remote Credential Guard is preferred, but it uses Restricted Admin mode (if supported) when Remote Credential Guard can't be used

> **Note**
>
> When *Restrict Credential Delegation* is enabled, the `/restrictedAdmin` switch will be ignored. Windows enforces the policy configuration instead and uses Remote Credential Guard.

To configure your clients, you can use:

- Microsoft Intune/MDM
- Group policy

The following instructions provide details about how to configure your devices. Select the option that best suits your needs.

Intune/CSP

To configure devices with Microsoft Intune, create a Settings catalog policy and use the following settings:

Expand table

Category	Setting name	Value
Administrative Templates > **System** > **Credentials Delegation**	Restrict delegation of credentials to remote servers	Select **Enabled** and in the dropdown, select one of the options: - **Restrict Credential Delegation** - **Require Remote Credential Guard**

Assign the policy to a group that contains as members the devices or users that you want to configure.

Alternatively, you can configure devices using a custom policy with the Policy CSP.

Expand table

Setting
- **OMA-URI:** `./Device/Vendor/MSFT/Policy/Config/ADMX_CredSsp/RestrictedRemoteAdministration` - **Data type:** string - **Value:** `<enabled/><data id="RestrictedRemoteAdministrationDrop" value="2"/>` Possible values for `RestrictedRemoteAdministrationDrop` are: - `0`: Disabled - `1`: Require Restricted Admin - `2`: Require Remote Credential Guard - `3`: Restrict credential delegation

User experience

Once a client receives the policy, you can connect to the remote host using Remote Credential Guard by opening the Remote Desktop Client (`mstsc.exe`). The user is automatically authenticated to the remote host:

https://learn-video.azurefd.net/vod/player?id=39cc96a2-5193-48be-a4f3-d491571fd9a1&locale=en-us&embedUrl=%2Fwindows%2Fsecurity%2Fidentity-protection%2Fremote-credential-guard

> **Note**
>
> The user must be authorized to connect to the remote server using the Remote Desktop protocol, for example by being a member of the Remote Desktop Users local group on the remote host.

Remote Desktop connections and helpdesk support scenarios

For helpdesk support scenarios in which personnel require administrative access via Remote Desktop sessions, it isn't recommended the use of Remote Credential Guard. If an RDP session is initiated to an already compromised client, the attacker could use that open channel to create sessions on the user's behalf. The attacker can access any of the user's resources for a limited time after the session disconnects.

We recommend using Restricted Admin mode option instead. For helpdesk support scenarios, RDP connections should only be initiated using the `/RestrictedAdmin` switch. This helps to ensure that credentials and other user resources aren't exposed to compromised remote hosts. For more information, see Mitigating Pass-the-Hash and Other Credential Theft v2.

To further harden security, we also recommend that you implement Windows Local Administrator Password Solution (LAPS), which automates local administrator password management. LAPS mitigates the risk of lateral escalation and other cyberattacks facilitated when customers use the same administrative local account and password combination on all their computers.

For more information about LAPS, see What is Windows LAPS.

Considerations

Here are some considerations for Remote Credential Guard:

- Remote Credential Guard doesn't support compound authentication. For example, if you're trying to access a file server from a remote host that requires a device claim, access is denied

- Remote Credential Guard can be used only when connecting to a device that is joined to an Active Directory domain. It can't be used when connecting to remote devices joined to Microsoft Entra ID
- Remote Credential Guard can be used from a Microsoft Entra joined client to connect to an Active Directory joined remote host, as long as the client can authenticate using Kerberos
- Remote Credential Guard only works with the RDP protocol
- No credentials are sent to the target device, but the target device still acquires Kerberos Service Tickets on its own
- The server and client must authenticate using Kerberos
- Remote Credential Guard is only supported for direct connections to the target machines. It isn't support for connections via Remote Desktop Connection Broker and Remote Desktop Gateway

---

Last updated on 03/29/2026

What is Windows LAPS?

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows 11, ✅ Windows 10

Windows Local Administrator Password Solution (Windows LAPS) is a Windows feature that automatically manages and backs up the password of a local administrator account on your Microsoft Entra-joined or Windows Server Active Directory-joined devices. You also can use Windows LAPS to automatically manage and back up the Directory Services Restore Mode (DSRM) account password on your Windows Server Active Directory domain controllers. An authorized administrator can retrieve the DSRM password and use it.

Windows LAPS supported platforms

Windows LAPS is available on the following OS platforms:

Windows client

- Windows 11 23H2 and later.
- Other versions of Windows client that have received the April 11, 2023 Update or later, including:
  - Windows 11 22H2 - April 11 2023 Update and later.
  - Windows 11 21H2 - April 11 2023 Update and later.
  - Windows 10 - April 11 2023 Update and later.

> **Note**
>
> The **Automatic Account Management** CSP settings require Windows 11 24H2 or later.

Windows Server

- Windows Server 2025 and later.
- Windows Server Annual Channel for Containers, 23H2 and later.
- Other versions of Windows Server that have received the April 11, 2023 Update or later, including:
  - Windows Server 2022 - April 11 2023 Update and later.
  - Windows Server 2019 - April 11 2023 Update and later.

All supported editions of these platforms have been updated with Windows LAPS, including long-term servicing channel (LTSC) editions. The introduction of the Windows LAPS feature

doesn't modify the standard Microsoft product lifecycle policies.

Windows LAPS and Microsoft Entra ID

Windows LAPS with Microsoft Entra ID and Microsoft Intune support is generally available as of October 23, 2023. For more information, see Windows Local Administrator Password Solution with Microsoft Entra ID now generally available! and Windows Local Administrator Password Solution in Microsoft Entra ID.

Benefits of using Windows LAPS

Use Windows LAPS to regularly rotate and manage local administrator account passwords and get these benefits:

- Protection against pass-the-hash and lateral-traversal attacks
- Improved security for remote help desk scenarios
- Ability to sign in to and recover devices that are otherwise inaccessible
- A fine-grained security model (access control lists and optional password encryption) for securing passwords that are stored in Windows Server Active Directory
- Support for the Microsoft Entra role-based access control model for securing passwords that are stored in Microsoft Entra ID

Informational videos

The following videos offer an informative way to see more about the Windows LAPS feature.

Windows Technical Takeoff presentation (November 2022):

https://www.youtube-nocookie.com/embed/jdEDIXm4JgU

Windows Tackling Tech discussion (August 2023):

https://www.youtube-nocookie.com/embed/bcs1gPB4dOQ

Key Windows LAPS scenarios

You can use Windows LAPS for several primary scenarios:

- Back up local administrator account passwords to Microsoft Entra ID (for Microsoft Entra-joined devices)

- Back up local administrator account passwords to Windows Server Active Directory (for Windows Server Active Directory-joined clients and servers)
- Back up DSRM account passwords to Windows Server Active Directory (for Windows Server Active Directory domain controllers)
- Back up local administrator account passwords to Windows Server Active Directory by using legacy Microsoft LAPS

In each scenario, you can apply different policy settings.

Understand device join state restrictions

Whether a device is joined to Microsoft Entra ID or Windows Server Active Directory determines how you can use Windows LAPS.

- Devices that are joined only to Microsoft Entra ID can back up passwords only to Microsoft Entra ID.
- Devices that are joined only to Windows Server Active Directory can back up passwords only to Windows Server Active Directory.
- Devices that are hybrid-joined (joined to both Microsoft Entra ID and Windows Server Active Directory) can back up their passwords either to Microsoft Entra ID or to Windows Server Active Directory.

You can't back up passwords to both Microsoft Entra ID and Windows Server Active Directory.

Windows LAPS doesn't support Microsoft Entra workplace-joined clients.

Set Windows LAPS policy

To set up and manage policy for your Windows LAPS deployment, you have multiple options:

- Windows LAPS configuration service provider (CSP)
- Windows LAPS Group Policy
- Legacy Microsoft LAPS

Manage and monitor Windows LAPS

You also have various options to manage and monitor Windows LAPS.

Options for Windows include:

- The Windows Server Active Directory Users and Computers properties dialog.

- A dedicated event log channel.
- A Windows PowerShell module that's specific to Windows LAPS.

Entra-based monitoring and reporting solutions are available when you back up passwords to Microsoft Entra ID.

Deprecation of the legacy Microsoft LAPS product

> **Important**
>
> The legacy Microsoft LAPS product is deprecated as of Windows 11 23H2 and later. Installation of the legacy Microsoft LAPS Microsoft Installer (MSI) package is blocked on newer OS versions, and Microsoft no longer considers code changes for the legacy Microsoft LAPS product.
>
> Use Windows LAPS for managing local administrator account passwords. Windows LAPS is available on Windows Server 2019 and later, and on supported Windows 10 and Windows 11 clients.
>
> Microsoft will continue to support the legacy Microsoft LAPS product on older versions of Windows (earlier than Windows 11 23H2) on which it was previously supported. That support will end upon the normal end of support for those OS versions.

Windows LAPS vs. legacy Microsoft LAPS

Windows LAPS inherits many design concepts from legacy Microsoft LAPS. If you're familiar with legacy Microsoft LAPS, many Windows LAPS features are familiar. A key difference is that Windows LAPS is an entirely separate implementation that's native to Windows. Windows LAPS also adds many features that aren't available in legacy Microsoft LAPS. You can use Windows LAPS to back up passwords to Microsoft Entra ID, encrypt passwords in Windows Server Active Directory, and store your password history.

> **Important**
>
> Windows LAPS doesn't require you to install legacy Microsoft LAPS. You can fully deploy and use all Windows LAPS features without installing or referring to legacy Microsoft LAPS. But to help migrate an existing legacy Microsoft LAPS deployment, Windows LAPS offers legacy Microsoft LAPS emulation mode.

> **Important**
>
> The legacy Microsoft LAPS product is deprecated on newer Microsoft OS versions. For more information, see Deprecation of the legacy Microsoft LAPS product.

Support statement

Microsoft released the legacy Microsoft LAPS product in calendar year 2016 on the Microsoft Download Center. Windows LAPS shipped as part of Windows Updates released on April 11, 2023, for the platforms listed in Windows LAPS and Microsoft Entra ID.

Microsoft and its support delivery organization offer assisted support for both Microsoft LAPS and Windows LAPS, including interoperability between the two products.

> **Important**
>
> The legacy Microsoft LAPS product is deprecated on newer Microsoft OS versions. For more information, see Deprecation of the legacy Microsoft LAPS product.

We strongly recommend that you begin planning now to migrate your Windows LAPS-capable systems from using legacy Microsoft LAPS over to the Windows LAPS feature. Windows LAPS offers many new security features and improved product servicing.

Questions about limitations and interoperability concerns between third-party local account password management tools and Windows LAPS should be directed to the third-party application developer, not Microsoft.

Licensing requirements

The Windows LAPS feature itself is available for free in all supported Windows platforms.

You can back up passwords to Windows Server Active Directory with no other licensing requirements.

You can back up passwords to Microsoft Entra ID with a Microsoft Entra ID Free or higher license.

Other Entra-related or Intune-related features can have other licensing requirements.

Submit feedback

Want to send us feedback? Feel free to submit document-specific questions via the feedback links at the bottom of this page.

You can also submit feedback and other requests via the Windows LAPS feedback Tech Community page.

If your feedback is specific to the Microsoft Entra ID-related or Intune-related LAPS functionality, you can submit feedback via the Microsoft Entra feedback forum.

If you aren't sure where your feedback should go, submit it by using any of these options.

See also

- Introducing Windows Local Administrator Password Solution with Microsoft Entra ID
- Windows Local Administrator Password Solution in Microsoft Entra ID
- Windows Local Administrator Password Solution with Microsoft Entra ID now generally available!
- Microsoft Intune support for Windows LAPS
- LAPS CSP
- Windows LAPS troubleshooting guidance
- LAPS PowerShell module reference
- Legacy Microsoft LAPS

Next steps

- Key concepts in Windows LAPS
- Get started with Windows LAPS and Windows Server Active Directory
- Get started with Windows LAPS and Microsoft Entra ID
- Get started with Windows LAPS in legacy Microsoft LAPS emulation mode

---

Last updated on 12/15/2025

Enhanced Phishing Protection in Microsoft Defender SmartScreen

Applies to: ✅ Windows 11, version 22H2

Starting in Windows 11, version 22H2, Enhanced Phishing Protection in Microsoft Defender SmartScreen helps protect Microsoft school or work passwords against phishing and unsafe usage on sites and apps.

If a user signs into Windows using a password, Enhanced Phishing Protection works alongside Windows security protections, and helps protect typed work or school password used to sign into Windows 11 in these ways:

- If users type or paste their work or school password on any browser, into a site deemed malicious by Microsoft Defender SmartScreen, Enhanced Phishing Protection alerts them. It also alerts them to change their password so attackers can't gain access to their account.
- Reusing work or school passwords makes it easy for attackers who compromise a user's password to gain access to their other accounts. Enhanced Phishing Protection can warn users if they reuse their work or school Microsoft account password on sites and apps and alert them to change their password.
- Since it's unsafe to store plaintext passwords in text editors, Enhanced Phishing Protection can warn users if they store their work or school password in Notepad, Word, or any Microsoft 365 Office app, and recommends they delete their password from the file.
- If users type their work or school password into a website or app that SmartScreen finds suspicious, Enhanced Phishing Protection can automatically collect information from that website or app to help identify security threats. For example, the content displayed, sounds played, and application memory.

> ⓘ **Note**
>
> When a user signs in to a device using a Windows Hello for Business PIN or biometric, Enhanced Phishing Protection doesn't alert the user or send events to Microsoft Defender for Endpoint (MDE).

Benefits of Enhanced Phishing Protection in Microsoft Defender SmartScreen

Enhanced Phishing Protection provides robust phishing protections for work or school passwords that are used to sign into Windows 11. The benefits of Enhanced Phishing Protection are:

- **Anti-phishing support:** Phishing attacks trick users through convincing imitations of safe content or through credential harvesting content hosted inside trusted sites and applications. Enhanced Phishing Protection helps protect users from reported phishing sites by evaluating the URLs a site or app is connecting to, along with other characteristics, to determine if they're known to distribute or host unsafe content.

- **Secure operating system integration:** Enhanced Phishing Protection is integrated directly into the Windows 11 operating system, so it can understand users' password entry context (including process connections, URLs, certificate information) in any browser or app. Because Enhanced Phishing Protection has unparalleled insight into what is happening at the OS level, it can identify when users type their work or school password unsafely. If users do use their work or school password unsafely, the feature empowers users to change their password to minimize chances of their compromised credential being weaponized against them.

- **Unparalleled telemetry shared throughout Microsoft's security suite:** Enhanced Phishing Protection is constantly learning from phishing attacks seen throughout the entire Microsoft security stack. It works alongside other Microsoft security products, to provide a layered approach to password security, especially for organizations early in their password-less authentication journey. If your organization uses Microsoft Defender for Endpoint, you can see valuable phishing sensors data in the Microsoft 365 Defender Portal. This portal lets you view Enhanced Phishing Protection alerts and reports for unsafe password usage in your environment.

- **Easy management through Group Policy and Microsoft Intune:** Enhanced Phishing Protection works with Group Policy and mobile device management (MDM) settings to help you manage your organization's computer settings. Based on how you set up Enhanced Phishing Protection, you can customize which phishing protection scenarios show users warning dialogs. For example, the Service Enabled setting determines whether the Enhanced Phishing Protection service is on or off. The feature is in audit mode if the other settings, which correspond to notification policies, aren't enabled.

Windows edition and licensing requirements

The following table lists the Windows editions that support Enhanced phishing protection with SmartScreen:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Enhanced phishing protection with SmartScreen license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Configure Enhanced Phishing Protection for your organization

Enhanced Phishing Protection can be configured via Microsoft Intune, Group Policy Objects (GPO), or Configuration Service Providers (CSP) with an MDM service. These settings are available to configure your devices using either Microsoft Intune, GPO, or CSP.

Expand table

Setting	Description
Automatic Data Collection	This policy setting determines whether Enhanced Phishing Protection can collect additional information-such as content displayed, sounds played, and application memory-when your users enter their work or school password into a suspicious website or app. This information is used only for security purposes and helps SmartScreen determine whether the website or app is malicious. • If you enable this policy setting, Enhanced Phishing Protection may automatically collect additional content for security analysis from a suspicious website or app when your users enter their work or school password into that website or app. • If you disable this policy setting, Enhanced Phishing Protection won't collect additional content for security analysis when your users enter their work or school password into a suspicious site or app. • If this policy isn't set, Enhanced Phishing Protection automatic data collection honors the end user's settings.
Service Enabled	This policy setting determines whether Enhanced Phishing Protection is in audit mode or off. Users don't see any notifications for any protection scenarios when Enhanced

Setting	Description
	Phishing Protection is in audit mode. In audit mode, Enhanced Phishing Protection captures unsafe password entry events and sends diagnostic data through Microsoft Defender. • If you enable or don't configure this setting, Enhanced Phishing Protection is enabled in audit mode, preventing users to turn it off. • If you disable this policy setting, Enhanced Phishing Protection is off. When off, Enhanced Phishing Protection doesn't capture events, send data, or notify users. Additionally, your users are unable to turn it on.
Notify Malicious	This policy setting determines whether Enhanced Phishing Protection warns your users if they type their work or school password into one of the following malicious scenarios: into a reported phishing site, into a sign-in URL with an invalid certificate, or into an application connecting to either a reported phishing site or a sign-in URL with an invalid certificate • If you enable this policy setting, Enhanced Phishing Protection warns your users if they type their work or school password into one of the malicious scenarios described above and encourages them to change their password. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they type their work or school password into one of the malicious scenarios described above.
Notify Password Reuse	This policy setting determines whether Enhanced Phishing Protection warns your users if they reuse their work or school password. • If you enable this policy setting, Enhanced Phishing Protection warns users if they reuse their work, or school password and encourages them to change it. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they reuse their work or school password.
Notify Unsafe App	This policy setting determines whether Enhanced Phishing Protection warns your users if they type their work or school passwords in Notepad or Microsoft 365 Office Apps. • If you enable this policy setting, Enhanced Phishing Protection warns your users if they store their password in Notepad or Microsoft 365 Office Apps. • If you disable or don't configure this policy setting, Enhanced Phishing Protection doesn't warn users if they store their password in Notepad or Microsoft 365 Office Apps.

Enhanced Phishing Protection allows organizations to add their custom identity provider sign-in URL as a recognized URL. Then Enhanced Phishing Protection doesn't consider Microsoft passwords typed into an internal identity provider (IdP) as unknown or password reuse. Without knowledge of an enterprise's custom identity provider URL, SmartScreen might not have enough information about the URL. If you configure warning dialogs for Enhanced Phishing Protection, it might show an unsafe password usage dialog to the user entering their Microsoft password into the URL.

To add your organization's custom sign-in URL to Enhanced Phishing Protection, configure the `EnableWebSignIn` policy in the Authentication Policy CSP. For more information, see Web sign-in for Windows.

Follow these instructions to configure your devices using either Microsoft Intune, GPO, or CSP.

Intune

To configure devices using Microsoft Intune, create a **Settings catalog** policy, and use the settings listed under the category `SmartScreen > Enhanced Phishing Protection`:

- Automatic Data Collection
- Service Enabled
- Notify Malicious
- Notify Password Reuse
- Notify Unsafe App

Assign the policy to a security group that contains as members the devices or users that you want to configure.

Recommended settings for your organization

By default, Enhanced Phishing Protection is deployed in audit mode, preventing notifications to the users for any protection scenarios. In audit mode, Enhanced Phishing Protection captures unsafe password entry events and sends diagnostic data through Microsoft Defender. Users aren't warned if they enter their work or school password into a phishing site, if they reuse their password, or if they unsafely store their password in applications. Because of this possibility, we recommend that you configure Enhanced Phishing Protection to warn users during all protection scenarios.

Expand table

Setting	Default Value	Recommendation
Automatic Data Collection	**Enabled** for domain joined devices or devices enrolled with MDM. **Disabled** for all other devices.	**Enabled**: Turns on collection of additional content for security analysis from a suspicious website or app to improve Microsoft's threat intelligence. This information is used only for security purposes and helps SmartScreen determine whether the website or app is malicious.
Service Enabled	**Enabled**	**Enabled**: Turns on Enhanced Phishing Protection in audit mode, which captures work or school password entry events and sends diagnostic data but doesn't show any notifications to your users.
Notify Malicious	**Disabled** for devices onboarded to MDE.	**Enabled**: Turns on Enhanced Phishing Protection notifications when users type their work or school password into one of the

Setting	Default Value	Recommendation
	Enabled for all other devices.	previously described malicious scenarios and encourages them to change their password.
Notify Password Reuse	**Disabled**	**Enabled**: Turns on Enhanced Phishing Protection notifications when users reuse their work or school password and encourages them to change their password.
Notify Unsafe App	**Disabled**	**Enabled**: Turns on Enhanced Phishing Protection notifications when users type their work or school passwords in Notepad and Microsoft 365 Office Apps.

To better help you protect your organization, we recommend turning on and using these specific Microsoft Defender SmartScreen settings.

Intune

Expand table

Settings catalog element	Recommended value
Automatic Data Collection	**Enabled**
Service Enabled	**Enabled**
Notify Malicious	**Enabled**
Notify Password Reuse	**Enabled**
Notify Unsafe App	**Enabled**

Related articles

- SmartScreen frequently asked questions
- WebThreatDefense CSP
- Threat protection

Last updated on 04/15/2025

Access control overview

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes access control in Windows, which is the process of authorizing users, groups, and computers to access objects on the network or computer. Key concepts that make up access control are:

- permissions
- ownership of objects
- inheritance of permissions
- user rights
- object auditing

Computers that are running a supported version of Windows can control the use of system and network resources through the interrelated mechanisms of authentication and authorization. After a user is authenticated, the Windows operating system uses built-in authorization and access control technologies to implement the second phase of protecting resources: determining if an authenticated user has the correct permissions to access a resource.

Shared resources are available to users and groups other than the resource's owner, and they need to be protected from unauthorized use. In the access control model, users and groups (also referred to as security principals) are represented by unique security identifiers (SIDs). They're assigned rights and permissions that inform the operating system what each user and group can do. Each resource has an owner who grants permissions to security principals. During the access control check, these permissions are examined to determine which security principals can access the resource and how they can access it.

Security principals perform actions (which include Read, Write, Modify, or Full control) on objects. Objects include files, folders, printers, registry keys, and Active Directory Domain Services (AD DS) objects. Shared resources use access control lists (ACLs) to assign permissions. This enables resource managers to enforce access control in the following ways:

- Deny access to unauthorized users and groups
- Set well-defined limits on the access that is provided to authorized users and groups

Object owners generally grant permissions to security groups rather than to individual users. Users and computers that are added to existing groups assume the permissions of that group. If an object (such as a folder) can hold other objects (such as subfolders and files), it's called a container. In a hierarchy of objects, the relationship between a container and its content is expressed by referring to the container as the parent. An object in the container is referred to as the child, and the child inherits the access control settings of the parent. Object owners often define permissions for container objects, rather than individual child objects, to ease access control management.

This content set contains:

- Dynamic Access Control Overview
- Security identifiers
- Security Principals
  - Local Accounts
  - Active Directory Accounts
  - Microsoft Accounts
  - Service Accounts
  - Active Directory Security Groups

Windows edition and licensing requirements

The following table lists the Windows editions that support Access Control (ACL/SACL):

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Access Control (ACL/SACL) license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

Practical applications

Administrators who use the supported version of Windows can refine the application and management of access control to objects and subjects to provide the following security:

- Protect a greater number and variety of network resources from misuse
- Provision users to access resources in a manner that is consistent with organizational policies and the requirements of their jobs
- Enable users to access resources from various devices in numerous locations
- Update users' ability to access resources regularly as an organization's policies change or as users' jobs change
- Account for a growing number of use scenarios (such as access from remote locations or from a rapidly expanding variety of devices, such as tablet computers and mobile phones)
- Identify and resolve access issues when legitimate users are unable to access resources that they need to perform their jobs

Permissions

Permissions define the type of access that is granted to a user or group for an object or object property. For example, the Finance group can be granted Read and Write permissions for a file named Payroll.dat.

By using the access control user interface, you can set NTFS permissions for objects such as files, Active Directory objects, registry objects, or system objects such as processes. Permissions can be granted to any user, group, or computer. It's a good practice to assign permissions to groups because it improves system performance when verifying access to an object.

For any object, you can grant permissions to:

- Groups, users, and other objects with security identifiers in the domain.
- Groups and users in that domain and any trusted domains.
- Local groups and users on the computer where the object resides.

The permissions attached to an object depend on the type of object. For example, the permissions that can be attached to a file are different from those that can be attached to a registry key. Some permissions, however, are common to most types of objects. These common permissions are:

- Read
- Modify
- Change owner

- Delete

When you set permissions, you specify the level of access for groups and users. For example, you can let one user read the contents of a file, let another user make changes to the file, and prevent all other users from accessing the file. You can set similar permissions on printers so that certain users can configure the printer and other users can only print.

When you need to change the permissions on a file, you can run Windows Explorer, right-click the file name, and select **Properties**. On the **Security** tab, you can change permissions on the file. For more information, see Managing Permissions.

> ⓘ **Note**
>
> Another kind of permissions, called share permissions, is set on the Sharing tab of a folder's **Properties** page or by using the Shared Folder Wizard. For more information, see Share and NTFS Permissions on a File Server.

Ownership of objects

An owner is assigned to an object when that object is created. By default, the owner is the creator of the object. No matter what permissions are set on an object, the owner of the object can always change the permissions. For more information, see Manage Object Ownership.

Inheritance of permissions

Inheritance allows administrators to easily assign and manage permissions. This feature automatically causes objects within a container to inherit all the inheritable permissions of that container. For example, the files within a folder inherit the permissions of the folder. Only permissions marked to be inherited are inherited.

User rights

User rights grant specific privileges and sign-in rights to users and groups in your computing environment. Administrators can assign specific rights to group accounts or to individual user accounts. These rights authorize users to perform specific actions, such as signing in to a system interactively or backing up files and directories.

User rights are different from permissions because user rights apply to user accounts, and permissions are associated with objects. Although user rights can apply to

individual user accounts, user rights are best administered on a group account basis. There's no support in the access control user interface to grant user rights. However, user rights assignment can be administered through **Local Security Settings**.

For more information about user rights, see User Rights Assignment.

Object auditing

With administrator's rights, you can audit users' successful or failed access to objects. You can select which object access to audit by using the access control user interface, but first you must enable the audit policy by selecting **Audit object access** under **Local Policies** in **Local Security Settings**. You can then view these security-related events in the Security log in Event Viewer.

For more information about auditing, see Security Auditing Overview.

See also

For more information about access control and authorization, see Access Control and Authorization Overview.

Feedback

Was this page helpful? Yes No

Provide product feedback

Local Accounts

Article • 04/07/2025 •

Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the default local user accounts for Windows operating systems, and how to manage the built-in accounts.

About local user accounts

Local user accounts are defined locally on a device, and can be assigned rights and permissions on the device only. Local user accounts are security principals that are used to secure and manage access to the resources on a device, for services or users.

Default local user accounts

The *default local user accounts* are built-in accounts that are created automatically when the operating system is installed. The default local user accounts can't be removed or deleted and don't provide access to network resources.

Default local user accounts are used to manage access to the local device's resources based on the rights and permissions that are assigned to the account. The default local user accounts, and the local user accounts that you create, are located in the *Users* folder. The Users folder is located in the Local Users and Groups folder in the local *Computer Management* Microsoft Management Console (MMC). *Computer Management* is a collection of administrative tools that you can use to manage a local or remote device.

Default local user accounts are described in the following sections. Expand each section for more information.

Administrator

The default local Administrator account is a user account for system administration. Every computer has an Administrator account (SID S-1-5-*domain*-500, display name Administrator). The Administrator account is the first account that is created during the Windows installation.

The Administrator account has full control of the files, directories, services, and other resources on the local device. The Administrator account can create other local users,

assign user rights, and assign permissions. The Administrator account can take control of local resources at any time by changing the user rights and permissions.

The default Administrator account can't be deleted or locked out, but it can be renamed or disabled.

Windows setup disables the built-in Administrator account and creates another local account that is a member of the Administrators group.

Members of the Administrators groups can run apps with elevated permissions without using the *Run as Administrator* option. Fast User Switching is more secure than using `runas` or different-user elevation.

Account group membership

By default, the Administrator account is a member of the Administrators group. It's a best practice to limit the number of users in the Administrators group because members of the Administrators group have Full Control permissions on the device.

The Administrator account can't be removed from the Administrators group.

Security considerations

Because the Administrator account is known to exist on many versions of the Windows operating system, it's a best practice to disable the Administrator account when possible to make it more difficult for malicious users to gain access to the server or client computer.

You can rename the Administrator account. However, a renamed Administrator account continues to use the same automatically assigned security identifier (SID), which can be discovered by malicious users. For more information about how to rename or disable a user account, see Disable or activate a local user account and Rename a local user account.

As a security best practice, use your local (non-Administrator) account to sign in and then use **Run as administrator** to accomplish tasks that require a higher level of rights than a standard user account. Don't use the Administrator account to sign in to your computer unless it's entirely necessary. For more information, see Run a program with administrative credentials.

Group Policy can be used to control the use of the local Administrators group automatically. For more information about Group Policy, see Group Policy Overview.

> **Important**
>
> - Blank passwords are not allowed
> - Even when the Administrator account is disabled, it can still be used to gain access to a computer by using safe mode. In the Recovery Console or in safe mode, the Administrator account is automatically enabled. When normal operations are resumed, it's disabled.

Guest

The Guest account lets occasional or one-time users, who don't have an account on the computer, temporarily sign in to the local server or client computer with limited user rights. By default, the Guest account is disabled and has a blank password. Since the Guest account can provide anonymous access, it's considered a security risk. For this reason, it's a best practice to leave the Guest account disabled, unless its use is necessary.

Guest account group membership

By default, the Guest account is the only member of the default Guests group `SID S-1-5-32-546`, which lets a user sign in to a device.

Guest account security considerations

When enabling the Guest account, only grant limited rights and permissions. For security reasons, the Guest account shouldn't be used over the network and made accessible to other computers.

In addition, the guest user in the Guest account shouldn't be able to view the event logs. After the Guest account is enabled, it's a best practice to monitor the Guest account frequently to ensure that other users can't use services and other resources. This includes resources that were unintentionally left available by a previous user.

HelpAssistant

The HelpAssistant account is a default local account that is enabled when a Remote Assistance session is run. This account is automatically disabled when no Remote Assistance requests are pending.

HelpAssistant is the primary account that is used to establish a Remote Assistance session. The Remote Assistance session is used to connect to another computer running the Windows operating system, and it's initiated by invitation. For solicited remote assistance, a user sends an invitation from their computer, through e-mail or as a file, to a person who can provide assistance. After the user's invitation for a Remote Assistance session is accepted, the default HelpAssistant account is automatically created to give the person who provides assistance limited access to the computer. The HelpAssistant account is managed by the Remote Desktop Help Session Manager service.

HelpAssistant account security considerations

The SIDs that pertain to the default HelpAssistant account include:

- SID: `S-1-5-<domain>-13`, display name *Terminal Server User*. This group includes all users who sign in to a server with Remote Desktop Services enabled.
- SID: `S-1-5-<domain>-14`, display name *Remote Interactive Logon*. This group includes all users who connect to the computer by using a remote desktop connection. This group is a subset of the Interactive group. Access tokens that contain the Remote Interactive Logon SID also contain the Interactive SID.

For the Windows Server operating system, Remote Assistance is an optional component that isn't installed by default. You must install Remote Assistance before it can be used.

For details about the HelpAssistant account attributes, see the following table.

HelpAssistant account attributes

Expand table

Attribute	Value
Well-Known SID/RID	`S-1-5-<domain>-13 (Terminal Server User), S-1-5-<domain>-14 (Remote Interactive Logon)`
Type	User
Default container	`CN=Users, DC=<domain>`
Default members	None
Default member of	Domain Guests Guests
Protected by ADMINSDHOLDER?	No

Attribute	Value
Safe to move out of default container?	Can be moved out, but we don't recommend it.
Safe to delegate management of this group to non-Service admins?	No

DefaultAccount

The DefaultAccount account, also known as the Default System Managed Account (DSMA), is a well-known user account type. DefaultAccount can be used to run processes that are either multi-user aware or user-agnostic.

The DSMA is disabled by default on the desktop editions and on the Server operating systems with the desktop experience.

The DSMA has a well-known RID of `503`. The security identifier (SID) of the DSMA will thus have a well-known SID in the following format: `S-1-5-21-\<ComputerIdentifier>-503`.

The DSMA is a member of the well-known group **System Managed Accounts Group**, which has a well-known SID of `S-1-5-32-581`.

The DSMA alias can be granted access to resources during offline staging even before the account itself is created. The account and the group are created during first boot of the machine within the Security Accounts Manager (SAM).

How Windows uses the DefaultAccount

From a permission perspective, the DefaultAccount is a standard user account. The DefaultAccount is needed to run multi-user-manifested-apps (MUMA apps). MUMA apps run all the time and react to users signing in and signing out of the devices. Unlike Windows Desktop where apps run in context of the user and get terminated when the user signs off, MUMA apps run by using the DSMA.

MUMA apps are functional in shared session SKUs such as Xbox. For example, Xbox shell is a MUMA app. Today, Xbox automatically signs in as Guest account and all apps run in this context. All the apps are multi-user-aware and respond to events fired by user manager. The apps run as the Guest account.

Similarly, Phone auto logs in as a *DefApps* account, which is akin to the standard user account in Windows but with a few extra privileges. Brokers, some services and apps run as this account.

In the converged user model, the multi-user-aware apps and multi-user-aware brokers will need to run in a context different from that of the users. For this purpose, the system creates DSMA.

How the DefaultAccount is created on domain controllers

If the domain was created with domain controllers running Windows Server 2016, the DefaultAccount exists on all domain controllers in the domain. If the domain was created with domain controllers running an earlier version of Windows Server, the DefaultAccount is created after the PDC Emulator role is transferred to a domain controller that runs Windows Server 2016. The DefaultAccount is then replicated to all other domain controllers in the domain.

Recommendations for managing the Default Account (DSMA)

Microsoft doesn't recommend changing the default configuration, where the account is disabled. There's no security risk with having the account in the disabled state. Changing the default configuration could hinder future scenarios that rely on this account.

Default local system accounts

SYSTEM

The *SYSTEM* account is used by the operating system and by services running under Windows. There are many services and processes in the Windows operating system that need the capability to sign in internally, such as during a Windows installation. The SYSTEM account was designed for that purpose, and Windows manages the SYSTEM account's user rights. It's an internal account that doesn't show up in User Manager, and it can't be added to any groups.

On the other hand, the SYSTEM account does appear on an NTFS file system volume in File Manager in the **Permissions** portion of the **Security** menu. By default, the SYSTEM account is granted Full Control permissions to all files on an NTFS volume. Here the SYSTEM account has the same functional rights and permissions as the Administrator account.

> ⓘ **Note**
>
> To grant the account Administrators group file permissions does not implicitly give permission to the SYSTEM account. The SYSTEM account's permissions can be

removed from a file, but we do not recommend removing them.

NETWORK SERVICE

The *NETWORK SERVICE* account is a predefined local account used by the service control manager (SCM). A service that runs in the context of the NETWORK SERVICE account presents the computer's credentials to remote servers. For more information, see NetworkService Account.

LOCAL SERVICE

The *LOCAL SERVICE* account is a predefined local account used by the service control manager. It has minimum privileges on the local computer and presents anonymous credentials on the network. For more information, see LocalService Account.

How to manage local user accounts

The default local user accounts, and the local user accounts you create, are located in the Users folder. The Users folder is located in Local Users and Groups. For more information about creating and managing local user accounts, see Manage Local Users.

You can use Local Users and Groups to assign rights and permissions on only the local server to limit the ability of local users and groups to perform certain actions. A right authorizes a user to perform certain actions on a server, such as backing up files and folders or shutting down a server. An access permission is a rule that is associated with an object, usually a file, folder, or printer. It regulates which users can have access to an object on the server and in what manner.

You can't use Local Users and Groups on a domain controller. However, you can use Local Users and Groups on a domain controller to target remote computers that aren't domain controllers on the network.

> **Note**
>
> You use Active Directory Users and Computers to manage users and groups in Active Directory.

You can also manage local users by using NET.EXE USER and manage local groups by using NET.EXE LOCALGROUP, or by using various PowerShell cmdlets and other scripting technologies.

Restrict and protect local accounts with administrative rights

An administrator can use many approaches to prevent malicious users from using stolen credentials such as a stolen password or password hash, for a local account on one computer from being used to authenticate on another computer with administrative rights. This is also called *lateral movement*.

The simplest approach is to sign in to your computer with a standard user account, instead of using the Administrator account for tasks. For example, use a standard account to browse the Internet, send email, or use a word processor. When you want to perform administrative tasks such as installing a new program or changing a setting that affects other users, you don't have to switch to an Administrator account. You can use User Account Control (UAC) to prompt you for permission or an administrator password before performing the task, as described in the next section.

The other approaches that can be used to restrict and protect user accounts with administrative rights include:

- Enforce local account restrictions for remote access
- Deny network logon to all local Administrator accounts
- Create unique passwords for local accounts with administrative rights

Each of these approaches is described in the following sections.

> **Note**
>
> These approaches do not apply if all administrative local accounts are disabled.

Enforce local account restrictions for remote access

User Account Control (UAC) is a security feature that informs you when a program makes a change that requires administrative permissions. UAC works by adjusting the permission level of your user account. By default, UAC is set to notify you when applications try to make changes to your computer, but you can change when UAC notifies you.

UAC makes it possible for an account with administrative rights to be treated as a standard user nonadministrator account until full rights, also called elevation, is requested and approved. For example, UAC lets an administrator enter credentials during a nonadministrator's user session to perform occasional administrative tasks without having to switch users, sign out, or use the *Run as* command.

In addition, UAC can require administrators to specifically approve applications that make system-wide changes before those applications are granted permission to run, even in the administrator's user session.

For example, a default feature of UAC is shown when a local account signs in from a remote computer by using Network logon (for example, by using NET.EXE USE). In this instance, it's issued a standard user token with no administrative rights, but without the ability to request or receive elevation. Consequently, local accounts that sign in by using Network logon can't access administrative shares such as C$, or ADMIN$, or perform any remote administration.

For more information about UAC, see User Account Control.

The following table shows the Group Policy and registry settings that are used to enforce local account restrictions for remote access.

Expand table

No.	Setting	Detailed Description
	Policy location	Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options
1	Policy name	User Account Control: Admin Approval Mode for the Built-in Administrator account
	Policy setting	Enabled
2	Policy location	Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options
	Policy name	User Account Control: Run all administrators in Admin Approval Mode
	Policy setting	Enabled
3	Registry key	HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
	Registry value name	LocalAccountTokenFilterPolicy
	Registry value type	DWORD

No.	Setting	Detailed Description
	Registry value data	0

> **Note**
>
> You can also enforce the default for LocalAccountTokenFilterPolicy by using the custom ADMX in Security Templates.

To enforce local account restrictions for remote access

1. Start the **Group Policy Management** Console (GPMC)
2. In the console tree, expand <*Forest*>\Domains<*Domain*>, and then **Group Policy Objects** where *forest* is the name of the forest, and *domain* is the name of the domain where you want to set the Group Policy Object (GPO)
3. In the console tree, right-click **Group Policy Objects** > **New**
4. In the **New GPO** dialog box, type <**gpo_name**>, and > **OK** where *gpo_name* is the name of the new GPO. The GPO name indicates that the GPO is used to restrict local administrator rights from being carried over to another computer
5. In the details pane, right-click <**gpo_name**>, and > **Edit**
6. Ensure that UAC is enabled and that UAC restrictions apply to the default Administrator account by following these steps:
    - Navigate to the **Computer Configuration** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**
    - Select **User Account Control: Run all administrators in Admin Approval Mode** > **Enabled** > **OK**
    - Select **User Account Control: Admin Approval Mode for the Built-in Administrator account** > **Enabled** > **OK**
7. Ensure that the local account restrictions are applied to network interfaces by following these steps:
    - Navigate to *Computer Configuration**Preferences and Windows Settings*, and > **Registry**
    - Right-click **Registry**, and > **New** > **Registry Item**

- In the **New Registry Properties** dialog box, on the **General** tab, change the setting in the **Action** box to **Replace**
- Ensure that the **Hive** box is set to **HKEY_LOCAL_MACHINE**
- Select (...), browse to the following location for **Key Path** > **Select** for: `SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System`
- In the **Value name** area, type `LocalAccountTokenFilterPolicy`
- In the **Value type** box, from the drop-down list, select **REG_DWORD** to change the value
- In the **Value data** box, ensure that the value is set to **0**
- Verify this configuration, and > **OK**

8. Link the GPO to the first **Workstations** organizational unit (OU) by doing the following:

   - Navigate to the `*Forest*\<Domains>\*Domain*\*OU*` path
   - Right-click the **Workstations** > **Link an existing GPO**
   - Select the GPO that you created, and > **OK**

9. Test the functionality of enterprise applications on the workstations in that first OU and resolve any issues caused by the new policy

10. Create links to all other OUs that contain workstations

11. Create links to all other OUs that contain servers

Deny network logon to all local Administrator accounts

Denying local accounts the ability to perform network logons can help prevent a local account password hash from being reused in a malicious attack. This procedure helps to prevent lateral movement by ensuring that stolen credentials for local accounts from a compromised operating system can't be used to compromise other computers that use the same credentials.

> ⓘ **Note**
>
> To perform this procedure, you must first identify the name of the local, default Administrator account, which might not be the default user name "Administrator", and any other accounts that are members of the local Administrators group.

The following table shows the Group Policy settings that are used to deny network logon for all local Administrator accounts.

Expand table

No.	Setting	Detailed Description
	Policy location	Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment
1	Policy name	Deny access to this computer from the network
	Policy setting	Local account and member of Administrators group
2	Policy location	Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment
	Policy name	Deny log on through Remote Desktop Services
	Policy setting	Local account and member of Administrators group

To deny network logon to all local administrator accounts

1. Start the **Group Policy Management** Console (GPMC)
2. In the console tree, expand <*Forest*>\Domains<*Domain*>, and then **Group Policy Objects**, where *forest* is the name of the forest, and *domain* is the name of the domain where you want to set the Group Policy Object (GPO)
3. In the console tree, right-click **Group Policy Objects**, and > **New**
4. In the **New GPO** dialog box, type <**gpo_name**>, and then > **OK** where *gpo_name* is the name of the new GPO indicates that it's being used to restrict the local administrative accounts from interactively signing in to the computer
5. In the details pane, right-click <**gpo_name**>, and > **Edit**
6. Configure the user rights to deny network logons for administrative local accounts as follows:
7. Navigate to the Computer Configuration\Windows Settings\Security Settings, and > **User Rights Assignment**
8. Double-click **Deny access to this computer from the network**
9. Select **Add User or Group**, type **Local account and member of Administrators group**, and > **OK**

10. Configure the user rights to deny Remote Desktop (Remote Interactive) logons for administrative local accounts as follows:
11. Navigate to Computer Configuration\Policies\Windows Settings and Local Policies, and then select **User Rights Assignment**
12. Double-click **Deny log on through Remote Desktop Services**
13. Select **Add User or Group**, type **Local account and member of Administrators group**, and > **OK**
14. Link the GPO to the first **Workstations** OU as follows:
    - Navigate to the <*Forest*>\Domains<*Domain*>\OU path
    - Right-click the **Workstations** OU, and > **Link an existing GPO**
    - Select the GPO that you created, and > **OK**
15. Test the functionality of enterprise applications on the workstations in that first OU and resolve any issues caused by the new policy
16. Create links to all other OUs that contain workstations
17. Create links to all other OUs that contain servers

> ⓘ **Note**
>
> You might have to create a separate GPO if the user name of the default Administrator account is different on workstations and servers.

Create unique passwords for local accounts with administrative rights

Passwords should be unique per individual account. While it's true for individual user accounts, many enterprises have identical passwords for common local accounts, such as the default Administrator account. This also occurs when the same passwords are used for local accounts during operating system deployments.

Passwords that are left unchanged or changed synchronously to keep them identical add a significant risk for organizations. Randomizing the passwords mitigates "pass-the-hash" attacks by using different passwords for local accounts, which hamper the ability of malicious users to use password hashes of those accounts to compromise other computers.

Passwords can be randomized by:

- Purchasing and implementing an enterprise tool to accomplish this task. These tools are commonly referred to as "privileged password management" tools
- Configuring Local Administrator Password Solution (LAPS) to accomplish this task
- Creating and implementing a custom script or solution to randomize local account passwords

---

Feedback

Was this page helpful? Yes No

Provide product feedback

Microsoft Entra joined devices

Any organization can deploy Microsoft Entra joined devices no matter the size or industry. Microsoft Entra join works even in hybrid environments, enabling access to both cloud and on-premises apps and resources.

Expand table

Microsoft Entra join	Description
Definition	• Joined only to Microsoft Entra ID requiring organizational account to sign in to the device
Primary audience	• Suitable for both cloud-only and hybrid organizations. • Applicable to all users in an organization
Device ownership	• Organization

Operating Systems	• All Windows 11 and Windows 10 devices except Home editions
	• Windows Enterprise multi-session Virtual Machines running in Azure
	• Windows Server 2019 and newer Virtual Machines running in Azure (Server core isn't supported)
	• Apple devices running macOS 13 or newer
	• Linux editions:
	  ◦ Ubuntu 22.04/24.04 LTS
	  ◦ Red Hat Enterprise Linux 8/9 LTS

Provisioning	• Self-service: Windows Out of Box Experience (OOBE) or Settings • Bulk enrollment • Windows Autopilot • (Public preview) Apple Automated Device Enrollment (applies to Apple devices only)
Device management	• Mobile Device Management (example: Microsoft Intune) • Configuration Manager standalone or co-management with Microsoft Intune
Key capabilities	• single sign-on (SSO) to both cloud and on-premises resources • Conditional Access • Self-service Password Reset and Windows Hello PIN reset on lock screen

Device sign in options

The following are the supported sign-in options for Microsoft Entra joined devices. The availability of these options depends on the device's operating system and configuration. For example, Windows Hello for Business requires additional setup and may not be available on all devices.

Expand table

Platform	Password	SmartCard	Microsoft Authenticator phone sign-in	Windows Hello for Business / Platform Credentials	Web Sign-In	FIDO2
Windows 10/11	✅	✅	✅	✅	✅	✅
macOS 13+	✅	✅				
Ubuntu 22.04/24.04 LTS	✅	✅	✅			
RHEL 8/9/10	✅	⚠	✅			

You sign in to Microsoft Entra joined devices using a Microsoft Entra account. Access to resources can be controlled based on your account and Conditional Access policies applied to the device.

Administrators can secure and further control Microsoft Entra joined devices using Mobile Device Management (MDM) tools like Microsoft Intune or in co-management scenarios using Microsoft Configuration Manager. These tools provide a means to enforce organization-required configurations like:

- Requiring storage to be encrypted
- Password complexity
- Software installation
- Software updates

Administrators can make organization applications available to Microsoft Entra joined devices using Configuration Manager to Manage apps from the Microsoft Store for Business and Education.

Microsoft Entra join can be accomplished using self-service options like the Out of Box Experience (OOBE), bulk enrollment, Apple Automated Device Enrollment (public preview), or Windows Autopilot.

Microsoft Entra joined devices can still maintain single sign-on access to on-premises resources when they are on the organization's network. Devices that are Microsoft Entra joined can still authenticate to on-premises servers like file, print, and other applications.

Scenarios

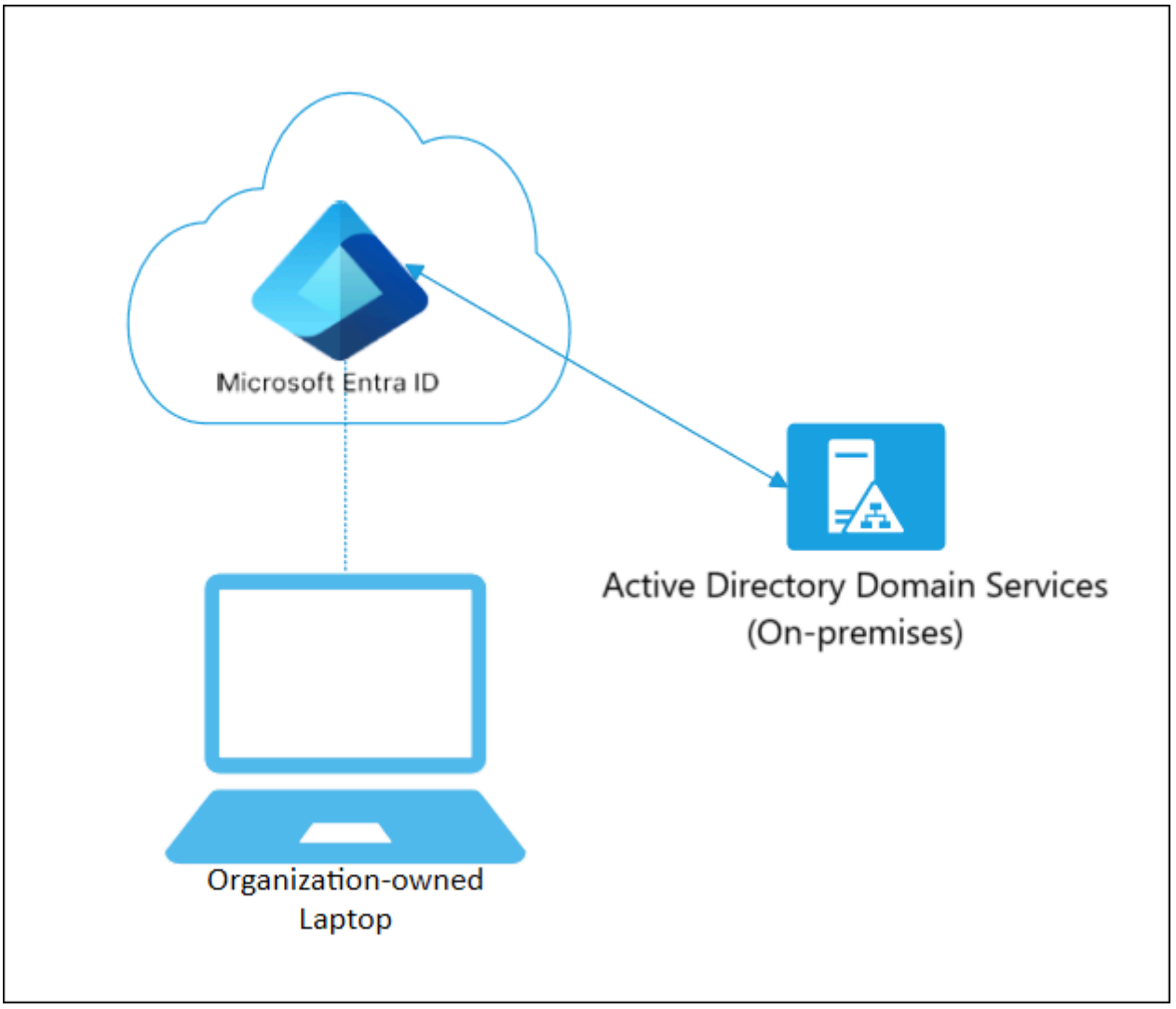
Microsoft Entra join can be used in various scenarios like:

- You want to transition to cloud-based infrastructure using Microsoft Entra ID and MDM like Intune.
- You can't use an on-premises domain join, for example, if you need to get mobile devices such as tablets and phones under control.
- Your users primarily need to access Microsoft 365 or other software as a service (SaaS) apps integrated with Microsoft Entra ID.
- You want to manage a group of users in Microsoft Entra ID instead of in Active Directory. This scenario can apply, for example, to seasonal workers, contractors, or students.
- You want to provide joining capabilities to workers who work from home or are in remote branch offices with limited on-premises infrastructure.

You can configure Microsoft Entra join for all Windows 11 and Windows 10 devices except for Home editions.

The goal of Microsoft Entra joined devices is to simplify:

- Windows and macOS deployments of work-owned devices
- Access to organizational apps and resources from any Windows or macOS device
- Cloud-based management of work-owned devices
- Users to sign in to their devices with their Microsoft Entra ID or synced Active Directory work or school accounts.



Microsoft Entra join can be deployed by using any of the following methods:

- Windows Autopilot
- Bulk deployment
- Self-service experience
- Apple Automated Device Enrollment (public preview)

Related content

- Plan your Microsoft Entra join implementation
- Co-management using Configuration Manager and Microsoft Intune
- How to manage the local administrators group on Microsoft Entra joined devices
- Manage device identities
- Manage stale devices in Microsoft Entra ID
- macOS Platform Single Sign-on (preview)

Last updated on 04/04/2026

Use security baselines to help secure Windows devices you manage with Microsoft Intune

With Microsoft Intune's security baselines, you can rapidly deploy a recommended security posture to your managed Windows devices for Windows security baselines to help you secure and protect your users and devices.

Even though Windows and Windows Server are designed to be secure out-of-the-box, many organizations still want more granular control over their security configurations. To navigate the large number of controls, organizations often seek guidance on configuring various security features. Microsoft provides this guidance in the form of security baselines.

This feature applies to:

- Windows 11
- Windows 10 version 1809 and later

> **Important**
>
> On October 14, 2025, Windows 10 reached end of support and won't receive quality and feature updates. Windows 10 is an **allowed** version in Intune. Devices running this version can still enroll in Intune and use eligible features, but functionality won't be guaranteed and can vary.

Intune security baseline overview

Each security baseline is a group of preconfigured Windows settings that help you apply and enforce granular security settings that the relevant security teams recommend. You can also customize each baseline you deploy to enforce only those settings and values you require. When you create a security baseline profile in Intune, you're creating a template that consists of multiple *device configuration* profiles.

The settings in each baseline are device configuration settings like those found in various Intune policies. Each setting in a baseline works with the configuration service provider for the

relevant product that is present on a managed windows device.

To learn more about why and when you might want to deploy security baselines, see Windows security baselines in the Windows security documentation.

You deploy security baselines to groups of users or devices in Intune, and the settings apply to devices that run Windows. For example, the default configuration of the *Security Baseline for Windows 10 and later* automatically enables BitLocker for removable drives, automatically requires a password to unlock a device, automatically disables basic authentication, and more. When a default value doesn't work for your environment, customize the baseline to apply the settings you need.

> **Note**
>
> In May 2023, Intune began rollout of a new security baseline format for each new baseline release or version update. The new format updates the baseline settings to directly take their name and configuration options from the configuration service provider (CSP) that the baseline setting manages.
>
> Intune also introduced a new process to help you migrate an existing security baseline profile to the newer baseline version. This new behavior is a one-time process that replaces the normal update behavior when you move from the most recent version of an older profile to a newer version that became available in May 2023 or later.

Benefits of using baselines: Security baselines can help you to have an end-to-end secure workflow when working with Microsoft 365. Some of the benefits include:

- By default, each security baseline is configured to meet the best practices and recommendations for the settings that affect security. Intune partners with the same Windows security team that creates group policy security baselines. These recommendations are based on guidance and extensive experience.
- If you're new to Intune, and not sure where to start, security baselines give you an advantage. You can quickly create and deploy a secure profile, knowing that you're helping protect your organization's resources and data.
- If you currently use group policy, migrating to Intune for management is easier with these baselines. These baselines are natively built into Intune, and include a modern management experience.

Default settings across multiple baselines: Separate baseline types, like the MDM security baseline for Windows and the baseline for Microsoft Defender, might include the same settings and use different default values for those settings. Intune can't determine which configuration is best for you, or even in which environment or scenario you might want to use one baselines default recommendation over another:

- It's important to understand the defaults in the baselines you use, and to then modify each baseline to fit your organizational needs.
- By default, each baseline is preconfigured using the recommendations that are specific to the product it applies to.
- In some cases, a configuration that Microsoft Defender recommends might not be the default configuration for similar settings when recommended by Windows. In such situations, it's important to review each setting so you can understand its intent based on the configuration service provider details, and larger scope of the two products.

In almost all scenarios, the default settings in the security baselines are the most restrictive. You should confirm that these settings don't conflict with other policy settings or features in your environment.

For example, the default settings for firewall configuration might not merge connection security rules and local policy rules with MDM rules. So, if you're using delivery optimization, then you should validate these configurations before assigning the security baseline.

> **Note**
>
> Microsoft doesn't recommend using preview versions of security baselines in a production environment. The settings in a preview baseline might change over the course of the preview.

Available security baselines

The following security baseline instances are available for use with Intune. Use the links to view the settings for recent instances of each baseline.

- **Security Baseline for Windows 10 and later**:
  - Version 25H2
  - Version 24H2
  - Version 23H2
  - November 2021

- December 2020
- August 2020

- **Microsoft Defender for Endpoint baseline**:
  (To use this baseline your environment must meet the prerequisites for using Microsoft Defender for Endpoint).
  - Version 24H1
  - Version 6
  - Version 5
  - Version 4
  - Version 3

  > ⓘ **Note**
  >
  > The Microsoft Defender for Endpoint security baseline is optimized for physical devices and is currently not recommended for use on virtual machines (VMs) or VDI endpoints. Certain baseline settings can impact remote interactive sessions on virtualized environments. For more information, see **Increase compliance to the Microsoft Defender for Endpoint security baseline** in the Windows documentation.

- **Microsoft 365 Apps for Enterprise**:
  - Version 2306 (Office baseline) *Released in November 2023*
  - May 2023 (Office baseline)

- **Microsoft Edge Baseline**:
  - Microsoft Edge version 128 - January 2025
  - Microsoft Edge version 117 - November 2023
  - Microsoft Edge version 112 and later - May 2023
  - Microsoft Edge version 85 and later - September 2020
  - Microsoft Edge version 80 and later - April 2020
  - Preview: Microsoft Edge version 77 and later - October 2019

- **HoloLens 2 - Advanced security baseline settings**:
  - Version 1 - HoloLens 2 advanced security - *January 2025*

- **HoloLens 2 - Standard security baseline settings**:
  - Version 1 - HoloLens 2 standard security - *January 2025*

- **Windows 365 Security Baseline**:

- Version 24H1
- November 2021

When a new version for a profile becomes available, settings in profiles based on the older versions become read-only. You can continue to use those older profiles. You can also edit the profile names, description, and assignments, but they don't support a change to their settings configuration and you can't create new profiles based on the older versions.

When you're ready to use the more recent baseline version, you can create new profiles or update your existing profiles to the new version. See Update a baseline profile to the latest version in the *Manage security baseline profiles* article.

About baseline versions and instances

Each new version instance of a baseline can add or remove settings or introduce other changes. For example, as new Windows settings become available with new versions of Windows, *Security Baseline for Windows 10 and later* might receive a new version instance that includes the newest settings.

You can view the list of available baselines in the Microsoft Intune admin center, under **Endpoint security** > **Security baselines**. The list includes:

- The name of each security baseline template.
- How many profiles you have that use that type of baseline.
- How many separate instances (versions) of the baseline type are available.
- A *Last Published* date that identifies when the latest version of the baseline template became available.

To view more information about the baseline versions you use, select a baseline type, like *Security Baseline for Windows 10 and later* to open its *Profiles* pane, and then select **Versions**. Intune displays details about the versions of that baseline that are in use by your profiles. The details include the most recent and current baseline version. You can select a single version to view deeper details about the profiles that use that version.

You can choose to change the version of a baseline that's in use with a given profile. When you change the version, you don't have to create a new baseline profile to take advantage of updated versions. Instead you can select a baseline profile and use the built-in option to change the instance version for that profile to a new one.

Avoid conflicts

You can use one or more of the available baselines in your Intune environment at the same time. You can also use multiple instances of the same security baselines that have different customizations.

When you use multiple security baselines, review the settings in each one to identify when your different baseline configurations introduce conflicting values for the same setting. Because you can deploy security baselines that are designed for different intents, and deploy multiple instances of the same baseline that includes customized settings, you might create configuration conflicts for devices that must be investigated and resolved.

In addition, security baselines often manage the same settings you might set with device configuration profiles or other types of policy. Therefore, remain aware of and consider your other policies and profiles for settings when seeking to avoid or resolve conflicts.

For information that can help you identify and resolve conflicts, see:

- Troubleshoot policies and profiles in Intune
- Monitor your security baselines

Q & A

Why these settings?

The Microsoft security team has years of experience working directly with Windows developers and the security community to create these recommendations. The settings in this baseline are considered the most relevant security-related configuration options. In each new build of Windows, the team adjusts its recommendations based on newly released features.

Is there a difference in the recommendations for Windows security baselines for group policy vs. Intune?

The same Microsoft security team chose and organized the settings for each baseline. Intune includes all the relevant settings in the Intune security baseline. There are some settings in the group policy baseline that are specific to an on-premises domain controller. These settings are excluded from Intune's recommendations. All the other settings are the same.

Are the Intune security baselines CIS or NIST compliant?

Strictly speaking, no. The Microsoft security team consults organizations, such as CIS, to compile its recommendations. However, there isn't a one-to-one mapping between "CIS-

compliant" and Microsoft baselines.

What certifications do Microsoft's security baselines have?

Microsoft continues to publish security baselines for group policies (GPOs) and the Security Compliance Toolkit, as it has for many years. These baselines are used by many organizations. The recommendations in these baselines are from the Microsoft security team's engagement with enterprise customers and external agencies, including the Department of Defense (DoD), National Institute of Standards and Technology (NIST), and more. We share our recommendations and baselines with these organizations. These organizations also have their own recommendations that closely mirror Microsoft's recommendations. As mobile device management (MDM) continues to grow into the cloud, Microsoft created equivalent MDM recommendations of these group policy baselines. Many of these baselines are built into Microsoft Intune, and include compliance reports on users, groups, and devices that follow (or don't follow) the baseline.

Many customers use the Intune baseline recommendations as a starting point, and then customize them to meet their IT and security demands. Microsoft's Windows 10 and later baseline template was the first baseline to release. This baseline is built as a generic infrastructure that allows customers to eventually import other security baselines based on CIS, NIST, and other standards.

Migrating from on-premises Active Directory group policies to a pure cloud solution using Microsoft Entra ID with Microsoft Intune is a journey. To help, use the various tools from the Security Compliance Toolkit that can help you identify cloud-based options from security baselines that can replace your on-premises GPO configurations.

Where can I find details about using or configuring the settings that are available in a security baseline?

Each security baseline manages device configurations by applying the options found in a configuration service provider on a device. For example, settings that apply to Microsoft Defender are taken from the Microsoft Defender CSP. Because Intune is a configuration vehicle for those options and doesn't determine their functionality or scope, the CSP documentation owns the content for how to configure each option.

Within the Intune security baseline policy UI, Intune provides information text that is taken from the source CSP and provides a link to that CSP. In some cases, the CSP might be part of a larger content set that includes proactive guidance that remains beyond the scope of Intune to

include or duplicate in our content. However, Intune does document the list of settings in each security baseline version and its default configuration.

Learn more

- Create security baseline profiles
- Check the status and monitor the baseline and profile
- View the settings in the latest versions of the available baselines:
  - Windows 10 and later - MDM security baseline
  - Microsoft Defender for Endpoint baseline
  - Microsoft 365 Apps for Enterprise security baseline (Office)
  - Microsoft Edge security baseline
  - Windows 365 Security Baseline

Last updated on 04/15/2026

RemoteWipe CSP

Article • 03/12/2025

The RemoteWipe configuration service provider can be used by mobile operators DM server or enterprise management server to remotely reset a device. The RemoteWipe configuration service provider can make the data stored in memory and hard disks difficult to recover if the device is remotely reset after being lost or stolen. Enterprise IT Professionals can update these settings by using the Exchange Server.

Windows edition and licensing requirements

The following table lists the Windows editions that support Remote wipe:

Expand table

Windows Pro	Windows Enterprise	Windows Pro Education/SE	Windows Education
Yes	Yes	Yes	Yes

Remote wipe license entitlements are granted by the following licenses:

Expand table

Windows Pro/Pro Education/SE	Windows Enterprise E3	Windows Enterprise E5	Windows Education A3	Windows Education A5
Yes	Yes	Yes	Yes	Yes

For more information about Windows licensing, see Windows licensing overview.

The following list shows the RemoteWipe configuration service provider nodes:

- ./Device/Vendor/MSFT/RemoteWipe
  - AutomaticRedeployment
    - doAutomaticRedeployment
    - LastError
    - Status
  - doWipe
  - doWipeCloud
  - doWipeCloudPersistProvisionedData
  - doWipeCloudPersistUserData
  - doWipePersistProvisionedData

- doWipePersistUserData
- doWipeProtected

AutomaticRedeployment

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/AutomaticRedeployment
```

Node for the Autopilot Reset operation.

Description framework properties:

Expand table

Property name	Property value
Format	`node`
Access Type	Get

AutomaticRedeployment/doAutomaticRedeployment

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/AutomaticRedeployment/doAutomaticRedeployment
```

Exec on this node triggers Autopilot Reset operation. This works like PC Reset, similar to other existing nodes in this RemoteWipe CSP, except that it keeps the device enrolled in Microsoft Entra ID and MDM, keeps Wi-Fi profiles, and a few other settings like region, language, keyboard.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

AutomaticRedeployment/LastError

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/AutomaticRedeployment/LastError
```

Error value, if any, associated with Automatic Redeployment operation (typically an HRESULT).

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Get
Default Value	0

AutomaticRedeployment/Status

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1809 [10.0.17763] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/AutomaticRedeployment/Status
```

Status value indicating current state of an Automatic Redeployment operation. 0: Never run (not started). The default state. 1: Complete. 10: Reset has been scheduled. 20: Reset is scheduled and waiting for a reboot. 30: Failed during CSP Execute ("Exec" in SyncML). 40: Failed: power requirements not met. 50: Failed: reset internals failed during reset attempt.

Description framework properties:

Expand table

Property name	Property value
Format	`int`
Access Type	Get
Default Value	0

doWipe

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1511 [10.0.10586] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipe
```

Exec on this node will perform a remote wipe on the device. The return status code shows whether the device accepted the Exec command. When used with OMA Client Provisioning, a dummy value of "1" should be included for this element.

A remote reset is equivalent to running **Reset this PC** > **Remove everything** from the **Settings** app, with **Clean Data** set to No and **Delete Files** set to Yes. If a doWipe reset is started and then interrupted, the PC will attempt to roll-back to the pre-reset state. If the PC can't be rolled-back, the recovery environment will take no additional actions and the PC could be in an unusable state and Windows will have to be reinstalled.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipeCloud

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education	✅ Windows 11, version 22H2 [10.0.22621] and later

Scope	Editions	Applicable OS
	✅ IoT Enterprise / IoT Enterprise LTSC	

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipeCloud
```

Exec on this node will perform a cloud-based remote wipe on the device. The return status code shows whether the device accepted the Exec command.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipeCloudPersistProvisionedData

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 11, version 22H2 [10.0.22621] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipeCloudPersistProvisionedData
```

Exec on this node will back up provisioning data to a persistent location and perform a cloud-based remote wipe on the device. The information that was backed up will be restored and applied to the device when it resumes. The return status code shows whether the device accepted the Exec command.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipeCloudPersistUserData

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 11, version 22H2 [10.0.22621] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipeCloudPersistUserData
```

Exec on this node will perform a cloud-based remote reset on the device and persist user accounts and data. The return status code shows whether the device accepted the Exec command.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipePersistProvisionedData

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1511 [10.0.10586] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipePersistProvisionedData
```

Exec on this node will back up provisioning data to a persistent location and perform a remote wipe on the device. The information that was backed up will be restored and applied to the device when it resumes. The return status code shows whether the device accepted the Exec command. When used with OMA Client Provisioning, a dummy value of "1" should be included for this element. The information that was backed up will be restored and applied to the device when it resumes. The return status code shows whether the device accepted the Exec command.

Provisioning packages are persisted in `%SystemDrive%\ProgramData\Microsoft\Provisioning` directory.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipePersistUserData

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1709 [10.0.16299] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipePersistUserData
```

Exec on this node will perform a remote reset on the device and persist user accounts and data. The return status code shows whether the device accepted the Exec command.

This setting is equivalent to selecting **Reset this PC** > **Keep my files** when manually starting a reset from the Settings app.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

doWipeProtected

Expand table

Scope	Editions	Applicable OS
✅ Device ❌ User	✅ Pro ✅ Enterprise ✅ Education ✅ IoT Enterprise / IoT Enterprise LTSC	✅ Windows 10, version 1703 [10.0.15063] and later

Device

```
./Device/Vendor/MSFT/RemoteWipe/doWipeProtected
```

Exec on this node will perform a remote wipe on the device and fully clean the internal drive. In some device configurations, this command may leave the device unable to boot. The return status code shows whether the device accepted the Exec command. The doWipeProtected is functionally similar to doWipe. But unlike doWipe, which can be easily circumvented by simply power cycling the device, doWipeProtected will keep trying to reset the device until it's done.

> **Note**
>
> Because doWipeProtected will clean the partitions in case of failure or interruption, use doWipeProtected in lost/stolen device scenarios.

Description framework properties:

Expand table

Property name	Property value
Format	`chr` (string)
Access Type	Exec

Related articles

Configuration service provider reference

Feedback

Was this page helpful? Yes No

Provide product feedback

Configuration Service Provider

Learn more about the configuration service provider (CSP) policies available on Windows devices.

Configuration service provider reference

REFERENCE

Support scenarios

Device description framework (DDF) files

Contribute to CSP reference

Declared Configuration protocol

Policy CSP

REFERENCE

Policy CSP

Policy DDF file

Policy CSP - Defender

Policy CSP - Update

Policy CSP support scenarios

REFERENCE

ADMX policies

Policies supported by group policy

Policies supported by HoloLens 2

Policies supported by Microsoft Surface Hub

Universal Print documentation

Universal Print is a secure, reliable, cloud-based print solution that simplifies management and lets users print from anywhere.



OVERVIEW
Discover Universal Print



HOW-TO...
Set up Universal Print



WHAT'S NEW
Latest updates (as of 7/18/2024)

Learn about Universal Print

- Why choose Universal Print?
- How does it work?

Get started

- Get access to Universal Print
- Set up Universal Print
- Troubleshoot issues

Integrations

- Printers
- Intune
- macOS
- Print solutions

Management tools



Universal Print in Azure Portal



Microsoft 365 admin center



Universal Print app on macOS



Connector

For Printer Manufacturers

Learn how to build Universal Print ready devices with guides, API references, and code samples.

Universal Print for Government

Universal Print is FedRAMP certified and available to government customers. Learn how to get access.

Security and privacy

Learn how Universal Print stores and processes your data securely.

Windows Autopatch documentation

Windows Autopatch is a cloud service that automates Windows, Microsoft 365 Apps for enterprise, Microsoft Edge, and Microsoft Teams updates to improve security and productivity across your organization.

About Windows Autopatch

OVERVIEW

What is Windows Autopatch?

Windows Autopatch FAQ

Articles and blog posts

TRAINING

[Blog] Get current and stay current with Windows Autopatch

Windows Autopilot documentation

Windows Autopilot and Windows Autopilot device preparation is a collection of technologies used to set up and pre-configure new devices, getting them ready for productive use.

Windows Autopilot solutions

GET STARTED

Compare Windows Autopilot device preparation and Windows Autopilot

Understand Windows Autopilot device preparation

OVERVIEW

Overview of Windows Autopilot device preparation

Requirements

Reporting and monitoring

Understand Windows Autopilot

OVERVIEW

Overview of Windows Autopilot

Requirements

Device registration overview

Enrollment Status Page

Manually register devices

Tutorials

TUTORIAL

Windows Autopilot scenarios

Windows Autopilot device preparation scenarios

Windows privacy documentation

Resources to help you understand data privacy considerations for Windows, including what tools and controls are available to your organization to view and manage Windows data collection.



OVERVIEW
Windows privacy & compliance...



HOW-TO ...
Configure Windows diagnostic data



HOW-TO ...
View Windows diagnostic data

Understand Windows diagnostic data in Windows 11 and Windows 10

Learn more about what Windows diagnostic data is collected under the different settings for the latest versions of Windows 11 and Windows 10.



Windows 11 required diagnostic data

Learn more about Windows 11 diagnostic data events and fields collected.



Windows 10 required diagnostic data

Learn more about Windows 10 diagnostic data events and fields collected.



Optional diagnostic data

Get examples of the types of optional diagnostic data collected from Windows.

View and manage Windows 11 connection endpoints

Manage Windows 11 connection endpoints

Manage connections from Windows to Microsoft services

Additional resources

Microsoft Trust Center

General Data Protection Regulation Summary

Service Trust Portal