Windows Server Security documentation

Windows Server Security provides layers of protection built into the operating system to safeguard against security breaches, help block malicious attacks, and enhance the security of your virtual machines, applications, and data.

An overview of Security in Windows Server

OVERVIEW

Windows Server 2025 Security Book

Datacenter and Private Cloud Security Blog

Addressing emerging threats and landscape shifts

Protecting Your Datacenter and Cloud from Emerging Threats blog post

Secure virtualization with Shielded VMs

VIDEO

Shielded VM Demo

Shielded Virtual Machines in Windows Server video walkthrough

HOW-TO GUIDE

Guarded Fabric Deployment Guide

Shielded VM and Guarded Fabric in Branch Offices

Shielded VM and Guarded Fabric Troubleshooting Guide

Hardening the OS and applications

OVERVIEW

Windows Defender Application Control (WDAC) Deployment Guide

Transport Layer Security Registry Settings

Control Flow Guard

Windows Defender

Secured-core server

VIDEO

Device Guard demo video

Privileged Access Management

OVERVIEW

Securing Privileged Access

Just in Time Administration with Microsoft Identity Manager

Just Enough Administration Article

VIDEO

Just Enough Administration demo video

Credential Protection

OVERVIEW

Protect derived domain credentials with Credential Guard

Protect Remote Desktop credentials with Remote Credential Guard

VIDEO

Credential Guard demo video

Detecting and Responding to Threats

VIDEO

Microsoft Operations Management Suite (OMS)

OMS and Windows Server

OVERVIEW

Microsoft Advanced Threat Analytics

Network Security

OVERVIEW

Datacenter Firewall Overview

What's New in DNS in Windows Server

Mapping security features to compliance regulations

DOWNLOAD

Hyper-V Shielded VMs compliance mapping whitepaper

JEA and JIT compliance mapping whitepaper

Credential Guard compliance mapping whitepaper

Credential Guard compliance mapping whitepaper

Windows Defender compliance mapping whitepaper

Older versions of Windows Server

OVERVIEW

Windows previous versions documentation

Search for specific information

Beginning your General Data Protection Regulation (GDPR) journey for Windows Server

Article • 08/29/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article provides info about the GDPR, including what it is, and the products Microsoft provides to help you to become compliant.

Introduction

On May 25, 2018, a European privacy law is due to take effect that sets a new global bar for privacy rights, security, and compliance.

The General Data Protection Regulation, or GDPR, is fundamentally about protecting and enabling the privacy rights of individuals. The GDPR establishes strict global privacy requirements governing how you manage and protect personal data while respecting individual choice — no matter where data is sent, processed, or stored.

Microsoft and our customers are now on a journey to achieve the privacy goals of the GDPR. At Microsoft, we believe privacy is a fundamental right, and we believe that the GDPR is an important step forward for clarifying and enabling individual privacy rights. But we also recognize that the GDPR will require significant changes by organizations all over the world.

We have outlined our commitment to the GDPR and how we are supporting our customers within the Get GDPR compliant with the Microsoft Cloud blog post by our Chief Privacy Officer Brendon Lynch and the Earning your trust with contractual commitments to the General Data Protection Regulation" blog post by Rich Sauer - Microsoft Corporate Vice President & Deputy General Counsel.

Although your journey to GDPR-compliance may seem challenging, we're here to help you. For specific information about the GDPR, our commitments and how to begin your journey, please visit the GDPR section of the Microsoft Trust Center.

GDPR and its implications

The GDPR is a complex regulation that may require significant changes in how you gather, use and manage personal data. Microsoft has a long history of helping our customers comply with complex regulations, and when it comes to preparing for the GDPR, we are your partner on this journey.

The GDPR imposes rules on organizations that offer goods and services to people in the European Union (EU), or that collect and analyze data tied to EU residents, no matter where those businesses are located. Among the key elements of the GDPR are the following:

- **Enhanced personal privacy rights.** Strengthened data protection for residents of EU by ensuring they have the right to access to their personal data, to correct inaccuracies in that data, to erase that data, to object to processing of their personal data, and to move it.
- **Increased duty for protecting personal data.** Reinforced accountability of organizations that process personal data, providing increased clarity of responsibility in ensuring compliance.
- **Mandatory personal data breach reporting.** Organizations that control personal data are required to report personal data breaches that pose a risk to the rights and freedoms of individuals to their supervisory authorities without undue delay, and, where feasible, no later than 72 hours once they become aware of the breach.

As you might anticipate, the GDPR can have a significant impact on your business, potentially requiring you to update privacy policies, implement and strengthen data protection controls and breach notification procedures, deploy highly transparent policies, and further invest in IT and training. Microsoft Windows 10 can help you effectively and efficiently address some of these requirements.

Personal and sensitive data

As part of your effort to comply with the GDPR, you will need to understand how the regulation defines personal and sensitive data and how those definitions relate to data held by your organization. Based on that understanding you'll be able to discover where that data is created, processed, managed and stored.

The GDPR considers personal data to be any information related to an identified or identifiable natural person. That can include both direct identification (such as, your legal name) and indirect identification (such as, specific information that makes it clear it is you the data references). The GDPR also makes clear that the concept of personal

data includes online identifiers (such as, IP addresses, mobile device IDs) and location data.

The GDPR introduces specific definitions for genetic data (such as, an individual's gene sequence) and biometric data. Genetic data and biometric data along with other sub categories of personal data (personal data revealing racial or ethnic origin, political opinions, religious or philosophical beliefs, or trade union membership: data concerning health; or data concerning a person's sex life or sexual orientation) are treated as sensitive personal data under the GDPR. Sensitive personal data is afforded enhanced protections and generally requires an individual's explicit consent where these data are to be processed.

Examples of info relating to an identified or identifiable natural person (data subject)

This list provides examples of several types of info that will be regulated through GDPR. This is not an exhaustive list.

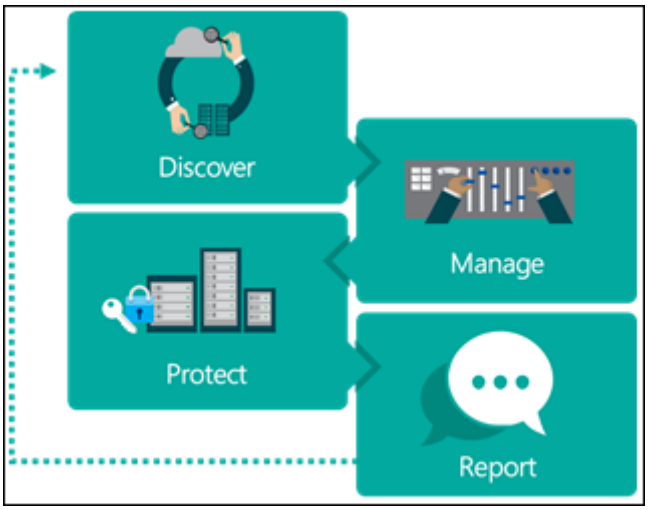
- Name
- Identification number (such as, SSN)
- Location data (such as, home address)
- Online identifier (such as, e-mail address, screen names, IP address, device IDs)
- Pseudonymous data (such as, using a key to identify individuals)
- Genetic data (such as, biological samples from an individual)
- Biometric data (such as, fingerprints, facial recognition)

Getting started on the journey towards GDPR compliance

Given how much is involved to become GDPR-compliant, we strongly recommend that you don't wait to prepare until enforcement begins. You should review your privacy and data management practices now. We recommend that you begin your journey to GDPR compliance by focusing on four key steps:

- **Discover.** Identify what personal data you have and where it resides.
- **Manage.** Govern how personal data is used and accessed.

- **Protect.** Establish security controls to prevent, detect, and respond to vulnerabilities and data breaches.
- **Report.** Act on data requests, report data breaches, and keep required documentation.



For each of the steps, we've outlined example tools, resources, and features in various Microsoft solutions, which can be used to help you address the requirements of that step. While this article isn't a comprehensive "how-to" guide, we've included links for you to find out more details, and more info is available in the GDPR section of the Microsoft Trust Center.

Windows Server security and privacy

The GDPR requires you to implement appropriate technical and organizational security measures to protect personal data and processing systems. In the context of the GDPR, your physical and virtual server environments are potentially processing personal and sensitive data. Processing can mean any operation or set of operations, such as data collection, storage, and retrieval.

Your ability to meet this requirement and to implement appropriate technical security measures must reflect the threats you face in today's increasingly hostile IT environment. Today's security threat landscape is one of aggressive and tenacious threats. In previous years, malicious attackers mostly focused on gaining community recognition through their attacks or the thrill of temporarily taking a system offline. Since then, attacker's motives have shifted toward making money, including holding devices and data hostage until the owner pays the demanded ransom.

Modern attacks increasingly focus on large-scale intellectual property theft; targeted system degradation that can result in financial loss; and now even cyberterrorism that threatens the security of individuals, businesses, and national interests all over the world.

These attackers are typically highly trained individuals and security experts, some of whom are in the employ of nation states that have large budgets and seemingly unlimited human resources. Threats like these require an approach that can meet this challenge.

Not only are these threats a risk to your ability to maintain control of any personal or sensitive data you may have, but they are a material risk to your overall business as well. Consider recent data from McKinsey, Ponemon Institute, Verizon, and Microsoft:

- The average cost of the type of data breach the GDPR will expect you to report is $3.5M.
- 63% of these breaches involve weak or stolen passwords that the GDPR expects you to address.
- Over 300,000 new malware samples are created and spread every day making your task to address data protection even more challenging.

As seen with the recent Ransomware attacks, once called the black plague of the Internet, attackers are going after bigger targets that can afford to pay more, with potentially catastrophic consequences. The GDPR includes penalties that make your systems, including desktops and laptops, that contain personal and sensitive data rich targets indeed.

Two key principles have guided and continue to guide the development of Windows:

- **Security.** The data our software and services store on behalf of our customers should be protected from harm and used or modified only in appropriate ways. Security models should be easy for developers to understand and build into their applications.
- **Privacy.** Users should be in control of how their data is used. Policies for information use should be clear to the user. Users should be in control of when and if they receive information to make best use of their time. It should be easy for users to specify appropriate use of their information including controlling the use of email they send.

Microsoft supports these principles as recently noted by Microsoft's CEO, Satya Nadella,

> "*As the world continues to change and business requirements evolve, some things are consistent: a customer's demand for security and privacy.*"

As you work to comply with the GDPR, understanding the role of your physical and virtual servers in creating, accessing, processing, storing and managing data that may

qualify as personal and potentially sensitive data under the GDPR is important. Windows Server provides capabilities that will help you comply with the GDPR requirements to implement appropriate technical and organizational security measures to protect personal data.

The security posture of Windows Server 2016 isn't a bolt-on; it's an architectural principle. And, it can be best understood in four principals:

- **Protect.** Ongoing focus and innovation on preventative measures; block known attacks and known malware.
- **Detect.** Comprehensive monitoring tools to help you spot abnormalities and respond to attacks faster.
- **Respond.** Leading response and recovery technologies plus deep consulting expertise.
- **Isolate.** Isolate operating system components and data secrets, limit administrator privileges, and rigorously measure host health.

With Windows Server, your ability to protect, detect and defend against the types of attacks that can lead to data breaches is greatly improved. Given the stringent requirements around breach notification within the GDPR, ensuring that your desktop and laptop systems are well defended will lower the risks you face that could result in costly breach analysis and notification.

In the section that follows, you will see how Windows Server provides capabilities that fit squarely in the "Protect" stage of your GDPR compliance journey. These capabilities fall into three protection scenarios:

- **Protect your credentials and limit administrator privileges.** Windows Server 2016 helps to implement these changes, to help prevent your system from being used as a launching point for further intrusions.
- **Secure the operating system to run your apps and infrastructure.** Windows Server 2016 provides layers of protection, which helps to block external attackers from running malicious software or exploiting vulnerabilities.
- **Secure virtualization.** Windows Server 2016 enables secure virtualization, using Shielded Virtual Machines and Guarded Fabric. This helps you encrypt and run your virtual machines on trusted hosts in your fabric, better protecting them from malicious attacks.

These capabilities, discussed in more detail below with references to specific GDPR requirements, are built on top of advanced device protection that helps maintain the

integrity and security of the operating system and data.

A key provision within the GDPR is data protection by design and by default, and helping with your ability to meet this provision are features within Windows 10 such as BitLocker Device Encryption. BitLocker uses the Trusted Platform Module (TPM) technology, which provides hardware-based, security-related functions. This crypto-processor 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.

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 key advantages of using TPM technology are that you can:

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

Additional advanced device protection relevant to your operating without data breaches include Windows Trusted Boot to help maintain the integrity of the system by ensuring malware is unable to start before system defenses.

Windows Server: Supporting your GDPR compliance journey

Key features within Windows Server can help you to efficiently and effectively implement the security and privacy mechanisms the GDPR requires for compliance. While the use of these features will not guarantee your compliance, they will support your efforts to do so.

The server operating system sits at a strategic layer in an organization's infrastructure, affording new opportunities to create layers of protection from attacks that could steal data and interrupt your business. Key aspects of the GDPR such as Privacy by Design, Data Protection, and Access Control need to be addressed within your IT infrastructure at the server level.

Working to help protect the identity, operating system, and virtualization layers, Windows Server 2016 helps block the common attack vectors used to gain illicit access to your systems: stolen credentials, malware, and a compromised virtualization fabric. In addition to reducing business risk, the security components built into Windows Server

2016 help address compliance requirements for key government and industry security regulations.

These identity, operating system, and virtualization protections enable you to better protect your datacenter running Windows Server as a VM in any cloud, and limit the ability of attackers to compromise credentials, launch malware, and remain undetected in your network. Likewise, when deployed as a Hyper-V host, Windows Server 2016 offers security assurance for your virtualization environments through Shielded Virtual Machines and distributed firewall capabilities. With Windows Server 2016, the server operating system becomes an active participant in your datacenter security.

Protect your credentials and limit administrator privileges

Control over access to personal data, and the systems that process that data, is an area with the GDPR that has specific requirements including access by administrators. Privileged identities are any accounts that have elevated privileges, such as user accounts that are members of the Domain Administrators, Enterprise Administrators, local Administrators, or even Power Users groups. Such identities can also include accounts that have been granted privileges directly, such as performing backups, shutting down the system, or other rights listed in the User Rights Assignment node in the Local Security Policy console.

As a general access control principle and in-line with the GDPR, you need to protect these privileged identities from compromise by potential attackers. First, it's important to understand how identities are compromised; then you can plan to prevent attackers from gaining access to these privileged identities.

How do privileged identities get compromised?

Privileged identities can get compromised when organizations don't have guidelines to protect them. The following are examples:

- **More privileges than are necessary.** One of the most common issues is that users have more privileges than are necessary to perform their job function. For example, a user who manages DNS might be an AD administrator. Most often, this is done to avoid the need to configure different administration levels. However, if such an account is compromised, the attacker automatically has elevated privileges.

- **Constantly signed in with elevated privileges.** Another common issue is that users with elevated privileges can use it for an unlimited time. This is very common with IT pros who sign in to a desktop computer using a privileged account, stay signed

in, and use the privileged account to browse the web and use email (typical IT work job functions). Unlimited duration of privileged accounts makes the account more susceptible to attack and increases the odds that the account will be compromised.

- **Social engineering research.** Most credential threats start out by researching the organization and then conducted through social engineering. For example, an attacker may perform an email phishing attack to compromise legitimate accounts (but not necessarily elevated accounts) that have access to an organization's network. The attacker then uses these valid accounts to perform additional research on your network and to identify privileged accounts that can perform administrative tasks.

- **Leverage accounts with elevated privileges.** Even with a normal, non-elevated user account in the network, attackers can gain access to accounts with elevated permissions. One of the more common methods of doing so is by using the Pass-the-Hash or Pass-the-Token attacks. For more information on the Pass-the-Hash and other credential theft techniques, see the resources on the Pass-the-Hash (PtH) page.

There are of course other methods that attackers can use to identify and compromise privileged identities (with new methods being created every day). It is therefore important that you establish practices for users to log on with least-privileged accounts to reduce the ability of attackers to gain access to privileged identities. The sections below outline functionality where Windows Server can mitigate these risks.

Just-in-Time Admin (JIT) and Just Enough Admin (JEA)

While protecting against Pass-the-Hash or Pass-the-Ticket attacks is important, administrator credentials can still be stolen by other means, including social engineering, disgruntled employees, and brute force. Therefore, in addition to isolating credentials as much as possible, you also want a way to limit the reach of administrator-level privileges in case they are compromised.

Today, too many administrator accounts are over-privileged, even if they have only one area of responsibility. For example, a DNS administrator, who requires a very narrow set of privileges to manage DNS servers, is often granted domain admin-level privileges. In addition, because these credentials are granted for perpetuity, there is no limit on how long they can be used.

Every account with unnecessary domain admin-level privileges increases your exposure to attackers seeking to compromise credentials. To minimize the surface area for attack,

you want to provide only the specific set of rights that an admin needs to do the job – and only for the window of time needed to complete it.

Using Just Enough Administration and Just-in-Time Administration, administrators can request the specific privileges they need for the exact window of time required. For a DNS administrator, for example, using PowerShell to enable Just Enough Administration lets you create a limited set of commands that are available for DNS management.

If the DNS administrator needs to make an update to one of her servers, she would request access to manage DNS using Microsoft Identity Manager 2016. The request workflow can include an approval process such as two-factor authentication, which could call the administrator's mobile phone to confirm her identity before granting the requested privileges. Once granted, those DNS privileges provide access to the PowerShell role for DNS for a specific time span.

Imagine this scenario if the DNS admin's credentials were stolen. First, since the credentials have no admin privileges attached to them, the attacker wouldn't be able to gain access to the DNS server – or any other systems – to make any changes. If the attacker tried to request privileges for the DNS server, second-factor authentication would ask them to confirm their identity. Since it isn't likely that the attacker has the DNS admin's mobile phone, authentication would fail. This would lock the attacker out of the system, and alert the IT organization that the credentials might be compromised.

In addition, many organizations use the free Local Administrator Password Solution (LAPS) as a simple yet powerful JIT administration mechanism for their server and client systems. The LAPS capability provides management of local account passwords of domain joined computers. Passwords are stored in Active Directory (AD) and protected by and Access Control List (ACL) so only eligible users can read it or request its reset.

As noted in the Windows Credential Theft Mitigation Guide,

> "*the tools and techniques criminals use to carry out credential theft and reuse attacks improve, malicious attackers are finding it easier to achieve their goals. Credential theft often relies on operational practices or user credential exposure, so effective mitigations require a holistic approach that addresses people, processes, and technology. In addition, these attacks rely on the attacker stealing credentials after compromising a system to expand or persist access, so organizations must contain breaches rapidly by implementing strategies that prevent attackers from moving freely and undetected in a compromised network.*"

An important design consideration for Windows Server was mitigating credential theft—in particular, derived credentials. Credential Guard provides significantly improved

security against derived credential theft and reuse by implementing a significant architectural change in Windows designed to help eliminate hardware-based isolation attacks rather than simply trying to defend against them.

While using Windows Defender Credential Guard, NTLM, and Kerberos derived credentials are protected using virtualization-based security, the credential theft attack techniques and tools used in many targeted attacks are blocked. Malware running in the operating system with administrative privileges cannot extract secrets that are protected by virtualization-based security. While Windows Defender Credential Guard is a powerful mitigation, persistent threat attacks will likely shift to new attack techniques and you should also incorporate Device Guard, as described below, along with other security strategies and architectures.

Windows Defender Credential Guard

Windows Defender Credential Guard uses virtualization-based security to isolate credential information, preventing password hashes or Kerberos tickets from being intercepted. It uses an entirely new isolated Local Security Authority (LSA) process, which is not accessible to the rest of the operating system. All binaries used by the isolated LSA are signed with certificates that are validated before launching them in the protected environment, making Pass-the-Hash type attacks completely ineffective.

Windows Defender Credential Guard uses:

- Virtualization-based security (required). Also required:
  - 64-bit CPU
  - CPU virtualization extensions, plus extended page tables
  - Windows hypervisor
- Secure boot (required)
- TPM 2.0 either discrete or firmware (preferred - provides binding to hardware)

You can use Windows Defender Credential Guard to help protect privileged identities by protecting the credentials and credential derivatives on Windows Server 2016. For more information on Windows Defender Credential Guard requirements, see Protect derived domain credentials with Windows Defender Credential Guard.

Windows Defender Remote Credential Guard

Windows Defender Remote Credential Guard on Windows Server 2016 and Windows 10 Anniversary Update also helps protect credentials for users with remote desktop connections. Previously, anyone using Remote Desktop Services would have to log on to their local machine and then be required to log on again when they performed a remote connection to their target machine. This second login would pass credentials to the target machine, exposing them to Pass-the-Hash or Pass-the-Ticket attacks.

With Windows Defender Remote Credential Guard, Windows Server 2016 implements single sign-on for Remote Desktop sessions, eliminating the requirement to re-enter your username and password. Instead, it leverages the credentials that you've already used to log on to your local machine. To use Windows Defender Remote Credential Guard, the Remote Desktop client and server must meet the following requirements:

- Must be joined to an Active Directory domain and be in the same domain or a domain with a trust relationship.
- Must use Kerberos authentication.
- Must be running at least Windows 10 version 1607 or Windows Server 2016.
- The Remote Desktop classic Windows app is required. The Remote Desktop Universal Windows Platform app doesn't support Windows Defender Remote Credential Guard.

You can enable Windows Defender Remote Credential Guard by using a registry setting on the Remote Desktop server and Group Policy or a Remote Desktop Connection parameter on the Remote Desktop client. For more information on enabling Windows Defender Remote Credential Guard, see Protect Remote Desktop credentials with Windows Defender Remote Credential Guard. As with Windows Defender Credential Guard, you can use Windows Defender Remote Credential Guard to help protect privileged identities on Windows Server 2016.

Secure the operating system to run your apps and infrastructure

Preventing cyber threats also requires finding and blocking malware and attacks that seek to gain control by subverting the standard operating practices of your infrastructure. If attackers can get an operating system or application to run in a non-predetermined, non-viable way, they are likely using that system to take malicious actions. Windows Server 2016 provides layers of protection that block external attackers running malicious software or exploiting vulnerabilities. The operating system takes an

active role in protecting infrastructure and applications by alerting administrators to activity that indicates a system has been breached.

Windows Defender Device Guard

Windows Server 2016 includes Windows Defender Device Guard to ensure that only trusted software can be run on the server. Using virtualization-based security, it can limit what binaries can run on the system based on the organization's policy. If anything, other than the specified binaries tries to run, Windows Server 2016 blocks it and logs the failed attempt so that administrators can see that there has been a potential breach. Breach notification is a critical part of the requirements for GDPR compliance.

Windows Defender Device Guard is also integrated with PowerShell so that you can authorize which scripts can run on your system. In earlier versions of Windows Server, administrators could bypass code integrity enforcement by simply deleting the policy from the code file. With Windows Server 2016, you can configure a policy that is signed by your organization so that only a person with access to the certificate that signed the policy can change the policy.

Control Flow Guard

Windows Server 2016 also includes built-in protection against some classes of memory corruption attacks. Patching your servers is important, but there is always a chance that malware could be developed for a vulnerability that has not yet been identified. Some of the most common methods for exploiting these vulnerabilities are to provide unusual or extreme data to a running program. For example, an attacker can exploit a buffer overflow vulnerability by providing more input to a program than expected and overrun the area reserved by the program to hold a response. This can corrupt adjacent memory that might hold a function pointer.

When the program calls through this function, it can then jump to an unintended location specified by the attacker. These attacks are also known as jump-oriented programming (JOP) attacks. Control Flow Guard prevents JOP attacks by placing tight restrictions on what application code can be executed – especially indirect call instructions. It adds lightweight security checks to identify the set of functions in the application that are valid targets for indirect calls. When an application runs, it verifies that these indirect call targets are valid.

If the Control Flow Guard check fails at runtime, Windows Server 2016 immediately terminates the program, breaking any exploit that attempts to indirectly call an invalid address. Control Flow Guard provides an important additional layer of protection to Device Guard. If an allowlisted application has been compromised, it would be able to

run unchecked by Device Guard, because the Device Guard screening would see that the application has been signed and is considered trusted.

But because Control Flow Guard can identify whether the application is executing in a non-predetermined, non-viable order, the attack would fail, preventing the compromised application from running. Together, these protections make it very difficult for attackers to inject malware into software running on Windows Server 2016.

Developers building applications where personal data will be handled are encouraged to enable Control Flow Guard (CFG) in their applications. This feature is available in Microsoft Visual Studio 2015, and runs on "CFG-Aware" versions of Windows—the x86 and x64 releases for Desktop and Server of Windows 10 and Windows 8.1 Update (KB3000850). You don't have to enable CFG for every part of your code, as a mixture of CFG enabled and non-CFG enabled code will execute fine. But failing to enable CFG for all code can open gaps in the protection. Furthermore, CFG enabled code works fine on "CFG-Unaware" versions of Windows and is therefore fully compatible with them.

Windows Defender Antivirus

Windows Server 2016 includes the industry leading, active detection capabilities of Windows Defender to block known malware. Windows Defender Antivirus (AV) works together with Windows Defender Device Guard and Control Flow Guard to prevent malicious code of any kind from being installed on your servers. It is turned on by default – the administrator does not need to take any action for it to start working. Windows Defender AV is also optimized to support the various server roles in Windows Server 2016. In the past, attackers used shells such as PowerShell to launch malicious binary code. In Windows Server 2016, PowerShell is now integrated with Windows Defender AV to scan for malware before launching the code.

Windows Defender AV is a built-in antimalware solution that provides security and antimalware management for desktops, portable computers, and servers. Windows Defender AV has been significantly improved since it was introduced in Windows 8. Windows Defender Antivirus in Windows Server uses a multi-pronged approach to improve antimalware:

- **Cloud-delivered protection** helps detect and block new malware within seconds, even if the malware has never been seen before.
- **Rich local context** improves how malware is identified. Windows Server informs Windows Defender AV not only about content like files and processes but also where the content came from, where it has been stored, and more.

- **Extensive global sensors** help keep Windows Defender AV current and aware of even the newest malware. This is accomplished in two ways: by collecting the rich local context data from end points and by centrally analyzing that data.
- **Tamper proofing** helps guard Windows Defender AV itself against malware attacks. For example, Windows Defender AV uses Protected Processes, which prevents untrusted processes from attempting to tamper with Windows Defender AV components, its registry keys, and so on.
- **Enterprise-level features** give IT pros the tools and configuration options necessary to make Windows Defender AV an enterprise-class antimalware solution.

Enhanced security auditing

Windows Server 2016 actively alerts administrators to potential breach attempts with enhanced security auditing that provides more detailed information, which can be used for faster attack detection and forensic analysis. It logs events from Control Flow Guard, Windows Defender Device Guard, and other security features in one location, making it easier for administrators to determine what systems may be at risk.

New event categories include:

- **Audit Group Membership.** Allows you to audit the group membership information in a user's login token. Events are generated when group memberships are enumerated or queried on the PC where the login session was created.
- **Audit PnP Activity.** Allows you to audit when plug and play detects an external device – which could contain malware. PnP events can be used to track down changes in system hardware. A list of hardware vendor IDs is included in the event.

Windows Server 2016 integrates easily with security incident event management (SIEM) systems, such as Microsoft Operations Management Suite (OMS), which can incorporate the information into intelligence reports on potential breaches. The depth of information provided by the enhanced auditing enables security teams to identify and respond to potential breaches more quickly and effectively.

Secure virtualization

Enterprises today virtualize everything they can, from SQL Server to SharePoint to Active Directory Domain Controllers. Virtual machines (VMs) simply make it easier to deploy, manage, service, and automate your infrastructure. But when it comes to security, compromised virtualization fabrics have become a new attack vector that is hard to

defend against – until now. From a GDPR perspective, you should think about protecting VMs as you would protect physical servers including the use of VM TPM technology.

Windows Server 2016 fundamentally changes how enterprises can secure virtualization, by including multiple technologies that allow you to create virtual machines that will run only on your own fabric; helping to protect from the storage, network, and host devices they run on.

Shielded Virtual Machines

The same things that make virtual machines so easy to migrate, backup, and replicate, also make them easier to modify and copy. A virtual machine is just a file, so it is not protected on the network, in storage, in backups, or elsewhere. Another issue is that fabric administrators – whether they are a storage administrator or a network administrator – have access to all the virtual machines.

A compromised administrator on the fabric can easily result in compromised data across virtual machines. All the attacker must do is use the compromised credentials to copy whatever VM files they like onto a USB drive and walk it out of the organization, where those VM files can be accessed from any other system. If any one of those stolen VMs were an Active Directory domain controller, for example, the attacker could easily view the content and use readily available brute force techniques to crack the passwords in the Active Directory database, ultimately giving them access to everything else within your infrastructure.

Windows Server 2016 introduces Shielded Virtual Machines (Shielded VMs) to help protect against scenarios like the one just described. Shielded VMs include a virtual TPM device, which enables organizations to apply BitLocker Encryption to the virtual machines and ensure they run only on trusted hosts to help protect against compromised storage, network, and host administrators. Shielded VMs are created using Generation 2 VMs, which support Unified Extensible Firmware Interface (UEFI) firmware and have virtual TPM.

Host Guardian service

Alongside Shielded VMs, the Host Guardian Service is an essential component for creating a secure virtualization fabric. Its job is to attest to the health of a Hyper-V host before it will allow a Shielded VM to boot or to migrate to that host. It holds the keys for Shielded VMs and will not release them until the security health is assured. There are two ways that you can require Hyper-V hosts to attest to the Host Guardian Service.

The first, and most secure, is hardware-trusted attestation. This solution requires that your Shielded VMs are running on hosts that have TPM 2.0 chips and UEFI 2.3.1. This hardware is required to provide the measured boot and operating system kernel integrity information required by the Host Guardian Service to ensure the Hyper-V host has not been tampered with.

IT organizations have the alternative of using Admin-trusted attestation, which may be desirable if TPM 2.0 hardware is not in use in your organization. This attestation model is easy to deploy because hosts are simply placed into a security group and the Host Guardian Service is configured to allow Shielded VMs to run on hosts that are members of the security group. With this method, there is no complex measurement to ensure that the host machine hasn't been tampered with. However, you do eliminate the possibility of unencrypted VMs walking out the door on USB drives or that the VM will run on an unauthorized host. This is because the VM files won't run on any machine other than those in the designated group. If you do not yet have TPM 2.0 hardware, you can start with Admin-trusted attestation and switch to hardware-trusted attestation when your hardware is upgraded.

Virtual Machine Trusted Platform Module

Windows Server 2016 supports TPM for virtual machines, which allows you to support advanced security technologies such as BitLocker® Drive Encryption in virtual machines. You can enable TPM support on any Generation 2 Hyper-V virtual machine by using Hyper-V Manager or the Enable-VMTPM Windows PowerShell cmdlet.

You can protect virtual TPM (vTPM) by using the local crypto keys stored on the host or stored in the Host Guardian Service. So, while the Host Guardian Service requires more infrastructure, it also provides more protection.

Distributed network firewall using software-defined networking

One way to improve protection in virtualized environments is to segment the network in a way that allows VMs to talk only to the specific systems required to function. For example, if your application doesn't need to connect with the Internet, you can partition it off, eliminating those systems as targets from external attackers. The software-defined networking (SDN) in Windows Server 2016 includes a distributed network firewall that allows you to dynamically create the security policies that can protect your applications from attacks coming from inside or outside a network. This distributed network firewall adds layers to your security by enabling you to isolate your applications in the network. Policies can be applied anywhere across your virtual network infrastructure, isolating VM-to-VM traffic, VM-to-host traffic, or VM-to-Internet traffic where necessary – either

for individual systems that may have been compromised or programmatically across multiple subnets. Windows Server 2016 software-defined networking capabilities also enable you to route or mirror incoming traffic to non-Microsoft virtual appliances. For example, you could choose to send all your email traffic through a Barracuda virtual appliance for additional spam filtering protection. This allows you to easily layer in additional security both on-premises or in the cloud.

Other GDPR considerations for servers

The GDPR includes explicit requirements for breach notification where a personal data breach means, "*a breach of security leading to the accidental or unlawful destruction, loss, alteration, unauthorized disclosure of, or access to, personal data transmitted, stored or otherwise processed.*" Obviously, you can't begin to move forward to meet the stringent GDPR notification requirements within 72 hours if you cannot detect the breach in the first place.

As noted in the Windows Security Center white paper, Post Breach: Dealing with Advanced Threats

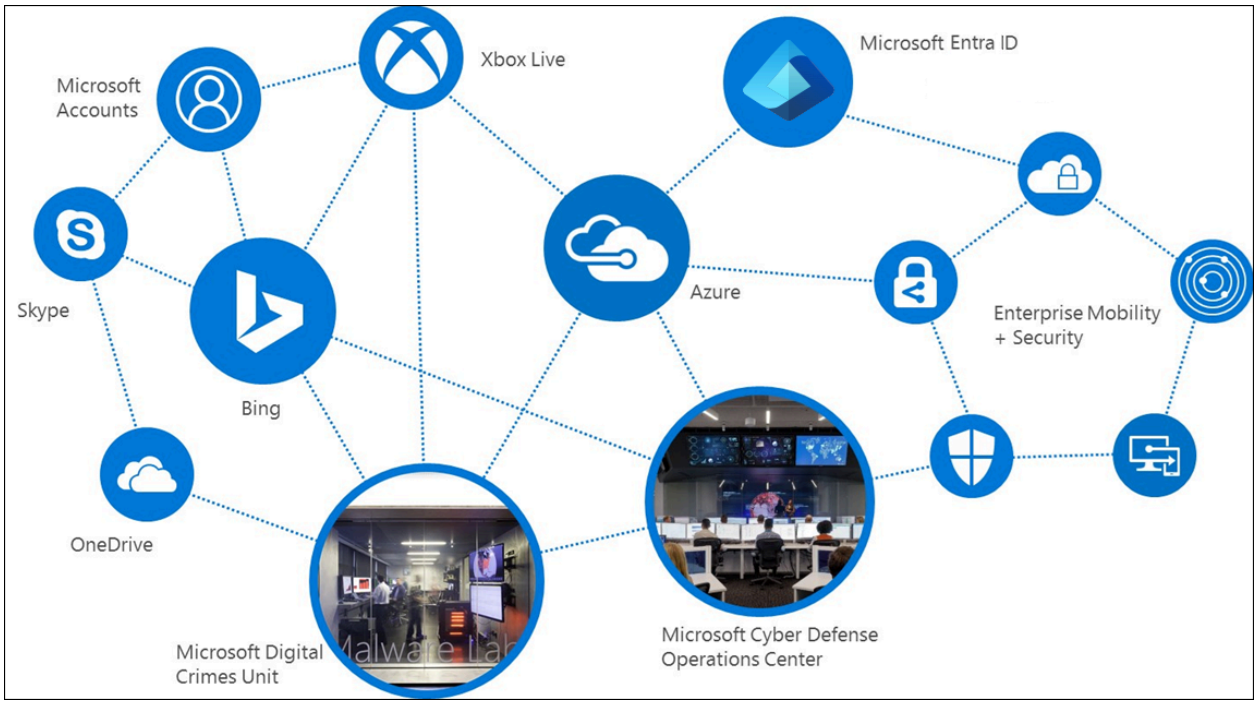
> "*Unlike pre-breach, post-breach assumes a breach has already occurred – acting as a flight recorder and Crime Scene Investigator (CSI). Post-breach provides security teams the information and toolset needed to identify, investigate, and respond to attacks that otherwise will stay undetected and below the radar.*"

In this section we will look at how Windows Server can help you meet your GDPR breach notification obligations. This starts with understanding the underlying threat data available to Microsoft that is gathered and analyzed for your benefit and how, through Windows Defender Advanced Threat Protection (ATP), that data can be critical to you.

Insightful security diagnostic data

For nearly two decades, Microsoft has been turning threats into useful intelligence that can help fortify its platform and protect customers. Today, with the immense computing advantages afforded by the cloud, we are finding new ways to use our rich analytics engines driven by threat intelligence to protect our customers.

By applying a combination of automated and manual processes, machine learning and human experts, we can create an Intelligent Security Graph that learns from itself and evolves in real-time, reducing our collective time to detect and respond to new incidents across our products.



The scope of Microsoft's threat intelligence spans, literally, billions of data points: 35 billion messages scanned monthly, 1 billion customers across enterprise and consumer segments accessing 200+ cloud services, and 14 billion authentications performed daily. All this data is pulled together on your behalf by Microsoft to create the Intelligent Security Graph that can help you protect your front door in a dynamic way to stay secure, remain productive and meet the requirements of the GDPR.

Detecting attacks and forensic investigation

Even the best endpoint defenses may be breached eventually, as cyberattacks become more sophisticated and targeted. Two capabilities can be used to help with potential breach detection - Windows Defender Advanced Threat Protection (ATP) and Microsoft Advanced Threat Analytics (ATA).

Windows Defender Advanced Threat Protection (ATP) helps you detect, investigate, and respond to advanced attacks and data breaches on your networks. The types of data breach the GDPR expects you to protect against through technical security measures to ensure the ongoing confidentiality, integrity, and availability of personal data and processing systems.

Among the key benefits of Windows Defender ATP are the following:

- **Detecting the undetectable.** Sensors built deep into the operating system kernel, Windows security experts, and unique optics from over 1 billion machines and signals across all Microsoft services.

- **Built in, not bolted on.** Agentless, with high performance and minimal impact, cloud-powered; easy management with no deployment.

- **Single pane of glass for Windows security.** Explore 6 months of rich, machine-timeline, unifying security events from Windows Defender ATP, Windows Defender Antivirus and Windows Defender Device Guard.
- **Power of the Microsoft graph.** Leverages the Microsoft Intelligence Security Graph to integrate detection and exploration with Office 365 ATP subscription, to track back and respond to attacks.

Read more at What's new in the Windows Defender ATP Creators Update preview.

ATA is an on-premises product that helps detect identity compromise in an organization. ATA can capture and parse network traffic for authentication, authorization, and information gathering protocols (such as Kerberos, DNS, RPC, NTLM, and other protocols). ATA uses this data to build a behavioral profile about users and other entities on a network so that it can detect anomalies and known attack patterns. The following table lists the attack types detected by ATA.

Expand table

Attack type	Description
Malicious attacks	These attacks are detected by looking for attacks from a known list of attack types, including: • Pass-the-Ticket (PtT) • Pass-the-Hash (PtH) • Overpass-the-Hash • Forged PAC (MS14-068) • Golden Ticket • Malicious replications • Reconnaissance • Brute force • Remote execution For a complete list of malicious attacks that can be detected and their description, see What threats does ATA look for?.
Abnormal behavior	These attacks are detected by using behavioral analysis and use machine learning to identify questionable activities, including: • Anomalous logins • Unknown threats • Password sharing • Lateral movement
Security issues and risks	These attacks are detected by looking at current network and system configuration, including: • Broken trust • Weak protocols

Attack type	Description
	• Known protocol vulnerabilities

You can use ATA to help detect attackers attempting to compromise privileged identities. For more information on deploying ATA, see the Plan, Design, and Deploy topics in the Advanced Threat Analytics documentation.

Related content for associated Windows Server 2016 solutions

- **Windows Defender Antivirus:**
  - Windows Defender Antivirus (YouTube video)
  - Microsoft Defender Antivirus in Windows
- **Windows Defender Advanced Threat Protection:**
- An overview of Windows Defender Advanced Threat Protection for Windows 10 Creators Update (YouTube video)
- Onboard Windows servers to the Microsoft Defender for Endpoint service
- **Windows Defender Device Guard:**
- Windows Defender Device Guard (YouTube video)
- Deploying Windows Defender Application Control (WDAC) policies
- **Windows Defender Credential Guard:** Windows Defender Credential Guard (YouTube video)
- Protect derived domain credentials with Windows Defender Credential Guard
- **Control Flow Guard:**
  - Control Flow Guard for platform security
- **Security and Assurance:**
- Windows Server Security documentation

Disclaimer

This article is a commentary on the GDPR, as Microsoft interprets it, as of the date of publication. We've spent a lot of time with GDPR and like to think we've been thoughtful

about its intent and meaning. But the application of GDPR is highly fact-specific, and not all aspects and interpretations of GDPR are well-settled.

As a result, this article is provided for informational purposes only and should not be relied upon as legal advice or to determine how GDPR might apply to you and your organization. We encourage you to work with a legally-qualified professional to discuss GDPR, how it applies specifically to your organization, and how best to ensure compliance.

MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS ARTICLE. This article is provided “as-is.” Information and views expressed in this article, including URL and other Internet website references, may change without notice.

This article does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this article for your internal, reference purposes only.

Published September 2017
Version 1.0
© 2017 Microsoft. All rights reserved.

Feedback

Was this page helpful? Yes No

What is Secured-core server?

Article • 12/05/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Azure Local, versions 23H2 and 22H2

Secured-core is a collection of capabilities that offers built-in hardware, firmware, driver and operating system security features. The protection provided by Secured-core systems begins before the operating system boots and continues whilst running. Secured-core server is designed to deliver a secure platform for critical data and applications.

Secured-core server is built on three key security pillars:

- Creating a hardware backed root of trust.
- Defense against firmware level attacks.
- Protecting the OS from the execution of unverified code.

What makes a Secured-core server

The Secured-core initiative started with Windows PCs through a deep collaboration between Microsoft and PC manufacturing partners to provide the most elevated Windows security ever. Microsoft has expanded the partnership further with server manufacturing partners to help ensure Windows Server delivers a secure operating system environment.

Windows Server integrates closely with hardware to provide increasing levels of security:

- Recommended baseline: The recommended minimum for all systems to provide foundational system integrity using TPM 2.0 for a hardware root of trust and Secure Boot. TPM2.0 and Secure boot are required for Windows Server hardware certification. To learn more, see Microsoft raises the security standard for next major Windows Server release
- Secured-core server: Recommended for systems and industries requiring higher levels of assurance. Secured-core server builds on the previous features and uses advanced processor capabilities to provide protection from firmware attacks.

The following table shows how each security concept and feature are used to create a Secured-core server.

Expand table

Concept	Feature	Requirement	Recommended baseline	Secured-Core server
Create a hardware backed root of trust				
	Secure Boot	Secure Boot is enabled in the Unified Extensible Firmware Interface (UEFI) BIOS by default.	✓	✓
	Trusted Platform Module (TPM) 2.0	Meet the latest Microsoft requirements for the Trusted Computing Group (TCG) specification.	✓	✓
	Certified for Windows Server	Demonstrates that a server system meets Microsoft's highest technical bar for security, reliability and manageability.	✓	✓
	Boot DMA protection	Support on devices that have the Input/Output Memory Management Unit (IOMMU). For example, Intel VT-D or AMD-Vi.		✓
Defend against firmware level attacks				
	System Guard Secure Launch	Enabled in the operating system with Dynamic Root of Trust for Measurement (DRTM) compatible Intel and AMD hardware.		✓
Protect the OS from execution of unverified code				

Concept	Feature	Requirement	Recommended baseline	Secured-Core server
	Virtualization-based Security (VBS)	Requires the Windows hypervisor, which is only supported on 64-bit processors with virtualization extensions, including Intel VT-X and AMD-v.	✓	✓
	Hypervisor Enhanced Code Integrity (HVCI)	Hypervisor Code Integrity (HVCI)-compatible drivers plus VBS requirements.	✓	✓

Create a hardware backed root of trust

UEFI Secure boot is a security standard that protects your servers from malicious rootkits by verifying your systems boot components. Secure boot verifies a trusted author has digitally signed the UEFI firmware drivers and applications. When the server is started, the firmware checks the signature of each boot component including firmware drivers and the OS. If the signatures are valid, the server boots and the firmware gives control to the OS.

To learn more about the boot process, see Secure the Windows boot process.

TPM 2.0 provides a secure, hardware-backed storage for sensitive keys and data. Every component loaded during the boot process is measured and the measurements stored in the TPM. By verifying the hardware root-of-trust it elevates the protection provided by capabilities like BitLocker, which uses TPM 2.0 and facilitates the creation of attestation-based workflows. These attestation-based workflows can be incorporated into zero-trust security strategies.

Learn more about Trusted Platform Modules and how Windows uses the TPM.

Along with Secure Boot and TPM 2.0, Windows Server Secured-core uses Boot DMA protection on compatible processors that have the Input/Output Memory Management Unit (IOMMU). For example, Intel VT-D or AMD-Vi. With boot DMA protection, systems are protected from Direct Memory Access (DMA) attacks during boot and during the operating system runtime.

Defend against firmware level attacks

Endpoint protection and detection solutions usually have limited visibility of firmware, given that firmware runs underneath of the operating system. Firmware has a higher level of access and privilege than operating system and hypervisor kernel, making it an attractive target for attackers. Attacks targeting firmware undermine other security measures implemented by the operating system, making it more difficult to identify when a system or user has been compromised.

Beginning with Windows Server 2022, System Guard Secure Launch protects the boot process from firmware attacks by using hardware capabilities from AMD and Intel. With processor support for Dynamic Root of Trust for Measurement (DRTM) technology, Secured-core servers put firmware in a hardware-backed sandbox helping to limit the effects of vulnerabilities in highly privileged firmware code. System Guard uses the DRTM capabilities that are built into compatible processors to launch the operating system, ensuring the system launches into a trusted stated using verified code.

Protect the OS from execution of unverified code

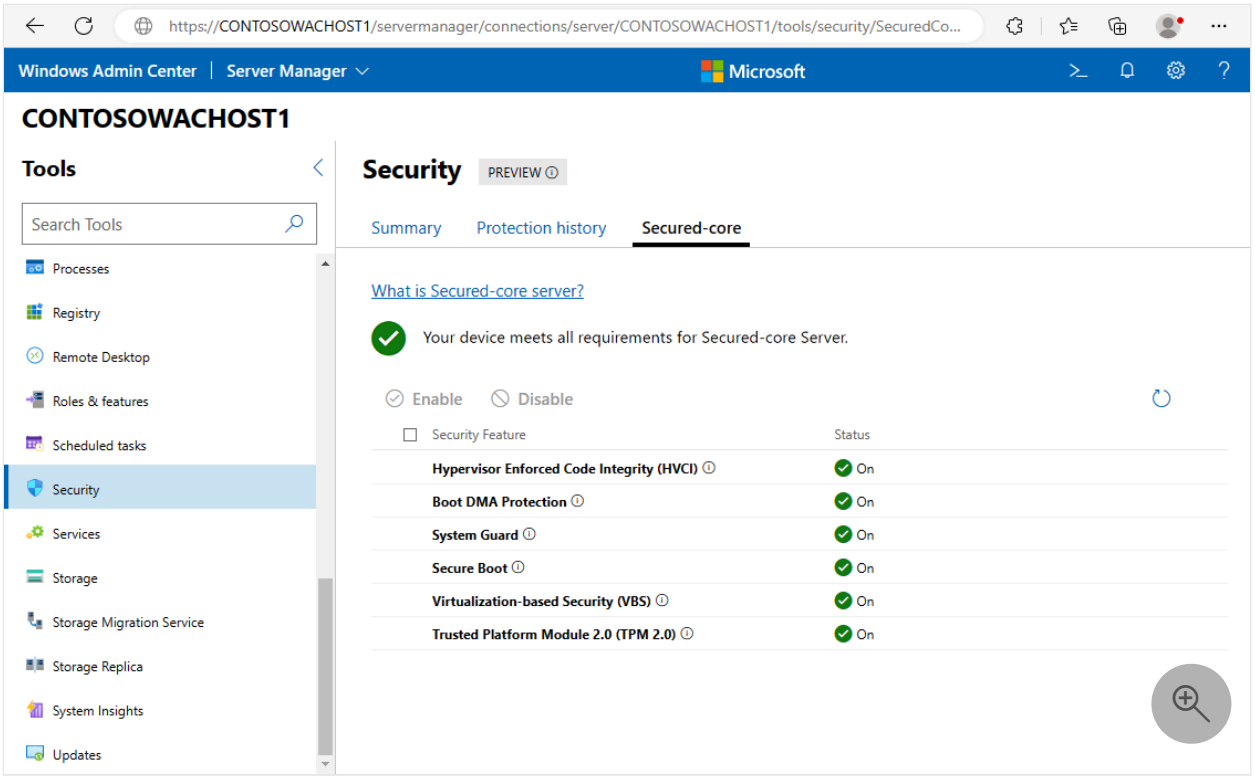
Secured-core server uses Virtualization Based Security (VBS) and hypervisor-protected code integrity (HVCI) to create and isolate a secure region of memory from the normal operating system. VBS uses the Windows hypervisor to create a Virtual Secure Mode (VSM) to offer security boundaries within the operating system, which can be used for other security solutions.

HVCI, commonly referred to as Memory integrity protection, is a security solution that helps ensure that only signed and trusted code is allowed to execute in the kernel. Using only signed and trusted code prevents attacks that attempt to modify the kernel mode code. For example, attacks that modify drivers, or exploits such as WannaCry that attempt to inject malicious code into the kernel.

To learn more about VBS and hardware requirements, see Virtualization-based Security.

Simplified management

You can view and configure the OS security features of Secured-core systems using Windows PowerShell or the security extension in Windows Admin Center. With Azure Local Integrated Systems, manufacturing partners have further simplified the configuration experience for customers so that Microsoft's best server security is available right out of the box.



Learn more about Windows Admin Center.

Preventative defense

You can proactively defend against and disrupt many of the paths attackers use to exploit systems by enabling Secured-core functionality. Secured-core server enables advanced security features at the bottom layers of the technology stack, protecting the most privileged areas of the system before many security tools are aware of exploits. It also occurs without the need for extra tasks or monitoring by IT and SecOps teams.

Begin your Secured-core journey

You can find hardware certified for Secured-core server from the Windows Server Catalog, and Azure Local servers in the Azure Local Catalog. These certified servers come fully equipped with industry-leading security mitigations built into the hardware, firmware, and the operating system to help thwart some of the most advanced attack vectors.

Next steps

Now you understand what Secured-core server is, here are some resources to get you started. Learn about how:

- Configure Secured-core server.

- Microsoft brings advanced hardware security to Server and Edge with Secured-core in the Microsoft Security Blog.
- New Secured-core servers are now available from the Microsoft ecosystem to help secure your infrastructure in the Microsoft Security Blog.
- Building Windows-compatible devices, systems, and filter drivers across all Windows Platforms in Windows Hardware Compatibility Program Specifications and Policies.

Feedback

Was this page helpful? Yes No

Configure Secured-core server

Article • 12/05/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Secured-core is a collection of capabilities that offers built-in hardware, firmware, driver and operating system security features. This article shows you how to configure Secured-core server by using Windows Admin Center, the Windows Server Desktop Experience, and Group Policy.

Secured-core server is designed to deliver a secure platform for critical data and applications. For more information, see What is Secured-core server?

Prerequisites

Before you can configure Secured-core server, you must have the following security components installed and enabled in the BIOS:

- Secure Boot.
- Trusted Platform Module (TPM) 2.0.
- System firmware must meet preboot DMA protection requirements and set appropriate flags in ACPI tables to opt into and enable Kernel DMA Protection. To learn more about Kernel DMA Protection, see Kernel DMA Protection (Memory Access Protection) for OEMs.
- A processor with support enabled in the BIOS for:
  - Virtualization extensions.
  - Input/Output Memory Management Unit (IOMMU).
  - Dynamic Root of Trust for Measurement (DRTM).
  - Transparent Secure Memory Encryption is also required for AMD based systems.

> **ⓘ Important**
>
> Enabling each of the security features in the BIOS can vary based on your hardware vendor. Make sure to check your hardware manufacturer's Secured-core server enablement guide.

You can find hardware certified for Secured-core server from the Windows Server Catalog, and Azure Local servers in the Azure Local Catalog.

Enable security features

To configure Secured-core server you need to enable specific Windows Server security features, select the relevant method and follow the steps.

GUI

Here's how to enable Secured-core server using the user interface.

1. From the Windows desktop, open the **Start** menu, select **Windows Administrative Tools**, open **Computer Management**.
2. In Computer management, select **Device Manager**, resolve any device error if necessary.
   a. For AMD based systems, confirm the DRTM Boot Driver device is present before continuing
3. From Windows desktop, open the **Start** menu, select **Windows Security**.
4. Select **Device security** > **Core isolation details**, then enable **Memory Integrity** and **Firmware Protection**. You might not be able to enable Memory Integrity until you've enabled Firmware Protection first and restarted your server.
5. Restart your server when prompted.

Once your server has restarted, your server is enabled for Secured-core server.

Verify Secured-core server configuration

Now that you've configured Secured-core server, select the relevant method to verify your configuration.

GUI

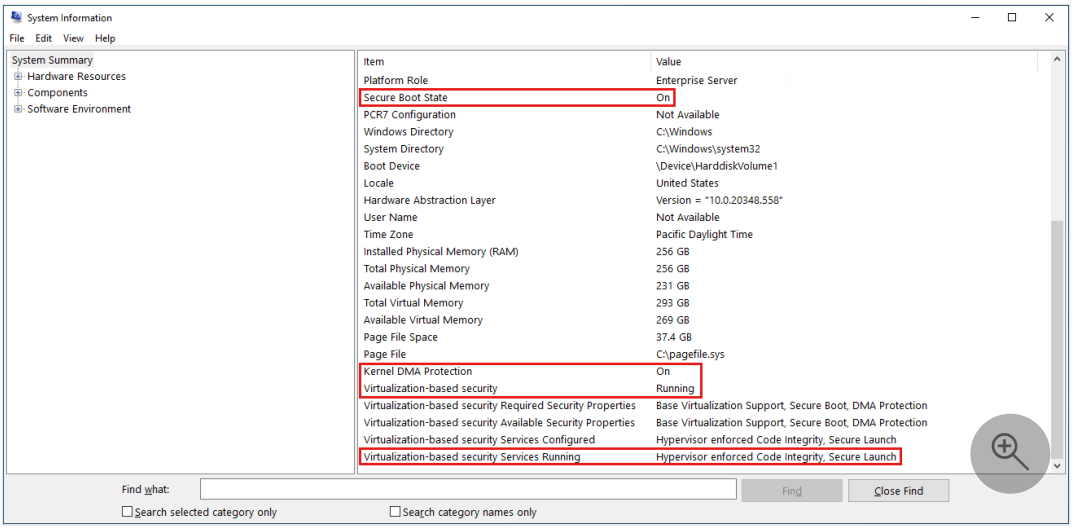
Here's how to verify your Secured-core server is configured using the user interface.

1. From the Windows desktop, open the **Start** menu, type `msinfo32.exe` to open System Information. From the System Summary page, confirm:

   a. **Secure Boot State** and **Kernel DMA Protection** is On.

   b. **Virtualization-based security** is Running.

   c. **Virtualization-based security Services** Running shows **Hypervisor enforced Code Integrity** and **Secure Launch**.



Next steps

Now that you've configured Secured-core server, here are some resources to learn more about:

- Virtualization-based Security (VBS)
- Memory integrity and VBS enablement
- System Guard Secure Launch

Feedback

Was this page helpful? Yes No

Kernel Mode Hardware-enforced Stack Protection

Article • 10/30/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> Applies To: Windows Server 2025

The security feature **Hardware-enforced Stack Protection** was introduced to protect user mode processes and help prevent hijacking on the stack in Windows 10. Hardware-enforced Stack Protection now extends to **Kernel Mode**, which protects stacks in the kernel from return-oriented programming-based attacks (ROP). ROP is a common way for attackers to hijack the flow of a program's execution, and continue their attack chain to execute attacker desired code.

Now that user mode stacks are protected, and prevent return address modification from kernel mode code, attackers can't exploit memory safety vulnerabilities. Customers are already seeing how Kernel Mode Hardware-enforced Stack Protection prevents drivers associated with viruses and malware from executing their malicious payload.

Kernel-mode Hardware-enforced Stack Protection is off by default, but customers can turn it on if the prerequisites are met. This article provides more information about Kernel-mode Hardware-enforced Stack Protection, and shows how to enable the feature in the Windows Security App and via Group Policy.

Prerequisites

- Windows 11 2022 update or newer
- Windows Security app version 1000.25330.0.9000 or newer
- Hardware that supports Intel Control-flow Enforcement Technology (CET) or AMD Shadow Stacks.
  - For Intel, 11th Gen Intel Core Mobile processors and AMD Zen 3 Core (and newer).
- Virtualization-based security (VBS) and Hypervisor-enforced code integrity (HVCI) are enabled.

Use shadow stacks to enforce integrity of control flow

With Kernel-mode Hardware-enforced Stack Protection, all kernel stacks have a corresponding shadow stack to enforce the integrity of its control flow. If attackers exploit a memory safety vulnerability, their next step is to redirect control flow of a program to an attacker's desired location.

Shadow stacks prevent control-flow hijacking. Windows uses Control Flow Guard to enforce integrity on indirect calls, and Hardware-enforced Stack Protection to enforce integrity on returns in order to protect against exploits which aim to redirect the flow of a program's execution. Control Flow Guard utilizes a bitmap to annotate valid jump targets, to prevent a compromised indirect call from redirecting control flow to arbitrary locations.

Shadow stack maintains a (hardware protected) secondary stack for all call stacks, and whenever a CALL or RET instruction pushes or pops a value onto the stack, a corresponding entry lives in the shadow stack. When a return address mismatch occurs, the system triggers a blue screen to prevent unintended program control behavior.

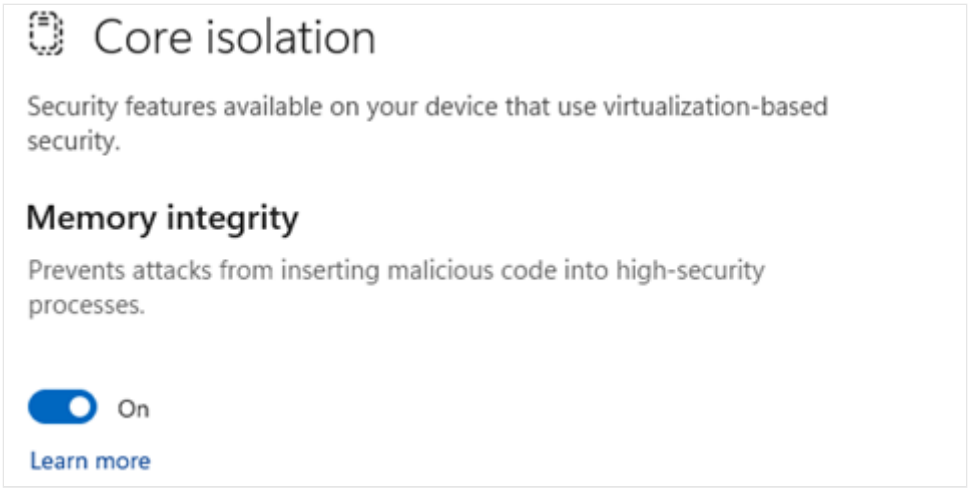
For more information, see the blog post on Understanding Hardware-enforced Stack Protection.

Enable Kernel-mode Hardware-enforced Stack Protection in Windows Security

Virtualization-Based Security (VBS) and Hypervisor-enforced Code Integrity (HVCI) are prerequisites for Kernel-mode Hardware-enforced Stack Protection, you must first ensure that these features are enabled before continuing. They are autoenabled on Windows systems that meet minimum hardware requirements.

Enable VBS and HVCI with the following steps:

1. Open the **Windows Security** app.
2. Navigate to **Device Security** > **Core isolation details** > **Memory integrity**.
3. Toggle the feature **On**.
4. After making this change, you need to restart your device.



Turn on Kernel-mode Hardware-enforced Stack Protection

1. Open the **Windows Security** app.
2. Navigate to **Device Security** > **Core isolation details** > **Kernel-mode Hardware-enforced Stack Protection**.
3. Toggle the feature **On**.

Enable Kernel-mode Hardware-enforced Stack Protection in the Local Group Policy Editor

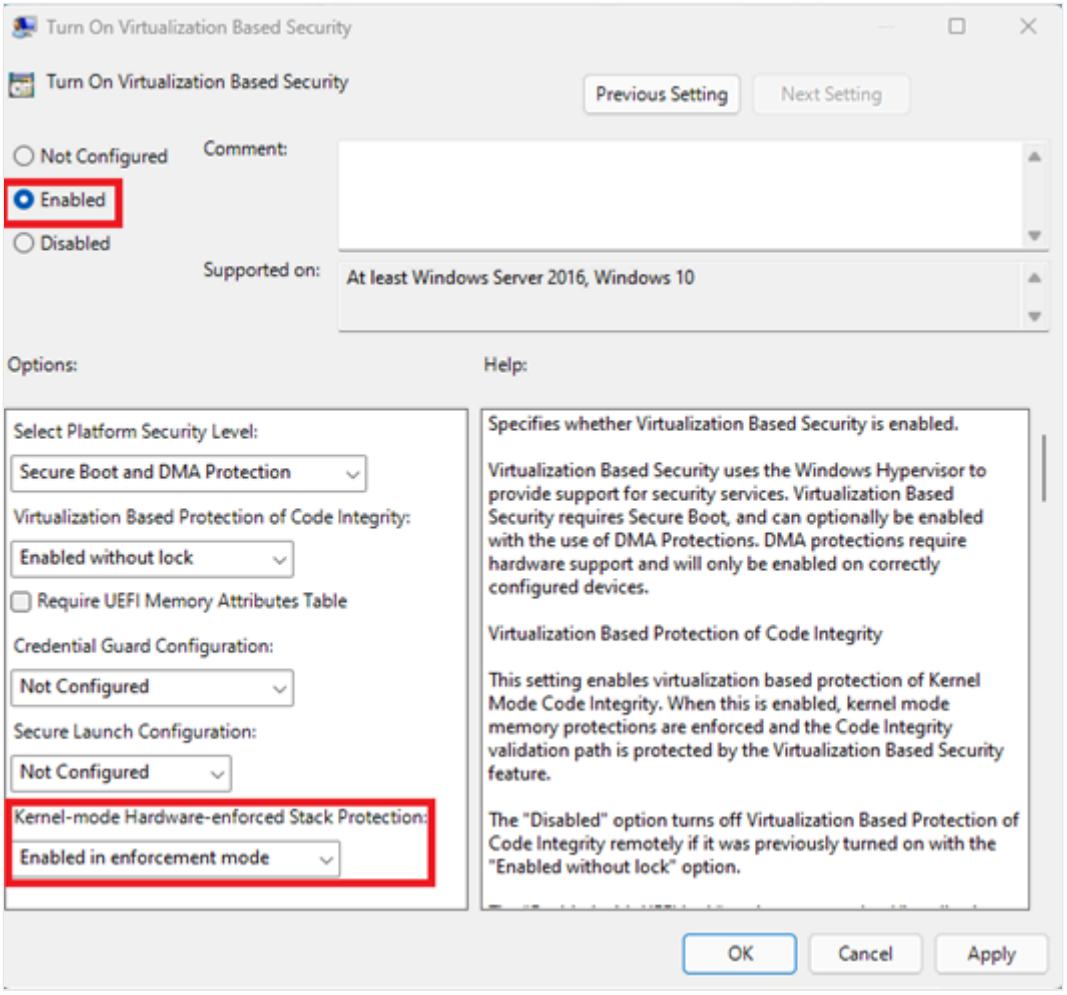
For enterprise customers, Kernel-mode Hardware-enforced Stack Protection can be enabled using Group Policy.

1. Open the Local Group Policy Editor.
2. Navigate to **Computer Configuration** > **Administrative Templates** > **System** > **Device Guard** > **Turn on Virtualization Based Security**.

3. Confirm that Virtualization Based Security is **Enabled**.

4. Under **Options** find **Kernel-mode Hardware-enforced Stack Protection**. Select **Enabled in enforcement mode**.

5. Select **Apply**. Then **OK**.



Incompatible drivers

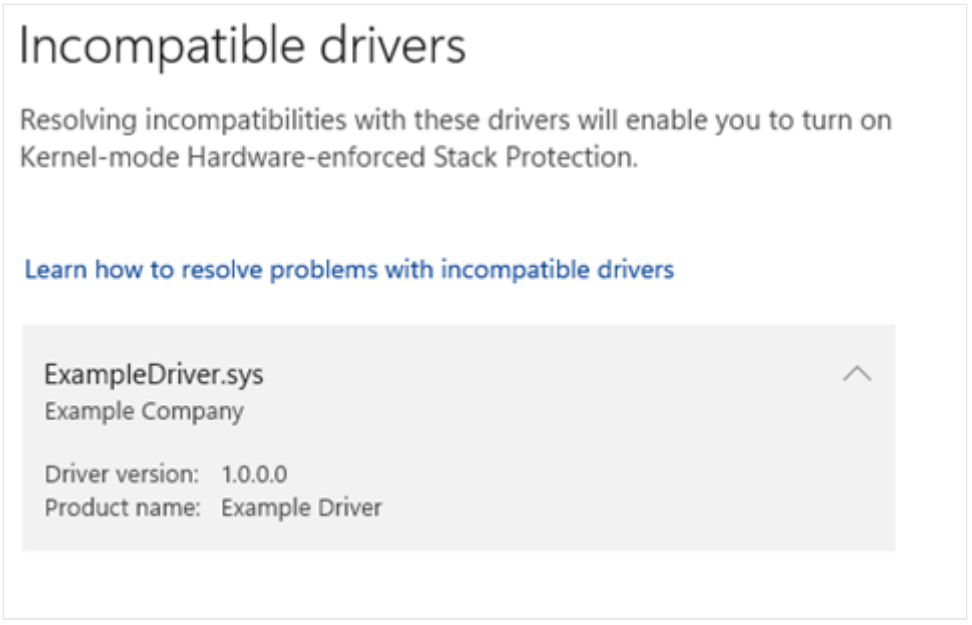
There's a small set of drivers that aren't yet compatible. Drivers that exhibit potentially malicious behavior like hijacking return addresses to get around control flow policies aren't compatible, and are added to the vulnerable driver blocklist for kernel-mode hardware-enforced stack protection. After working with driver vendors to perform code obfuscation in a shadow stack compliant way, those drivers are allowed.

To provide good user experience and avoid machine blue screens, Windows maintains a known-incompatible driver blocklist for kernel-mode hardware-enforced stack protection. These are drivers that are known to hijack return addresses in the kernel. When this feature is turned on, the driver isn’t allowed to load (as opposed to a blue screen when a return address hijack is attempted). Additionally, if the system already has

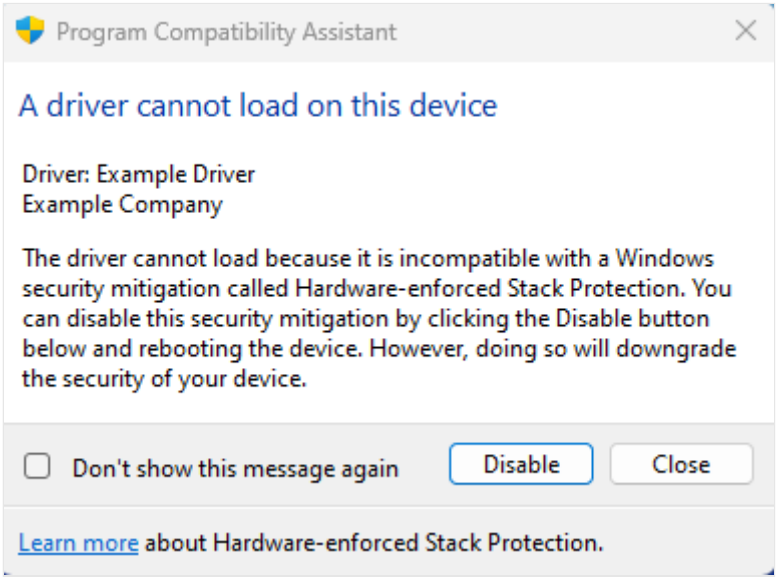
a driver installed on the blocklist, this feature fails to enable. You can enable this feature by uninstalling the associated driver.

Review incompatible drivers

The feature can't be enabled until the incompatibilities are resolved, either with an updated version from the driver vendor or by removing the application that installed the driver. To see the list of incompatible drivers, select "Review incompatible drivers".



Certain apps use drivers that are incompatible with Kernel-mode Hardware-enforced Stack Protection. For example, apps that use obfuscation engines to protect IP and obfuscate control flow, which are incompatible with shadow stacks. When the insecure driver attempts to load with this security feature enabled, you see a prompt saying "A driver cannot load on this device".



You can optionally disable the security feature, although doing so downgrades the security of your device. You can always re-enable this feature in the Windows Security application.

Related content

- Understanding Hardware-enforced Stack Protection
- Developer Guidance for Hardware-enforced Stack Protection
- Enable virtualization-based protection of code integrity
- Memory integrity and VBS enablement

Feedback

Was this page helpful? Yes No

AIA URL retrieval in Windows

07/22/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Authority Information Access (AIA) extension in X.509 certificates specifies a URL where the issuing Certificate Authority (CA) certificate can be obtained. If a system is unable to build a complete certificate trust chain using only the certificates in its local store, it might attempt to download missing intermediate certificates from the network using the AIA URL. AIA automates the retrieval of missing certificates and minimizes the manual effort required to update local certificate stores.

The automatic retrieval helps complete the trust chain but can introduce security risks if not properly managed. To enhance security, ensure that your certificate stores are properly configured and contain only certificates from trusted CAs. You can further strengthen your environment by disabling AIA retrieval system-wide. This forces Windows APIs to build certificate chains exclusively from certificates already present in the local store, reducing the risk of inadvertently trusting unverified or malicious certificates retrieved from external sources. Adopting this defense-in-depth strategy helps safeguard your systems against unauthorized or unexpected certificate trust paths.

Best practices for managing your CAs include:

- Keep the list of trusted root and intermediate CAs up-to-date. This involves regularly updating the local certificate store with new or updated certificates from trusted sources.
- Ensure intermediate certificates are consistently added to the local store, especially if AIA retrieval is disabled. This prevents potential disruptions in certificate validation.
- Conduct regular audits of the certificate store to identify and remove expired or untrusted certificates. This practice helps maintain trustworthiness and reduces security risks associated with outdated certificates.

Prerequisites

To make changes to the AIA, you must be a member of the **Administrators**, **Domain Admins**, or **Enterprise Admins** security group.

Manage AIA retrieval

To disable AIA retrieval system-wide in your environment, follow one of these methods:

Group Policy

1. Select **Start**, type **gpedit**, and select **Edit group policy**.
2. Navigate to **Computer Configuration\Windows Settings\Security Settings\Public Key Policies** and open **Certificate Path Validation Settings**.
3. Select the **Network Retrieval** tab. Select **Define these policy settings**, uncheck **Allow issuer certificate (AIA) retrieval during path validation**, then select **OK**.

If you need to reenable AIA retrieval, uncheck **Define these policy settings** and then select **OK**.

Retrieve AIA status

By default, AIA is enabled on your device. AIA retrieval is triggered by the **CCertChainEngine::GetIssuerUrlStore()** function in **crypt32.dll**. The **CERT_CHAIN_DISABLE_AIA** and **CERT_CHAIN_ENSABLE_AIA** is set in the **dwFlags** passed to create the engine or in the call to **CertGetCertificateChain**. To retrieve your current AIA status, run the following command:

PowerShell

```
$registryPath =
"HKLM:\Software\Policies\Microsoft\SystemCertificates\ChainEngine\Config"
$propertyName = "Options"

try {
    $regCheck = (Get-ItemProperty -Path $registryPath -Name $propertyName -
ErrorAction Stop).$propertyName

    switch ($value) {
        0 { "AIA is Enabled" }
        2 { "AIA is Disabled" }
        default { "Unexpected value: $regCheck" }
    }
} catch {
    "AIA has not been set"
}
```

See also

- Active Directory Certificate Services documentation
- Microsoft's Zero Trust Best Practices

OSConfig security configuration for Windows Server

Applies to: ✅ Windows Server 2025

OSConfig is a security configuration tool for Windows Server that applies and maintains a known good security state on your devices. It uses *scenarios*, which are predefined groups of settings, to deliver a consistent configuration to on-premises and Azure Arc-connected devices across their life cycle.

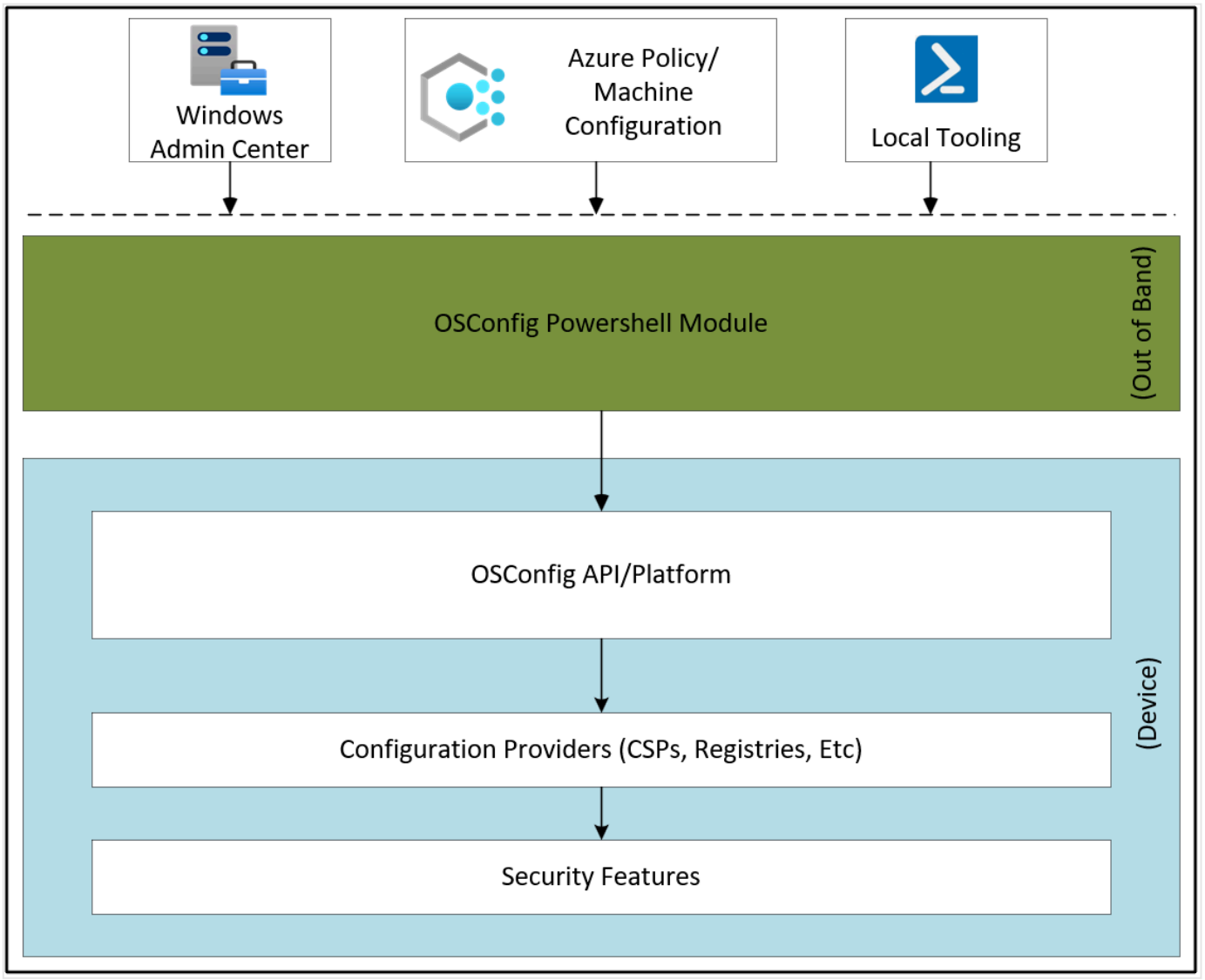
Over time, servers drift away from their intended security configuration, and applying settings consistently across many devices and management tools is hard to do by hand. OSConfig helps you keep each device in its intended state by detecting and correcting this drift automatically, so your servers stay secure and compliant.

This article explains what OSConfig is, how it works, and the scenarios it supports, so you can decide how to use it to secure and maintain your servers.

How OSConfig works

The OSConfig stack consists of base cmdlets, native APIs, and a scenario definition that defines the desired state configuration. The scenario definition is a data-driven description of configurations. The configurations are groups of settings that use name/value pairs with a predefined order and dependencies that correspond to subareas.

The object model design is data driven, which allows mapping to various providers in the Windows operating system (OS) for device configuration. The following diagram describes the OSConfig flow.



On Windows Server 2025, OSConfig integrates with Azure Policy, Microsoft Defender, Windows Admin Center, and Azure Policy machine configuration to support monitoring and compliance reporting. It also enables improved mapping or even direct conversion with other preexisting management definitions. These definitions include `.admx` files in Group Policy, `.mof` files in Windows Management Instrumentation (WMI), and Device Description Framework (DDF) files in the configuration service provider (CSP).

As a single platform, OSConfig:

- Applies configuration payloads across the device life cycle, including configuration, remediation, monitoring, reporting, and versioning.
- Works consistently across Windows Admin Center, Azure Arc, PowerShell, Azure Policy, and Azure Policy machine configuration.
- Honors prerequisites and dependencies among settings through orchestration directives.
- Maintains the desired state through configuration drift detection, reporting, and correction.

OSConfig drift control

One of the main features of OSConfig is *drift control*. It helps ensure that the system starts and remains in a known good security state. When you turn it on, OSConfig automatically corrects any system changes that deviate from the desired state. OSConfig makes the correction through a refresh task.

When you turn off the feature, OSConfig also disables the refresh task. You can then use other tools, with or without OSConfig, to modify the system. Each management tool can serve various purposes, and different actors can use it, so multiple authorities can manage the same set of device settings. For instance, authorities can use Azure Policy for cloud or Azure Arc-enabled resources at scale, whereas they can use Windows Admin Center for local management.

To address multiple authorities, an orchestrator ensures deterministic configuration in an environment where multiple authorities use various IT admin tools. Under this model, each authority receives a precedence order. This precedence order doesn't just apply from a configuration perspective. It also allows drift control per authority and even per scenario document.

If you use cloud or Azure Arc-enabled resources, the precedence order is:

1. Cloud authority (Azure Policy)
2. Local authority (Windows Admin Center and Windows PowerShell)
3. Any other deployment tool

OSConfig supported scenarios

OSConfig delivers configuration through scenarios. On Windows Server 2025, the supported scenarios fall into two areas: security baselines and App Control for Business.

Security baselines

OSConfig-driven security baselines apply a recommended, role-aware security posture to your servers and virtual machines. Throughout the device life cycle, you apply them by using PowerShell or Windows Admin Center, and drift control helps keep each server in its known good state. This area includes:

- The Windows Server security baseline, tailored to the server role.
- Microsoft Defender Antivirus configuration.
- Windows Local Administrator Password Solution (LAPS).
- Secured-core server configuration.

Applying the security baselines helps you:

- Strengthen the security posture by disabling legacy protocols and ciphers.
- Reduce operating effort through built-in drift control for consistent, at-scale monitoring.
- Work toward the Center for Internet Security (CIS) Benchmarks and Defense Information Systems Agency Security Technical Implementation Guides (DISA STIGs) recommendations for the OS security baseline.

App Control for Business

App Control for Business helps you control which applications and code can run on your servers, helping to reduce the risk from unauthorized or untrusted software. OSConfig provides a scenario to apply and maintain this configuration as part of the same desired-state model.

Related content

- Deploy security baselines with OSConfig
- Deploy App Control for Business with OSConfig

ⓘ**Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 07/02/2026

Deploy Windows Server 2025 security baselines locally with OSConfig

Applies to: ✅ Windows Server 2025

A security baseline is a recommended, role-aware collection of security settings that hardens the servers and virtual machines in your environment. Deploying the Windows Server 2025 security baseline gives you a comprehensive, standardized starting point that aligns with industry-standard security requirements, instead of configuring hundreds of settings by hand.

OSConfig powers the baseline experience. To learn how OSConfig delivers and maintains security configuration across the device lifecycle, see OSConfig overview.

In this article, you install the OSConfig PowerShell module, apply the security baseline that matches your server's role, and verify, customize, and remove the baseline. OSConfig doesn't support versions of Windows Server earlier than Windows Server 2025.

Prerequisites

- A device running **Windows Server 2025**. OSConfig doesn't support earlier versions of Windows Server.
- Administrator rights on the device. Installing the OSConfig PowerShell module and applying, verifying, or removing baselines all require an elevated PowerShell session.

Why security baselines matter

The security baseline goes beyond checklist compliance. It aims to strengthen protection across five areas, and no control is absolute. The goal is to raise the cost and reduce the likelihood of an attack succeeding. The baseline organizes the settings into five protection categories. Each category maps to common attacker techniques that typically succeed on a default (unhardened) server but are prevented, impaired, or visibly logged on a baseline-hardened server.

Network exposure reduction

The baseline reduces the server's exposed attack surface by disabling legacy protocols, enforcing modern cryptography, and tightening inbound access:

- The baseline enables the Windows Firewall on all profiles (Domain, Private, Public) with a default-deny stance for inbound traffic. Only ports with explicit allow rules remain open.
- The baseline disables SMBv1 on both client and server and enforces a minimum of SMB 3.0.
- The baseline shuts down legacy name-resolution protocols: it disables LLMNR and NetBIOS over TCP/IP, and the system relies exclusively on DNS.
- The baseline reduces anonymous and unauthenticated access: it blocks anonymous SAM enumeration, disables insecure guest logons, and disables the Guest account.
- The baseline restricts TLS to version 1.2 or higher with modern cipher suites and disables IP source routing.

For example, these settings help defend against an attacker running an SMB enumeration tool such as `enum4linux`, attempting the EternalBlue (MS17-010) exploit against a disabled protocol, or running Responder to poison LLMNR or NetBIOS broadcasts that the hardened server no longer sends.

Credential theft resistance

This baseline helps protect in-memory and on-disk credentials so that an attacker who gains a foothold has a harder time harvesting passwords or hashes:

- Credential Guard uses virtualization-based security (VBS) to help isolate NTLM hashes, Kerberos ticket-granting tickets, and stored domain credentials from the operating system. Credential Guard requires compatible hardware and VBS to be available and enabled.
- LSASS runs as a Protected Process Light (PPL) to help block code injection and memory reads.
- The baseline restricts authentication to NTLMv2, so the system doesn't generate or store legacy LM and NTLMv1 hashes, and it doesn't store passwords with reversible encryption.
- The baseline hardens credential delegation so delegated credentials remain non-exportable, and enforces CredSSP encryption-oracle protection.

For example, these protections aim to make credential-dumping tools such as Mimikatz (`sekurlsa::logonpasswords`) less effective, because protected memory is harder to read, and extracting the SAM database yields strong NTLMv2 hashes rather than trivially crackable LM hashes.

Lateral movement prevention

This baseline aims to contain breaches by making it harder for attackers to pivot between machines with stolen credentials or sessions:

- Remote UAC filtering strips administrative privileges from local accounts that authenticate over the network, helping defend against pass-the-hash with local admin credentials.
- The baseline requires SMB signing on both clients and servers, helping defend against relay and on-path attacks.
- The baseline signs and encrypts secure channel data to the domain controller, and hardened UNC paths protect NETLOGON and SYSVOL.
- An SMB authentication rate limiter adds a delay after each failed attempt, and the account lockout policy locks accounts after three failed attempts for 15 minutes.

For example, a pass-the-hash attempt with a stolen local admin hash receives only a standard-user token and is denied access to administrative shares, an NTLM relay attempt is impaired because the target requires valid SMB signatures, and brute-force tooling is slowed substantially by the authentication rate limiter.

Persistence and tamper blocking

This baseline makes it harder for attackers to establish long-term footholds or tamper with security mechanisms:

- Secured-core protections build on a Microsoft-signed boot chain, Secure Boot with DMA protection, the UEFI Memory Attributes Table, and kernel-mode hardware-enforced stack protection (kernel shadow stacks) via VBS. The baseline initially enables some of these protections, such as kernel shadow stacks, in audit mode before you switch them to block mode, and several require compatible hardware.
- The baseline enables exploit mitigations such as SEHOP and untrusted-font blocking.
- The baseline disables AutoRun and AutoPlay for all drive types, helping block USB-based malware.
- The baseline reduces insecure installation behaviors - it disables "Always install with elevated privileges" - and blocks consumer Microsoft account authentication.

For example, these settings help prevent a bootkit from loading unsigned boot components and help stop malware on a dropped USB stick from auto-executing, while logging the device insertion.

Auditing and visibility

This baseline ensures the system detects any activity that it doesn't fully block:

- Nearly all advanced audit policy subcategories capture Success and Failure events, and the baseline forces the system to use per-subcategory audit settings.

- The baseline audits logon, credential validation, account management, and sensitive privilege use.
- The baseline audits process creation with command-line capture (Event ID 4688 includes the full command line), so tooling activity leaves a forensic trail.
- The baseline logs file share access, removable storage, and Plug and Play device activity, and the firewall logs dropped and successful connections on all profiles.
- The baseline hardens event log sizes and retention, with the Security log sized to at least 192 MB.

For example, a brute-force attempt generates account lockout and failed-logon events with the source IP and account name, and a privilege-escalation attempt is recorded by sensitive privilege use auditing.

What to expect after applying the baseline

Use the following guidance to plan and carry out a baseline deployment. For at-scale operations, use Azure Policy and Azure Policy machine configuration to monitor and see your compliance score.

> **Important**
>
> After you apply the security baseline, your system's security settings change along with default behaviors. Test carefully before applying these changes in production environments.
>
> You must change your local administrator password after you apply the security baseline for Member server and Workgroup member scenarios.

The following changes are some of the more noticeable ones after you apply the baselines:

- You must change the local administrator password. The new password policy must meet the complexity requirements and a minimum length of 14 characters. This rule applies only to local user accounts; when signing in with a domain account, domain requirements prevail.
- TLS connections are subject to a minimum of TLS/DTLS 1.2 or higher, which might prevent connections to older systems.
- The baseline disables the ability to copy and paste files from RDP sessions. If you need to use this function, run the following command and then restart your device:

  PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/<ServerRoleBeingApplied> -Name
RemoteDesktopServicesDoNotAllowDriveRedirection -Value 0
```

- Connections are subject to a minimum of SMB 3.0 or higher. Connecting to non-Windows systems, such as Linux SAMBA, requires support for SMB 3.0, or you must adjust the baseline.
- If you're currently configuring the same settings with two different methods, one being OSConfig, expect conflicts. With drift control involved, you must remove one of the sources if the parameters differ, to prevent the settings from constantly changing between sources.
- You might run into SID translation errors in specific domain configurations. These errors don't impact the rest of the security baseline definition, and you can ignore them.

Install the OSConfig PowerShell module

Before you can apply a security baseline for the first time, you need to install the OSConfig module from an elevated PowerShell window. Two installation methods are available: online and offline. Follow these instructions for your environment.

Online method

1. Select **Start**, type **PowerShell**, hover over **Windows PowerShell**, and select **Run as administrator**.
2. Run the following command to install the OSConfig module:

   PowerShell

   ```
   Install-Module -Name Microsoft.OSConfig -Scope AllUsers -Force
   ```

   If you're prompted to install or update the NuGet provider, select **Yes**.
3. To verify that the OSConfig module is installed, run the following command:

   PowerShell

   ```
   Get-Module -ListAvailable -Name Microsoft.OSConfig
   ```

Available security baseline scenarios

OSConfig identifies each baseline by a scenario name in the form `SecurityBaseline/WindowsServer/2025/<Role>`. Apply the scenario that matches the role of your device:

Expand table

Server role	Scenario name
Domain controller	`SecurityBaseline/WindowsServer/2025/DomainController`
Member server	`SecurityBaseline/WindowsServer/2025/MemberServer`
Workgroup member	`SecurityBaseline/WindowsServer/2025/WorkgroupMember`

In addition to the role-based security baselines, OSConfig ships companion scenarios you can apply independently:

Expand table

Scenario	Scenario name
Secured-core	`SecuredCore`
Microsoft Defender Antivirus	`Defender/Antivirus/WindowsServer/2025`
Windows LAPS	`LAPS/WindowsServer/2025/MemberServer`

> **Note**
>
> The Windows LAPS scenario is available for member servers only. For standalone (non-domain-joined) systems, you can use Windows LAPS through Azure Arc-enabled servers.

Manage Windows Server 2025 security baselines

Apply the appropriate security baselines based on the Windows Server role of your device:

- Domain controller (DC)
- Member server (domain-joined)
- Workgroup member server (non-domain-joined)

OSConfig powers the baseline experience. After you apply a baseline, OSConfig automatically protects your security baseline settings from drift, which is one of the key features of its security platform.

> **Note**
>
> For Azure Arc-connected devices, you can apply the security baselines before or after connecting. But if the role of your server changes after the connection, you must delete and reapply the assignment to make sure that the machine configuration platform can detect the role change. For more information about deleting an assignment, see Deletion of guest assignments from Azure Policy.

To apply a baseline, verify that a baseline is in place, remove a baseline, or view detailed compliance information for OSConfig in PowerShell, use the commands on the following tabs.

Configure

To apply the baseline for a domain-joined device, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/MemberServer -Default
```

To apply the baseline for a device that's in a workgroup, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/WorkgroupMember -Default
```

To apply the baseline for a device that's configured as the DC, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/DomainController -Default
```

To apply the Secured-core baseline for a device, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario SecuredCore -Default
```

To apply the Microsoft Defender Antivirus baseline for a device, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario Defender/Antivirus/WindowsServer/2025
-Default
```

To apply the Windows LAPS baseline for a member server, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario LAPS/WindowsServer/2025/MemberServer -
Default
```

> **Note**
>
> - When you *apply* or *remove* a security baseline, you must restart the device for changes to take effect.
> - When you *customize* a security baseline, you might need to restart the device for changes to take effect, depending on which security features you modify.
> - During the *removal* process, when OSConfig reverts security settings, it can't guarantee that these settings return to their premanaged configuration. The outcome depends on the specific settings within the security baseline. This behavior aligns with the capabilities that Microsoft Intune policies provide. To learn more, see Remove a security baseline assignment.

Customize Windows Server 2025 security baselines

After you complete the security baseline configuration, you can modify the security settings while maintaining drift control. Customizing the security values gives you more control over your organization's security policies, depending on your environment's specific needs.

To change the default value of `AuditDetailedFileShare` from `2` to `3` for your member server, run the following command:

PowerShell

```
Set-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/MemberServer -Setting AuditDetailedFileShare -
Value 3
```

To verify that the new value is in effect, run the following command:

PowerShell

```
Get-OSConfigDesiredConfiguration -Scenario
SecurityBaseline/WindowsServer/2025/MemberServer -Setting AuditDetailedFileShare
```

> **Note**
>
> Depending on which security settings you customize, you need to provide certain input. These inputs are:
>
> - `MessageTextUserLogon`
> - `MessageTextUserLogonTitle`
> - `RenameAdministratorAccount`
> - `RenameGuestAccount`
>
> After you provide the necessary input, select the **Enter** key to proceed.

Security baseline versioning

The OSConfig security baselines are versioned and ship inside the OSConfig PowerShell module. A newer module brings the latest baseline version, which fully supersedes the previous one rather than acting as an incremental patch on top of it. When you apply a scenario, OSConfig configures the complete set of settings defined by the baseline version in the module installed on the device.

The scenario name doesn't include a version token. Applying `SecurityBaseline/WindowsServer/2025/MemberServer`, for example, always configures the current Windows Server 2025 baseline version present in your installed OSConfig module.

To see which OSConfig module version is installed, run:

PowerShell

```
Get-Module -ListAvailable -Name Microsoft.OSConfig
```

Update between versions

To move a device from an older baseline version to a newer one, update the OSConfig module and then reapply the scenario. Because each baseline version fully supersedes the previous one, reapplying brings the device to the complete new set of settings.

1. Update the OSConfig module to the latest version:

   PowerShell

   ```
   Update-Module -Name Microsoft.OSConfig
   ```

2. Reapply the scenario that matches the role of your device. For example, for a member server, run:

   PowerShell

   ```
   Set-OSConfigDesiredConfiguration -Scenario
   SecurityBaseline/WindowsServer/2025/MemberServer -Default
   ```

   > **Note**
   >
   > If an earlier version of the baseline is already applied, you're prompted to remove the previously applied version before OSConfig configures the new one. Confirm the prompt to continue.

3. Restart the device for the changes to take effect.

Windows Server 2025 security baseline changelog

The current Windows Server 2025 baseline expands coverage over the previous version. The number of settings increases for each server role:

Expand table

Server role	Previous version	Current version
Domain controller	319	347

Server role	Previous version	Current version
Member server	318	344
Workgroup member	294	319

For the complete list of settings included in each baseline, see the OSConfig security baselines on GitHub.

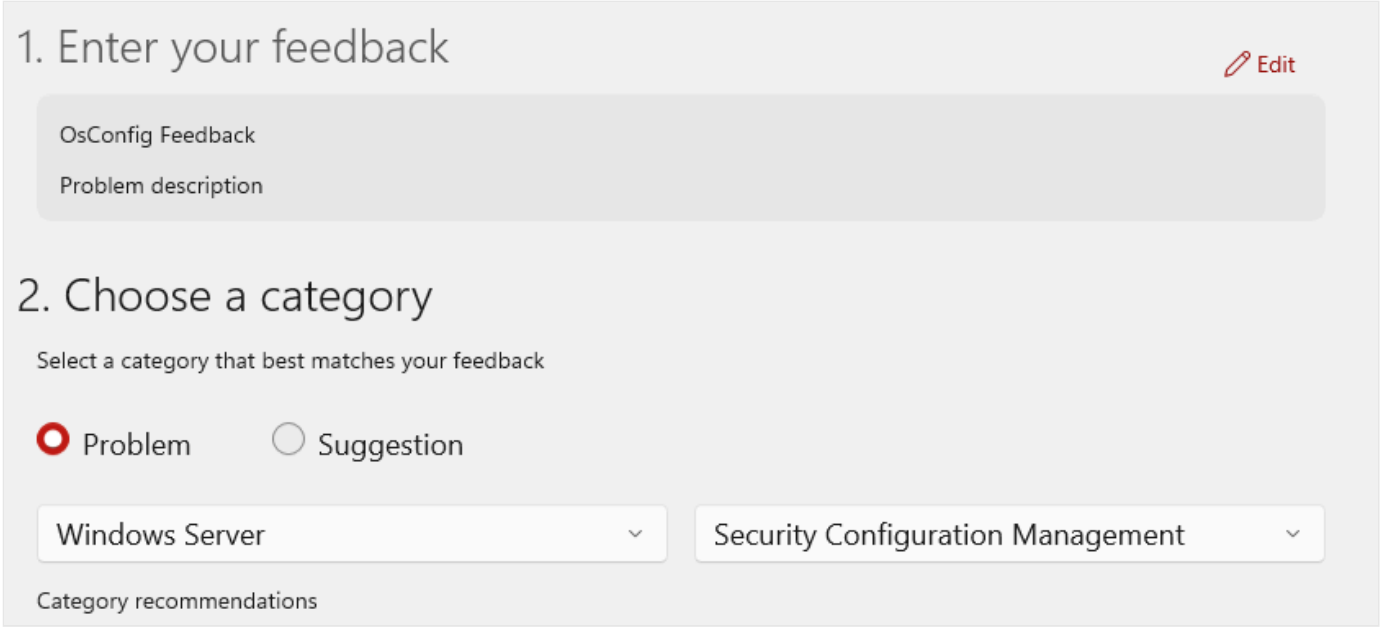
Provide feedback for OSConfig

If you experience a problem or have a suggestion after applying a security baseline, submit feedback through Feedback Hub. Both Windows Server and the Windows client offer this capability. To open Feedback Hub, select **Start**, type **Feedback Hub**, and open the app, or go to the Feedback Hub website. For more information about submitting feedback, see Deeper look at feedback.

To submit OSConfig feedback:

1. Under **Enter your feedback**, provide a clear title and description. Include the installed OSConfig module version (which `Get-Module -ListAvailable -Name Microsoft.OSConfig` returns) so the team can triage your report.
2. Under **Choose a category**, select **Problem** or **Suggestion**.
3. In the category drop-down list, select **Windows Server**, and then select **Security Configuration Management** in the secondary drop-down list.
4. Add any relevant details or attachments, and then submit your feedback.

The following example shows the Feedback Hub with the **Windows Server** category and the **Security Configuration Management** subcategory selected.



Related content

- OSConfig overview

Last updated on 07/02/2026

Configure App Control for Business by using OSConfig

Article • 05/20/2025 • Applies to: ✅ Windows Server 2025

App Control for Business is a software-based security layer that reduces attack surface by enforcing an explicit list of software that's allowed to run. Microsoft developed a default policy for Windows Server 2025, which you can implement on the server by using Windows PowerShell cmdlets. App Control implementation is facilitated through the OSConfig security configuration platform and provides two main operation modes:

- **Audit mode**: Allows untrusted code to run while events are recorded.
- **Enforcement mode**: Disallows untrusted code from running while events are recorded.

To learn more about these security event logs, see Understanding App Control events and Understanding App Control event tags.

Because App Control is a component of Windows Server 2025, deployment of its policies provides easier access to audit mode and enforcement mode through PowerShell. By default, audit-mode App Control policies aren't set in Window Server 2025. Instead, organizations can use businesses can use the OSConfig tool to add these policies. OSConfig includes base policies (unsigned) that can be customized with supplemental policies to meet specific business needs. To learn more about this security enhancement, see Application Control for Windows.

> ⓘ **Note**
>
> An Azure Monitor workbook simplifies the process of reviewing audit or block events that the operating system emits when App Control is activated. This workbook provides insights on file audit and block activity, along with:
>
> - Collecting and sending Windows event logs for App Control for Business to your Log Analytics workspace.
> - Identifying file and policy event activities by using various dashboards, charts, filters, and export capabilities. These features help you analyze and troubleshoot the effects and the status of your App Control policies.
> - Refining your App Control policies by exporting the workbook data and ingesting it in the Microsoft App Control Wizard.
>
> To begin using the Azure Monitor workbook for App Control for Business, see How to get insights into App Control for Business (WDAC) events.

Prerequisites

- You must be running a production-signed Windows Server 2025 build on your device. This requirement ensures compliance with the App Control for Business policies.

  > **⊗ Caution**
  >
  > Flight-signed binaries aren't permitted. Failure to comply with this requirement results in the inability to start your device.

- The OSConfig PowerShell module must be installed on your server device. See Install the OSConfig PowerShell module for details.

- You must be running Windows 10 version 1909 or later on your client device, and have the App Control Wizard installed.

> **ⓘ Note**
>
> If the client device doesn't have .NET Desktop Runtime 8.0 or later installed, the App Control Wizard prompts you to download and install this application.

Manage default policies

Configure

To configure the default App Control policies in audit mode, open PowerShell as admin and run the following command:

```PowerShell
Set-OSConfigDesiredConfiguration -Scenario
AppControl\WS2025\DefaultPolicy\Audit -Default
Set-OSConfigDesiredConfiguration -Scenario
AppControl\WS2025\AppBlockList\Audit -Default
```

To configure the default App Control policies in enforcement mode, open PowerShell as admin and run the following command:

```PowerShell
Set-OSConfigDesiredConfiguration -Scenario
AppControl\WS2025\DefaultPolicy\Enforce -Default
```

```
Set-OSConfigDesiredConfiguration -Scenario
AppControl\WS2025\AppBlockList\Enforce -Default
```

Monitor event logs

To view captured events after you apply the App Control policy, choose any third-party application that you want to run on your device. If you set the App Control policy in audit mode, check if the operating system emitted event ID **3076** for any third-party applications. If you set the policy in enforcement mode, check if the operating system emitted event ID **3077**.

The system detects attempts by the third-party application to access restricted content and takes measures to block access. To view and export these event logs, see the following steps:

View

1. Right-click **Start**, and then select **Event Viewer**.
2. On the left pane, go to **Applications and Service Logs\Microsoft\Windows\CodeIntegrity\Operational**.
3. On the middle pane, look for event ID **3076** for audit mode or **3077** for enforcement mode.

Configure supplemental policies

To create supplemental App Control policies, follow these steps:

1. Copy the `.evtx` log file from your server to your client device.
2. On the client device, open the **App Control Wizard**.
3. On the **Home** screen, select **Policy Editor**.
4. On the **Policy Editor** screen, select **Convert Event Logs to a Policy**. Then, under **Parse Event Log evtx Files to Policy**, select **Parse Log File(s)**.
5. In the **Choose event logs to convert to policy** dialog, locate your `.evtx` file, and then select **Open**. Select **OK** in the prompt, and then select **Next**.
6. On the **File Rules** screen, under **Filename**, select the file that you want to add to the policy.
7. Under **Rule Type**, select **Path**, and then select **+ Add Allow**.
8. Repeat step 7 for all items that you want to add to the policy, and then select **Next**.

ⓘ Note

By default, supplemental policies are stored in **C:\Users\Username\Documents** in XML format.

After the supplemental policy is generated, copy the XML file to your server and run the following script:

PowerShell

```
$policyPath = "<Path to the XML policy file>"

# Reset GUID (best practice)

Set-CIPolicyIdInfo -FilePath $policyPath -ResetPolicyID

# Set policy version (VersionEx in the XML file)

$policyVersion = "1.0.0.1"

Set-CIPolicyVersion -FilePath $policyPath -Version $policyVersion

# Set policy info (PolicyName and PolicyID in the XML file)

Set-CIPolicyIdInfo -FilePath $policyPath -PolicyID "<App name>-
Policy_$policyVersion" -PolicyName "<App name>-Policy" # E.g. Set-CIPolicyIdInfo -
FilePath $policyPath -PolicyID "Chrome-Policy_$policyVersion" -PolicyName "Chrome-
Policy"

$base = "{9214D8EE-9B0F-4972-9073-A04E917D7989}"

Set-CIPolicyIdInfo -FilePath $policyPath -SupplementsBasePolicyID $base

#Set the new policy into the system

Set-OSConfigDesiredConfiguration -Scenario AppControl -Name Policies -Value
$policyPath
```

To verify that the supplemental policies were applied, monitor event IDs **3076** and **3077** as described earlier in Monitor event logs. Verify that the operating system didn't generate any new events.

Query policies

To view the policies that are currently in effect in your environment, run the following command:

PowerShell

```
(Get-OSConfigDesiredConfiguration -Scenario AppControl).Value.PolicyInfo | Where-
Object { $_.IsEffective -eq $true }
```

To view policies that are currently inactive in your environment, run the following command:

PowerShell

```
(Get-OSConfigDesiredConfiguration -Scenario AppControl).Value.PolicyInfo | Where-
Object { $_.IsEffective -eq $false }
```

Output for these queries varies based on your policy configuration needs.

Related content

- CiTool technical reference
- ConfigCI PowerShell reference

Guarded fabric and shielded VMs

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

One of the most important goals of providing a hosted environment is to guarantee the security of the virtual machines running in the environment. As a cloud service provider or enterprise private cloud administrator, you can use a guarded fabric to provide a more secure environment for VMs. A guarded fabric consists of one Host Guardian Service (HGS) - typically, a cluster of three nodes - plus one or more guarded hosts, and a set of shielded virtual machines (VMs).

> **ⓘ Important**
>
> Ensure that you have installed the latest cumulative update before you deploy shielded virtual machines in production.

Videos, blog, and overview topic about guarded fabrics and shielded VMs

- Video: How to protect your virtualization fabric from insider threats with Windows Server 2019
- Video: Introduction to Shielded Virtual Machines in Windows Server 2016
- Video: Dive into Shielded VMs with Windows Server 2016 Hyper-V
- Video: Deploying Shielded VMs and a Guarded Fabric with Windows Server 2016
- Blog: Datacenter and Private Cloud Security Blog
- Overview: Guarded fabric and shielded VMs overview

Planning topics

- Planning guide for hosters
- Planning guide for tenants

Deployment topics

- Deployment Guide
  - Quick start

- Deploy HGS
- Deploy guarded hosts
  - Configuring the fabric DNS for hosts that will become guarded hosts
  - Deploy a guarded host using AD mode
  - Deploy a guarded host using TPM mode
  - Confirm guarded hosts can attest
  - Shielded VMs - Hosting service provider deploys guarded hosts in VMM
- Deploy shielded VMs
  - Create a shielded VM template
  - Prepare a VM Shielding helper VHD
  - Set up Windows Azure Pack
  - Create a shielding data file
  - Deploy a shielded VM by using Windows Azure Pack
  - Deploy a shielded VM by using Virtual Machine Manager

Operations and management topic

- Managing the Host Guardian Service

Feedback

Was this page helpful? Yes No

Guarded fabric and shielded VMs overview

Article • 02/17/2023 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

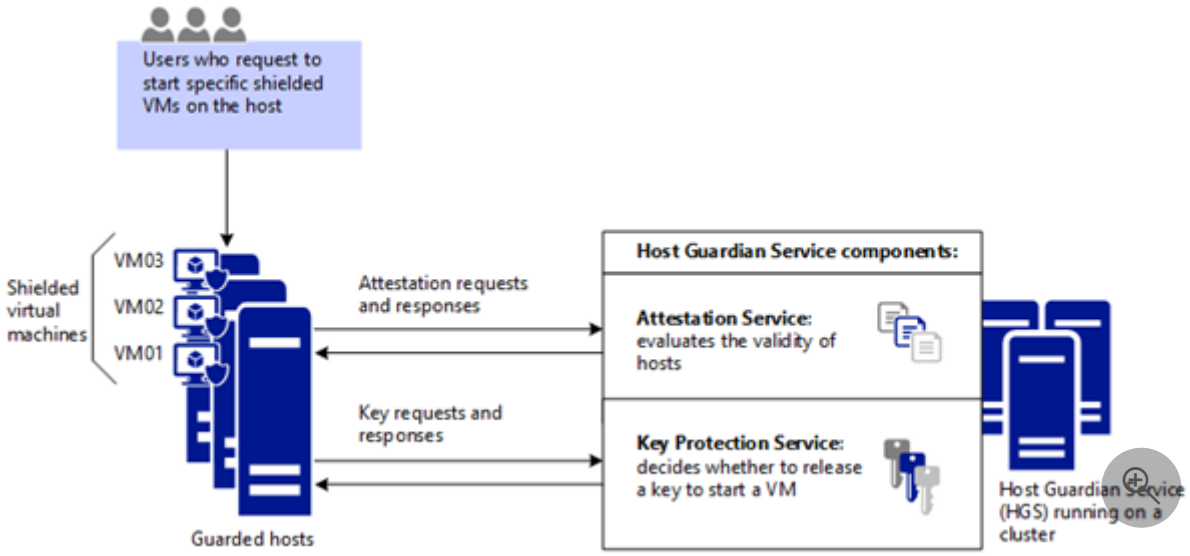
Virtualization security is a major investment area in Hyper-V. In addition to protecting hosts or other virtual machines from a virtual machine running malicious software, we also need to protect virtual machines from a compromised host. This is a fundamental danger for every virtualization platform today, whether it's Hyper-V, VMware or any other. If a virtual machine gets out of an organization (either maliciously or accidentally), that virtual machine can be run on any other system. Protecting high value assets in your organization, such as domain controllers, sensitive file servers, and HR systems, is a top priority.

To help protect against compromised virtualization fabric, Windows Server 2016 Hyper-V introduced shielded VMs. A shielded VM is a generation 2 VM (supported on Windows Server 2012 and later) that has a virtual TPM, is encrypted using BitLocker, and can run only on healthy and approved hosts in the fabric. Shielded VMs and guarded fabric enable cloud service providers or enterprise private cloud administrators to provide a more secure environment for tenant VMs.

A guarded fabric consists of:

- 1 Host Guardian Service (HGS) (typically, a cluster of 3 nodes).
- 1 or more guarded hosts.
- A set of shielded virtual machines. The diagram below shows how the Host Guardian Service uses attestation to ensure that only known, valid hosts can start the shielded VMs, and key protection to securely release the keys for shielded VMs.

When a tenant creates shielded VMs that run on a guarded fabric, the Hyper-V hosts and the shielded VMs themselves are protected by the HGS. The HGS provides two distinct services: attestation and key protection. The Attestation service ensures only trusted Hyper-V hosts can run shielded VMs while the Key Protection Service provides the keys necessary to power them on and to live migrate them to other guarded hosts.



To learn more, see this video on Introduction to shielded virtual machines.

Attestation modes in the Guarded Fabric solution

The HGS supports different attestation modes for a guarded fabric:

- TPM-trusted attestation (hardware-based)
- Host key attestation (based on asymmetric key pairs)

TPM-trusted attestation is recommended because it offers stronger assurances, as explained in the following table, but it requires that your Hyper-V hosts have TPM 2.0. If you currently don't have TPM 2.0 or any TPM, you can use host key attestation. If you decide to move to TPM-trusted attestation when you acquire new hardware, you can switch the attestation mode on the Host Guardian Service with little or no interruption to your fabric.

Expand table

Attestation mode you choose for hosts	Host assurances
TPM-trusted attestation: Offers the strongest possible protections but also requires more configuration steps. Host hardware and firmware must include TPM 2.0 and UEFI 2.3.1 with Secure Boot enabled.	Guarded hosts are approved based on their TPM identity, Measured Boot sequence, and code integrity policies to ensure they only run approved code.
Host key attestation: Intended to support existing host hardware where TPM 2.0 isn't available. Requires fewer configuration steps and is compatible with commonplace server hardware.	Guarded hosts are approved based on possession of the key.

Another mode named **Admin-trusted attestation** is deprecated beginning with Windows Server 2019. This mode was based on guarded host membership in a designated Active Directory Domain Services (AD DS) security group. Host key attestation provides similar host identification and is easier to set up.

Assurances provided by the Host Guardian Service

HGS, together with the methods for creating shielded VMs, help provide the following assurances.

Expand table

Type of assurance for VMs	Shielded VM assurances, from Key Protection Service and from creation methods for shielded VMs
BitLocker encrypted disks (OS disks and data disks)	Shielded VMs use BitLocker to protect their disks. The BitLocker keys needed to boot the VM and decrypt the disks are protected by the shielded VM's virtual TPM using industry-proven technologies such as secure measured boot. While shielded VMs only automatically encrypt and protect the operating system disk, you can encrypt data drives attached to the shielded VM as well.
Deployment of new shielded VMs from "trusted" template disks/images	When deploying new shielded VMs, tenants are able to specify which template disks they trust. Shielded template disks have signatures that are computed at a point in time when their content is deemed trustworthy. The disk signatures are then stored in a signature catalog, which tenants securely provide to the fabric when creating shielded VMs. During provisioning of shielded VMs, the signature of the disk is computed again and compared to the trusted signatures in the catalog. If the signatures match, the shielded VM is deployed. If the signatures don't match, the shielded template disk is deemed untrustworthy, and deployment fails.
Protection of passwords and other secrets when a shielded VM is created	When creating VMs, it's necessary to ensure that VM secrets, such as the trusted disk signatures, RDP certificates, and the password of the VM's local Administrator account, are not divulged to the fabric. These secrets are stored in an encrypted file called a shielding data file (a .PDK file), which is protected by tenant keys and uploaded to the fabric by the tenant. When a shielded VM is created, the tenant selects the shielding data to use which securely provides these secrets only to the trusted components within the guarded fabric.
Tenant control of where the VM can be started	Shielding data also contains a list of the guarded fabrics on which a particular shielded VM is permitted to run. This is useful, for example, in cases where a shielded VM typically resides in an on-premises private cloud but may need to be migrated to another (public or private) cloud for

Type of assurance for VMs	Shielded VM assurances, from Key Protection Service and from creation methods for shielded VMs
	disaster recovery purposes. The target cloud or fabric must support shielded VMs and the shielded VM must permit that fabric to run it.

What is shielding data and why is it necessary?

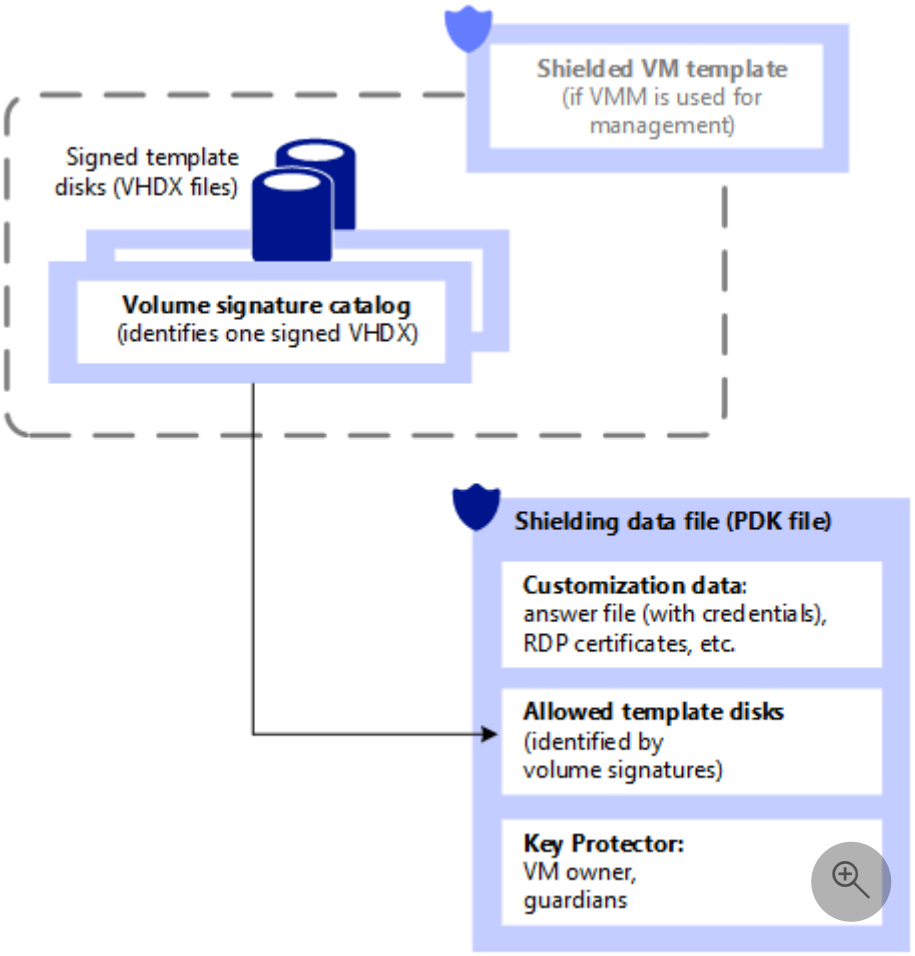
A shielding data file (also called a provisioning data file or PDK file) is an encrypted file that a tenant or VM owner creates to protect important VM configuration information, such as the administrator password, RDP and other identity-related certificates, domain-join credentials, and so on. A fabric administrator uses the shielding data file when creating a shielded VM, but is unable to view or use the information contained in the file.

Among others, shielding data files contain secrets such as:

- Administrator credentials
- An answer file (unattend.xml)
- A security policy that determines whether VMs created using this shielding data are configured as shielded or encryption supported
    - Remember, VMs configured as shielded are protected from fabric admins whereas encryption supported VMs aren't
- An RDP certificate to secure remote desktop communication with the VM
- A volume signature catalog that contains a list of trusted, signed template-disk signatures that a new VM is allowed to be created from
- A Key Protector (or KP) that defines which guarded fabrics a shielded VM is authorized to run on

The shielding data file (PDK file) provides assurances that the VM will be created in the way the tenant intended. For example, when the tenant places an answer file (unattend.xml) in the shielding data file and delivers it to the hosting provider, the hosting provider can't view or make changes to that answer file. Similarly, the hosting provider can't substitute a different VHDX when creating the shielded VM, because the shielding data file contains the signatures of the trusted disks that shielded VMs can be created from.

The following figure shows the shielding data file and related configuration elements.



What are the types of virtual machines that a guarded fabric can run?

Guarded fabrics are capable of running VMs in one of three possible ways:

1. A normal VM offering no protections above and beyond previous versions of Hyper-V
2. An encryption-supported VM whose protections can be configured by a fabric admin
3. A shielded VM whose protections are all switched on and can't be disabled by a fabric admin

Encryption-supported VMs are intended for use where the fabric administrators are fully trusted. For example, an enterprise might deploy a guarded fabric in order to ensure VM disks are encrypted at-rest for compliance purposes. Fabric administrators can continue to use convenient management features, such VM console connections, PowerShell Direct, and other day-to-day management and troubleshooting tools.

Shielded VMs are intended for use in fabrics where the data and state of the VM must be protected from both fabric administrators and untrusted software that might be running on the Hyper-V hosts. For example, shielded VMs will never permit a VM

console connection whereas a fabric administrator can turn this protection on or off for encryption supported VMs.

The following table summarizes the differences between encryption-supported and shielded VMs.

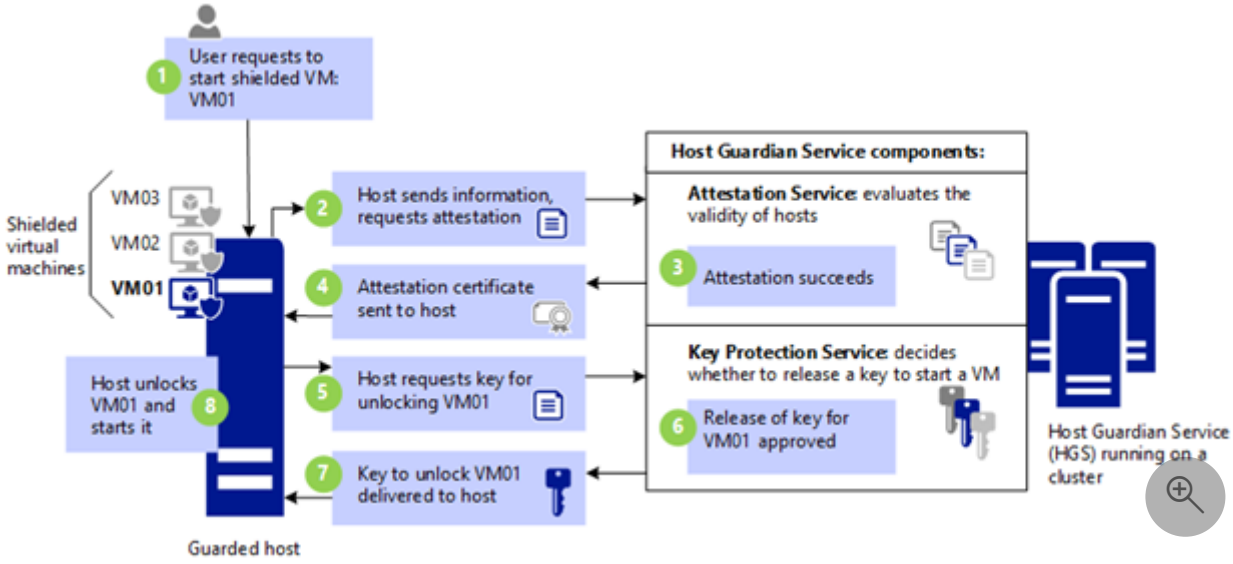
Expand table

Capability	Generation 2 Encryption Supported	Generation 2 Shielded
Secure Boot	Yes, required but configurable	Yes, required and enforced
Vtpm	Yes, required but configurable	Yes, required and enforced
Encrypt VM state and live migration traffic	Yes, required but configurable	Yes, required and enforced
Integration components	Configurable by fabric admin	Certain integration components blocked (e.g. data exchange, PowerShell Direct)
Virtual Machine Connection (Console), HID devices (e.g. keyboard, mouse)	On, can't be disabled	Enabled on hosts beginning with Windows Server version 1803; Disabled on earlier hosts
COM/Serial ports	Supported	Disabled (can't be enabled)
Attach a debugger (to the VM process)[1]	Supported	Disabled (can't be enabled)

[1] Traditional debuggers that attach directly to a process, such as WinDbg.exe, are blocked for shielded VMs because the VM's worker process (VMWP.exe) is a protected process light (PPL). Alternative debugging techniques, such as those used by LiveKd.exe, aren't blocked. Unlike shielded VMs, the worker process for encryption supported VMs doesn't run as a PPL so traditional debuggers, such as WinDbg.exe, will continue to function normally.

Both shielded VMs and encryption-supported VMs continue to support commonplace fabric management capabilities, such as Live Migration, Hyper-V replica, VM checkpoints, and so on.

The Host Guardian Service in action: How a shielded VM is powered on



1. **VM01 is powered on.** Before a guarded host can power on a shielded VM, it must first be affirmatively attested that it's healthy. To prove it's healthy, it must present a certificate of health to the Key Protection service (KPS). The certificate of health is obtained through the attestation process.

2. **Host requests attestation.** The guarded host requests attestation. The mode of attestation is dictated by the Host Guardian Service:

    - **TPM-trusted attestation**: Hyper-V host sends information that includes:

        - TPM-identifying information (its endorsement key)
        - Information about processes that were started during the most recent boot sequence (the TCG log)
        - Information about the Code Integrity (CI) policy that was applied on the host.

        Attestation happens when the host starts and every 8 hours thereafter. If for some reason a host doesn't have an attestation certificate when a VM tries to start, this also triggers attestation.

    - **Host key attestation**: Hyper-V host sends the public half of the key pair. HGS validates the host key is registered.

    - **Admin-trusted attestation**: Hyper-V host sends a Kerberos ticket, which identifies the security groups that the host is in. HGS validates that the host

belongs to a security group that was configured earlier by the trusted HGS admin.

3. **Attestation succeeds (or fails).** The attestation mode determines which checks are needed to successfully attest the host is healthy. With TPM-trusted attestation, the host's TPM identity, boot measurements, and code integrity policy are validated. With host key attestation, only registration of the host key is validated.

4. **Attestation certificate sent to host.** Assuming attestation was successful, a health certificate is sent to the host and the host is considered "guarded" (authorized to run shielded VMs). The host uses the health certificate to authorize the Key Protection Service to securely release the keys needed to work with shielded VMs

5. **Host requests VM key.** Guarded host does not have the keys needed to power on a shielded VM (VM01 in this case). To obtain the necessary keys, the guarded host must provide the following to KPS:

    - The current health certificate
    - An encrypted secret (a Key Protector or KP) that contains the keys necessary to power on VM01. The secret is encrypted using other keys that only KPS knows.

6. **Release of key.** KPS examines the health certificate to determine its validity. The certificate must not have expired and KPS must trust the attestation service that issued it.

7. **Key is returned to host.** If the health certificate is valid, KPS attempts to decrypt the secret, and securely return the keys needed to power on the VM. The keys are encrypted to the guarded host's VBS.

8. **Host powers on VM01.**

Guarded fabric and shielded VM glossary

Expand table

Term	Definition
Host Guardian Service (HGS)	A Windows Server role that is installed on a secured cluster of bare-metal servers that is able to measure the health of a Hyper-V host and release keys to healthy Hyper-V hosts when powering-on or live migrating shielded VMs. These two capabilities are fundamental to a shielded VM solution and are referred to as the **Attestation service** and **Key Protection Service** respectively.

Term	Definition
guarded host	A Hyper-V host on which shielded VMs can run. A host can only be considered *guarded* when it has been deemed healthy by HGS' Attestation service. Shielded VMs can't be powered-on or live migrated to a Hyper-V host that has not yet attested or that failed attestation.
guarded fabric	This is the collective term used to describe a fabric of Hyper-V hosts and their Host Guardian Service that has the ability to manage and run shielded VMs.
shielded virtual machine (VM)	A virtual machine that can only run on guarded hosts and is protected from inspection, tampering and theft from malicious fabric admins and host malware.
fabric administrator	A public or private cloud administrator that can manage virtual machines. In the context of a guarded fabric, a fabric administrator doesn't have access to shielded VMs, or the policies that determine which hosts shielded VMs can run on.
HGS administrator	A trusted administrator in the public or private cloud that has the authority to manage the policies and cryptographic material for guarded hosts, that is, hosts on which a shielded VM can run.
provisioning data file or shielding data file (PDK file)	An encrypted file that a tenant or user creates to hold important VM configuration information and to protect that information from access by others. For example, a shielding data file can contain the password that will be assigned to the local Administrator account when the VM is created.
Virtualization-based Security (VBS)	A Hyper-V based processing and storage environment that is protected from administrators. Virtual Secure Mode provides the system with the ability to store operating system keys that aren't visible to an operating system administrator.
virtual TPM	A virtualized version of a Trusted Platform Module (TPM). Beginning with Hyper-V in Windows Server 2016, you can provide a virtual TPM 2.0 device so that virtual machines can be encrypted, just as a physical TPM allows a physical machine to be encrypted.

Additional References

- Guarded fabric and shielded VMs
- Blog: Datacenter and Private Cloud Security Blog
- Video: Introduction to Shielded Virtual Machines
- Video: Dive into Shielded VMs with Windows Server 2016 Hyper-V

Feedback

Was this page helpful? Yes No

Planning a Guarded Fabric

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The following topics cover planning for the deployment of a guarded fabric and shielded virtual machines (VMs):

- Guarded Fabric and Shielded VM Planning Guide for Hosters
- Compatible hardware with Windows Server 2016 Virtualization-based protection of Code Integrity
- Guarded Fabric and Shielded VM Planning Guide for Tenants

Feedback

Was this page helpful? Yes No

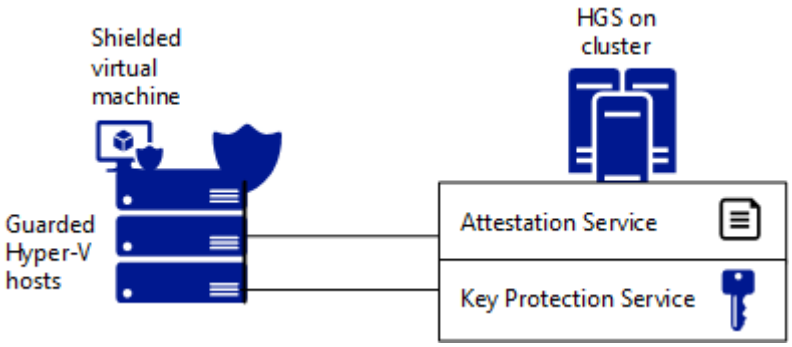
Guarded Fabric and Shielded VM Planning Guide for Hosters

Article • 06/09/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic covers planning decisions that will need to be made to enable shielded virtual machines to run on your fabric. Whether you upgrade an existing Hyper-V fabric or create a new fabric, running shielded VMs consists of two main components:

- The Host Guardian Service (HGS) provides attestation and key protection so that you can make sure that shielded VMs will run only on approved and healthy Hyper-V hosts.
- Approved and healthy Hyper-V hosts on which shielded VMs (and regular VMs) can run — these are known as guarded hosts.



Decision #1: Trust level in the fabric

How you implement the Host Guardian Service and guarded Hyper-V hosts will depend mainly on the strength of trust that you are looking to achieve in your fabric. The strength of trust is governed by the attestation mode. There are two mutually-exclusive options:

1. TPM-trusted attestation

   If your goal is to help protect virtual machines from malicious admins or a compromised fabric, then you will use TPM-trusted attestation. This option works well for multi-tenant hosting scenarios as well as for high-value assets in enterprise environments, such as domain controllers or content servers like SQL or SharePoint. Hypervisor-protected code integrity (HVCI) policies are measured and their validity enforced by HGS before the host is permitted to run shielded VMs.

2. Host key attestation

   If your requirements are primarily driven by compliance that requires virtual machines be encrypted both at rest as well as in-flight, then you will use host key attestation. This option works well for general purpose datacenters where you are comfortable with Hyper-V host and fabric administrators having access to the guest operating systems of virtual machines for day-to-day maintenance and operations.

   In this mode, the fabric admin is solely responsible for ensuring the health of the Hyper-V hosts. Since HGS plays no part in deciding what is or is not allowed to run, malware and debuggers will function as designed.

   However, debuggers that attempt to attach directly to a process (such as WinDbg.exe) are blocked for shielded VMs because the VM's worker process (VMWP.exe) is a protected process light (PPL). Alternative debugging techniques, such as those used by LiveKd.exe, are not blocked. Unlike shielded VMs, the worker process for encryption supported VMs does not run as a PPL so traditional debuggers like WinDbg.exe will continue to function normally.

   A similar attestation mode named Admin-trusted attestation (Active Directory-based) is deprecated beginning with Windows Server 2019.

The trust level you choose will dictate the hardware requirements for your Hyper-V hosts as well as the policies that you apply on the fabric. If necessary, you can deploy your guarded fabric using existing hardware and admin-trusted attestation and then convert it to TPM-trusted attestation when the hardware has been upgraded and you need to strengthen fabric security.

Decision #2: Existing Hyper-V fabric versus a new separate Hyper-V fabric

If you have an existing fabric (Hyper-V or otherwise), it is very likely that you can use it to run shielded VMs along with regular VMs. Some customers choose to integrate shielded VMs into their existing tools and fabrics while others separate the fabric for business reasons.

HGS admin planning for the Host Guardian Service

Deploy the Host Guardian Service (HGS) in a highly secure environment, whether that be on a dedicated physical server, a shielded VM, a VM on an isolated Hyper-V host (separated from the fabric it's protecting), or one logically separated by using a different Azure subscription.

Expand table

Area	Details
Installation requirements	• One server (three-node cluster recommended for high availability) • For fallback, at least two HGS servers are required • Servers can be either virtual or physical (physical server with TPM 2.0 recommended; TPM 1.2 also supported) • Server Core installation of Windows Server 2016 or later • Network line of sight to the fabric allowing HTTP or fallback configuration • HTTPS certificate recommended for access validation
Sizing	Each mid-size (8 core/4 GB) HGS server node can handle 1,000 Hyper-V hosts.
Management	Designate specific people who will manage HGS. They should be separate from fabric administrators. For comparison, HGS clusters can be thought of in the same manner as a Certificate Authority (CA) in terms of administrative isolation, physical deployment and overall level of security sensitivity.
Host Guardian Service Active Directory	By default, HGS installs its own internal Active Directory for management. This is a self-contained, self-managed forest and is the recommended configuration to help isolate HGS from your fabric. If you already have a highly privileged Active Directory forest that you use for isolation, you can use that forest instead of the HGS default forest. It is important that HGS is not joined to a domain in the same forest as the Hyper-V hosts or your fabric management tools. Doing so could allow a fabric admin to gain control over HGS.
Disaster recovery	There are three options: 1. Install a separate HGS cluster in each datacenter and authorize shielded VMs to run in both the primary and the backup datacenters. This avoids the need to stretch the cluster across a WAN and allows you to isolate virtual machines such that they run only in their designated site. 2. Install HGS on a stretch cluster between two (or more) datacenters. This provides resiliency if the WAN goes down, but pushes the limits of failover clustering. You cannot isolate workloads to one site; a VM authorized to run in one site can run on any other. 3. Register your Hyper-V host with another HGS as failover.

Area	Details
	You should also backup every HGS by exporting its configuration so that you can always recover locally. For more information, see Export-HgsServerState and Import-HgsServerState.
Host Guardian Service keys	A Host Guardian Service uses two asymmetric key pairs — an encryption key and a signing key — each represented by an SSL certificate. There are two options to generate these keys: 1. Internal certificate authority – you can generate these keys using your internal PKI infrastructure. This is suitable for a datacenter environment. 2. Publicly trusted certificate authorities – use a set of keys obtained from a publicly trusted certificate authority. This is the option that hosters should use. Note that while it is possible to use self-signed certificates, it is not recommended for deployment scenarios other than proof-of-concept labs. In addition to having HGS keys, a hoster can use "bring your own key," where tenants can provide their own keys so that some (or all) tenants can have their own specific HGS key. This option is suitable for hosters that can provide an out-of-band process for tenants to upload their keys.
Host Guardian Service key storage	For the strongest possible security, we recommend that HGS keys are created and stored exclusively in a Hardware Security Module (HSM). If you are not using HSMs, applying BitLocker on the HGS servers is strongly recommended.

Fabric admin planning for guarded hosts

Expand table

Area	Details
Hardware	• Host key attestation: You can use any existing hardware as your guarded host. There are a few exceptions (to make sure that your host can use new security mechanisms beginning with Windows Server 2016, see Compatible hardware with Windows Server 2016 Virtualization-based protection of Code Integrity. • TPM-trusted attestation: You can use any hardware that has the Hardware Assurance Additional Qualification as long as it is configured appropriately (see Server configurations that are compliant with Shielded VMs and Virtualization-based protection of code integrity for the specific configuration). This includes TPM 2.0, and UEFI version 2.3.1c and above.
OS	We recommend using Server Core option for the Hyper-V host OS.

Area	Details
Performance implications	Based on performance testing, we anticipate a roughly 5% density-difference between running shielded VMs and non-shielded VMs. This means that if a given Hyper-V host can run 20 non-shielded VMs, we expect that it can run 19 shielded VMs. Make sure to verify sizing with your typical workloads. For example, there might be some outliers with intensive write-oriented IO workloads that will further affect the density difference.
Branch office considerations	Beginning with Windows Server version 1709, you can specify a fallback URL for a virtualized HGS server running locally as a shielded VM in the branch office. The fallback URL can be used when the branch office loses connectivity to HGS servers in the datacenter. On previous versions of Windows Server, a Hyper-V host running in a branch office needs connectivity to the Host Guardian Service to power-on or to live migrate shielded VMs. For more information, see Branch office considerations.

Feedback

Was this page helpful? Yes No

Compatible hardware with Windows Server Virtualization-based protection of Code Integrity

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows Server 2016 introduced a new Virtualization-based code protection to help protect physical and virtual machines from attacks that modify system code. To achieve this high protection level, Microsoft works in tandem with the computer hardware manufactures (Original Equipment Manufacturers, or OEMs) to prevent malicious writes into system execution code. This protection can be applied to any system and is being used as one of the building blocks for implementing the Hyper-V host health for shielded virtual machines (VMs).

As with any hardware based protection, some systems might not be compliant due to issues such as incorrect marking of memory pages as executables or by actually trying to modify code at run time, which may result in unexpected failures including data loss or a blue screen error (also called a stop error).

To be compatible and fully support the new security feature, OEMs need to implement the Memory Address Table defined in UEFI 2.6, which was published in Jan. 2016. The adoption of the new UEFI standard takes time; meanwhile, to prevent customers encountering issues, we want to provide information about systems and configurations that we have tested this feature set with as well as systems that we know to be not compatible.

Non-compatible systems

The following configurations are known to be non-compatible with Virtualization-based protection of code integrity and cannot be used as a host for Shielded VMs:

- Dell PowerEdge Servers running PERC H330 RAID Controllers For more information, see the following article from Dell Support H330 – Enabling "Host Guardian Hyper-V Support" or "Device Guard" on Win 2016 OS causes OS boot failure.

Compatible systems

These are the systems we and our partners have been testing in our environment. Please make sure that you verify the system works as expected in your environment:

- Virtual Machines – You can enable Virtualization-based protection of code integrity on virtual machines that run on a Hyper-V host beginning with Windows Server 2016.

Feedback

Was this page helpful? Yes No

Guarded Fabric and Shielded VM Planning Guide for Tenants

Article • 05/19/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic focuses on VM owners who would like to protect their virtual machines (VMs) for compliance and security purposes. Regardless of whether the VMs run on a hosting provider's guarded fabric or a private guarded fabric, VM owners need to control the security level of their shielded VMs, which includes maintaining the ability to decrypt them if needed.

There are three areas to consider when using shielded VMs:

- The security level for the VMs
- The cryptographic keys used to protect them
- Shielding data—sensitive information used to create shielded VMs

Security level for the VMs

When deploying shielded VMs, one of two security levels must be selected:

- Shielded
- Encryption Supported

Both shielded and encryption-supported VMs have a virtual TPM attached to them and those that run Windows are protected by BitLocker. The primary difference is that shielded VMs block access by fabric administrators while encryption-supported VMs permit fabric administrators the same level of access as they would have to a regular VM. For more details about these differences, see Guarded fabric and shielded VMs overview.

Choose **Shielded VMs** if you are looking to protect the VM from a compromised fabric (including compromised administrators). They should be used in environments where fabric administrators and the fabric itself are not trusted. Choose **Encryption Supported VMs** if you are looking to meet a compliance bar that might require both encryption at-rest and encryption of the VM on the wire (e.g., during live migration).

Encryption-supported VMs are ideal in environments where fabric administrators are fully trusted but encryption remains a requirement.

You can run a mixture of regular VMs, shielded VMs, and encryption-supported VMs on a guarded fabric and even on the same Hyper-V host.

Whether a VM is shielded or encryption-supported is determined by the shielding data that is selected when creating the VM. VM owners configure the security level when creating the shielding data (see the Shielding data section). Note that once this choice has been made, it cannot be changed while the VM remains on the virtualization fabric.

Cryptographic keys used for shielded VMs

Shielded VMs are protected from virtualization fabric attack vectors using encrypted disks and various other encrypted elements which can only be decrypted by:

- An Owner key – this is a cryptographic key maintained by the VM-owner that is typically used for last-resort recovery or troubleshooting. VM owners are responsible for maintaining owner keys in a secure location.
- One or more Guardians (Host Guardian keys) – each Guardian represents a virtualization fabric on which an owner authorizes shielded VMs to run. Enterprises often have both a primary and a disaster recovery (DR) virtualization fabric and would typically authorize their shielded VMs to run on both. In some cases, the secondary (DR) fabric might be hosted by a public cloud provider. The private keys for any guarded fabric are maintained only on the virtualization fabric, while its public keys can be downloaded and are contained within its Guardian.

How do I create an owner key? An owner key is represented by two certificates. A certificate for encryption and a certificate for signing. You can create these two certificates using your own PKI infrastructure or obtain SSL certificates from a public certificate authority (CA). For test purposes, you can also create a self-signed certificate on any computer beginning with Windows 10 or Windows Server 2016.

How many owner keys should you have? You can use a single owner key or multiple owner keys. Best practices recommend a single owner key for a group of VMs that share the same security, trust or risk level, and for administrative control. You can share a single owner key for all your domain-joined shielded VMs and escrow that owner key to be managed by the domain administrators.

Can I use my own keys for the Host Guardian? Yes, you can "Bring Your Own" key to the hosting provider and use that key for your shielded VMs. This enables you to use your specific keys (vs. using the hosting provider key) and can be used when you have specific security or regulations that you need to abide by. For key hygiene purposes, the Host Guardian keys should be different than the Owner key.

Shielding data

Shielding data contains the secrets necessary to deploy shielded or encryption-supported VMs. It is also used when converting regular VMs to shielded VMs.

Shielding data is created using the Shielding Data File Wizard and is stored in PDK files which VM owners upload to the guarded fabric.

Shielded VMs help protect against attacks from a compromised virtualization fabric, so we need a safe mechanism to pass sensitive initialization data, such as the administrator's password, domain join credentials, or RDP certificates, without revealing these to the virtualization fabric itself or to its administrators. In addition, shielding data contains the following:

1. Security level – Shielded or encryption-supported
2. Owner and list of trusted Host Guardians where the VM can run
3. Virtual machine initialization data (unattend.xml, RDP certificate)
4. List of trusted signed template disks for creating the VM in the virtualization environment

When creating a shielded or encryption-supported VM or converting an existing VM, you will be asked to select the shielding data instead of being prompted for the sensitive information.

How many shielding data files do I need? A single shielding data file can be used to create every shielded VM. If, however, a given shielded VM requires that any of the four items be different, then an additional shielding data file is necessary. For example, you might have one shielding data file for your IT department and a different shielding data file for the HR department because their initial administrator password and RDP certificates differed.

While using separate shielding data files for each shielded VM is possible, it is not necessarily the optimal choice and should be done for the right reasons. For example, if every shielded VM needs to have a different administrator password, consider instead using a password management service or tool such as Microsoft's Local Administrator Password Solution (LAPS).

Creating a shielded VM on a virtualization fabric

There are several options for creating a shielded VM on a virtualization fabric (the following is relevant for both shielded and encryption-supported VMs):

1. Create a shielded VM in your environment and upload it to the virtualization fabric
2. Create a new shielded VM from a signed template on the virtualization fabric
3. Shield an existing VM (the existing VM must be generation 2 and must be running Windows Server 2012 or later)

Creating new VMs from a template is normal practice. However, since the template disk that is used to create new Shielded VM resides on the virtualization fabric, additional measures are necessary to ensure that it has not been tampered with by a malicious fabric administrator or by malware running on the fabric. This problem is solved using signed template disks—signed template disks and their disk signatures are created by trusted administrators or the VM owner. When a shielded VM is created, the template disk's signature is compared with the signatures contained within the specified shielding data file. If any of the shielding data file's signatures match the template disk's signature, the deployment process continues. If no match can be found, the deployment process is aborted, ensuring that VM secrets will not be compromised because of an untrustworthy template disk.

When using signed template disks to create shielded VMs, two options are available:

1. Use an existing signed template disk that is provided by your virtualization provider. In this case, the virtualization provider maintains signed template disks.
2. Upload a signed template disk to the virtualization fabric. The VM owner is responsible for maintaining signed template disks.

Feedback

Was this page helpful? Yes No

Deploying the Host Guardian Service

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

One of the most important goals of providing a hosted environment is to guarantee the security of the virtual machines running in the environment. As a cloud service provider or enterprise private cloud administrator, you can use a guarded fabric to provide a more secure environment for VMs. A guarded fabric consists of one Host Guardian Service (HGS) - typically, a cluster of three nodes - plus one or more guarded hosts, and a set of shielded virtual machines (VMs).

Video: Deploying a guarded fabric

https://learn-video.azurefd.net/vod/player?id=ecb8912d-d8fa-4b3e-8673-2f5be964d12a&locale=en-us&embedUrl=%2Fwindows-server%2Fsecurity%2Fguarded-fabric-shielded-vm%2Fguarded-fabric-deploying-hgs-overview

Deployment tasks for guarded fabrics and shielded VMs

The following table breaks down the tasks to deploy a guarded fabric and create shielded VMs according to different administrator roles. Note that when the HGS admin configures HGS with authorized Hyper-V hosts, a fabric admin will collect and provide identifying information about the hosts at the same time.

Expand table

Step and link to content	Image
1 - Verify HGS prerequisites	
2 - Configure first HGS node	
3 - Configure additional HGS nodes	
4 - Configure fabric DNS	

Step and link to content	Image
5 - Verify host prerequisites (Key) and Verify host prerequisites (TPM)	
6 - Create host key (Key) andCollect host information (TPM)	
7 - Configure HGS with host information	
8 - Confirm hosts can attest	
9 - Configure VMM (optional)	
10 - Create template disks	
11 - Create a VM shielding helper disk for VMM (optional)	
12 - Set up Windows Azure Pack (optional)	
13 - Create shielding data file	
14 - Create shielded VMs using Windows Azure Pack	
15 - Create shielded VMs using VMM	

Additional References

- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Quick start for guarded fabric deployment

Article • 05/12/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic explains what a guarded fabric is, its requirements, and a summary of the deployment process. For detailed deployment steps, see Deploying the Host Guardian Service for guarded hosts and shielded VMs.

Prefer video? See the Microsoft Virtual Academy course Deploying Shielded VMs and a Guarded Fabric with Windows Server 2016.

What is a guarded fabric

A *guarded fabric* is a Windows Server 2016 Hyper-V fabric capable of protecting tenant workloads against inspection, theft, and tampering from malware running on the host, as well as from system administrators. These virtualized tenant workloads—protected both at rest and in-flight—are called *shielded VMs*.

What are the requirements for a guarded fabric

The requirements for a guarded fabric include:

- **A place to run shielded VMs that is free from malicious software.**

  These are called *guarded hosts*. Guarded hosts are Windows Server 2016 Datacenter edition Hyper-V hosts that can run shielded VMs only if they can prove they are running in a known, trusted state to an external authority called the Host Guardian Service (HGS). The HGS is a new server role in Windows Server 2016, and is typically deployed as a three-node cluster.

- **A way to verify a host is in a healthy state.**

  The HGS performs *attestation*, where it measures the health of guarded hosts.

- **A process to securely release keys to healthy hosts.**

  The HGS performs *key protection and key release*, where it releases the keys back to healthy hosts.

- **Management tools to automate the secure provisioning and hosting of shielded VMs.**

  Optionally, you can add these management tools to a guarded fabric:

- System Center 2016 Virtual Machine Manager (VMM). VMM is recommended because it provides additional management tooling beyond what you get from using just the PowerShell cmdlets that come with Hyper-V and the guarded fabric workloads).
- System Center 2016 Service Provider Foundation (SPF). This is an API layer between Windows Azure Pack and VMM, and a prerequisite for using Windows Azure Pack.
- Windows Azure Pack provides a good graphical web interface to manage a guarded fabric and shielded VMs.

In practice, one decision must be made up front: the mode of attestation used by the guarded fabric. There are two means—two mutually exclusive modes—by which HGS can measure that a Hyper-V host is healthy. When you initialize HGS, you need to choose the mode:

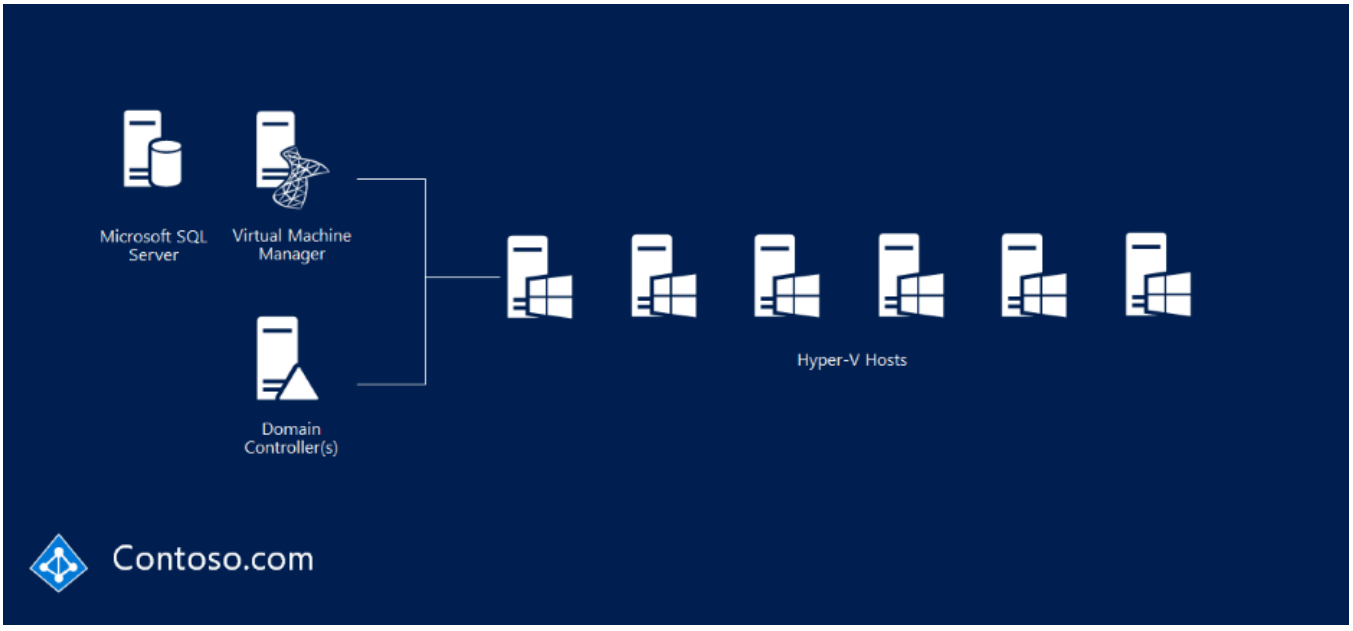
- Host key attestation, or key mode, is less secure but easier to adopt
- TPM-based attestation, or TPM mode, is more secure but requires more configuration and specific hardware

If necessary, you can deploy in key mode using existing Hyper-V hosts that have been upgraded to Windows Server 2019 Datacenter edition, and then convert to the more secure TPM mode when supporting server hardware (including TPM 2.0) is available.

Now that you know what the pieces are, let's walk through an example of the deployment model.

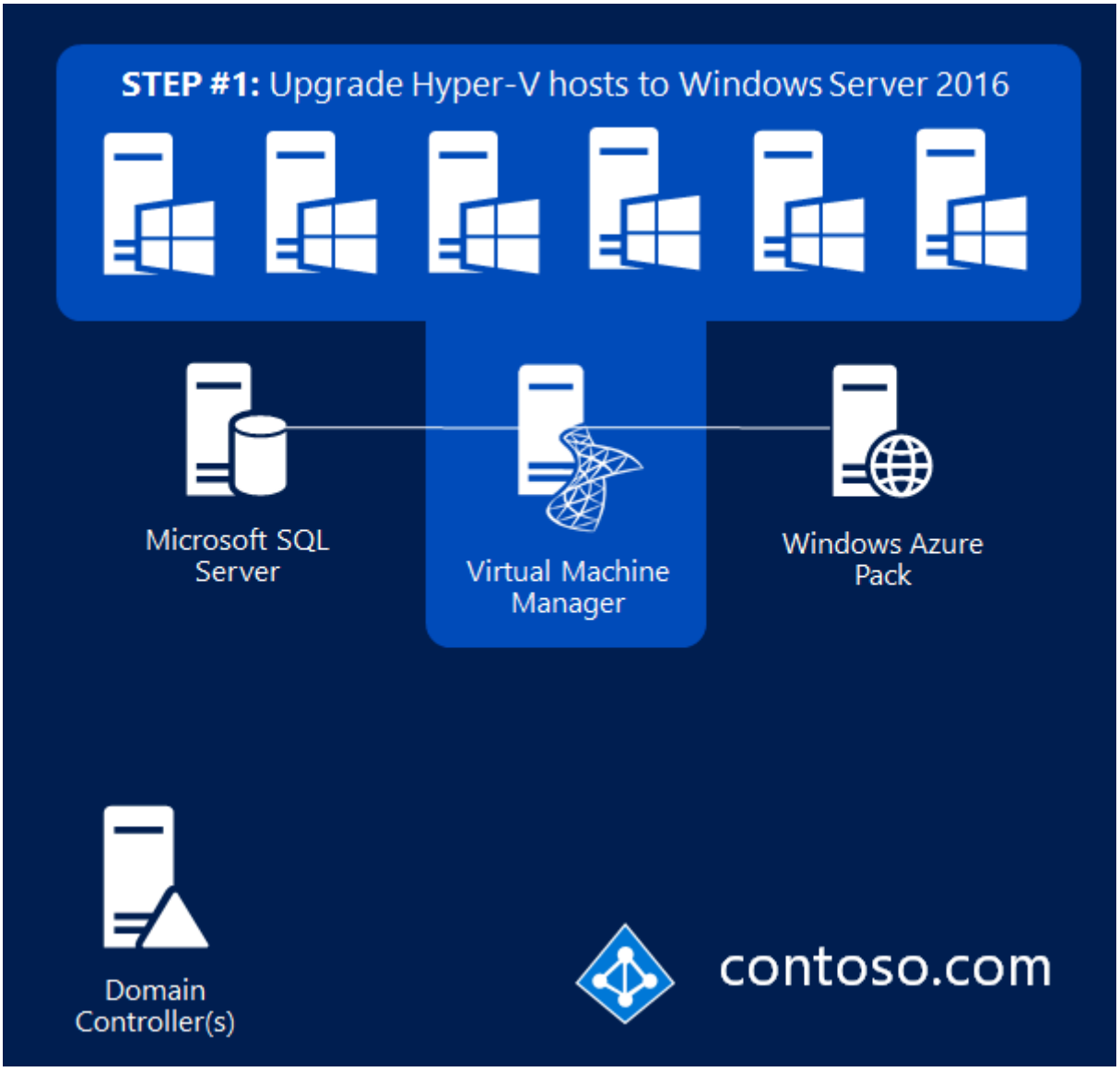
How to get from a current Hyper-V fabric to a guarded fabric

Let's imagine this scenario—you have an existing Hyper-V fabric, like Contoso.com in the following picture, and you want to build a Windows Server 2016 guarded fabric.



Step 1: Deploy the Hyper-V hosts running Windows Server 2016

The Hyper-V hosts need to run Windows Server 2016 Datacenter edition or later. If you are upgrading hosts, you can upgrade from Standard edition to Datacenter edition.



Step 2: Deploy the Host Guardian Service (HGS)

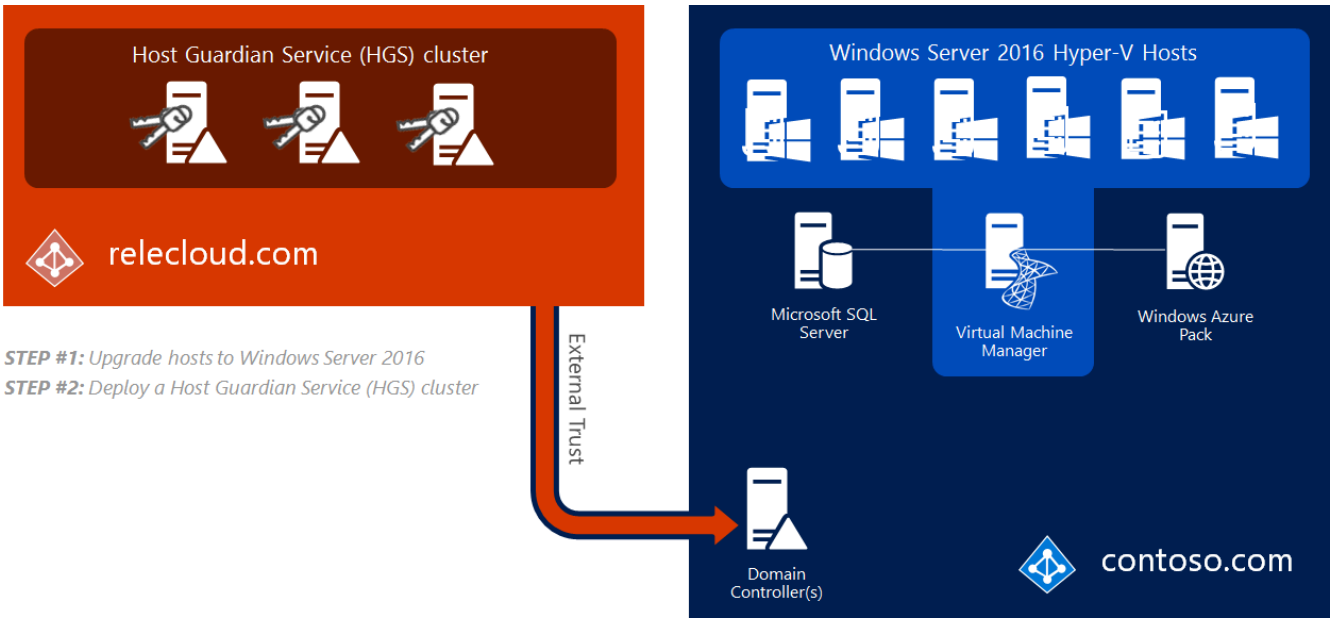
Then install the HGS server role and deploy it as a three-node cluster, like the relecloud.com example in the following picture. This requires three PowerShell cmdlets:

- To add the HGS role, use `Install-WindowsFeature`
- To install the HGS, use `Install-HgsServer`
- To initialize the HGS with your chosen mode of attestation, use `Initialize-HgsServer`

If your existing Hyper-V servers don't meet the prerequisites for TPM mode (for example, they do not have TPM 2.0), you can initialize HGS using Admin-based attestation (AD mode), which

requires an Active Directory trust with the fabric domain.

In our example, let's say Contoso initially deploys in AD mode in order to immediately meet compliance requirements, and plans to convert to the more secure TPM-based attestation after suitable server hardware can be purchased.



Step 3: Extract identities, hardware baselines, and code integrity policies

The process to extract identities from Hyper-V hosts depends on the attestation mode being used.

For AD mode, the ID of the host is its domain-joined computer account, which must be a member of a designated security group in the fabric domain. Membership in the designated group is the only determination of whether the host is healthy or not.

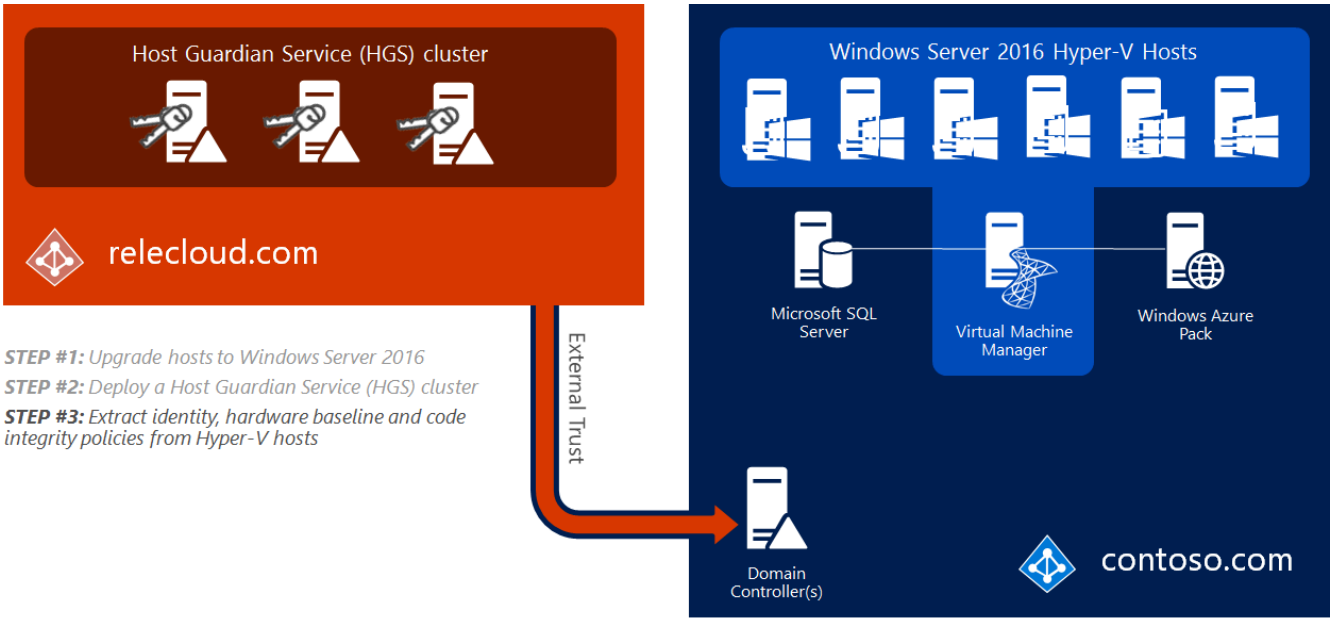
In this mode, the fabric admin is solely responsible for ensuring the health of the Hyper-V hosts. Since HGS plays no part in deciding what is or is not allowed to run, malware and debuggers will function as designed.

However, debuggers that attempt to attach directly to a process (such as WinDbg.exe) are blocked for shielded VMs because the VM's worker process (VMWP.exe) is a protected process light (PPL). Alternative debugging techniques, such as those used by LiveKd.exe, are not blocked. Unlike shielded VMs, the worker process for encryption supported VMs does not run as a PPL so traditional debuggers like WinDbg.exe will continue to function normally.

Stated another way, the rigorous validation steps used for TPM mode are not used for AD mode in any way.

For TPM mode, three things are required:

1. A *public endorsement key* (or *EKpub*) from the TPM 2.0 on each and every Hyper-V host. To capture the EKpub, use `Get-PlatformIdentifier`.
2. A *hardware baseline*. If each of your Hyper-V hosts are identical, then a single baseline is all you need. If they are not, then you'll need one for each class of hardware. The baseline is in the form of a Trustworthy Computing Group logfile, or TCGlog. The TCGlog contains everything that the host did, from the UEFI firmware, through the kernel, right up to where the host is entirely booted. To capture the hardware baseline, install the Hyper-V role and the Host Guardian Hyper-V Support feature and use `Get-HgsAttestationBaselinePolicy`.
3. A *Code Integrity policy*. If each of your Hyper-V hosts are identical, then a single CI policy is all you need. If they are not, then you'll need one for each class of hardware. Windows Server 2016 and Windows 10 both have a new form of enforcement for CI policies, called *Hypervisor-enforced Code Integrity (HVCI)*. HVCI provides strong enforcement and ensures that a host is only allowed to run binaries that a trusted admin has allowed it to run. Those instructions are wrapped in a CI policy that is added to HGS. HGS measures each host's CI policy before they're permitted to run shielded VMs. To capture a CI policy, use `New-CIPolicy`. The policy must then be converted to its binary form using `ConvertFrom-CIPolicy`.



That's all—the guarded fabric is built, in terms of the infrastructure to run it. Now you can create a shielded VM template disk and a shielding data file so shielded VMs can be provisioned simply and securely.

Step 4: Create a template for shielded VMs

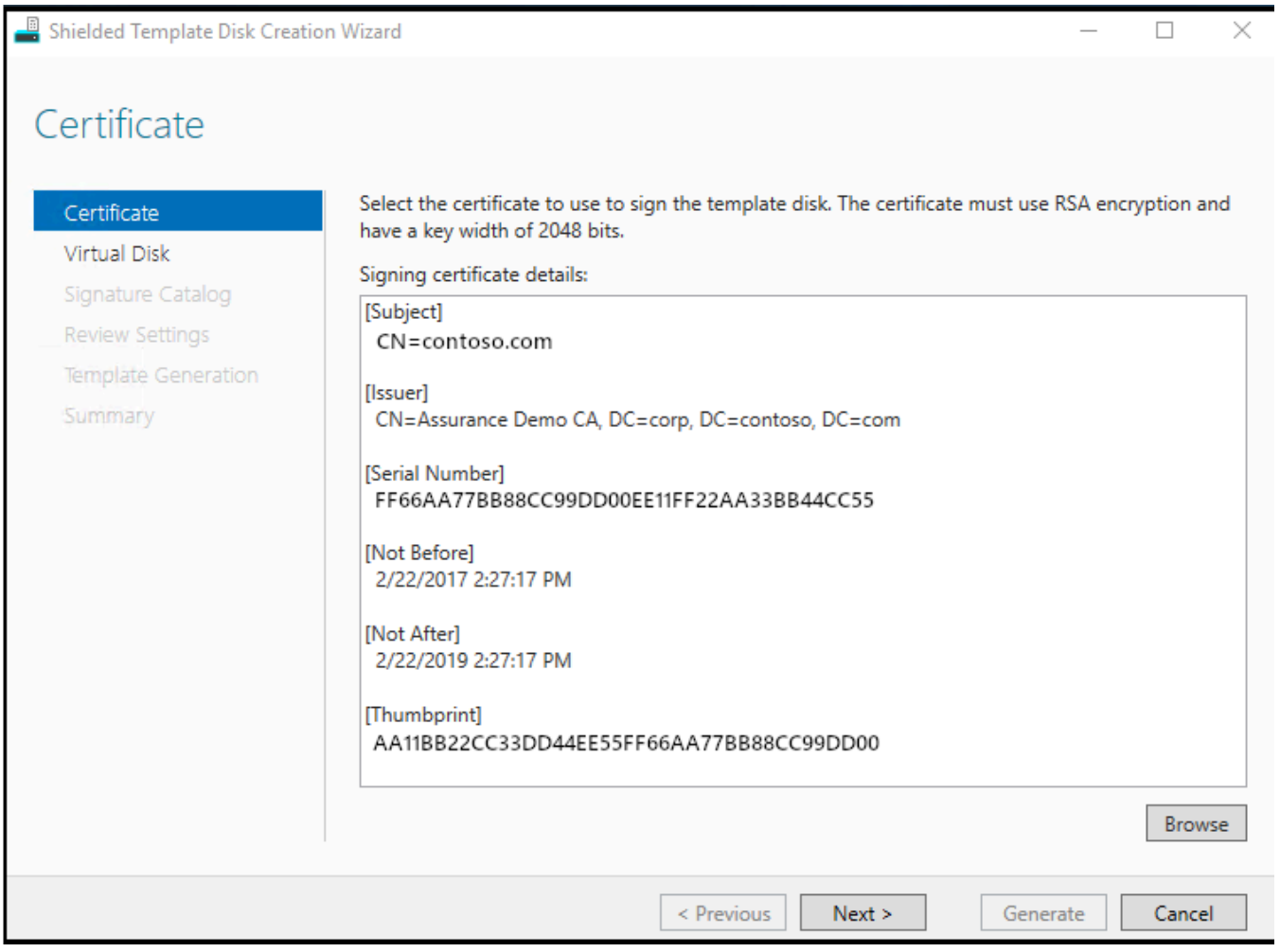
A shielded VM template protects template disks by creating a signature of the disk at a known trustworthy point in time. If the template disk is later infected by malware, its signature will differ original template which will be detected by the secure shielded VM provisioning process. Shielded template disks are created by running the **Shielded Template Disk Creation Wizard** or `Protect-TemplateDisk` against a regular template disk.

Each is included with the **Shielded VM Tools** feature in the Remote Server Administration Tools for Windows 10. After you download RSAT, run this command to install the **Shielded VM Tools** feature:

PowerShell

```
Install-WindowsFeature RSAT-Shielded-VM-Tools -Restart
```

A trustworthy administrator, such as the fabric administrator or the VM owner, will need a certificate (often provided by a Hosting Service Provider) to sign the VHDX template disk.

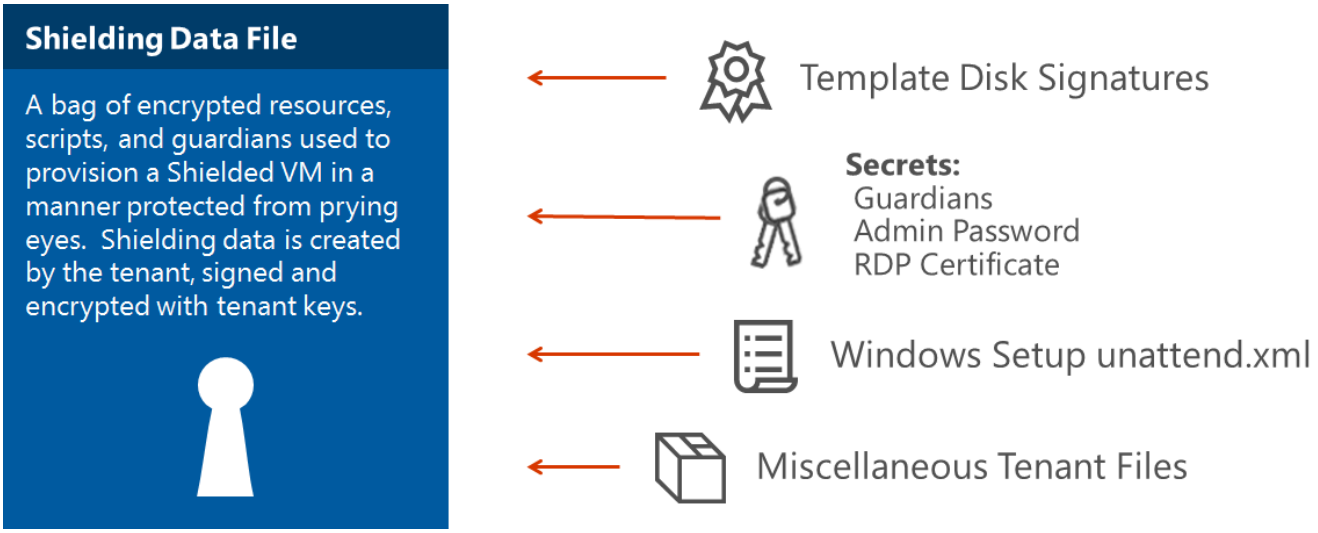


The disk signature is computed over the OS partition of the virtual disk. If anything changes on the OS partition, the signature will also change. This allows users to strongly identify which disks they trust by specifying the appropriate signature.

Review the template disk requirements before you get started.

Step 5: Create a shielding data file

A shielding data file, also known as a .pdk file, captures sensitive information about the virtual machine, such as the Administrator password.



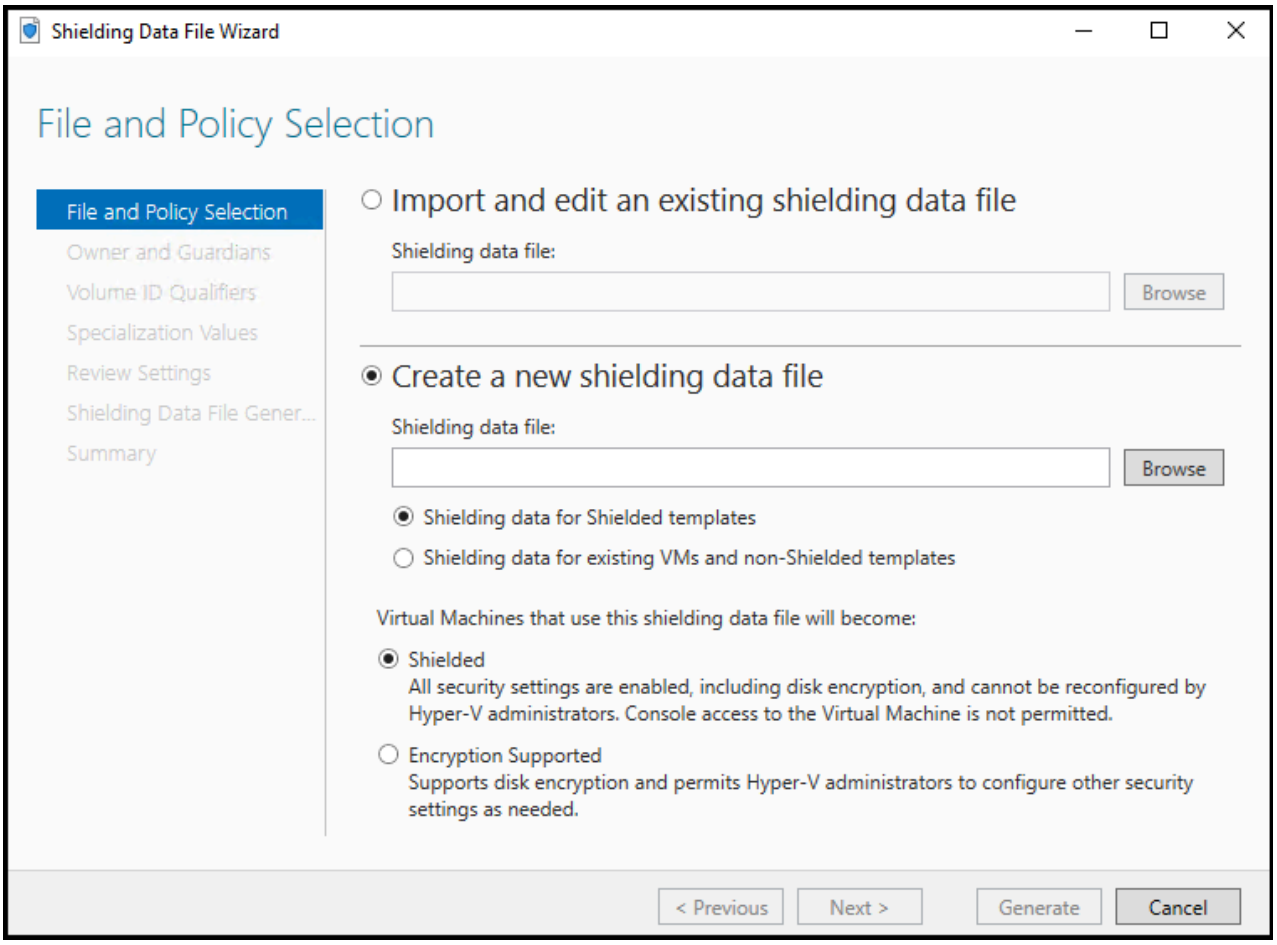
The shielding data file also includes the security policy setting for the shielded VM. You must choose one of two security policies when you create a shielding data file:

- Shielded

  The most secure option, which eliminates many administrative attack vectors.

- Encryption supported

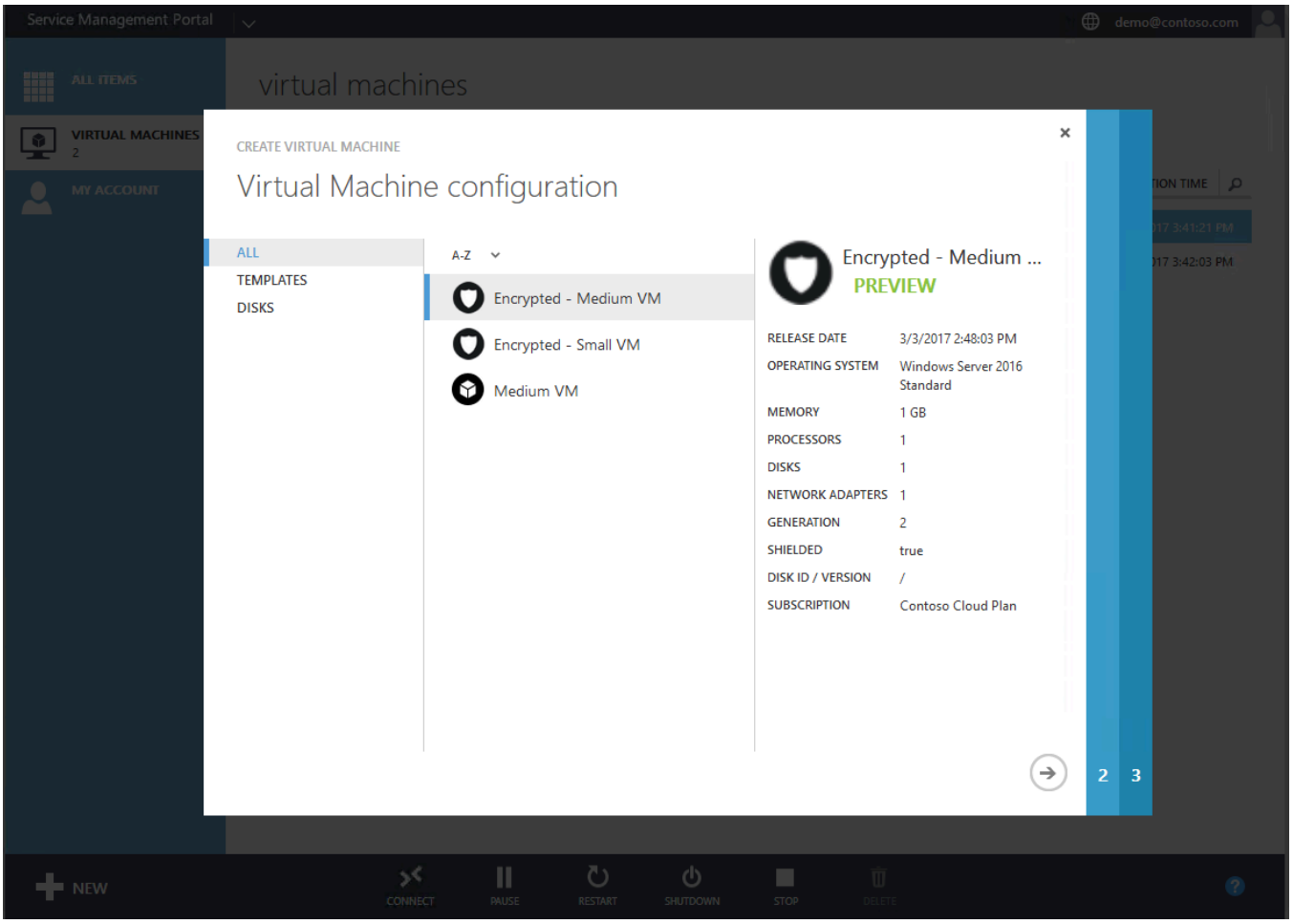
  A lesser level of protection that still provides the compliance benefits of being able to encrypt a VM, but allows Hyper-V admins to do things like use VM console connection and PowerShell Direct.



You can add optional management pieces like VMM or Windows Azure Pack. If you'd like to create a VM without installing those pieces, see Step by step – Creating Shielded VMs without VMM.

Step 6: Create a shielded VM

Creating shielded virtual machines differs very little from regular virtual machines. In Windows Azure Pack, the experience is even easier than creating a regular VM because you only need to supply a name, shielding data file (containing the rest of the specialization information), and the VM network.



Next step

HGS Prerequisites

Deploy the Host Guardian Service (HGS)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

To deploy the HGS, complete the following tasks:

- Prepare for the Host Guardian Service deployment
- Install HGS
- Initialize HGS
- Configure Https (optional)
- Add nodes

Additional References

- Deploying the Host Guardian Service for guarded hosts and shielded VMs
- Configuration steps for Hyper-V hosts that will become guarded hosts

---

Feedback

Was this page helpful? Yes No

Review prerequisites for the Host Guardian Service

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic covers HGS prerequisites and initial steps to prepare for the HGS deployment.

Prerequisites

- **Hardware**: HGS can be run on physical or virtual machines, but physical machines are recommended.

  If you want to run HGS as a three-node physical cluster (for availability), you must have three physical servers. (As a best practice for clustering, the three servers should have very similar hardware.)

- **Operating system**: Host key attestation requires Windows Server 2019 Standard or Datacenter edition operating with v2 attestation. For TPM-based attestation, HGS can run Windows Server 2019 or Windows Server 2016, Standard or Datacenter edition.

- **Server Roles**: Host Guardian Service and supporting server roles.

- **Configuration permissions/privileges for the fabric (host) domain**: You will need to configure DNS forwarding between the fabric (host) domain and the HGS domain.

Upgrading HGS

If you've already deployed HGS and want to upgrade its operating system, follow the upgrade guidance to upgrade your HGS and Hyper-V servers to the latest OS.

Next step

Obtain certificates for HGS

Feedback

Was this page helpful? Yes No

Obtain certificates for HGS

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

When you deploy HGS, you will be asked to provide signing and encryption certificates that are used to protect the sensitive information needed to start up a shielded VM. These certificates never leave HGS, and are only used to decrypt shielded VM keys when the host on which they're running has proven it is healthy. Tenants (VM owners) use the public half of the certificates to authorize your datacenter to run their shielded VMs. This section covers the steps required to obtain compatible signing and encryption certificates for HGS.

Request certificates from your certificate authority

While not required, it is strongly recommended that you obtain your certificates from a trusted certificate authority. Doing so helps VM owners verify that they are authorizing the correct HGS server (i.e. service provider or datacenter) to run their shielded VMs. In an enterprise scenario, you may choose to use your own enterprise CA to issue these certs. Hosters and service providers should consider using a well-known, public CA instead.

Both the signing and encryption certificates must be issued with the following certificiate properties (unless marked "recommended"):

Expand table

Certificate Template Property	Required value
Crypto provider	Any Key Storage Provider (KSP). Legacy Cryptographic Service Providers (CSPs) are **not** supported.
Key algorithm	RSA
Minimum key size	2048 bits
Signature algorithm	Recommended: SHA256
Key usage	Digital signature *and* data encipherment
Enhanced key usage	Server authentication

Certificate Template Property	Required value
Key renewal policy	Renew with the same key. Renewing HGS certificates with different keys will prevent shielded VMs from starting up.
Subject name	Recommended: your company's name or web address. This information will be shown to VM owners in the shielding data file wizard.

These requirements apply whether you are using certificates backed by hardware or software. For security reasons, it is recommended that you create your HGS keys in a Hardware Security Module (HSM) to prevent the private keys from being copied off the system. Follow the guidance from your HSM vendor to request certificates with the above attributes and be sure to install and authorize the HSM KSP on every HGS node.

Every HGS node will require access to the same signing and encryption certificates. If you are using software-backed certificates, you can export your certificates to a PFX file with a password and allow HGS to manage the certificates for you. You can also choose to install the certs into the local machine's certificate store on each HGS node and provide the thumbprint to HGS. Both options are explained in the Initialize the HGS Cluster topic.

Create self signed certificates for test scenarios

If you are creating an HGS lab environment and do not have or want to use a certificate authority, you can create self-signed certificates. You will receive a warning when importing the certificate information in the shielding data file wizard, but all functionality will remain the same.

To create self-signed certificates and export them to a PFX file, run the following commands in PowerShell:

PowerShell

```
$certificatePassword = Read-Host -AsSecureString -Prompt 'Enter a password
for the PFX file'

$signCert = New-SelfSignedCertificate -Subject 'CN=HGS Signing Certificate'
-KeyUsage DataEncipherment, DigitalSignature
Export-PfxCertificate -FilePath '.\signCert.pfx' -Password
$certificatePassword -Cert $signCert

# Remove the certificate from "Personal" container
Remove-Item $signCert.PSPath
# Remove the certificate from "Intermediate certification authorities"
container
```

```
Remove-Item -Path "Cert:\LocalMachine\CA\$($signCert.Thumbprint)"

$encCert = New-SelfSignedCertificate -Subject 'CN=HGS Encryption
Certificate' -KeyUsage DataEncipherment, DigitalSignature
Export-PfxCertificate -FilePath '.\encCert.pfx' -Password
$certificatePassword -Cert $encCert

# Remove the certificate from "Personal" container
Remove-Item $encCert.PSPath
# Remove the certificate from "Intermediate certification authorities"
container
Remove-Item -Path "Cert:\LocalMachine\CA\$($encCert.Thumbprint)"
```

Request an SSL certificate

All keys and sensitive information transmitted between Hyper-V hosts and HGS are encrypted at the message level -- that is, the information is encrypted with keys known either to HGS or Hyper-V, preventing someone from sniffing your network traffic and stealing keys to your VMs. However, if you have compliance requirements or simply prefer to encrypt all communications between Hyper-V and HGS, you can configure HGS with an SSL certificate which will encrypt all data at the transport level.

Both the Hyper-V hosts and HGS nodes will need to trust the SSL certificate you provide, so it is recommended that you request the SSL certificate from your enterprise certificate authority. When requesting the certificate, be sure to specify the following:

Expand table

SSL Certificate Property	Required value
Subject name	Address that HGS clients (that is, Guarded hosts) will be using to access the HGS server. This is typically the DNS address of your HGS cluster, known as the distributed network name or virtual computer object (VCO). This will be the concatenation of your HGS service name provided to `Initialize-HgsServer` and your HGS domain name.
Subject alternative name	If you will be using a different DNS name to reach your HGS cluster (e.g. if it is behind a load balancer, or you are using different addresses for a subset of nodes in complex topology), be sure to include those DNS names in the SAN field of your certificate request. Note that if SAN extension is populated, the Subject name is ignored, and hence SAN should include all values, including the one you would normally put in Subject name.

The options for specifying this certificate when initializing the HGS server are covered in Configure the first HGS node. You can also add or change the SSL certificate at a later time using the Set-HgsServer cmdlet.

Next step

Install HGS

Feedback

Was this page helpful? Yes No

Choose whether to install HGS in its own dedicated forest or in an existing bastion forest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Active Directory forest for HGS is sensitive because its administrators have access to the keys that control shielded VMs. The default installation will set up a new forest dedicated for HGS and configure other dependencies. This option is recommended because the environment is self-contained and known to be secure when it is created.

The only technical requirement for installing HGS in an existing forest is that it be added to the root domain; non-root domains are not supported. But there are also operational requirements and security-related best practices for using an existing forest. Suitable forests are purposely built to serve one sensitive function, such as the forest used by Privileged Access Management for AD DS or an Enhanced Security Administrative Environment (ESAE) forest. Such forests usually exhibit the following characteristics:

- They have few admins (separate from fabric admins)
- They have a low number of logons
- They are not general-purpose in nature

General purpose forests such as production forests are not suitable for use by HGS. Fabric forests are also unsuitable because HGS needs to be isolated from fabric administrators.

Next step

Choose the installation option that best suits your environment:

- Install HGS in its own dedicated forest
- Install HGS in an existing bastion forest

Feedback

Was this page helpful? Yes No

Install HGS in a new forest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Add the HGS server role

Run the following commands in an elevated PowerShell session to add the HGS server role and install HGS.

Add the Host Guardian Service role by running the following command:

PowerShell

```
Install-WindowsFeature -Name HostGuardianServiceRole -IncludeManagementTools
-Restart
```

Install HGS

The Host Guardian Service should be installed in a separate Active Directory forest. Ensure that the HGS machine is **not** joined to a domain before you start and sign in as the local machine Administrtor.

Run the following commands to install the Host Guardian Service and configure its domain. The password you specify here will only apply to the Directory Services Repair Mode password for Active Directory; it will *not* change your admin account's login password. You may provide any domain name of your choosing for -HgsDomainName.

PowerShell

```
$adminPassword = ConvertTo-SecureString -AsPlainText '<password>' -Force

Install-HgsServer -HgsDomainName 'bastion.local' -
SafeModeAdministratorPassword $adminPassword -Restart
```

Next steps

- For the next steps to set up TPM-based attestation, see Initialize the HGS cluster using TPM mode in a new dedicated forest (default).

- For the next steps to set up host key attestation, see Initialize the HGS cluster using key mode in a new dedicated forest (default).
- For the next steps to set up Admin-based attestation (deprecated in Windows Server 2019), see Initialize the HGS cluster using AD mode in a new dedicated forest (default).

Next step

Initialize HGS

Feedback

Was this page helpful? Yes No

Install HGS in an existing bastion forest

Article • 06/09/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Join the HGS server to the root domain

In an existing bastion forest, HGS must be added to the root domain. Use Server Manager or Add-Computer to join your HGS server to the root domain.

Add the HGS server role

Run all commands in this topic in an elevated PowerShell session.

Add the Host Guardian Service role by running the following command:

PowerShell

```
Install-WindowsFeature -Name HostGuardianServiceRole -IncludeManagementTools
-Restart
```

If your datacenter has a secure bastion forest where you want to join HGS nodes, follow these steps. You can also use these steps to configure 2 or more independent HGS clusters that are joined to the same domain.

Join the HGS server to the desired domain

Use Server Manager or Add-Computer to join the HGS servers to the desired domain.

Prepare Active Directory objects

Create a group managed service account and 2 security groups. You can also pre-stage the cluster objects if the account you are initializing HGS with does not have permission to create computer objects in the domain.

Group managed service account

The group managed service account (gMSA) is the identity used by HGS to retrieve and use its certificates. Use New-ADServiceAccount to create a gMSA. If this is the first gMSA in the domain, you will need to add a Key Distribution Service root key.

Each HGS node will need to be permitted to access the gMSA password. The easiest way to configure this is to create a security group that contains all of your HGS nodes and grant that security group access to retrieve the gMSA password.

You must reboot your HGS server after adding it to a security group to ensure it obtains its new group membership.

PowerShell

```
# Check if the KDS root key has been set up
if (-not (Get-KdsRootKey)) {
    # Adds a KDS root key effective immediately (ignores normal 10 hour
waiting period)
    Add-KdsRootKey -EffectiveTime ((Get-Date).AddHours(-10))
}

# Create a security group for HGS nodes
$hgsNodes = New-ADGroup -Name 'HgsServers' -GroupScope DomainLocal -PassThru

# Add your HGS nodes to this group
# If your HGS server object is under an organizational unit, provide the
full distinguished name instead of "HGS01"
Add-ADGroupMember -Identity $hgsNodes -Members "HGS01"

# Create the gMSA
New-ADServiceAccount -Name 'HGSgMSA' -DnsHostName 'HGSgMSA.yourdomain.com' -
PrincipalsAllowedToRetrieveManagedPassword $hgsNodes
```

The gMSA will require the right to generate events in the security log on each HGS server. If you use Group Policy to configure User Rights Assignment, ensure that the gMSA account is granted the generate audit events privilege on your HGS servers.

> **Note**
>
> Group managed service accounts are available beginning with the Windows Server 2012 Active Directory schema. For more information, see group managed service account requirements.

JEA security groups

When you set up HGS, a Just Enough Administration (JEA) PowerShell endpoint is configured to allow admins to manage HGS without needing full local administrator privileges. You are not required to use JEA to manage HGS, but it still must be configured when running Initialize-HgsServer. The configuration of the JEA endpoint consists of designating 2 security groups that contain your HGS admins and HGS reviewers. Users who belong to the admin group can add, change, or remove policies on HGS; reviewers can only view the current configuration.

Create 2 security groups for these JEA groups using Active Directory admin tools or New-ADGroup.

PowerShell

```
New-ADGroup -Name 'HgsJeaReviewers' -GroupScope DomainLocal
New-ADGroup -Name 'HgsJeaAdmins' -GroupScope DomainLocal
```

Cluster objects

If the account you are using to set up HGS does not have permission to create new computer objects in the domain, you will need to pre-stage the cluster objects. These steps are explained in Prestage Cluster Computer Objects in Active Directory Domain Services.

To set up your first HGS node, you will need to create one Cluster Name Object (CNO) and one Virtual Computer Object (VCO). The CNO represents the name of the cluster, and is primarily used internally by Failover Clustering. The VCO represents the HGS service that resides on top of the cluster and will be the name registered with the DNS server.

> **ⓘ Important**
>
> The user who will run `Initialize-HgsServer` requires **Full Control** over the CNO and VCO objects in Active Directory.

To quickly prestage your CNO and VCO, have an Active Directory admin run the following PowerShell commands:

PowerShell

```
# Create the CNO
$cno = New-ADComputer -Name 'HgsCluster' -Description 'HGS CNO' -Enabled
$false -Passthru
```

```
# Create the VCO
$vco = New-ADComputer -Name 'HgsService' -Description 'HGS VCO' -Passthru

# Give the CNO full control over the VCO
$vcoPath = Join-Path "AD:\" $vco.DistinguishedName
$acl = Get-Acl $vcoPath
$ace = New-Object System.DirectoryServices.ActiveDirectoryAccessRule
$cno.SID, "GenericAll", "Allow"
$acl.AddAccessRule($ace)
Set-Acl -Path $vcoPath -AclObject $acl

# Allow time for your new CNO and VCO to replicate to your other Domain
Controllers before continuing
```

Security baseline exceptions

If you are deploying HGS into a highly locked down environment, certain Group Policy settings may prevent HGS from operating normally. Check your Group Policy objects for the following settings and follow the guidance if you are affected:

Network Logon

Policy Path: Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignments

Policy Name: Deny access to this computer from the network

Required value: Ensure the value does not block network logons for all local accounts. You can safely block local administrator accounts, however.

Reason: Failover Clustering relies on a non-administrator local account called CLIUSR to manage cluster nodes. Blocking network logon for this user will prevent the cluster from operating correctly.

Kerberos Encryption

Policy Path: Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options

Policy Name: Network Security: Configure encryption types allowed for Kerberos

Action: If this policy is configured, you must update the gMSA account with Set-ADServiceAccount to use only the supported encryption types in this policy. For instance, if your policy only allows AES128_HMAC_SHA1 and AES256_HMAC_SHA1, you

should run `Set-ADServiceAccount -Identity HGSgMSA -KerberosEncryptionType AES128,AES256`.

Next steps

- For the next steps to set up TPM-based attestation, see Initialize the HGS cluster using TPM mode in an existing bastion forest.
- For the next steps to set up host key attestation, see Initialize the HGS cluster using key mode in an existing bastion forest.
- For the next steps to set up Admin-based attestation (deprecated in Windows Server 2019), see Initialize the HGS cluster using AD mode in an existing bastion forest.

Feedback

Was this page helpful? Yes No

Initialize the Host Guardian Service (HGS)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

When you initialize HGS, you specify the mode that HGS will use to measure the health of guarded hosts. There are two mutually exclusive options. For background information about which mode to choose, see Guarded Fabric and Shielded VM Planning Guide for Hosters.

The following topics cover deployment steps for each mode:

- TPM-trusted attestation (TPM mode)
- Host key attestation (Key mode)
- Admin-trusted attestation (AD mode)

You should perform these steps on a physical server.

Feedback

Was this page helpful? Yes No

Initialize HGS using TPM-trusted attestation

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

These steps vary depending on whether you are initializing HGS in a new forest or an existing bastion forest:

1. Initialize the HGS cluster in a new forest (default)

   -Or-

   Initialize the HGS cluster in an existing bastion forest

2. Install trusted TPM root certificates

3. Configure the fabric DNS

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using TPM mode in a new dedicated forest (default)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

1. Clients can easily contact any HGS node by using the failover clustering distributed network name (DNN). You'll need to choose a DNN. This name will be registered in the HGS DNS service. As an example, if you have 3 HGS nodes with hostnames HGS01, HGS02, and HGS03, you might decide to choose "hgs" or "HgsCluster" for the DNN.

2. Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

3. Run Initialize-HgsServer in an elevated PowerShell window on the first HGS node. The syntax of this cmdlet supports many different inputs, but the 2 most common invocations are below:

    - If you are using PFX files for your signing and encryption certificates, run the following commands:

    PowerShell

    ```
    $signingCertPass = Read-Host -AsSecureString -Prompt "Signing
    certificate password"
    $encryptionCertPass = Read-Host -AsSecureString -Prompt
    "Encryption certificate password"

    Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
    SigningCertificatePath '.\signCert.pfx' -
    SigningCertificatePassword $signingCertPass -
    EncryptionCertificatePath '.\encCert.pfx' -
    EncryptionCertificatePassword $encryptionCertPass -TrustTpm
    ```

- If you are using non-exportable certificates that are installed in the local certificate store, run the following command. If you do not know the thumbprints of your certificates, you can list available certificates by running `Get-ChildItem Cert:\LocalMachine\My`.

```
PowerShell

Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
SigningCertificateThumbprint '1A2B3C4D5E6F...' -
EncryptionCertificateThumbprint '0F9E8D7C6B5A...' -TrustTpm
```

4. If you provided any certificates to HGS using thumbprints, you will be instructed to grant HGS read access to the private key of those certificates. On a server with Desktop Experience installed, complete the following steps:
   a. Open the local computer certificate manager (**certlm.msc**)
   b. Find the certificate(s) > right-click > all tasks > manage private keys
   c. Click **Add**
   d. In the object picker window, click **Object types** and enable **service accounts**
   e. Enter the name of the service account mentioned in the warning text from `Initialize-HgsServer`
   f. Ensure the gMSA has "Read" access to the private key.

   On server core, you will need to download a PowerShell module to assist in setting the private key permissions.

   a. Run `Install-Module GuardedFabricTools` on the HGS server if it has Internet connectivity, or run `Save-Module GuardedFabricTools` on another computer and copy the module over to the HGS server.

   b. Run `Import-Module GuardedFabricTools`. This will add additional properties to certificate objects found in PowerShell.

   c. Find your certificate thumbprint in PowerShell with `Get-ChildItem Cert:\LocalMachine\My`

   d. Update the ACL, replacing the thumbprint with your own and the gMSA account in the code below with the account listed in the warning text of `Initialize-HgsServer`.

```
PowerShell

$certificate = Get-Item "Cert:\LocalMachine\1A2B3C..."
$certificate.Acl = $certificate.Acl | Add-AccessRule "HgsSvc_1A2B3C"
```

```
Read Allow
```

If you are using HSM-backed certificates, or certificates stored in a third party key storage provider, these steps may not apply to you. Consult your key storage provider's documentation to learn how to manage permissions on your private key. In some cases, there is no authorization, or authorization is provided to the entire computer when the certificate is installed.

5. That's it! In a production environment, you should continue to add additional HGS nodes to your cluster.

Next step

Install TPM root certs

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using TPM mode in an existing bastion forest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

To initialize the HGS cluster using TPM mode in an existing bastion forest, follow the steps below. Active Directory Domain Services will be installed on the machine, but should remain unconfigured.

Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

Before you continue, ensure that you have prestaged your cluster objects for the Host Guardian Service and granted the logged in user **Full Control** over the VCO and CNO objects in Active Directory. The virtual computer object name needs to be passed to the `-HgsServiceName` parameter, and the cluster name to the `-ClusterName` parameter.

> **Tip**
>
> Double check your AD Domain Controllers to ensure your cluster objects have replicated to all DCs before continuing.

If you are using PFX-based certificates, run the following commands on the HGS server:

PowerShell

```
$signingCertPass = Read-Host -AsSecureString -Prompt "Signing certificate
password"
$encryptionCertPass = Read-Host -AsSecureString -Prompt "Encryption
certificate password"

Install-ADServiceAccount -Identity 'HGSgMSA'

Initialize-HgsServer -UseExistingDomain -ServiceAccount 'HGSgMSA' -
JeaReviewersGroup 'HgsJeaReviewers' -JeaAdministratorsGroup 'HgsJeaAdmins' -
HgsServiceName 'HgsService' -SigningCertificatePath '.\signCert.pfx' -
```

```
SigningCertificatePassword $signPass -EncryptionCertificatePath
'.\encCert.pfx' -EncryptionCertificatePassword $encryptionCertPass -TrustTpm
```

If you are using certificates installed on the local machine (such as HSM-backed certificates and non-exportable certificates), use the `-SigningCertificateThumbprint` and `-EncryptionCertificateThumbprint` parameters instead.

In a production environment, you should continue to add additional HGS nodes to your cluster.

Next step

Install TPM root certs

Feedback

Was this page helpful? Yes No

Install trusted TPM root certificates

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

When you configure HGS to use TPM attestation, you also need to configure HGS to trust the vendors of the TPMs in your servers. This extra verification process ensures only authentic, trustworthy TPMs are able to attest with your HGS. If you try to register an untrusted TPM with `Add-HgsAttestationTpmHost`, you will receive an error indicating the TPM vendor is untrusted.

To trust your TPMs, the root and intermediate signing certificates used to sign the endorsement key in your servers' TPMs need to be installed on HGS. If you use more than one TPM model in your datacenter, you may need to install different certificates for each model. HGS will look in the "TrustedTPM_RootCA" and "TrustedTPM_IntermediateCA" certificate stores for the vendor certificates.

> **Note**
>
> The TPM vendor certificates are different from those installed by default in Windows and represent the specific root and intermediate certificates used by TPM vendors.

A collection of trusted TPM root and intermediate certificates is published by Microsoft for your convenience. You can use the steps below to install these certificates. If your TPM certificates are not included in the package below, contact your TPM vendor or server OEM to obtain the root and intermediate certificates for your specific TPM model.

Repeat the following steps on **every HGS server**:

1. Download the latest package from https://go.microsoft.com/fwlink/?linkid=2097925.

2. Verify the signature of the cab file to ensure its authenticity. Do not proceed if the signature is not valid.

   PowerShell

   ```
   Get-AuthenticodeSignature .\TrustedTpm.cab
   ```

   Here's some example output:

```
Directory: C:\Users\Administrator\Downloads

SignerCertificate                         Status
Path
-----------------                         ------
----
0DD6D4D4F46C0C7C2671962C4D361D607E370940  Valid
TrustedTpm.cab
```

3. Expand the cab file.

```
mkdir .\TrustedTPM
expand.exe -F:* <Path-To-TrustedTpm.cab> .\TrustedTPM
```

4. By default, the configuration script will install certificates for every TPM vendor. If you only want to import certificates for your specific TPM vendor, delete the folders for TPM vendors not trusted by your organization.

5. Install the trusted certificate package by running the setup script in the expanded folder.

```
cd .\TrustedTPM
.\setup.cmd
```

To add new certificates or ones intentionally skipped during an earlier installation, simply repeat the above steps on every node in your HGS cluster. Existing certificates will remain trusted but new certificates found in the expanded cab file will be added to the trusted TPM stores.

Next step

Configure fabric DNS

Feedback

Was this page helpful? Yes No

Next step

05/01/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

There are many ways to configure name resolution for the fabric domain. One simple way is to set up a conditional forwarder zone in DNS for the fabric. To set up this zone, run the following commands in an elevated Windows PowerShell console on a fabric DNS server. Substitute the names and addresses in the Windows PowerShell syntax below as needed for your environment. Add the parameter `-MasterServers` for the additional HGS nodes.

```
Add-DnsServerConditionalForwarderZone -Name 'bastion.local' -ReplicationScope
"Forest" -MasterServers <IP addresses of HGS server>
```

Configure HTTPS

Additional References

- Configuration steps for Hyper-V hosts that will become guarded hosts
- Deployment tasks for guarded fabrics and shielded VMs

Initialize HGS using host key attestation

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This step depends on whether you are initializing HGS in a new forest or an existing bastion forest:

- Initialize the HGS cluster in a new forest (default)

  -Or-

- Initialize the HGS cluster in an existing bastion forest

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using key mode in a new dedicated forest (default)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

1. Clients can easily contact any HGS node by using the failover clustering distributed network name (DNN). You'll need to choose a DNN. This name will be registered in the HGS DNS service. As an example, if you have 3 HGS nodes with hostnames HGS01, HGS02, and HGS03, you might decide to choose "hgs" or "HgsCluster" for the DNN.

2. Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

3. Run Initialize-HgsServer in an elevated PowerShell window on the first HGS node. The syntax of this cmdlet supports many different inputs, but the 2 most common invocations are below:

    - If you are using PFX files for your signing and encryption certificates, run the following commands:

    PowerShell

    ```
    $signingCertPass = Read-Host -AsSecureString -Prompt "Signing
    certificate password"
    $encryptionCertPass = Read-Host -AsSecureString -Prompt
    "Encryption certificate password"

    Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
    SigningCertificatePath '.\signCert.pfx' -
    SigningCertificatePassword $signingCertPass -
    EncryptionCertificatePath '.\encCert.pfx' -
    EncryptionCertificatePassword $encryptionCertPass -TrustHostkey
    ```

- If you are using non-exportable certificates that are installed in the local certificate store, run the following command. If you do not know the thumbprints of your certificates, you can list available certificates by running `Get-ChildItem Cert:\LocalMachine\My`.

```
PowerShell

Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
SigningCertificateThumbprint '1A2B3C4D5E6F...' -
EncryptionCertificateThumbprint '0F9E8D7C6B5A...' --TrustHostKey
```

4. If you provided any certificates to HGS using thumbprints, you will be instructed to grant HGS read access to the private key of those certificates. On a server with Desktop Experience installed, complete the following steps:
   a. Open the local computer certificate manager (**certlm.msc**)
   b. Find the certificate(s) > right-click > all tasks > manage private keys
   c. Click **Add**
   d. In the object picker window, click **Object types** and enable **service accounts**
   e. Enter the name of the service account mentioned in the warning text from `Initialize-HgsServer`
   f. Ensure the gMSA has "Read" access to the private key.

   On server core, you will need to download a PowerShell module to assist in setting the private key permissions.

   a. Run `Install-Module GuardedFabricTools` on the HGS server if it has Internet connectivity, or run `Save-Module GuardedFabricTools` on another computer and copy the module over to the HGS server.

   b. Run `Import-Module GuardedFabricTools`. This will add additional properties to certificate objects found in PowerShell.

   c. Find your certificate thumbprint in PowerShell with `Get-ChildItem Cert:\LocalMachine\My`

   d. Update the ACL, replacing the thumbprint with your own and the gMSA account in the code below with the account listed in the warning text of `Initialize-HgsServer`.

```
PowerShell

$certificate = Get-Item "Cert:\LocalMachine\1A2B3C..."
$certificate.Acl = $certificate.Acl | Add-AccessRule "HgsSvc_1A2B3C"
```

```
Read Allow
```

If you are using HSM-backed certificates, or certificates stored in a third party key storage provider, these steps may not apply to you. Consult your key storage provider's documentation to learn how to manage permissions on your private key. In some cases, there is no authorization, or authorization is provided to the entire computer when the certificate is installed.

5. That's it! In a production environment, you should continue to add additional HGS nodes to your cluster.

Next step

Create host key

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using key mode in an existing bastion forest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

« Install HGS in a new forest | Create host key »

Active Directory Domain Services will be installed on the machine, but should remain unconfigured.

Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

Before you continue, ensure that you have prestaged your cluster objects for the Host Guardian Service and granted the logged in user **Full Control** over the VCO and CNO objects in Active Directory. The virtual computer object name needs to be passed to the `-HgsServiceName` parameter, and the cluster name to the `-ClusterName` parameter.

> **Tip**
>
> Double check your AD Domain Controllers to ensure your cluster objects have replicated to all DCs before continuing.

If you are using PFX-based certificates, run the following commands on the HGS server:

PowerShell

```
$signingCertPass = Read-Host -AsSecureString -Prompt "Signing certificate
password"
$encryptionCertPass = Read-Host -AsSecureString -Prompt "Encryption
certificate password"

Install-ADServiceAccount -Identity 'HGSgMSA'

Initialize-HgsServer -UseExistingDomain -ServiceAccount 'HGSgMSA' -
```

```
JeaReviewersGroup 'HgsJeaReviewers' -JeaAdministratorsGroup 'HgsJeaAdmins' -
HgsServiceName 'HgsService' -ClusterName 'HgsCluster' -
SigningCertificatePath '.\signCert.pfx' -SigningCertificatePassword
$signPass -EncryptionCertificatePath '.\encCert.pfx' -
EncryptionCertificatePassword $encryptionCertPass -TrustHostKey
```

If you are using certificates installed on the local machine (such as HSM-backed certificates and non-exportable certificates), use the `-SigningCertificateThumbprint` and `-EncryptionCertificateThumbprint` parameters instead.

Feedback

Was this page helpful? Yes No

Initialize HGS using Admin-trusted attestation

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Important**
>
> Admin-trusted attestation (AD mode) is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

These steps vary depending on whether you are initializing HGS in a new forest or an existing bastion forest:

1. Initialize the HGS cluster in a new forest (default)

   -Or-

   Initialize the HGS cluster in an existing bastion forest

2. Configure DNS forwarding in the fabric domain

3. Configure DNS forwarding and a one-way trust in the HGS domain

---

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using AD mode in a new dedicated forest (default)

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **ⓘ Important**
>
> Admin-trusted attestation (AD mode) is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

1. Clients can easily contact any HGS node by using the failover clustering distributed network name (DNN). You'll need to choose a DNN. This name will be registered in the HGS DNS service. As an example, if you have 3 HGS nodes with hostnames HGS01, HGS02, and HGS03, you might decide to choose "hgs" or "HgsCluster" for the DNN.

2. Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

3. Run Initialize-HgsServer in an elevated PowerShell window on the first HGS node. The syntax of this cmdlet supports many different inputs, but the 2 most common invocations are below:

    - If you are using PFX files for your signing and encryption certificates, run the following commands:

```
PowerShell

$signingCertPass = Read-Host -AsSecureString -Prompt "Signing
certificate password"
$encryptionCertPass = Read-Host -AsSecureString -Prompt
"Encryption certificate password"
```

```
Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
SigningCertificatePath '.\signCert.pfx' -
SigningCertificatePassword $signingCertPass -
EncryptionCertificatePath '.\encCert.pfx' -
EncryptionCertificatePassword $encryptionCertPass -
TrustActiveDirectory
```

- If you are using non-exportable certificates that are installed in the local certificate store, run the following command. If you do not know the thumbprints of your certificates, you can list available certificates by running `Get-ChildItem Cert:\LocalMachine\My`.

PowerShell

```
Initialize-HgsServer -HgsServiceName 'MyHgsDNN' -
SigningCertificateThumbprint '1A2B3C4D5E6F...' -
EncryptionCertificateThumbprint '0F9E8D7C6B5A...' --
TrustActiveDirectory
```

4. If you provided any certificates to HGS using thumbprints, you will be instructed to grant HGS read access to the private key of those certificates. On a server with Desktop Experience installed, complete the following steps:
   a. Open the local computer certificate manager (**certlm.msc**)
   b. Find the certificate(s) > right-click > all tasks > manage private keys
   c. Click **Add**
   d. In the object picker window, click **Object types** and enable **service accounts**
   e. Enter the name of the service account mentioned in the warning text from `Initialize-HgsServer`
   f. Ensure the gMSA has "Read" access to the private key.

   On server core, you will need to download a PowerShell module to assist in setting the private key permissions.

   a. Run `Install-Module GuardedFabricTools` on the HGS server if it has Internet connectivity, or run `Save-Module GuardedFabricTools` on another computer and copy the module over to the HGS server.

   b. Run `Import-Module GuardedFabricTools`. This will add additional properties to certificate objects found in PowerShell.

   c. Find your certificate thumbprint in PowerShell with `Get-ChildItem Cert:\LocalMachine\My`

d. Update the ACL, replacing the thumbprint with your own and the gMSA account in the code below with the account listed in the warning text of `Initialize-HgsServer`.

```PowerShell
$certificate = Get-Item "Cert:\LocalMachine\1A2B3C..."
$certificate.Acl = $certificate.Acl | Add-AccessRule "HgsSvc_1A2B3C"
Read Allow
```

If you are using HSM-backed certificates, or certificates stored in a third party key storage provider, these steps may not apply to you. Consult your key storage provider's documentation to learn how to manage permissions on your private key. In some cases, there is no authorization, or authorization is provided to the entire computer when the certificate is installed.

5. That's it! In a production environment, you should continue to add additional HGS nodes to your cluster.

Next step

Configure fabric DNS

Feedback

Was this page helpful? Yes No

Initialize the HGS cluster using AD mode in an existing bastion forest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Important**
>
> Admin-trusted attestation (AD mode) is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

Active Directory Domain Services will be installed on the machine, but should remain unconfigured.

Locate your HGS guardian certificates. You will need one signing certificate and one encryption certificate to initialize the HGS cluster. The easiest way to provide certificates to HGS is to create a password-protected PFX file for each certificate which contains both the public and private keys. If you are using HSM-backed keys or other non-exportable certificates, make sure the certificate is installed into the local machine's certificate store before continuing. For more information about which certificates to use, see Obtain certificates for HGS.

Before you continue, ensure that you have prestaged your cluster objects for the Host Guardian Service and granted the logged in user **Full Control** over the VCO and CNO objects in Active Directory. The virtual computer object name needs to be passed to the `-HgsServiceName` parameter, and the cluster name to the `-ClusterName` parameter.

> **Tip**
>
> Double check your AD Domain Controllers to ensure your cluster objects have replicated to all DCs before continuing.

If you are using PFX-based certificates, run the following commands on the HGS server:

PowerShell

```
$signingCertPass = Read-Host -AsSecureString -Prompt "Signing certificate password"
```

```
$encryptionCertPass = Read-Host -AsSecureString -Prompt "Encryption
certificate password"

Install-ADServiceAccount -Identity 'HGSgMSA'

Initialize-HgsServer -UseExistingDomain -ServiceAccount 'HGSgMSA' -
JeaReviewersGroup 'HgsJeaReviewers' -JeaAdministratorsGroup 'HgsJeaAdmins' -
HgsServiceName 'HgsService' -ClusterName 'HgsCluster' -
SigningCertificatePath '.\signCert.pfx' -SigningCertificatePassword
$signPass -EncryptionCertificatePath '.\encCert.pfx' -
EncryptionCertificatePassword $encryptionCertPass -TrustActiveDirectory
```

If you are using certificates installed on the local machine (such as HSM-backed certificates and non-exportable certificates), use the `-SigningCertificateThumbprint` and `-EncryptionCertificateThumbprint` parameters instead.

Next step

Configure fabric DNS

Feedback

Was this page helpful? Yes No

Configure the fabric DNS for guarded hosts (AD)

05/01/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **ⓘ Important**
>
> AD mode is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

A fabric administrator needs to configure the fabric DNS takes to allow guarded hosts to resolve the HGS cluster. The HGS cluster must already be set up by the HGS administrator.

There are many ways to configure name resolution for the fabric domain. One simple way is to set up a conditional forwarder zone in DNS for the fabric. To set up this zone, run the following commands in an elevated Windows PowerShell console on a fabric DNS server. Substitute the names and addresses in the Windows PowerShell syntax below as needed for your environment. Add the parameter `-MasterServers` for the additional HGS nodes.

```
Add-DnsServerConditionalForwarderZone -Name 'bastion.local' -ReplicationScope
"Forest" -MasterServers <IP addresses of HGS server>
```

Next step

Configure HGS DNS and a one-way trust

Configure DNS forwarding in the HGS domain and a one-way trust with the fabric domain

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **ⓘ Important**
>
> AD mode is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

Use the following steps to set up DNS forwarding and establish a one-way trust with the fabric domain. These steps allow the HGS to locate the fabric domain controllers and validate group membership of the Hyper-V hosts.

1. Run the following command in an elevated PowerShell session to configure DNS forwarding. Replace fabrikam.com with the name of the fabric domain and type the IP addresses of DNS servers in the fabric domain. For higher availability, point to more than one DNS server.

   PowerShell

   ```
   Add-DnsServerConditionalForwarderZone -Name "fabrikam.com" -
   ReplicationScope "Forest" -MasterServers <DNSserverAddress1>,
   <DNSserverAddress2>
   ```

2. To create a one-way forest trust, run the following command in an elevated Command Prompt:

   Replace `bastion.local` with the name of the HGS domain and `fabrikam.com` with the name of the fabric domain. Provide the password for an admin of the fabric domain.

   PowerShell

   ```
   netdom trust bastion.local /domain:fabrikam.com
   /userD:fabrikam.com\Administrator /passwordD:<password> /add
   ```

Next step

Configure HTTPS

Feedback

Was this page helpful? Yes No

Configure HGS for HTTPS communications

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

By default, when you initialize the HGS server it will configure the IIS web sites for HTTP-only communications. All sensitive material being transmitted to and from HGS are always encrypted using message-level encryption, however if you desire a higher level of security you can also enable HTTPS by configuring HGS with an SSL certificate.

First, obtain an SSL certificate for HGS from your certificate authority. Each host machine will need to trust the SSL certificate, so it is recommended that you issue the SSL certificate from your company's public key infrastructure or a third party CA. Any SSL certificate supported by IIS is supported by HGS, however **the subject name on the certificate must match the fully qualified HGS service name** (cluster distributed network name). For instance, if the HGS domain is "bastion.local" and your HGS service name is "hgs", your SSL certificate should be issued for "hgs.bastion.local". You can add additional DNS names to the certificate's subject alternative name field if necessary.

Once you have the SSL certificate, open an elevated PowerShelll session and either provide the certificate path when you run Set-HgsServer:

PowerShell

```
$sslPassword = Read-Host -AsSecureString -Prompt "SSL Certificate Password"
Set-HgsServer -Http -Https -HttpsCertificatePath
'C:\temp\HgsSSLCertificate.pfx' -HttpsCertificatePassword $sslPassword
```

Or, if you have already installed the certificate into the local certificate store, you can reference it by thumbprint:

PowerShell

```
Set-HgsServer -Http -Https -HttpsCertificateThumbprint 'A1B2C3D4E5F6...'
```

> **ⓘ Important**
>
> Configuring HGS with an SSL certificate does not disable the HTTP endpoint. If you wish to only allow use of the HTTPS endpoint, configure Windows Firewall to block

inbound connections to port 80. **Do not modify the IIS bindings** for HGS websites to remove the HTTP endpoint; it is unsupported to do so.

Feedback

Was this page helpful? Yes No

Configure additional HGS nodes

Article • 09/21/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

In production environments, HGS should be set up in a high availability cluster to ensure that shielded VMs can be powered on even if an HGS node goes down. For test environments, secondary HGS nodes are not required.

Use one of these methods to add HGS nodes, as best suited for your environment.

Expand table

Environment	Option 1	Option 2
New HGS forest	Using PFX files	Using certificate thumbprints
Existing bastion forest	Using PFX files	Using certificate thumbprints

Prerequisites

Make sure that each additional node:

- Has the same hardware and software configuration as the primary node
- Is connected to the same network as the other HGS servers
- Can resolve the other HGS servers by their DNS names

Dedicated HGS forest with PFX certificates

1. Promote the HGS node to a domain controller
2. Initialize the HGS server

Promote the HGS node to a domain controller

1. Run Install-HgsServer to join the domain and promote the node to a domain controller.

```
PowerShell

$adSafeModePassword = ConvertTo-SecureString -AsPlainText '<password>'
-Force
```

```
$cred = Get-Credential 'relecloud\Administrator'

Install-HgsServer -HgsDomainName 'bastion.local' -HgsDomainCredential
$cred -SafeModeAdministratorPassword $adSafeModePassword -Restart
```

2. When the server reboots, log in with a domain administrator account.

Initialize the HGS server

Run the following command to join the existing HGS cluster.

PowerShell

```
Initialize-HgsServer -HgsServerIPAddress <IP address of first HGS Server>
```

Dedicated HGS forest with certificate thumbprints

1. Promote the HGS node to a domain controller
2. Initialize the HGS server
3. Install the private keys for the certificates

Promote the HGS node to a domain controller

1. Run Install-HgsServer to join the domain and promote the node to a domain controller.

   PowerShell

   ```
   $adSafeModePassword = ConvertTo-SecureString -AsPlainText '<password>'
   -Force

   $cred = Get-Credential 'relecloud\Administrator'

   Install-HgsServer -HgsDomainName 'bastion.local' -HgsDomainCredential
   $cred -SafeModeAdministratorPassword $adSafeModePassword -Restart
   ```

2. When the server reboots, log in with a domain administrator account.

Initialize the HGS server

Run the following command to join the existing HGS cluster.

PowerShell

```
Initialize-HgsServer -HgsServerIPAddress <IP address of first HGS Server>
```

Install the private keys for the certificates

If you did not provide a PFX file for either the encryption or signing certificates on the first HGS server, only the public key will be replicated to this server. You will need to install the private key by importing a PFX file containing the private key into the local certificate store or, in the case of HSM-backed keys, configuring the Key Storage Provider and associating it with your certificates per your HSM manufacturer's instructions.

Existing bastion forest with PFX certificates

1. Join the node to the existing domain
2. Grant the machine rights to retrieve gMSA password and run Install-ADServiceAccount
3. Initialize the HGS server

Join the node to the existing domain

1. Ensure at least one NIC on the node is configured to use the DNS server on your first HGS server.
2. Join the new HGS node to the same domain as your first HGS node.

Grant the machine rights to retrieve gMSA password and run Install-ADServiceAccount

1. Have a directory services admin add the computer account for your new node to the security group containing all of your HGS servers that is permissioned to allow those servers to use the HGS gMSA account.

2. Reboot the new node to obtain a new Kerberos ticket that includes the computer's membership in that security group. After the reboot completes, sign in with a domain identity that belongs to the local administrators group on the computer.

3. Install the HGS group managed service account on the node.

   PowerShell

```
Install-ADServiceAccount -Identity <HGSgMSAAccount>
```

Initialize the HGS server

Run the following command to join the existing HGS cluster.

PowerShell

```
Initialize-HgsServer -HgsServerIPAddress <IP address of first HGS Server>
```

Existing bastion forest with certificate thumbprints

1. Join the node to the existing domain
2. Grant the machine rights to retrieve gMSA password and run Install-ADServiceAccount
3. Initialize the HGS server
4. Install the private keys for the certificates

Join the node to the existing domain

1. Ensure at least one NIC on the node is configured to use the DNS server on your first HGS server.
2. Join the new HGS node to the same domain as your first HGS node.

Grant the machine rights to retrieve gMSA password and run Install-ADServiceAccount

1. Have a directory services admin add the computer account for your new node to the security group containing all of your HGS servers that is permissioned to allow those servers to use the HGS gMSA account.

2. Reboot the new node to obtain a new Kerberos ticket that includes the computer's membership in that security group. After the reboot completes, sign in with a domain identity that belongs to the local administrators group on the computer.

3. Install the HGS group managed service account on the node.

PowerShell

```
Install-ADServiceAccount -Identity <HGSgMSAAccount>
```

Initialize the HGS server

Run the following command to join the existing HGS cluster.

PowerShell

```
Initialize-HgsServer -HgsServerIPAddress <IP address of first HGS Server>
```

It will take up to 10 minutes for the encryption and signing certificates from the first HGS server to replicate to this node.

Install the private keys for the certificates

If you did not provide a PFX file for either the encryption or signing certificates on the first HGS server, only the public key will be replicated to this server. You will need to install the private key by importing a PFX file containing the private key into the local certificate store or, in the case of HSM-backed keys, configuring the Key Storage Provider and associating it with your certificates per your HSM manufacturer's instructions.

Configure HGS for HTTPS communications

If you want to secure HGS endpoints with an SSL certificate, you must configure the SSL certificate on this node, as well as every other node in the HGS cluster. SSL certificates *are not* replicated by HGS and do not need to use the same keys for every node (i.e. you can have different SSL certs for each node).

When requesting an SSL cert, ensure the cluster fully qualified domain name (as shown in the output of `Get-HgsServer`) is either the subject common name of the cert, or included as a subject alternative DNS name. When you've obtained a certificate from your certificate authority, you can configure HGS to use it with Set-HgsServer.

PowerShell

```
$sslPassword = Read-Host -AsSecureString -Prompt "SSL Certificate Password"
Set-HgsServer -Http -Https -HttpsCertificatePath
'C:\temp\HgsSSLCertificate.pfx' -HttpsCertificatePassword $sslPassword
```

If you already installed the certificate into the local certificate store and want to reference it by thumbprint, run the following command instead:

```PowerShell
Set-HgsServer -Http -Https -HttpsCertificateThumbprint 'A1B2C3D4E5F6...'
```

HGS will always expose both the HTTP and HTTPS ports for communication. It is unsupported to remove the HTTP binding in IIS, however you can use the Windows Firewall or other network firewall technologies to block communications over port 80.

Decommission an HGS node

To decommission an HGS node:

1. Clear the HGS configuration.

   This removes the node from the cluster and uninstalls the attestation and key protection services. If it's the last node in the cluster, -Force is needed to signify you do want to remove the last node and destroy the cluster in Active Directory.

   If HGS is deployed in a bastion forest (default), that's the only step. You can optionally unjoin the machine from the domain and remove the gMSA account from Active Directory.

2. If HGS created its own domain, you should also uninstall HGS to unjoin the domain and demote the domain controller.

Feedback

Was this page helpful? Yes No

Deploy guarded hosts

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The topics in this section describe the steps that a fabric administrator takes to configure Hyper-V hosts to work with the Host Guardian Service (HGS). Before you can start these steps, at least one node in the HGS cluster must be set up.

For TPM-trusted attestation:

1. Configure the fabric DNS: Tells how to set up a DNS forwarder from the fabric domain to the HGS domain.
2. Capture information required by HGS: Tells how to capture TPM identifiers (also called platform identifiers), create a Code Integrity policy, and create a TPM baseline. Then you will provide this information to the HGS administrator to configure attestation.
3. Confirm guarded hosts can attest

For host key attestation:

1. Create a host key: Tells how to set up a DNS forwarder from the fabric domain to the HGS domain.
2. Add the host key to the attestation service: Tells how to set up an Active Directory security group in the fabric domain, add guarded hosts as members of that group, and provide that group identifier to the HGS administrator.
3. Confirm guarded hosts can attest

For Admin-trusted attestation:

1. Configure the fabric DNS: Tells how to set up a DNS forwarder from the fabric domain to the HGS domain.
2. Create a security group: Tells how to set up an Active Directory security group in the fabric domain, add guarded hosts as members of that group, and provide that group identifier to the HGS administrator.
3. Confirm guarded hosts can attest

Additional References

- Deployment tasks for guarded fabrics and shielded VMs

Feedback

Was this page helpful? Yes No

Prerequisites for guarded hosts

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Review the host prerequisites for the mode of attestation you've chosen, then click the next step to add guarded hosts.

TPM-trusted attestation

Guarded hosts using TPM mode must meet the following prerequisites:

- **Hardware**: One host is required for initial deployment. To test Hyper-V live migration for shielded VMs, you must have at least two hosts.

  Hosts must have:
  - IOMMU and Second Level Address Translation (SLAT)
  - TPM 2.0
  - UEFI 2.3.1 or later
  - Configured to boot using UEFI (not BIOS or "legacy" mode)
  - Secure boot enabled

- **Operating system**: Windows Server 2016 Datacenter edition or later

  > **ⓘ Important**
  >
  > Make sure you install the latest cumulative update.

- **Role and features**: Hyper-V role and the Host Guardian Hyper-V Support feature. The Host Guardian Hyper-V Support feature is only available on Datacenter editions of Windows Server.

> **⚠ Warning**
>
> The Host Guardian Hyper-V Support feature enables Virtualization-based protection of code integrity that may be incompatible with some devices. We strongly recommend testing this configuration in your lab before enabling this feature. Failure to do so may result in unexpected failures up to and including data loss or a blue screen error (also called a stop error). For more information, see

Compatible hardware with Windows Server Virtualization-based protection of Code Integrity.

Next step:

Capture TPM info

Host key attestation

Guarded hosts using host key attestation must meet the following prerequisites:

- **Hardware**: Any server capable of running Hyper-V beginning with Windows Server 2019
- **Operating system**: Windows Server 2019 Datacenter edition
- **Role and features**: Hyper-V role and the Host Guardian Hyper-V Support feature

The host can be joined to either a domain or a workgroup.

For host key attestation, HGS must be running Windows Server 2019 and operating with v2 attestation. For more information see HGS prerequisites.

Next step:

Create a key pair

Admin-trusted attestation

> **Important**
>
> Admin-trusted attestation (AD mode) is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

Hyper-V hosts must meet the following prerequisites for AD mode:

- **Hardware**: Any server capable of running Hyper-V beginning with Windows Server 2016. One host is required for initial deployment. To test Hyper-V live migration for shielded VMs, you need at least two hosts.

- **Operating system**: Windows Server 2016 Datacenter edition

> **Important**
>
> Install the latest cumulative update.

- **Role and features**: Hyper-V role and the Host Guardian Hyper-V Support feature, which is only available in Windows Server 2016 Datacenter edition.

> **Warning**
>
> The Host Guardian Hyper-V Support feature enables Virtualization-based protection of code integrity that may be incompatible with some devices. We strongly recommend testing this configuration in your lab before enabling this feature. Failure to do so may result in unexpected failures up to and including data loss or a blue screen error (also called a stop error). For more information, see Compatible hardware with Windows Server 2016 Virtualization-based protection of Code Integrity.

Next step:

Place guarded hosts in a security group

Feedback

Was this page helpful? Yes No

Authorize guarded hosts using TPM-based attestation

Article • 01/25/2023 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

TPM mode uses a TPM identifier (also called a platform identifier or endorsement key [EKpub]) to begin determining whether a particular host is authorized as "guarded." This mode of attestation uses Secure Boot and code integrity measurements to ensure that a given Hyper-V host is in a healthy state and is running only trusted code. In order for attestation to understand what is and is not healthy, you must capture the following artifacts:

1. TPM identifier (EKpub)
    - This information is unique to each Hyper-V host
2. TPM baseline (boot measurements)
    - This is applicable to all Hyper-V hosts that run on the same class of hardware
3. Code integrity policy (an allowlist of allowed binaries)
    - This is applicable to all Hyper-V hosts that share common hardware and software

We recommend that you capture the baseline and CI policy from a "reference host" that is representative of each unique class of Hyper-V hardware configuration within your datacenter. Beginning with Windows Server version 1709, sample CI policies are included at C:\Windows\schemas\CodeIntegrity\ExamplePolicies.

Versioned attestation policies

Windows Server 2019 introduces a new method for attestation, called *v2 attestation*, where a TPM certificate must be present in order to add the EKPub to HGS. The v1 attestation method used in Windows Server 2016 allowed you to override this safety check by specifying the -Force flag when you run Add-HgsAttestationTpmHost or other TPM attestation cmdlets to capture the artifacts. Beginning with Windows Server 2019, v2 attestation is used by default and you need to specify the -PolicyVersion v1 flag when you run Add-HgsAttestationTpmHost if you need to register a TPM without a certificate. The -Force flag does not work with v2 attestation.

A host can only attest if all artifacts (EKPub + TPM baseline + CI Policy) use the same version of attestation. V2 attestation is tried first, and if that fails, v1 attestation is used. This means if you need to register a TPM identifier by using v1 attestation, you need to also specify the -PolicyVersion v1 flag to use v1 attestation when you capture the TPM baseline and create the CI policy. If the TPM baseline and CI policy were created by using v2 attestation and then later you need to add a guarded host without a TPM certificate, you need to re-create each artifact with the -PolicyVersion v1 flag.

Capture the TPM identifier (platform identifier or EKpub) for each host

1. In the fabric domain, make sure the TPM on each host is ready for use - that is, the TPM is initialized and ownership obtained. You can check the status of the TPM by opening the TPM Management Console (tpm.msc) or by running **Get-Tpm** in an elevated Windows PowerShell window. If your TPM is not in the **Ready** state, you will need to initialize it and set its ownership. This can be done in the TPM Management Console or by running **Initialize-Tpm**.

2. On each guarded host, run the following command in an elevated Windows PowerShell console to obtain its EKpub. For `<HostName>`, substitute the unique host name with something suitable to identify this host - this can be its hostname or the name used by a fabric inventory service (if available). For convenience, name the output file using the host's name.

   PowerShell

   ```
   (Get-PlatformIdentifier -Name '<HostName>').InnerXml | Out-file <Path>
   <HostName>.xml -Encoding UTF8
   ```

3. Repeat the preceding steps for each host that will become a guarded host, being sure to give each XML file a unique name.

4. Provide the resulting XML files to the HGS administrator.

5. In the HGS domain, open an elevated Windows PowerShell console on an HGS server and run the following command. Repeat the command for each of the XML files.

   PowerShell

   ```
   Add-HgsAttestationTpmHost -Path <Path><Filename>.xml -Name <HostName>
   ```

> ⓘ **Note**
>
> If you encounter an error when adding a TPM identifier regarding an untrusted Endorsement Key Certificate (EKCert), ensure that the **trusted TPM root certificates have been added** to the HGS node. Additionally, some TPM vendors do not use EKCerts. You can check if an EKCert is missing by opening the XML file in an editor such as Notepad and checking for an error message indicating no EKCert was found. If this is the case, and you trust that the TPM in your machine is authentic, you can use the `-Force` parameter to add the host identifier to HGS. In Windows Server 2019, you need to also use the `-PolicyVersion v1` parameter when using `-Force`. This creates a policy consistent with the Windows Server 2016 behavior and will require you to use `-PolicyVersion v1` when registering the CI policy and the TPM baseline as well.

Create and apply a code integrity policy

A code integrity policy helps ensure that only the executables you trust to run on a host are allowed to run. Malware and other executables outside the trusted executables are prevented from running.

Each guarded host must have a code integrity policy applied in order to run shielded VMs in TPM mode. You specify the exact code integrity policies you trust by adding them to HGS. Code integrity policies can be configured to enforce the policy, blocking any software that does not comply with the policy, or simply audit (log an event when software not defined in the policy is executed).

Starting with Windows Server version 1709, sample code integrity policies are included with Windows at C:\Windows\schemas\CodeIntegrity\ExamplePolicies. Two policies are recommended for Windows Server:

- **AllowMicrosoft**: Allows all files signed by Microsoft. This policy is recommended for server applications such as SQL or Exchange, or if the server is monitored by agents published by Microsoft.
- **DefaultWindows_Enforced**: Allows only files that shipped in Windows and doesn't permit other applications released by Microsoft, such as Office. This policy is recommended for servers that run only built-in server roles and features such as Hyper-V.

It is recommended that you first create the CI policy in audit (logging) mode to see if it's missing anything, then enforce the policy for host production workloads.

If you use the New-CIPolicy cmdlet to generate your own code integrity policy, you will need to decide the rule levels to use. We recommend a primary level of **Publisher** with fallback to **Hash**, which allows most digitally signed software to be updated without changing the CI policy. New software written by the same publisher can also be installed on the server without changing the CI policy. Executables that are not digitally signed will be hashed. Updates to these files will require you to create a new CI policy. For more information about the available CI policy rule levels, see Deploy code integrity policies: policy rules and file rules and cmdlet help.

1. On the reference host, generate a new code integrity policy. The following commands create a policy at the **Publisher** level with fallback to **Hash**. It then converts the XML file to the binary file format Windows and HGS need to apply and measure the CI policy, respectively.

   PowerShell

   ```
   New-CIPolicy -Level Publisher -Fallback Hash -FilePath
   'C:\temp\HW1CodeIntegrity.xml' -UserPEs

   ConvertFrom-CIPolicy -XmlFilePath 'C:\temp\HW1CodeIntegrity.xml' -
   BinaryFilePath 'C:\temp\HW1CodeIntegrity.p7b'
   ```

   > **Note**
   >
   > The above command creates a CI policy in audit mode only. It will not block unauthorized binaries from running on the host. You should only use enforced policies in production.

2. Keep your Code Integrity policy file (XML file) where you can easily find it. You will need to edit this file later to enforce the CI policy or merge in changes from future updates made to the system.

3. Apply the CI policy to your reference host:

   a. Run the following command to configure the machine to use your CI policy. You can also deploy the CI policy with Group Policy or System Center Virtual Machine Manager.

      PowerShell

```
Invoke-CimMethod -Namespace root/Microsoft/Windows/CI -ClassName
PS_UpdateAndCompareCIPolicy -MethodName Update -Arguments @{
FilePath = "C:\temp\HW1CodeIntegrity.p7b" }
```

b. Restart the host to apply the policy.

4. Test the code integrity policy by running a typical workload. This may include running VMs, any fabric management agents, backup agents, or troubleshooting tools on the machine. Check if there are any code integrity violations and update your CI policy if necessary.

5. Change your CI policy to enforced mode by running the following commands against your updated CI policy XML file.

```
PowerShell

Set-RuleOption -FilePath 'C:\temp\HW1CodeIntegrity.xml' -Option 3 -
Delete

ConvertFrom-CIPolicy -XmlFilePath 'C:\temp\HW1CodeIntegrity.xml' -
BinaryFilePath 'C:\temp\HW1CodeIntegrity_enforced.p7b'
```

6. Apply the CI policy to all of your hosts (with identical hardware and software configuration) using the following commands:

```
PowerShell

Invoke-CimMethod -Namespace root/Microsoft/Windows/CI -ClassName
PS_UpdateAndCompareCIPolicy -MethodName Update -Arguments @{ FilePath =
"C:\temp\HW1CodeIntegrity.p7b" }

Restart-Computer
```

> **Note**
>
> Be careful when applying CI policies to hosts and when updating any software on these machines. Any kernel mode drivers that are non-compliant with the CI Policy may prevent the machine from starting up.

7. Provide the binary file (in this example, HW1CodeIntegrity_enforced.p7b) to the HGS administrator.

8. In the HGS domain, copy the code integrity policy to an HGS server and run the following command.

For `<PolicyName>`, specify a name for the CI policy that describes the type of host it applies to. A best practice is to name it after the make/model of your machine and any special software configuration running on it. For `<Path>`, specify the path and filename of the code integrity policy.

PowerShell

```
Add-HgsAttestationCIPolicy -Path <Path> -Name '<PolicyName>'
```

> **Note**
>
> If you're using a signed code integrity policy, register an unsigned copy of the same policy with HGS. The signature on code integrity policies is used to control updates to the policy, but is not measured into the host TPM and therefore cannot be attested to by HGS.

Capture the TPM baseline for each unique class of hardware

A TPM baseline is required for each unique class of hardware in your datacenter fabric. Use a "reference host" again.

1. On the reference host, make sure that the Hyper-V role and the Host Guardian Hyper-V Support feature are installed.

   > **Warning**
   >
   > The Host Guardian Hyper-V Support feature enables Virtualization-based protection of code integrity that may be incompatible with some devices. We strongly recommend testing this configuration in your lab before enabling this feature. Failure to do so may result in unexpected failures up to and including data loss or a blue screen error (also called a stop error).

   PowerShell

   ```
   Install-WindowsFeature Hyper-V, HostGuardian -IncludeManagementTools -
   Restart
   ```

2. To capture the baseline policy, run the following command in an elevated Windows PowerShell console.

```PowerShell
Get-HgsAttestationBaselinePolicy -Path 'HWConfig1.tcglog'
```

> **Note**
>
> You will need to use the **-SkipValidation** flag if the reference host does not have Secure Boot enabled, an IOMMU present, Virtualization Based Security enabled and running, or a code integrity policy applied. These validations are designed to make you aware of the minimum requirements of running a shielded VM on the host. Using the -SkipValidation flag does not change the output of the cmdlet; it merely silences the errors.

3. Provide the TPM baseline (TCGlog file) to the HGS administrator.

4. In the HGS domain, copy the TCGlog file to an HGS server and run the following command. Typically, you will name the policy after the class of hardware it represents (for example, "Manufacturer Model Revision").

```PowerShell
Add-HgsAttestationTpmPolicy -Path <Filename>.tcglog -Name
'<PolicyName>'
```

Next step

Confirm attestation

Feedback

Was this page helpful? Yes No

Create a host key and add it to HGS

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic covers how to prepare Hyper-V hosts to become guarded hosts using host key attestation (Key mode). You'll create a host key pair (or use an existing certificate) and add the public half of the key to HGS.

Create a host key

1. Install Windows Server 2019 on your Hyper-V host machine.

2. Install the Hyper-V and Host Guardian Hyper-V Support features:

   PowerShell

   ```
   Install-WindowsFeature Hyper-V, HostGuardian -IncludeManagementTools -Restart
   ```

3. Generate a host key automatically, or select an existing certificate. If you are using a custom certificate, it should have at least a 2048-bit RSA key, Client Authentication EKU, and Digital Signature key usage.

   PowerShell

   ```
   Set-HgsClientHostKey
   ```

   Alternatively, you can specify a thumbprint if you want to use your own certificate. This can be useful if you want to share a certificate across multiple machines, or use a certificate bound to a TPM or an HSM. Here's an example of creating a TPM-bound certificate (which prevents it from having the private key stolen and used on another machine and requires only a TPM 1.2):

   PowerShell

   ```
   $tpmBoundCert = New-SelfSignedCertificate -Subject "Host Key Attestation ($env:computername)" -Provider "Microsoft Platform Crypto Provider"
   Set-HgsClientHostKey -Thumbprint $tpmBoundCert.Thumbprint
   ```

4. Get the public half of the key to provide to the HGS server. You can use the following cmdlet or, if you have the certificate stored elsewhere, provide a .cer containing the public half of the key. Note that we are only storing and validating the public key on HGS; we do not keep any certificate information nor do we validate the certificate chain or expiration date.

   PowerShell

   ```
   Get-HgsClientHostKey -Path "C:\temp\$env:hostname-HostKey.cer"
   ```

5. Copy the .cer file to your HGS server.

Add the host key to the attestation service

This step is done on the HGS server and allows the host to run shielded VMs. It is recommended that you set the name to the FQDN or resource identifier of the host machine, so you can easily refer to which host the key is installed on.

PowerShell

```
Add-HgsAttestationHostKey -Name MyHost01 -Path "C:\temp\MyHost01-
HostKey.cer"
```

Next step

- Confirm hosts can attest successfully

Additional References

- Deploying the Host Guardian Service for guarded hosts and shielded VMs

---

Feedback

Was this page helpful? Yes No

Create a security group for guarded hosts and register the group with HGS

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **ⓘ Important**
>
> AD mode is deprecated beginning with Windows Server 2019. For environments where TPM attestation is not possible, configure host key attestation. Host key attestation provides similar assurance to AD mode and is simpler to set up.

This topic describes the intermediate steps to prepare Hyper-V hosts to become guarded hosts using Admin-trusted attestation (AD mode). Before taking these steps, complete the steps in Configuring the fabric DNS for hosts that will become guarded hosts.

Create a security group and add hosts

1. Create a new **GLOBAL** security group in the fabric domain and add Hyper-V hosts that will run shielded VMs. Restart the hosts to update their group membership.
2. Use Get-ADGroup to obtain the security identifier (SID) of the security group and provide it to the HGS administrator.

```
PowerShell

Get-ADGroup "Guarded Hosts"
```

```
PS C:\> Get-ADGroup 'Guarded Hosts'


DistinguishedName : CN=Guarded Hosts,CN=Users,DC=contoso,DC=com
GroupCategory     : Security
GroupScope        : Global
Name              : Guarded Hosts
ObjectClass       : group
ObjectGUID        : b00c72cc-69dd-4010-9c97-f31062199f9c
SamAccountName    : Guarded Hosts
SID               : S-1-5-21-947661373-489659989-2802621743-1119
```

Register the SID of the security group with HGS

1. On an HGS server, run the following command to register the security group with HGS. Re-run the command if necessary for additional groups. Provide a friendly name for the group. It does not need to match the Active Directory security group name.

   PowerShell

   ```
   Add-HgsAttestationHostGroup -Name "<GuardedHostGroup>" -Identifier "
   <SID>"
   ```

2. To verify the group was added, run Get-HgsAttestationHostGroup.

Next step

Confirm attestation

Additional References

- Deploying the Host Guardian Service for guarded hosts and shielded VMs

---

Feedback

Was this page helpful? Yes No

Confirm guarded hosts can attest

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

A fabric administrator needs to confirm that Hyper-V hosts can run as guarded hosts. Complete the following steps on at least one guarded host:

1. If you have not already installed the Hyper-V role and Host Guardian Hyper-V Support feature, install them with the following command:

   PowerShell

   ```
   Install-WindowsFeature Hyper-V, HostGuardian -IncludeManagementTools -Restart
   ```

2. Make sure the Hyper-V host can resolve the HGS DNS name and has network connectivity to reach port 80 (or 443 if you set up HTTPS) on the HGS server.

3. Configure the host's Key Protection and Attestation URLs:

   - **Through Windows PowerShell**: You can configure the Key Protection and Attestation URLs by executing the following command in an elevated Windows PowerShell console. For <FQDN>, use the Fully Qualified Domain Name (FQDN) of your HGS cluster (for example, hgs.bastion.local, or ask the HGS administrator to run the **Get-HgsServer** cmdlet on the HGS server to retrieve the URLs).

     PowerShell

     ```
     Set-HgsClientConfiguration -AttestationServerUrl
     'http://<FQDN>/Attestation' -KeyProtectionServerUrl
     'http://<FQDN>/KeyProtection'
     ```

     To configure a fallback HGS server, repeat this command and specify the fallback URLs for the Key Protection and Attestation services. For more information, see Fallback configuration.

   - **Through VMM**: If you are using System Center Virtual Machine Manager (VMM), you can configure Attestation and Key Protection URLs in VMM. For details, see Configure global HGS settings in **Provision guarded hosts in VMM**.

> **Notes**
>
> - If the HGS administrator enabled HTTPS on the HGS server, begin the URLs with `https://`.
> - If the HGS administrator enabled HTTPS on the HGS server and used a self-signed certificate, you will need to import the certificate into the Trusted Root Certificate Authorities store on every host. To do this, run the following command on each host: `PowerShell Import-Certificate -FilePath "C:\temp\HttpsCertificate.cer" -CertStoreLocation Cert:\LocalMachine\Root`
> - If you've configured HGS Client to use HTTPS and have disabled TLS 1.0 systemwide, see our modern TLS guidance

4. To initiate an attestation attempt on the host and view the attestation status, run the following command:

```PowerShell
Get-HgsClientConfiguration
```

The output of the command indicates whether the host passed attestation and is now guarded. If `IsHostGuarded` does not return **True**, you can run the HGS diagnostics tool, Get-HgsTrace, to investigate. To run diagnostics, enter the following command in an elevated Windows PowerShell prompt on the host:

```PowerShell
Get-HgsTrace -RunDiagnostics -Detailed
```

> **ⓘ Important**
>
> If you're using Windows Server 2019 or Windows 10, version 1809 or later, and are using code integrity policies, `Get-HgsTrace` return a failure for the **Code Integrity Policy Active** diagnostic. You can safely ignore this result when it is the only failing diagnostic.

Next step

Deploy shielded VMs

Additional References

- Deploy the Host Guardian Service (HGS)
- Deploy shielded VMs

Feedback

Was this page helpful? Yes No

Provision guarded hosts in VMM

This article describes how to deploy guarded Hyper-V hosts in a System Center Virtual Machine Manager (VMM) compute fabric. Learn more about guarded fabric.

There are a couple of ways to set up guarded Hyper-V hosts in a VMM fabric.

- **Configure an existing host to be a guarded host**: You can configure an existing host to run shielded VMs.
- **Add or provision a new guarded host**: This host could be:
  - An existing Windows Server computer (with or without the Hyper-V role)
  - A bare-metal computer

You set up guarded hosts in the VMM fabric as follows:

1. **Configure global HGS settings**: VMM connects all the guarded hosts to the same Host Guardian Service (HGS) server so that you can successfully migrate shielded VMs between the hosts. You specify the global HGS settings that apply to all the guarded hosts, and you can specify the host-specific settings that override the global settings. Settings include:

   - **Attestation URL**: The URL that the host uses to connect to the HGS attestation service. This service authorizes a host to run shielded VMs.
   - **Key protection server URL**: The URL that the host uses to retrieve the key needed to decrypt VMs. The host must pass attestation to retrieve keys.
   - **Code integrity policies**: A code integrity policy restricts the software that can run on a guarded host. When HGS is configured to use TPM attestation, guarded hosts must be configured to use a code integrity policy authorized by the HGS server. You can specify the location of code integrity policies in VMM and deploy them to your hosts. This is optional and isn't required to manage a guarded fabric.
   - **VM shielding helper VHD**: A specially prepared virtual hard disk that is used to convert the existing VMs to shielded VMs. You must configure this setting if you wish to shield the existing VMs.

2. **Configure the cloud**: If the guarded host will be included in a VMM cloud, you need to enable the cloud to support shielded VMs.

Before you start

Ensure that you've deployed and configured the Host Guardian Service before proceeding. Learn more about configuring HGS in the Windows Server documentation.

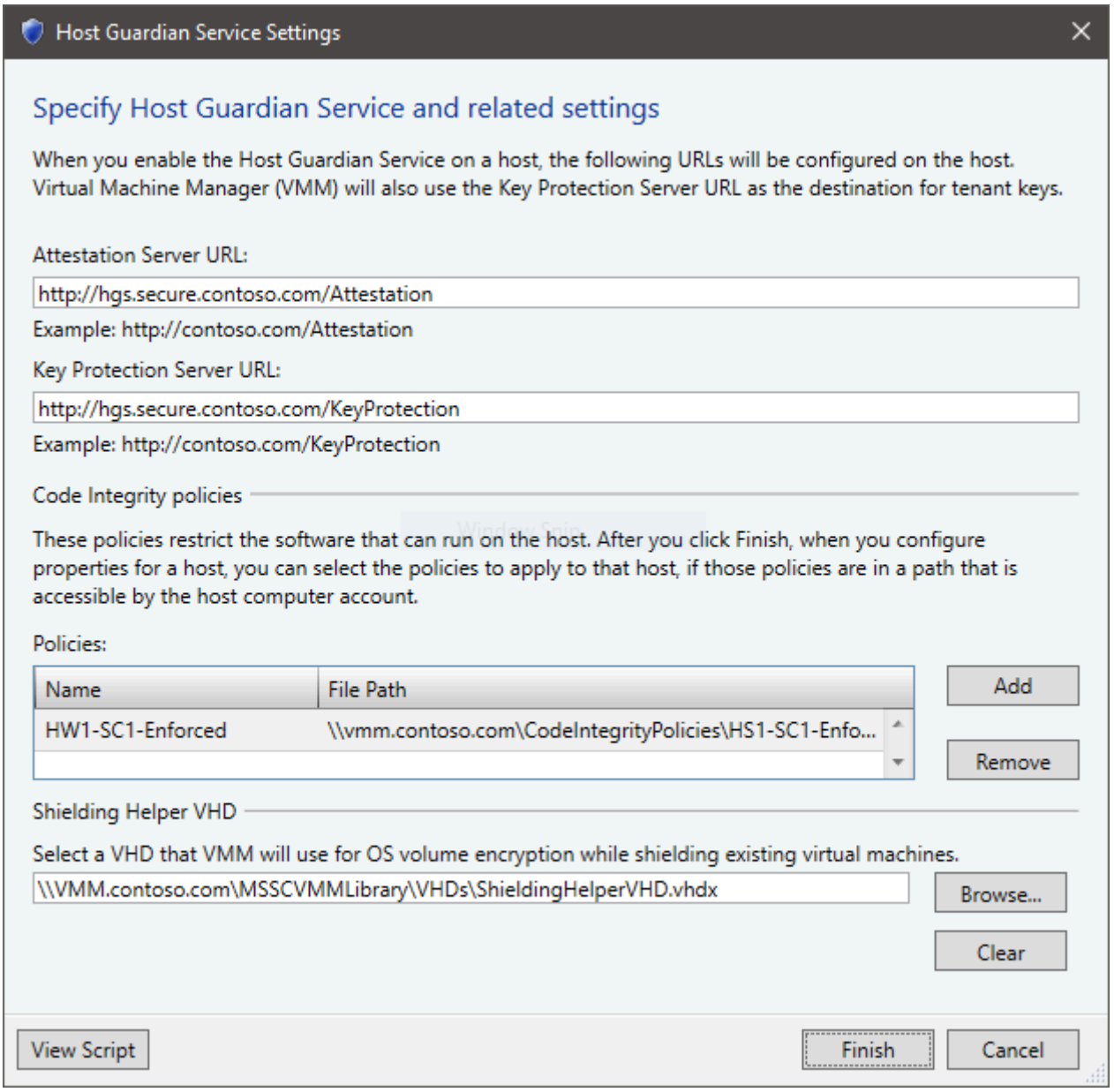
Additionally, ensure any hosts that will become guarded hosts meet the guarded host prerequisites:

- **Operating system**: Host servers must run Windows Server Datacenter. It's recommended to use Server Core for guarded hosts.
- **Role and features**: Host servers should be running the Hyper-V role and the Host Guardian Hyper-V Support feature. Host Guardian Hyper-V Support lets the host communicate with HGS to attest to its health and request keys for shielded VMs. If your host is running Nano Server, it should have the Compute, SCVMM-Package, SCVMM-Compute, SecureStartup, and ShieldedVM packages installed.
- **TPM-attestation**: If your HGS is configured to use TPM attestation, the host servers must:
  - Use UEFI 2.3.1c and a TPM 2.0 module
  - Boot in UEFI mode (not BIOS or **legacy** mode)
  - Enable Secure Boot
- **HGS registration**: Hyper-V hosts must be registered with HGS. How they're registered depends on whether HGS is using AD or TPM attestation. Learn more
- **Live migration**: If you want to live migrate shielded VMs, you need to deploy two or more guarded hosts.
- **Domain**: Guarded hosts and the VMM server must be in the same domain or in domains with a two-way trust.

Configure global HGS settings

Before you can add guarded hosts to your VMM compute fabric, you must configure VMM with information about the HGS for the fabric. The same HGS will be used for all guarded hosts managed by VMM.

1. Obtain the attestation and key protection URLs for your fabric from your HGS administrator.
2. In the VMM console, select **Settings** > **Host Guardian Service Settings**.
3. Enter the attestation and key protection URLs in the respective fields. You don't need to configure the code integrity policies and VM shielding helper VHD sections at this time.



4. Select **Finish** to save the configuration.

Add or provision a new guarded host

1. Add the host:

   - If you want to add an existing server running Windows Server as a guarded Hyper-V host, add it to the fabric.
   - If you want to provision a Hyper-V host from a bare-metal computer, follow these prerequisites and instructions.

   > ⓘ **Note**

> You can deploy the host as guarded when you provision it (Add Resource Wizard > **OS Settings** > **Configure as guarded host**).

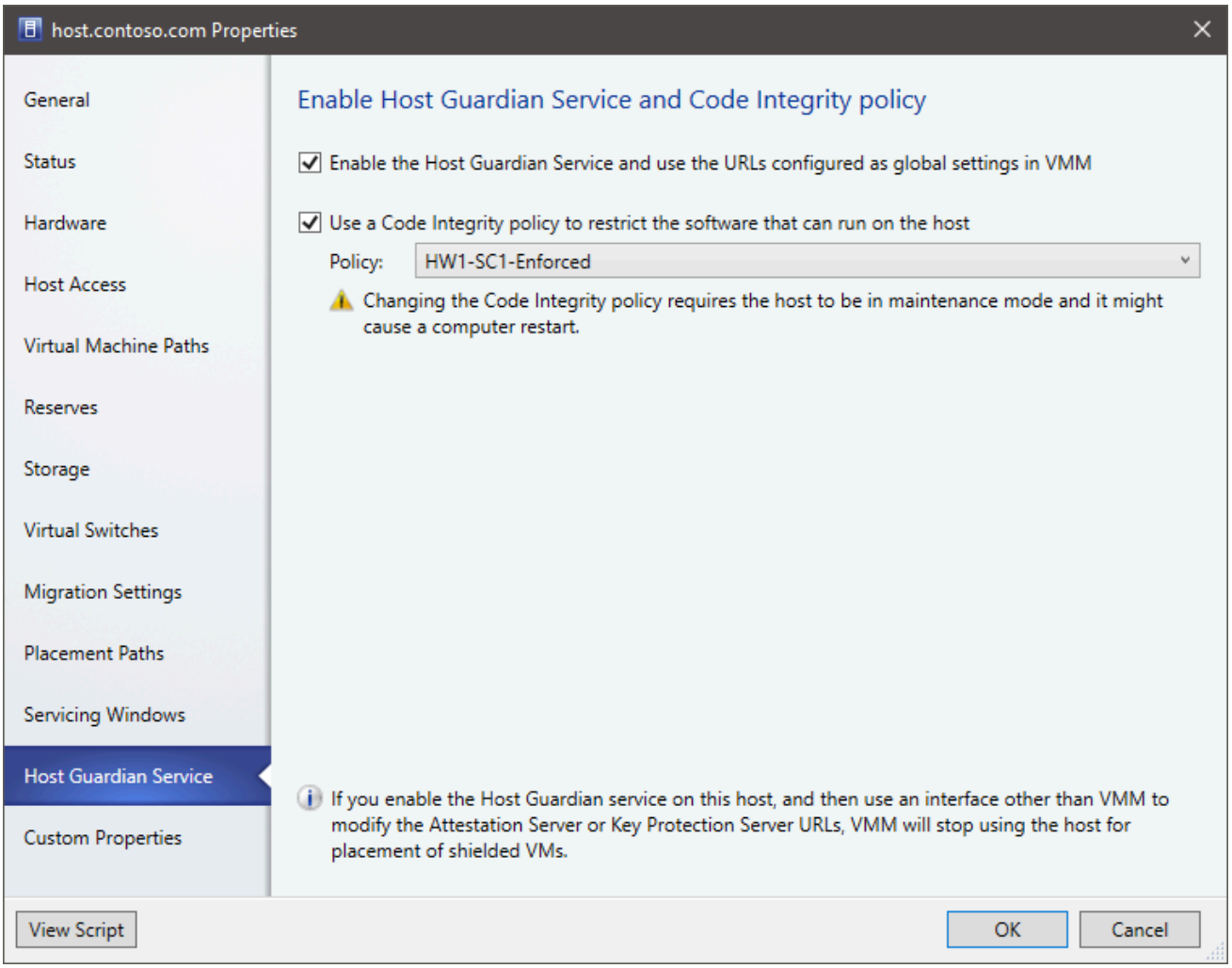
2. Continue to the next section to configure the host as a guarded host.

Configure an existing host to be a guarded host

To configure an existing Hyper-V host managed by VMM to be a guarded host, complete the following steps:

1. Place the host in maintenance mode.

2. In **All Hosts**, right-click the host > **Properties** > **Host Guardian Service**.



3. Select to enable the Host Guardian Hyper-V Support feature and configure the host.

> ⓘ **Note**
>
> - The global attestation and key protection server URLs will be set on the host.

- If you modify these URLs outside the VMM console, you need to update them in VMM too. If you don't, VMM won't place shielded VMs on the host until the URLs match again. You can also uncheck and re-check the **Enable** box to reconfigure the host with the URLs configured in VMM.

4. If you're using VMM to manage code integrity policies, you can enable the second checkbox and select the appropriate policy for the system.

5. Select **OK** to update the host's configuration.

6. Take the host out of the maintenance mode.

VMM checks that the host passes attestation when you add it and every time the host status is refreshed. VMM only deploys and migrates shielded VMs on hosts that have passed attestation. You can check the attestation status of a host in **Properties** > **Status** > **HGS Client Overall**.

Enable guarded hosts on a VMM cloud

Enable a cloud to support guarded hosts:

1. In the VMM console, select **VMs and Services** > **Clouds**. Right-click the cloud name > **Properties**.
2. In **General** > **Shielded VM support**, select **Supported on this private cloud**.

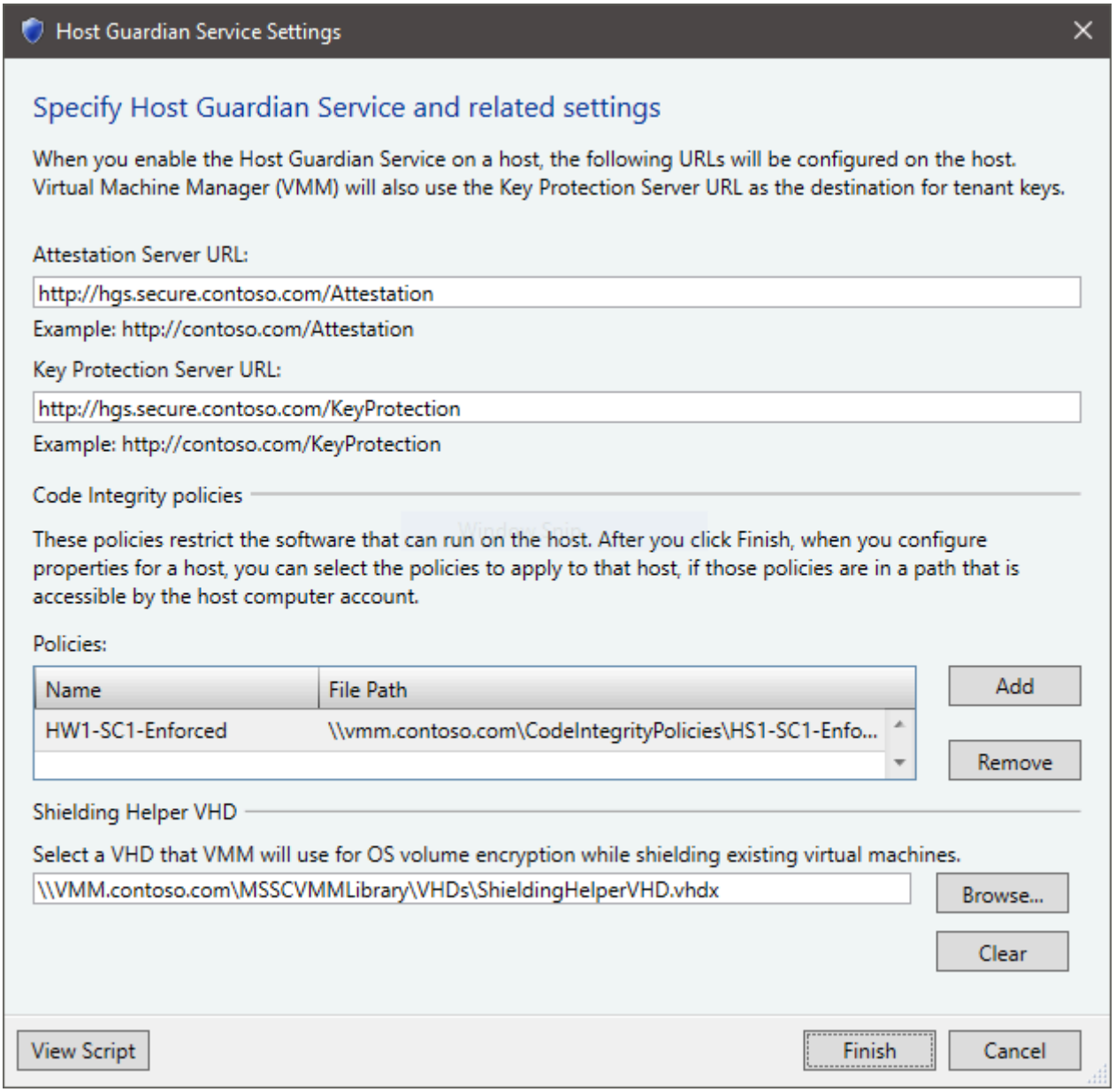
Manage and deploy code integrity policies with VMM

In guarded fabrics configured to use TPM attestation, each host must be configured with a code integrity policy that is trusted by the Host Guardian Service. To ease the management of code integrity policies, you can optionally use VMM to deploy new or updated policies to your guarded hosts.

To deploy a code integrity policy to a guarded host managed by VMM, complete the following steps:

1. Create a code integrity policy for each reference host in your environment. You need a different CI policy for each unique hardware and software configuration of your guarded hosts.
2. Store the CI policies in a secure file share. The computer accounts for each guarded host require **read access** to the share. Only trusted administrators should have write access.
3. In the VMM console, select **Settings** > **Host Guardian Service Settings**.

4. Under the Code Integrity Policies section, select **Add** and specify a friendly name and the path to a CI policy. Repeat this step for each unique CI policy. Ensure to name your policies in a manner that will help you identify which policy should be applied to which hosts.

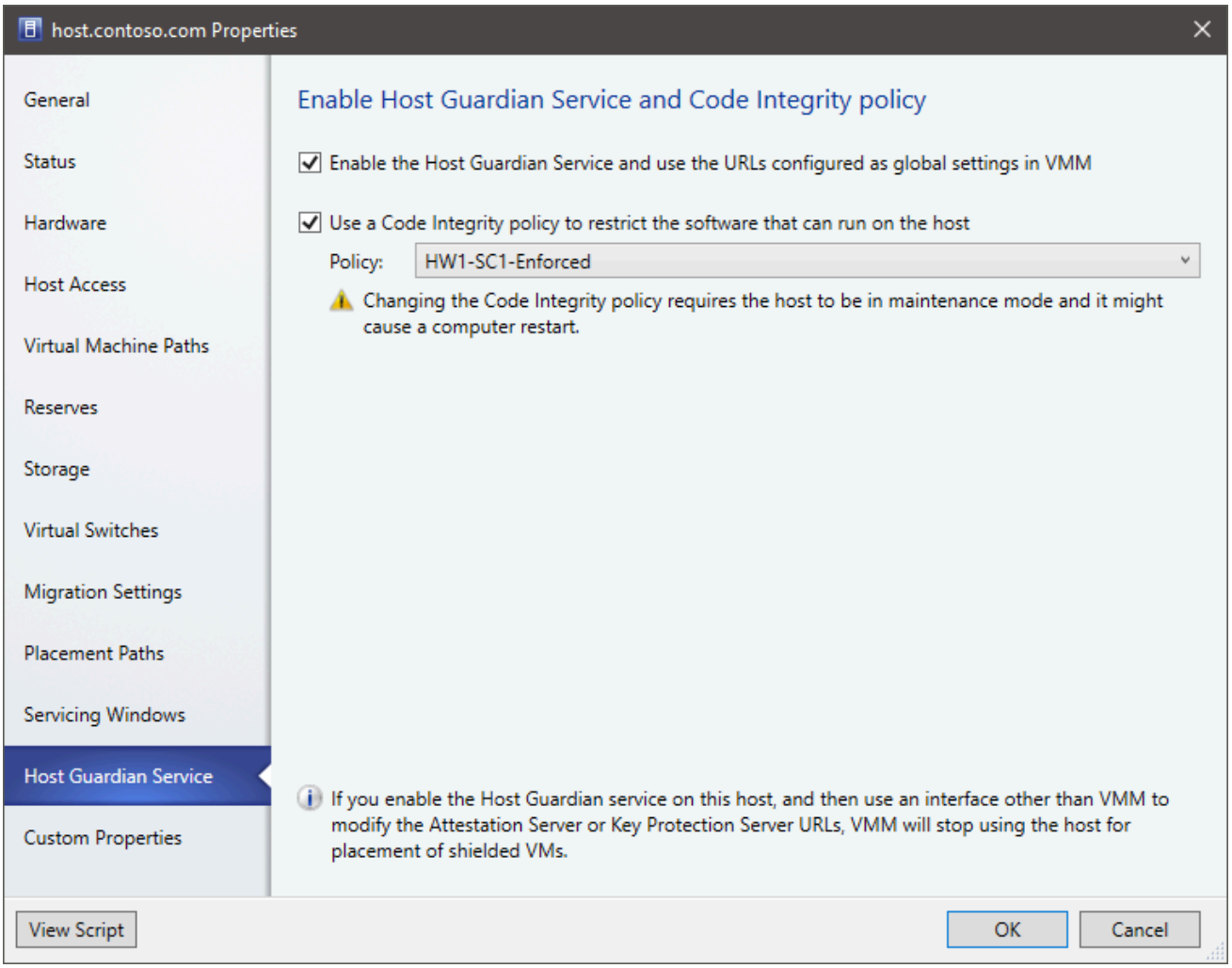


5. Select **Finish** to save the configuration.

Now, for each guarded host, complete the following steps to apply a code integrity policy:

1. Place the host in maintenance mode.

2. In **All Hosts**, right-click the host > **Properties** > **Host Guardian Service**.



3. Select to enable the option to configure the host with a code integrity policy. Then select the appropriate policy for the system.
4. Select **OK** to apply the configuration change. The host can restart to apply the new policy.
5. Take the host out of maintenance mode.

> ⚠ **Warning**
>
> Ensure that you select the correct code integrity policy for the host. If an incompatible policy is applied to the host, some applications, drivers, or operating system components may no longer work.

If you update the code integrity policy in the file share and wish to also update the guarded hosts, you can do so by completing the following steps:

1. Place the host in maintenance mode.
2. In **All Hosts**, right-click the host > **Apply Latest Code Integrity Policy**.
3. Take the host out of maintenance mode.

Next steps

- Set up a shielded template disk, utility disk, and VM template.

Last updated on 07/17/2024

Deploy shielded VMs

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The following topics describe how a tenant can work with shielded VMs.

1. (Optional) Create a Windows template disk or create a Linux template disk. The template disk can be created by either the tenant or the hosting service provider.
2. (Optional) Convert an existing Windows VM to a shielded VM.
3. Create shielding data to define a shielded VM.

   For a description and diagram of a shielding data file, see What is shielding data and why is it necessary?

   For information about creating an answer file to include in a shielded data file, see Shielded VMs - Generate an answer file by using the New-ShieldingDataAnswerFile function.
4. Create a shielded VM:

   - Using **Windows Azure Pack**: Deploy a shielded VM by using Windows Azure Pack
   - Using **Virtual Machine Manager**: Deploy a shielded VM by using Virtual Machine Manager

Next step

Create a shielded VM template

Additional References

- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Create a Windows shielded VM template disk

Article • 10/08/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

As with regular VMs, you can create a VM template (for example, a VM template in Virtual Machine Manager (VMM)) to make it easy for tenants and administrators to deploy new VMs on the fabric using a template disk. Because shielded VMs are security-sensitive assets, there are additional steps to create a VM template that supports shielding. This topic covers the steps to create a shielded template disk and a VM template in VMM.

To understand how this topic fits in the overall process of deploying shielded VMs, see Hosting service provider configuration steps for guarded hosts and shielded VMs.

Prepare an operating system VHDX

First prepare an OS disk that you will then run through the Shielded Template Disk Creation Wizard. This disk will be used as the OS disk in your tenant's VMs. You can use any existing tooling to create this disk, such as Microsoft Desktop Image Service Manager (DISM), or manually set up a VM with a blank VHDX and install the OS onto that disk. When setting up the disk, it must adhere to the following requirements that are specific to generation 2 and/or shielded VMs:

Expand table

Requirement for VHDX	Reason
Must be a GUID Partition Table (GPT) disk	Needed for generation 2 virtual machines to support UEFI
Disk type must be **Basic** as opposed to **Dynamic**. Note: This refers to the logical disk type, not the "dynamically expanding" VHDX feature supported by Hyper-V.	BitLocker does NOT support dynamic disks.
The disk has at least two partitions. One partition must include the drive on which Windows is installed. This is the drive that BitLocker will encrypt. The other partition is the active partition, which contains the bootloader and remains unencrypted so that the computer can be started.	Needed for BitLocker

Requirement for VHDX	Reason
File system is NTFS	Needed for BitLocker
The operating system installed on the VHDX is one of the following: - Windows Server 2019, Windows Server 2016, Windows Server 2012 R2, or Windows Server 2012 - Windows 10, Windows 8.1, Windows 8	Needed to support generation 2 virtual machines and the Microsoft Secure Boot template
Operating system must be generalized (run sysprep.exe)	Template provisioning involves specializing VMs for a specific tenant's workload

> **Note**
>
> If you use VMM, do not copy the template disk into the VMM library at this stage.

Run Windows Update on the template operating system

On the template disk, verify that the operating system has all of the latest Windows updates installed. Recently released updates improve the reliability of the end-to-end shielding process - a process that may fail to complete if the template operating system is not up-to-date.

Prepare and protect the VHDX with the template disk wizard

To use a template disk with shielded VMs, the disk must be prepared and encrypted with BitLocker by using the Shielded Template Disk Creation Wizard. This wizard will generate a hash for the disk and add it to a volume signature catalog (VSC). The VSC is signed using a certificate you specify and is used during the provisioning process to ensure the disk being deployed for a tenant has not been altered or replaced with a disk the tenant does not trust. Finally, BitLocker is installed on the disk's operating system (if it is not already there) to prepare the disk for encryption during VM provisioning.

> **Note**
>
> The template disk wizard will modify the template disk you specify in-place. You may want to make a copy of the unprotected VHDX before running the wizard to

make updates to the disk at a later time. You will not be able to modify a disk that has been protected with the template disk wizard.

Perform the following steps on a computer running Windows Server 2016, Windows 10 (with Remote Server Management Tools, RSAT installed) or later (does not need to be a guarded host or a VMM server):

1. Copy the generalized VHDX created in Prepare an operating system VHDX to the server, if it is not already there.

2. To administer the server locally, install the **Shielded VM Tools** feature from **Remote Server Administration Tools** on the server.

   ```
   Install-WindowsFeature RSAT-Shielded-VM-Tools -Restart
   ```

   You can also administer the server from a client computer on which you have installed the Windows 10 Remote Server Administration Tools.

3. Obtain or create a certificate to sign the VSC for the VHDX that will become the template disk for new shielded VMs. Details about this certificate will be shown to tenants when they create their shielding data files and are authorizing disks they trust. Therefore, it is important to obtain this certificate from a certificate authority mutually trusted by you and your tenants. In enterprise scenarios where you are both the host and tenant, you might consider issuing this certificate from your PKI.

   If you are setting up a test environment and just want to use a self-signed certificate to prepare your template disk, run a command similar to the following:

   ```
   New-SelfSignedCertificate -DnsName publisher.fabrikam.com
   ```

4. Start the **Template Disk Wizard** from the **Administrative Tools** folder on the Start menu or by typing **TemplateDiskWizard.exe** into a command prompt.

5. On the **Certificate** page, click **Browse** to display a list of certificates. Select the certificate with which to prepare the disk template. Click **OK** and then click **Next**.

6. On the Virtual Disk page, click **Browse** to select the VHDX that you have prepared, then click **Next**.

7. On the Signature Catalog page, provide a friendly **disk name** and **version.** These fields are present to help you identify the disk once it has been prepared.

   For example, for **disk name** you could type *WS2016* and for **Version**, *1.0.0.0*

8. Review your selections on the Review Settings page of the wizard. When you click **Generate**, the wizard will enable BitLocker on the template disk, compute the hash of the disk, and create the Volume Signature Catalog, which is stored in the VHDX metadata.

   Wait until the prep process has finished before attempting to mount or move the template disk. This process may take a while to complete, depending on the size of your disk.

   > **Important**
   >
   > Template disks can only be used with the secure shielded VM provisioning process. Attempting to boot a regular (unshielded) VM using a template disk will likely result in a stop error (blue screen) and is unsupported.

9. On the **Summary** page, information about the disk template, the certificate used to sign the VSC, and the certificate issuer is shown. Click **Close** to exit the wizard.

If you use VMM, follow the steps in the remaining sections in this topic to incorporate a template disk into a shielded VM template in VMM.

Copy the template disk to the VMM Library

If you use VMM, after you create a template disk, you need to copy it to a VMM library share so hosts can download and use the disk when provisioning new VMs. Use the following procedure to copy the template disk into the VMM library and then refresh the library.

1. Copy the VHDX file to the VMM library share folder. If you used the default VMM configuration, copy the template disk to \<*vmmserver*>*MSSCVMMLibrary**VHDs*.

2. Refresh the library server. Open the **Library** workspace, expand **Library Servers**, right-click on the library server that you want to refresh, and click **Refresh**.

3. Next, provide VMM with information about the operating system installed on the template disk:

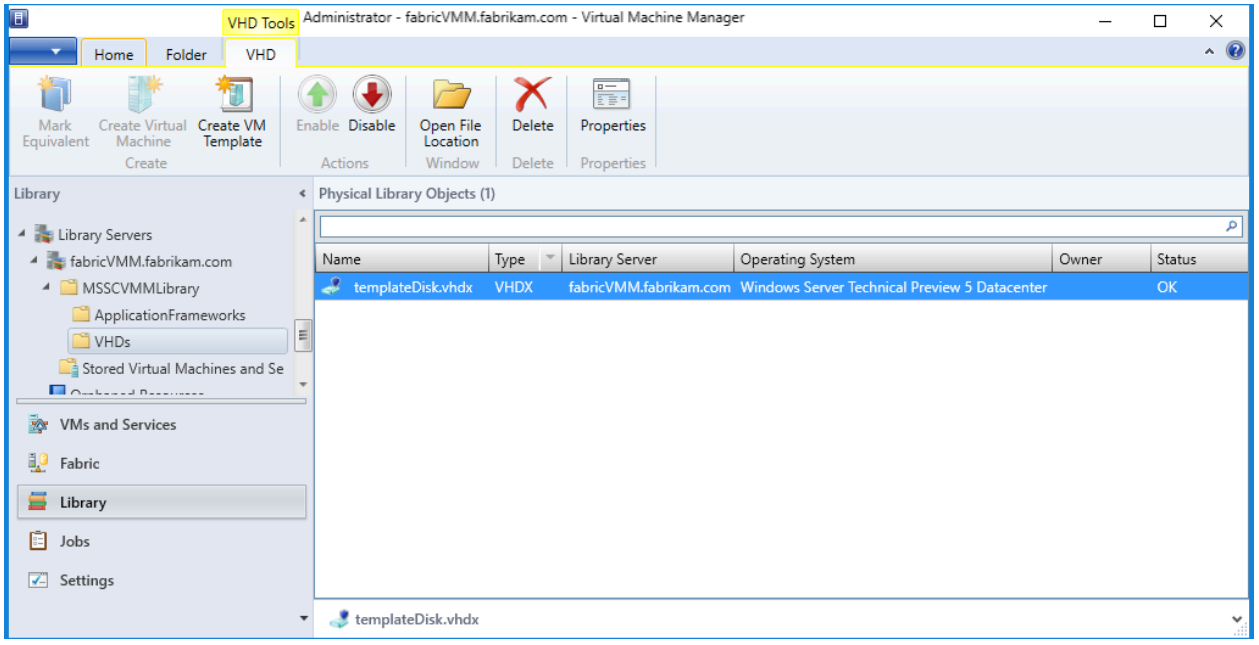
a. Find your newly imported template disk on your library server in the **Library** workspace.

b. Right-click the disk and then click **Properties**.

c. For **operating system**, expand the list and select the operating system installed on the disk. Selecting an operating system indicates to VMM that the VHDX is not blank.

d. When you have updated the properties, click **OK**.

The small shield icon next to the disk's name denotes the disk as a prepared template disk for shielded VMs. You can also right click the column headers and toggle the **Shielded** column to see a textual representation indicating whether a disk is intended for regular or shielded VM deployments.



Create the shielded VM template in VMM using the prepared template disk

With a prepared template disk in your VMM library, you are ready to create a VM template for shielded VMs. VM templates for shielded VMs differ slightly from traditional VM templates in that certain settings are fixed (generation 2 VM, UEFI and Secure Boot enabled, and so on) and others are unavailable (tenant customization is limited to a few, select properties of the VM). To create the VM template, perform the following steps:

1. In the **Library** workspace, click **Create VM Template** on the home tab at the top.

2. On the **Select Source** page, click **Use an existing VM template or a virtual hard disk stored in the library**, and then click **Browse**.

3. In the window that appears, select a prepared template disk from the VMM library. To more easily identify which disks are prepared, right-click a column header and enable the **Shielded** column. Click **OK** then **Next**.

4. Specify a VM template name and optionally a description, and then click **Next**.

5. On the **Configure Hardware** page, specify the capabilities of VMs created from this template. Ensure that at least one NIC is available and configured on the VM template. The only way for a tenant to connect to a shielded VM is through Remote Desktop Connection, Windows Remote Management, or other pre-configured remote management tools that work over networking protocols.

   If you choose to leverage static IP pools in VMM instead of running a DHCP server on the tenant network, you will need to alert your tenants to this configuration. When a tenant supplies their shielding data file, which contains the unattend file for the VMM, they will need to provide special placeholder values for the static IP pool information. For more information about VMM placeholders in tenant unattend files, see Create an answer file.

6. On the **Configure Operating System** page, VMM will only show a few options for shielded VMs, including the product key, time zone, and computer name. Some secure information, such as the administrator password and domain name, is specified by the tenant through a shielding data file (.PDK file).

   > ⓘ **Note**
   >
   > If you choose to specify a product key on this page, ensure it is valid for the operating system on the template disk. If an incorrect product key is used, the VM creation will fail.

After the template is created, tenants can use it to create new virtual machines. You will need to verify that the VM template is one of the resources available to the Tenant Administrator user role (in VMM, user roles are in the **Settings** workspace).

Prepare and protect the VHDX using PowerShell

As an alternative to running the Template Disk Wizard, you can copy your template disk and certificate to a computer running RSAT and run Protect-TemplateDisk to initiate the signing process. The following example uses the name and version information specified by the *TemplateName* and *Version* parameters. The VHDX you provide to the `-Path` parameter will be overwritten with the updated template disk, so be sure to make a copy before running the command.

PowerShell

```
# Replace "THUMBPRINT" with the thumbprint of your template disk signing
certificate in the line below
$certificate = Get-Item Cert:\LocalMachine\My\THUMBPRINT

Protect-TemplateDisk -Certificate $certificate -Path "WindowsServer2019-
ShieldedTemplate.vhdx" -TemplateName "Windows Server 2019" -Version 1.0.0.0
```

Your template disk is now ready to be used to provision shielded VMs. If you are using System Center Virtual Machine Manager to deploy your VM, you can now copy the VHDX to your VMM library.

You may also want to extract the volume signature catalog from the VHDX. This file is used to provide information about the signing certificate, disk name, and version to VM owners who want to use your template. They need to import this file into the Shielding Data File Wizard to authorize you, the template author in possession of the signing certificate, to create this and future template disks for them.

To extract the volume signature catalog, run the following command in PowerShell:

PowerShell

```
Save-VolumeSignatureCatalog -TemplateDiskPath 'C:\temp\MyLinuxTemplate.vhdx'
-VolumeSignatureCatalogPath 'C:\temp\MyLinuxTemplate.vsc'
```

Next step

Create a shielding data file

Additional References

- Hosting service provider configuration steps for guarded hosts and shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Create a Linux shielded VM template disk

Article • 09/21/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic explains how to prepare a template disk for Linux shielded VMs that can be used to instantiate one or more tenant VMs.

Prerequisites

To prepare and test a Linux shielded VM, you will need the following resources available:

- A server with virtualization capabilities running Windows Server, version 1709 or later
- A second computer (Windows 10 or Windows Server 2016) capable of running Hyper-V Manager to connect to the running VM's console
- An ISO image for one of the supported Linux shielded VM OSes:
  - Ubuntu 16.04 LTS with the 4.4 kernel
  - Red Hat Enterprise Linux 7.3
  - SUSE Linux Enterprise Server 12 Service Pack 2
- Internet access to download the lsvmtools package and OS updates

> **ⓘ Important**
>
> Newer versions of the preceding Linux OSes may include a known TPM driver bug which will prevent them from successfully provisioning as shielded VMs. It is not recommended that you update your templates or shielded VMs to a newer release until a fix is available. The list of supported OSes above will be updated when the updates are made public.

Prepare a Linux VM

Shielded VMs are created from secure template disks. Template disks contain the operating system for the VM and metadata, including a digital signature of the /boot and /root partitions, to ensure core OS components are not modified before deployment.

To create a template disk, you must first create a regular (unshielded) VM that you will prepare as the base image for future shielded VMs. The software you install and configuration changes you make to this VM will apply to all shielded VMs created from this template disk. These steps will walk you through the bare minimum requirements to get a Linux VM ready for templatization.

> **Note**
>
> Linux disk encryption is configured when the disk is partitioned. This means that you must create a new VM that is pre-encrypted using dm-crypt to create a Linux shielded VM template disk.

1. On the virtualization server, ensure that Hyper-V and the Host Guardian Hyper-V Support features are installed by running the following commands in an elevated PowerShell console:

   ```PowerShell
   Install-WindowsFeature Hyper-V, HostGuardian -IncludeManagementTools -Restart
   ```

2. Download the ISO image from a trustworthy source and store it on your virtualization server, or on a file share accessible to your virtualization server.

3. On your management computer running Windows Server version 1709, install the Shielded VM Remote Server Administration Tools by running the following command:

   ```PowerShell
   Install-WindowsFeature RSAT-Shielded-VM-Tools
   ```

4. Open **Hyper-V Manager** on your management computer and connect to your virtualization server. You can do this by clicking "Connect to Server..." in the Actions pane or by right clicking on Hyper-V Manager and choosing "Connect to Server..." Provide the DNS name for your Hyper-V server and, if necessary, the credentials needed to connect to it.

5. Using Hyper-V Manager, configure an external switch on your virtualization server so the Linux VM can access the Internet to obtain updates.

6. Next, create a new virtual machine to install the Linux OS onto. In the Actions pane, click **New** > **Virtual Machine** to bring up the wizard. Provide a friendly name for your VM, such as "Pre-templatized Linux" and click **Next**.

7. On the second page of the Wizard, select **Generation 2** to ensure the VM is provisioned with a UEFI-based firmware profile.

8. Complete the rest of the wizard according to your preferences. Do not use a differencing disk for this VM; shielded VM template disks cannot use differencing disks. Lastly, connect the ISO image you downloaded earlier to the virtual DVD drive for this VM so that you can install the OS.

9. In Hyper-V Manager, select your newly-created VM and click **Connect...** in the Actions pane to attach to a virtual console of the VM. In the window that appears, click **Start** to turn on the virtual machine.

10. Proceed through the setup process for your selected Linux distribution. While each Linux distribution uses a different setup wizard, the following requirements must be met for VMs that will become Linux shielded VM template disks:

    - The disk must be partitioned using the GUID Partitioning Table (GPT) layout
    - The root partition must be encrypted with dm-crypt. The passphrase should be set to **passphrase** (all lowercase). This passphrase will be randomized and the partition re-encrypted when a shielded VM is provisioned.
    - The boot partition must use the **ext2** file system

11. Once your Linux OS has fully booted and you have signed in, it is recommended that you install the linux-virtual kernel and associated Hyper-V integration services packages. Additionally, you will want to install an SSH server or other remote management tool to access the VM once it is shielded.

    On Ubuntu, run the following command to install these components:

    ```
    Bash

    sudo apt-get install linux-virtual linux-tools-virtual linux-cloud-
    tools-virtual linux-image-extra-virtual openssh-server
    ```

    On RHEL, run the following command instead:

    ```
    Bash

    sudo yum install hyperv-daemons openssh-server
    sudo service sshd start
    ```

And on SLES, run the following command:

```
Bash

sudo zypper install hyper-v
sudo chkconfig hv_kvp_daemon on
sudo systemctl enable sshd
```

12. Configure your Linux OS as desired. Any software you install, user accounts you add, and systemwide configuration changes you make will apply to all future VMs created from this template disk. You should avoid saving any secrets or unnecessary packages to the disk.

13. If you are planning to use System Center Virtual Machine Manager to deploy your VMs, install the VMM guest agent to enable VMM to specialize your OS during VM provisioning. Specialization allows each VM to be set up securely with different users and SSH keys, networking configurations, and custom setup steps. Learn how to obtain and install the VMM guest agent in the VMM documentation.

14. Next, add the Microsoft Linux Software Repository to your package manager.

15. Using your package manager, install the lsvmtools package which contains the Linux shielded VM bootloader shim, provisioning components, and disk preparation tool.

```
Bash

# Ubuntu 16.04
sudo apt-get install lsvmtools

# SLES 12 SP2
sudo zypper install lsvmtools

# RHEL 7.3
sudo yum install lsvmtools
```

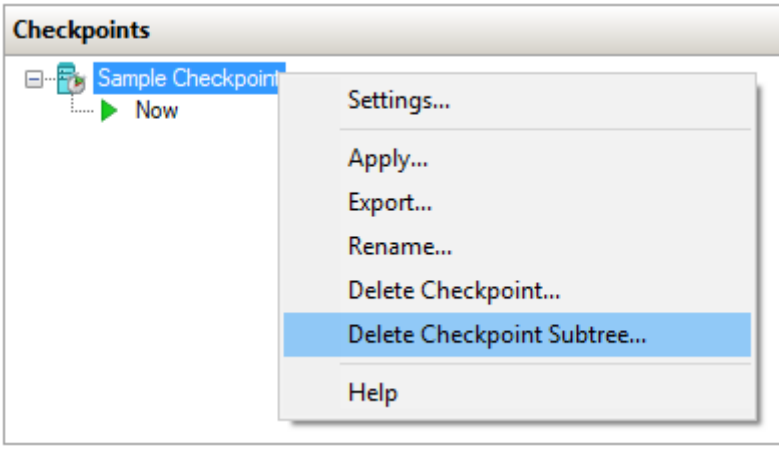
16. When you're done customizing the Linux OS, locate the lsvmprep installation program on your system and run it.

```
Bash

# The path below may change based on the version of lsvmprep installed
# Run "find /opt -name lsvmprep" to locate the lsvmprep executable
sudo /opt/lsvmtools-1.0.0-x86-64/lsvmprep
```

17. Shut down your VM.

18. If you took any checkpoints of your VM (including automatic checkpoints created by Hyper-V with the Windows 10 Fall Creators Update), be sure to delete them before continuing. Checkpoints create differencing disks (.avhdx) that are not supported by the Template Disk Wizard.

    To delete checkpoints, open **Hyper-V Manager**, select your VM, right click the topmost checkpoint in the Checkpoints pane, then click **Delete Checkpoint Subtree**.



Protect the template disk

The VM you prepared in the previous section is almost ready to be used as a Linux shielded VM template disk. The last step is to run the disk through the Template Disk Wizard, which will hash and digitally sign the current state of the root and boot partitions. The hash and digital signature are verified when a shielded VM is provisioned to ensure that no unauthorized changes were made to the two partitions in between template creation and deployment.

Obtain a certificate to sign the disk

In order to digitally sign the disk measurements, you will need to obtain a certificate on the computer where you will run the Template Disk Wizard. The certificate must meet the following requirements:

Expand table

Certificate Property	Required value
Key Algorithm	RSA
Minimum key size	2048 bits

Certificate Property	Required value
Signature algorithm	SHA256 (Recommended)
Key Usage	Digital Signature

Details about this certificate will be shown to tenants when they create their shielding data files and are authorizing disks they trust. Therefore, it is important to obtain this certificate from a certificate authority mutually trusted by you and your tenants. In enterprise scenarios where you are both the hoster and tenant, you might consider issuing this certificate from your enterprise certificate authority. Protect this certificate carefully, as anyone in possession of this certificate can create new template disks that are trusted the same as your authentic disk.

In a test lab environment, you can create a self-signed certificate with the following PowerShell command:

PowerShell

```powershell
New-SelfSignedCertificate -Subject "CN=Linux Shielded VM Template Disk Signing Certificate"
```

Process the disk with the Template Disk Wizard cmdlet

Copy your template disk and certificate to a computer running Windows Server, version 1709, then run the following commands to initiate the signing process. The VHDX you provide to the `-Path` parameter will be overwritten with the updated template disk, so be sure to make a copy before running the command.

> **ⓘ Important**
>
> The Remote Server Administration Tools available on Windows Server 2016 or Windows 10 cannot be used to prepare a Linux shielded VM template disk. Only use the Protect-TemplateDisk cmdlet available on Windows Server, version 1709 or the Remote Server Administration Tools available on Windows Server 2019 to prepare a Linux shielded VM template disk.

PowerShell

```powershell
# Replace "THUMBPRINT" with the thumbprint of your template disk signing certificate in the line below
$certificate = Get-Item Cert:\LocalMachine\My\THUMBPRINT
```

```
Protect-TemplateDisk -Path 'C:\temp\MyLinuxTemplate.vhdx' -TemplateName
'Ubuntu 16.04' -Version 1.0.0.0 -Certificate $certificate -
ProtectedTemplateTargetDiskType PreprocessedLinux
```

Your template disk is now ready to be used to provision Linux shielded VMs. If you are using System Center Virtual Machine Manager to deploy your VM, you can now copy the VHDX to your VMM library.

You may also want to extract the volume signature catalog from the VHDX. This file is used to provide information about the signing certificate, disk name, and version to VM owners who want to use your template. They need to import this file into the Shielding Data File Wizard to authorize you, the template author in possession of the signing certificate, to create this and future template disks for them.

To extract the volume signature catalog, run the following command in PowerShell:

PowerShell

```
Save-VolumeSignatureCatalog -TemplateDiskPath 'C:\temp\MyLinuxTemplate.vhdx'
-VolumeSignatureCatalogPath 'C:\temp\MyLinuxTemplate.vsc'
```

Feedback

Was this page helpful? Yes No

Shielded VMs - Hosting service provider sets up Windows Azure Pack

Article • 06/09/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic describes how a hosting service provider can configure Windows Azure Pack so that tenants can use it to deploy shielded VMs. Windows Azure Pack is a web portal that extends the functionality of System Center Virtual Machine Manager to allow tenants to deploy and manage their own VMs through a simple web interface. Windows Azure Pack fully supports shielded VMs and makes it even easier for your tenants to create and manage their shielding data files.

To understand how this topic fits in the overall process of deploying shielded VMs, see Hosting service provider configuration steps for guarded hosts and shielded VMs.

Setting up Windows Azure Pack

You will complete the following tasks to set up Windows Azure Pack in your environment:

1. Complete configuration of System Center 2016 - Virtual Machine Manager (VMM) for your hosting fabric. This includes setting up VM templates and a VM cloud, which will be exposed through Windows Azure Pack:

   Scenario - Deploy guarded hosts and shielded virtual machines in VMM

2. Install and configure System Center 2016 - Service Provider Foundation (SPF). This software enables Windows Azure Pack to communicate with your VMM servers:

   Deploying Service Provider Foundation - SPF

3. Install Windows Azure Pack and configure it to communicate with SPF:

   - Install Windows Azure Pack (in this topic)
   - Configure Windows Azure Pack (in this topic)

4. Create one or more hosting plans in Windows Azure Pack to allow tenants access to your VM clouds:

   Create a plan in Windows Azure Pack (in this topic)

Install Windows Azure Pack

Install and configure Windows Azure Pack (WAP) on the machine where you wish to host the web portal for your tenants. This machine will need to be able to reach the SPF server and be reachable by your tenants.

1. Reviewing WAP system requirements and install the prerequisite software.
2. Download and install the Web Platform Installer. If the machine is not connected to the Internet, follow the offline installation instructions.
3. Open the Web Platform Installer and find **Windows Azure Pack: Portal and API Express** under the **Products** tab. Click **Add**, then **Install** at the bottom of the window.
4. Proceed through the installation. After the installation completes, the configuration site (*https://<wapserver>:30101/*) opens in your web browser. On this website, provide information about your SQL server and finish configuring WAP.

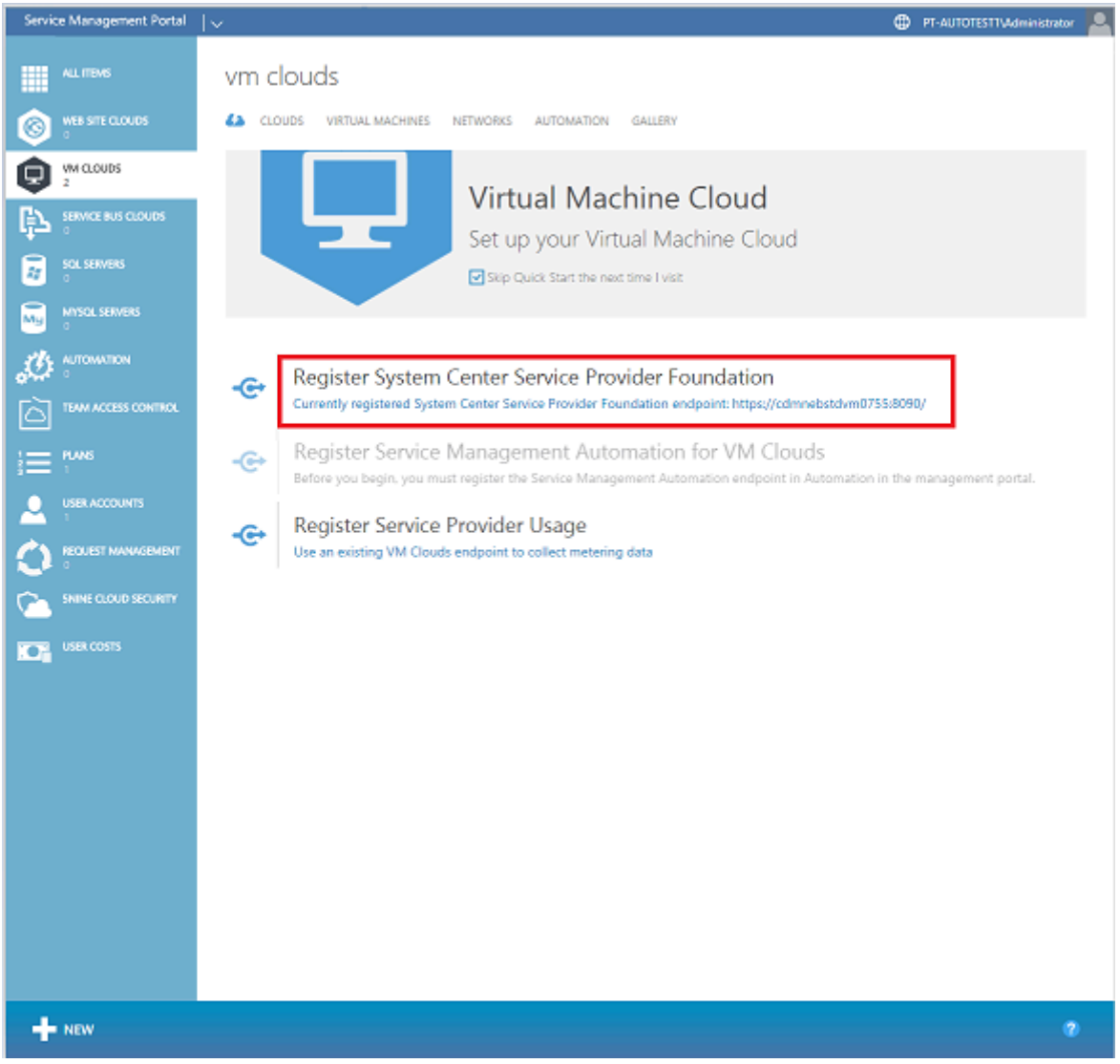
For help setting up Windows Azure Pack, see Install an express deployment of Windows Azure Pack.

> **Note**
>
> If you already run Windows Azure Pack in your environment, you may use your existing installation. In order to work with the latest shielded VM features, however, you will need to upgrade your installation to at least Update Rollup 10.

Configure Windows Azure Pack

Before you use Windows Azure Pack, you should already have it installed and configured for your infrastructure.

1. Navigate to the Windows Azure Pack admin portal at *https://<wapserver>:30091*, and then log in using your administrator credentials.
2. In the left pane, click **VM Clouds**.
3. Connect Windows Azure Pack to the Service Provider Foundation instance by clicking **Register System Center Service Provider Foundation**. You will need to specify the URL for Service Provider Foundation, as well as a username and password.



4. Once completed, you should be able to see the VM clouds set up in your VMM environment. Ensure you have at least one VM cloud that supports shielded VMs available to WAP before continuing.

Create a plan in Windows Azure Pack

In order to allow tenants to create VMs in WAP, you must first create a hosting plan to which tenants can subscribe. Plans define the allowed VM clouds, templates, networks, and billing entities for your tenants.

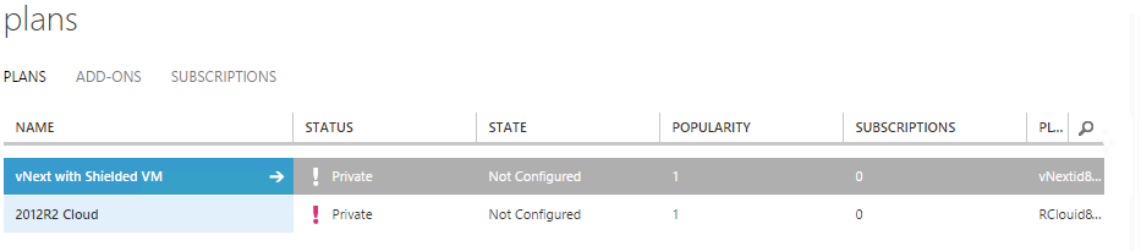
1. On the lower pane of the portal, click **+NEW** > **PLAN** > **CREATE PLAN**.

2. In the first step of the wizard, choose a name for your Plan. This is the name your tenants will see when subscribing.

3. In the second step, select **VIRTUAL MACHINE CLOUDS** as one of the services to offer in the plan.

4. Skip the step about selecting any add-ons for the plan.

5. Click **OK** (check mark) to create the plan. Although this creates the plan, it is not yet in a configured state.

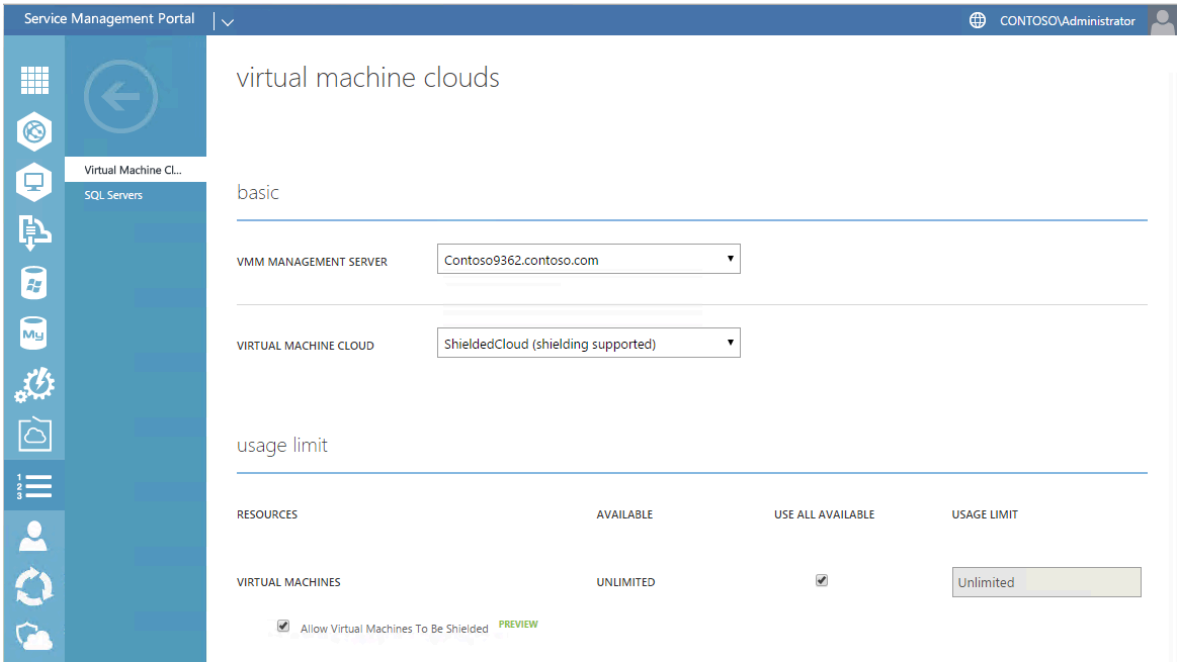


6. To begin configuring the Plan, click its name.

7. On the next page, under **plan services**, click **Virtual Machine Clouds**. This opens the page where you can configure quotas for this plan.

8. Under **basic**, select the VMM Management Server and Virtual Machine Cloud you wish to offer to your tenants. Clouds that can offer shielded VMs will be displayed with **(shielding supported)** next to their name.

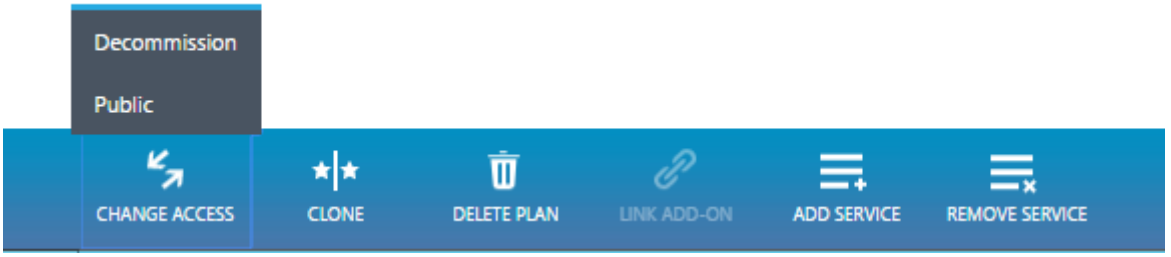
9. Select the quotas you want to apply in this Plan. (For example, limits on CPU core and RAM usage). Make sure to leave the **Allow Virtual Machines To Be Shielded** checkbox selected.



10. Scroll down to the section titled **templates**, and then select one or more templates to offer to your tenants. You can offer both shielded and unshielded templates to tenants, but a shielded template must be offered to give tenants end-to-end assurances about the integrity of the VM and their secrets.

11. In the **networks** section, add one or more networks for your tenants.

12. After setting any other settings or quotas for the Plan, click **Save** at the bottom.
13. At the top left of the screen, click on the arrow to take you back to the **Plan** page.
14. At the bottom of the screen, change the Plan from being **Private** to **Public** so that tenants can subscribe to the Plan.



At this point, Windows Azure Pack is configured and tenants will be able to subscribe to the plan you just created and deploy shielded VMs. For additional steps that tenants need to complete, see Shielded VMs for tenants - Deploying a shielded VM by using Windows Azure Pack.

Additional References

- Hosting service provider configuration steps for guarded hosts and shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Create OS specialization answer file

Article • 09/21/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

In preparation to deploy shielded VMs, you may need to create an operating system specialization answer file. On Windows, this is commonly known as the "unattend.xml" file. The **New-ShieldingDataAnswerFile** Windows PowerShell function helps you do this. You can then use the answer file when you're creating shielded VMs from a template by using System Center Virtual Machine Manager (or any other fabric controller).

For general guidelines for Unattend files for shielded VMs, see Create an answer file.

Downloading the New-ShieldingDataAnswerFile function

You can obtain the **New-ShieldingDataAnswerFile** function from the PowerShell Gallery. If your computer has Internet connectivity, you can install it from PowerShell with the following command:

PowerShell

```
Install-Module GuardedFabricTools -Repository PSGallery -MinimumVersion
1.0.0
```

The `unattend.xml` output can be packaged into the shielding data, along with additional artifacts, so that it can be used to create shielded VMs from templates.

The following sections show how you can use the function parameters for an `unattend.xml` file containing various options:

- Basic Windows Answer File
- Windows answer file with domain join
- Windows answer file with static IPv4 addresses
- Windows answer file with a custom locale
- Basic Linux answer file

Basic Windows answer file

The following commands create a Windows answer file that simply sets the administrator account password and hostname. The VM network adapters will use DHCP to obtain IP addresses, and the VM will not be joined to an Active Directory domain. When prompted to enter an administrator credential, specify the desired username and password. Use "Administrator" for the username if you wish to configure the built-in Administrator account.

PowerShell

```
$adminCred = Get-Credential -Message "Local administrator account"

New-ShieldingDataAnswerFile -Path '.\ShieldedVMAnswerFile.xml' -
AdminCredentials $adminCred
```

Windows answer file with domain join

The following commands create a Windows answer file that joins the shielded VM to an Active Directory domain. The VM network adapters will use DHCP to obtain IP addresses.

The first credential prompt will ask for the local administrator account information. Use "Administrator" for the username if you wish to configure the built-in Administrator account.

The second credential prompt will ask for credentials that have the right to join the machine to the Active Directory domain.

Be sure to change the value of the "-DomainName" parameter to the FQDN of your Active Directory domain.

PowerShell

```
$adminCred = Get-Credential -Message "Local administrator account"
$domainCred = Get-Credential -Message "Domain join credentials"

New-ShieldingDataAnswerFile -Path '.\ShieldedVMAnswerFile.xml' -
AdminCredentials $adminCred -DomainName 'my.contoso.com' -
DomainJoinCredentials $domainCred
```

Windows answer file with static IPv4 addresses

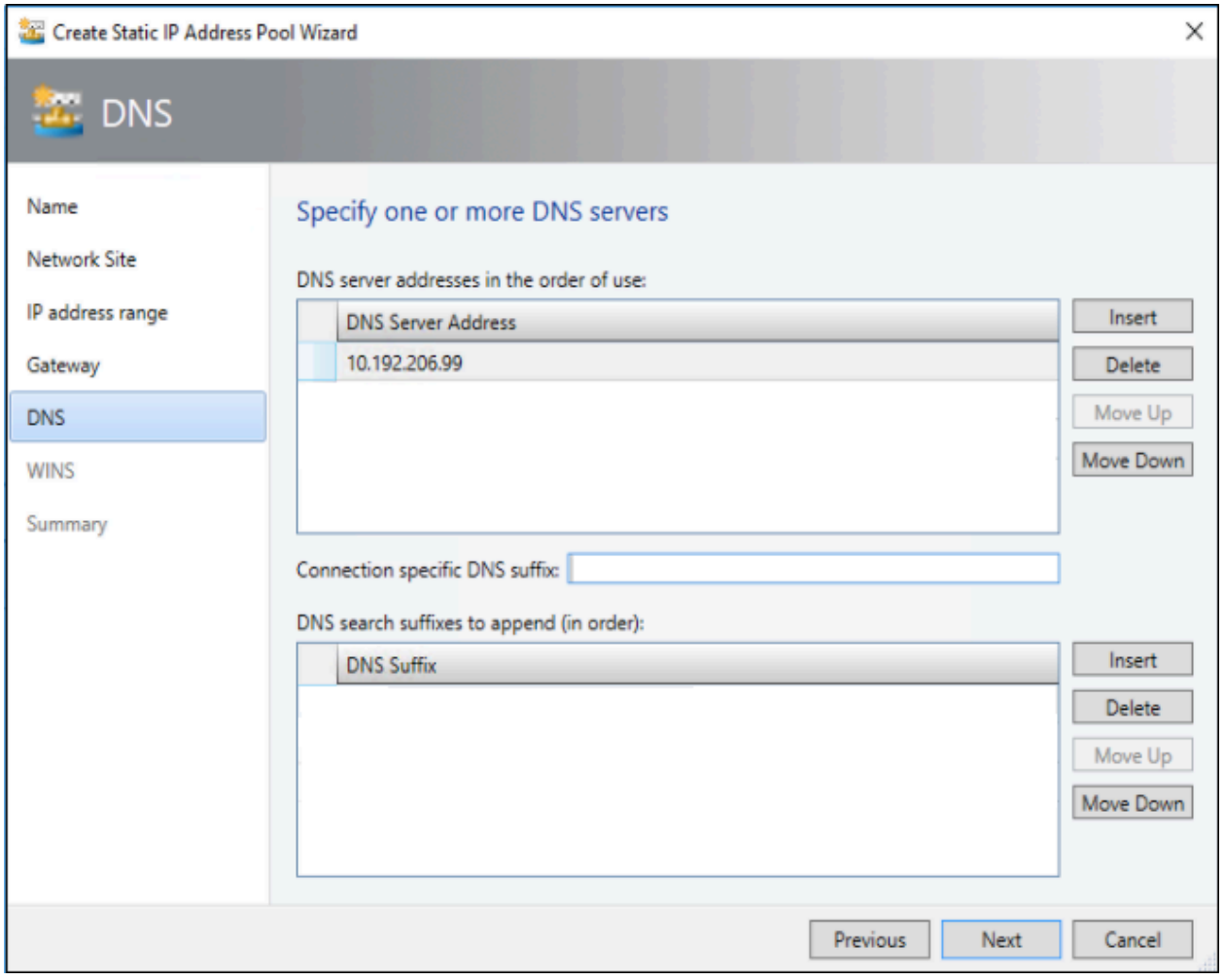
The following commands create a Windows answer file that uses static IP addresses provided at deployment time by the fabric manager, such as System Center Virtual

Machine Manager.

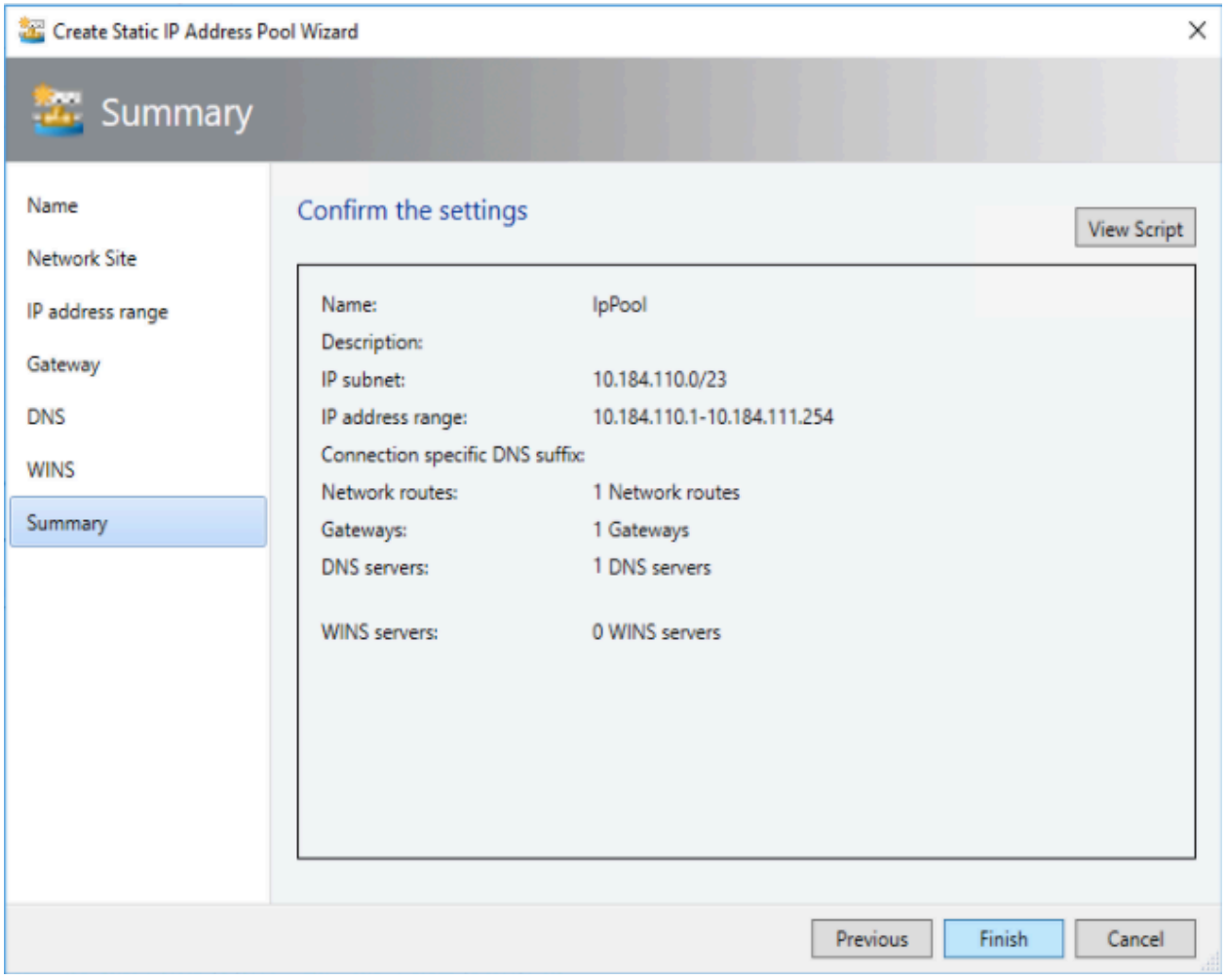
Virtual Machine Manager provides three components to the static IP address by using an IP pool: IPv4 address, IPv6 address, gateway address, and DNS address. If you want any additional fields to be included or require a custom network configuration, you will need to manually edit the answer file produced by the script.

The following screenshots show the IP pools that you can configure in Virtual Machine Manager. These pools are necessary if you want to use static IP.

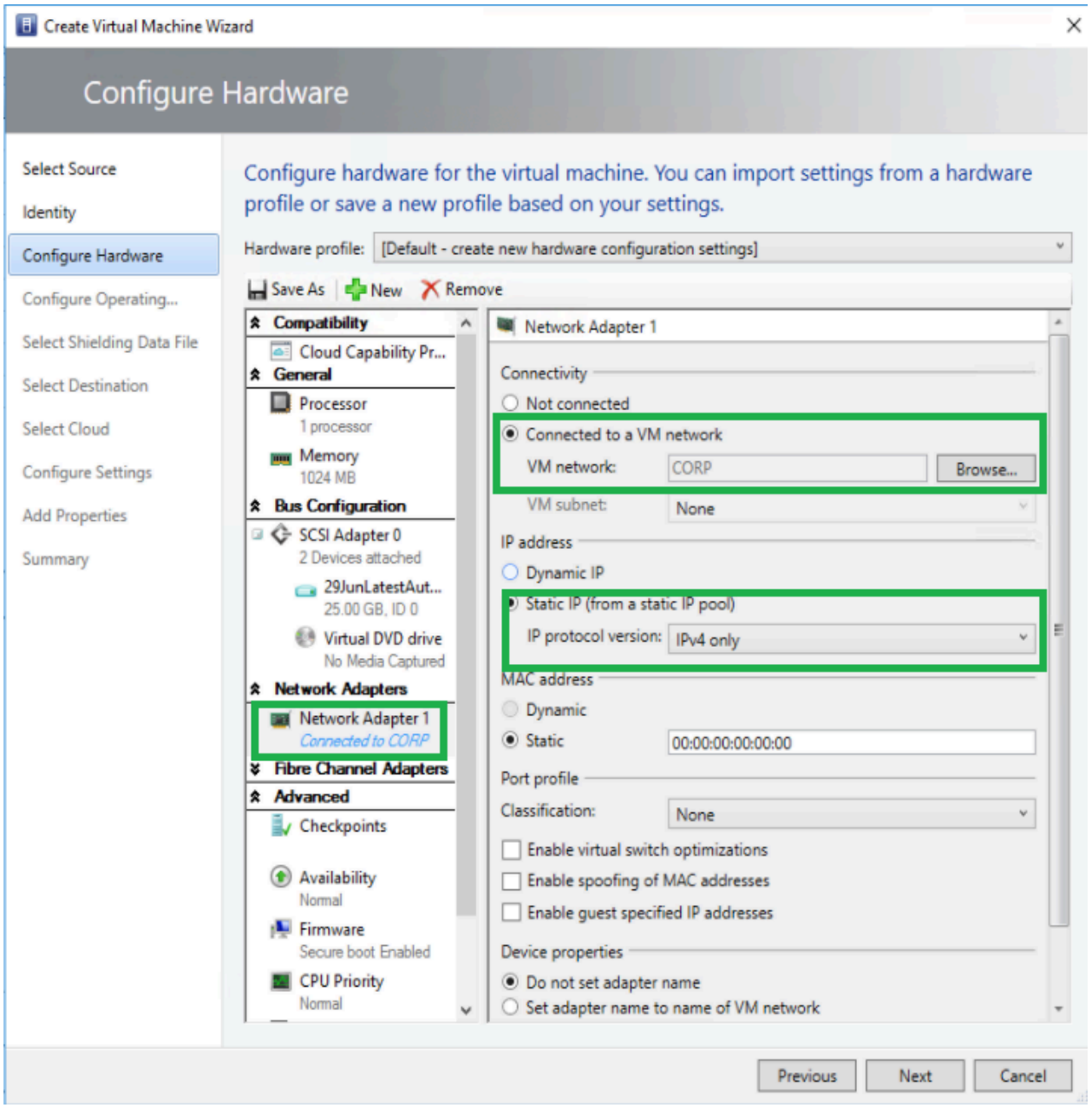
Currently, the function supports only one DNS server. Here is what your DNS settings would look like:



Here is what your summary for creating the static IP address pool would look like. In short, you must have only one network route, one gateway, and one DNS server - and you must specify your IP address.



You need to configure your network adapter for your virtual machine. The following screenshot shows where to set that configuration and how to switch it to static IP.



Then, you can use the `-StaticIPPool` parameter to include the static IP elements in the answer file. The parameters `@IPAddr-1@`, `@NextHop-1-1@`, and `@DNSAddr-1-1@` in the answer file will then be replaced with the real values that you specified in Virtual Machine Manager at deployment time.

PowerShell

```
$adminCred = Get-Credential -Message "Local administrator account"

New-ShieldingDataAnswerFile -Path '.\ShieldedVMAnswerFile.xml' -
AdminCredentials $adminCred -StaticIPPool IPv4Address
```

Windows answer file with a custom locale

The following commands create a Windows answer file with a custom locale.

When prompted to enter an administrator credential, specify the desired username and password. Use "Administrator" for the username if you wish to configure the built-in Administrator account.

PowerShell

```
$adminCred = Get-Credential -Message "Local administrator account"
$domainCred = Get-Credential -Message "Domain join credentials"

New-ShieldingDataAnswerFile -Path '.\ShieldedVMAnswerFile.xml' -
AdminCredentials $adminCred -Locale es-ES
```

Basic Linux answer file

Starting with Windows Server version 1709, you can run certain Linux guest OSes in shielded VMs. If you are using the System Center Virtual Machine Manager Linux agent to specialize those VMs, the New-ShieldingDataAnswerFile cmdlet can create compatible answer files for it.

In a Linux answer file, you will typically include the root password, root SSH key, and optionally static IP pool information. Replace the path to the public half of your SSH key before running the script below.

PowerShell

```
$rootPassword = Read-Host -Prompt "Root password" -AsSecureString

New-ShieldingDataAnswerFile -Path '.\ShieldedVMAnswerFile.xml' -RootPassword
$rootPassword -RootSshKey '~\.ssh\id_rsa.pub'
```

Additional References

- Deploy shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Shielded VMs for tenants - Creating shielding data to define a shielded VM

Article • 06/09/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

A shielding data file (also called a provisioning data file or PDK file) is an encrypted file that a tenant or VM owner creates to protect important VM configuration information, such as the administrator password, RDP and other identity-related certificates, domain-join credentials, and so on. This topic provides information about how to create a shielding data file. Before you can create the file, you must either obtain a template disk from your hosting service provider, or create a template disk as described in Shielded VMs for tenants - Creating a template disk (optional).

For a list and a diagram of the contents of a shielding data file, see What is shielding data and why is it necessary?.

> ⓘ **Important**
>
> The steps in this section should be completed on a separate, trusted machine outside of the guarded fabric. Typically, the VM owner (tenant) would create the shielding data for their VMs, not the fabric administrators.

To prepare to create a shielding data file, take the following steps:

- Obtain a certificate for Remote Desktop Connection
- Create an answer file
- Get the volume signature catalog file
- Select trusted fabrics

Then you can create the shielding data file:

- Create a shielding data file and add guardians

(Optional) Obtain a certificate for Remote Desktop Connection

Since tenants are only able to connect to their shielded VMs using Remote Desktop Connection or other remote management tools, it is important to ensure that tenants

can verify they are connecting to the right endpoint (that is, there is not a "man in the middle" intercepting the connection).

One way to verify you are connecting to the intended server is to install and configure a certificate for Remote Desktop Services to present when you initiate a connection. The client machine connecting to the server will check whether it trusts the certificate and show a warning if it does not. Generally, to ensure the connecting client trusts the certificate, RDP certificates are issued from the tenant's PKI. More information about Using certificates in Remote Desktop Services can be found on TechNet.

To help you decide if you need to obtain a custom RDP certificate, consider the following:

- If you're just testing shielded VMs in a lab environment, you **do not** need a custom RDP certificate.
- If your VM is configured to join an Active Directory domain, a computer certificate will typically be issued by your organization's certificate authority automatically and used to identify the computer during RDP connections. You **do not** need a custom RDP certificate.
- If your VM is not domain joined but you want a way to verify you're connecting to the correct machine when you use Remote Desktop, you **should consider** using custom RDP certificates.

> **Tip**
>
> When selecting an RDP certificate to include in your shielding data file, be sure to use a wildcard certificate. One shielding data file may be used to create an unlimited number of VMs. Since each VM will share the same certificate, a wildcard certificate ensures the certificate will be valid regardless of the VM's hostname.

Create an answer file

Since the signed template disk in VMM is generalized, tenants are required to provide an answer file to specialize their shielded VMs during the provisioning process. The answer file (often called the unattend file) can configure the VM for its intended role - that is, it can install Windows features, register the RDP certificate created in the previous step, and perform other custom actions. It will also supply required information for Windows setup, including the default administrator's password and product key.

For information about obtaining and using the **New-ShieldingDataAnswerFile** function to generate an answer file (Unattend.xml file) for creating shielded VMs, see Generate an

answer file by using the New-ShieldingDataAnswerFile function. Using the function, you can more easily generate an answer file that reflects choices such as the following:

- Is the VM intended to be domain joined at the end of the initialization process?
- Will you be using a volume license or specific product key per VM?
- Are you using DHCP or static IP?
- Will you use a custom Remote Desktop Protocol (RDP) certificate that will be used to prove that the VM belongs to your organization?
- Do you want to run a script at the end of the initialization?

Answer files used in shielding data files will be used on every VM created using that shielding data file. Therefore, you should make sure that you do not hard code any VM-specific information into the answer file. VMM supports some substitution strings (see the table below) in the unattend file to handle specialization values that may change from VM to VM. You are not required to use these; however, if they are present VMM will take advantage of them.

When creating an unattend.xml file for shielded VMs, keep in mind the following restrictions:

- If you're using VMM to manage your datacenter, the unattend file must result in the VM being turned off after it has been configured. This is to allow VMM to know when it should report to the tenant that the VM finished provisioning and is ready for use. VMM will automatically power the VM back on once it detects it has been turned off during provisioning.

- Be sure to enable RDP and the corresponding firewall rule so you can access the VM after it has been configured. You cannot use the VMM console to access shielded VMs, so you will need RDP to connect to your VM. If you prefer to manage your systems with Windows PowerShell remoting, ensure WinRM is enabled, too.

- The only substitution strings supported in shielded VM unattend files are the following:

Expand table

Replaceable Element	Substitution String
ComputerName	@ComputerName@
TimeZone	@TimeZone@
ProductKey	@ProductKey@

Replaceable Element	Substitution String
IPAddr4-1	@IP4Addr-1@
IPAddr6-1	@IP6Addr-1@
MACAddr-1	@MACAddr-1@
Prefix-1-1	@Prefix-1-1@
NextHop-1-1	@NextHop-1-1@
Prefix-1-2	@Prefix-1-2@
NextHop-1-2	@NextHop-1-2@

If you have more than one NIC, you can add multiple substitution strings for the IP configuration by incrementing the first digit. For example, to set the IPv4 address, subnet, and gateway for 2 NICs, you would use the following substitution strings:

Expand table

Substitution String	Example substitution
@IP4Addr-1@	192.168.1.10/24
@MACAddr-1@	Ethernet
@Prefix-1-1@	24
@NextHop-1-1@	192.168.1.254
@IP4Addr-2@	10.0.20.30/24
@MACAddr-2@	Ethernet 2
@Prefix-2-1@	24
@NextHop-2-1@	10.0.20.1

When using substitution strings, it is important to ensure that the strings will be populated during the VM provisioning process. If a string such as @ProductKey@ is not supplied at deployment time, leaving the <ProductKey> node in the unattend file blank, the specialization process will fail and you will be unable to connect to your VM.

Also, note that the networking-related substitution strings towards the end of the table are only used if you are leveraging VMM Static IP Address Pools. Your hosting service provider should be able to tell you if these substitution strings are required. For more

information about static IP addresses in VMM templates, see the following in the VMM documentation:

- Guidelines for IP address pools
- Set up static IP address pools in the VMM fabric

Finally, it is important to note that the shielded VM deployment process will only encrypt the OS drive. If you deploy a shielded VM with one or more data drives, it is strongly recommended that you add an unattend command or Group Policy setting in the tenant domain to automatically encrypt the data drives.

Get the volume signature catalog file

Shielding data files also contain information about the template disks a tenant trusts. Tenants acquire the disk signatures from trusted template disks in the form of a volume signature catalog (VSC) file. These signatures are then validated when a new VM is deployed. If none of the signatures in the shielding data file match the template disk trying to be deployed with the VM (i.e. it was modified or swapped with a different, potentially malicious disk), the provisioning process will fail.

> **ⓘ Important**
>
> While the VSC ensures that a disk has not been tampered with, it is still important for the tenant to trust the disk in the first place. If you are the tenant and the template disk is provided by your hoster, deploy a test VM using that template disk and run your own tools (antivirus, vulnerability scanners, and so on) to validate the disk is, in fact, in a state that you trust.

There are two ways to acquire the VSC of a template disk:

1. The hoster (or tenant, if the tenant has access to VMM) uses the VMM PowerShell cmdlets to save the VSC and gives it to the tenant. This can be performed on any machine with the VMM console installed and configured to manage the hosting fabric's VMM environment. The PowerShell cmdlets to save the VSC are:

   PowerShell

   ```
   $disk = Get-SCVirtualHardDisk -Name "templateDisk.vhdx"

   $vsc = Get-SCVolumeSignatureCatalog -VirtualHardDisk $disk

   $vsc.WriteToFile(".\templateDisk.vsc")
   ```

2. The tenant has access to the template disk file. This may be the case if the tenant creates a template disk to uploaded to a hosting service provider or if the tenant can download the hoster's template disk. In this case, without VMM in the picture, the tenant would run the following cmdlet (installed with the Shielded VM Tools feature, part of Remote Server Administration Tools):

```PowerShell
Save-VolumeSignatureCatalog -TemplateDiskPath templateDisk.vhdx -VolumeSignatureCatalogPath templateDisk.vsc
```

Select trusted fabrics

The last component in the shielding data file relates to the owner and guardians of a VM. Guardians are used to designate both the owner of a shielded VM and the guarded fabrics on which it is authorized to run.

To authorize a hosting fabric to run a shielded VM, you must obtain the guardian metadata from the hosting service provider's Host Guardian Service. Often, the hosting service provider will provide you with this metadata through their management tools. In an enterprise scenario, you may have direct access to obtain the metadata yourself.

You or your hosting service provider can obtain the guardian metadata from HGS by performing one of the following actions:

- Obtain the guardian metadata directly from HGS by running the following Windows PowerShell command, or browsing to the website and saving the XML file that is displayed:

```PowerShell
Invoke-WebRequest 'http://hgs.bastion.local/KeyProtection/service/metadata/2014-07/metadata.xml' -OutFile .\RelecloudGuardian.xml
```

- Obtain the guardian metadata from VMM using the VMM PowerShell cmdlets:

```PowerShell
$relecloudmetadata = Get-SCGuardianConfiguration
$relecloudmetadata.InnerXml | Out-File .\RelecloudGuardian.xml -Encoding UTF8
```

Obtain the guardian metadata files for each guarded fabric you wish to authorize your shielded VMs to run on before continuing.

Create a shielding data file and add guardians using the Shielding Data File wizard

Run the Shielding Data File wizard to create a shielding data (PDK) file. Here, you'll add the RDP certificate, unattend file, volume signature catalogs, owner guardian and the downloaded guardian metadata obtained in the preceding step.

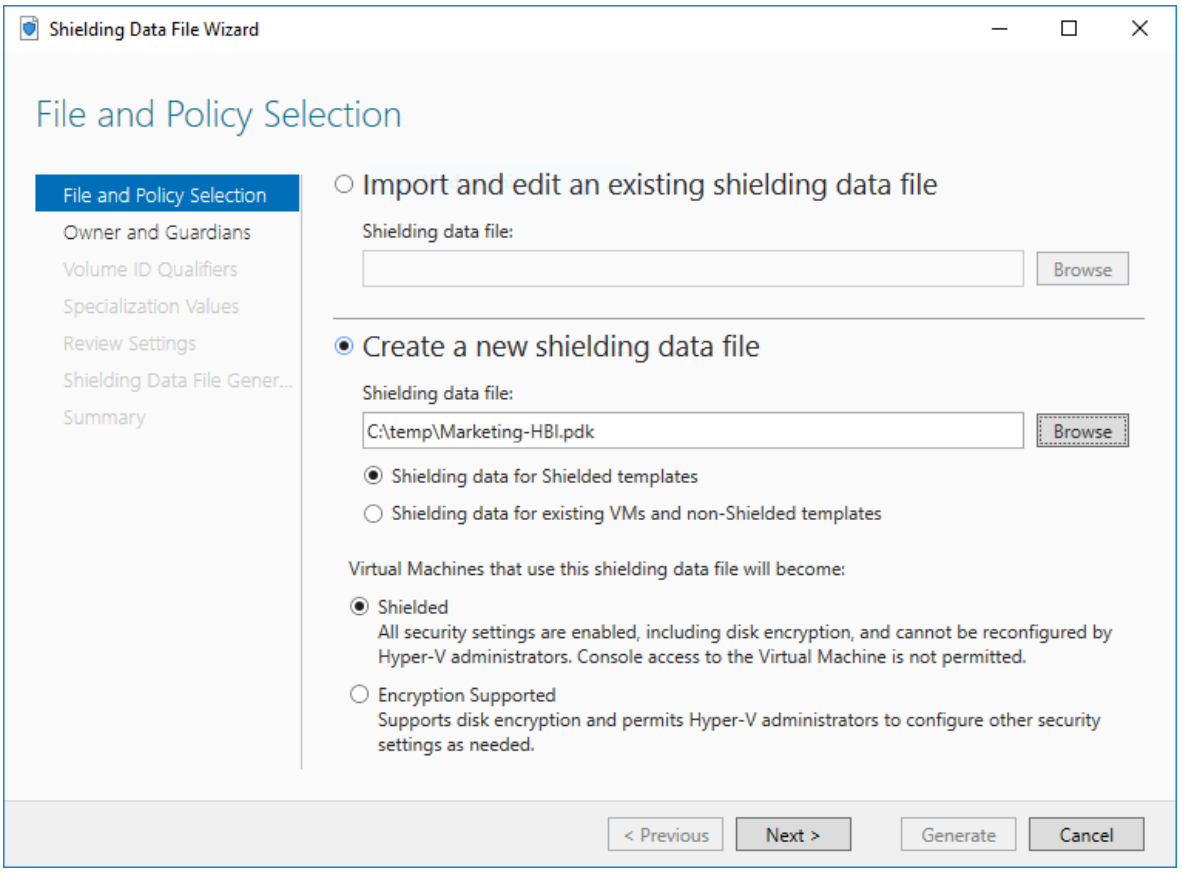
1. Install **Remote Server Administration Tools** > **Feature Administration Tools** > **Shielded VM Tools** on your machine using Server Manager or the following Windows PowerShell command:

   ```
   PowerShell

   Install-WindowsFeature RSAT-Shielded-VM-Tools
   ```

2. Open the Shielding Data File Wizard from the Administrator Tools section on your Start menu or by running the following executable **C:\Windows\System32\ShieldingDataFileWizard.exe**.

3. On the first page, use the second file selection box to choose a location and file name for your shielding data file. Normally, you would name a shielding data file after the entity who owns any VMs created with that shielding data (for example, HR, IT, Finance) and the workload role it is running (for example, file server, web server, or anything else configured by the unattend file). Leave the radio button set to **Shielding data for Shielded templates**.

   > ⓘ **Note**
   >
   > In the Shielding Data File Wizard you will notice the two options below:
   >
   > - **Shielding data for Shielded templates**
   > - **Shielding data for existing VMs and non-Shielded templates**
   >
   >   The first option is used when creating new shielded VMs from shielded templates. The second option allows you to create shielding data that can only be used when converting existing VMs or creating shielded VMs from non-shielded templates.



Additionally, you must choose whether VMs created using this shielding data file will be truly shielded or configured in "encryption supported" mode. For more information about these two options, see What are the types of virtual machines that a guarded fabric can run?.

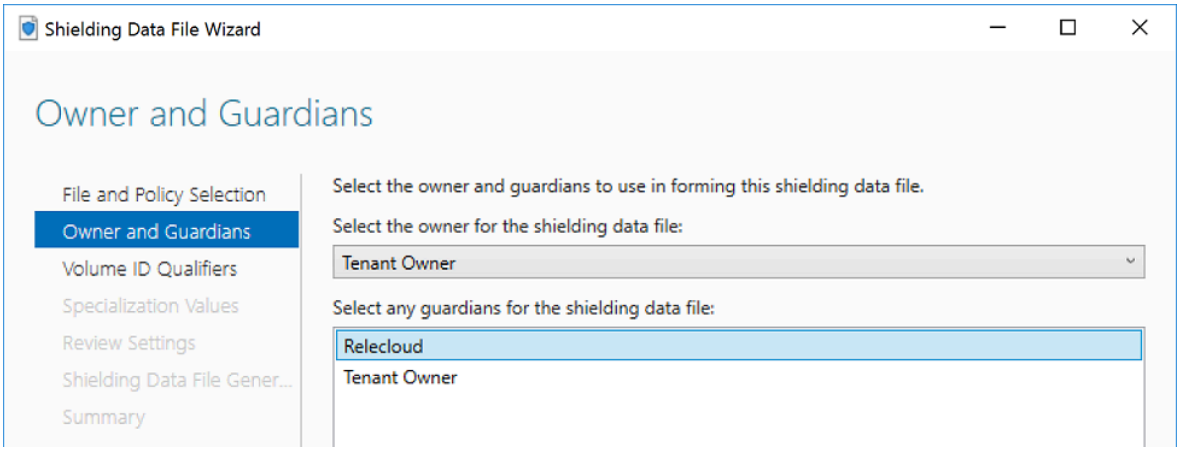
> **ⓘ Important**
>
> Pay careful attention to the next step as it defines the owner of your shielded VMs and which fabrics your shielded VMs will be authorized to run on. Possession of **owner guardian** is required in order to later change an existing shielded VM from **Shielded** to **Encryption Supported** or vice-versa.

4. Your goal in this step is two-fold:

   - Create or select an owner guardian that represents you as the VM owner
   - Import the guardian that you downloaded from the hosting provider's (or your own) Host Guardian Service in the preceding step

   To designate an existing owner guardian, select the appropriate guardian from the drop down menu. Only guardians installed on your local machine with the private keys intact will show up in this list. You can also create your own owner guardian by selecting **Manage Local Guardians** in the lower right corner and clicking **Create** and completing the wizard.

Next, we import the guardian metadata downloaded earlier again using the **Owner and Guardians** page. Select **Manage Local Guardians** from the lower right corner. Use the **Import** feature to import the guardian metadata file. Click **OK** once you have imported or added all of the necessary guardians. As a best practice, name guardians after the hosting service provider or enterprise datacenter they represent. Finally, select all the guardians that represent the datacenters in which your shielded VM is authorized to run. You do not need to select the owner guardian again. Click **Next** once finished.



5. On the Volume ID Qualifiers page, click **Add** to authorize a signed template disk in your shielding data file. When you select a VSC in the dialog box, it will show you information about that disk's name, version, and the certificate that was used to sign it. Repeat this process for each template disk you wish to authorize.

6. On the **Specialization Values** page, click **Browse** to select your unattend.xml file that will be used to specialize your VMs.

   Use the **Add** button at the bottom to add any additional files to the PDK that are needed during the specialization process. For example, if your unattend file is installing an RDP certificate onto the VM (as described in Generate an answer file by using the New-ShieldingDataAnswerFile function), you should add the RDP certificate PFX file and the RDPCertificateConfig.ps1 script here. Note that any files you specify here will automatically be copied to C:\temp\ on the VM that is created. Your unattend file should expect the files to be in that folder when referencing them by path.

7. Review your selections on the next page, and then click **Generate**.

8. Close the wizard after it has completed.

Create a shielding data file and add guardians using PowerShell

As an alternative to the Shielding Data File wizard, you can run New-ShieldingDataFile to create a shielding data file.

All shielding data files need to be configured with the correct owner and guardian certificates to authorize your shielded VMs to be run on a guarded fabric. You can check if you have any guardians installed locally by running Get-HgsGuardian. Owner guardians have private keys while guardians for your datacenter typically do not.

If you need to create an owner guardian, run the following command:

PowerShell

```
New-HgsGuardian -Name "Owner" -GenerateCertificates
```

This command creates a pair of signing and encryption certificates in the local machine's certificate store under the "Shielded VM Local Certificates" folder. You will need the owner certificates and their corresponding private keys to unshield a virtual machine, so ensure these certificates are backed up and protected from theft. An attacker with access to the owner certificates can use them to start up your shielded virtual machine or change its security configuration.

If you need to import guardian information from a guarded fabric where you want to run your virtual machine (your primary datacenter, backup datacenters, etc.), run the following command for each metadata file retrieved from your guarded fabrics.

PowerShell

```
Import-HgsGuardian -Name 'EAST-US Datacenter' -Path '.\EastUSGuardian.xml'
```

> **Tip**
>
> If you used self-signed certificates or the certificates registered with HGS are expired, you may need to use the `-AllowUntrustedRoot` and/or `-AllowExpired` flags with the Import-HgsGuardian command to bypass the security checks.

You will also need to obtain a volume signature catalog for each template disk you want to use with this shielding data file and a shielding data answer file to allow the operating system to complete its specialization tasks automatically. Lastly, decide if you want your VM to be fully shielded or just vTPM-enabled. Use `-Policy Shielded` for a fully shielded VM or `-Policy EncryptionSupported` for a vTPM enabled VM that allows basic console connections and PowerShell Direct.

Once everything is ready, run the following command to create your shielding data file:

PowerShell

```
$viq = New-VolumeIDQualifier -VolumeSignatureCatalogFilePath
'C:\temp\marketing-ws2016.vsc' -VersionRule Equals
New-ShieldingDataFile -ShieldingDataFilePath "C:\temp\Marketing-LBI.pdk" -
Policy EncryptionSupported -Owner 'Owner' -Guardian 'EAST-US Datacenter' -
VolumeIDQualifier $viq -AnswerFile 'C:\temp\marketing-ws2016-answerfile.xml'
```

> **Tip**
>
> If you are using a custom RDP certificate, SSH keys, or other files that need to be included with your shielding data file, use the `-OtherFile` parameter to include them. You can provide a comma separated list of file paths, like `-OtherFile "C:\source\myRDPCert.pfx", "C:\source\RDPCertificateConfig.ps1"`

In the above command, the guardian named "Owner" (obtained from Get-HgsGuardian) will be able to change the security configuration of the VM in the future, while 'EAST-US Datacenter' can run the VM but not change its settings. If you have more than one guardian, separate the names of the guardians with commas like `'EAST-US Datacenter', 'EMEA Datacenter'`. The volume ID qualifier specifies whether you trust only the exact version (Equals) of the template disk or future versions (GreaterThanOrEquals) as well. The disk name and signing certificate must match exactly for the version comparison to considered at deployment time. You can trust more than one template disk by providing a comma-separated list of volume ID qualifiers to the `-VolumeIDQualifier` parameter. Finally, if you have other files that need to accompany the answer file with the VM, use the `-OtherFile` parameter and provide a comma-separated list of file paths.

See the cmdlet documentation for New-ShieldingDataFile and New-VolumeIDQualifier to learn about additional ways to configure your shielding data file.

Additional References

- Deploy shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Create a shielded VM using PowerShell

Article • 09/21/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

In production, you would typically use a fabric manager (e.g. VMM) to deploy shielded VMs. However, the steps illustrated below allow you to deploy and validate the entire scenario without a fabric manager.

In a nutshell, you will create a template disk, a shielding data file, an unattended installation answer file and other security artifacts on any machine, then copy these files to a guarded host and provision the shielded VM.

Create a signed template disk

To create a new shielded VM, you first need a shielded VM template disk that is pre-encrypted with its OS volume (or boot and root partitions on Linux) signed. Follow the links below for more information on how to create a template disk.

- Prepare a Windows template disk
- Prepare a Linux template disk

You will also need a copy of the disk's volume signature catalog to create the shielding data file. To save this file, run the following command on the machine where you created the template disk:

PowerShell

```
Save-VolumeSignatureCatalog -TemplateDiskPath "C:\temp\MyTemplateDisk.vhdx"
-VolumeSignatureCatalogPath "C:\temp\MyTemplateDiskCatalog.vsc"
```

Download guardian metadata

For each of the virtualization fabrics where you wish to run your shielded VM, you will need to obtain guardian metadata for the fabrics' HGS clusters. Your hosting provider should be able to provide this information for you.

If you are in an enterprise environment and can communicate with the HGS server, the guardian metadata is available at *http://<HGSCLUSTERNAME>/KeyProtection/service/metadata/2014-07/metadata.xml*

Create Shielding Data (PDK) file

Shielding Data is created and owned by tenant VM owners and contains secrets needed to create shielded VMs that must be protected from the fabric admin, such as the shielded VM's administrator password. Shielding Data is encrypted such that only the HGS servers and tenant can decrypt it. Once created by the tenant/VM owner, the resulting PDK file must be copied to the guarded fabric. For more information, see What is shielding data and why is it necessary?.

In addition, you will need an unattended installation answer file (unattend.xml for Windows, varies for Linux). See Create an answer file for guidance on what to include in the answer file.

Run the following cmdlets on a machine with the Remote Server Administration Tools for Shielded VMs installed. If you are creating a PDK for a Linux VM, you must do this on a server running Windows Server, version 1709 or later.

PowerShell

```
# Create owner certificate, don't lose this!
# The certificate is stored at Cert:\LocalMachine\Shielded VM Local
Certificates
$Owner = New-HgsGuardian –Name 'Owner' –GenerateCertificates

# Import the HGS guardian for each fabric you want to run your shielded VM
$Guardian = Import-HgsGuardian -Path C:\HGSGuardian.xml -Name 'TestFabric'

# Create the PDK file
# The "Policy" parameter describes whether the admin can see the VM's
console or not
# Use "EncryptionSupported" if you are testing out shielded VMs and want to
debug any issues during the specialization process
New-ShieldingDataFile -ShieldingDataFilePath 'C:\temp\Contoso.pdk' -Owner
$Owner –Guardian $guardian –VolumeIDQualifier (New-VolumeIDQualifier -
VolumeSignatureCatalogFilePath 'C:\temp\MyTemplateDiskCatalog.vsc' -
VersionRule Equals) -WindowsUnattendFile 'C:\unattend.xml' -Policy Shielded
```

Provision shielded VM on a guarded host

On a host that is running Windows Server 2016, you can monitor for the VM to shut down to signal completion of the provisioning task, and consult the Hyper-V event logs for error information if the provisioning is unsuccessful.

You can also create a new template disk every time before creating new Shielded VM.

Copy the template disk file (ServerOS.vhdx) and the PDK file (contoso.pdk) to the guarded host to get ready for deployment.

On the guarded host, install the Guarded Fabric Tools PowerShell module, which contains the New-ShieldedVM cmdlet to simplify the provisioning process. If your guarded host has access to the Internet, run the following command:

PowerShell

```
Install-Module GuardedFabricTools -Repository PSGallery -MinimumVersion
1.0.0
```

You can also download the module on another computer that has Internet access and copy the resulting module to `C:\Program Files\WindowsPowerShell\Modules` on the guarded host.

PowerShell

```
Save-Module GuardedFabricTools -Repository PSGallery -MinimumVersion 1.0.0 -
Path C:\temp\
```

Once the module is installed, you're ready to provision your shielded VM.

PowerShell

```
New-ShieldedVM -Name 'MyShieldedVM' -TemplateDiskPath
'C:\temp\MyTemplateDisk.vhdx' -ShieldingDataFilePath 'C:\temp\Contoso.pdk' -
Wait
```

If your shielding data answer file includes specialization values, you can provide the replacement values to New-ShieldedVM. In this example, the answer file is configured with placeholder values for a static IPv4 address.

PowerShell

```
$specializationValues = @{
    "@IP4Addr-1@" = "192.168.1.10/24"
    "@MacAddr-1@" = "Ethernet"
    "@Prefix-1-1@" = "24"
    "@NextHop-1-1@" = "192.168.1.254"
}
New-ShieldedVM -Name 'MyStaticIPVM' -TemplateDiskPath
'C:\temp\MyTemplateDisk.vhdx' -ShieldingDataFilePath 'C:\temp\Contoso.pdk' -
SpecializationValues $specializationValues -Wait
```

If your template disk contains a Linux-based OS, include the `-Linux` flag when running the command:

```
PowerShell

New-ShieldedVM -Name 'MyLinuxVM' -TemplateDiskPath
'C:\temp\MyTemplateDisk.vhdx' -ShieldingDataFilePath 'C:\temp\Contoso.pdk' -
Wait -Linux
```

Check the help content using `Get-Help New-ShieldedVM -Full` to learn more about other options you can pass to the cmdlet.

Once the VM finishes provisioning, it will enter the OS-specific specialization phase, after which it will be ready for use. Be sure to connect the VM to a valid network so you can connect to it once it is running (using RDP, PowerShell, SSH, or your preferred management tool).

Running Shielded VMs on a Hyper-V cluster

If you are trying to deploy shielded VMs on clustered guarded hosts (using a Windows Failover Cluster), you can configure the shielded VM to be highly available using the following cmdlet:

```
PowerShell

Add-ClusterVirtualMachineRole -VMName 'MyShieldedVM' -Cluster <Hyper-V
cluster name>
```

The shielded VM can now be live migrated within the cluster.

Next step

Deploy a shielded using VMM

Feedback

Was this page helpful? Yes No

Shielded VMs for tenants - Deploying a shielded VM by using Virtual Machine Manager

Article • 09/21/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

If you have access to System Center 2016 - Virtual Machine Manager (VMM), you can deploy a shielded VM for which a shielded VM template has already been created.

To deploy a shielded VM in VMM, use the instructions in Provision a new shielded VM.

Additional References

- Deploy shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Shielded VMs for tenants - Deploying a shielded VM by using Windows Azure Pack

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

If your hosting service provider supports it, you can use Windows Azure Pack to deploy a shielded VM.

Complete the following steps:

1. Subscribe to one or more plans offered in Windows Azure Pack.
2. Create a shielded VM by using Windows Azure Pack.

   Use shielded virtual machines, which is described in the following topics:

   - Create shielding data (and upload the shielding data file, as described in the second procedure in the topic).

     > ⓘ **Note**
     >
     > As part of creating shielding data, you will download your guardian key file, which will be an XML file in UTF-8 format. Do not change the file to UTF-16.

   - Create a shielded virtual machine - with **Quick Create**, through a shielded template, or through a regular template.

     > ⚠ **Warning**
     >
     > If you **Create a shielded virtual machine by using a regular template**, it is important to note that the VM is provisioned *unshielded*. This means that the template disk is not verified against the list of trusted disks in your shielding data file, nor are the secrets in your shielding data file used to provision the VM. If a shielded template is available, it is preferable to deploy a shielded VM with a shielded template to provide end-to-end protection of your secrets.

- Convert a Generation 2 virtual machine to a shielded virtual machine

> **Note**
>
> If you convert a virtual machine to a shielded virtual machine, existing checkpoints and backups are not encrypted. You should delete old checkpoints when possible to prevent access to your old, decrypted data.

Additional References

- Hosting service provider configuration steps for guarded hosts and shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Shielded VMs - Preparing a VM Shielding Helper VHD

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

> **Important**
>
> Before beginning these procedures, ensure that you have installed the latest cumulative update for Windows Server 2016 or are using the latest Windows 10 Remote Server Administration Tools. Otherwise, the procedures will not work.

This section outlines steps performed by a hosting service provider to enable support for converting existing VMs to shielded VMs.

To understand how this topic fits in the overall process of deploying shielded VMs, see Hosting service provider configuration steps for guarded hosts and shielded VMs.

Which VMs can be shielded?

The shielding process for existing VMs is only available for VMs that meet the following prerequisites:

- The guest OS is Windows Server 2012, 2012 R2, 2016, or a Semi-Annual Channel release. Existing Linux VMs cannot be converted to shielded VMs.
- The VM is a generation 2 VM (UEFI firmware)
- The VM does not use differencing disks for its OS volume.

Prepare Helper VHD

1. On a machine with Hyper-V and the Remote Server Administration Tools feature **Shielded VM Tools** installed, create a new generation 2 VM with a blank VHDX and install Windows Server 2016 on it using the Windows Server ISO installation media. This VM should not be shielded and must run Server Core or Server with Desktop Experience.

   > **Important**

> The VM Shielding Helper VHD **must not** be related to the template disks you created in [Hosting service provider creates a shielded VM template](). If you re-use a template disk, there will be a disk signature collision during the shielding process because both disks will have the same GPT disk identifier. You can avoid this by creating a new (blank) VHD and installing Windows Server 2016 onto it using your ISO installation media.

2. Start the VM, complete any setup steps, and log into the desktop. Once you have verified the VM is in a working state, shut down the VM.

3. In an elevated Windows PowerShell window, run the following command to prepare the VHDX created earlier to become a VM shielding helper disk. Update the path with the correct path for your environment.

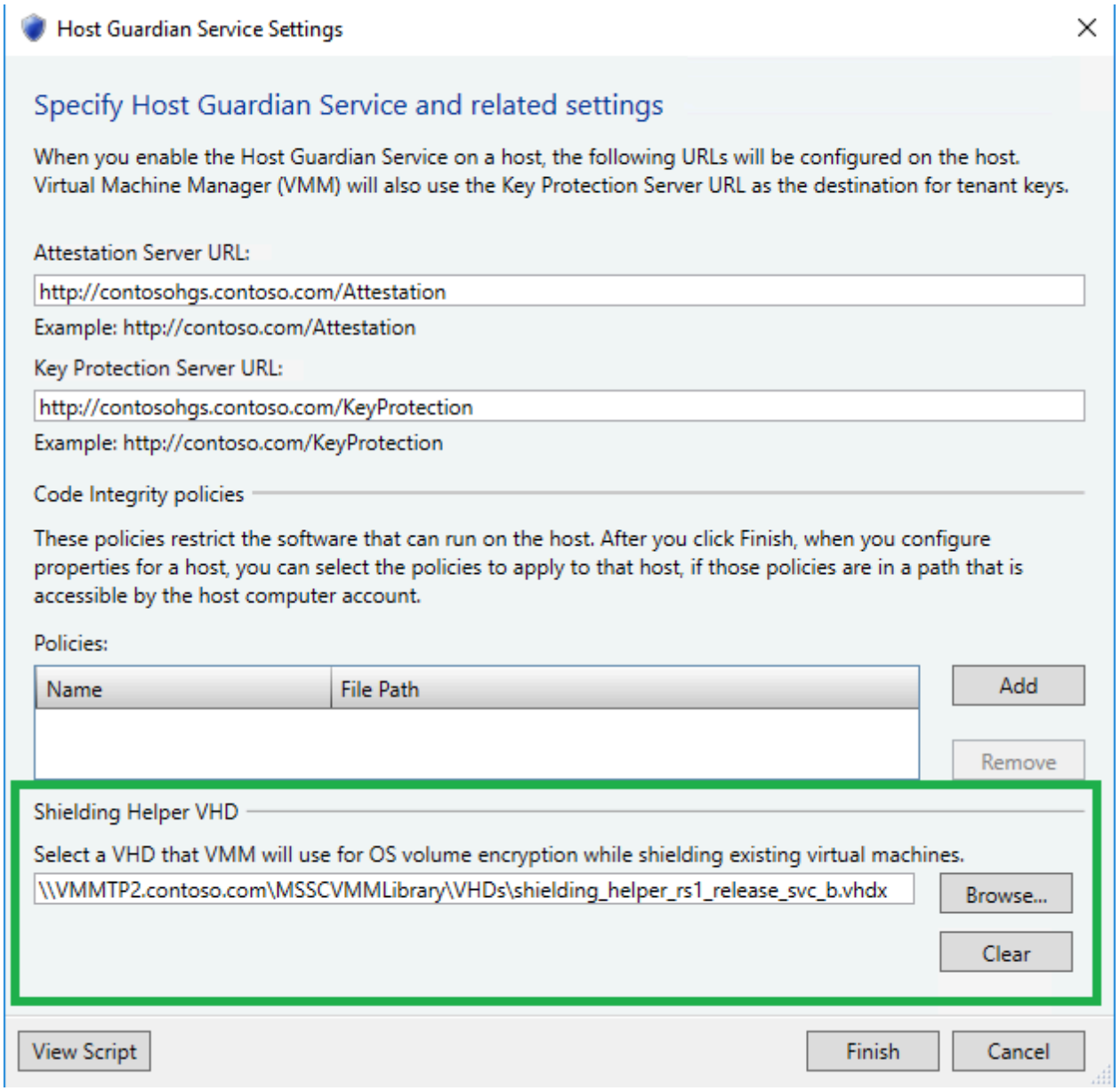
   ```PowerShell
   Initialize-VMShieldingHelperVHD -Path 'C:\VHD\shieldingHelper.vhdx'
   ```

4. Once the command has completed successfully, copy the VHDX to your VMM library share. **Do not** start up the VM from step 1 again. Doing so will corrupt the helper disk.

5. You can now delete the VM from step 1 in Hyper-V.

Configure VMM Host Guardian Server Settings

In the VMM Console, open the settings pane and then **Host Guardian Service Settings** under **General**. At the bottom of this window, there is a field to configure the location of your helper VHD. Use the browse button to select the VHD from your library share. If you do not see your disk in the share, you may need to manually refresh the library in VMM for it to show up.



Additional References

- Hosting service provider configuration steps for guarded hosts and shielded VMs
- Guarded fabric and shielded VMs

Feedback

Was this page helpful? Yes No

Managing a Guarded Fabric

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The following topics describe how to manage a guarded fabric.

- Manage the Host Guardian Service
- Branch office considerations

Additional References

- Deploying a guarded fabric

Feedback

Was this page helpful? Yes No

Managing the Host Guardian Service

Article • 04/21/2023 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Host Guardian Service (HGS) is the centerpiece of the guarded fabric solution. It is responsible for ensuring that Hyper-V hosts in the fabric are known to the hoster or enterprise and running trusted software and for managing the keys used to start up shielded VMs. When a tenant decides to trust you to host their shielded VMs, they are placing their trust in your configuration and management of the Host Guardian Service. Therefore, it is very important to follow best practices when managing the Host Guardian Service to ensure the security, availability and reliability of your guarded fabric. The guidance in the following sections addresses the most common operational issues facing administrators of HGS.

Limiting admin access to HGS

Due to the security sensitive nature of HGS, it is important to ensure that its administrators are highly trusted members of your organization and, ideally, separate from the administrators of your fabric resources. Additionally, it is recommended that you only manage HGS from secure workstations using secure communication protocols, such as WinRM over HTTPS.

Separation of Duties

When setting up HGS, you are given the option of creating an isolated Active Directory forest just for HGS or to join HGS to an existing, trusted domain. This decision, as well as the roles you assign the admins in your organization, determine the trust boundary for HGS. Whoever has access to HGS, whether directly as an admin or indirectly as an admin of something else (e.g. Active Directory) that can influence HGS, has control over your guarded fabric. HGS admins choose which Hyper-V hosts are authorized to run shielded VMs and manage the certificates necessary to start up shielded VMs. An attacker or malicious admin who has access to HGS can use this power to authorize compromised hosts to run shielded VMs, initiate a denial-of-service attack by removing key material, and more.

To avoid this risk, it is *strongly* recommended that you limit the overlap between the admins of your HGS (including the domain to which HGS is joined) and Hyper-V environments. By ensuring no one admin has access to both systems, an attacker would need to compromise 2 different accounts from 2 individuals to complete his mission to

change the HGS policies. This also means that the domain and enterprise admins for the two Active Directory environments should not be the same person, nor should HGS use the same Active Directory forest as your Hyper-V hosts. Anyone who can grant themselves access to more resources poses a security risk.

Using Just Enough Administration

HGS comes with Just Enough Administration (JEA) roles built in to help you manage it more securely. JEA helps by allowing you to delegate admin tasks to non-admin users, meaning the people who manage HGS policies need not actually be admins of the entire machine or domain. JEA works by limiting what commands a user can run in a PowerShell session and using a temporary local account behind the scenes (unique for each user session) to run the commands which normally require elevation.

HGS ships with 2 JEA roles preconfigured:

- **HGS Administrators** which allows users to manage all HGS policies, including authorizing new hosts to run shielded VMs.
- **HGS Reviewers** which only allows users the right to audit existing policies. They cannot make any changes to the HGS configuration.

To use JEA, you first need to create a new standard user and make them a member of either the HGS admins or HGS reviewers group. If you used `Install-HgsServer` to set up a new forest for HGS, these groups will be named "*servicename*Administrators" and "*servicename*Reviewers", respectively, where *servicename* is the network name of the HGS cluster. If you joined HGS to an existing domain, you should refer to the group names you specified in `Initialize-HgsServer`.

Create standard users for the HGS administrator and reviewer roles

PowerShell

```
$hgsServiceName = (Get-ClusterResource HgsClusterResource | Get-
ClusterParameter DnsName).Value
$adminGroup = $hgsServiceName + "Administrators"
$reviewerGroup = $hgsServiceName + "Reviewers"

New-ADUser -Name 'hgsadmin01' -AccountPassword (Read-Host -AsSecureString -
Prompt 'HGS Admin Password') -ChangePasswordAtLogon $false -Enabled $true
Add-ADGroupMember -Identity $adminGroup -Members 'hgsadmin01'

New-ADUser -Name 'hgsreviewer01' -AccountPassword (Read-Host -AsSecureString
-Prompt 'HGS Reviewer Password') -ChangePasswordAtLogon $false -Enabled
$true
Add-ADGroupMember -Identity $reviewerGroup -Members 'hgsreviewer01'
```

Audit policies with the reviewer role

On a remote machine that has network connectivity to HGS, run the following commands in PowerShell to enter the JEA session with the reviewer credentials. It is important to note that since the reviewer account is just a standard user, it cannot be used for regular Windows PowerShell remoting, Remote Desktop access to HGS, etc.

PowerShell

```
Enter-PSSession -ComputerName <hgsnode> -Credential
'<hgsdomain>\hgsreviewer01' -ConfigurationName 'microsoft.windows.hgs'
```

You can then check which commands are allowed in the session with `Get-Command` and run any allowed commands to audit the configuration. In the below example, we are checking which policies are enabled on HGS.

PowerShell

```
Get-Command

Get-HgsAttestationPolicy
```

Type the command `Exit-PSSession` or its alias, `exit`, when you are done working with the JEA session.

Add a new policy to HGS using the administrator role

To actually change a policy, you need to connect to the JEA endpoint with an identity that belongs to the 'hgsAdministrators' group. In the below example, we show how you can copy a new code integrity policy to HGS and register it using JEA. The syntax may be different from what you are used to. This is to accommodate some of the restrictions in JEA like not having access to the full file system.

PowerShell

```
$cipolicy = Get-Item "C:\temp\cipolicy.p7b"
$session = New-PSSession -ComputerName <hgsnode> -Credential
'<hgsdomain>\hgsadmin01' -ConfigurationName 'microsoft.windows.hgs'
Copy-Item -Path $cipolicy -Destination 'User:' -ToSession $session

# Now that the file is copied, we enter the interactive session to register
it with HGS
Enter-PSSession -Session $session
Add-HgsAttestationCiPolicy -Name 'New CI Policy via JEA' -Path
'User:\cipolicy.p7b'

# Confirm it was added successfully
```

```
Get-HgsAttestationPolicy -PolicyType CiPolicy

# Finally, remove the PSSession since it is no longer needed
Exit-PSSession
Remove-PSSession -Session $session
```

Monitoring HGS

Event sources and forwarding

Events from HGS will show up in the Windows event log under 2 sources:

- **HostGuardianService-Attestation**
- **HostGuardianService-KeyProtection**

You can view these events by opening Event Viewer and navigating to Microsoft-Windows-HostGuardianService-Attestation and Microsoft-Windows-HostGuardianService-KeyProtection.

In a large environment, it is often preferable to forward events to a central Windows Event Collector to make analyzation of the events easier. For more information, check out the Windows Event Forwarding documentation.

Using System Center Operations Manager

You can also use System Center 2016 - Operations Manager to monitor HGS and your guarded hosts. The guarded fabric management pack has event monitors to check for common misconfigurations that can lead to datacenter downtime, including hosts not passing attestation and HGS servers reporting errors.

To get started, install and configure SCOM 2016 and download the guarded fabric management pack. The included management pack guide explains how to configure the management pack and understand the scope of its monitors.

Backing up and restoring HGS

Disaster recovery planning

When drafting your disaster recovery plans, it is important to consider the unique requirements of the Host Guardian Service in your guarded fabric. Should you lose some or all of your HGS nodes, you may face immediate availability problems that will prevent

users from starting up their shielded VMs. In a scenario where you lose your entire HGS cluster, you will need to have complete backups of the HGS configuration on hand to restore your HGS cluster and resume normal operations. This section covers the steps necessary to prepare for such a scenario.

First, it's important to understand what about HGS is important to back up. HGS retains several pieces of information that help it determine which hosts are authorized to run shielded VMs. This includes:

1. Active Directory security identifiers for the groups containing trusted hosts (when using Active Directory attestation);
2. Unique TPM identifiers for each host in your environment;
3. TPM policies for each unique configuration of host; and
4. Code integrity policies that determine which software is allowed to run on your hosts.

These attestation artifacts require coordination with the admins of your hosting fabric to obtain, potentially making it difficult to get this information again after a disaster.

Additionally, HGS requires access to 2 or more certificates used to encrypt and sign the information required to start up a shielded VM (the key protector). These certificates are well known (used by the owners of shielded VMs to authorize your fabric to run their VMs) and must be restored after a disaster for a seamless recovery experience. Should you not restore HGS with the same certificates after a disaster, each VM would need to be updated to authorize your new keys to decrypt their information. For security reasons, only the VM owner can update the VM configuration to authorize these new keys, meaning failure to restore your keys after a disaster will result in each VM owner needing to take action to get their VMs running again.

Preparing for the worst

To prepare for a complete loss of HGS, there are 2 steps you must take:

1. Back up the HGS attestation policies
2. Back up the HGS keys

Guidance on how to perform both of these steps is provided in the Backing up HGS section.

It is additionally recommended, but not required, that you back up the list of users authorized to manage HGS in its Active Directory domain or Active Directory itself.

Backups should be taken regularly to ensure the information is up to date and stored securely to avoid tampering or theft.

It is **not recommended** to back up or attempt to restore an entire system image of an HGS node. In the event you have lost your entire cluster, the best practice is to set up a brand new HGS node and restore just the HGS state, not the entire server OS.

Recovering from the loss of one node

If you lose one or more nodes (but not every node) in your HGS cluster, you can simply add nodes to your cluster following the guidance in the deployment guide. The attestation policies will sync automatically, as will any certificates which were provided to HGS as PFX files with accompanying passwords. For certificates added to HGS using a thumbprint (non-exportable and hardware backed certificates, commonly), you will need to ensure each new node has access to the private key of each certificate.

Recovering from the loss of the entire cluster

If your entire HGS cluster goes down and you are unable to bring it back online, you will need to restore HGS from a backup. Restoring HGS from a backup involves first setting up a new HGS cluster per the guidance in the deployment guide. It is highly recommended, but not required, to use the same cluster name when setting up the recovery HGS environment to assist with name resolution from hosts. Using the same name avoids having to reconfigure hosts with new attestation and key protection URLs. If you restored objects to the Active Directory domain backing HGS, it is recommended that you remove the objects representing the HGS cluster, computers, service account and JEA groups before initializing the HGS server.

Once you have set up your first HGS node (e.g. it has been installed and initialized), you will follow the procedures under Restoring HGS from a backup to restore the attestation policies and public halves of the key protection certificates. You will need to restore the private keys for your certificates manually according to the guidance of your certificate provider (e.g. import the certificate in Windows, or configure access to HSM-backed certificates). After the first node is set up, you can continue to install additional nodes to the cluster until you have reached the capacity and resiliency you desire.

Backing up HGS

The HGS administrator should be responsible for backing up HGS on a regular basis. A complete backup will contain sensitive key material that must be appropriately secured.

Should an untrusted entity gain access to these keys, they could use that material to set up a malicious HGS environment for the purpose of compromising shielded VMs.

Backing up the attestation policies To back up the HGS attestation policies, run the following command on any working HGS server node. You will be prompted to provide a password. This password is used to encrypt any certificates added to HGS using a PFX file (instead of a certificate thumbprint).

PowerShell

```
Export-HgsServerState -Path C:\temp\HGSBackup.xml
```

> **Note**
>
> If you are using admin-trusted attestation, you must separately back up membership in the security groups used by HGS to authorize guarded hosts. HGS will only back up the SID of the security groups, not the membership within them. In the event these groups are lost during a disaster, you will need to recreate the group(s) and add each guarded host to them again.

Backing up certificates

The `Export-HgsServerState` command will back up any PFX-based certificates added to HGS at the time the command is run. If you added certificates to HGS using a thumbprint (typical for non-exportable and hardware-backed certificates), you will need to manually back up the private keys for your certificates. To identify which certificates are registered with HGS and need to be backed up manually, run the following PowerShell command on any working HGS server node.

PowerShell

```
Get-HgsKeyProtectionCertificate | Where-Object {
$_.CertificateData.GetType().Name -eq 'CertificateReference' } | Format-
Table Thumbprint, @{ Label = 'Subject'; Expression = {
$_.CertificateData.Certificate.Subject } }
```

For each of the certificates listed, you will need to manually back up the private key. If you are using software-based certificate that is non-exportable, you should contact your certificate authority to ensure they have a backup of your certificate and/or can reissue it on demand. For certificates created and stored in hardware security modules, you should consult the documentation for your device for guidance on disaster recovery planning.

You should store the certificate backups alongside your attestation policy backups in a secure location so that both pieces can be restored together.

Additional configuration to back up

The backed up HGS server state will not include the name of your HGS cluster, any information from Active Directory, or any SSL certificates used to secure communications with the HGS APIs. These settings are important for consistency but not critical to get your HGS cluster back online after a disaster.

To capture the name of the HGS service, run `Get-HgsServer` and note the flat name in the Attestation and Key Protection URLs. For example, if the Attestation URL is "https://hgs.contoso.com/Attestation", "hgs" is the HGS service name.

The Active Directory domain used by HGS should be managed like any other Active Directory domain. When restoring HGS after a disaster, you will not necessarily need to recreate the exact objects that are present in the current domain. However, it will make recovery easier if you back up Active Directory and keep a list of the JEA users authorized to manage the system as well as the membership of any security groups used by admin-trusted attestation to authorize guarded hosts.

To identify the thumbprint of the SSL certificates configured for HGS, run the following command in PowerShell. You can then back up those SSL certificates according to your certificate provider's instructions.

PowerShell

```
Get-WebBinding -Protocol https | Select-Object certificateHash
```

Restoring HGS from a backup

The following steps describe how to restore HGS settings from a backup. The steps are relevant to both situations where you are trying to undo changes made to your already-running HGS instances and when you are standing up a brand new HGS cluster after a complete loss of your previous one.

Set up a replacement HGS cluster

Before you can restore HGS, you need to have an initialized HGS cluster to which you can restore the configuration. If you are simply importing settings that were accidentally deleted to an existing (running) cluster, you can skip this step. If you are recovering from

a complete loss of HGS, you will need to install and initialize at least one HGS node following the guidance in the deployment guide.

Specifically, you will need to:

1. Set up the HGS domain or join HGS to an existing domain
2. Initialize the HGS server using your existing keys *or* a set of temporary keys. You can remove the temporary keys after importing your actual keys from the HGS backup files.
3. Import HGS settings from your backup to restore the trusted host groups, code integrity policies, TPM baselines, and TPM identifiers

> **Tip**
>
> The new HGS cluster does not need to use the same certificates, service name, or domain as the HGS instance from which your backup file was exported.

Import settings from a backup

To restore attestation policies, PFX-based certificates, and the public keys of non-PFX certificates to your HGS node from a backup file, run the following command on an initialized HGS server node. You will be prompted to enter the password you specified when creating the backup.

PowerShell

```
Import-HgsServerState -Path C:\Temp\HGSBackup.xml
```

If you only want to import admin-trusted attestation policies or TPM-trusted attestation policies, you can do so by specifying the `-ImportActiveDirectoryModeState` or `-ImportTpmModeState` flags to Import-HgsServerState.

Ensure the latest cumulative update for Windows Server 2016 is installed before running `Import-HgsServerState`. Failure to do so may result in an import error.

> **Note**
>
> If you restore policies on an existing HGS node that already has one or more of those policies installed, the import command will show an error for each duplicate policy. This is an expected behavior and can be safely ignored in most cases.

Reinstall private keys for certificates

If any of the certificates used on the HGS from which the backup was created were added using thumbprints, only the public key of those certificates will be included in the backup file. This means that you will need to manually install and/or grant access to the private keys for each of those certificates before HGS can service requests from Hyper-V hosts. The actions necessary to complete that step varies depending on how your certificate was originally issued. For software-backed certificates issued by a certificate authority, you will need to contact your CA to get the private key and install it on **each** HGS node per their instructions. Similarly, if your certificates are hardware-backed, you will need to consult your hardware security module vendor's documentation to install the necessary driver(s) on each HGS node to connect to the HSM and grant each machine access to the private key.

As a reminder, certificates added to HGS using thumbprints require manual replication of the private keys to each node. You will need to repeat this step on each additional node you add to the restored HGS cluster.

Review imported attestation policies

After you've imported your settings from a backup, it is recommended to closely review all the imported policies using `Get-HgsAttestationPolicy` to make sure only the hosts you trust to run shielded VMs will be able to successfully attest. If you find any policies which no longer match your security posture, you can disable or remove them.

Run diagnostics to check system state

After you have finished setting up and restoring the state of your HGS node, you should run the HGS diagnostics tool to check the state of the system. To do this, run the following command on the HGS node where you restored the configuration:

PowerShell

```
Get-HgsTrace -RunDiagnostics
```

If the "Overall Result" is not "Pass", additional steps are required to finish configuring the system. Check the messages reported in the subtest(s) that failed for more information.

Patching HGS

It is important to keep your Host Guardian Service nodes up to date by installing the latest cumulative update when it comes out. If you are setting up a brand new HGS node, it is highly recommended that you install any available updates before installing the HGS role or configuring it. This will ensure any new or changed functionality will take effect immediately.

When patching your guarded fabric, it is strongly recommended that you first upgrade *all* Hyper-V hosts **before upgrading HGS**. This is to ensure that any changes to the attestation policies on HGS are made *after* the Hyper-V hosts have been updated to provide the information needed for them. If an update is going to change the behavior of policies, they will not automatically be enabled to avoid disrupting your fabric. Such updates require that you follow the guidance in the following section to activate the new or changed attestation policies. We encourage you to read the release notes for Windows Server and any cumulative updates you install to check if the policy updates are required.

Updates requiring policy activation

If an update for HGS introduces or significantly changes the behavior of an attestation policy, an additional step is required to activate the changed policy. Policy changes are only enacted after exporting and importing the HGS state. You should only activate the new or changed policies after you have applied the cumulative update to all hosts and all HGS nodes in your environment. Once every machine has been updated, run the following commands on any HGS node to trigger the upgrade process:

PowerShell

```
$password = Read-Host -AsSecureString -Prompt "Enter a temporary password"
Export-HgsServerState -Path .\temporaryExport.xml -Password $password
Import-HgsServerState -Path .\temporaryExport.xml -Password $password
```

If a new policy was introduced, it will be disabled by default. To enable the new policy, first find it in the list of Microsoft policies (prefixed with 'HGS_') and then enable it using the following commands:

PowerShell

```
Get-HgsAttestationPolicy

Enable-HgsAttestationPolicy -Name <Hgs_NewPolicyName>
```

Managing attestation policies

HGS maintains several attestation policies which define the minimum set of requirements a host must meet in order to be deemed "healthy" and allowed to run shielded VMs. Some of these policies are defined by Microsoft, others are added by you to define the allowable code integrity policies, TPM baselines, and hosts in your environment. Regular maintenance of these policies is necessary to ensure hosts can continue attesting properly as you update and replace them, and to ensure any untrusted hosts or configurations are blocked from successfully attesting.

For admin-trusted attestation, there is only one policy which determines if a host is healthy: membership in a known, trusted security group. TPM attestation is more complicated, and involves various policies to measure the code and configuration of a system before determining if it is healthy.

A single HGS can be configured with both Active Directory and TPM policies at once, but the service will only check the policies for the current mode which it is configured for when a host tries attesting. To check the mode of your HGS server, run `Get-HgsServer`.

Default policies

For TPM-trusted attestation, there are several built-in policies configured on HGS. Some of these policies are "locked" -- meaning that they cannot be disabled for security reasons. The table below explains the purpose of each default policy.

Expand table

Policy Name	Purpose
Hgs_SecureBootEnabled	Requires hosts to have Secure Boot enabled. This is necessary to measure the startup binaries and other UEFI-locked settings.
Hgs_UefiDebugDisabled	Ensures hosts do not have a kernel debugger enabled. User-mode debuggers are blocked with code integrity policies.
Hgs_SecureBootSettings	Negative policy to ensure hosts match at least one (admin-defined) TPM baseline.
Hgs_CiPolicy	Negative policy to ensure hosts are using one of the admin-defined CI policies.
Hgs_HypervisorEnforcedCiPolicy	Requires the code integrity policy to be enforced by the hypervisor. Disabling this policy weakens your protections against kernel-mode code integrity policy attacks.
Hgs_FullBoot	Ensures the host did not resume from sleep or hibernation. Hosts must be properly restarted or shut down to pass this

Policy Name	Purpose
	policy.
Hgs_VsmIdkPresent	Requires virtualization based security to be running on the host. The IDK represents the key necessary to encrypt information sent back to the host's secure memory space.
Hgs_PageFileEncryptionEnabled	Requires pagefiles to be encrypted on the host. Disabling this policy could result in information exposure if an unencrypted pagefile is inspected for tenant secrets.
Hgs_BitLockerEnabled	Requires BitLocker to be enabled on the Hyper-V host. This policy is disabled by default for performance reasons and is not recommended to be enabled. This policy has no bearing on the encryption of the shielded VMs themselves.
Hgs_IommuEnabled	Requires that the host have an IOMMU device in use to prevent direct memory access attacks. Disabling this policy and using hosts without an IOMMU enabled can expose tenant VM secrets to direct memory attacks.
Hgs_NoHibernation	Requires hibernation to be disabled on the Hyper-V host. Disabling this policy could allow hosts to save shielded VM memory to an unencrypted hibernation file.
Hgs_NoDumps	Requires memory dumps to be disabled on the Hyper-V host. If you disable this policy, it is recommended that you configure dump encryption to prevent shielded VM memory from being saved to unencrypted crash dump files.
Hgs_DumpEncryption	Requires memory dumps, if enabled on the Hyper-V host, to be encrypted with an encryption key trusted by HGS. This policy does not apply if dumps are not enabled on the host. If this policy and *Hgs_NoDumps* are both disabled, shielded VM memory could be saved to an unencrypted dump file.
Hgs_DumpEncryptionKey	Negative policy to ensure hosts configured to allow memory dumps are using an admin-defined dump file encryption key known to HGS. This policy does not apply when *Hgs_DumpEncryption* is disabled.

Authorizing new guarded hosts

To authorize a new host to become a guarded host (e.g. attest successfully), HGS must trust the host and (when configured to use TPM-trusted attestation) the software running on it. The steps to authorize a new host differ based on the attestation mode for which HGS is currently configured. To check the attestation mode for your guarded fabric, run `Get-HgsServer` on any HGS node.

Software configuration

On the new Hyper-V host, ensure that Windows Server 2016 Datacenter edition is installed. Windows Server 2016 Standard cannot run shielded VMs in a guarded fabric. The host may be installed Desktop Experience or Server Core.

On server with desktop experience and Server Core, you need to install the Hyper-V and Host Guardian Hyper-V Support server roles:

```
PowerShell

Install-WindowsFeature Hyper-V, HostGuardian -IncludeManagementTools -
Restart
```

Admin-trusted attestation

To register a new host in HGS when using admin-trusted attestation, you must first add the host to a security group in the domain to which it's joined. Typically, each domain will have one security group for guarded hosts. If you have already registered that group with HGS, the only action you need to take is to restart the host to refresh its group membership.

You can check which security groups are trusted by HGS by running the following command:

```
PowerShell

Get-HgsAttestationHostGroup
```

To register a new security group with HGS, first capture the security identifier (SID) of the group in the host's domain and register the SID with HGS.

```
PowerShell

Add-HgsAttestationHostGroup -Name "Contoso Guarded Hosts" -Identifier "S-1-
5-21-3623811015-3361044348-30300820-1013"
```

Instructions on how to set up the trust between the host domain and HGS are available in the deployment guide.

TPM-trusted attestation

When HGS is configured in TPM mode, hosts must pass all locked policies and "enabled" policies prefixed with "Hgs_", as well as at least one TPM baseline, TPM identifier, and code integrity policy. Each time you add a new host, you will need to register the new TPM identifier with HGS. As long as the host is running the same software (and has the same code integrity policy applied) and TPM baseline as another host in your environment, you will not need to add new CI policies or baselines.

Adding the TPM identifier for a new host On the new host, run the following command to capture the TPM identifier. Be sure to specify a unique name for the host that will help you look it up on HGS. You will need this information if you decommission the host or want to prevent it from running shielded VMs in HGS.

PowerShell

```
(Get-PlatformIdentifier -Name "Host01").InnerXml | Out-File
C:\temp\host01.xml -Encoding UTF8
```

Copy this file to your HGS server, then run the following command to register the host with HGS.

PowerShell

```
Add-HgsAttestationTpmHost -Name 'Host01' -Path C:\temp\host01.xml
```

Adding a new TPM baseline If the new host is running a new hardware or firmware configuration for your environment, you may need to take a new TPM baseline. To do this, run the following command on the host.

PowerShell

```
Get-HgsAttestationBaselinePolicy -Path 'C:\temp\hardwareConfig01.tcglog'
```

> ⓘ **Note**
>
> If you receive an error saying your host failed validation and will not successfully attest, do not worry. This is a prerequisite check to make sure your host can run shielded VMs, and likely means that you have not yet applied a code integrity policy or other required setting. Read the error message, make any changes suggested by it, then try again. Alternatively, you can skip the validation at this time by adding the `-SkipValidation` flag to the command.

Copy the TPM baseline to your HGS server, then register it with the following command. We encourage you to use a naming convention that helps you understand the hardware and firmware configuration of this class of Hyper-V host.

PowerShell

```
Add-HgsAttestationTpmPolicy -Name 'HardwareConfig01' -Path
'C:\temp\hardwareConfig01.tcglog'
```

Adding a new code integrity policy If you have changed the code integrity policy running on your Hyper-V hosts, you will need to register the new policy with HGS before those hosts can successfully attest. On a reference host, which serves as a master image for the trusted Hyper-V machines in your environment, capture a new CI policy using the `New-CIPolicy` command. We encourage you to use the **FilePublisher** level and **Hash** fallback for Hyper-V host CI policies. You should first create a CI policy in audit mode to ensure that everything is working as expected. After validating a sample workload on the system, you can enforce the policy and copy the enforced version to HGS. For a complete list of code integrity policy configuration options, consult the Device Guard documentation.

PowerShell

```
# Capture a new CI policy with the FilePublisher primary level and Hash
fallback and enable user mode code integrity protections
New-CIPolicy -FilePath 'C:\temp\ws2016-hardware01-ci.xml' -Level
FilePublisher -Fallback Hash -UserPEs

# Apply the CI policy to the system
ConvertFrom-CIPolicy -XmlFilePath 'C:\temp\ws2016-hardware01-ci.xml' -
BinaryFilePath 'C:\temp\ws2016-hardware01-ci.p7b'
Copy-Item 'C:\temp\ws2016-hardware01-ci.p7b'
'C:\Windows\System32\CodeIntegrity\SIPolicy.p7b'
Restart-Computer

# Check the event log for any untrusted binaries and update the policy if
necessary
# Consult the Device Guard documentation for more details

# Change the policy to be in enforced mode
Set-RuleOption -FilePath 'C:\temp\ws2016-hardare01-ci.xml' -Option 3 -Delete

# Apply the enforced CI policy on the system
ConvertFrom-CIPolicy -XmlFilePath 'C:\temp\ws2016-hardware01-ci.xml' -
BinaryFilePath 'C:\temp\ws2016-hardware01-ci.p7b'
Copy-Item 'C:\temp\ws2016-hardware01-ci.p7b'
'C:\Windows\System32\CodeIntegrity\SIPolicy.p7b'
Restart-Computer
```

Once you have your policy created, tested and enforced, copy the binary file (.p7b) to your HGS server and register the policy.

```PowerShell
Add-HgsAttestationCiPolicy -Name 'WS2016-Hardware01' -Path 'C:\temp\ws2016-hardware01-ci.p7b'
```

Adding a memory dump encryption key

When the *Hgs_NoDumps* policy is disabled and *Hgs_DumpEncryption* policy is enabled, guarded hosts are allowed to have memory dumps (including crash dumps) to be enabled as long as those dumps are encrypted. Guarded hosts will only pass attestation if they either have memory dumps disabled or are encrypting them with a key known to HGS. By default, no dump encryption keys are configured on HGS.

To add a dump encryption key to HGS, use the `Add-HgsAttestationDumpPolicy` cmdlet to provide HGS with the hash of your dump encryption key. If you capture a TPM baseline on a Hyper-V host configured with dump encryption, the hash is included in the tcglog and can be provided to the `Add-HgsAttestationDumpPolicy` cmdlet.

```PowerShell
Add-HgsAttestationDumpPolicy -Name 'DumpEncryptionKey01' -Path 'C:\temp\TpmBaselineWithDumpEncryptionKey.tcglog'
```

Alternatively, you can directly provide the string representation of the hash to the cmdlet.

```PowerShell
Add-HgsAttestationDumpPolicy -Name 'DumpEncryptionKey02' -PublicKeyHash '<paste your hash here>'
```

Be sure to add each unique dump encryption key to HGS if you choose to use different keys across your guarded fabric. Hosts that are encrypting memory dumps with a key not known to HGS will not pass attestation.

Consult the Hyper-V documentation for more information about configuring dump encryption on hosts.

Check if the system passed attestation

After registering the necessary information with HGS, you should check if the host passes attestation. On the newly-added Hyper-V host, run `Set-HgsClientConfiguration` and supply the correct URLs for your HGS cluster. These URLs can be obtained by running `Get-HgsServer` on any HGS node.

PowerShell

```
Set-HgsClientConfiguration -KeyProtectionServerUrl
'https://hgs.bastion.local/KeyProtection' -AttestationServerUrl
'https://hgs.bastion.local/Attestation'
```

If the resulting status does not indicate "IsHostGuarded : True" you will need to troubleshoot the configuration. On the host that failed attestation, run the following command to get a detailed report about issues that may help you resolve the failed attestation.

PowerShell

```
Get-HgsTrace -RunDiagnostics -Detailed
```

> **Important**
>
> If you're using Windows Server 2019 or Windows 10, version 1809 and are using code integrity policies, `Get-HgsTrace` may return a failure for the **Code Integrity Policy Active** diagnostic. You can safely ignore this result when it is the only failing diagnostic.

Review attestation policies

To review the current state of the policies configured on HGS, run the following commands on any HGS node:

PowerShell

```
# List all trusted security groups for admin-trusted attestation
Get-HgsAttestationHostGroup

# List all policies configured for TPM-trusted attestation
Get-HgsAttestationPolicy
```

If you find a policy enabled that no longer meets your security requirement (e.g. an old code integrity policy which is now deemed unsafe), you can disable it by replacing the

name of the policy in the following command:

```PowerShell
Disable-HgsAttestationPolicy -Name 'PolicyName'
```

Similarly, you can use `Enable-HgsAttestationPolicy` to re-enable a policy.

If you no longer need a policy and wish to remove it from all HGS nodes, run `Remove-HgsAttestationPolicy -Name 'PolicyName'` to permanently delete the policy.

Changing attestation modes

If you started your guarded fabric using admin-trusted attestation, you will likely want to upgrade to the much-stronger TPM attestation mode as soon as you have enough TPM 2.0-compatible hosts in your environment. When you're ready to switch, you can pre-load all of the attestation artifacts (CI policies, TPM baselines and TPM identifiers) in HGS while continuing to run HGS with admin-trusted attestation. To do this, simply follow the instruction in the authorizing a new guarded host section.

Once you've added all of your policies to HGS, the next step is to run a synthetic attestation attempt on your hosts to see if they would pass attestation in TPM mode. This does not affect the current operational state of HGS. The commands below must be run on a machine that has access to all of the hosts in the environment and at least one HGS node. If your firewall or other security policies prevent this, you can skip this step. When possible, we recommend running the synthetic attestation to give you a good indication of whether "flipping" to TPM mode will cause downtime for your VMs.

```PowerShell
# Get information for each host in your environment
$hostNames = 'host01.contoso.com', 'host02.contoso.com',
'host03.contoso.com'
$credential = Get-Credential -Message 'Enter a credential with admin
privileges on each host'
$targets = @()
$hostNames | ForEach-Object { $targets += New-HgsTraceTarget -Credential
$credential -Role GuardedHost -HostName $_ }

$hgsCredential = Get-Credential -Message 'Enter an admin credential for HGS'
$targets += New-HgsTraceTarget -Credential $hgsCredential -Role
HostGuardianService -HostName 'HGS01.bastion.local'

# Initiate the synthetic attestation attempt
```

```
Get-HgsTrace -RunDiagnostics -Target $targets -Diagnostic
GuardedFabricTpmMode
```

After the diagnostics complete, review the outputted information to determine if any hosts would have failed attestation in TPM mode. Re-run the diagnostics until you get a "pass" from each host, then proceed to change HGS to TPM mode.

Changing to TPM mode takes just a second to complete. Run the following command on any HGS node to update the attestation mode.

PowerShell

```
Set-HgsServer -TrustTpm
```

If you run into problems and need to switch back to Active Directory mode, you can do so by running `Set-HgsServer -TrustActiveDirectory`.

Once you have confirmed everything is working as expected, you should remove all trusted Active Directory host groups from HGS and remove the trust between the HGS and fabric domains. If you leave the Active Directory trust in place, you risk someone re-enabling the trust and switching HGS to Active Directory mode, which could allow untrusted code to run unchecked on your guarded hosts.

Key management

The guarded fabric solution uses several public/private key pairs to validate the integrity of various components in the solution and encrypt tenant secrets. The Host Guardian Service is configured with at least two certificates (with public and private keys), which are used for signing and encrypting the keys used to start up shielded VMs. Those keys must be carefully managed. If the private key is acquired by an adversary, they will be able to unshield any VMs running on your fabric or set up an imposter HGS cluster that uses weaker attestation policies to bypass the protections you put in place. Should you lose the private keys during a disaster and not find them in a backup, you will need to set up a new pair of keys and have each VM re-keyed to authorize your new certificates.

This section covers general key management topics to help you configure your keys so they are functional and secure.

Adding new keys

While HGS must be initialized with one set of keys, you can add more than one encryption and signing key to HGS. The two most common reasons why you would add

new keys to HGS are:

1. To support "bring your own key" scenarios where tenants copy their private keys to your hardware security module and only authorize their keys to start up their shielded VMs.
2. To replace the existing keys for HGS by first adding the new keys and keeping both sets of keys until each VM configuration has been updated to use the new keys.

The process to add your new keys differs based on the type of certificate you are using.

Option 1: Adding a certificate stored in an HSM

Our recommended approach for securing HGS keys is to use certificates created in a hardware security module (HSM). HSMs ensure use of your keys is tied to physical access to a security-sensitive device in your datacenter. Each HSM is different and has a unique process to create certificates and register them with HGS. The steps below are intended to provide rough guidance for using HSM backed certificates. Consult your HSM vendor's documentation for exact steps and capabilities.

1. Install the HSM software on each HGS node in your cluster. Depending on whether you have a network or local HSM device, you may need to configure the HSM to grant your machine access to its key store.

2. Create 2 certificates in the HSM with **2048 bit RSA keys** for encryption and signing
   a. Create an encryption certificate with the **Data Encipherment** key usage property in your HSM
   b. Create a signing certificate with the **Digital Signature** key usage property in your HSM

3. Install the certificates in each HGS node's local certificate store per your HSM vendor's guidance.

4. If your HSM uses granular permissions to grant specific applications or users permission to use the private key, you will need to grant your HGS group managed service account access to the certificate. You can find the name of the HGS gMSA account by running `(Get-IISAppPool -Name KeyProtection).ProcessModel.UserName`

5. Add the signing and encryption certificates to HGS by replacing the thumbprints with those of your certificates' in the following commands:

   PowerShell

   ```
   Add-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
   "AABBCCDDEEFF00112233445566778899"
   ```

```
Add-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
"9988776655443322110OFFEEDDCCBBAA"
```

Option 2: Adding non-exportable software certificates

If you have a software-backed certificate issued by your company's or a public certificate authority that has a non-exportable private key, you will need to add your certificate to HGS using its thumbprint.

1. Install the certificate on your machine according to your certificate authority's instructions.

2. Grant the HGS group managed service account read-access to the private key of the certificate. You can find the name of the HGS gMSA account by running `(Get-IISAppPool -Name KeyProtection).ProcessModel.UserName`

3. Register the certificate with HGS using the following command and substituting in your certificate's thumbprint (change *Encryption* to *Signing* for signing certificates):

   PowerShell

   ```
   Add-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
   "AABBCCDDEEFF00112233445566778899"
   ```

> **ⓘ Important**
>
> You will need to manually install the private key and grant read access to the gMSA account on each HGS node. HGS cannot automatically replicate private keys for *any* certificate registered by its thumbprint.

Option 3: Adding certificates stored in PFX files

If you have a software-backed certificate with an exportable private key that can be stored in the PFX file format and secured with a password, HGS can automatically manage your certificates for you. Certificates added with PFX files are automatically replicated to every node of your HGS cluster and HGS secures access to the private keys. To add a new certificate using a PFX file, run the following commands on any HGS node (change *Encryption* to *Signing* for signing certificates):

PowerShell

```
$certPassword = Read-Host -AsSecureString -Prompt "Provide the PFX file
password"
```

```
Add-HgsKeyProtectionCertificate -CertificateType Encryption -CertificatePath
"C:\temp\encryptionCert.pfx" -CertificatePassword $certPassword
```

Identifying and changing the primary certificates While HGS can support multiple signing and encryption certificates, it uses one pair as its "primary" certificates. These are the certificates that will be used if someone downloads the guardian metadata for that HGS cluster. To check which certificates are currently marked as your primary certificates, run the following command:

PowerShell

```
Get-HgsKeyProtectionCertificate -IsPrimary $true
```

To set a new primary encryption or signing certificate, find the thumbprint of the desired certificate and mark it as primary using the following commands:

PowerShell

```
Get-HgsKeyProtectionCertificate
Set-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
"AABBCCDDEEFF00112233445566778899" -IsPrimary
Set-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
"99887766554433221100FFEEDDCCBBAA" -IsPrimary
```

Renewing or replacing keys

When you create the certificates used by HGS, the certificates will be assigned an expiration date according to your certificate authority's policy and your request information. Normally, in scenarios where the validity of the certificate is important such as securing HTTP communications, certificates must be renewed before they expire to avoid a service disruption or worrisome error message. HGS does not use certificates in that sense. HGS is simply using certificates as a convenient way to create and store an asymmetric key pair. An expired encryption or signing certificate on HGS does not indicate a weakness or loss of protection for shielded VMs. Further, certificate revocation checks are not performed by HGS. If an HGS certificate or issuing authority's certificate is revoked, it will not impact HGS' use of the certificate.

The only time you need to worry about an HGS certificate is if you have reason to believe that its private key has been stolen. In that case, the integrity of your shielded VMs is at risk because possession of the private half of the HGS encryption and signing key pair is enough to remove the shielding protections on a VM or stand up a fake HGS server that has weaker attestation policies.

If you find yourself in that situation, or are required by compliance standards to refresh certificate keys regularly, the following steps outline the process to change the keys on an HGS server. Please note that the following guidance represents a significant undertaking that will result in a disruption of service to each VM served by the HGS cluster. Proper planning for changing HGS keys is required to minimize service disruption and ensure the security of tenant VMs.

On an HGS node, perform the following steps to register a new pair of encryption and signing certificates. See the section on adding new keys for detailed information the various ways to add new keys to HGS.

1. Create a new pair of encryption and signing certificates for your HGS server. Ideally, these will be created in a hardware security module.

2. Register the new encryption and signing certificates with **Add-HgsKeyProtectionCertificate**

   PowerShell

   ```
   Add-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
   <Thumbprint>
   Add-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
   <Thumbprint>
   ```

3. If you used thumbprints, you'll need to go to each node in the cluster to install the private key and grant the HGS gMSA access to the key.

4. Make the new certificates the default certificates in HGS

   PowerShell

   ```
   Set-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
   <Thumbprint> -IsPrimary
   Set-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
   <Thumbprint> -IsPrimary
   ```

At this point, shielding data created with metadata obtained from the HGS node will use the new certificates, but existing VMs will continue to work because the older certificates are still there.

In order to ensure all existing VMs will work with the new keys, you will need to update the key protector on each VM.

This is an action that requires the VM owner (person or entity in possession of the "owner" guardian) to be involved. For each shielded VM, perform the following steps:

1. Shut down the VM. The VM cannot be turned back on until the remaining steps are complete or else you will need to start the process over again.

2. Save the current key protector to a file: `Get-VMKeyProtector -VMName 'VM001' | Out-File '.\VM001.kp'`

3. Transfer the KP to the VM owner

4. Have the owner download the updated guardian info from HGS and import it on their local system

5. Read the current KP into memory, grant the new guardian access to the KP, and save it to a new file by running the following commands:

```
PowerShell

$kpraw = Get-Content -Path .\VM001.kp
$kp = ConvertTo-HgsKeyProtector -Bytes $kpraw
$newGuardian = Get-HgsGuardian -Name 'UpdatedHgsGuardian'
$updatedKP = Grant-HgsKeyProtectorAccess -KeyProtector $kp -Guardian
$newGuardian
$updatedKP.RawData | Out-File .\updatedVM001.kp
```

6. Copy the updated KP back to the hosting fabric.

7. Apply the KP to the original VM:

```
PowerShell

$updatedKP = Get-Content -Path .\updatedVM001.kp
Set-VMKeyProtector -VMName VM001 -KeyProtector $updatedKP
```

8. Finally, start the VM and ensure it runs successfully.

> **Note**
>
> If the VM owner sets an incorrect key protector on the VM and does not authorize your fabric to run the VM, you will be unable to start up the shielded VM. To return to the last known good key protector, run `Set-VMKeyProtector -RestoreLastKnownGoodKeyProtector`

Once all VMs have been updated to authorize the new guardian keys, you can disable and remove the old keys.

9. Get the thumbprints of the old certificates from `Get-HgsKeyProtectionCertificate -IsPrimary $false`

10. Disable each certificate by running the following commands:

```
PowerShell

Set-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
<Thumbprint> -IsEnabled $false
Set-HgsKeyProtectionCertificate -CertificateType Encryption -Thumbprint
<Thumbprint> -IsEnabled $false
```

11. After ensuring VMs are still able to start with the certificates disabled, remove the certificates from HGS by running the following commands:

```
PowerShell

Remove-HgsKeyProtectionCertificate -CertificateType Signing -Thumbprint
<Thumbprint>`
Remove-HgsKeyProtectionCertificate -CertificateType Encryption -
Thumbprint <Thumbprint>`
```

> **ⓘ Important**
>
> VM backups will contain old key protector information that allow the old certificates to be used to start up the VM. If you are aware that your private key has been compromised, you should assume that the VM backups can be compromised, too, and take appropriate action. Destroying the VM configuration from the backups (.vmcx) will remove the key protectors, at the cost of needing to use the BitLocker recovery password to boot the VM the next time.

Key replication between nodes

Every node in the HGS cluster must be configured with the same encryption, signing, and (when configured) SSL certificates. This is necessary to ensure Hyper-V hosts reaching out to any node in the cluster can have their requests serviced successfully.

If you initialized HGS server with PFX-based certificates then HGS will automatically replicate both the public and private key of those certificates across every node in the cluster. You only need to add the keys on one node.

If you initialized HGS server with certificate references or thumbprints, then HGS will only replicate the *public* key in the certificate to each node. Additionally, HGS cannot

grant itself access to the private key on any node in this scenario. Therefore, it is your responsibility to:

1. Install the private key on each HGS node
2. Grant the HGS group managed service account (gMSA) access to the private key on each node These tasks add extra operational burden, however they are required for HSM-backed keys and certificates with non-exportable private keys.

SSL Certificates are never replicated in any form. It is your responsibility to initialize each HGS server with the same SSL certificate and update each server whenever you choose to renew or replace the SSL certificate. When replacing the SSL certificate, it is recommended that you do so using the Set-HgsServer cmdlet.

Unconfiguring HGS

If you need to decommission or significantly reconfigure an HGS server, you can do so using the Clear-HgsServer or Uninstall-HgsServer cmdlets.

Clearing the HGS configuration

To remove a node from the HGS cluster, use the Clear-HgsServer cmdlet. This cmdlet will make the following changes on the server where it is run:

- Unregisters the attestation and key protection services
- Removes the "microsoft.windows.hgs" JEA management endpoint
- Removes the local computer from the HGS failover cluster

If the server is the last HGS node in the cluster, the cluster and its corresponding Distributed Network Name resource will also be destroyed.

PowerShell

```
# Removes the local computer from the HGS cluster
Clear-HgsServer
```

After the clear operation completes, the HGS server can be re-initialized with Initialize-HgsServer. If you used Install-HgsServer to set up an Active Directory Domain Services domain, that domain will remain configured and operational after the clear operation.

Uninstalling HGS

If you wish to remove a node from the HGS cluster **and** demote the Active Directory Domain Controller running on it, use the Uninstall-HgsServer cmdlet. This cmdlet will make the following changes on the server where it is run:

- Unregisters the attestation and key protection services
- Removes the "microsoft.windows.hgs" JEA management endpoint
- Removes the local computer from the HGS failover cluster
- Demotes the Active Directory Domain Controller, if configured

If the server is the last HGS node in the cluster, the domain, failover cluster, and the cluster's Distributed Network Name resource will also be destroyed.

PowerShell

```
# Removes the local computer from the HGS cluster and demotes the ADDC
(restart required)
$newLocalAdminPassword = Read-Host -AsSecureString -Prompt "Enter a new
password for the local administrator account"
Uninstall-HgsServer -LocalAdministratorPassword $newLocalAdminPassword -
Restart
```

After the uninstall operation is complete and the computer has been restarted, you can reinstall ADDC and HGS using Install-HgsServer or join the computer to a domain and initialize the HGS server in that domain with Initialize-HgsServer.

If you no longer intend to use the computer as a HGS node, you can remove the role from Windows.

PowerShell

```
Uninstall-WindowsFeature HostGuardianServiceRole
```

Feedback

Was this page helpful? Yes No

Branch office considerations

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes best practices for running shielded virtual machines in branch offices and other remote scenarios where Hyper-V hosts may have periods of time with limited connectivity to HGS.

Fallback configuration

Starting with Windows Server version 1709, you can configure an additional set of Host Guardian Service URLs on Hyper-V hosts for use when your primary HGS is unresponsive. This allows you to run a local HGS cluster that is used as a primary server for better performance with the ability to fall back to your corporate datacenter's HGS if the local servers are down.

To use the fallback option, you'll need to set up two HGS servers. They can run Windows Server 2019 or Windows Server 2016 and either be part of the same or different clusters. If they are different clusters, you will want to establish operational practices to ensure the attestation policies are in sync between the two servers. They both need to be able to correctly authorize the Hyper-V host to run shielded VMs and have the key material needed to start up the shielded VMs. You can choose to either have a pair of shared encryption and signing certificates between the two clusters, or use separate certificates and configure the HGS shielded VM to authorize both guardians (encryption/signing certificate pairs) in the shielding data file.

Then upgrade your Hyper-V hosts to Windows Server version 1709 or Windows Server 2019 and run the following command:

PowerShell

```
# Replace https://hgs.primary.com and https://hgs.backup.com with your own
domain names and protocols
Set-HgsClientConfiguration -KeyProtectionServerUrl
'https://hgs.primary.com/KeyProtection' -AttestationServerUrl
'https://hgs.primary.com/Attestation' -FallbackKeyProtectionServerUrl
'https://hgs.backup.com/KeyProtection' -FallbackAttestationServerUrl
'https://hgs.backup.com/Attestation'
```

To unconfigure a fallback server, simply omit both fallback parameters:

PowerShell

```
Set-HgsClientConfiguration -KeyProtectionServerUrl
'https://hgs.primary.com/KeyProtection' -AttestationServerUrl
'https://hgs.primary.com/Attestation'
```

In order for the Hyper-V host to pass attestation with both the primary and fallback servers, you will need to ensure that your attestation information is up to date with both HGS clusters. Additionally, the certificates used to decrypt the virtual machine's TPM need to be available in both HGS clusters. You can configure each HGS with different certificates and configure the VM to trust both, or add a shared set of certificates to both HGS clusters.

For additional information about configuring HGS in a branch office using fallback URLs, see the blog post Improved branch office support for shielded VMs in Windows Server, version 1709.

Offline mode

Offline mode allows your shielded VM to turn on when HGS cannot be reached, so long as the security configuration of your Hyper-V host has not changed. Offline mode works by caching a special version of the VM TPM key protector on the Hyper-V host. The key protector is encrypted to the current security configuration of the host (using the Virtualization Based Security identity key). If your host is unable to communicate with HGS and its security configuration has not changed, it will be able to use the cached key protector to start up the shielded VM. When security settings change on the system, such as a new code integrity policy being applied or Secure Boot being disabled, the cached key protectors will be invalidated and the host will have to attest with an HGS before any shielded VMs can be started offline again.

Offline mode requires Insider Preview build 17609 or newer for both the Host Guardian Service cluster and Hyper-V host. It is controlled by a policy on HGS, which is disabled by default. To enable support for offline mode, run the following command on an HGS node:

PowerShell

```
Set-HgsKeyProtectionConfiguration -AllowKeyMaterialCaching:$true
```

Since the cacheable key protectors are unique to each shielded VM, you will need to fully shut down (not restart) and start up your shielded VMs to obtain a cacheable key protector after this setting is enabled on HGS. If your shielded VM migrates to a Hyper-

V host running an older version of Windows Server, or obtains a new key protector from an older version of HGS, it will not be able to start itself up in offline mode, but can continue running in online mode when access to HGS is available.

Feedback

Was this page helpful? Yes No

Upgrade a guarded fabric to Windows Server 2019

Article • 05/16/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This article describes the steps necessary to upgrade an existing guarded fabric from Windows Server 2016, Windows Server version 1709 or Windows Server version 1803 to Windows Server 2019.

What's new in Windows Server 2019

When you run a guarded fabric on Windows Server 2019, you can take advantage of several new features:

Host Key Attestation is our newest attestation mode, designed to make it easier to run shielded VMs when your Hyper-V hosts do not have TPM 2.0 devices available for TPM attestation. Host Key Attestation uses key pairs to authenticate hosts with HGS, removing the requirement for hosts to be joined to an Active Directory domain, eliminating the AD trust between HGS and the corporate forest, and reducing the number of open firewall ports. Host Key Attestation replaces Active Directory attestation, which is deprecated in Windows Server 2019.

V2 Attestation Version - To support Host Key Attestation and new features in the future, we've introduced versioning to HGS. A fresh install of HGS on Windows Server 2019 will result in the server using v2 attestation, which means it can support Host Key attestation for Windows Server 2019 hosts and still support v1 hosts on Windows Server 2016. In-place upgrades to 2019 will remain at version v1 until you manually enable v2. Most cmdlets now have a -HgsVersion parameter that lets you specify if you want to work with legacy or modern attestation policies.

Support for Linux shielded VMs - Hyper-V hosts running Windows Server 2019 can run Linux shielded VMs. While Linux shielded VMs have been around since Windows Server version 1709, Windows Server 2019 is the first Long Term Servicing Channel release to support them.

Branch office improvements - We've made it easier to run shielded VMs in branch offices with support for offline shielded VMs and fallback configurations on Hyper-V hosts.

TPM host binding - For the most secure workloads, where you want a shielded VM to run only on the first host where it was created but no other, you can now bind the VM to that host using the host's TPM. This is best used for privileged access workstations and branch offices, rather than general datacenter workloads that need to migrate between hosts.

Compatibility matrix

Before you upgrade your guarded fabric to Windows Server 2019, review the following compatibility matrix to see if your configuration is supported.

Expand table

	WS2016 HGS	**WS2019 HGS**
WS2016 Hyper-V Host	Supported	Supported[1]
WS2019 Hyper-V Host	Unsupported[2]	Supported

[1] Windows Server 2016 hosts can only attest against Windows Server 2019 HGS servers using the v1 attestation protocol. New features that are exclusively available in the v2 attestation protocol, including Host Key Attestation, are not supported for Windows Server 2016 hosts.

[2] Microsoft is aware of an issue preventing Windows Server 2019 hosts using TPM attestation from successfully attesting against a Windows Server 2016 HGS server. This limitation will be addressed in a future update for Windows Server 2016.

Upgrade HGS to Windows Server 2019

We recommend upgrading your HGS cluster to Windows Server 2019 before you upgrade your Hyper-V hosts to ensure that all hosts, whether they are running Windows Server 2016 or 2019, can continue to attest successfully.

Upgrading your HGS cluster will require you to temporarily remove one node from the cluster at a time while it is upgraded. This will reduce the capacity of your cluster to respond to requests from your Hyper-V hosts and could result in slow response times or service outages for your tenants. Ensure you have sufficient capacity to handle your attestation and key release requests before upgrading an HGS server.

To upgrade your HGS cluster, perform the following steps on each node of your cluster, one node at a time:

1. Remove the HGS server from your cluster by running `Clear-HgsServer` in an elevated PowerShell prompt. This cmdlet will remove the HGS replicated store, HGS websites, and node from the failover cluster.
2. If your HGS server is a domain controller (default configuration), you will need to run `adprep /forestprep` and `adprep /domainprep` on the first node being upgraded to prepare the domain for an OS upgrade. See the Active Directory Domain Services upgrade documentation for more information.
3. Perform an in-place upgrade to Windows Server 2019.
4. Run Initialize-HgsServer to join the node back to the cluster.

Once all nodes have been upgraded to Windows Server 2019, you can optionally upgrade the HGS version to v2 to support new features such as Host Key Attestation.

PowerShell

```
Set-HgsServerVersion  v2
```

Upgrade Hyper-V hosts to Windows Server 2019

Before you upgrade your Hyper-V hosts to Windows Server 2019, ensure that your HGS cluster is already upgraded to Windows Server 2019 and that you've moved all VMs off the Hyper-V server.

1. If you are using Windows Defender Application Control code integrity policies on your server (always the case when using TPM attestation), ensure that the policy is either in audit mode or disabled before attempting to upgrade the server. Learn how to disable a WDAC policy
2. Follow the guidance in the Windows Server upgrade content to upgrade your host to Windows Server 2019. If your Hyper-V host is part of a Failover Cluster, consider using a Cluster Operating System Rolling Upgrade.
3. Test and re-enable your Windows Defender Application Control policy, if you had one enabled before the upgrade.
4. Run `Get-HgsClientConfiguration` to check if **IsHostGuarded** = **True**, meaning the host is successfully passing attestation with your HGS server.
5. If you're using TPM attestation, you may need to re-capture the TPM baseline or code integrity policy after the upgrade to pass attestation.
6. Start running shielded VMs on the host again!

Switch to Host Key Attestation

Follow the steps below if you are currently running Active Directory-based attestation and want to upgrade to Host Key Attestation. Note that Active Directory-based attestation is deprecated in Windows Server 2019 and may be removed in a future release.

1. Ensure that your HGS server is operating in v2 attestation mode by running the following command. Existing v1 hosts will continue to attest even when the HGS server is upgraded to v2.

   ```
   PowerShell

   Set-HgsServerVersion v2
   ```

2. Generate host keys from each of your Hyper-V hosts and register them with HGS. Because HGS is still operating in Active Directory mode, you will receive a warning that the new host keys are not immediately effective. This is intentional, as you do not want to change to host key mode until all of your hosts can attest with host keys successfully.

3. Once host keys have been registered for every host, you can configure HGS to use host key attestation mode:

   ```
   PowerShell

   Set-HgsServer -TrustHostKey
   ```

   If you run into trouble with host key mode and need to revert back to Active Directory-based attestation, run the following command on HGS:

   ```
   PowerShell

   Set-HgsServer -TrustActiveDirectory
   ```

Feedback

Was this page helpful? Yes No

Device Health Attestation

Article • 10/25/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Introduced in Windows 10, version 1507, Device Health Attestation (DHA) included the following:

- Integrates with Windows 10 Mobile Device Management (MDM) framework in alignment with Open Mobile Alliance (OMA) standards.
- Supports devices that have a Trusted Module Platform (TPM) provisioned in a firmware or discrete format.
- Enables enterprises to raise the security bar of their organization to hardware monitored and attested security, with minimal or no impact on operation cost.

Starting with Windows Server 2016, you can now run the DHA service as a server role within your organization. Use this article to learn how to install and configure the Device Health Attestation server role.

Overview

You can use DHA to assess device health for:

- Windows 10 and Windows 10 Mobile devices that support TPM 1.2 or 2.0.
- On-premises devices that are managed by using Active Directory with Internet access, devices that are managed by using Active Directory without Internet access, devices managed by Microsoft Entra ID, or a hybrid deployment using both Active Directory and Microsoft Entra ID.

DHA service

The DHA service validates the TPM and PCR logs for a device and then issues a DHA report. Microsoft offers the DHA service in three ways:

- **DHA cloud service**. A Microsoft-managed DHA service that is free, geo-load-balanced, and optimized for access from different regions of the world.
- **DHA on-premises service**. A new server role introduced in Windows Server 2016. It's available for free to customers that have a Windows Server 2016 license.

- **DHA Azure cloud service**. A virtual host in Microsoft Azure. To do this, you need a virtual host and licenses for the DHA on-premises service.

The DHA service integrates with MDM solutions and provides the following:

- Combine the info they receive from devices (through existing device management communication channels) with the DHA report
- Make a more secure and trusted security decision, based on hardware attested and protected data

Here's an example that shows how you can use DHA to help raise the security protection bar for your organization's assets.

1. You create a policy that checks the following boot configuration/attributes:

    - Secure Boot
    - BitLocker
    - ELAM

2. The MDM solution enforces this policy and triggers a corrective action based on the DHA report data. For example, it could verify the following:

    - Secure Boot was enabled, the device loaded trusted code that is authentic, and the Windows boot loader was not tampered with.
    - Trusted Boot successfully verified the digital signature of the Windows kernel and the components that were loaded while the device started.
    - Measured Boot created a TPM-protected audit trail that could be verified remotely.
    - BitLocker was enabled and that it protected the data when the device was turned off.
    - ELAM was enabled at early boot stages and is monitoring the runtime.

DHA cloud service

The DHA cloud service provides the following benefits:

- Reviews the TCG and PCR device boot logs it receives from a device that is enrolled with an MDM solution.
- Creates a tamper resistant and tamper evident report (DHA report) that describes how the device started based on data that is collected and protected by a device's TPM chip.
- Delivers the DHA report to the MDM server that requested the report in a protected communication channel.

DHA on-premises service

The DHA on-premises service offer all the capabilities that are offered by DHA cloud service. It also enables customers to:

- Optimize performance by running DHA service in your own data center
- Ensure that the DHA report does not leave your network

DHA Azure cloud service

This service provides the same functionality as the DHA on-premises service, except that the DHA Azure cloud service runs as a virtual host in Microsoft Azure.

DHA validation modes

You can set up the DHA on-premises service to run in either EKCert or AIKCert validation mode. When the DHA service issues a report, it indicates if it was issued in AIKCert or EKCert validation mode. AIKCert and EKCert validation modes offer the same security assurance as long as the EKCert chain of trust is kept up-to-date.

EKCert validation mode

EKCert validation mode is optimized for devices in organizations that are not connected to the Internet. Devices connecting to a DHA service running in EKCert validation mode do **not** have direct access to the Internet.

When DHA is running in EKCert validation mode, it relies on an enterprise managed chain of trust that needs to be updated occasionally (approximately 5 - 10 times per year).

Microsoft publishes aggregated packages of trusted Roots and intermediate CAs for approved TPM manufacturers (as they become available) in a publicly accessible archive in .cab archive. You need to download the feed, validate its integrity, and install it on the server running Device Health Attestation.

An example archive is https://go.microsoft.com/fwlink/?linkid=2097925.

AIKCert validation mode

AIKCert Validation Mode is optimized for operational environments that do have access to the Internet. Devices connecting to a DHA service running in AIKCert validation mode

must have direct access to the Internet and are able to get an AIK certificate from Microsoft.

Install and configure the DHA service on Windows Server 2016

Use the following sections to get DHA installed and configured on Windows Server 2016.

Prerequisites

In order to set up and verify a DHA on-premises service, you need:

- A server running Windows Server 2016.
- One (or more) Windows 10 client devices with a TPM (either 1.2 or 2.0) that is in a clear/ready state running the latest Windows Server 2025 build.
- Decide if you are going to run in EKCert or AIKCert validation mode.
- The following certificates:
  - **DHA SSL certificate**. A x.509 SSL certificate that chains to an enterprise trusted root with an exportable private key. This certificate protects DHA data communications in transit including server to server (DHA service and MDM server) and server to client (DHA service and a Windows 10 device) communications.
  - **DHA signing certificate**. A x.509 certificate that chains to an enterprise trusted root with an exportable private key. The DHA service uses this certificate for digital signing.
  - **DHA encryption certificate**. A x.509 certificate that chains to an enterprise trusted root with an exportable private key. The DHA service also uses this certificate for encryption.

Install Windows Server 2016

Install Windows Server 2016 using your preferred installation method, such as Windows Deployment Services, or running the installer from bootable media, a USB drive, or the local file system. If this is the first time you are configuring the DHA on-premises service, you should install Windows Server 2016 using the **Desktop Experience** installation option.

Add the Device Health Attestation server role

You can install the Device Health Attestation server role and its dependencies by using Server Manager.

After you've installed Windows Server 2016, the device restarts and opens Server Manager. If Server manager doesn't start automatically, click **Start**, and then click **Server Manager**.

1. Click **Add roles and features**.
2. On the **Before you begin** page, click **Next**.
3. On the **Select installation type** page, click **Role-based or feature-based installation**, and then click **Next**.
4. On the **Select destination server** page, click **Select a server from the server pool**, select the server, and then click **Next**.
5. On the **Select server roles** page, select the **Device Health Attestation** check box.
6. Click **Add Features** to install other required role services and features.
7. Click **Next**.
8. On the **Select features** page, click **Next**.
9. On the **Web Server Role (IIS)** page, click **Next**.
10. On the **Select role services** page, click **Next**.
11. On the **Device Health Attestation Service** page, click **Next**.
12. On the **Confirm installation selections** page, click **Install**.
13. When the installation is done, click **Close**.

Install the signing and encryption certificates

Using the following Windows PowerShell script to install the signing and encryption certificates. For more information about the thumbprint, see How to: Retrieve the Thumbprint of a Certificate.

```
$key = Get-ChildItem Cert:\LocalMachine\My | Where-Object {$_.Thumbprint -
like "<thumbprint>"}
$keyname = $key.PrivateKey.CspKeyContainerInfo.UniqueKeyContainerName
$keypath = $env:ProgramData + "\Microsoft\Crypto\RSA\MachineKeys\" +
$keyname
icacls $keypath /grant <username>`:R

#<thumbprint>: Certificate thumbprint for encryption certificate or signing
certificate
#<username>: Username for web service app pool, by default IIS_IUSRS
```

Install the trusted TPM roots certificate package

To install the trusted TPM roots certificate package, you must extract it, remove any trusted chains that are not trusted by your organization, and then run setup.cmd.

Download the trusted TPM roots certificate package

Before you install the certificate package, you can download the latest list of trusted TPM roots from https://go.microsoft.com/fwlink/?linkid=2097925.

> **Important:** Before installing the package, verify that it is digitally signed by Microsoft.

Extract the trusted certificate package

Extract the trusted certificate package by running the following commands.

```
mkdir .\TrustedTpm
expand -F:* .\TrustedTpm.cab .\TrustedTpm
```

Remove the trust chains for TPM vendors that are *not* trusted by your organization (Optional)

Delete the folders for any TPM vendor trust chains that are not trusted by your organization.

> **Note:** If using AIK Certificate mode, the Microsoft folder is required to validate Microsoft issued AIK certificates.

Install the trusted certificate package

Install the trusted certificate package by running the setup script from the .cab file.

```
.\setup.cmd
```

Configure the Device Health Attestation service

You can use Windows PowerShell to configure the DHA on-premises service.

```
Install-DeviceHealthAttestation -EncryptionCertificateThumbprint
<encryption> -SigningCertificateThumbprint <signing> -
SslCertificateStoreName My -SslCertificateThumbprint <ssl> -
SupportedAuthenticationSchema "<schema>"

#<encryption>: Thumbprint of the encryption certificate
#<signing>: Thumbprint of the signing certificate
#<ssl>: Thumbprint of the SSL certificate
#<schema>: Comma-delimited list of supported schemas including
AikCertificate, EkCertificate, and AikPub
```

Configure the certificate chain policy

Configure the certificate chain policy by running the following Windows PowerShell script:

```
$policy = Get-DHASCertificateChainPolicy
$policy.RevocationMode = "NoCheck"
Set-DHASCertificateChainPolicy -CertificateChainPolicy $policy
```

DHA management commands

Here are some Windows PowerShell examples that can help you manage the DHA service.

Configure the DHA service for the first time

```
Install-DeviceHealthAttestation -SigningCertificateThumbprint "<HEX>" -
EncryptionCertificateThumbprint "<HEX>" -SslCertificateThumbprint "<HEX>" -
Force
```

Remove the DHA service configuration

```
```

```
Uninstall-DeviceHealthAttestation -RemoveSslBinding -Force
```

Get the active signing certificate

```

Get-DHASActiveSigningCertificate
```

Set the active signing certificate

```

Set-DHASActiveSigningCertificate -Thumbprint "<hex>" -Force
```

> **Note:** This certificate must be deployed on the server running the DHA service in the **LocalMachine\My** certificate store. When the active signing certificate is set, the existing active signing certificate is moved to the list of inactive signing certificates.

List the inactive signing certificates

```

Get-DHASInactiveSigningCertificate
```

Remove any inactive signing certificates

```

Remove-DHASInactiveSigningCertificate -Force
Remove-DHASInactiveSigningCertificate  -Thumbprint "<hex>" -Force
```

> **Note:** Only *one* inactive certificate (of any type) may exist in the service at any time. Certificates should be removed from the list of inactive certificates once they are no longer required.

Get the active encryption certificate

```

```

```
Get-DHASActiveEncryptionCertificate
```

Set the active encryption certificate

```
Set-DHASActiveEncryptionCertificate -Thumbprint "<hex>" -Force
```

The certificate must be deployed on the device in the **LocalMachine\My** certificate store.

When the active encryption certificate is set, the existing active encryption certificate is moved to the list of inactive encryption certificates.

List the inactive encryption certificates

```
Get-DHASInactiveEncryptionCertificate
```

Remove any inactive encryption certificates

```
Remove-DHASInactiveEncryptionCertificate -Force
Remove-DHASInactiveEncryptionCertificate -Thumbprint "<hex>" -Force
```

Get the X509ChainPolicy configuration

```
Get-DHASCertificateChainPolicy
```

Change the X509ChainPolicy configuration

```
$certificateChainPolicy = Get-DHASInactiveEncryptionCertificate
$certificateChainPolicy.RevocationFlag = <X509RevocationFlag>
```

```
$certificateChainPolicy.RevocationMode = <X509RevocationMode>
$certificateChainPolicy.VerificationFlags = <X509VerificationFlags>
$certificateChainPolicy.UrlRetrievalTimeout = <TimeSpan>
Set-DHASCertificateChainPolicy = $certificateChainPolicy
```

DHA service reporting

The following are a list of messages that are reported by the DHA service to the MDM solution:

- **200** HTTP OK. The certificate is returned.
- **400** Bad request. Invalid request format, invalid health certificate, certificate signature does not match, invalid Health Attestation Blob, or an invalid Health Status Blob. The response also contains a message, as described by the response schema, with an error code and an error message that can be used for diagnostics.
- **500** Internal server error. This can happen if there are issues that prevent the service from issuing certificates.
- **503** Throttling is rejecting requests to prevent server overloading.

Feedback

Was this page helpful? Yes No

Guidance on disabling system services on Windows Server 2016 with Desktop Experience

Article • 08/22/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows operating systems (OSes) include many important system services. Different services have different default startup policies:

- Automatic, when the service starts by default
- Manual, when the user activates the service
- Disabled by default, which means the user must enable the service before it can run

Microsoft chose these default settings carefully for each service to balance performance, functionality, and security for customers.

However, a more security-focused deployment may require changing these settings. For example, you might want to disable all programs that you don't need to run by default. The guidance in this article describes how to change these settings to make your deployment more secure.

The guidance in this article only applies to Windows Server 2016 with Desktop Experience, unless you're using it as a desktop replacement for end-users. As of Windows Server 2019, the configurations in this article are actually the default settings in the OS.

Each service is sorted into the following categories:

- **Should Disable**, which means you should disable this service and not use it to make your system more secure.
- **OK to Disable**, which means that while you don't have to disable this service for security reasons, you can still safely do so if you don't need it.
- **Do Not Disable**, which means disabling this service impacts essential system functions or prevents specific roles or features from working as intended. You must always keep this service enabled.

- **(No guidance)**, for some services, the impact of disabling hasn't been fully evaluated, so it's recommended to leave them at their default configuration.

You can configure your Windows devices and servers to disable selected services by using Security Templates in Group Policies or by running PowerShell cmdlets. In some cases, this guide describes specific Group Policy settings you can change to disable services directly.

We recommend you disable the following services and their related scheduled tasks on Windows Server 2016 with Desktop Experience:

- Xbox Live Auth Manager
  - \Microsoft\XblGameSave\XblGameSaveTask
- Xbox Live Game Save
  - \Microsoft\XblGameSave\XblGameSaveTaskLogon

Disabling non-default services

We don't recommend applying policies that disable services that aren't by default by Microsoft.

There are a few reasons why we recommend against disabling non-default services:

- A service your admins installed after the initial setup is usually one that you need but wasn't in the original deployment. In these cases, you should tell your admins not to install the service in the first place as opposed to only disabling it after it's installed.
- Blocking the Windows version of a service doesn't stop admins or non-admins from installing third-party equivalents that pose higher security risks.
- Establishing a baseline that disables non-default Windows services, such as the World Wide Web Publishing Service (W2SVC) or Internet Information Services (IIS), may give auditors the impression that they're inherently insecure and should never be used.
- If the service is never installed, disabling it adds unnecessary work to establishing the baseline and verification for your system.

How to read this guide

This article describes each service one by one, using the following variables.

- The *service name* is the internal name of the service.
- The *description* describes what each service is.
- *Installation* describes which OS the service exists on.
  - If the service is *always installed*, it's available on both Windows Server 2016 Core and Windows Server 2016 with Desktop Experience.
  - If the service is *only with Desktop Experience*, it isn't available on Windows Server 2016 Core.
- The *startup type* is how Windows determines when and how the service is started.
- *Recommendation* lists Microsoft's recommended configuration for this service on a typical Windows Server 2016 deployment in an enterprise that's not using the server as an end-user desktop replacement.
  - *Do not disable* means you should never disable this service.
  - *OK to disable* means you can disable this service if you don't need it.
  - *Already disabled* means Windows disabled this service by default.
  - *Should be disabled* means you should always disable this service to increase security.
- The *comments* show extra explanations and notes if you need more context.

Default local services

The following tables describes the default services found in Windows Server 2016. To edit the configuration of these services, select **Start**, enter **services.msc**, then select **Services**.

ActiveX Installer (AxInstSV)

Expand table

Name	Description
Service name	AxInstSV
Description	Provides User Account Control validation for the installation of ActiveX controls from the Internet and enables management of ActiveX control installation based on Group Policy settings. This service is started on demand,

Name	Description
	and, if disabled, the installation of ActiveX controls behave according to default browser settings.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	OK to disable if feature not needed

AllJoyn Router Service

Expand table

Name	Description
Service name	AJRouter
Description	Routes AllJoyn messages for the local AllJoyn clients. If you stop this service, the AllJoyn clients that don't have their own bundled routers are unable to run.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

App Readiness

Expand table

Name	Description
Service name	AppReadiness
Description	Gets apps ready for use the first time a user signs in to this PC and when adding new apps.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	Do not disable

Name	Description
Comments	None

Application Identity

Expand table

Name	Description
Service name	AppIDSvc
Description	Determines and verifies the identity of an application. Disabling this service prevents AppLocker from being enforced.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Application Information

Expand table

Name	Description
Service name	Appinfo
Description	Facilitates the running of interactive applications with additional administrative privileges. If you stop this service, users can't launch applications with the additional administrative privileges they may require to perform desired user tasks.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	Supports UAC same-desktop elevation

Application Layer Gateway Service

Expand table

Name	Description
Service name	ALG
Description	Provides support for third-party protocol plug-ins for Internet Connection Sharing.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Application Management

Expand table

Name	Description
Service name	AppMgmt
Description	Processes installation, removal, and enumeration requests for software deployed through Group Policy. If the service is disabled, users are unable to install, remove, or enumerate software deployed through Group Policy, and any services that explicitly depend on Application Management are unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

AppX Deployment Service (AppXSVC)

Expand table

Name	Description
Service name	AppXSvc

Name	Description
Description	Provides infrastructure support for deploying Store applications. This service starts on demand and, if disabled, Store applications don't deploy to the system and may not function properly.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Auto Time Zone Updater

Expand table

Name	Description
Service name	tzautoupdate
Description	Automatically sets the system time zone.
Installation	Only with Desktop Experience
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Background Intelligent Transfer Service

Expand table

Name	Description
Service name	BITS
Description	Transfers files in the background using idle network bandwidth. If the service is disabled, then any applications that depend on BITS, such as Windows Update or MSN Explorer, are unable to automatically download programs and other information.
Installation	Always installed
Startup type	Manual

Name	Description
Recommendation	No guidance
Comments	None

Background Tasks Infrastructure Service

Expand table

Name	Description
Service name	BrokerInfrastructure
Description	Windows infrastructure service that controls which background tasks can run on the system.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	No guidance
Comments	None

Base Filtering Engine

Expand table

Name	Description
Service name	BFE
Description	The Base Filtering Engine (BFE) is a service that manages firewall and Internet Protocol security (IPsec) policies and implements user mode filtering. Stopping or disabling the BFE service significantly reduces system security and causes unpredictable behavior in IPsec management and firewall applications.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Bluetooth Support Service

Expand table

Name	Description
Service name	bthserv
Description	The Bluetooth service supports discovery and association of remote Bluetooth devices. Stopping or disabling this service may cause already installed Bluetooth devices to fail to operate properly and prevent new devices from being discovered or associated.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	OK to disable if not used. Another disabling mechanism: Disabling Bluetooth and Infrared Beaming

CDPUserSvc

Expand table

Name	Description
Service name	CDPUserSvc
Description	This user service is used for Connected Devices Platform scenarios.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	OK to disable
Comments	User service template

Certificate Propagation

Expand table

Name	Description
Service name	CertPropSvc

Name	Description
Description	Copies user certificates and root certificates from smart cards into the current user's certificate store, detects when a smart card is inserted into a smart card reader, and, if needed, installs the smart card Plug and Play minidriver.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Client License Service (ClipSVC)

Expand table

Name	Description
Service name	ClipSVC
Description	Provides infrastructure support for the Microsoft Store. This service starts on demand, and, if disabled, applications bought using Microsoft Store don't behave correctly.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

CNG Key Isolation

Expand table

Name	Description
Service name	KeyIso
Description	The CNG key isolation service is hosted in the LSA process. The service provides key process isolation to private keys and associated cryptographic operations, as required by the Common Criteria. The service stores and uses long-lived keys in a secure process that complies with Common Criteria requirements.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

COM+ Event System

Expand table

Name	Description
Service name	EventSystem
Description	Supports System Event Notification Service (SENS), which provides automatic distribution of events to subscribing Component Object Model (COM) components. If the service is stopped, SENS closes and is unable to provide logon and logoff notifications. If you disable this service, any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

COM+ System Application

Expand table

Name	Description
Service name	COMSysApp
Description	Manages the configuration and tracking of Component Object Model (COM)-based components. If the service stops, most COM+-based components stop functioning properly. If you disable this service, any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Manual

Name	Description
Recommendation	No guidance
Comments	None

Computer Browser

Expand table

Name	Description
Service name	Browser
Description	Maintains an updated list of computers on the network and supplies this list to computers designated as browsers. If this service stops, this list doesn't update or maintain itself, and any services that explicitly depend on Computer Browser are unable to start.
Installation	Always installed
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Connected Devices Platform Service

Expand table

Name	Description
Service name	CDPSvc
Description	This service is used for Connected Devices and Universal Glass scenarios.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	No guidance
Comments	None

Connected User Experiences and Telemetry

Expand table

Name	Description
Service name	DiagTrack
Description	The Connected User Experiences and Telemetry service enables features that support in-application and connected user experiences. When the diagnostics and usage privacy option settings are enabled under Feedback and Diagnostics, this service also manages the event-driven collection and transmission of diagnostic and usage information that Microsoft uses to improve the experience and quality of the Windows Platform.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Contact Data

Expand table

Name	Description
Service name	PimIndexMaintenanceSvc
Description	Indexes contact data for fast contact searching. If you stop or disable this service, contacts might be missing from your search results.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	User service template

CoreMessaging

Expand table

Name	Description
Service name	CoreMessagingRegistrar

Name	Description
Description	Manages communication between system components.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Credential Manager

Expand table

Name	Description
Service name	VaultSvc
Description	Provides secure storage and retrieval of credentials to users, applications, and security service packages.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Cryptographic Services

Expand table

Name	Description
Service name	CryptSvc
Description	Provides three management services: Catalog Database Service, which confirms the signatures of Windows files and allows new programs to be installed; Protected Root Service, which adds and removes Trusted Root Certification Authority certificates from this computer; and Automatic Root Certificate Update Service, which retrieves root certificates from Windows Update and enable scenarios such as SSL. If you stop this service, these management services are unable to function properly and any services that explicitly depend on it are unable to start.

Name	Description
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Data Sharing Service

Expand table

Name	Description
Service name	DsSvc
Description	Provides data brokering between applications.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

DataCollectionPublishingService

Expand table

Name	Description
Service name	DcpSvc
Description	The Data Collection and Publishing (DCP) service supports first-party apps to upload data to cloud.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

DCOM Server Process Launcher

Expand table

Name	Description
Service name	DcomLaunch
Description	The DCOMLAUNCH service launches Component Object Model (COM) and Distributed Component Object Model (DCOM) servers in response to object activation requests. If you stop or disable this service, programs using COM or DCOM are unable to function properly. We highly recommend that you let the DCOMLAUNCH service run.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Device Association Service

Expand table

Name	Description
Service name	DeviceAssociationService
Description	Enables pairing between the system and wired or wireless devices.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Device Install Service

Expand table

Name	Description
Service name	DeviceInstall

Name	Description
Description	Enables a computer to recognize and adapt to hardware changes with little or no user input. Stopping or disabling this service results in system instability.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Device Management Enrollment Service

Expand table

Name	Description
Service name	DmEnrollmentSvc
Description	Performs Device Enrollment Activities for Device Management
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Device Setup Manager

Expand table

Name	Description
Service name	DsmSvc
Description	Enables the detection, download, and installation of device-related software. If you disable this service, devices may be configured with outdated software and not work correctly.
Installation	Only with Desktop Experience
Startup type	Manual

Name	Description
Recommendation	No guidance
Comments	None

DevQuery Background Discovery Broker

Expand table

Name	Description
Service name	DevQueryBroker
Description	Enables apps to discover devices with a background task.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

DHCP Client

Expand table

Name	Description
Service name	Dhcp
Description	Registers and updates IP addresses and DNS records for this computer. If you stop this service, this computer doesn't receive dynamic IP addresses and DNS updates. If you disable this service, then any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Diagnostic Policy Service

Expand table

Name	Description
Service name	DPS
Description	The Diagnostic Policy Service enables problem detection, troubleshooting, and resolution for Windows components. If you stop this service, diagnostics stops functioning.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Diagnostic Service Host

Expand table

Name	Description
Service name	WdiServiceHost
Description	The Diagnostic Service Host is used by the Diagnostic Policy Service to host diagnostics that need to run in a Local Service context. If you stop this service, any diagnostics that depend on it stop functioning.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Diagnostic System Host

Expand table

Name	Description
Service name	WdiSystemHost
Description	The Diagnostic Policy Service uses the Diagnostic System host to host diagnostics that need to run in a Local System context. If you stop this

Name	Description
	service, any diagnostics that depend on it stop functioning.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Distributed Link Tracking Client

Expand table

Name	Description
Service name	TrkWks
Description	Maintains links between NTFS files within a computer or across computers in a network.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	No guidance
Comments	None

Distributed Transaction Coordinator

Expand table

Name	Description
Service name	MSDTC
Description	Coordinates transactions that span multiple resource managers, such as databases, message queues, and file systems. Stopping this service blocks these transactions from happening. If you disable this service, any services that explicitly depend on it can't start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance

Name	Description
Comments	None

dmwappushsvc

Expand table

Name	Description
Service name	dmwappushservice
Description	Wireless Application Protocol (WAP) Push Message Routing Service
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Service required on client devices for Intune, Mobile Device Management (MDM), and similar management technologies, and for Unified Write Filter. Not needed for Server.

DNS Client

Expand table

Name	Description
Service name	Dnscache
Description	The DNS Client service (dnscache) caches Domain Name System (DNS) names and registers the full computer name for this computer. If you stop this service, the dnscache continues to resolve DNS names. However, the results of DNS name queries aren't cached, so the computer's name isn't registered. If you disable this service, any services that explicitly depend on it don't start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Downloaded Maps Manager

Expand table

Name	Description
Service name	MapsBroker
Description	Windows service for application access to downloaded maps. This service is started on-demand by applications accessing downloaded maps. Disabling this service stops apps from accessing maps.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	OK to disable
Comments	Disabling this service breaks apps that rely on it; OK to disable if apps not relying on it

Embedded Mode

Expand table

Name	Description
Service name	embeddedmode
Description	The Embedded Mode service enables scenarios related to Background Applications. Disabling this service prevents Background Applications from activating.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Encrypting File System (EFS)

Expand table

Name	Description
Service name	EFS
Description	Provides the core file encryption technology used to store encrypted files on NTFS file system volumes. If you stop or disable this service, applications can't access encrypted files.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Enterprise App Management Service

Expand table

Name	Description
Service name	EntAppSvc
Description	Enables enterprise application management.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Extensible Authentication Protocol

Expand table

Name	Description
Service name	EapHost
Description	The Extensible Authentication Protocol (EAP) service provides network authentication in such scenarios as 802.1x wired and wireless, VPN, and Network Access Protection (NAP). EAP also provides the application programming interfaces (APIs) that network access clients use during the authentication process, including wireless and Virtual Private Network (VPN)

Name	Description
	clients. If you disable this service, the computer can't access networks requiring EAP authentication.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Function Discovery Provider Host

Expand table

Name	Description
Service name	fdPHost
Description	The Function Discovery Provider Host (FDPHOST) service hosts the Function Discovery (FD) network discovery providers. These FD providers supply network discovery services for the Simple Services Discovery Protocol (SSDP) and Web Services - Discovery (WS-D) protocol. Stopping or disabling the FDPHOST service disables network discovery for these protocols when using FD. When this service is unavailable, network services using FD and relying on these discovery protocols are unable to find network devices or resources.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Function Discovery Resource Publication

Expand table

Name	Description
Service name	FDResPub
Description	Publishes this computer and resources attached to this computer so that other computers can discover them over the network. If you stop this service,

Name	Description
	network resources are no longer published and can't be discovered by other computers on the network.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Geolocation Service

Expand table

Name	Description
Service name	lfsvc
Description	This service monitors the current location of the system and manages geofences (a geographical location with associated events). If you turn off this service, applications can't use or receive notifications for geolocation or geofences.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Disabling breaks apps that rely on the service; OK to disable if apps not relying on it

Group Policy Client

Expand table

Name	Description
Service name	gpsvc
Description	The service is responsible for applying settings that administrators configured for the computer and users through the Group Policy component. If you disable the service, the settings aren't applied and applications and Group Policy can't manage the components. Any components or applications that depend on the Group Policy component might not function as intended.

Name	Description
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Human Interface Device Service

Expand table

Name	Description
Service name	hidserv
Description	Activates and maintains the use of hot buttons on keyboards, remote controls, and other multimedia devices. We recommend you keep this service running.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

HV Host Service

Expand table

Name	Description
Service name	HvHost
Description	Provides an interface for the Hyper-V hypervisor to provide per-partition performance counters to the host operating system.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Performance enhancers for guest VMs. Not used today except for explicitly

Name	Description
	populated VMs, but Application Guard can also use it.

Hyper-V Data Exchange Service

Expand table

Name	Description
Service name	vmickvpexchange
Description	Provides a mechanism to exchange data between the virtual machine and the operating system running on the physical computer.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Guest Service Interface

Expand table

Name	Description
Service name	vmicguestinterface
Description	Provides an interface for the Hyper-V host to interact with specific services running inside the virtual machine.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Guest Shutdown Service

Expand table

Name	Description
Service name	vmicshutdown
Description	Lets you shut down the operating system of this virtual machine from the management interfaces on the physical computer.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Heartbeat Service

Expand table

Name	Description
Service name	vmicheartbeat
Description	Monitors the state of this virtual machine by reporting a heartbeat at regular intervals. This service helps you identify running virtual machines that have stopped responding.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V PowerShell Direct Service

Expand table

Name	Description
Service name	vmicvmsession
Description	Lets you manage virtual machines with PowerShell by using a VM session without a virtual network.
Installation	Always installed

Name	Description
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Remote Desktop Virtualization Service

Expand table

Name	Description
Service name	vmicrdv
Description	Provides a platform for communication between the virtual machine and the operating system running on the physical computer.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Time Synchronization Service

Expand table

Name	Description
Service name	vmictimesync
Description	Synchronizes the system time of this virtual machine with the system time of the physical computer.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

Hyper-V Volume Shadow Copy Requestor

Expand table

Name	Description
Service name	vmicvss
Description	Coordinates communications required to use Volume Shadow Copy Service to back up applications and data on this virtual machine from the operating system on the physical computer.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	See HvHost

IKE and AuthIP IPsec Keying Modules

Expand table

Name	Description
Service name	IKEEXT
Description	The IKEEXT service hosts the Internet Key Exchange (IKE) and Authenticated Internet Protocol (AuthIP) keying modules. The system uses these keying modules for authentication and key exchange in Internet Protocol security (IPsec). Stopping or disabling the IKEEXT service disables the IKE and AuthIP key exchange with peer computers. IPsec is typically configured to use IKE or AuthIP; therefore, stopping or disabling the IKEEXT service might result in an IPsec failure, which compromises system security. We highly recommend that you leave the IKEEXT service running.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Interactive Services Detection

Expand table

Name	Description
Service name	UI0Detect
Description	Enables user notification of user input for interactive services, which enables access to dialogs created by interactive services when they appear. If you stop this service, notifications of new interactive service dialogs stop functioning, which can also prevent access to interactive service dialogs. If you disable this service, both notifications of and access to new interactive service dialogs stop working.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Internet Connection Sharing (ICS)

Expand table

Name	Description
Service name	SharedAccess
Description	Provides network address translation, addressing, name resolution, and intrusion prevention services for a home or small office network.
Installation	Always installed
Startup type	Manual
Recommendation	OK to disable
Comments	Required for clients used as WiFi hotspots and both ends of Miracast projection. You can block ICS with the GPO setting called **Prohibit use of Internet Connection Sharing on your DNS domain network**.

IP Helper

Expand table

Name	Description
Service name	iphlpsvc

Name	Description
Description	Provides tunnel connectivity using IPv6 transition technologies (6to4, ISATAP, Port Proxy, and Teredo) and IP-HTTPS. If you stop this service, the computer loses the enhanced connectivity benefits that these technologies offer.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

IPsec Policy Agent

Expand table

Name	Description
Service name	PolicyAgent
Description	Internet Protocol security (IPsec) supports network-level peer authentication, data origin authentication, data integrity, data confidentiality (encryption), and replay protection. This service enforces IPsec policies created through the IP Security Policies snap-in or the `netsh ipsec` command-line tool. If you stop this service, you may experience network connectivity issues if your policy requires that connections use IPsec. Also, remote management of Windows Firewall isn't available when this service isn't running.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

KDC Proxy Server service (KPS)

Expand table

Name	Description
Service name	KPSSVC
Description	KDC Proxy Server service runs on edge servers to proxy Kerberos protocol messages to domain controllers on the corporate network.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

KtmRm for Distributed Transaction Coordinator

Expand table

Name	Description
Service name	KtmRm
Description	Coordinates transactions between the Distributed Transaction Coordinator (MSDTC) and the Kernel Transaction Manager (KTM). If you don't need this service, we recommend that you stop this service. If you do need it, both MSDTC and KTM start this service automatically. If you disable this service, any MSDTC transaction interacting with a Kernel Resource Manager doesn't work and any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Link-Layer Topology Discovery Mapper

Expand table

Name	Description
Service name	lltdsvc
Description	Creates a Network Map consisting of PC and device topology (connectivity) information and metadata describing each PC and device. If you disable this service, the Network Map doesn't function properly.
Installation	Always installed
Startup type	Manual

Name	Description
Recommendation	OK to disable
Comments	OK to disable if no dependencies on Network Map

Local Session Manager

Expand table

Name	Description
Service name	LSM
Description	Core Windows Service that manages local user sessions. Stopping or disabling this service causes system instability.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Microsoft (R) Diagnostics Hub Standard Collector

Expand table

Name	Description
Service name	diagnosticshub.standardcollector.service
Description	Diagnostics Hub Standard Collector Service. When running, this service collects real-time Event Tracing for Windows (ETW) events and processes them.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Microsoft Account Sign-in Assistant

Expand table

Name	Description
Service name	wlidsvc
Description	Enables user sign-in through Microsoft account identity services. If you stop this service, users can't sign in to the computer with their Microsoft account.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Microsoft Accounts are N/A on Windows Server

Microsoft App-V Client

Expand table

Name	Description
Service name	AppVClient
Description	Manages App-V users and virtual applications
Installation	Only with Desktop Experience
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Microsoft iSCSI Initiator Service

Expand table

Name	Description
Service name	MSiSCSI
Description	Manages Internet SCSI (iSCSI) sessions from this computer to remote iSCSI target devices. If you stop this service, this computer can't sign in or access iSCSI targets. If you disable this service, any services that explicitly depend on it are unable to start.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Our diagnostic data indicates the system uses this service on both client and server, so there's no benefit to disabling it.

Microsoft Passport

Expand table

Name	Description
Service name	NgcSvc
Description	Provides process isolation for cryptographic keys the system uses to authenticate to a user's associated identity providers. If you disable this service, all uses and management of these keys become available, which includes the ability to sign in to the machine and single-sign on for apps and websites. This service starts and stops automatically. We recommend you don't reconfigure this service.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Needed to sign in with PIN and Windows Hello, which Windows Server doesn't support.

Microsoft Passport Container

Expand table

Name	Description
Service name	NgcCtnrSvc
Description	Manages local user identity keys used to authenticate user to identity providers and TPM virtual smart cards. If you disable this service, the system can't access local user identity keys and TPM virtual smart cards. We recommend you don't reconfigure this service.

Name	Description
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Microsoft Software Shadow Copy Provider

Expand table

Name	Description
Service name	swprv
Description	Manages software-based volume shadow copies taken by the Volume Shadow Copy service. If you stop this service, the system becomes unable to manage software-based volume shadow copies. If you disable this service, any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Microsoft Storage Spaces SMP

Expand table

Name	Description
Service name	smphost
Description	Host service for the Microsoft Storage Spaces management provider. If you stop or disable this service, the system can't manage Storage Spaces.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable

Name	Description
Comments	Storage management APIs fail without this service. For example, the `Get-WmiObject -class MSFT_Disk -Namespace Root\Microsoft\Windows\Storage` command stops working.

Net.Tcp Port Sharing Service

Expand table

Name	Description
Service name	NetTcpPortSharing
Description	Provides ability to share TCP ports over the net.tcp protocol.
Installation	Always installed
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Netlogon

Expand table

Name	Description
Service name	Netlogon
Description	Maintains a secure channel between this computer and the domain controller for authenticating users and services. If you stop this service, the computer may not authenticate users and services and the domain controller can't register DNS records. If you disable this service, any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Network Connection Broker

Expand table

Name	Description
Service name	NcbService
Description	Brokers connections that allow Microsoft Store Apps to receive notifications from the internet.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Network Connections

Expand table

Name	Description
Service name	Netman
Description	Manages objects in the Network and Dial-Up Connections folder, in which you can view both local area network and remote connections.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Network Connectivity Assistant

Expand table

Name	Description
Service name	NcaSvc
Description	Provides DirectAccess status notification for UI components
Installation	Always installed

Name	Description
Startup type	Manual
Recommendation	No guidance
Comments	None

Network List Service

Expand table

Name	Description
Service name	netprofm
Description	Identifies the networks the computer has connected to, collects and stores properties for these networks, and notifies applications when these properties change.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Network Location Awareness

Expand table

Name	Description
Service name	NlaSvc
Description	Collects and stores configuration information for the network and notifies programs when this information is modified. If you stop this service, configuration information might be unavailable. If you disable this service, any services that explicitly depend on it are unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Network Setup Service

Expand table

Name	Description
Service name	NetSetupSvc
Description	The Network Setup Service manages the installation of network drivers and permits the configuration of low-level network settings. If you stop this service, the system may cancel any in-progress driver installations.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Network Store Interface Service

Expand table

Name	Description
Service name	nsi
Description	This service delivers network notifications, such as interface addition and deletions, to user mode clients. Stopping this service causes the system to lose network connectivity. If you disable this service, any other services that explicitly depend on this service become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Offline Files

Expand table

Name	Description
Service name	CscService
Description	The Offline Files service performs maintenance activities on the Offline Files cache, responds to user logon and logoff events, implements the internals of the public API, and dispatches interesting events to those interested in Offline Files activities and changes in cache state.
Installation	Only with Desktop Experience
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Optimize drives

Expand table

Name	Description
Service name	defragsvc
Description	Helps the computer run more efficiently by optimizing files on storage drives.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Performance Counter DLL Host

Expand table

Name	Description
Service name	PerfHost
Description	Enables remote users and 64-bit processes to query performance counters provided by 32-bit DLLs. If you stop this service, only local users and 32-bit processes can query performance counters provided by 32-bit DLLs.
Installation	Always installed

Name	Description
Startup type	Manual
Recommendation	No guidance
Comments	None

Performance Logs and Alerts

Expand table

Name	Description
Service name	pla
Description	Performance Logs and Alerts Collects performance data from local or remote computers based on preconfigured schedule parameters, then writes the data to a log or triggers an alert. If you stop this service, the system stops collecting performance information. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Phone Service

Expand table

Name	Description
Service name	PhoneSvc
Description	Manages the telephony state on the device.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Used by modern Voice over Internet Protocol (VoIP) apps.

Plug and Play

Expand table

Name	Description
Service name	PlugPlay
Description	Enables a computer to recognize and adapt to hardware changes with little or no user input. Stopping or disabling this service will result in system instability.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Portable Device Enumerator Service

Expand table

Name	Description
Service name	WPDBusEnum
Description	Enforces group policy for removable mass-storage devices. Enables applications such as Windows Media Player and Image Import Wizard to transfer and synchronize content using removable mass-storage devices.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Power

Expand table

Name	Description
Service name	Power

Name	Description
Description	Manages power policy and power policy notification delivery.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Print Spooler

Expand table

Name	Description
Service name	Spooler
Description	This service spools print jobs and handles interaction with the printer. If you turn off this service, you can't print or see your printers.
Installation	Always installed
Startup type	Automatic
Recommendation	OK to disable if not a print server or a DC
Comments	On a domain controller, installing the DC role adds a thread to the spooler service that's responsible for performing *print pruning*, which removes the stale print queue objects from the Active Directory. If the spooler service isn't running on at least one DC in each site, then the AD can't remove old queues that no longer exist. Learn more at "Disabling Unnecessary Services? A Word to the Wise".

Printer Extensions and Notifications

Expand table

Name	Description
Service name	PrintNotify
Description	This service opens custom printer dialog boxes and handles notifications from a remote print server or a printer. If you turn off this service, you can't see printer extensions or notifications.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	OK to disable if not a print server
Comments	None

Problem Reports and Solutions Control Panel Support

Expand table

Name	Description
Service name	wercplsupport
Description	This service provides support for viewing, sending and deletion of system-level problem reports for the Problem Reports and Solutions control panel.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Program Compatibility Assistant Service

Expand table

Name	Description
Service name	PcaSvc
Description	This service provides support for the Program Compatibility Assistant (PCA). PCA monitors programs installed and run by the user and detects known compatibility problems. If you stop this service, PCA stops working.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	OK to disable
Comments	None

Quality Windows Audio Video Experience

Expand table

Name	Description
Service name	QWAVE
Description	Quality Windows Audio Video Experience (qWave) is a networking platform for Audio Video (AV) streaming applications on IP home networks. qWave enhances AV streaming performance and reliability by ensuring network quality-of-service (QoS) for AV applications. It provides mechanisms for admission control, runtime monitoring and enforcement, application feedback, and traffic prioritization.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	Client-side QoS service

Radio Management Service

Expand table

Name	Description
Service name	RmSvc
Description	Radio Management and Airplane Mode Service
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Remote Access Auto Connection Manager

Expand table

Name	Description
Service name	RasAuto
Description	Creates a connection to a remote network whenever a program references a remote DNS or NetBIOS name or address.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Remote Access Connection Manager

Expand table

Name	Description
Service name	RasMan
Description	Manages dial-up and VPN connections from this computer to the Internet or other remote networks. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Remote Desktop Configuration

Expand table

Name	Description
Service name	SessionEnv
Description	Remote Desktop Configuration service (RDCS) is responsible for all Remote Desktop Services and Remote Desktop related configuration and session maintenance activities that require SYSTEM context. These include per-session temporary folders, RD themes, and RD certificates.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	None

Remote Desktop Services

Expand table

Name	Description
Service name	TermService
Description	Allows users to connect interactively to a remote computer. Remote Desktop and Remote Desktop Session Host Server depend on this service. To prevent remote use of this computer, clear the checkboxes on the Remote tab of the System properties control panel item.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	None

Remote Desktop Services UserMode Port Redirector

Expand table

Name	Description
Service name	UmRdpService
Description	Allows the redirection of Printers, Drives, or Ports for RDP connections
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Supports redirections on the server side of the connection.

Remote Procedure Call (RPC)

Expand table

Name	Description
Service name	RpcSs
Description	The RPCSS service is the Service Control Manager for COM and DCOM servers. It performs object activations requests, object exporter resolutions and distributed garbage collection for COM and DCOM servers. If you stop or disable this service, programs using COM or DCOM stop functioning properly. We highly recommend that you leave the RPCSS service running.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Remote Procedure Call (RPC) Locator

Expand table

Name	Description
Service name	RpcLocator
Description	In Windows 2003 and earlier, the Remote Procedure Call (RPC) Locator service manages the RPC name service database. In Windows Vista or later, this service doesn't provide any functionality and is only there to keep the OS compatible with certain applications.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Remote Registry

Expand table

Name	Description
Service name	RemoteRegistry
Description	Enables remote users to modify registry settings on this computer. If you stop this service, only users on this computer can modify the registry. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	Do not disable
Comments	None

Resultant Set of Policy Provider

Expand table

Name	Description
Service name	RSoPProv
Description	Provides a network service that processes requests to simulate application of Group Policy settings for a target user or computer in various situations and computes the Resultant Set of Policy settings.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Routing and Remote Access

Expand table

Name	Description
Service name	RemoteAccess
Description	Offers routing services to businesses in local area and wide area network environments.

Name	Description
Installation	Always installed
Startup type	Disabled
Recommendation	Already disabled
Comments	Already disabled

RPC Endpoint Mapper

Expand table

Name	Description
Service name	RpcEptMapper
Description	Resolves Remote Procedure Call (RPC) interfaces identifiers to transport endpoints. If you stop or disable this service, programs using RPC services stop functioning properly.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Secondary Logon

Expand table

Name	Description
Service name	seclogon
Description	Enables starting processes under alternate credentials. If you stop this service, this method of signing in becomes unavailable. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance

Name	Description
Comments	None

Secure Socket Tunneling Protocol Service

Expand table

Name	Description
Service name	SstpSvc
Description	Provides support for the Secure Socket Tunneling Protocol (SSTP) to connect to remote computers using VPN. If you disable this service, users can't use SSTP to access remote servers.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Disabling breaks RRAS

Security Accounts Manager

Expand table

Name	Description
Service name	SamSs
Description	The startup of this service signals other services that the Security Accounts Manager (SAM) is ready to accept requests. Disabling this service prevents the system from notifying other services when the SAM is ready, which can prevent those services from starting correctly. You shouldn't disable this service.
Installation	Always installed
Startup type	Automatic
Recommendation	Do not disable
Comments	None

Sensor Data Service

Expand table

Name	Description
Service name	SensorDataService
Description	Delivers data from a variety of sensors
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Sensor Monitoring Service

Expand table

Name	Description
Service name	SensrSvc
Description	Monitors various sensors to expose data and adapt to system and user state. If you stop or disable this service, the display brightness doesn't adapt to lighting conditions. Stopping this service can also affect other system functionality and features.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Sensor Service

Expand table

Name	Description
Service name	SensorService
Description	Manages the functionality of different sensors, including Simple Device Orientation (SDO) and History. Loads the SDO sensor that reports device

Name	Description
	orientation changes. If you stop or disable this service, the SDO sensor isn't loaded, which prevents history collection and auto-rotation from working.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Server

Expand table

Name	Description
Service name	LanmanServer
Description	Supports file, print, and named-pipe sharing over the network for this computer. If you stop this service, these functions become unavailable. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	Do not disable
Comments	Needed for remote management, IPC$, SMB file sharing

Shell Hardware Detection

Expand table

Name	Description
Service name	ShellHWDetection
Description	Provides notifications for AutoPlay hardware events.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	OK to disable

Name	Description
Comments	None

Smart Card

Expand table

Name	Description
Service name	SCardSvr
Description	Manages access to smart cards read by this computer. If you stop this service, this computer becomes unable to read smart cards. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Smart Card Device Enumeration Service

Expand table

Name	Description
Service name	ScDeviceEnum
Description	Creates software device nodes for all smart card readers accessible to a given session. If you disable this service, WinRT APIs can't enumerate smart card readers.
Installation	Always installed
Startup type	Manual
Recommendation	OK to disable
Comments	Needed almost exclusively for WinRT apps

Smart Card Removal Policy

Expand table

Name	Description
Service name	SCPolicySvc
Description	Allows the system to be configured to lock the user desktop upon smart card removal.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

SNMP Trap

Expand table

Name	Description
Service name	SNMPTRAP
Description	Receives trap messages generated by local or remote Simple Network Management Protocol (SNMP) agents and forwards the messages to SNMP management programs running on this computer. If you stop this service, SNMP-based programs on this computer can't receive SNMP trap messages. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Software Protection

Expand table

Name	Description
Service name	sppsvc

Name	Description
Description	Enables the download, installation and enforcement of digital licenses for Windows and Windows applications. If you disable the service, the operating system and licensed applications may run in a notification mode. We highly recommend that you don't disable the Software Protection service.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Special Administration Console Helper

Expand table

Name	Description
Service name	sacsvr
Description	Allows administrators to remotely access a command prompt using Emergency Management Services.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Spot Verifier

Expand table

Name	Description
Service name	svsvc
Description	Verifies potential file system corruptions.
Installation	Always installed
Startup type	Manual

Name	Description
Recommendation	No guidance
Comments	None

SSDP Discovery

Expand table

Name	Description
Service name	SSDPSRV
Description	Discovers networked devices and services that use the Simple Service Discovery Protocol (SSDP), such as Universal Plug and Play (UPnP) devices. Also announces SSDP devices and services running on the local computer. If you stop this service, the computer can't discover SSDP-based devices. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

State Repository Service

Expand table

Name	Description
Service name	StateRepository
Description	Provides required infrastructure support for the application model.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Still Image Acquisition Events

Expand table

Name	Description
Service name	WiaRpc
Description	Launches applications associated with still image acquisition events.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Storage Service

Expand table

Name	Description
Service name	StorSvc
Description	Provides enabling services for storage settings and external storage expansion.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Storage Tiers Management

Expand table

Name	Description
Service name	TieringEngineService
Description	Optimizes data placement in storage tiers on all tiered storage spaces in the system.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Superfetch

Expand table

Name	Description
Service name	SysMain
Description	Maintains and improves system performance over time.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Sync Host

Expand table

Name	Description
Service name	OneSyncSvc
Description	This service synchronizes mail, contacts, calendar, and various other user data. Mail and other applications dependent on this functionality don't work properly when this service isn't running.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	OK to disable
Comments	User service template

System Event Notification Service

Expand table

Name	Description
Service name	SENS
Description	Monitors system events and notifies subscribers to COM+ Event System of these events.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

System Events Broker

Expand table

Name	Description
Service name	SystemEventsBroker
Description	Coordinates execution of background work for WinRT applications. If you stop or disable this service, then background work might not be triggered.
Installation	Always installed
Startup type	Automatic
Recommendation	Do not disable
Comments	Despite this service's description implying it's only for WinRT apps, it's also needed for task scheduler, broker infrastructure service, and other internal components.

Task Scheduler

Expand table

Name	Description
Service name	Schedule

Name	Description
Description	Lets users configure and schedule automated tasks on this computer. The service also hosts multiple Windows system-critical tasks. If you stop or disable this service, these tasks don't run at their scheduled times, and any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

TCP/IP NetBIOS Helper

Expand table

Name	Description
Service name	lmhosts
Description	Provides support for the NetBIOS over TCP/IP (NetBT) services and NetBIOS name resolution for clients on the network, which lets users share files, print, and sign in to the network. If you stop this service, these functions might become unavailable. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Telephony

Expand table

Name	Description
Service name	TapiSrv
Description	Provides Telephony API (TAPI) support for programs that control telephony devices on the local computer and, through the Local Area Network (LAN), on servers that are also running the service.

Name	Description
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	Do not disable
Comments	Disabling breaks Routing and Remote Access Service (RRAS)

Themes

Expand table

Name	Description
Service name	Themes
Description	Provides user experience theme management.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	Do not disable
Comments	Can't set accessibility themes when this service is disabled

Tile Data model server

Expand table

Name	Description
Service name	tiledatamodelsvc
Description	Tile Server for tile updates.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	Do not disable
Comments	Start menu breaks If you disable this service

Time Broker

Expand table

Name	Description
Service name	TimeBrokerSvc
Description	Coordinates execution of background work for WinRT applications. If you stop or disable this service, then background work might not be triggered.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Despite this service's name implying it's only for WinRT apps, it's needed for task scheduler, broker infrastructure service, and other internal components.

Touch Keyboard and Handwriting Panel Service

Expand table

Name	Description
Service name	TabletInputService
Description	Enables Touch Keyboard and Handwriting Panel pen and ink functionality.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Update Orchestrator Service for Windows Update

Expand table

Name	Description
Service name	UsoSvc
Description	Manages Windows Updates. If stopped, your devices will not be able to download and install latest updates.
Installation	Always installed

Name	Description
Startup type	Manual
Recommendation	Do not disable
Comments	Service description was missing in v1607; Windows Update, including Windows Server Update Services (WSUS), depends on this service.

UPnP Device Host

Expand table

Name	Description
Service name	upnphost
Description	Allows this computer to host UPnP devices. If you stop this service, any hosted UPnP devices stop functioning and the computer can't add any more hosted devices.If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

User Access Logging Service

Expand table

Name	Description
Service name	UALSVC
Description	This service logs unique client access requests in the form of IP addresses and user names of installed products and roles on the local server. Any administrators who need to quantify client demand of server software for offline Client Access License (CAL) management can query these logs using PowerShell. If the service is disabled, the system stops logging client requests, preventing Powershell queries from retrieving them. Stopping the service doeesn't affect or delete historical data queries. The local system administrator must consult their Windows Server license terms to determine how many CALs the server software needs to be appropriately licensed.

Name	Description
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

User Data Access

Expand table

Name	Description
Service name	UserDataSvc
Description	Provides apps access to structured user data, including contact info, calendars, messages, and other content. If you stop or disable this service, apps that use this data might not work correctly.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	User service template

User Data Storage

Expand table

Name	Description
Service name	UnistoreSvc
Description	Handles storage of structured user data, including contact info, calendars, messages, and other content. If you stop or disable this service, apps that use this data might not work correctly.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable

Name	Description
Comments	User service template

User Experience Virtualization Service

Expand table

Name	Description
Service name	UevAgentService
Description	Provides support for application and OS settings roaming.
Installation	Only with Desktop Experience
Startup type	Disabled
Recommendation	Already disabled
Comments	None

User Manager

Expand table

Name	Description
Service name	UserManager
Description	User Manager provides the runtime components required for multi-user interaction. If you stop this service, some applications may not operate correctly.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

User Profile Service

Expand table

Name	Description
Service name	ProfSvc
Description	This service is responsible for loading and unloading user profiles. If you stop or disable this service, users become unable to successfully sign in or sign out, apps might have problems getting to users' data, and components registered to receive profile event notifications don't receive them.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Virtual Disk

Expand table

Name	Description
Service name	vds
Description	Provides management services for disks, volumes, file systems, and storage arrays.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Volume Shadow Copy

Expand table

Name	Description
Service name	VSS
Description	Manages and implements Volume Shadow Copies used for backup and other purposes. If you stop this service, shadow copies become unavailable for backup, causing the backup to fail. If you disable this service, any services that explicitly depend on it become unable to start.

Name	Description
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

WalletService

Expand table

Name	Description
Service name	WalletService
Description	Hosts objects used by wallet clients.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Windows Audio

Expand table

Name	Description
Service name	Audiosrv
Description	Manages audio for Windows-based programs. If you stop this service, audio devices and effects stop functioning properly. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Windows Audio Endpoint Builder

Expand table

Name	Description
Service name	AudioEndpointBuilder
Description	Manages audio devices for the Windows Audio service. If you stop this service, audio devices and effects stop functioning properly. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Windows Biometric Service

Expand table

Name	Description
Service name	WbioSrvc
Description	The Windows biometric service gives client applications the ability to capture, compare, manipulate, and store biometric data without gaining direct access to any biometric hardware or samples. The service is hosted in a privileged SVCHOST process.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows Camera Frame Server

Expand table

Name	Description
Service name	FrameServer
Description	Enables multiple clients to access video frames from camera devices.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Windows Connection Manager

Expand table

Name	Description
Service name	Wcmsvc
Description	Makes automatic connect or disconnect decisions based on the network connectivity options currently available to the PC and enables management of network connectivity based on Group Policy settings.
Installation	Only with Desktop Experience
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Defender Network Inspection Service

Expand table

Name	Description
Service name	WdNisSvc
Description	Helps guard against intrusion attempts targeting known and newly discovered vulnerabilities in network protocols.
Installation	Always installed
Startup type	Manual

Name	Description
Recommendation	No guidance
Comments	None

Windows Defender Service

Expand table

Name	Description
Service name	WinDefend
Description	Helps protect users from malware and other potentially unwanted software.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Driver Foundation - User-mode Driver Framework

Expand table

Name	Description
Service name	wudfsvc
Description	Creates and manages user-mode driver processes. You can't stop this service.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows Encryption Provider Host Service

Expand table

Name	Description
Service name	WEPHOSTSVC
Description	Windows Encryption Provider Host Service brokers encryption related functionalities from third-party Encryption Providers to processes that need to evaluate and apply EAS policies. Stopping this process compromises EAS compliancy checks established by the connected Mail Accounts.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows Error Reporting Service

Expand table

Name	Description
Service name	WerSvc
Description	Allows the system to report errors when programs stop working or responding and allows delivery of existing solutions. Also allows log generation for diagnostic and repair services. If you stop this service, error reporting might not work correctly and results of diagnostic services and repairs might not be displayed.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Collects and sends data when a program crashes or stops responding, which both Microsoft and third-party Independent Software Vendors (ISVs) or Independent Network Vendors (INVs) can use to diagnose crash-inducing bugs and security bugs. Also needed for Corporate Error Reporting.

Windows Event Collector

Expand table

Name	Description
Service name	Wecsvc
Description	This service manages persistent subscriptions to events from remote sources that support the Windows Server Management protocol. Event sources include Windows Vista event logs and hardware and IPMI-enabled event sources. The service stores forwarded events in a local Event Log. If you stop or disable this service, the system can't create event subscriptions or accept forwarded events.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Collects ETW events, including security events, for manageability and diagnostics. Lots of features and third-party tools rely on it, including security audit tools.

Windows Event Log

Expand table

Name	Description
Service name	EventLog
Description	This service manages events and event logs. It supports logging events, querying events, subscribing to events, archiving event logs, and managing event metadata. Windows Event Log can display events in both XML and plaintext format. Stopping this service may compromise system security and reliability.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Firewall

Expand table

Name	Description
Service name	MpsSvc
Description	Windows Firewall protects your computer by preventing unauthorized users from gaining access to your computer through the Internet or a network.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Font Cache Service

Expand table

Name	Description
Service name	FontCache
Description	Optimizes performance of applications by caching commonly used font data. Applications automatically start this service if it's not already running. You can disable this service, but disabling it degrades application performance.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Image Acquisition (WIA)

Expand table

Name	Description
Service name	stisvc
Description	Provides image acquisition services for scanners and cameras.
Installation	Only with Desktop Experience
Startup type	Manual

Name	Description
Recommendation	OK to disable
Comments	None

Windows Insider Service

Expand table

Name	Description
Service name	wisvc
Description	wisvc
Installation	Always installed
Startup type	Manual
Recommendation	OK to disable
Comments	Server doesn't support flighting, so it's a no-op on Server. You can also disable the feature using Group Policy.

Windows Installer

Expand table

Name	Description
Service name	msiserver
Description	Adds, modifies, and removes applications provided as a Windows Installer (*.msi, *.msp) package. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows License Manager Service

Expand table

Name	Description
Service name	LicenseManager
Description	Provides infrastructure support for the Microsoft Store. This service is started on demand and if disabled then content acquired through the Microsoft Store will not function properly.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows Management Instrumentation

Expand table

Name	Description
Service name	Winmgmt
Description	Provides a common interface and object model to access management information about operating system, devices, applications and services. If you stop this service, most Windows-based software will not function properly. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Windows Mobile Hotspot Service

Expand table

Name	Description
Service name	icssvc

Name	Description
Description	Provides the ability to share a cellular data connection with another device.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	None

Windows Modules Installer

Expand table

Name	Description
Service name	TrustedInstaller
Description	Enables installation, modification, and removal of Windows updates and optional components. If you disable this service, install or uninstall of Windows updates might fail for this computer.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Windows Push Notifications System Service

Expand table

Name	Description
Service name	WpnService
Description	This service runs in session 0 and hosts the notification platform and connection provider which handles the connection between the device and WNS server.
Installation	Only with Desktop Experience
Startup type	Automatic

Name	Description
Recommendation	OK to disable
Comments	Needed for live tiles and other features

Windows Push Notifications User Service

Expand table

Name	Description
Service name	WpnUserService
Description	This service hosts Windows notification platform which provides support for local and push notifications. Supported notifications are tile, toast and raw.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	OK to disable
Comments	User service template

Windows Remote Management (WS-Management)

Expand table

Name	Description
Service name	WinRM
Description	Windows Remote Management (WinRM) service implements the WS-Management protocol for remote management. WS-Management is a standard web services protocol used for remote software and hardware management. The WinRM service listens on the network for WS-Management requests and processes them. The WinRM Service needs to be configured with a listener using winrm.cmd command line tool or through Group Policy in order for it to listen over the network. The WinRM service provides access to WMI data and enables event collection. Event collection and subscription to events require that the service is running. WinRM messages use HTTP and HTTPS as transports. The WinRM service does not depend on IIS but is preconfigured to share a port with IIS on the same machine. The WinRM service reserves the /wsman URL prefix. To prevent conflicts with IIS, administrators should ensure that any websites hosted on IIS do not use the /wsman URL prefix.

Name	Description
Installation	Always installed
Startup type	Automatic
Recommendation	Do not disable
Comments	Needed for remote management

Windows Search

Expand table

Name	Description
Service name	WSearch
Description	Provides content indexing, property caching, and search results for files, e-mail, and other content.
Installation	Only with Desktop Experience
Startup type	Disabled
Recommendation	Already disabled
Comments	None

Windows Time

Expand table

Name	Description
Service name	W32Time
Description	Maintains date and time synchronization on all clients and servers in the network. If you stop this service, date and time synchronization will be unavailable. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance

Name	Description
Comments	None

Windows Update

Expand table

Name	Description
Service name	wuauserv
Description	Enables the detection, download, and installation of updates for Windows and other programs. If you disable this service, users of this computer will not be able to use Windows Update or its automatic updating feature, and programs will not be able to use the Windows Update Agent (WUA) API.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

WinHTTP Web Proxy Auto-Discovery Service

Expand table

Name	Description
Service name	WinHttpAutoProxySvc
Description	WinHTTP implements the client HTTP stack and provides developers with a Win32 API and COM Automation component for sending HTTP requests and receiving responses. In addition, WinHTTP provides support for auto-discovering a proxy configuration via its implementation of the Web Proxy Auto-Discovery (WPAD) protocol.
Installation	Always installed
Startup type	Manual
Recommendation	Do not disable
Comments	Anything that uses the network stack can have a functional dependency on this service. Many organizations rely on this to configure their internal

Name	Description
	networks' HTTP proxy routing. Without it, internally-originating HTTP connections to the Internet will all fail.

Wired AutoConfig

Expand table

Name	Description
Service name	dot3svc
Description	The Wired AutoConfig (DOT3SVC) service is responsible for performing IEEE 802.1X authentication on Ethernet interfaces. If your current wired network deployment enforces 802.1X authentication, the DOT3SVC service should be configured to run for establishing Layer 2 connectivity and/or providing access to network resources. Wired networks that do not enforce 802.1X authentication are unaffected by the DOT3SVC service.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	No guidance
Comments	None

WMI Performance Adapter

Expand table

Name	Description
Service name	wmiApSrv
Description	Provides performance library information from Windows Management Instrumentation (WMI) providers to clients on the network. This service only runs when Performance Data Helper is activated.
Installation	Always installed
Startup type	Manual
Recommendation	No guidance
Comments	None

Workstation

Expand table

Name	Description
Service name	LanmanWorkstation
Description	Creates and maintains client network connections to remote servers using the SMB protocol. If you stop this service, these connections will be unavailable. If you disable this service, any services that explicitly depend on it become unable to start.
Installation	Always installed
Startup type	Automatic
Recommendation	No guidance
Comments	None

Xbox Live Auth Manager

Expand table

Name	Description
Service name	XblAuthManager
Description	Provides authentication and authorization services for interacting with Xbox Live. If you stop this service, some applications may not operate correctly.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	Should be disabled
Comments	None

Xbox Live Game Save

Expand table

Name	Description
Service name	XblGameSave

Name	Description
Description	This service syncs save data for Xbox Live save enabled games. If you stop this service, game save data will not upload to or download from Xbox Live.
Installation	Only with Desktop Experience
Startup type	Manual
Recommendation	Should be disabled
Comments	This service syncs save data for Xbox Live save enabled games. If you stop this service, game save data will not upload to or download from Xbox Live.

Feedback

Was this page helpful? Yes No

Per-user services in Windows

Article • 10/01/2024 • Applies to: ✅ Windows 11, ✅ Windows 10, ✅ Windows Server

When a user signs in to Windows, the OS creates *per-user services*. When the user signs out, these services are stopped and deleted. They run in the security context of the user account instead of a built-in security principal. This behavior provides better resource management than the previous approach of running these services associated with a preconfigured account or as tasks.

> **Note**
>
> Per-user services are only available in Windows Server if you install the Desktop Experience. For more information, see Server Core vs Server with Desktop Experience install options.

Windows creates these per-user services based on templates defined in the registry. If you need to manage or control behaviors of these services, you can adjust the template. For example, you can set a template service's startup type to **Disabled**. In this example, Windows creates the per-user service in a stopped and disabled state.

> **Important**
>
> Carefully test any changes to the template service's configuration before you broadly deploy them to a production environment.

Use the information in this article to understand per-user services, configure user service templates, and manage per-user services through group policy and security templates.

List of per-user services

The following table lists per-user services in the current version of Windows. Other versions of Windows 10/11 might not have the same services available.

Before you reconfigure any of these services, review this information to understand the implications. For example, if you disable the per-user service, there might be dependent apps that don't work correctly.

Expand table

Display name	Service name	Default start type	Dependencies	Description
Agent Activation Runtime	AarSvc	Manual		Runtime for activating conversational agent applications.
Bluetooth User Support Service	BluetoothUserService	Manual		Supports proper functionality of Bluetooth features relevant to each user session.
OneCore Capture Service	CaptureService	Manual		Enables optional screen capture functionality for applications that call screen capture APIs of the Windows.Graphics.Capture namespace.
Clipboard User Service	cbdhsvc	Automated (Delayed Start)		Windows uses this user service for clipboard scenarios. For example, clipboard history or sync across devices. For more information, see Clipboard in Windows.
Cloud Backup and Restore Service	CloudBackupRestoreSvc	Manual		Monitors the system for changes in application and setting states. When required, this service does cloud backup and restore operations.
Connected Devices Platform User Service	CDPUserSvc	Automatic	- Network Connection Broker - Remote Procedure Call (RPC) - TCP/IP Protocol Driver	This service allows the user to connect, manage, and control connected devices. These connected devices include mobile, Xbox, HoloLens, or smart/IoT devices. For one specific example, see Share things with nearby devices in Windows.
Consent UX User Service	ConsentUxUserSvc	Manual		Allows the system to request user consent to allow apps to access sensitive resources and information such as the device's location.
Contact Data	PimIndexMaintenanceSvc	Manual	UnistoreSvc	Indexes contact data for fast contact searching. If

Display name	Service name	Default start type	Dependencies	Description
				you stop or disable this service, contacts might be missing from your search results.
Credential Enrollment Manager	CredentialEnrollmentManagerUserSvc	Manual		This service supports the secure storage and retrieval of user credentials. For example, tokens for web sites, remote desktop connections, or other apps.
Device Association Broker	DeviceAssociationBrokerSvc	Manual	- DevicePicker - Shell Pairing UX	Supports in-app pairing and access checks for new device scenarios.
Device Picker	DevicePickerUserSvc	Manual		Windows uses this user service to manage Miracast, Digital Living Network Alliance (DLNA), and Discovery and Launch (DIAL) experiences.
Devices Flow	DevicesFlowUserSvc	Manual		Allows the Connect user interface and Settings app to connect and pair with WiFi displays and Bluetooth devices.
Game DVR and Broadcast User Service	BcastDVRUserService	Manual		Windows uses this user service for game recordings and live broadcasts.
Messaging Service	MessagingService	Manual		This service supports text messaging and related functionality.
Now Playing Session Manager	NPSMSvc	Manual		The *now playing session manager* (NPSM) service manages media sessions running on the device.
Plan 9 Redirector Service	P9RdrService	Manual		Enables trigger-starting plan9 file servers, which are supported by Windows Subsystem for Linux. For more

Display name	Service name	Default start type	Dependencies	Description
				information, see Plan 9 from Bell Labs.
Pen Service	PenService	Manual		When you press the tail button on a pen input device, this service responds to those actions. It can launch applications or take another action that you customize in Settings. For more information, see user documentation on How to use your Surface Pen or hardware developer documentation on Pen devices.
Print Workflow	PrintWorkflowUserSvc	Manual		Provides support for Print Workflow applications. If you turn off this service, some printing functions might not work successfully.
Sync Host	OneSyncSvc	Automated (Delayed Start)		This service synchronizes mail, contacts, calendar, and other user data. When this service is stopped, mail and other applications dependent on this functionality don't work properly.
UDK User Service	UdkUserSvc	Manual		Windows uses this service to coordinate between shell experiences.
User Data Access	UserDataSvc	Manual	UnistoreSvc	Provides apps access to structured user data, including contact info, calendars, messages, and other content. If you stop or disable this service, apps that use this data might not work correctly.
User Data Storage	UnistoreSvc	Manual		Handles storage of structured user data, including contact info, calendars, messages, and other content. If you stop

Display name	Service name	Default start type	Dependencies	Description
				or disable this service, apps that use this data might not work correctly.
Web Threat Defense User Service	webthreatdefusersvc	Automatic		This service helps protect your computer by warning the user when unauthorized entities attempt to gain access to their credentials.
Windows Push Notifications User Service	WpnUserService	Automatic		This service hosts the Windows push notification services (WNS) platform, which provides support for local and push notifications. Supported notifications are tile, toast, and raw.

View per-user services

You can't view the user service templates outside of the Windows Registry, but you can see the user-specific per-user services. Windows displays these services with the following format: `<service name>_LUID` where `<service name>` is the display name of the user service and `LUID` is a locally unique identifier for the user context.

For example, you might see the following per-user service names:

- `Contact Data_443f50`
- `Sync Host_443f50`
- `User Data Access_443f50`
- `User Data Storage_443f50`

> **Note**
>
> The display name and the service name for all per-user services include the same LUID suffix.

View per-user services in the Windows Services console

When you sign in to Windows, run `services.msc` to open the Services console. When you view the local machine, you can see these services for your user account.

View per-user services using Windows PowerShell

The following PowerShell script is an example of how to query for per-user services. It queries for service type values that include the `64` bit value.

PowerShell

```
# Define the bit value for per-user services in the ServiceType property of a
service object
$flag = 64

# Define an empty array to store the resulting services that match the criteria
$serviceList = @()

# Get all services on the computer and store them in the variable
$services = Get-Service

# Loop through each service in the array of services.
foreach ( $service in $services ) {
  # For each specific service, check if the service type property includes the 64
bit using the bitwise AND operator (-band).
  # If the result equals the flag value, then the service is a per-user service.
  if ( ( $service.ServiceType -band $flag ) -eq $flag ) {
    # When a per-user service is found, then add that service object to the results
array.
    $serviceList += $service
  }
}

# Display the results array, sorted by display name, in a table format with the
specified properties.
$serviceList | Sort-Object DisplayName | Format-Table DisplayName, Name, StartType,
ServiceType
```

View per-user services from the command line

Run `cmd.exe` to open a Windows command prompt. Use the `sc qc` command to query these services. The **Type** value indicates whether the service is a user-service template or user-service instance.

The following example queries for the template and user-specific instance of the **Game DVR and Broadcast User Service** (`BcastDVRUserService`) service:

Windows Command Prompt

```
sc qc BcastDVRUserService
sc qc BcastDVRUserService_18f113
```

```
PS C:\> sc.exe qc BcastDVRUserService
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: BcastDVRUserService
        TYPE               : 60  USER_SHARE_PROCESS TEMPLATE
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k BcastDVRUserService
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : GameDVR and Broadcast User Service
        DEPENDENCIES       : RpcSs
        SERVICE_START_NAME : LocalSystem
PS C:\> sc.exe qc BcastDVRUserService_660dc
[SC] QueryServiceConfig SUCCESS

SERVICE_NAME: BcastDVRUserService_660dc
        TYPE               : e0  USER_SHARE_PROCESS INSTANCE
        START_TYPE         : 3   DEMAND_START
        ERROR_CONTROL      : 1   NORMAL
        BINARY_PATH_NAME   : C:\Windows\system32\svchost.exe -k BcastDVRUserService
        LOAD_ORDER_GROUP   :
        TAG                : 0
        DISPLAY_NAME       : GameDVR and Broadcast User Service_660dc
        DEPENDENCIES       :
        SERVICE_START_NAME :
```

How to disable per-user services

The templates for user services aren't displayed in the **Services** console (services.msc). To disable a per-user service, you need to directly edit the registry, either with group policy or a scripted solution. The templates are located in the registry at `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services`.

> ⓘ **Note**
>
> When you disable a per-user service, Windows still creates it when the user signs in, but in a stopped and disabled state. When the user signs out, Windows removes the per-user service.

You can't manage all of the per-user service templates using normal group policy management methods. Because the per-user services aren't displayed in the **Services** management console, they're also not displayed in the group policy services policy editor.

Additionally, there are four user services that you can't manage with a security template:

- PimIndexMaintenanceSvc
- UnistoreSvc
- UserDataSvc
- WpnUserService

With these restrictions, you can use the following methods to manage per-user service templates:

- A combination of a security template and a script, or group policy preferences registry policy.
- Group policy preferences for all of the services.
- A script for all of the services.

Manage template services using a security template

You can manage the CDPUserSvc and OneSyncSvc per-user services with a security template.

For example:

ini

```
[Unicode]
Unicode=yes
[Version]
signature="$CHICAGO$"
Revision=1
[Service General Setting]
"CDPUserSVC".4,""
```

Use a script to manage per-user services

You can create a script to change the startup type for the per-user services. Then use group policy or another management solution such as Microsoft Configuration Manager to deploy the script to targeted devices.

Example 1: Use the `Set-Service` PowerShell cmdlet

The following sample script uses the Set-Service PowerShell cmdlet to configure the PimIndexMaintenanceSvc service start type to disabled:

PowerShell

```
Set-Service -Name PimIndexMaintenanceSvc -StartupType Disabled
```

Example 2: Use the `sc.exe config` command line

The following sample script uses sc.exe config to configure the PimIndexMaintenanceSvc service start type to disabled:

Windows Command Prompt

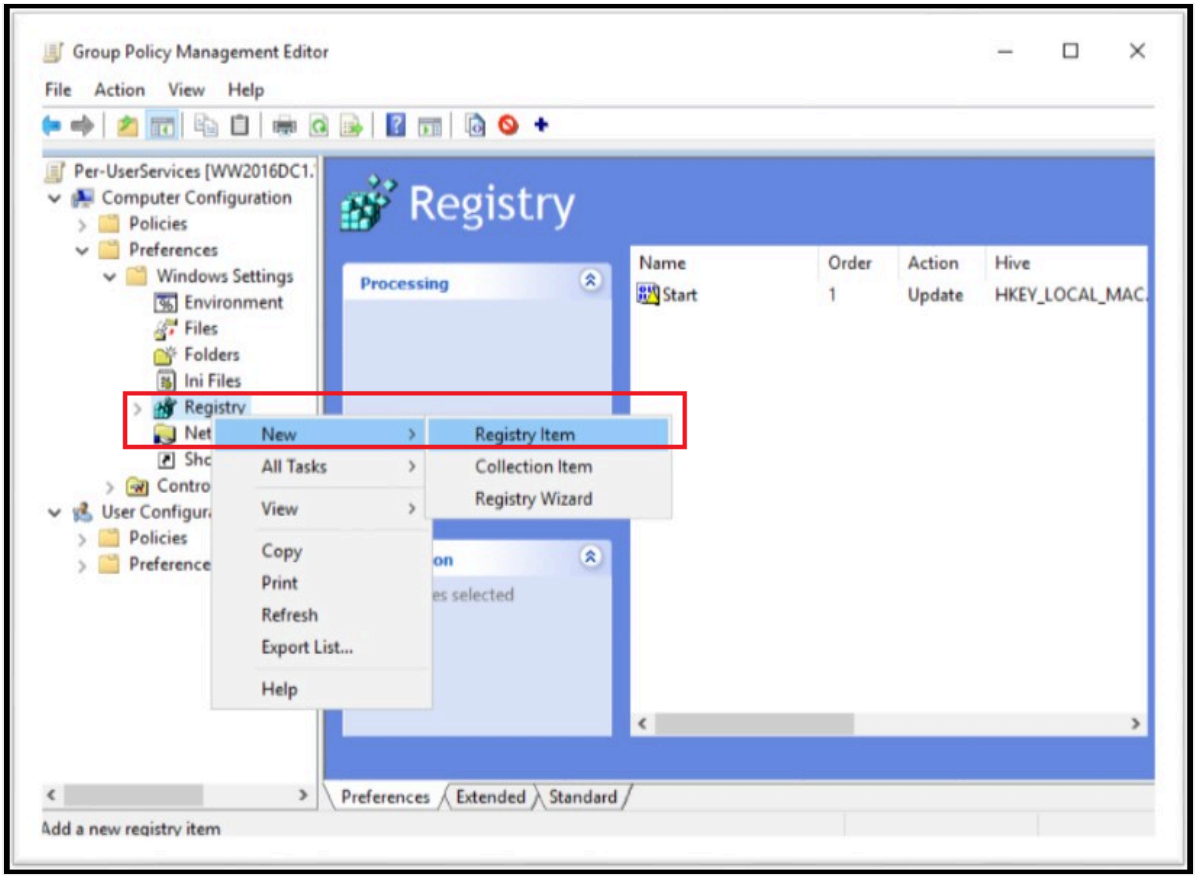
```
sc.exe configure PimIndexMaintenanceSvc start= disabled
```

> ⓘ **Note**
>
> The space after `=` is intentional.

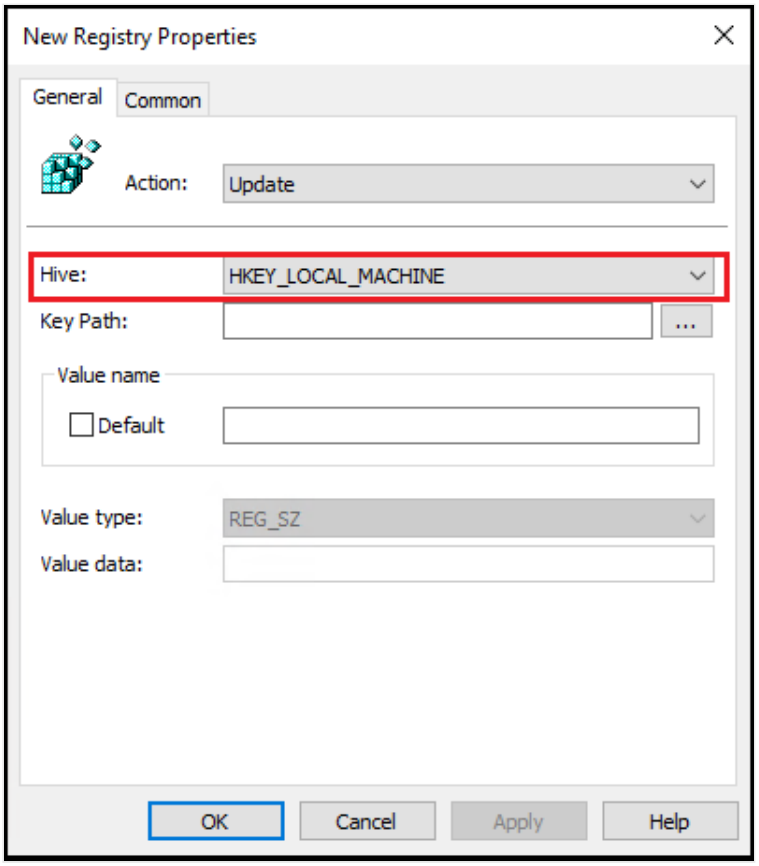
Manage template services using group policy preferences

If you can't disable a per-user service with the security template, use group policy preferences.

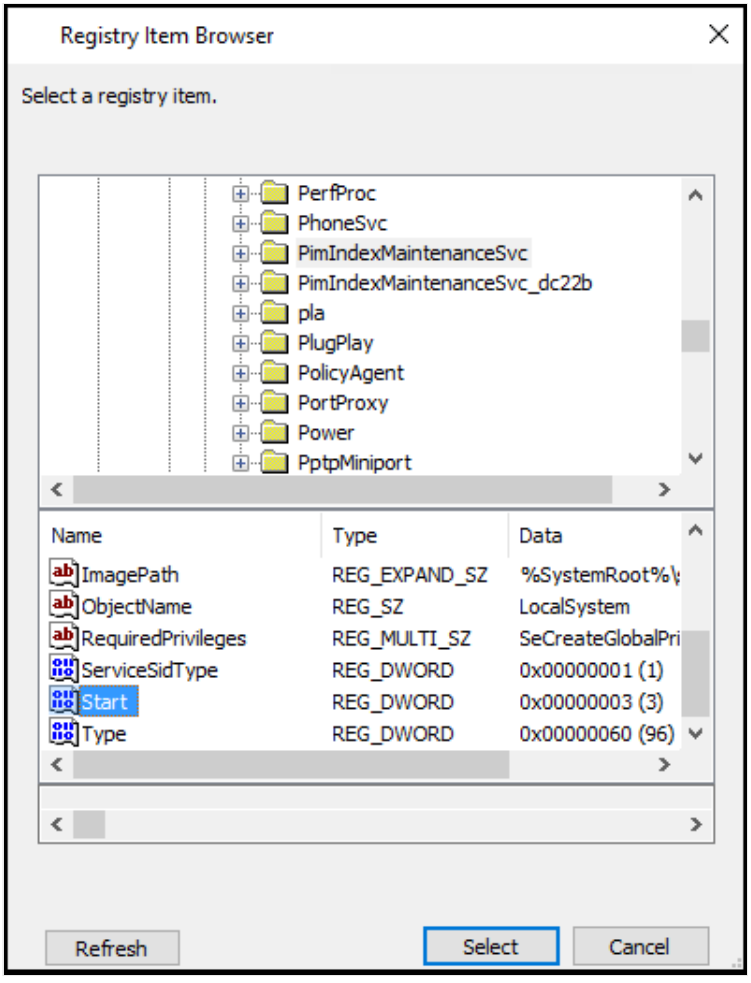
1. Open the **Group Policy Management Console** (gpmc.msc).
2. Create a new group policy object (GPO) or use an existing GPO.
3. **Edit** the GPO to launch the group policy object editor.
4. Depending on how you want to target the group policy, under **Computer configuration** or **User configuration** browse to **Preferences**, **Windows Settings**, and select **Registry**.
5. Go to the **Action** menu, select **New**, and select **Registry Item**.



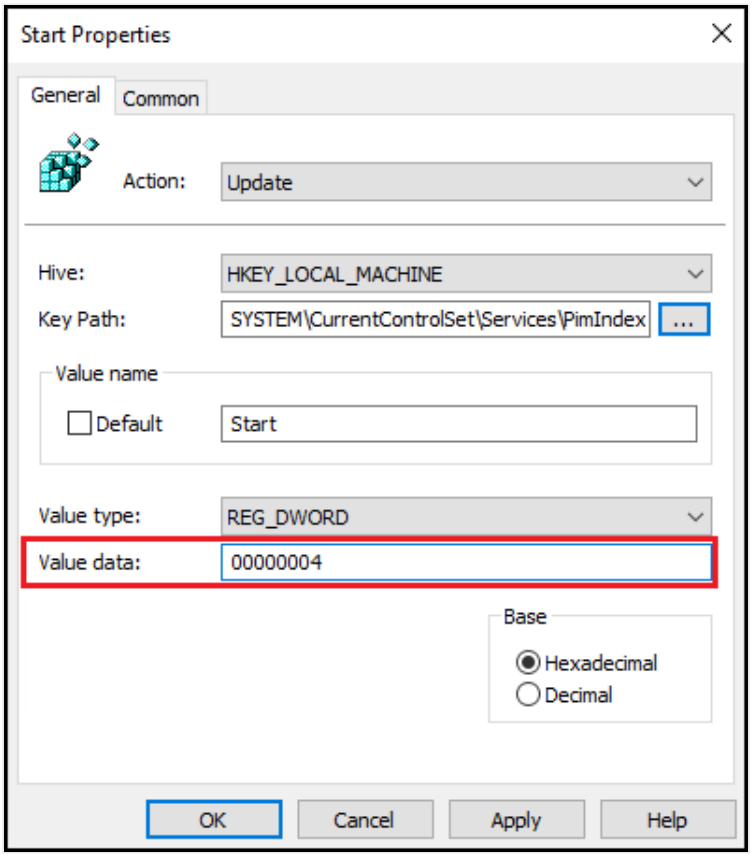
6. For the **Hive** select `HKEY_LOCAL_MACHINE`.



7. Select the ellipses (`...`) next to **Key Path**. Browse to `System\CurrentControlSet\Services` and then select the user service template. For example, `PimIndexMaintenanceSvc`. In the list of values, highlight **Start** and select **Select**.



8. In the **Start Properties** window, Change **Value data** from `00000003` to `00000004` and select **OK**. Note setting the Value data to **4** = **Disabled**.



> ⓘ **Note**
>
> The service start value `4` is **Disabled**.

9. To add the other services that can't be managed with group policy templates, edit the policy and repeat the previous steps.

Managing user service templates with the Windows Registry

If you can't use group policy preferences to manage the per-user services, you can edit the Windows Registry. To disable a user service template, change the startup type for each service to `4`, which is **Disabled**.

> ⊗ **Caution**
>
> Don't directly edit the registry unless there's no other alternative. The Registry Editor or Windows don't validate these manual modifications to the registry. Incorrect values can be stored, which can result in unrecoverable errors in the system. When possible, instead of editing the registry directly, use group policy or other supported Windows tools to accomplish these tasks. If you must edit the registry, use extreme caution.

Example 1: Use the `reg.exe` command line command to edit the registry

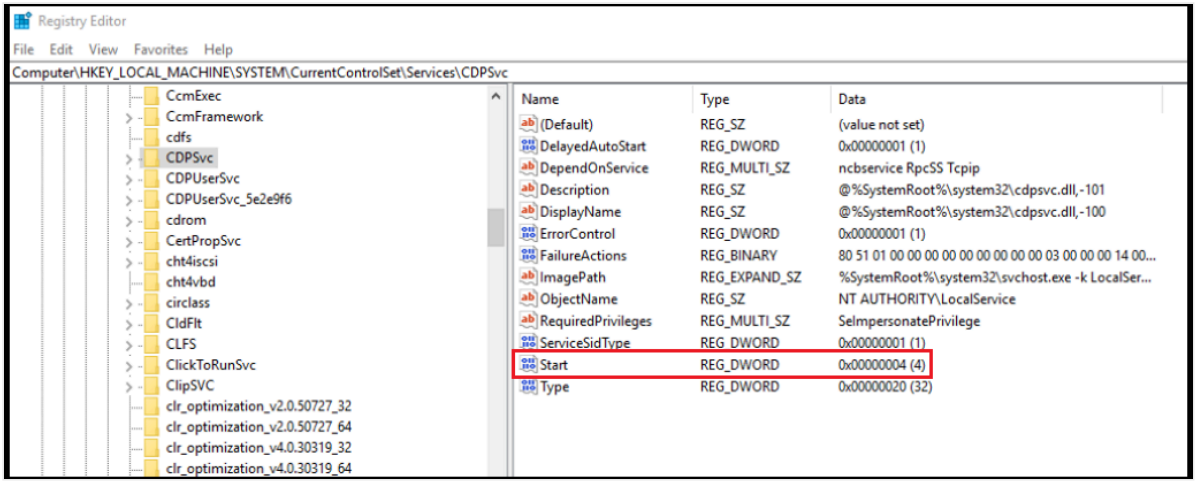
1. As an administrator, run `cmd.exe` to open a Windows command prompt.
2. The following example includes multiple commands that disable the specified Windows services by changing their **Start** value in the Windows Registry to `4`:

Windows Command Prompt

```
REG.EXE ADD HKLM\System\CurrentControlSet\Services\CDPUserSvc /v Start /t
REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\OneSyncSvc /v Start /t
REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\PimIndexMaintenanceSvc /v
Start /t REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\UnistoreSvc /v Start /t
REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\UserDataSvc /v Start /t
REG_DWORD /d 4 /f
REG.EXE ADD HKLM\System\CurrentControlSet\Services\WpnUserService /v Start /t
REG_DWORD /d 4 /f
```

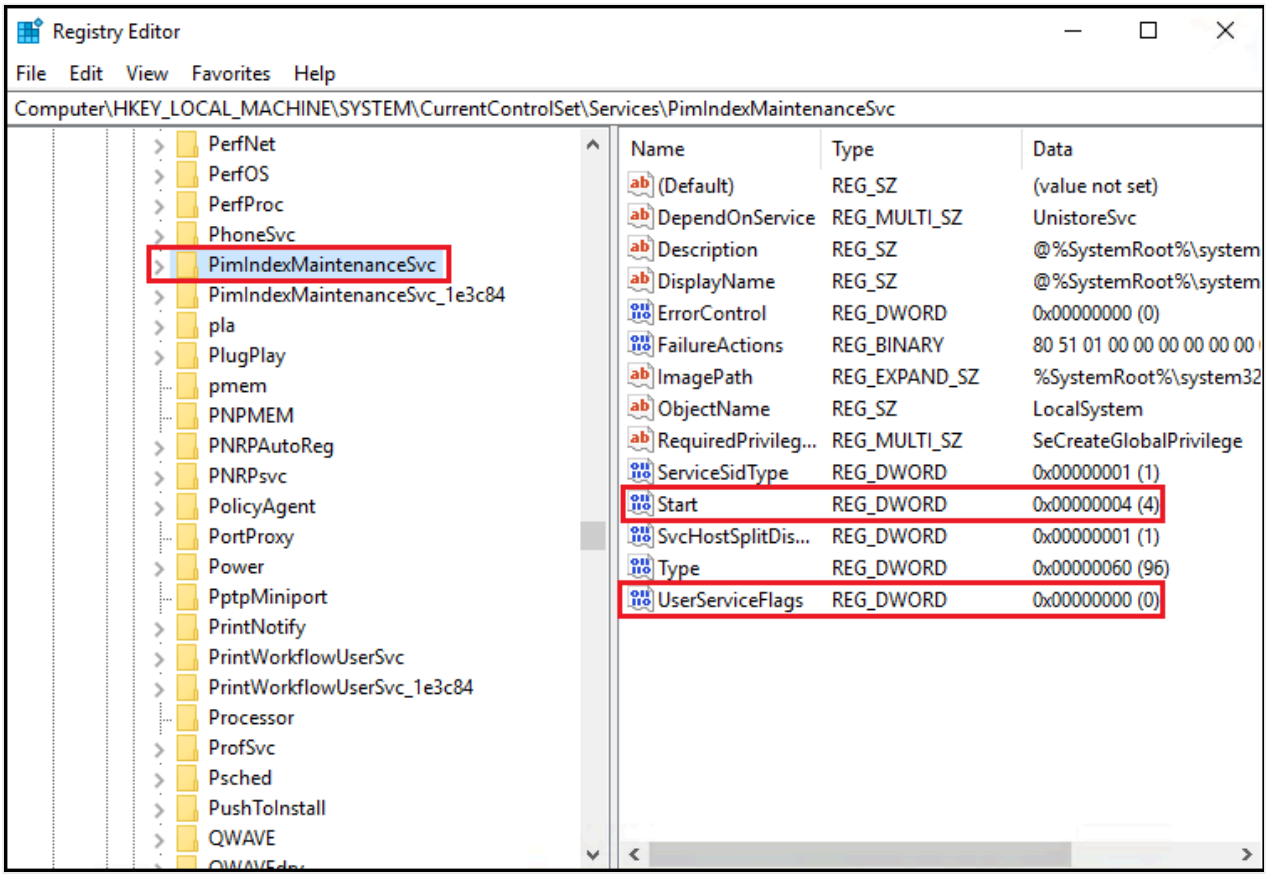
Example 2: Use the Registry Editor user interface to edit the registry

1. As an administrator, run `regedit.exe` to open the Registry Editor.
2. Browse to `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services` and then select the user service template. For example, `CDPSvc`.
3. In the list of values, open the **Start** value.
4. Change the **Value data** to `4`.



Example 3: Prevent the creation of per-user services

You can prevent Windows from creating a per-user service when a user signs on. In the same service template node of the registry, set `UserServiceFlags` to `0`.



Next steps

For more information about disabling system services for Windows Server, see Guidance on disabling system services on Windows Server with Desktop Experience.

Feedback

Was this page helpful? Yes No

Provide product feedback

Windows authentication overview

07/29/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This navigation topic for the IT professional lists documentation resources for Windows authentication and logon technologies that include product evaluation, getting started guides, procedures, design and deployment guides, technical references, and command references.

Feature description

Authentication is a process for verifying the identity of an object, service or person. When you authenticate an object, the goal is to verify that the object is genuine. When you authenticate a service or person, the goal is to verify that the credentials presented are authentic.

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 either a key only the user knows - 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. Active Directory Domain Services is the recommended and default technology for storing identity information (including the cryptographic keys that are the user's credentials). Active Directory is required for default NTLM and Kerberos implementations.

Authentication techniques range from a simple logon, which identifies users based on something that only the user knows - like a password, to more powerful security mechanisms that use something that the user has - like tokens, public key certificates, and biometrics. In a business environment, services or users might access multiple applications or resources on many types of servers within a single location or across multiple locations. For these reasons, authentication must support environments for other platforms and for other Windows operating systems.

The Windows operating system implements a default set of authentication protocols, including Kerberos, NTLM, Transport Layer Security/Secure Sockets Layer (TLS/SSL), and Digest, as part of an extensible architecture. In addition, some protocols are combined into authentication packages such as Negotiate and the Credential Security Support Provider. These protocols and packages enable authentication of users, computers, and services; the authentication process, in turn, enables authorized users and services to access resources in a secure manner.

For more information about Windows Authentication including

- Windows Authentication Concepts
- Windows Logon Scenarios
- Windows Authentication Architecture
- Security Support Provider Interface Architecture
- Credentials Processes in Windows Authentication
- Group Policy Settings Used in Windows Authentication

See the Windows Authentication Technical Overview.

Practical applications

Windows Authentication is used to verify that the information comes from a trusted source, whether from a person or computer object, such as another computer. Windows provides many different methods to achieve this goal as described below.

Expand table

To...	Feature	Description
Authenticate within an Active Directory domain	Kerberos	The Microsoft Windows Server operating systems implement the Kerberos version 5 authentication protocol and extensions for public key authentication. The Kerberos authentication client is implemented as a security support provider (SSP) and can be accessed through the Security Support Provider Interface (SSPI). Initial user authentication is integrated with the Winlogon single sign-on architecture. The Kerberos Key Distribution Center (KDC) is integrated with other Windows Server security services running on the domain controller. The KDC uses the domain's Active Directory directory service database as its security account database. Active Directory is required for default Kerberos implementations. For additional resources, see Kerberos Authentication Overview.
Secure authentication on the web	TLS/SSL as implemented in the Schannel Security Support Provider	The Transport Layer Security (TLS) protocol versions 1.0, 1.1, and 1.2, Secure Sockets Layer (SSL) protocol, versions 2.0 and 3.0, Datagram Transport Layer Security protocol version 1.0, and the Private Communications Transport (PCT) protocol, version 1.0, are based on public key cryptography. The Secure Channel (Schannel) provider authentication protocol suite provides these protocols. All Schannel protocols use a client and server model.

To...	Feature	Description
		For additional resources, see TLS - SSL (Schannel SSP) Overview.
Authenticate to a web service or application	Integrated Windows Authentication	For additional resources, see Integrated Windows Authentication.
Authenticate to legacy applications	NTLM	NTLM is a challenge-response style authentication protocol. In addition to authentication, the NTLM protocol optionally provides for session security--specifically message integrity and confidentiality through signing and sealing functions in NTLM. For additional resources, see NTLM Overview.
Leverage multifactor authentication	Smart card support Biometric support	Smart cards are a tamper-resistant and portable way to provide security solutions for tasks such as client authentication, logging on to domains, code signing, and securing e-mail. Biometrics relies on measuring an unchanging physical characteristic of a person to uniquely identify that person. Fingerprints are one of the most frequently used biometric characteristics, with millions of fingerprint biometric devices that are embedded in personal computers and peripherals. For additional resources, see Smart Card Technical Reference.
Provide local management, storage and reuse of credentials	Credentials management Local Security Authority Passwords	Credential management in Windows ensures that credentials are stored securely. Credentials are collected on the Secure Desktop (for local or domain access), through apps or through websites so that the correct credentials are presented for every time a resource is accessed.
Extend modern authentication protection to legacy systems	Extended Protection for Authentication	This feature enhances the protection and handling of credentials when authenticating network connections by using Integrated Windows Authentication (IWA).

Software requirements

Windows Authentication is designed to be compatible with previous versions of the Windows operating system. However, improvements with each release aren't necessarily applicable to previous versions. Refer to documentation about specific features for more information.

Server Manager information

Many authentication features can be configured using Group Policy, which can be installed using Server Manager. The Windows Biometric Framework feature is installed using Server

Manager. Other server roles which are dependent upon authentication methods, such as Web Server (IIS) and Active Directory Domain Services, can also be installed using Server Manager.

Related resources

Expand table

Authentication technologies	Resources
Windows authentication	Windows Authentication Technical Overview Includes topics addressing differences between versions, general authentication concepts, logon scenarios, architectures for supported versions, and applicable settings.
Kerberos	Kerberos Authentication Overview Kerberos Constrained Delegation Overview Kerberos forum
TLS/SSL and DTLS (Schannel security support provider)	TLS - SSL (Schannel SSP) Overview Schannel Security Support Provider Technical Reference
NTLM	NTLM Overview Contains links to current and past resources
PKU2U	Introducing PKU2U in Windows
Smart Card	Smart Card Technical Reference
Credentials	Credentials Protection and Management Contains links to current and past resources Passwords Overview Contains links to current and past resources

Windows Authentication Technical Overview

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional provides links to topics for the Windows Authentication Technical Overview. Windows authentication is the process to prove the authenticity of a user or service attempting to access Windows.

This collection of topics describes Windows authentication architecture and its components.

To digitally save or print pages from this library, click **Export** (in the upper-right corner of the page), and then follow the instructions.

- Differences in Windows Authentication Between Windows Operating Systems

  Describes the significant differences in the authentication architecture and processes.

- Windows Authentication Concepts

  Describes the concepts on which Windows authentication is based.

- Windows Logon Authentication Scenarios

  Summarizes the various logon scenarios.

- Windows Authentication Architecture

  Describes the significant differences in the authentication architecture and processes for Windows operating systems.

  - Security Support Provider Interface Architecture

    Describes the SSPI architecture.

  - Credentials Processes in Windows Authentication

    Describes the different credential management processes.

- Group Policies used in Windows Authentication

  Describes the use and impact of Group Policies in the authentication process.

What is not covered

This collection of topics does not cover procedures for designing, implementing, or monitoring your authentication technologies within a Windows environment.

- For design information on Windows authorization strategies, see Designing a Resource Authorization Strategy.
- For design information on Windows authentication strategies, see Designing an Authentication Strategy.
- For design information on Windows public key infrastructure implementation strategies, see Designing a Public Key Infrastructure.
- For configuring and monitoring security, including authentication, in your Windows environment, see:
  - Windows Vista Security Baseline
  - Windows Server 2003 Security Baseline and the Threats and Countermeasures Guide
  - Windows Server 2008 R2 Security Baseline
- For information about auditing logon and authentication events in Windows, see Auditing Security Events.

Feedback

Was this page helpful? Yes No

Windows Authentication Concepts

Article • 09/24/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This reference overview topic describes the concepts on which Windows authentication is based.

Authentication is a process for verifying the identity of an object or person. When you authenticate an object, the goal is to verify that the object is genuine. When you authenticate a person, the goal is to verify that the person is not 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 either a key only the user knows (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. Active Directory is the recommended and default technology for storing identity information, which include the cryptographic keys that are the user's credentials. Active Directory is required for default NTLM and Kerberos implementations.

Authentication techniques range from a simple logon to an operating system or a sign-in to a service or application, which identifies users based on something that only the user knows, such as a password, to more powerful security mechanisms that use something that the user has'such as tokens, public key certificates, pictures, or biological attributes. In a business environment, users might access multiple applications on many types of servers within a single location or across multiple locations. For these reasons, authentication must support environments for other platforms and for other Windows operating systems.

Authentication and authorization: A travel analogy

A travel analogy can help explain how authentication works. A few preparatory tasks are usually necessary to begin the journey. The traveler must prove their true identity to their host authorities. This proof can be in the form of proof of citizenship, birth place, a personal voucher, photographs, or whatever is required by the law of the host country.

The traveler's identity is validated by the issuance of a passport, which is analogous to a system account issued and administered by an organization--the security principal. The passport and the intended destination are based on a set of rules and regulations issued by the governmental authority.

The journey

When the traveler arrives at the international border, a border guard asks for credentials, and the traveler presents his or her passport. The process is two-fold:

- The guard authenticates the passport by verifying that it was issued by a security authority that the local government trusts (trusts, at least, to issue passports) and by verifying that the passport has not been modified.
- The guard authenticates the traveler by verifying that the face matches the face of the person pictured on the passport and that other required credentials are in good order.

If the passport proves to be valid and the traveler proves to be its owner, authentication is successful, and the traveler can be allowed access across the border.

Transitive trust between security authorities is the foundation of authentication; the type of authentication that takes place at an international border is based on trust. The local government does not know the traveler, but it trusts that the host government does. When the host government issued the passport, it did not know the traveler either. It trusted the agency that issued the birth certificate or other documentation. The agency that issued the birth certificate, in turn, trusted the physician who signed the certificate. The physician witnessed the traveler's birth and stamped the certificate with direct proof of the identity, in this case with the newborn's footprint. Trust that is transferred in this way, through trusted intermediaries, is transitive.

Transitive trust is the foundation for network security in Windows client/server architecture. A trust relationship flows throughout a set of domains, such as a domain tree, and forms a relationship between a domain and all domains that trust that domain. For example, if domain A has a transitive trust with domain B, and if domain B trusts domain C, then domain A trusts domain C.

There is a difference between authentication and authorization. With authentication, the system proves that you are who you say you are. With authorization, the system verifies that you have rights to do what you want to do. To take the border analogy to the next step, merely authenticating that the traveler is the proper owner of a valid passport does not necessarily authorize the traveler to enter a country. Residents of a particular country are allowed to enter another country by simply presenting a passport only in

situations where the country being entered grants unlimited permission for all citizens of that particular country to enter.

Similarly, you can grant all users from a certain domain permissions to access a resource. Any user who belongs to that domain has access to the resource, just as Canada lets U.S. citizens enter Canada. However, U.S. citizens attempting to enter Brazil or India find that they cannot enter those countries merely by presenting a passport because both of those countries require visiting U.S. citizens to have a valid visa. Thus, authentication does not guarantee access to resources or authorization to use resources.

Credentials

> **⊗ Caution**
>
> When a user performs a local logon, their credentials are verified locally against a cached copy before being authenticated with an identity provider over the network. If the cache verification is successful, the user gains access to the desktop even if the device is offline. However, if the user changes their password in the cloud, the cached verifier is not updated, which means that they can still access their local machine using their old password.

A passport and possibly associated visas are the accepted credentials for a traveler. However, those credentials might not let a traveler enter or access all resources within a country. For instance, additional credentials are required to attend a conference. In Windows, credentials can be managed to make it possible for account holders to access resources over the network without repeatedly having to supply their credentials. This type of access lets users be authenticated one time by the system to access all applications and data sources that they are authorized to use without entering another account identifier or password. The Windows platform capitalizes on the ability to use a single user identity (maintained by Active Directory) across the network by locally caching user credentials in the operating system's Local Security Authority (LSA). When a user logs on to the domain, Windows authentication packages transparently use the credentials to provide single sign-on when authenticating the credentials to network resources. For more information about credentials, see Credentials Processes in Windows Authentication.

A form of multi-factor authentication for the traveler might be the requirement to carry and present multiple documents to authenticate his identity such as a passport and conference registration information. Windows implements this form or authentication through smart cards, virtual smart cards, and biometric technologies.

Security principals and accounts

In Windows, any user, service, group, or computer that can initiate action is a security principal. Security principals have accounts, which can be local to a computer or be domain-based. For example, Windows client domain-joined computers can participate in a network domain by communicating with a domain controller even when no human user is logged on. To initiate communications, the computer must have an active account in the domain. Before accepting communications from the computer, the local security authority on the domain controller authenticates the computer's identity, and then defines the computer's security context just as it would for a human security principal. This security context defines the identity and capabilities of a user or service on a particular computer or a user, service, group, or computer on a network. For example, it defines the resources, such as a file share or printer, that can be accessed and the actions, such as Read, Write, or Modify, that can be performed by a user, service, or computer on that resource. For more information, see Security Principals.

An account is a means to identify a claimant--the human user or service--requesting access or resources. The traveler who holds the authentic passport possesses an account with the host country. Users, groups of users, objects, and services can all have individual accounts or share accounts. Accounts can be member of groups and can be assigned specific rights and permissions. Accounts can be restricted to the local computer, workgroup, network, or be assigned membership to a domain.

Built-in accounts and the security groups, of which they are members, are defined on each version of Windows. By using security groups, you can assign the same security permissions to many users who are successfully authenticated, which simplifies access administration. Rules for issuing passports might require that the traveler be assigned to certain groups, such as business, or tourist, or government. This process ensures consistent security permissions across all members of a group. By using security groups to assign permissions means that access control of resources remains constant and easy to manage and audit. By adding and removing users who require access from the appropriate security groups as needed, you can minimize the frequency of changes to access control lists (ACLs).

Standalone managed service accounts and virtual accounts were introduced in Windows Server 2008 R2 and Windows 7 to provide necessary applications, such as Microsoft Exchange Server and Internet Information Services (IIS), with the isolation of their own domain accounts, while eliminating the need for an administrator to manually administer the service principal name (SPN) and credentials for these accounts. Group managed service accounts were introduced in Windows Server 2012 and provides the same functionality within the domain but also extends that functionality over multiple servers. When connecting to a service hosted on a server farm, such as Network Load

Balance, the authentication protocols supporting mutual authentication require that all instances of the services use the same principal.

For more information about accounts, see:

- Active Directory Accounts
- Active Directory Security Groups
- Local Accounts
- Microsoft Accounts
- Service Accounts
- Special Identities

Delegated authentication

To use the travel analogy, countries might issue the same access to all members of an official governmental delegation, just as long as the delegates are well-known. This delegation lets one member act on the authority of another member. In Windows, delegated authentication occurs when a network service accepts an authentication request from a user and assumes the identity of that user in order to initiate a new connection to a second network service. To support delegated authentication, you must establish front-end or first-tier servers, such as web servers, that are responsible for handling client authentication requests and back-end or n-tier servers, such as large databases, that are responsible for storing information. You can delegate the right to set up delegated authentication to users in your organization to reduce the administrative load on your administrators.

By establishing a service or computer as trusted for delegation, you let that service or computer complete delegated authentication, receive a ticket for the user who is making the request, and then access information for that user. This model restricts data access on back-end servers just to those users or services that present credentials with the correct access control tokens. In addition, it allows for access auditing of those back-end resources. By requiring that all data be accessed by means of credentials that are delegated to the server for use on behalf of the client, you ensure that the server cannot be compromised and that you can gain access to sensitive information that is stored on other servers. Delegated authentication is useful for multitier applications that are designed to use single sign-on capabilities across multiple computers.

Authentication in trust relationships between domains

Most organizations that have more than one domain have a legitimate need for users to access shared resources that are located in a different domain, just as the traveler is permitted travel to different regions in the country. Controlling this access requires that users in one domain can also be authenticated and authorized to use resources in another domain. To provide authentication and authorization capabilities between clients and servers in different domains, there must be a trust between the two domains. Trusts are the underlying technology by which secured Active Directory communications occur and are an integral security component of the Windows Server network architecture.

When a trust exists between two domains, the authentication mechanisms for each domain trust the authentications coming from the other domain. Trusts help provide for controlled access to shared resources in a resource domain--the trusting domain--by verifying that incoming authentication requests come from a trusted authority--the trusted domain. In this way, trusts act as bridges that let only validated authentication requests travel between domains.

How a specific trust passes authentication requests depends on how it is configured. Trust relationships can be one-way, by providing access from the trusted domain to resources in the trusting domain, or two-way, by providing access from each domain to resources in the other domain. Trusts are also either nontransitive, in which case trust exists only between the two trust partner domains, or transitive, in which case trust automatically extends to any other domains that either of the partners trusts.

For information about how a trust works, see How Domain and Forest Trusts Work.

Protocol transition

Protocol transition assists application designers by letting applications support different authentication mechanisms at the user authentication tier and by switching to the Kerberos protocol for security features, such as mutual authentication and constrained delegation, in the subsequent application tiers.

For more information about protocol transition, see Kerberos Protocol Transition and Constrained Delegation.

Constrained delegation

Constrained delegation gives administrators the ability to specify and enforce application trust boundaries by limiting the scope where application services can act on behalf of a user. You can specify particular services from which a computer that is trusted for delegation can request resources. The flexibility to constrain authorization

rights for services helps improve application security design by reducing the opportunities for compromise by untrusted services.

For more information about constrained delegation, see Kerberos Constrained Delegation Overview.

Additional References

Windows Logon and Authentication Technical Overview

Feedback

Was this page helpful? Yes No

Windows logon scenarios

07/08/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This reference article for IT professionals summarizes common Windows logon and sign-in scenarios.

Windows operating systems require all users to log on to the computer with a valid account to access local and network resources. Windows-based computers secure resources by implementing the logon process, in which users are authenticated. After a user is authenticated, authorization and access control technologies implement the second phase of protecting resources: determining if the authenticated user is authorized to access a resource.

The contents of this article apply to versions of Windows designated in the **Applies to** list at the beginning of this article.

In addition, applications and services can require users to sign in to access the resources that are offered by the application or service. The sign-in process is similar to the logon process, in that a valid account and correct credentials are required, but logon information is stored in the Security Account Manager (SAM) database on the local computer and in Active Directory where applicable. Sign-in account and credential information is managed by the application or service, and optionally can be stored locally in Credential Locker.

To understand how authentication works, see Windows Authentication Concepts.

This article describes the following scenarios:

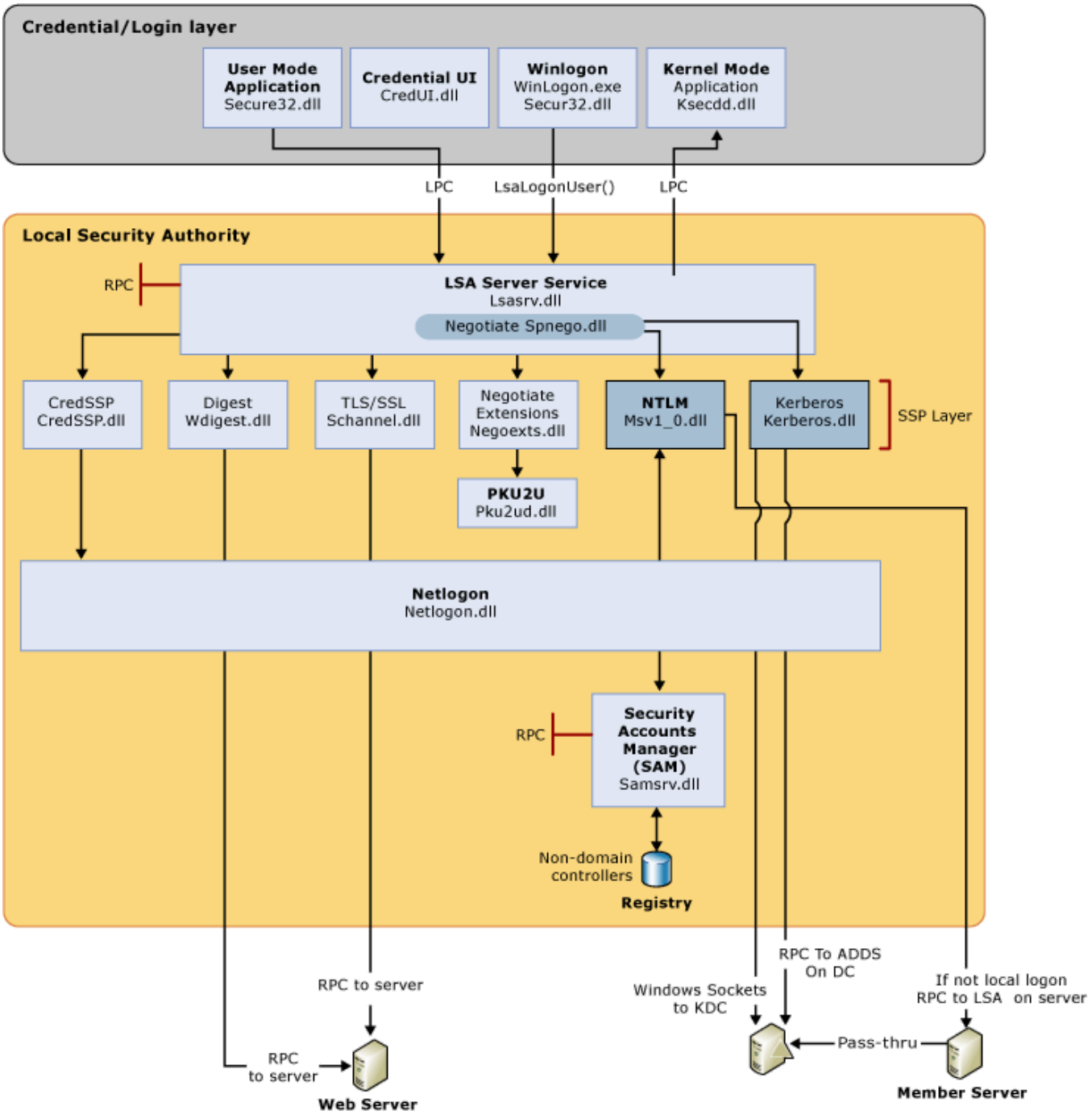
- Interactive logon
- Network logon
- Smart card logon
- Biometric logon

> ⊗ **Caution**
>
> When a user performs a local logon, their credentials are verified locally against a cached copy before being authenticated with an identity provider over the network. If the cache verification is successful, the user gains access to the desktop even if the device is offline. However, if the user changes their password in the cloud, the cached verifier isn't updated, which means that they can still access their local machine by using their old password.

Interactive logon

The logon process begins when a user enters credentials in the credentials entry dialog, when the user inserts a smart card into the smart card reader, or when the user interacts with a biometric device. Users can perform an interactive logon by using a local user account or a domain account to log on to a computer.

The following diagram shows the interactive logon elements and logon process.



Local and domain logon

Credentials that the user presents for a domain logon contain all the elements necessary for a local logon, such as account name and password or certificate, and Active Directory domain

information. The process confirms the user's identification to the security database on the user's local computer or to an Active Directory domain. This mandatory logon process can't be turned off for users in a domain.

Users can perform an interactive logon to a computer in two ways:

- Locally, when the user has direct physical access to the computer, or when the computer is part of a network of computers.

  A local logon grants a user permission to access Windows resources on the local computer. A local logon requires that the user has a user account in Security Accounts Manager (SAM) on the local computer. SAM protects and manages user and group information in the form of security accounts stored in the local computer registry. The computer can have network access, but it's not required. Local user account and group membership information is used to manage access to local resources.

  A network logon grants a user permission to access Windows resources on the local computer in addition to any resources on networked computers, as defined by the credential's access token. Both a local logon and a network logon require that the user has a user account in Security Accounts Manager (SAM) on the local computer. Local user account and group membership information is used to manage access to local resources, and the access token for the user defines what resources can be accessed on networked computers.

  A local logon and a network logon aren't sufficient to grant the user and computer permission to access and to use domain resources.

- Remotely, through Terminal Services or Remote Desktop Services (RDS), in which case the logon is further qualified as remote interactive.

After an interactive logon, Windows runs applications on behalf of the user, and the user can interact with those applications.

A local logon grants a user permission to access resources on the local computer or resources on networked computers. If the computer is joined to a domain, the Winlogon functionality attempts to log on to that domain.

A domain logon grants a user permission to access local and domain resources. A domain logon requires that the user has a user account in Active Directory. The computer must have an account in the Active Directory domain and be physically connected to the network. Users must also have the user rights to log on to a local computer or a domain. Domain user account information and group membership information are used to manage access to domain and local resources.

Remote logon

In Windows, accessing another computer through remote logon relies on Remote Desktop Protocol (RDP). Because the user must already have successfully logged on to the client computer before attempting a remote connection, interactive logon processes have successfully finished.

RDP manages the credentials that the user enters by using the Remote Desktop client. Those credentials are intended for the target computer, and the user must have an account on that target computer. In addition, the target computer must be configured to accept a remote connection. The target computer credentials are sent to attempt to perform the authentication process. If authentication is successful, the user is connected to local and network resources that are accessible by using the supplied credentials.

Network logon

A network logon can only be used after user, service, or computer authentication has occurred. During network logon, the process doesn't use the credentials entry dialog boxes to collect data. Instead, previously established credentials or another method to collect credentials is used. This process confirms the user's identity to any network service that the user is attempting to access. This process is typically invisible to the user unless alternative credentials have to be provided.

To provide this type of authentication, the security system includes these authentication mechanisms:

- Kerberos version 5 protocol
- Public key certificates
- Secure Sockets Layer/Transport Layer Security (SSL/TLS)
- Digest
- NTLM, for compatibility with Microsoft Windows NT 4.0-based systems

For information about the elements and processes, see the interactive logon diagram earlier in this article.

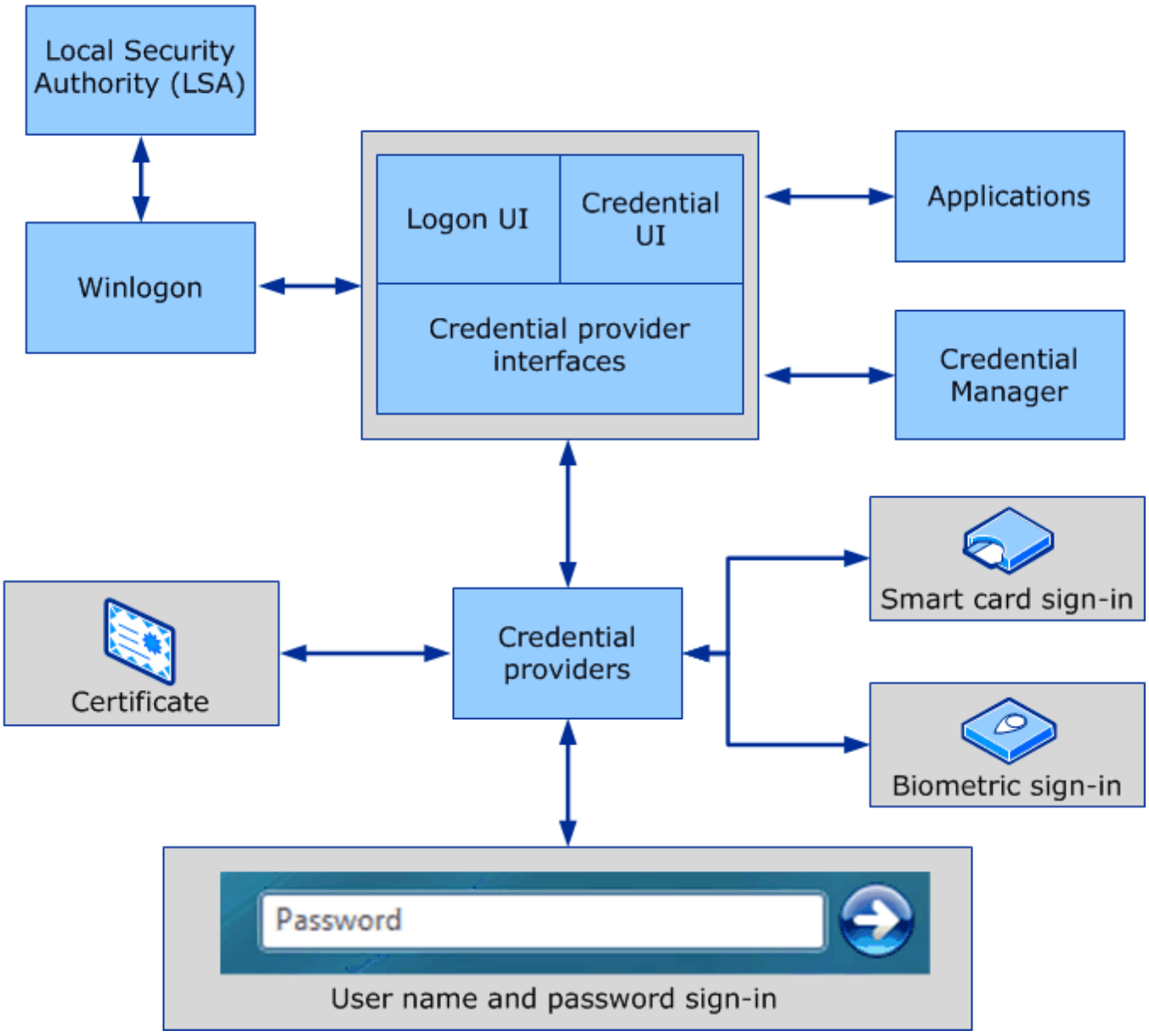
Smart card logon

Smart cards can be used to log on only to domain accounts, not local accounts. Smart card authentication requires the use of the Kerberos authentication protocol. Starting in Windows

2000 Server, in Windows-based operating systems a public key extension to the Kerberos protocol's initial authentication request is implemented. In contrast to shared secret key cryptography, public key cryptography is asymmetric. That is, two different keys are needed: one to encrypt, another to decrypt. Together, the keys that are required to perform both operations make up a private/public key pair.

To initiate a typical logon session, a user must prove their identity by providing information known only to the user and the underlying Kerberos protocol infrastructure. The secret information is a cryptographic shared key derived from the user's password. A shared secret key is symmetric, which means that the same key is used for both encryption and decryption.

The following diagram shows the elements and processes required for smart card logon.



When a smart card is used instead of a password, a private/public key pair stored on the user's smart card is substituted for the shared secret key, which is derived from the user's password. The private key is stored only on the smart card. The public key can be made available to anyone with whom the owner wants to exchange confidential information.

For more information about the smart card logon process in Windows, see How smart card logon works in Windows.

Biometric logon

A device is used to capture and build a digital characteristic of an artifact, such as a fingerprint. This digital representation is then compared to a sample of the same artifact, and when the two match, authentication can occur. Computers running any of the operating systems designated in the **Applies to** list at the beginning of this article can be configured to accept this form of logon. However, if biometric logon is only configured for local logon, the user needs to present domain credentials when accessing an Active Directory domain.

Related content

For information about how Windows manages credentials submitted during the logon process, see Credentials Management in Windows Authentication.

Windows Logon and Authentication Technical Overview

Windows Authentication Architecture

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This overview topic for the IT professional explains the basic architectural scheme for Windows authentication.

Authentication is the process by which the system validates a user's logon or sign-in information. A user's name and password are compared against an authorized list, and if the system detects a match, access is granted to the extent specified in the permission list for that user.

As part of an extensible architecture, the Windows Server operating systems implement a default set of authentication security support providers, which include Negotiate, the Kerberos protocol, NTLM, Schannel (secure channel), and Digest. The protocols used by these providers enable authentication of users, computers, and services, and the authentication process enables authorized users and services to access resources in a secure manner.

In Windows Server, applications authenticate users by using the SSPI to abstract calls for authentication. Thus, developers do not need to understand the complexities of specific authentication protocols or build authentication protocols into their applications.

Windows Server operating systems include a set of security components that make up the Windows security model. These components ensure that applications cannot gain access to resources without authentication and authorization. The following sections describe the elements of the authentication architecture.

Local Security Authority

The Local Security Authority (LSA) is a protected subsystem that authenticates and signs in users to the local computer. In addition, LSA maintains information about all aspects of local security on a computer (these aspects are collectively known as the local security policy). It also provides various services for translation between names and security identifiers (SIDs).

The security subsystem keeps track of the security policies and the accounts that are on a computer system. In the case of a domain controller, these policies and accounts are those that are in effect for the domain in which the domain controller is located. These policies and accounts are stored in Active Directory. The LSA subsystem provides

services for validating access to objects, checking user rights, and generating audit messages.

Security Support Provider Interface

The Security Support Provider Interface (SSPI) is the API that obtains integrated security services for authentication, message integrity, message privacy, and security quality-of-service for any distributed application protocol.

SSPI is the implementation of the Generic Security Service API (GSSAPI). SSPI provides a mechanism by which a distributed application can call one of several security providers to obtain an authenticated connection without knowledge of the details of the security protocol.

Additional References

- Security Support Provider Interface Architecture
- Credentials Processes in Windows Authentication
- Windows Authentication Technical Overview

Feedback

Was this page helpful? Yes No

Security Support Provider Interface Architecture

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This reference topic for the IT professional describes the Windows authentication protocols that are used within the Security Support Provider Interface (SSPI) architecture.

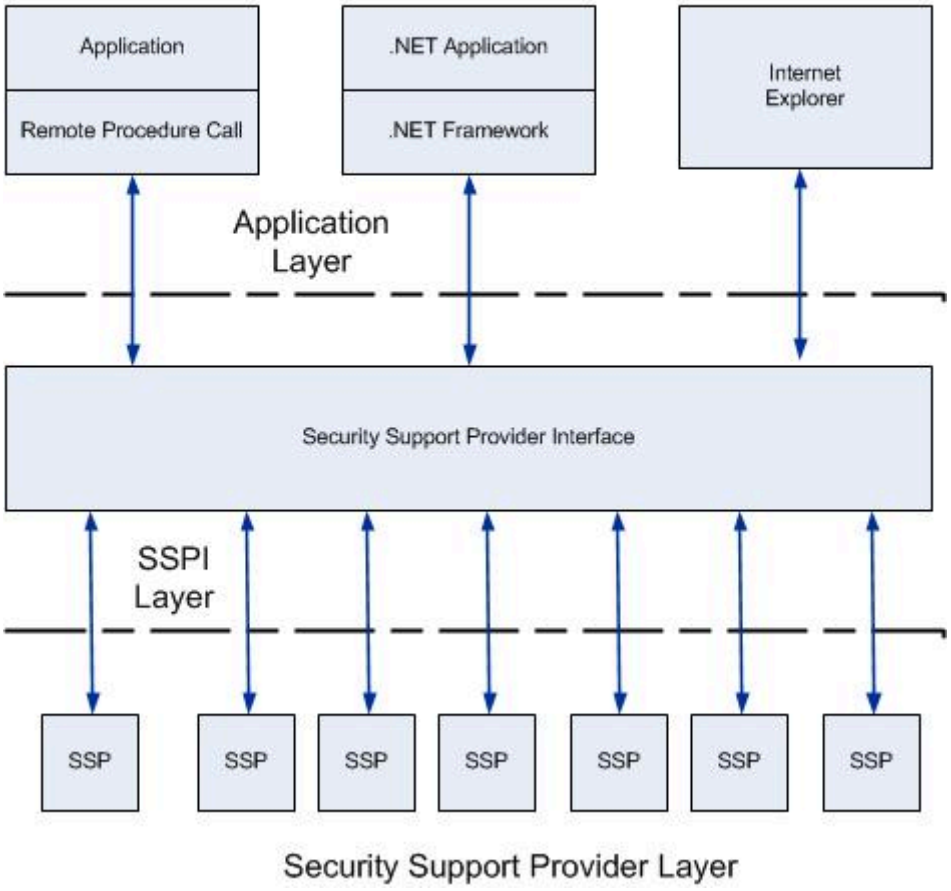
The Microsoft Security Support Provider Interface (SSPI) is the foundation for Windows authentication. Applications and infrastructure services that require authentication use SSPI to provide it.

SSPI is the implementation of the Generic Security Service API (GSSAPI) in Windows Server operating systems. For more information about GSSAPI, see RFC 2743 and RFC 2744 in the IETF RFC Database.

The default Security Support Providers (SSPs) that invoke specific authentication protocols in Windows are incorporated into the SSPI as DLLs. These default SSPs are described in the following sections. Additional SSPs can be incorporated if they can operate with the SSPI.

As shown in the following image, the SSPI in Windows provides a mechanism that carries authentication tokens over the existing communication channel between the client computer and the server. When two computers or devices need to be authenticated so that they can communicate securely, the requests for authentication are routed to the SSPI, which completes the authentication process, regardless of the network protocol currently in use. The SSPI returns transparent binary large objects. These are passed between the applications, at which point they can be passed to the SSPI layer. Thus, the SSPI enables an application to use various security models available on a computer or network without changing the interface to the security system.

Security Support Provider Interface Architecture



The following sections describe the default SSPs that interact with the SSPI. The SSPs are used in different ways in Windows operating systems to promote secure communication in an unsecure network environment.

- Kerberos Security Support Provider
- NTLM Security Support Provider
- Digest Security Support Provider
- Schannel Security Support Provider
- Negotiate Security Support Provider
- Credential Security Support Provider
- Negotiate Extensions Security Support Provider
- PKU2U Security Support Provider

Also included in this topic:

Security Support Provider selection

Kerberos Security Support Provider

This SSP uses only the Kerberos version 5 protocol as implemented by Microsoft. This protocol is based on the Network Working Group's RFC 4120 and draft revisions. It is an industry standard protocol that is used with a password or a smart card for an interactive logon. It is also the preferred authentication method for services in Windows.

Because the Kerberos protocol has been the default authentication protocol since Windows 2000, all domain services support the Kerberos SSP. These services include:

- Active Directory queries that use the Lightweight Directory Access Protocol (LDAP)
- Remote server or workstation management that uses the Remote Procedure Call service
- Print services
- Client-server authentication
- Remote file access that uses the Server Message Block (SMB) protocol (also known as Common Internet File System or CIFS)
- Distributed file system management and referral
- Intranet authentication to Internet Information Services (IIS)
- Security authority authentication for Internet Protocol security (IPsec)
- Certificate requests to Active Directory Certificate Services for domain users and computers

Location: %Windir%\System32\kerberos.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, plus Windows Server 2003 and Windows XP.

Additional resources for the Kerberos protocol and the Kerberos SSP

- Microsoft Kerberos (Windows)
- [MS-KILE]: Kerberos Protocol Extensions
- [MS-SFU]: Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification
- Kerberos SSP/AP (Windows)

- Kerberos Enhancements for Windows Vista
- Changes in Kerberos Authentication for Windows 7
- Kerberos Authentication Technical Reference

NTLM Security Support Provider

The NTLM Security Support Provider (NTLM SSP) is a binary messaging protocol used by the Security Support Provider Interface (SSPI) to allow NTLM challenge-response authentication and to negotiate integrity and confidentiality options. NTLM is used wherever SSPI authentication is used, including for Server Message Block or CIFS authentication, HTTP Negotiate authentication (for example, Internet Web Authentication), and the Remote Procedure Call service. The NTLM SSP includes the NTLM and NTLM version 2 (NTLMv2) authentication protocols.

The supported Windows operating systems can use the NTLM SSP for the following:

- Client/server authentication
- Print services
- File access by using CIFS (SMB)
- Secure Remote Procedure Call service or DCOM service

Location: %Windir%\System32\msv1_0.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, plus Windows Server 2003 and Windows XP.

Additional resources for the NTLM protocol and the NTLM SSP

- MSV1_0 Authentication Package (Windows)
- Changes in NTLM Authentication in Windows 7
- Microsoft NTLM (Windows)
- Auditing and restricting NTLM usage guide

Digest Security Support Provider

Digest authentication is an industry standard that is used for Lightweight Directory Access Protocol (LDAP) and web authentication. Digest authentication transmits credentials across the network as an MD5 hash or message digest.

Digest SSP (Wdigest.dll) is used for the following:

- Internet Explorer and Internet Information Services (IIS) access
- LDAP queries

Location: %Windir%\System32\Wdigest.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, plus Windows Server 2003 and Windows XP.

Additional resources for the Digest protocol and the Digest SSP

- Microsoft Digest Authentication (Windows)
- [MS-DPSP]: Digest Protocol Extensions

Schannel Security Support Provider

The Secure Channel (Schannel) is used for web-based server authentication, such as when a user attempts to access a secure web server.

The TLS protocol, SSL protocol , the Private Communications Technology (PCT) protocol, and the Datagram Transport Layer (DTLS) protocol are based on public key cryptography. Schannel provides all these protocols. All Schannel protocols use a client/server model. The Schannel SSP uses public key certificates to authenticate parties. When authenticating parties, Schannel SSP selects a protocol in the following order of preference:

- Transport Layer Security (TLS) version 1.0
- Transport Layer Security (TLS) version 1.1
- Transport Layer Security (TLS) version 1.2
- Secure Socket Layer (SSL) version 2.0
- Secure Socket Layer (SSL) version 3.0
- Private Communications Technology (PCT)

  Note PCT is disabled by default.

The protocol that is selected is the preferred authentication protocol that the client and the server can support. For example, if a server supports all the Schannel protocols and the client supports only SSL 3.0 and SSL 2.0, the authentication process uses SSL 3.0.

DTLS is used when explicitly called by the application. For more information about DTLS and the other protocols that are used by the Schannel provider, see Schannel Security Support Provider Technical Reference.

Location: %Windir%\System32\Schannel.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, plus Windows Server 2003 and Windows XP.

> **Note**
>
> TLS 1.2 was introduced in this provider in Windows Server 2008 R2 and Windows 7. DTLS was introduced in this provider in Windows Server 2012 and Windows 8.

Additional resources for the TLS and SSL protocols and the Schannel SSP

- Secure Channel (Windows)
- TLS/SSL Technical Reference
- [MS-TLSP]: Transport Layer Security (TLS) Profile

Negotiate Security Support Provider

The Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) forms the basis for the Negotiate SSP, which can be used to negotiate a specific authentication protocol. When an application calls into SSPI to log on to a network, it can specify an SSP to process the request. If the application specifies the Negotiate SSP, it analyzes the request and picks the appropriate provider to handle the request, based on customer-configured security policies.

SPNEGO is specified in RFC 2478.

In supported versions of the Windows operating systems, the Negotiate security support provider selects between the Kerberos protocol and NTLM. Negotiate selects the Kerberos protocol by default unless that protocol cannot be used by one of the systems involved in the authentication, or the calling application did not provide sufficient information to use the Kerberos protocol.

Location: %Windir%\System32\lsasrv.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, plus Windows Server 2003 and Windows XP.

Additional resources for the Negotiate SSP

- Microsoft Negotiate (Windows)
- [MS-SPNG]: Simple and Protected GSS-API Negotiation Mechanism (SPNEGO) Extensions
- [MS-N2HT]: Negotiate and Nego2 HTTP Authentication Protocol Specification

Credential Security Support Provider

The Credential Security Service Provider (CredSSP) provides a single sign-on (SSO) user experience when starting new Terminal Services and Remote Desktop Services sessions. CredSSP enables applications to delegate users' credentials from the client computer (by using the client-side SSP) to the target server (through the server-side SSP), based on the client's policies. CredSSP policies are configured by using Group Policy, and the delegation of credentials is turned off by default.

Location: %Windir%\System32\credssp.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic.

Additional resources for the Credentials SSP

- [MS-CSSP]: Credential Security Support Provider (CredSSP) Protocol Specification
- Credential Security Service Provider and SSO for Terminal Services Logon

Negotiate Extensions Security Support Provider

Negotiate Extensions (NegoExts) is an authentication package that negotiates the use of SSPs, other than NTLM or the Kerberos protocol, for applications and scenarios implemented by Microsoft and other software companies.

This extension to the Negotiate package permits the following scenarios:

- **Rich client availability within a federated system.** Documents can be accessed on SharePoint sites, and they can be edited by using a full-featured Microsoft Office application.
- **Rich client support for Microsoft Office services.** Users can sign in to Microsoft Office services and use a full-featured Microsoft Office application.

- **Hosted Microsoft Exchange Server and Outlook.** There is no domain trust established because Exchange Server is hosted on the web. Outlook uses the Windows Live service to authenticate users.
- **Rich client availability between client computers and servers.** The operating system's networking and authentication components are used.

The Windows Negotiate package treats the NegoExts SSP in the same manner as it does for Kerberos and NTLM. NegoExts.dll is loaded into the Local System Authority (LSA) at startup. When an authentication request is received, based on the request's source, NegoExts negotiates between the supported SSPs. It gathers the credentials and policies, encrypts them, and sends that information to the appropriate SSP, where the security token is created.

The SSPs supported by NegoExts are not stand-alone SSPs such as Kerberos and NTLM. Therefore, within the NegoExts SSP, when the authentication method fails for any reason, an authentication failure message will be displayed or logged. No renegotiation or fallback authentication methods are possible.

Location: %Windir%\System32\negoexts.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, excluding Windows Server 2008 and Windows Vista.

PKU2U Security Support Provider

The PKU2U protocol was introduced and implemented as an SSP in Windows 7 and Windows Server 2008 R2 . This SSP enables peer-to-peer authentication, particularly through the media and file sharing feature called HomeGroup, which was introduced in Windows 7 . The feature permits sharing between computers that are not members of a domain.

Location: %Windir%\System32\pku2u.dll

This provider is included by default in versions designated in the **Applies to** list at the beginning of this topic, excluding Windows Server 2008 and Windows Vista.

Additional resources for the PKU2U protocol and the PKU2U SSP

- Introducing Online Identity Integration

Security Support Provider selection

The Windows SSPI can use any of the protocols that are supported through the installed Security Support Providers. However, because not all operating systems support the same SSP packages as any given computer running Windows Server, clients and servers must negotiate to use a protocol that they both support. Windows Server prefers client computers and applications to use the Kerberos protocol, a strong standards-based protocol, when possible, but the operating system continues to allow client computers and client applications that do not support the Kerberos protocol to authenticate.

Before authentication can take place the two communicating computers must agree on a protocol that they both can support. For any protocol to be usable through the SSPI, each computer must have the appropriate SSP. For example, for a client computer and server to use the Kerberos authentication protocol, they must both support Kerberos v5. Windows Server uses the function **EnumerateSecurityPackages** to identify which SSPs are supported on a computer and what the capabilities of those SSPs are.

The selection of an authentication protocol can be handled in one of the following two ways:

1. Single authentication protocol
2. Negotiate option

Single authentication protocol

When a single acceptable protocol is specified on the server, the client computer must support the protocol specified or the communication fails. When a single acceptable protocol is specified, the authentication exchange takes place as follows:

1. The client computer requests access to a service.
2. The server replies to the request and specifies the protocol that will be used.
3. The client computer examines the contents of the reply and checks to determine whether it supports the specified protocol. If the client computer does support the specified protocol, the authentication continues. If the client computer does not support the protocol, the authentication fails, regardless of whether the client computer is authorized to access the resource.

Negotiate option

The negotiate option can be used to allow the client and server to attempt to find an acceptable protocol. This is based on the Simple and Protected GSS-API Negotiation

Mechanism (SPNEGO). When the authentication begins with the option to negotiate for an authentication protocol, the SPNEGO exchange takes place as follows:

1. The client computer requests access to a service.
2. The server replies with a list of authentication protocols that it can support and an authentication challenge or response, based on the protocol that is its first choice. For example, the server might list the Kerberos protocol and NTLM, and send a Kerberos authentication response.
3. The client computer examines the contents of the reply and checks to determine whether it supports any of the specified protocols.
    - If the client computer supports the preferred protocol, authentication proceeds.
    - If the client computer does not support the preferred protocol, but it does support one of the other protocols listed by the server, the client computer lets the server know which authentication protocol it supports, and the authentication proceeds.
    - If the client computer does not support any of the listed protocols, the authentication exchange fails.

Additional References

Windows Authentication Architecture

Feedback

Was this page helpful? Yes No

Credentials processes in Windows authentication

06/30/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows authentication is a cornerstone of secure access in enterprise environments, enabling users and services to interact with systems while safeguarding sensitive information. This article provides IT professionals with a comprehensive overview of how Windows processes credentials during authentication. It covers key components, mechanisms, and architectures involved in credential management, validation, and storage, ensuring a clear understanding of the authentication process across various Windows operating systems.

Specifically, these components are covered:

- Windows credentials authentication overview
- Credential input for user logon
- Credential input for application and service logon
- Local Security Authority
- Cached credentials and validation
- Credential storage and validation
- Security Accounts Manager database
- Local domains and trusted domains
- Certificates in Windows authentication

Windows credentials authentication overview

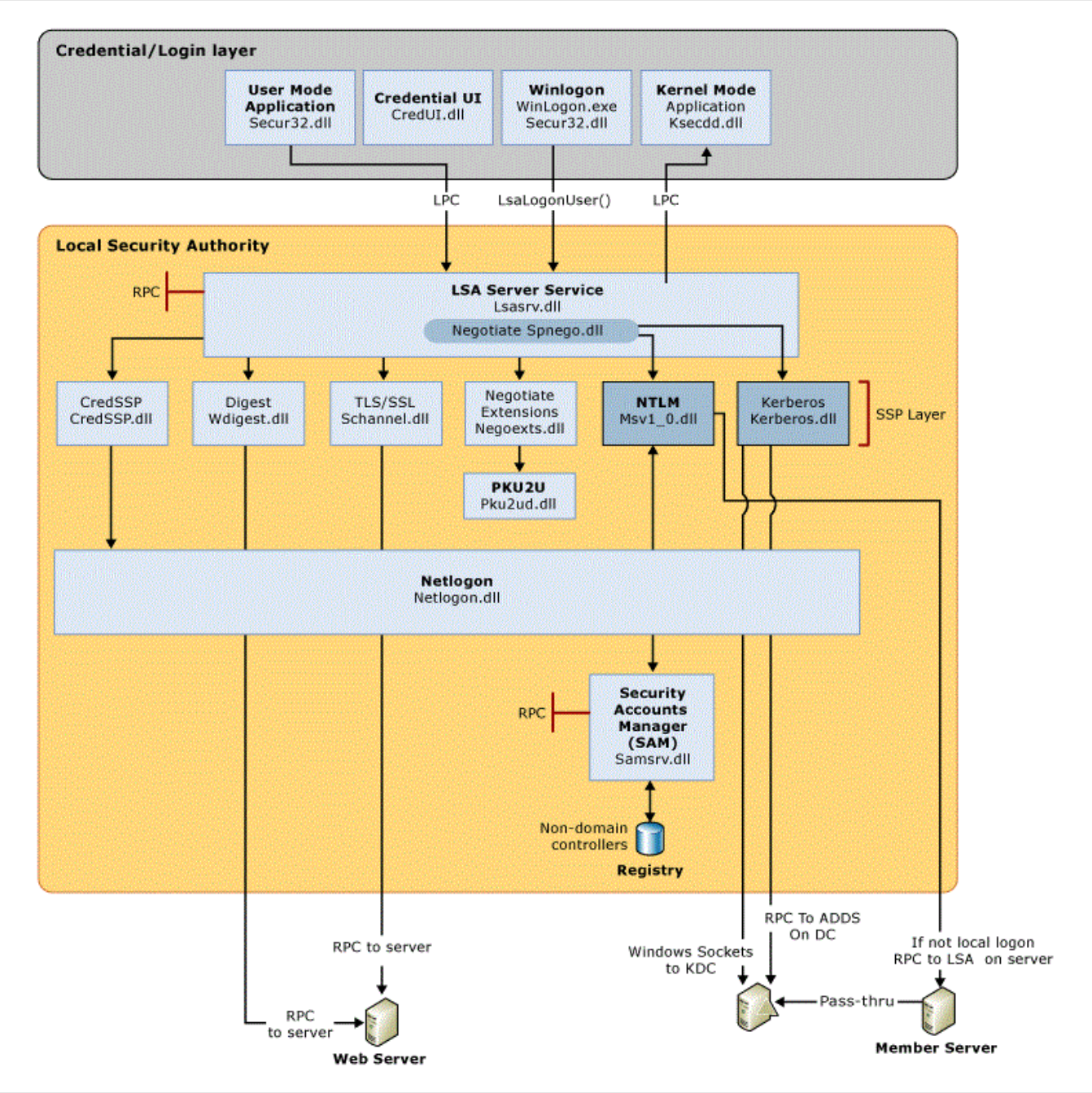
Windows credentials management is the process by which the operating system receives the credentials from the service or user and secures that information for future presentation to the authenticating target. If a computer is domain-joined, the authenticating target is the domain controller. The credentials used in authentication are digital documents that associate the user's identity to some form of proof of authenticity, such as a certificate, a password, or a PIN.

By default, Windows credentials are validated against the Security Accounts Manager (SAM) database on the local computer, or against Active Directory on a domain-joined computer,

through the Winlogon service. Credentials are collected through user input on the logon user interface or programmatically via the application programming interface (API) to be presented to the authenticating target.

Local security information is stored in the registry under **HKEY_LOCAL_MACHINE\SECURITY**. Stored information includes policy settings, default security values, and account information, such as cached logon credentials. A copy of the SAM database is also stored here, although it's write-protected.

The following diagram shows the components that are required and the paths that credentials take through the system to authenticate the user or process for a successful logon.



The following table describes each component that manages credentials in the authentication process at the point of logon for all systems.

Expand table

Component	Description
User logon	Winlogon.exe is the executable file responsible for managing secure user interactions. The Winlogon service initiates the logon process for Windows operating systems by passing the credentials collected by user action on the secure desktop (Logon UI) to the Local Security Authority (LSA) through `secur32.dll`.
Application logon	Application or service logons that don't require interactive logon. Most processes initiated by the user run in user mode by using `secur32.dll` whereas processes initiated at startup, such as services, run in kernel mode by using Ksecdd.sys. For more information about user mode and kernel mode, see Applications and User Mode or Services and Kernel Mode.
`secur32.dll`	The multiple authentication providers that form the foundation of the authentication process.
`lsasrv.dll`	The LSA Server service, which enforces security policies and acts as the security package manager for the LSA. The LSA contains the Negotiate function, which selects either the NTLM or Kerberos protocol after determining which protocol is to be successful.
Security Support Providers	A set of providers that can individually invoke one or more authentication protocols. The default set of providers can change with each version of the Windows operating system, and custom providers can be written.
`netlogon.dll`	The services that the Net Logon service performs are as follows: • Maintains the computer's secure channel (not to be confused with `Schannel`) to a domain controller. • Passes the user's credentials through a secure channel to the domain controller and returns the domain security identifiers (SIDs) and user rights for the user. • Publishes service resource records in the Domain Name System (DNS) and uses DNS to resolve names to the Internet Protocol (IP) addresses of domain controllers. • Implements the replication protocol based on remote procedure call (RPC) for synchronizing primary domain controllers (PDCs) and backup domain controllers (BDCs).
`samsrv.dll`	The Security Accounts Manager (SAM), which stores local security accounts, enforces locally stored policies and supports APIs.
Registry	The Registry contains a copy of the SAM database, local security policy settings, default security values, and account information that is only accessible to the system.

Credential input for user logon

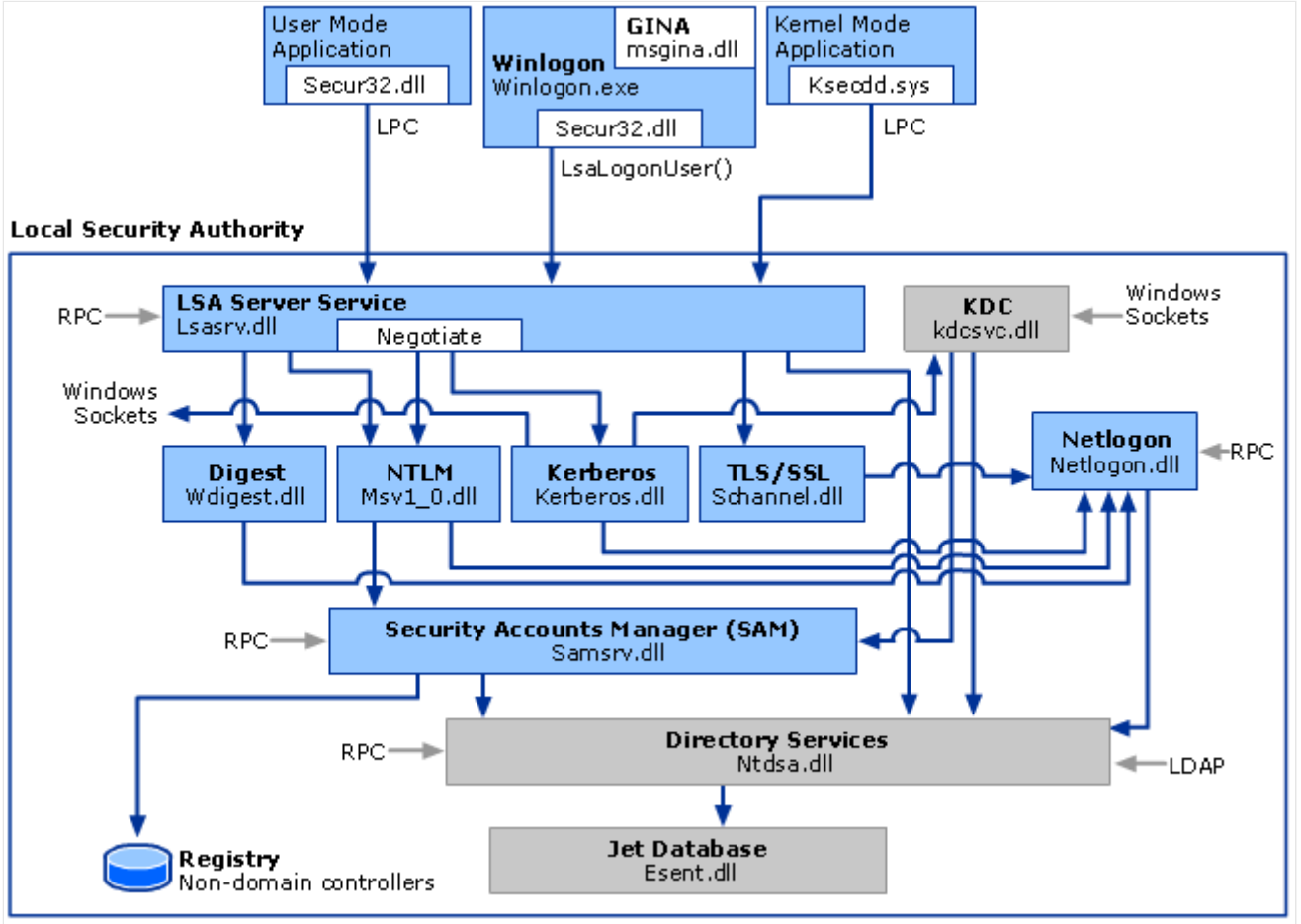
The credential provider architecture, which was introduced in Windows Server 2008, is the primary mechanism for collecting user credentials during logon. It allows for a flexible and extensible way to gather credentials from users, such as passwords, smart cards, or biometrics. The credential provider architecture replaces the older Graphical Identification and Authentication (GINA) architecture used in earlier versions of Windows.

▼ To learn about the Graphical Identification and Authentication (GINA) architecture in older versions of Windows, expand this section.

The Graphical Identification and Authentication (GINA) architecture applies to the Windows Server 2003, Microsoft Windows 2000 Server, Windows XP, and Windows 2000 Professional operating systems. In these systems, every interactive logon session creates a separate instance of the Winlogon service. The GINA architecture is loaded into the process space used by Winlogon, receives and processes the credentials, and makes the calls to the authentication interfaces through LSALogonUser.

The instances of Winlogon for an interactive logon run in Session 0. Session 0 hosts system services and other critical processes, including the Local Security Authority (LSA) process.

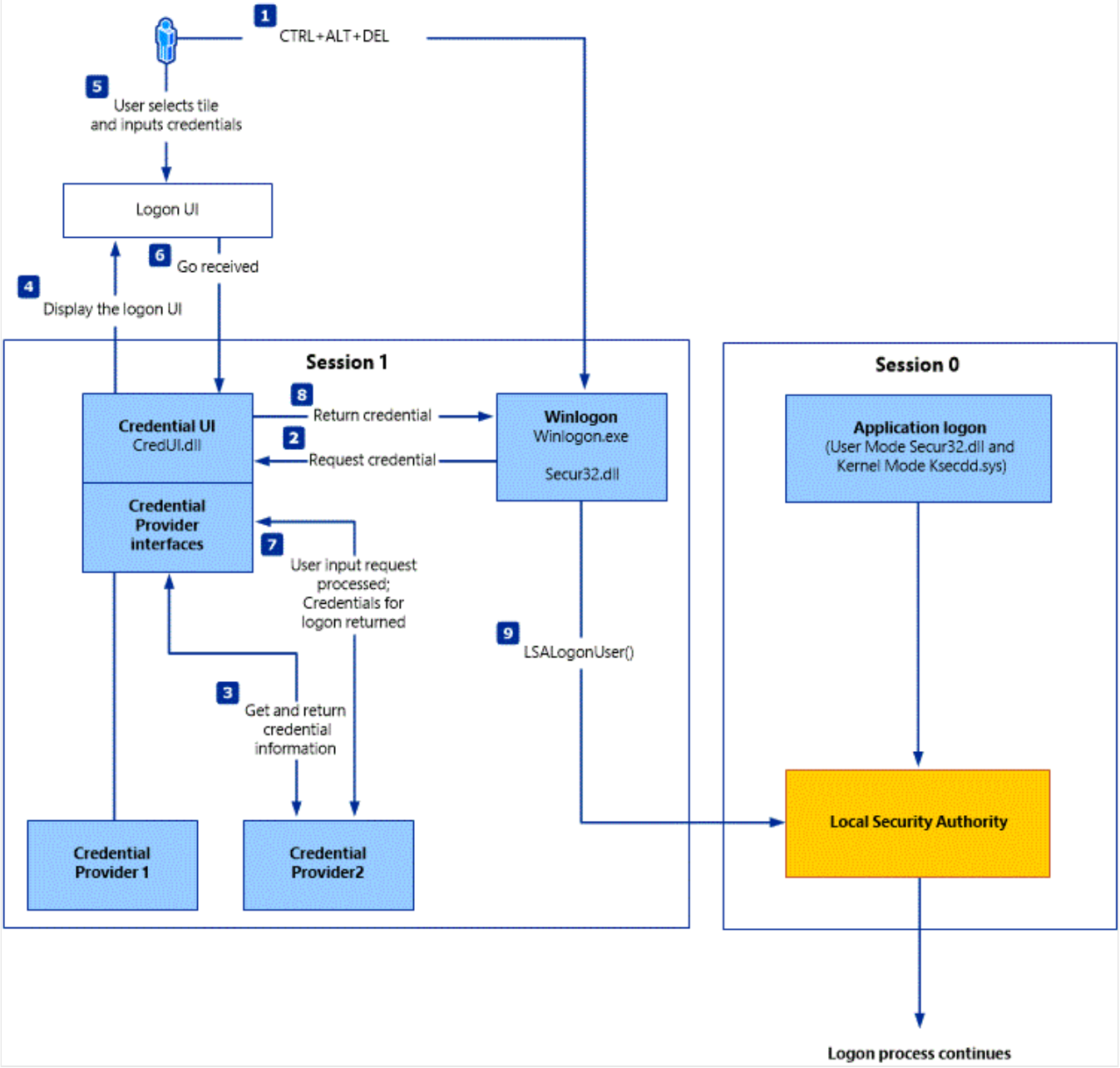
The following diagram shows the credential process for Windows Server 2003, Microsoft Windows 2000 Server, Windows XP, and Microsoft Windows 2000 Professional.



Credential provider architecture

The credential provider architecture uses an extensible design by using credential providers. Different logon tiles represent these providers on the secure desktop that permit any number of logon scenarios, including different accounts for the same user and different authentication methods, such as password, smart card, and biometrics.

The following diagram shows the credential process for the credential provider architecture:



Winlogon always starts the process Logon UI after it receives a secure attention sequence event. Logon UI queries each credential provider for the number of different credential types the provider is configured to enumerate. Credential providers have the option of specifying one of these tiles as the default. After all providers enumerate their tiles, Logon UI displays them to the user. The user interacts with a tile to supply their credentials. Logon UI submits these credentials for authentication.

Credential providers aren't enforcement mechanisms; they're used to gather and serialize credentials. The Local Security Authority and authentication packages enforce security.

Credential providers are registered on the computer and are responsible for the following tasks:

- Describing the credential information required for authentication.
- Handling communication and logic with external authentication authorities.
- Packaging credentials for interactive and network logon.

Packaging credentials for interactive and network logon includes the process of serialization. By serializing credentials, multiple logon tiles can be displayed to a user. Therefore, your organization can control the logon display such as users, target systems for logon, pre-logon access to the network and workstation lock/unlock policies, by using customized credential providers. Multiple credential providers can coexist on the same computer. Single sign-on (SSO) providers can be developed as a standard credential provider or as a Pre-Logon Access Provider (PLAP).

Each version of Windows contains one default credential provider and one default Pre-Logon Access Provider, also known as the SSO provider. The SSO provider permits users to make a connection to a network before logging on to the local computer. When this provider is implemented, the provider doesn't enumerate tiles on Logon UI.

An SSO provider is intended to be used in the following scenarios:

- **Network authentication and computer logon are handled by different credential providers**, including the following variations to this scenario:
    - A user has the option of connecting to a network, such as connecting to a virtual private network (VPN), before logging on to the computer but isn't required to make this connection.
    - Network authentication is required to retrieve information used during interactive authentication on the local computer.
    - Multiple network authentications are followed by one of the other scenarios. For example, a user authenticates to an Internet service provider (ISP), authenticates to a VPN, and then uses their user account credentials to log on locally.
    - Cached credentials are disabled, and a Remote Access Services connection through VPN is required before local logon to authenticate the user.

    - A domain user doesn't have a local account set up on a domain-joined computer and must establish a Remote Access Services connection through VPN connection before completing interactive logon.

- **Network authentication and computer logon are handled by the same credential provider**, where the user is required to connect to the network before logging on to the computer.

Logon tile enumeration

The credential provider enumerates logon tiles in the following instances:

- For workstation logon. The credential provider typically serializes credentials for authentication to the local security authority. This process displays tiles specific for each user and specific to each user's target systems.

- The logon and authentication architecture lets a user use tiles enumerated by the credential provider to unlock a workstation. Typically, the currently logged-on user is the default tile, but if more than one user is logged on, numerous tiles are displayed.

- The credential provider enumerates tiles in response to a user request to change their password or other private information, such as a PIN. Typically, the currently logged-on user is the default tile, but if more than one user is logged on, numerous tiles are displayed.

- The credential provider enumerates tiles based on the serialized credentials to be used for authentication on remote computers. Credential UI doesn't use the same instance of the provider as the Logon UI, unlocking workstation, or changing a password. Therefore, state information can't be maintained in the provider between instances of Credential UI. This structure results in one tile for each remote computer logon, assuming the credentials are correctly serialized. This scenario is also used in User Account Control (UAC), which can help prevent unauthorized changes to a computer by prompting the user for permission or an administrator password before permitting actions that could potentially affect the computer's operation or that could change settings that affect other users of the computer.

Credential input for application and service logon

Windows authentication is designed to manage credentials for applications or services that don't require user interaction. Applications in user mode are limited in terms of what system resources they have access to, while services can have unrestricted access to the system memory and external devices.

System services and transport-level applications access a Security Support Provider (SSP) through the Security Support Provider Interface (SSPI) in Windows, which provides functions for enumerating the security packages available on a system, selecting a package, and using that package to obtain an authenticated connection.

When a client/server connection is authenticated:

- The application on the client side of the connection sends credentials to the server by using the SSPI function `InitializeSecurityContext (General)`.
- The application on the server side of the connection responds with the SSPI function `AcceptSecurityContext (General)`.
- The SSPI functions `InitializeSecurityContext (General)` and `AcceptSecurityContext (General)` are repeated until all the necessary authentication messages are exchanged to either succeed or fail authentication.
- After the connection is authenticated, the LSA on the server uses information from the client to build the security context, which contains an access token.
- The server can then call the SSPI function `ImpersonateSecurityContext` to attach the access token to an impersonation thread for the service.

Applications and user mode

User mode in Windows is composed of two systems capable of passing I/O requests to the appropriate kernel-mode drivers: the environment system, which runs applications written for many different types of operating systems, and the integral system, which operates system-specific functions on behalf of the environment system.

The integral system manages operating system-specific functions on behalf of the environment system and consists of a security system process (the LSA), a workstation service, and a server service. The security system process deals with security tokens, grants or denies permissions to access user accounts based on resource permissions, handles logon requests and initiates logon authentication, and determines which system resources the operating system needs to audit.

Applications can run in user mode where the application can run as any principal, including in the security context of Local System (SYSTEM). Applications can also run in kernel mode where the application can run in the security context of Local System (SYSTEM).

SSPI is available through the `secur32.dll` module, which is an API used for obtaining integrated security services for authentication, message integrity, and message privacy. It

provides an abstraction layer between application-level protocols and security protocols. Because different applications require different ways of identifying or authenticating users and different ways of encrypting data as it travels across a network, SSPI provides a way to access dynamic-link libraries (DLLs) that contain different authentication and cryptographic functions. These DLLs are called Security Support Providers (SSPs).

Managed service accounts and virtual accounts were introduced in Windows Server 2008 R2 and Windows 7 to provide crucial applications, such as Microsoft SQL Server and Internet Information Services (IIS), with the isolation of their own domain accounts, while eliminating the need for an administrator to manually administer the service principal name (SPN) and credentials for these accounts. For more information about these features and their role in authentication, see Managed Service Accounts Documentation for Windows 7 and Windows Server 2008 R2 and Group Managed Service Accounts Overview.

Services and kernel mode

Even though most Windows applications run in the security context of the user who starts them, this isn't true of services. The service controller many Windows services, such as network and printing services, when the user starts the computer. These services might run as Local Service or Local System and might continue to run after the last human user logs off.

> **Note**
>
> Services normally run in security contexts known as Local System (SYSTEM), Network Service, or Local Service. Windows Server 2008 R2 introduced services that run under a managed service account, which are domain principals.

Before starting a service, the service controller logs on by using the account that is designated for the service, and then presents the service's credentials for authentication by the LSA. The Windows service implements a programmatic interface that the service controller manager can use to control the service. A Windows service can be started automatically when the system is started or manually with a service control program. For example, when a Windows client computer joins a domain, the messenger service on the computer connects to a domain controller and opens a secure channel to it. To obtain an authenticated connection, the service must have credentials that the remote computer's Local Security Authority (LSA) trusts. When communicating with other computers in the network, LSA uses the credentials for the local computer's domain account, as do all other services running in the security context of the Local System and Network Service. Services on the local computer run as SYSTEM so credentials don't need to be presented to the LSA.

The file `ksecdd.sys` manages and encrypts these credentials and uses a local procedure call into the LSA. The file type is DRV (driver) and is known as the kernel-mode Security Support Provider (SSP) and is **FIPS 140-2 Level 1** compliant, starting in Windows Server 2008.

Kernel mode has full access to the hardware and system resources of the computer. The kernel mode stops user-mode services and applications from accessing critical areas of the operating system that they shouldn't have access to.

Local Security Authority

The Local Security Authority (LSA) is a protected system process that authenticates and logs users on to the local computer. In addition, LSA maintains information about all aspects of local security on a computer (these aspects are collectively known as the local security policy), and it provides various services for translation between names and security identifiers (SIDs). The security system process, Local Security Authority Server Service (LSASS), keeps track of the security policies and the accounts that are in effect on a computer system.

The LSA validates a user's identity based on which of the following two entities issued the user's account:

- **Local Security Authority**: the LSA can validate user information by checking the Security Accounts Manager (SAM) database located on the same computer. Any workstation or member server can store local user accounts and information about local groups. However, these accounts can be used for accessing only that workstation or computer.
- **Security authority for the local domain or for a trusted domain**: the LSA contacts the entity that issued the account and requests verification that the account is valid and that the request originated from the account holder.

The Local Security Authority Subsystem Service (LSASS) stores credentials in memory on behalf of users with active Windows sessions. The stored credentials let users seamlessly access network resources, such as file shares, Exchange Server mailboxes, and SharePoint sites, without reentering their credentials for each remote service.

LSASS can store credentials in multiple forms, including:

- Reversibly encrypted plaintext.
- Kerberos tickets (ticket-granting tickets (TGTs), service tickets).
- NT hash.
- LAN Manager (LM) hash.

If the user logs on to Windows by using a smart card, LSASS doesn't store a plaintext password, but it stores the corresponding NT hash value for the account and the plaintext PIN for the smart card. If the account attribute is enabled for a smart card that is required for interactive logon, a random NT hash value is automatically generated for the account instead of the original password hash. The password hash that is automatically generated when the attribute is set doesn't change.

If a user logs on to a Windows-based computer with a password that is compatible with LAN Manager (LM) hashes, this authenticator is present in memory. The storage of plaintext credentials in memory can't be disabled, even if the credential providers that require them are disabled.

The stored credentials are directly associated with the Local Security Authority Subsystem Service (LSASS) logon sessions that are started after the last restart and aren't closed. For example, LSA sessions with stored LSA credentials are created when a user does any of the following actions:

- Logs on to a local session or Remote Desktop Protocol (RDP) session on the computer.
- Runs a task by using the **RunAs** option.
- Runs an active Windows service on the computer.
- Runs a scheduled task or batch job.
- Runs a task on the local computer by using a remote administration tool.

In some circumstances, the LSA secrets, which are secret pieces of data that are accessible only to SYSTEM account processes, are stored on the hard disk drive. Some of these secrets are credentials that must persist after reboot, and they're stored in encrypted form on the hard disk drive. Credentials stored as LSA secrets might include:

- Account password for the computer's Active Directory Domain Services (AD DS) account.
- Account passwords for Windows services that are configured on the computer.
- Account passwords for configured scheduled tasks.
- Account passwords for IIS application pools and websites.
- Passwords for Microsoft accounts.

The Windows client operating system provides extra protection for the LSA to prevent reading memory and code injection by non-protected processes, which was introduced in Windows 8.1. This protection increases security for the credentials that the LSA stores and manages.

For more information about these extra protections, see Configuring Additional LSA Protection.

Cached credentials and validation

Validation mechanisms rely on the presentation of credentials at the time of logon. However, when the computer is disconnected from a domain controller, and the user is presenting domain credentials, Windows uses the process of cached credentials in the validation mechanism.

Each time a user logs on to a domain, Windows caches the credentials supplied and stores them in the security hive in the registry of the operation system. With cached credentials, the user can log on to a domain member without being connected to a domain controller within that domain.

Credential storage and validation

It isn't always desirable to use one set of credentials for access to different resources. For example, an administrator might want to use administrative rather than user credentials when accessing a remote server. Similarly, if a user accesses external resources, such as a bank account, they can only use credentials that are different than their domain credentials.

Remote logon credential processes

The Remote Desktop Protocol (RDP) manages the credentials of the user who connects to a remote computer by using the Remote Desktop Client, which was introduced in Windows 8. The credentials in plaintext form are sent to the target host where the host attempts to perform the authentication process, and, if successful, connects the user to allowed resources. RDP doesn't store the credentials on the client, but the user's domain credentials are stored in the LSASS.

Restricted Admin mode provides extra security to remote logon scenarios, which was introduced in Windows Server 2012 R2 and Windows 8.1. This mode of Remote Desktop causes the client application to perform a network logon challenge-response with the NT one-way function (NTOWF) or use a Kerberos service ticket when authenticating to the remote host. After the administrator is authenticated, the administrator doesn't have the respective account credentials in LSASS because they weren't supplied to the remote host. Instead, the administrator has the computer account credentials for the session. Administrator credentials aren't supplied to the remote host, so actions are performed as the computer account. Resources are also limited to the computer account, and the administrator can't access resources with their own account.

Automatic restart sign-on credential process

When a user signs in, LSA saves the user credentials in encrypted memory that are accessible only by LSASS.exe, which was introduced in Windows 8.1. When Windows Update initiates an automatic restart without user presence, these credentials are used to configure Autologon for the user.

On restart, the user is automatically signed in via the Autologon mechanism, and then the computer is additionally locked to protect the user's session. The locking is initiated through Winlogon whereas the credential management is done by LSA. By automatically signing in and locking the user's session on the console, the user's lock screen applications is restarted and available.

For more information about ARSO, see Winlogon Automatic Restart Sign-On (ARSO).

Windows Vault and Credential Manager

Credential Manager is a Control Panel feature to store and manage user names and passwords, which was introduced in Windows Server 2008 R2 and Windows 7. Credential Manager lets users store credentials relevant to other systems and websites in the secure Windows Vault.

Users can save and store credentials from supported browsers and Windows applications on the local computer to make it convenient when they need to sign in to these resources. Credentials are saved in special encrypted folders on the computer under the user's profile. Applications that support this feature (by using the Credential Manager APIs), such as web browsers and apps, can present the correct credentials to other computers and websites during the logon process.

When a website, an application, or another computer requests authentication through NTLM or the Kerberos protocol, a dialog box appears in which you select the **Update Default Credentials** or **Save Password** check box. An application that supports the Credential Manager APIs generates this dialog box that lets a user save credentials locally. If the user selects the **Save Password** check box, Credential Manager keeps track of the user's user name, password, and related information for the authentication service that is in use.

The next time the service is used, Credential Manager automatically supplies the credential that is stored in the Windows Vault. If it isn't accepted, the user is prompted for the correct access information. If access is granted with the new credentials, Credential Manager overwrites the previous credential with the new one and then stores the new credential in the Windows Vault.

Security Accounts Manager database

The Security Accounts Manager (SAM) is a database that stores local user accounts and groups. It's present in every Windows operating system; however, when a computer is joined to a domain, Active Directory manages domain accounts in Active Directory domains.

For example, client computers running a Windows operating system participate in a network domain by communicating with a domain controller even when no human user is logged on. To initiate communications, the computer must have an active account in the domain. Before accepting communications from the computer, the LSA on the domain controller authenticates the computer's identity and then constructs the computer's security context just as it does for a human security principal. This security context defines the identity and capabilities of a user or service on a particular computer or a user, service, or computer on a network. For example, the access token contained within the security context defines the resources (such as a file share or printer) that can be accessed and the actions (such as Read, Write, or Modify) that a principal (a user, computer, or service) can perform on that resource.

The security context of a user or computer can vary from one computer to another, such as when a user logs on to a server or a workstation other than the user's own primary workstation. It can also vary from one session to another, such as when an administrator modifies the user's rights and permissions. In addition, the security context is usually different when a user or computer is operating on a stand-alone basis, in a network, or as part of an Active Directory domain.

Local domains and trusted domains

When a trust exists between two domains, the authentication mechanisms for each domain rely on the validity of the authentications coming from the other domain. Trusts help to provide controlled access to shared resources in a resource domain (the trusting domain) by verifying that incoming authentication requests come from a trusted authority (the trusted domain). In this way, trusts act as bridges that let only validated authentication requests travel between domains.

How a specific trust passes authentication requests depends on how it's configured. Trust relationships can be one-way, by providing access from the trusted domain to resources in the trusting domain, or two-way, by providing access from each domain to resources in the other domain. Trusts are also either non-transitive, in which case a trust exists only between the two trust partner domains, or transitive, in which case a trust automatically extends to any other domains that either of the partners trusts.

For information about domain and forest trust relationships regarding authentication, see Delegated Authentication and Trust Relationships.

Certificates in Windows authentication

A public key infrastructure (PKI) is the combination of software, encryption technologies, processes, and services that enable an organization to secure its communications and business transactions. The ability of a PKI to secure communications and business transactions is based on the exchange of digital certificates between authenticated users and trusted resources.

A digital certificate is an electronic document that contains information about the entity it belongs to, the entity who issued it, a unique serial number or some other unique identification, issuance and expiration dates, and a digital fingerprint.

Authentication is the process of determining if a remote host can be trusted. To establish its trustworthiness, the remote host must provide an acceptable authentication certificate.

Remote hosts establish their trustworthiness by obtaining a certificate from a certification authority (CA). The CA can, in turn, have certification from a higher authority, which creates a chain of trust. To determine whether a certificate is trustworthy, an application must determine the identity of the root CA, and then determine if it's trustworthy.

Similarly, the remote host or local computer must determine if the certificate presented by the user or application is authentic. The certificate presented by the user through the LSA and SSPI is evaluated for authenticity on the local computer for local logon, on the network, or on the domain through the certificate stores in Active Directory.

To produce a certificate, authentication data passes through hash algorithms, such as Secure Hash Algorithm (SHA), to produce a message digest. The message digest is then digitally signed by using the sender's private key to prove that the sender produced the message digest.

Smart card authentication

Smart card technology is an example of certificate-based authentication. Logging on to a network with a smart card provides a strong form of authentication because it uses cryptography-based identification and proof of possession when authenticating a user to a domain. Active Directory Certificate Services (AD CS) provides the cryptographic-based identification through the issuance of a logon certificate for each smart card.

Virtual smart card technology was introduced in Windows 8. It stores the smart card's certificate in the PC, and then protects it by using the device's tamper-proof Trusted Platform Module (TPM) security chip. In this way, the PC actually becomes the smart card that must receive the user's PIN in order to be authenticated.

For information about smart card authentication, see the Windows Smart Card Technical Reference.

Remote and wireless authentication

Remote and wireless network authentication is another technology that uses certificates for authentication. The Internet Authentication Service (IAS) and virtual private network servers use Extensible Authentication Protocol-Transport Level Security (EAP-TLS), Protected Extensible Authentication Protocol (PEAP), or Internet Protocol security (IPsec) to perform certificate-based authentication for many types of network access, including virtual private network (VPN) and wireless connections.

For information about certificate-based authentication in networking, see Network access authentication and certificates.

Related content

- Windows Authentication Concepts
- Windows authentication technical overview

ⓘ **Note:** The author created this article with assistance from AI. Learn more

Group Policy Settings Used in Windows Authentication

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This reference topic for the IT professional describes the use and impact of Group Policy settings in the authentication process.

You can manage authentication in Windows operating systems by adding user, computer, and service accounts to groups, and then by applying authentication policies to those groups. These policies are defined as local security policies and as administrative templates, also known as Group Policy settings. Both sets can be configured and distributed throughout your organization by using Group Policy.

> **Note**
>
> Features introduced in Windows Server 2012 R2 , let you configure authentication policies for targeted services or applications, commonly called authentication silos, by using protected accounts. For information about how to do this in Active Directory, see How to Configure Protected Accounts.

For example, you can apply the following policies to groups, based on their function in the organization:

- Log on locally or to a domain
- Log on over a network
- Reset accounts
- Create accounts

The following table lists policy groups relevant to authentication and provides links to documentation that can help you configure those policies.

Expand table

Policy group	Location	Description
Password Policy	Local Computer Policy\Computer Configuration\Windows	Password policies affect the characteristics and behavior of

Policy group	Location	Description
	Settings\Security Settings\Account Policies	passwords. Password policies are used for domain accounts or local user accounts. They determine settings for passwords, such as enforcement and lifetime. For information about specific settings, see Password Policy.
Account Lockout Policy	Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Account Policies	Account lockout policy options disable accounts after a set number of failed logon attempts. Using these options can help you detect and block attempts to break passwords. For information about account lockout policy options, see Account Lockout Policy.
Kerberos Policy	Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Account Policies	Kerberos-related settings include ticket lifetime and enforcement rules. Kerberos policy does not apply to local account databases because the Kerberos authentication protocol is not used to authenticate local accounts. Therefore, the Kerberos policy settings can be configured only by means of the default domain Group Policy Object (GPO), where it affects domain logons. For information about Kerberos Policy options for the domain controller, see Kerberos Policy.
Audit Policy	Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Audit Policy	Auditing policy lets you control and understand access to objects, such as files and folders, and to manage user and group accounts and user logons and logoffs. Auditing policies can specify the categories of events that you want to audit, set the size and behavior of the security log, and determine of which objects you want to monitor access and what type of access you want to monitor.
User Rights Assignment	Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\User Rights Assignment	User rights are typically assigned on the basis of the security groups to which a user belongs, such as Administrators, Power Users, or Users. The policy settings in this category are typically

Policy group	Location	Description
		used to grant or deny permission to access a computer based on the method of access and security group memberships.
Security Options	Local Computer Policy\Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options	Policies relevant to authentication include: - Devices - Domain controller - Domain member - Interactive logon - Microsoft network server - Network access - Network security - Recovery console - Shutdown
Credentials Delegation	Computer Configuration\Administrative Templates\System\Credentials Delegation	The delegation of credentials is a mechanism that lets local credentials be used on other systems, most notably member servers and domain controllers within a domain. These settings apply to applications by using the Credential Security Support Provider (Cred SSP). Remote Desktop Connection is an example.
KDC	Computer Configuration\Administrative Templates\System\KDC	These policy settings affect how the Key Distribution Center (KDC), which is a service on the domain controller, handles Kerberos authentication requests.
Kerberos	Computer Configuration\Administrative Templates\System\Kerberos	These policy settings affect how Kerberos is configured to handle support for claims, Kerberos armoring, compound authentication, identifying proxy servers, and other configurations.
Logon	Computer Configuration\Administrative Templates\System\Logon	These policy settings control how the system presents the logon experience for users.
Net Logon	Computer Configuration\Administrative Templates\System\Net Logon	These policy settings control how the system handles network logon requests including how the Domain Controller Locator behaves.

Policy group	Location	Description
		For more information about how the Domain Controller Locator fits into replication processes, see Understanding Replication Between Sites.
Biometrics	Computer Configuration\Administrative Templates\Windows Components\Biometrics	These policy settings generally permit or deny the use of Biometrics as an authentication method. For information about the Windows implementation of biometrics, see Windows Biometric Framework Overview.
Credential User Interface	Computer Configuration\Administrative Templates\Windows Components\Credential User Interface	These policy settings control how credentials are managed at the point of entry.
Password Synchronization	Computer Configuration\Administrative Templates\Windows Components\Password Synchronization	These policy settings determine how the system manages the synchronization of passwords between Windows and UNIX-based operating systems. For more information, see Password Synchronization.
Smart Card	Computer Configuration\Administrative Templates\Windows Components\Smart Card	These policy settings control how the system manages smart card logons.
Windows Logon Options	Computer Configuration\Administrative Templates\Windows Components\Windows Logon Options	These policy settings control when and how logon opportunities are available.
Ctrl+Alt+Del Options	Computer Configuration\Administrative Templates\Windows Components\Ctrl+Alt+Del Options	These policy settings affect the appearance of and accessibility to features on the logon UI (Secure Desktop), such as Task Manager and the keyboard lock of the computer.
Logon	Computer Configuration\Administrative Templates\Windows Components\Logon	These policy settings determine if or which processes can run when the user logs on.

Additional References

Windows Authentication Technical Overview

Feedback

Was this page helpful? Yes No

Credentials Protection and Management

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional discusses features and methods introduced in Windows Server 2012 R2 and Windows 8.1 for credential protection and domain authentication controls to reduce credential theft.

Restricted Admin mode for Remote Desktop Connection

Restricted Admin mode provides a method of interactively logging on to a remote host server without transmitting your credentials to the server. This prevents your credentials from being harvested during the initial connection process if the server has been compromised.

Using this mode with administrator credentials, the remote desktop client attempts to interactively logon to a host that also supports this mode without sending credentials. When the host verifies that the user account connecting to it has administrator rights and supports Restricted Admin mode, the connection succeeds. Otherwise, the connection attempt fails. Restricted Admin mode does not at any point send plain text or other re-usable forms of credentials to remote computers.

LSA protection

The Local Security Authority (LSA), which resides within the Local Security Authority Security Service (LSASS) process, validates users for local and remote sign-ins and enforces local security policies. The Windows 8.1 operating system provides additional protection for the LSA to prevent code injection by non-protected processes. This provides added security for the credentials that the LSA stores and manages. This protected process setting for LSA can be configured in Windows 8.1 but is on by default in Windows RT 8.1 and cannot be changed.

For information about configuring LSA protection, see Configuring Additional LSA Protection.

Protected Users security group

This new domain global group triggers new non-configurable protection on devices and host computers running Windows Server 2012 R2 and Windows 8.1. The Protected Users

group enables additional protections for domain controllers and domains in Windows Server 2012 R2 domains. This greatly reduces the types of credentials available when users are signed in to computers on the network from a non-compromised computer.

Members of the Protected Users group are limited further by the following methods of authentication:

- A member of the Protected Users group can only sign on using the Kerberos protocol. The account cannot authenticate using NTLM, Digest Authentication, or CredSSP. On a device running Windows 8.1, passwords are not cached, so the device that uses any one of these Security Support Providers (SSPs) will fail to authenticate to a domain when the account is a member of the Protected User group.
- The Kerberos protocol will not use the weaker DES or RC4 encryption types in the preauthentication process. This means that the domain must be configured to support at least the AES cypher suite.
- The user's account cannot be delegated with Kerberos constrained or unconstrained delegation. This means that former connections to other systems may fail if the user is a member of the Protected Users group.
- The default Kerberos Ticket Granting Tickets (TGTs) lifetime setting of four hours is configurable using Authentication Policies and Silos accessed through the Active Directory Administrative Center (ADAC). This means that when four hours has passed, the user must authenticate again.

> **⚠ Warning**
>
> Accounts for services and computers should not be members of the Protected Users group. This group provides no local protection because the password or certificate is always available on the host. Authentication will fail with the error "the user name or password is incorrect" for any service or computer that is added to the Protected Users group.

For more information about this group, see Protected Users Security Group.

Authentication Policy and Authentication Policy Silos

Forest-based Active Directory policies are introduced, and they can be applied to accounts in a domain with a Windows Server 2012 R2 domain functional level. These authentication policies can control which hosts a user can use to sign in. They work in

conjunction with the Protect Users security group, and admins can apply access control conditions for authentication to the accounts. These authentication policies isolate related accounts to constrain the scope of a network.

The new Active Directory object class, Authentication Policy, allows you to apply authentication configuration to account classes in domains with a Windows Server 2012 R2 domain functional level. Authentication policies are enforced during the Kerberos AS or the TGS exchange. Active Directory account classes are:

- User
- Computer
- Managed Service Account
- Group Managed Service Account

For more information, see Authentication Policies and Authentication Policy Silos.

For more information how to configure protected accounts, see How to Configure Protected Accounts.

Additional References

For more information about the LSA and the LSASS, see the Windows Logon and Authentication Technical Overview.

Feedback

Was this page helpful? Yes No

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**, Credential Guard is enabled by default on domain-joined, non-DC systems that meet hardware requirements. This default enablement automatically turns on VBS, and Secure Boot is not a hard prerequisite for VBS to enter a running state, although it is recommended for stronger security guarantees.

The default enablement is **without UEFI Lock**, thus allowing administrators to disable Credential Guard remotely if needed.

> ⓘ 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 04/27/2026

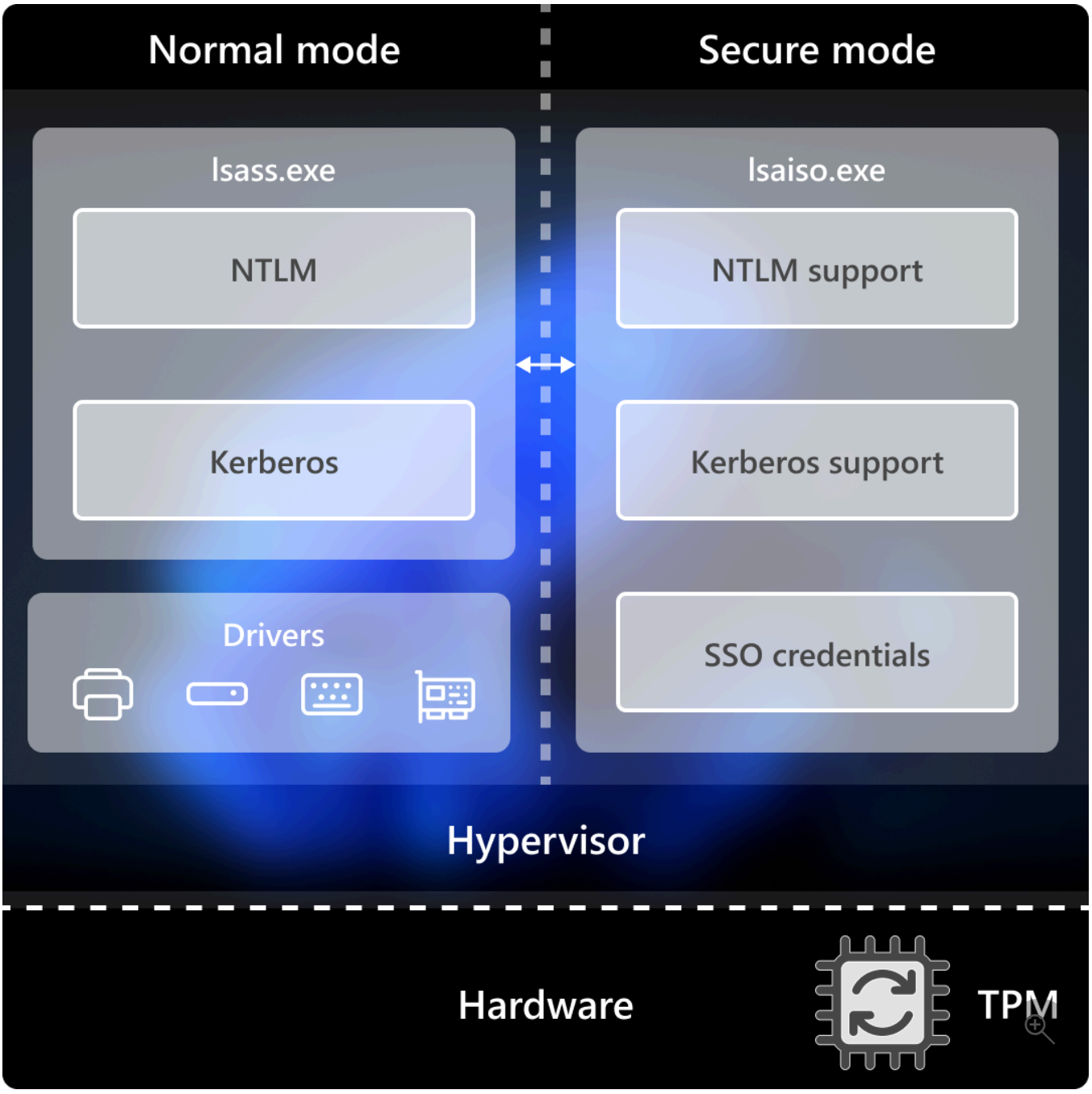
How Credential Guard works

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 by 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 04/23/2026

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

Protected Users security group

06/27/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Protected Users is a global security group for Active Directory that's designed to protect against credential theft attacks. The group triggers nonconfigurable protection on devices and host computers to prevent credentials from being cached when group members sign in.

Prerequisites

Your system must meet the following prerequisites before you can deploy a Protected Users group:

- Hosts must be running one of the following operating systems:
  - Windows 10 or Windows 11
  - Windows Server 2012 R2 or later with the most recent security updates installed
- The domain functional level must be Windows Server 2012 R2 or later. For more information about functional levels, see Forest and domain functional levels.

> **Note**
>
> The built-in domain Administrator, `S-1-5-<domain>-500`, is always exempt from authentication policies, even when they're assigned to an authentication policy silo. For more information, see How to Configure Protected Accounts.

- Protected Users global security group memberships restrict members to only use Advanced Encryption Standard (AES) for Kerberos. Members of the Protected Users group must be able to authenticate by using AES.

Protections applied by Active Directory

Becoming a member of the Protected Users group means Active Directory automatically applies certain preconfigured controls that the users won't be able to change unless they stop being group members.

Device protections for signed in Protected Users

When the signed in user is a member of the Protected Users group, the group provides the following protections:

- Credential delegation (CredSSP) doesn't cache the user's plain text credentials even when the user enables the **Allow delegating default credentials** Group Policy setting.
- Windows Digest doesn't cache the user's plaintext credentials even when they've enabled Windows Digest.
- NTLM stops caching the user's plaintext credentials or NT one-way function (NTOWF).
- Kerberos stops creating Data Encryption Standard (DES) or RC4 keys. Kerberos also doesn't cache the user's plaintext credentials, or long-term keys after acquiring the initial Ticket Granting Ticket (TGT).
- The system doesn't create a cached verifier at user sign-in or unlock, so member systems no longer support offline sign-in.

After you add a new user account to the Protected Users group, these protections activate when the new Protected User signs in to their device.

Domain controller protections for Protected Users

Protected User accounts that authenticate to a domain running Windows Server are unable to do the following:

- Authenticate with NTLM authentication.
- Use DES or RC4 encryption types in Kerberos preauthentication.
- Delegate with unconstrained or constrained delegation.
- Renew Kerberos TGTs beyond their initial four-hour lifetime.

The Protected Users group applies nonconfigurable settings to TGT expiration for every member account. Normally, the domain controller sets the TGT lifetime and renewal based on the following two domain policies:

- Maximum lifetime for user ticket
- Maximum lifetime for user ticket renewal

For Protected Users members, the group automatically sets these lifetime limits to 240 minutes. The user can't change this limit unless they leave the group.

How the Protected Users group works

You can add users to the Protected Users group by using the following methods:

- UI tools, such as Active Directory Administrative Center or Active Directory Users and Computers.
- PowerShell, by using the Add-ADGroupMember cmdlet.

> **Important**
>
> - Never add accounts for services and computers to the Protected Users group. For those accounts, membership doesn't provide local protections because the password and certificate is always available on the host.
> - Don't add accounts that are already members of highly privileged groups, such as the Enterprise Admins or Domain Admins groups, until you can guarantee that adding them won't have negative consequences. Highly privileged users in Protected Users are subject to the same limitations and restrictions as regular users, and it's not possible to work around or change those settings. If you add all members of those groups to the Protected Users group, it's possible to accidentally lock out their accounts. It's important to test your system to make sure the mandatory setting changes won't interfere with account access for these privileged user groups.

Members of the Protected Users group can only authenticate using Kerberos with AES. This method requires AES keys for the account in Active Directory. The built-in Administrator doesn't have an AES key unless the password for the domain running Windows Server 2008 or later changes. Any account that has its password changed by a domain controller running an earlier version of Windows Server is locked out of authentication.

To avoid lockouts and missing AES keys, we recommend that you follow these guidelines:

- Don't run tests in domains unless all domain controllers run Windows Server 2008 or later.
- If you have migrated accounts from other domains, you need to reset the password so the accounts have AES hashes. Otherwise, these accounts become unable to authenticate.
- Users need to change passwords after switching to a domain functional level of Windows Server 2008 or later. Doing so ensures that they have AES password hashes after they become members of the Protected Users group.

Protected Users group Active Directory properties

The following table specifies the Active Directory properties of the Protected Users group.

Expand table

Attribute	Value
Well-known SID/RID	S-1-5-21-<domain>-525
Type	Domain Global
Default container	CN=Users, DC=<domain>, DC=
Default members	None
Default member of	None
Protected by AdminSDHolder?	No
Safe to move out of default container?	Yes
Safe to delegate management of this group to non-service admins?	No
Default user rights	No default user rights

Event logs

Two operational administrative logs are available to help troubleshoot events that are related to Protected Users. These new logs are located in Event Viewer and are disabled by default. They're located under **Applications and Services Logs\Microsoft\Windows\Authentication**.

To enable capturing these logs:

1. Right-click **Start** and then select **Event Viewer**.
2. Open **Applications and Services Logs\Microsoft\Windows\Authentication**.
3. For each log that you want to enable, right-click the log name and then select **Enable Log**.

Expand table

Event ID and log	Description
104 **ProtectedUser-Client**	Reason: The security package on the client doesn't contain the credentials. The error is logged in the client computer when the account is a member of the Protected Users security group. This event indicates that the security package doesn't cache the credentials that are needed to authenticate to the server.

Event ID and log	Description
	Displays the package name, user name, domain name, and server name.
304 **ProtectedUser-Client**	Reason: The security package doesn't store the protected user's credentials. An informational event is logged in the client to indicate that the security package doesn't cache the user's sign-in credentials. It's expected that Digest (WDigest), Credential Delegation (CredSSP), and NTLM fail to have sign-on credentials for Protected Users members. Applications can still succeed if they prompt for credentials. Displays the package name, user name, and domain name.
100 **ProtectedUserFailures-DomainController**	Reason: An NTLM sign-in failure occurs for an account in the Protected Users security group. An error is logged in the domain controller to indicate that NTLM authentication failed because the account was a member of the Protected Users security group. Displays the account name and device name.
104 **ProtectedUserFailures-DomainController**	Reason: DES or RC4 encryption types are used for Kerberos authentication and a sign-in failure occurs for a user in the Protected Users security group. Kerberos preauthentication failed because DES and RC4 encryption types can't be used when the account is a member of the Protected Users security group. (AES is acceptable.)
303 **ProtectedUserSuccesses-DomainController**	Reason: A Kerberos TGT was successfully issued for a member of the Protected Users group.

Related content

- Credentials Protection and Management
- Authentication Policies and Authentication Policy Silos
- How to Configure Protected Accounts

Authentication Policies and Authentication Policy Silos

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

This topic for the IT professional describes authentication policy silos and the policies that can restrict accounts to those silos. It also explains how authentication policies can be used to restrict the scope of accounts.

Authentication policy silos and the accompanying policies provide a way to contain high-privilege credentials to systems that are only pertinent to selected users, computers, or services. Silos can be defined and managed in Active Directory Domain Services (AD DS) by using the Active Directory Administrative Center and the Active Directory Windows PowerShell cmdlets.

Authentication policy silos are containers to which administrators can assign user accounts, computer accounts, and service accounts. Sets of accounts can then be managed by the authentication policies that have been applied to that container. This reduces the need for the administrator to track access to resources for individual accounts, and helps prevent malicious users from accessing other resources through credential theft.

Capabilities introduced in Windows Server 2012 R2 , allow you to create authentication policy silos, which host a set of high-privilege users. You can then assign authentication policies for this container to limit where privileged accounts can be used in the domain. When accounts are in the Protected Users security group, additional controls are applied, such as the exclusive use of the Kerberos protocol.

With these capabilities, you can limit high-value account usage to high-value hosts. For example, you could create a new Forest Administrators silo that contains enterprise, schema, and domain administrators. Then you could configure the silo with an authentication policy so that password and smartcard-based authentication from systems other than domain controllers and domain administrator consoles would fail.

For information about configuring authentication policy silos and authentication policies, see How to Configure Protected Accounts.

About authentication policy silos

An authentication policy silo controls which accounts can be restricted by the silo and defines the authentication policies to apply to the members. You can create the silo based on the requirements of your organization. The silos are Active Directory objects for users, computers, and services as defined by the schema in the following table.

Active Directory schema for authentication policy silos

Expand table

Display Name	Description
Authentication Policy Silo	An instance of this class defines authentication policies and related behaviors for assigned users, computers, and services.
Authentication Policy Silos	A container of this class can contain authentication policy silo objects.
Authentication Policy Silo Enforced	Specifies whether the authentication policy silo is enforced. When not enforced, the policy by default is in audit mode. Events that indicate potential successes and failures are generated, but protections are not applied to the system.
Assigned Authentication Policy Silo Backlink	This attribute is the back link for msDS-AssignedAuthNPolicySilo.
Authentication Policy Silo Members	Specifies which principals are assigned to the AuthNPolicySilo.
Authentication Policy Silo Members Backlink	This attribute is the back link for msDS-AuthNPolicySiloMembers.

Authentication policy silos can be configured by using the Active Directory Administrative Console or Windows PowerShell. For more information, see How to Configure Protected Accounts.

About authentication policies

An authentication policy defines the Kerberos protocol ticket-granting ticket (TGT) lifetime properties and authentication access control conditions for an account type. The policy is built on and controls the AD DS container known as the authentication policy silo.

Authentication policies control the following:

- The TGT lifetime for the account, which is set to be non-renewable.

- The criteria that device accounts need to meet to sign in with a password or a certificate.
- The criteria that users and devices need to meet to authenticate to services running as part of the account.

The Active Directory account type determines the caller's role as one of the following:

- **User**

  Users should always be members of the Protected Users security group, which by default rejects attempts to authentication using NTLM.

  Policies can be configured to set the TGT lifetime of a user account to a shorter value or restrict the devices to which a user account can sign in. Rich expressions can be configured in the authentication policy to control the criteria that the users and their devices need to meet to authenticate to the service.

  For more information see Protected Users Security Group.

- **Service**

  Standalone managed service accounts, group managed service accounts, or a custom account object that is derived from these two types of service accounts are used. Policies can set a device's access control conditions, which are used to restrict managed service account credentials to specific devices with an Active Directory identity. Services should never be members of the Protected Users security group because all incoming authentication will fail.

- **Computer**

  The computer account object or the custom account object that is derived from the computer account object is used. Policies can set the access control conditions that are required to allow authentication to the account based on user and device properties. Computers should never be members of the Protected Users security group because all incoming authentication will fail. By default, attempts to use NTLM authentication are rejected. A TGT lifetime should not be configured for computer accounts.

> ⓘ **Note**
>
> It is possible to set an authentication policy on a set of accounts without associating the policy to an authentication policy silo. You can use this strategy when you have a single account to protect.

Active Directory schema for authentication policies

The policies for the Active Directory objects for users, computers, and services are defined by the schema in the following table.

Expand table

Type	Display Name	Description
Policy	Authentication Policy	An instance of this class defines authentication policy behaviors for assigned principals.
Policy	Authentication Policies	A container of this class can contain authentication policy objects.
Policy	Authentication Policy Enforced	Specifies whether the authentication policy is enforced. When not enforced, the policy by default is in audit mode, and events that indicate potential successes and failures are generated, but protections are not applied to the system.
Policy	Assigned Authentication Policy Backlink	This attribute is the back link for msDS-AssignedAuthNPolicy.
Policy	Assigned Authentication Policy	Specifies which AuthNPolicy should be applied to this principal.
User	User Authentication Policy	Specifies which AuthNPolicy should be applied to users who are assigned to this silo object.
User	User Authentication Policy Backlink	This attribute is the back link for msDS-UserAuthNPolicy.
User	ms-DS-User-Allowed-To-Authenticate-To	This attribute is used to determine the set of principals allowed to authenticate to a service running under the user account.
User	ms-DS-User-Allowed-To-Authenticate-From	This attribute is used to determine the set of devices to which a user account has permission to sign in.
User	User TGT Lifetime	Specifies the maximum age of a Kerberos TGT that is issued to a user (expressed in seconds). Resultant TGTs are non-renewable.
Computer	Computer Authentication Policy	Specifies which AuthNPolicy should be applied to computers that are assigned to this silo object.
Computer	Computer Authentication Policy Backlink	This attribute is the back link for msDS-ComputerAuthNPolicy.

Type	Display Name	Description
Computer	ms-DS-Computer-Allowed-To-Authenticate-To	This attribute is used to determine the set of principals that are allowed to authenticate to a service running under the computer account.
Computer	Computer TGT Lifetime	Specifies the maximum age of a Kerberos TGT that is issued to a computer (expressed in seconds). It is not recommended to change this setting.
Service	Service Authentication Policy	Specifies which AuthNPolicy should be applied to services that are assigned to this silo object.
Service	Service Authentication Policy Backlink	This attribute is the back link for msDS-ServiceAuthNPolicy.
Service	ms-DS-Service-Allowed-To-Authenticate-To	This attribute is used to determine the set of principals that are allowed to authenticate to a service running under the service account.
Service	ms-DS-Service-Allowed-To-Authenticate-From	This attribute is used to determine the set of devices to which a service account has permission to sign in.
Service	Service TGT Lifetime	Specifies the maximum age of a Kerberos TGT that is issued to a service (expressed in seconds).

Authentication policies can be configured for each silo by using the Active Directory Administrative Console or Windows PowerShell. For more information, see How to Configure Protected Accounts.

How it works

This section describes how authentication policy silos and authentication policies work in conjunction with the Protected Users security group and implementation of the Kerberos protocol in Windows.

- How the Kerberos protocol is used with authentication silos and policies
- How restricting a user sign-in works
- How restricting service ticket issuance works

Protected accounts

The Protected Users security group triggers non-configurable protection on devices and host computers running Windows Server 2012 R2 and Windows 8.1, and on domain

controllers in domains with a primary domain controller running Windows Server 2012 R2 . Depending on the domain functional level of the account, members of the Protected Users security group are further protected because of changes in the authentication methods that are supported in Windows.

- The member of the Protected Users security group cannot authenticate by using NTLM, Digest Authentication, or CredSSP default credential delegation. On a device running Windows 8.1 that uses any one of these Security Support Providers (SSPs), authentication to a domain will fail when the account is a member of the Protected Users security group.
- The Kerberos protocol will not use the weaker DES or RC4 encryption types in the preauthentication process. This means that the domain must be configured to support at least the AES encryption type.
- The user's account cannot be delegated with Kerberos constrained or unconstrained delegation. This means that former connections to other systems may fail if the user is a member of the Protected Users security group.
- The default Kerberos TGTs lifetime setting of four hours is configurable by using authentication policies and silos, which can be accessed through the Active Directory Administrative Center. This means that when four hours has passed, the user must authenticate again.

For more information about this security group, see How the Protected Users group works.

Silos and authentication policies

Authentication policy silos and authentication policies leverage the existing Windows authentication infrastructure. The use of the NTLM protocol is rejected, and the Kerberos protocol with newer encryption types is used. Authentication policies complement the Protected Users security group by providing a way to apply configurable restrictions to accounts, in addition to providing restrictions for accounts for services and computers. Authentication policies are enforced during the Kerberos protocol authentication service (AS) or ticket-granting service (TGS) exchange. For more information about how Windows uses the Kerberos protocol, and what changes have been made to support authentication policy silos and authentication policies, see:

- How the Kerberos Version 5 Authentication Protocol Works
- Changes in Kerberos Authentication (Windows Server 2008 R2 and Windows 7)

How the Kerberos protocol is used with authentication policy silos and policies

When a domain account is linked to an authentication policy silo, and the user signs in, the Security Accounts Manager adds the claim type of Authentication Policy Silo that includes the silo as the value. This claim on the account provides the access to the targeted silo.

When an authentication policy is enforced and the authentication service request for a domain account is received on the domain controller, the domain controller returns a non-renewable TGT with the configured lifetime (unless the domain TGT lifetime is shorter).

> ⓘ **Note**
>
> The domain account must have a configured TGT lifetime and must be either directly linked to the policy or indirectly linked through the silo membership.

When an authentication policy is in audit mode and the authentication service request for a domain account is received on the domain controller, the domain controller checks if authentication is allowed for the device so that it can log a warning if there is a failure. An audited authentication policy does not alter the process, so authentication requests will not fail if they do not meet the requirements of the policy.

> ⓘ **Note**
>
> The domain account must be either directly linked to the policy or indirectly linked through the silo membership.

When an authentication policy is enforced and the authentication service is armored, the authentication service request for a domain account is received on the domain controller, the domain controller checks if authentication is allowed for the device. If it fails, the domain controller returns an error message and logs an event.

> ⓘ **Note**
>
> The domain account must be either directly linked to the policy or indirectly linked through the silo membership.

When an authentication policy is in audit mode and a ticket-granting service request is received by the domain controller for a domain account, the domain controller checks if authentication is allowed based on the request's ticket Privilege Attribute Certificate (PAC) data, and it logs a warning message if it fails. The PAC contains various types of authorization data, including groups that the user is a member of, rights the user has, and what policies apply to the user. This information is used to generate the user's access token. If it is an enforced authentication policy which allows authentication to a user, device, or service, the domain controller checks if authentication is allowed based on the request's ticket PAC data. If it fails, the domain controller returns an error message and logs an event.

> **Note**
>
> The domain account must be either directly linked or linked through silo membership to an audited authentication policy which allows authentication to a user, device or service,

You can use a single authentication policy for all members of a silo, or you can use separate policies for users, computers, and managed service accounts.

Authentication policies can be configured for each silo by using the Active Directory Administrative Console or Windows PowerShell. For more information, see How to Configure Protected Accounts.

How restricting a user sign-in works

Because these authentication policies are applied to an account, it also applies to accounts that are used by services. If you want to limit the usage of a password for a service to specific hosts, this setting is useful. For example, group managed service accounts are configured where the hosts are allowed to retrieve the password from Active Directory Domain Services. However, that password can be used from any host for initial authentication. By applying an access control condition, an additional layer of protection can be achieved by limiting the password to only the set of hosts that can retrieve the password.

When services that run as system, network service, or other local service identity connect to network services, they use the host's computer account. Computer accounts cannot be restricted. So even if the service is using a computer account that is not for a Windows host, it cannot be restricted.

Restricting user sign-in to specific hosts requires the domain controller to validate the host's identity. When using Kerberos authentication with Kerberos armoring (which is part of Dynamic Access Control), the Key Distribution Center is provided with the TGT of the host from which the user is authenticating. The content of this armored TGT is used to complete an access check to determine if the host is allowed.

When a user signs in to Windows or enters their domain credentials in a credential prompt for an application, by default, Windows sends an unarmored AS-REQ to the domain controller. If the user is sending the request from a computer that does not support armoring, such as computers running Windows 7 or Windows Vista, the request fails.

The following list describes the process:

- The domain controller in a domain running Windows Server 2012 R2 queries for the user account and determines if it is configured with an authentication policy that restricts initial authentication that requires armored requests.
- The domain controller will fail the request.
- Because armoring is required, the user can attempt to sign in by using a computer running Windows 8.1 or Windows 8, which is enabled to support Kerberos armoring to retry the sign-in process.
- Windows detects that the domain supports Kerberos armoring and sends an armored AS-REQ to retry the sign-in request.
- The domain controller performs an access check by using the configured access control conditions and the client operating system's identity information in the TGT that was used to armor the request.
- If the access check fails, the domain controller rejects the request.

Even when operating systems support Kerberos armoring, access control requirements can be applied and must be met before access is granted. Users sign in to Windows or enter their domain credentials in a credential prompt for an application. By default, Windows sends an unarmored AS-REQ to the domain controller. If the user is sending the request from a computer that supports armoring, such as Windows 8.1 or Windows 8, authentication policies are evaluated as follows:

1. The domain controller in a domain running Windows Server 2012 R2 queries for the user account and determines if it is configured with an authentication policy that restricts initial authentication that requires armored requests.

2. The domain controller performs an access check by using the configured access control conditions and the system's identity information in the TGT that is used to armor the request. The access check succeeds.

   > ⓘ **Note**
   >
   > If legacy workgroup restrictions are configured, those also need to be met.

3. The domain controller replies with an armored reply (AS-REP), and the authentication continues.

How restricting service ticket issuance works

When an account is not allowed and a user who has a TGT attempts to connect to the service (such as by opening an application that requires authentication to a service that is identified by the service's service principal name (SPN), the following sequence occurs:

1. In an attempt to connect to SPN1 from SPN, Windows sends a TGS-REQ to the domain controller that is requesting a service ticket to SPN1.
2. The domain controller in a domain running Windows Server 2012 R2 looks up SPN1 to find the Active Directory Domain Services account for the service and determines that the account is configured with an authentication policy that restricts service ticket issuance.
3. The domain controller performs an access check by using the configured access control conditions and the user's identity information in the TGT. The access check fails.
4. The domain controller rejects the request.

When an account is allowed because the account meets the access control conditions that are set by the authentication policy, and a user who has a TGT attempts to connect to the service (such as by opening an application that requires authentication to a service that is identified by the service's SPN), the following sequence occurs:

1. In an attempt to connect to SPN1, Windows sends a TGS-REQ to the domain controller that is requesting a service ticket to SPN1.
2. The domain controller in a domain running Windows Server 2012 R2 looks up SPN1 to find the Active Directory Domain Services account for the service and determines that the account is configured with an authentication policy that restricts service ticket issuance.

3. The domain controller performs an access check by using the configured access control conditions and the user's identity information in the TGT. The access check succeeds.

4. The domain controller replies to the request with a ticket-granting service reply (TGS-REP).

Associated error and informational event messages

The following table describes the events that are associated with Protected Users security group and the authentication policies that are applied to authentication policy silos.

The events are recorded in the Applications and Services Logs at **Microsoft\Windows\Authentication**.

For troubleshooting steps that use these events, see Troubleshoot Authentication Policies and Troubleshoot events related to Protected Users.

Expand table

Event ID and Log	Description
101 **AuthenticationPolicyFailures-DomainController**	Reason: An NTLM sign-in failure occurs because the authentication policy is configured. An event is logged in the domain controller to indicate that NTLM authentication failed because access control restrictions are required, and those restrictions cannot be applied to NTLM. Displays the account, device, policy, and silo names.
105 **AuthenticationPolicyFailures-DomainController**	Reason: A Kerberos restriction failure occurs because the authentication from a particular device was not permitted. An event is logged in the domain controller to indicate that a Kerberos TGT was denied because the device did not meet the enforced access control restrictions. Displays the account, device, policy, silo names, and TGT lifetime.
305 **AuthenticationPolicyFailures-DomainController**	Reason: A potential Kerberos restriction failure might occur because the authentication from a particular device was not permitted.

Event ID and Log	Description
	In audit mode, an informational event is logged in the domain controller to determine if a Kerberos TGT will be denied because the device did not meet the access control restrictions. Displays the account, device, policy, silo names, and TGT lifetime.
106 **AuthenticationPolicyFailures-DomainController**	Reason: A Kerberos restriction failure occurs because the user or device was not allowed to authenticate to the server. An event is logged in the domain controller to indicate that a Kerberos service ticket was denied because the user, device, or both do not meet the enforced access control restrictions. Displays the device, policy, and silo names.
306 **AuthenticationPolicyFailures-DomainController**	Reason: A Kerberos restriction failure might occur because the user or device was not allowed to authenticate to the server. In audit mode, an informational event is logged on the domain controller to indicate that a Kerberos service ticket will be denied because the user, device, or both do not meet the access control restrictions. Displays the device, policy, and silo names.

Additional References

How to Configure Protected Accounts

Credentials Protection and Management

Protected Users Security Group

Feedback

Was this page helpful? Yes No

Kerberos authentication overview in Windows Server

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Kerberos is an authentication protocol that is used to verify the identity of a user or host. This topic contains information about Kerberos authentication in Windows Server and Windows.

The Windows Server operating systems implement the Kerberos version 5 authentication protocol and extensions for public key authentication, transporting authorization data, and delegation. The Kerberos authentication client is implemented as a security support provider (SSP), and it can be accessed through the Security Support Provider Interface (SSPI). Initial user authentication is integrated with the Winlogon single sign-on architecture.

The Kerberos Key Distribution Center (KDC) is integrated with other Windows Server security services that run on the domain controller. The KDC uses the domain's Active Directory Domain Services database as its security account database. Active Directory Domain Services is required for default Kerberos implementations within the domain or forest.

How Kerberos authentication works

Kerberos authentication involves three parties: the client that acts for a security principal, the service the principal wants to reach, and the Key Distribution Center (KDC) that both parties trust. A *security principal* is an identity, such as a user, computer, or service, that the system can authenticate. In Windows Server, the KDC runs on every domain controller and uses the domain's Active Directory Domain Services database as its account database.

When a user signs in, the Kerberos client contacts the KDC to prove the user's identity and obtain a ticket-granting ticket (TGT). The TGT is a credential that lets the client request access to services without presenting the user's password again. The client stores the TGT in its ticket cache and reuses it until the ticket expires.

To connect to a service, the client presents its TGT to the KDC and requests a service ticket for that specific service. The KDC returns a service ticket, which the client presents directly to the target service. Because the ticket already contains the information that the service needs to verify

the client's identity, the service can authenticate the client without contacting a domain controller for each connection.

Windows Server implements the industry-standard Kerberos version 5 protocol that the Internet Engineering Task Force (IETF) defines. This standards-track foundation lets the Windows implementation interoperate with other systems that use Kerberos. On top of the standard, Windows adds extensions for public key authentication, transporting authorization data, and delegation. Microsoft documents these Windows-specific behaviors in the Windows Protocols specification [MS-KILE]: Kerberos Protocol Extensions, which describes where the Windows implementation extends or differs from the standard.

Practical applications

The benefits gained by using Kerberos for domain-based authentication are described in the following sections.

Delegated authentication

Services that run on Windows operating systems can impersonate a client computer when accessing resources on the client's behalf. In many cases, a service can complete its work for the client by accessing resources on the local computer. When a client computer authenticates to the service, NTLM and Kerberos protocol provide the authorization information that a service needs to impersonate the client computer locally. However, some distributed applications are designed so that a front-end service must use the client computer's identity when it connects to back-end services on other computers. Kerberos authentication supports a delegation mechanism that enables a service to act on behalf of its client when connecting to other services.

Single sign on

Using Kerberos authentication within a domain or in a forest allows the user or service access to resources permitted by administrators without multiple requests for credentials. After initial domain sign on through Winlogon, Kerberos manages the credentials throughout the forest whenever access to resources is attempted.

Interoperability

The implementation of the Kerberos V5 protocol by Microsoft is based on standards-track specifications that are recommended to the Internet Engineering Task Force (IETF). As a result, in Windows operating systems, the Kerberos protocol lays a foundation for interoperability with

other networks in which the Kerberos protocol is used for authentication. In addition, Microsoft publishes Windows Protocols documentation for implementing the Kerberos protocol. The documentation contains the technical requirements, limitations, dependencies, and Windows-specific protocol behavior for Microsoft's implementation of the Kerberos protocol.

More efficient authentication to servers

Before Kerberos, NTLM authentication could be used, which requires an application server to connect to a domain controller to authenticate every client computer or service. With the Kerberos protocol, renewable session tickets replace pass-through authentication. The server is not required to go to a domain controller, unless it needs to validate a Privilege Attribute Certificate (PAC). Instead, the server can authenticate the client computer by examining credentials presented by the client. Client computers can obtain credentials for a particular server once and then reuse those credentials throughout a network logon session.

Mutual authentication

By using the Kerberos protocol, a party at either end of a network connection can verify that the party on the other end is the entity it claims to be. NTLM does not enable clients to verify a server's identity or enable one server to verify the identity of another. NTLM authentication was designed for a network environment in which servers were assumed to be genuine. The Kerberos protocol makes no such assumption.

Encryption types

Beginning with Windows Server 2025, Kerberos no longer honors the legacy registry key `REG_DWORD SupportedEncryptionTypes` found in the path **HKEY_LOCAL_MACHINE\CurrentControlSet\Control\Lsa\Kerberos\Parameters**. Microsoft recommends using group policy instead. To learn more about the group policy settings, see Network security: Configure encryption types allowed for Kerberos.

Related content

- Windows Authentication Overview

ⓘ**Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 09/15/2026

Domain-joined Device Public Key Authentication

Article • 05/12/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Kerberos added support for domain-joined devices to sign-in using a certificate beginning with Windows Server 2012 and Windows 8. This change allows 3rd party vendors to create solutions to provision and initialize certificates for domain-joined devices to use for domain authentication.

Automatic public key provisioning

Beginning with Windows 10 version 1507 and Windows Server 2016, domain-joined devices automatically provision a bound public key to a Windows Server 2016 domain controller (DC). Once a key is provisioned, then Windows can use public key authentication to the domain.

Key generation

If the device is running Credential Guard, then a public/private key pair is created protected by Credential Guard.

If Credential Guard is not available and a TPM is, then a public/private key pair is created protected by the TPM.

If neither is available, then a key pair is not generated and the device can only authenticate using password.

Provisioning computer account public key

When Windows starts up, it checks if a public key is provisioned for its computer account. If not, then it generates a bound public key and configures it for its account in AD using a Windows Server 2016 or higher DC. If all the DCs are down-level, then no key is provisioned.

Configuring device to only use public key

If the Group Policy setting **Support for device authentication using certificate** is set to **Force**, then the device needs to find a DC that runs Windows Server 2016 or later to authenticate. The setting is under Administrative Templates > System > Kerberos.

Configuring device to only use password

If the Group Policy setting **Support for device authentication using certificate** is disabled, then password is always used. The setting is under Administrative Templates > System > Kerberos.

Domain-joined device authentication using public key

When Windows has a certificate for the domain-joined device, Kerberos first authenticates using the certificate and on failure retries with password. This allows the device to authenticate to down-level DCs.

Since the automatically provisioned public keys have a self-signed certificate, certificate validation fails on domain controllers that do not support Key Trust account mapping. By default, Windows retries authentication using the device's domain password.

Kerberos Constrained Delegation Overview

08/25/2025

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article describes new capabilities for Kerberos constrained delegation in Windows Server.

Feature description

Kerberos constrained delegation was introduced in Windows Server to provide a safer form of delegation that could be used by services. When it's configured, constrained delegation restricts the services to which the specified server can act on the behalf of a user. This requires domain admin privileges to configure a domain account for a service and restricts the account to a single domain. In modern enterprises, front-end services aren't designed to integrate only with services in their domain.

In earlier operating systems, domain admins configured the service, and service admins had no way to know which front-end services delegated to the resource services they owned. And any front-end service that could delegate to a resource service represented a potential attack point. If a server that hosted a front-end service was compromised, and it was configured to delegate to resource services, the resource services could also be compromised.

In Windows Server 2012 R2 and Windows Server 2012, the ability to configure constrained delegation for the service is transferred from the domain admin to the service admin. In this way, the back-end service administrator can allow or deny front-end services.

The Windows Server 2012 R2 and Windows Server 2012 implementation of the Kerberos protocol includes extensions specifically for constrained delegation. Service for User to Proxy (S4U2Proxy) allows a service to use its Kerberos service ticket for a user to obtain a service ticket from the Key Distribution Center (KDC) to a back-end service. These extensions allow constrained delegation to be configured on the back-end service's account, which can be in another domain. For more information about these extensions, see [MS-SFU]: Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification in the MSDN Library.

Practical applications

Constrained delegation lets service admins specify and enforce application trust boundaries by limiting where application services can act on a user's behalf. Service administrators can configure which front-end service accounts can delegate to their back-end services.

Front-end services like Microsoft Internet Security and Acceleration (ISA) Server, Microsoft Forefront Threat Management Gateway, Microsoft Exchange Outlook Web Access (OWA), and Microsoft SharePoint Server can use constrained delegation to authenticate to servers in other domains. This provides support for across domains service solutions by using an existing Kerberos infrastructure. Kerberos constrained delegation can be managed by domain administrators or service administrators.

Resource-based constrained delegation across domains

Kerberos constrained delegation lets you provide constrained delegation when the front-end service and the resource services aren't in the same domain. Service administrators can configure the new delegation by specifying the domain accounts of the front-end services that can impersonate users on the account objects of the resource services.

What value does this change add?

Services can use constrained delegation to authenticate to servers in other domains instead of using unconstrained delegation. This provides authentication support for cross-domain service solutions using an existing Kerberos infrastructure without requiring trust in front-end services to delegate to any service.

This also shifts the decision of whether a server should trust the source of a delegated identity from the delegating-from domain administrator to the resource owner.

What works differently?

A change in the underlying protocol allows constrained delegation across domains. The Windows Server 2012 R2 and Windows Server 2012 implementation of the Kerberos protocol includes extensions to the Service for User to Proxy (S4U2Proxy) protocol. This is a set of extensions to the Kerberos protocol that allows a service to use its Kerberos service ticket for a user to obtain a service ticket from the Key Distribution Center (KDC) to a back-end service.

For implementation information about these extensions, see [MS-SFU]: Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification in MSDN.

For more information about the basic message sequence for Kerberos delegation with a forwarded ticket-granting ticket (TGT) as compared to Service for User (S4U) extensions, see section 1.3.3 Protocol Overview in the [MS-SFU]: Kerberos Protocol Extensions: Service for User and Constrained Delegation Protocol Specification.

Security Implications of Resource-based Constrained Delegation

Resource-based constrained delegation gives control of delegation to the administrator who owns the resource being accessed. It depends on attributes of the resource service rather than the service being trusted to delegate. As a result, resource-based constrained delegation can't use the Trusted-to-Authenticate-for-Delegation bit that previously controlled protocol transition. The KDC always allows protocol transition when performing resource-based constrained delegation as though the bit were set.

Because the KDC doesn't limit protocol transition, two new well-known SIDs give this control to the resource administrator. These SIDs identify whether protocol transition has occurred, and can be used with standard access control lists to grant or limit access as needed.

Expand table

SID	Description
AUTHENTICATION_AUTHORITY_ASSERTED_IDENTITY S-1-18-1	A SID that means the client's identity is asserted by an authentication authority based on proof of possession of client credentials.
SERVICE_ASSERTED_IDENTITY S-1-18-2	A SID that means the client's identity is asserted by a service.

A backend service can use standard ACL expressions to determine how the user was authenticated.

How do you configure Resource-based Constrained Delegation?

To configure a resource service to let front-end services access resources on behalf of users, use Windows PowerShell cmdlets.

- To retrieve a list of principals, use the **Get-ADComputer**, **Get-ADServiceAccount**, and **Get-ADUser** cmdlets with the **Properties PrincipalsAllowedToDelegateToAccount** parameter.
- To configure the resource service, use the **New-ADComputer**, **New-ADServiceAccount**, **New-ADUser**, **Set-ADComputer**, **Set-ADServiceAccount**, and **Set-ADUser** cmdlets with the **PrincipalsAllowedToDelegateToAccount** parameter.

Detect and remediate RC4 usage in Kerberos

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Kerberos authentication in Active Directory (AD) environments historically relied on the RC4 encryption algorithm. RC4 is a stream cipher that's useful due to its compatibility with legacy systems. However, RC4 is now considered insecure and is being phased out.

This article explains how to detect RC4 usage in your domain, audit device and user accounts that still depend on RC4, and take steps to remediate usage in favor of stronger encryption types or manage RC4 dependencies. Use these instructions to prepare your environment for upcoming changes to Kerberos defaults and improve your organization's security posture.

At a high level, IT administrators should take the following actions to prepare for the default changes to Kerberos support for RC4:

1. **Audit current RC4 usage**: use the auditing tools and event logs described in this article to identify accounts, services, or devices still using RC4.
2. **Address legacy devices**: migrate or replace unsupported devices, especially those running Windows Server 2003 or earlier, as they lack AES-SHA1 support.
3. **Disable RC4**: where possible, explicitly disable RC4.

RC4 usage in Windows and its risks

RC4 is typically used in Windows environments when accounts or devices don't support stronger encryption types such as AES-SHA1. RC4 use can occur with legacy systems, accounts created before AES-SHA1 support was introduced, or when encryption settings aren't explicitly configured. RC4 was the default encryption type making it a common choice for compatibility with older infrastructure. However, reliance on RC4 exposes environments to security risks, and its usage is being actively phased out in favor of more secure algorithms.

In response to CVE-2022-37966, Windows Updates released on or after November 8, 2022 changed the default encryption type in Kerberos to use Advanced Encryption Standard (AES)-SHA1 instead of RC4 for accounts where an encryption type wasn't explicitly set. This change resulted in a significant reduction in RC4 usage, but RC4 is still used by default for accounts that don't support AES-SHA1. For more information, see KB5021131.

This update adds the capability to set a registry value to control the default supported encryption type in Kerberos. If you don't set this registry value, the default supported encryption types are DES, RC4, and AES session keys.

The Kerberos RC4 default encryption type can be abused to carry out an attack method known as *Kerberoasting*, which targets service tickets in Microsoft Active Directory. In Kerberoasting, attackers capture these tickets and crack their encryption offline to steal user credentials, potentially compromising the security of the entire network. Changing the encryption type of Kerberos to use AES-SHA1 by default instead of RC4 helps protect customers against this attack.

> **Important**
>
> Microsoft plans to disable RC4 use as the default assumed supported encryption type for Active Directory domain controllers by the end of the second quarter of 2026. To learn more about how to prepare for disabling RC4, see How to manage Kerberos KDC usage of RC4 for service account ticket issuance changes related to CVE-2026-20833.

Prerequisites

Before you begin, you need to meet the following prerequisites:

- You need permissions to access Security Event Logs on domain controllers, such as being a member of the **Domain Admins** group or equivalent.
- If you want to run the PowerShell commands and scripts in this article, you need to have the ActiveDirectory module installed on the device where you run the commands.

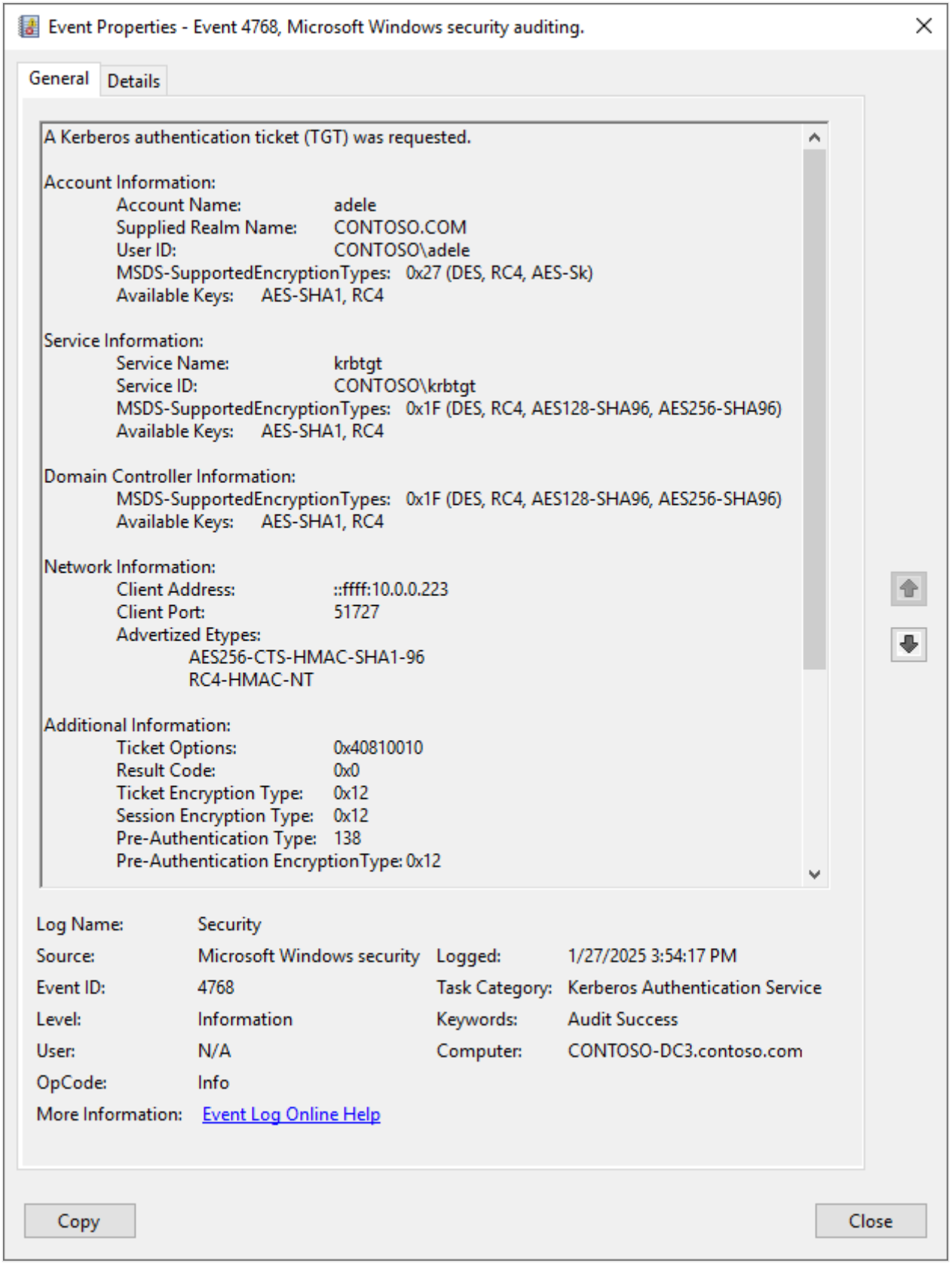
Audit RC4 usage

Detail about RC4 usage is stored in the Security Event Logs on Kerberos Key Distribution Centers (KDCs) for Windows Server 2019 and later. RC4 usage in Event Logs was also added to Windows Server 2016 in the January 2025 cumulative update. The following event IDs identify RC4 usage and accounts that are only able to use RC4:

- Event ID 4768, which is for the requested *Kerberos authentication ticket* (TGT).
- Event ID 4769, which is for the requested *Kerberos service ticket*.

Event Log fields for RC4 usage

In each entry for event IDs 4768 and 4769, there are several fields that provide insight into the encryption types supported by the account and the encryption type used for the ticket. Here's an example screenshot. The following list describes the relevant fields in both of these events.



- **Account Information** and **Service Information** sections:
  - **MSDS-SupportedEncryptionTypes**: is an Active Directory attribute that denotes the encryption types the account supports. If `msDS-SupportedEncryptionTypes` isn't defined

for the account, the KDC applies the value from `DefaultSupportedEncryptionTypes`. This field is sometimes referred to as `msds-SET`.

This value is a processed value, meaning that some accounts have extra conditions specified outside of the `msDS-SupportedEncryptionTypes` value set in the AD database. For example, on Windows Server 2022 and earlier, `msDS-SupportedEncryptionTypes` always shows Data Encryption Standard (DES) and RC4 to ensure compatibility with older domain controllers, regardless of your settings. Starting with Windows Server 2025, only AES-SHA1 and stronger algorithms are displayed. It's normal if the processed value for these security principals doesn't exactly match what's in the Active Directory attribute.

This field is only populated during account lookup. Account lookup populates this field for the account information and service information during an event with ID 4768 and lists it in the service information during an event with ID 4769. For more information, see Supported Encryption Types Bit Flags.

- **Available Keys**: Keys available for the account in AD. The password is hashed with the displayed algorithms. The population of this field is under the same restrictions as `msDS-SupportedEncryptionTypes`. RC4 is shown regardless of usage.

- **Domain Controller Information** section. This section also uses processed values and follows the same guidance described previously
  - **MSDS-SupportedEncryptionTypes**: Includes the `msDS-SupportedEncryptionTypes` for the domain controller.
  - **Account Keys**: Includes account keys for the domain controller.
- **Network Information** section:
  - **Advertized Etypes**: Enumeration of the encryption types that the client machine advertises as supported for the current operation.
- **Additional Information** section:
  - **Session Encryption Type**: Encryption type used for the Ticket Session Key.

You can use the `msDS-SupportedEncryptionTypes` or the `Advertized Etypes` fields to determine if a client or target machine doesn't support AES-SHA1 or only supports RC4.

▼ Expand this section to view a table that maps the encryption type to the algorithm that was used.

Expand table

Decimal Value	Hex Value	Supported Encryption Types
0	0x0	Not defined - defaults to RC4_HMAC_MD5
1	0x1	DES_CBC_CRC
2	0x2	DES_CBC_MD5
3	0x3	DES_CBC_CRC, DES_CBC_MD5
4	0x4	RC4
5	0x5	DES_CBC_CRC, RC4
6	0x6	DES_CBC_MD5, RC4
7	0x7	DES_CBC_CRC, DES_CBC_MD5, RC4
8	0x8	AES 128
9	0x9	DES_CBC_CRC, AES 128
10	0xA	DES_CBC_MD5, AES 128
11	0xB	DES_CBC_CRC, DES_CBC_MD5, AES 128
12	0xC	RC4, AES 128
13	0xD	DES_CBC_CRC, RC4, AES 128
14	0xE	DES_CBC_MD5, RC4, AES 128
15	0xF	DES_CBC_CRC, DES_CBC_MD5, RC4, AES 128
16	0x10	AES 256
17	0x11	DES_CBC_CRC, AES 256
18	0x12	DES_CBC_MD5, AES 256
19	0x13	DES_CBC_CRC, DES_CBC_MD5, AES 256
20	0x14	RC4, AES 256
21	0x15	DES_CBC_CRC, RC4, AES 256
22	0x16	DES_CBC_MD5, RC4, AES 256
23	0x17	DES_CBC_CRC, DES_CBC_MD5, RC4, AES 256
24	0x18	AES 128, AES 256
25	0x19	DES_CBC_CRC, AES 128, AES 256

Decimal Value	Hex Value	Supported Encryption Types
26	0x1A	DES_CBC_MD5, AES 128, AES 256
27	0x1B	DES_CBC_CRC, DES_CBC_MD5, AES 128, AES 256
28	0x1C	RC4, AES 128, AES 256
29	0x1D	DES_CBC_CRC, RC4, AES 128, AES 256
30	0x1E	DES_CBC_MD5, RC4, AES 128, AES 256
31	0x1F	DES_CBC_CRC, DES_CBC_MD5, RC4-HMAC, AES128-CTS-HMAC-SHA1-96, AES256-CTS-HMAC-SHA1-96

▼ Expand this section to view a table that maps the values in the events to the encryption type of the issued tickets.

Expand table

Type Value	Encryption Type
0x1	DES_CBC_CRC
0x3	DES_CBC_MD5
0x11	AES128-CTS-HMAC-SHA1-96
0x12	AES256-CTS-HMAC-SHA1-96
0x17	RC4-HMAC
0x18	RC4-HMAC-EXP

For a more in-depth breakdown of `msDS-SupportedEncryptionTypes` and encryption types, see our blog article Decrypting the Selection of Supported Kerberos Encryption Types.

Use PowerShell to audit RC4 usage

Manually looking through event logs to identify RC4 usage is a cumbersome task. For more efficient auditing, there are two PowerShell scripts you can download to help:

- `List-AccountKeys.ps1` to query event logs to enumerate available encryption keys for accounts.
- `Get-KerbEncryptionUsage.ps1` to identify Kerberos encryption types in use, with filtering options for specific algorithms like RC4.

Microsoft published these scripts as open source and they're available in Microsoft's Kerberos-Crypto GitHub repository. Alternatively, you can also identify RC4 usage using a security information and event management (SIEM) solution, like Microsoft Sentinel, or built-in Windows forwarding as described in our blog post So, you think you're ready for enforcing AES for Kerberos?.

1. Download the two PowerShell scripts listed from Microsoft's Kerberos-Crypto GitHub repository.

2. Open a PowerShell session as an administrator on a domain controller, or connect remotely using the Enter-PSSession cmdlet on a device you use to manage the domain controllers.

3. Run the script `List-AccountKeys.ps1` with the following command. This script queries the Security Event Log and enumerates the keys that are available for the accounts it finds.

   PowerShell

   ```
   .\List-AccountKeys.ps1
   ```

   In this example output, you can see the time at which the event happened, the account name, account type, and the account's keys. In this case, you can see that AES-SHA1 keys are available and that they would support using AES-SHA1.

   Output

   ```
   Time                  Name         Type  Keys
   ----                  ----         ----  ----
   1/21/2025 2:00:10 PM  VM01$      Machine {RC4, AES128-SHA96, AES256-SHA96,
   AES128-SHA256...}
   1/21/2025 2:00:10 PM  AdminUser     User {RC4, AES128-SHA96, AES256-SHA96,
   AES128-SHA256...}
   1/21/2025 6:50:34 PM  VM01$      Machine {RC4, AES128-SHA96, AES256-SHA96,
   AES128-SHA256...}
   1/21/2025 6:50:34 PM  AdminUser     User {RC4, AES128-SHA96, AES256-SHA96,
   AES128-SHA256...}
   1/21/2025 6:50:34 PM  VM01$      Machine {RC4, AES128-SHA96, AES256-SHA96,
   AES128-SHA256...}
   ```

4. Run the script `Get-KerbEncryptionUsage.ps1` with the following command. This script queries the same events to see the Kerberos usage that was found within the environment.

   PowerShell

   ```
   .\Get-KerbEncryptionUsage.ps1
   ```

Output

```
Time       : 1/21/2025 2:00:10 PM
Requestor  : ::1
Source     : AdminUser@CONTOSO.COM
Target     : VM01$
Type       : TGS
Ticket     : AES256-SHA96
SessionKey : AES256-SHA96

Time       : 1/21/2025 2:00:10 PM
Requestor  : 192.168.1.1
Source     : AdminUser
Target     : krbtgt
Type       : AS
Ticket     : AES256-SHA96
SessionKey : AES256-SHA96
```

This script includes filtering options for specific algorithms and can be filtered for RC4 usage, as shown in the following example:

PowerShell

```powershell
.\Get-KerbEncryptionUsage.ps1 -Encryption RC4
```

Options to handle RC4 usage

If you find that you still use RC4, there are a few things you can do depending on why RC4 was used.

A user account is using RC4 because it doesn't have AES-SHA1 keys

If a user account was created before support for AES-SHA1 keys in Windows Kerberos and the password was never reset after support was added, the account would be missing AES-SHA1 keys. AES-SHA1 support in Windows Kerberos was added in Windows Server 2003. Changing the account password generates these keys.

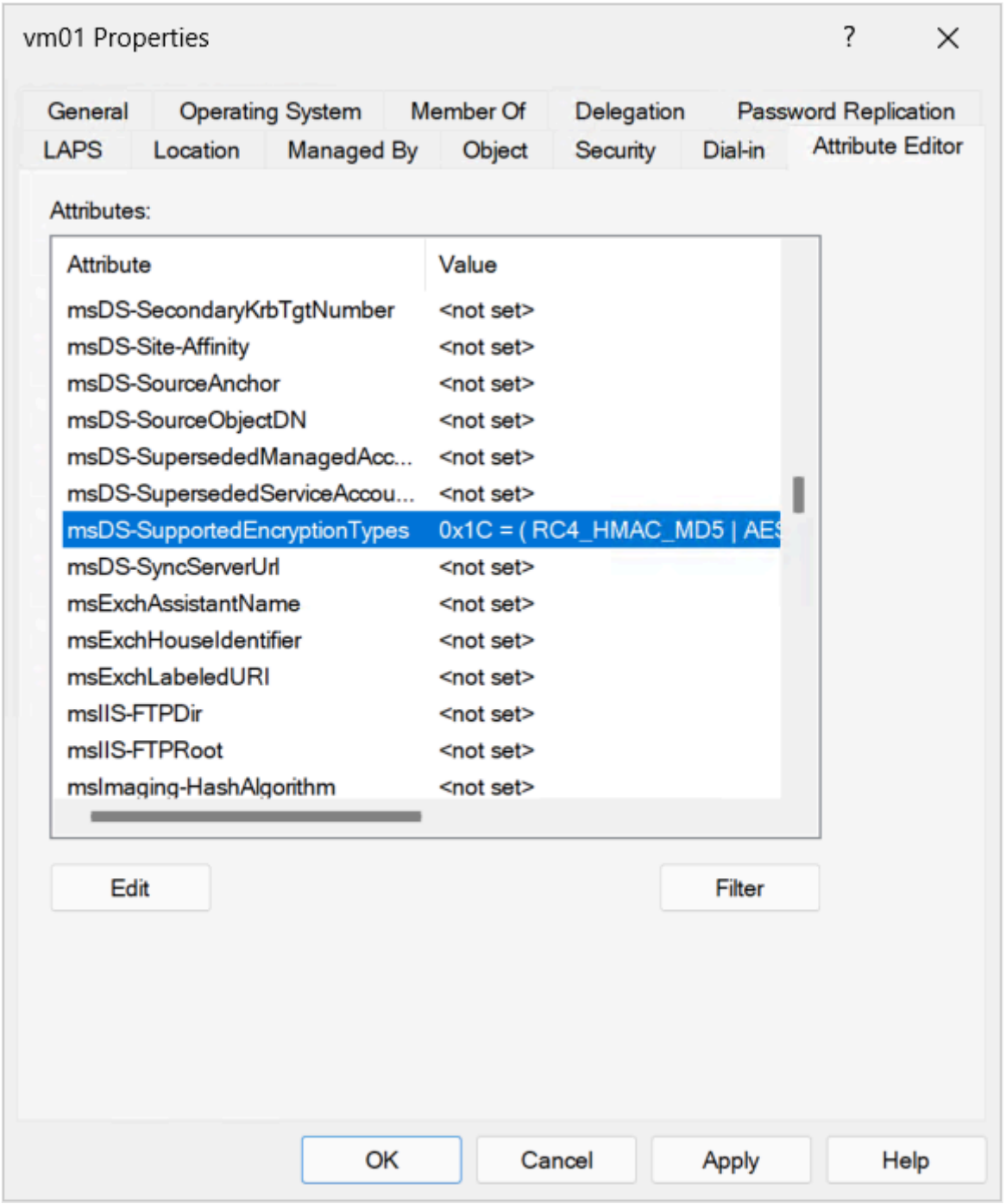
You don't need to set a value for `msDS-SupportedEncryptionTypes` for user accounts without a service principal name (SPN). The device's configuration determines the encryption type for service ticket requests and session keys.

The processed msDS-SupportedEncryptionTypes value doesn't include the AES-SHA1 bits

If the processed `msDS-SupportedEncryptionTypes` value doesn't include AES-SHA1, it could be the result of a handful of conditions.

1. Case: **the source and/or target machine's supported encryption types don't include AES-SHA1**. In this case, confirm the configuration of policy and update it so that it includes AES-SHA1. You can view the configured `msDS-SupportedEncryptionTypes` in the properties of the account in Active Directory Users and Computers console or by using PowerShell.

    - For Active Directory Users and Computers, make sure the **Advanced Features** view is enabled, then open the properties of the computer account and navigate to the **Attribute Editor** tab. Look for the `msDS-SupportedEncryptionTypes` attribute:



- For PowerShell, run the following command to retrieve the `msDS-SupportedEncryptionTypes` attribute. Be sure to replace `<computer account name>` with the actual name of the account you want to check.

  PowerShell

  ```
  $accountName = "<computer account name>"
  $parameters = @{
       Filter      = "Name -eq '$($accountName)' -and (ObjectClass -eq
  'Computer' -or ObjectClass -eq 'User')"
       Properties = "msDS-SupportedEncryptionTypes"
  }

  Get-ADObject @parameters | FL "DistinguishedName","msDS-
  SupportedEncryptionTypes","Name","ObjectClass"
  ```

  Here's an example of the output. The value for `msDS-SupportedEncryptionTypes` is in decimal instead of hexadecimal, which you can convert using the table provided in the section Event Log fields for RC4 usage.

  Output

  ```
  DistinguishedName             : CN=vm01,CN=Computers,DC=contoso,DC=com
  msDS-SupportedEncryptionTypes : 28
  Name                          : vm01
  ObjectClass                   : computer
  ```

If the `msDS-SupportedEncryptionTypes` value in AD for the source and/or target machine isn't defined, the KDC falls back to the `DefaultDomainSupportedEncTypes` value, as described in the second case.

To determine what values best fit your environment, we recommend reading Active Directory Hardening Series - Part 4 – Enforcing AES for Kerberos | Microsoft Community Hub. After finding the right combination for your environment, restart the machine and it updates its `msDS-SupportedEncryptionTypes` attributes in the AD database.

2. Case: **the `msDS-SupportedEncryptionTypes` value in AD for the source machine isn't defined and the KDC falls back to the `DefaultDomainSupportedEncTypes` value**. This case is more complex and requires a holistic understanding of your environment to resolve. The `DefaultDomainSupportedEncTypes` value is used to define the **assumed** supported encryption types for devices that don't have a value for `msDS-SupportedEncryptionTypes`. There are two approaches to resolving this situation:

   - Define the specific `msDS-SupportedEncryptionTypes` value in the account properties to ensure it isn't falling back to the `DefaultDomainSupportedEncTypes` value.

- Alternatively, set the `DefaultDomainSupportedEncTypes` value to include AES-SHA1.

The correct method is dependent on your individual risk tolerance because updating the `DefaultDomainSupportedEncTypes` value changes the behavior of **all** accounts that don't have a value.

A device doesn't support AES-SHA1

For Windows devices, the last version of Windows that didn't support AES-SHA1 was Windows Server 2003, which is out of support. You should migrate to a version of Windows that supports AES-SHA1. For accounts created before AES-SHA1 support, reset passwords to generate modern encryption keys, such as AES-SHA1. For accounts that don't support SHA-1, manually configure supported encryption types to include RC4.

> **Tip**
>
> If you have a third-party device that doesn't support AES-SHA1, reach out to stillneedrc4@microsoft.com with information about the device and scenario.

Limit or disable RC4 usage

The change in default Kerberos encryption type has resulted in a significant reduction in RC4 usage. However, if you want to further reduce RC4 usage, you can scope limited usage of RC4 by Group Policy or disable RC4 entirely in your domain.

> **Caution**
>
> Starting with Windows Server 2025, domain controllers don't issue RC4 Ticket Granting Tickets. While you can authenticate **to** legacy devices with RC4, the legacy device isn't able to authenticate using Kerberos. You need to use earlier versions of Windows Server for your domain controllers.

To limit or disable RC4 usage:

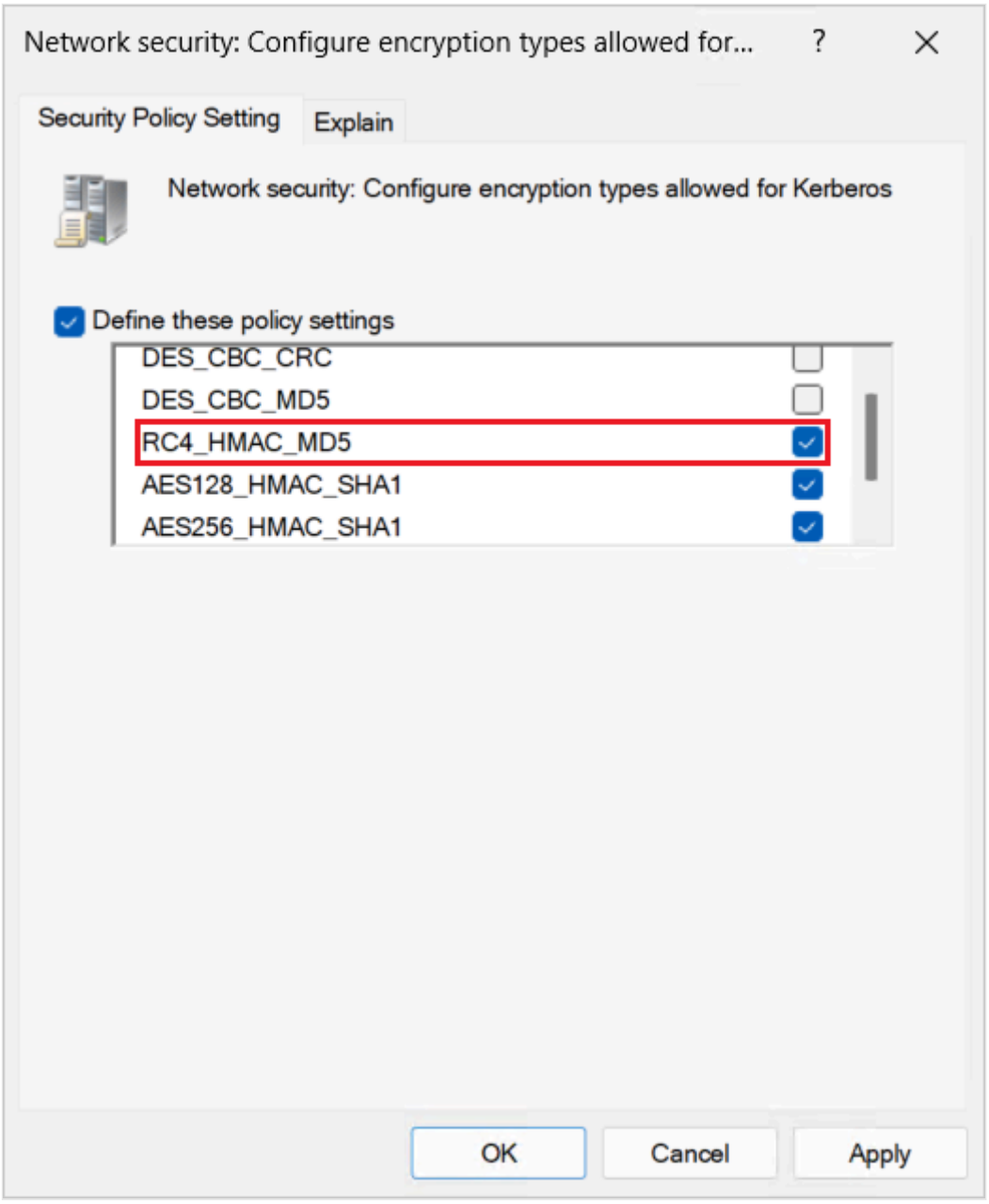
1. Create a Group Policy Object (GPO) using the Group Policy Management Console. For more information, see Group Policy Management.
2. Edit the GPO and navigate to **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Local Policies** > **Security Options**.

3. Locate the policy setting **Network security: Configure encryption types allowed for Kerberos**. Double-click the policy to open its properties.

4. Specify the encryption types you want to allow. Here are some examples:

    - To only allow AES-SHA1, check the boxes for **AES128_HMAC_SHA1** and **AES256_HMAC_SHA1**.

    - To allow AES-SHA1 and RC4, check the boxes for **AES128_HMAC_SHA1**, **AES256_HMAC_SHA1**, and **RC4_HMAC_MD5**.



5. Scope the GPO to the appropriate Organizational Units (OUs) or groups that contain the devices you want to apply the policy to. Ensure it propagates to the target devices, then restart the devices to apply the new settings.

6. After the policy is applied, monitor authentication events to ensure that no unexpected authentication failures occur. You can use the PowerShell scripts mentioned in the section

Use PowerShell to audit RC4 usage to verify that RC4 is no longer being used.

Alternatively, you can also disable RC4 on domain controllers by setting the following registry value, which only allows `AES128_HMAC_SHA1` and `AES256_HMAC_SHA1`:

- Key: `HKEY_LOCAL_MACHINE\System\CurrentControlSet\services\KDC`
- Value name: `DefaultDomainSupportedEncTypes`
- Type: `REG_DWORD`
- Value data: `0x38`

Identify authentication failures after disabling RC4

When RC4 is disabled, authentication might fail for some systems. This section covers common scenarios and indicators to identify a potential issue caused by disabling RC4. Here are two common scenarios where authentication failures might occur:

- **Server Message Block (SMB)**: when you attempt to access a share using SMB and RC4 is the only value for `msDS-SupportedEncryptionTypes`, an authentication failure generates a network error, as shown in the following screenshot:

- **Windows Management Instrumentation (WMI)**: when you attempt to create a remote PowerShell session using the New-PSSession cmdlet, which uses WMI, and RC4 is the only value for `msDS-SupportedEncryptionTypes`, an authentication failure generates an error message, as shown in the following example:

Output

```
New-PSSession : [vm01.contoso.com] Connecting to remote server vm01.contoso.com
failed with the following
error message : WinRM cannot process the request. The following error with
errorcode 0x80090342 occurred
while using Kerberos authentication: An unknown security error occurred.

Possible causes are:
   -The user name or password specified are invalid.
```

```
   -Kerberos is used when no authentication method and no user name are
specified.
   -Kerberos accepts domain user names, but not local user names.
   -The Service Principal Name (SPN) for the remote computer name and port does
not exist.
   -The client and remote computers are in different domains and there is no
trust between the two domains.
After checking for the above issues, try the following:
   -Check the Event Viewer for events related to authentication.
   -Change the authentication method; add the destination computer to the WinRM
TrustedHosts configuration
setting or use HTTPS transport.
Note that computers in the TrustedHosts list might not be authenticated.
-For more information about WinRM configuration, run the following command:
winrm help config. For more
information, see the about_Remote_Troubleshooting Help topic.
At line:1 char:6
+ $s = New-PSSession vm01.contoso.com
+      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
     + CategoryInfo          : OpenError:
(System.Manageme....RemoteRunspace:RemoteRunspace) [New-PSSession],
     PSRemotingTransportException
     + FullyQualifiedErrorId : err ,PSSessionOpenFailed
```

To work through either of these scenarios, follow these steps:

1. Attempt to make a direct service ticket request to this endpoint using the klist Windows command, which provides more detailed error for the failure. The following example shows a request for a service ticket to the HOST service on vm01.contoso.com.

   Open a PowerShell session and run the following command from a device that's trying to authenticate to the target device:

   Windows Command Prompt

   ```
   klist get HOST/vm01.contoso.com
   ```

   Here's an example of the output, showing that KDC doesn't support the requested encryption type:

   Output

   ```
   Current LogonId is 0:0xd6ed18
   Error calling API LsaCallAuthenticationPackage (GetTicket substatus):
   0x80090342

   klist failed with 0xc00002fd/-1073741059: The encryption type requested is not
   supported by the KDC.
   ```

2. Open Event ID 4769 on the KDC for the same request, you can see error code `0xE`, which maps to the error name `KDC_ERR_ETYPE_NOTSUPP`. This error means that the KDC doesn't support the encryption type. You can correlate more errors in Appendix C: Kerberos and LDAP Error Messages.

   Output

   ```
   A Kerberos service ticket was requested.

   Account Information:
       Account Name:       adele@CONTOSO.COM
       Account Domain:     CONTOSO.COM
       Logon GUID:     {00000000-0000-0000-0000-000000000000}
       MSDS-SupportedEncryptionTypes:  -
       Available Keys: -

   Service Information:
       Service Name:       HOST/vm01.contoso.com
       Service ID:     NULL SID
       MSDS-SupportedEncryptionTypes:  -
       Available Keys: -

   Domain Controller Information:
       MSDS-SupportedEncryptionTypes:  -
       Available Keys: -

   Network Information:
       Client Address:     ::ffff:192.168.1.112
       Client Port:        60090
       Advertized Etypes:  -

   Additional Information:
       Ticket Options:     0x40810000
       Ticket Encryption Type: 0xFFFFFFFF
       Session Encryption Type:    0x2D
       Failure Code:       0xE
       Transited Services: -
   ```

3. Follow the steps in the section The processed msDS-SupportedEncryptionTypes value doesn't include the AES-SHA1 bits to determine supported encryption type for the target device. In this example, the `msDS-SupportedEncryptionTypes` value for `vm01.contoso.com` is `0x4`, which indicates that only RC4 is configured for the account.

ⓘ **Note:** The author created this article with assistance from AI. Learn more

---

Last updated on 02/06/2026

Preventing Kerberos change password that uses RC4 secret keys

Article • 05/12/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional explains some limitations in the Kerberos protocol that could lead to a malicious user taking control of a user's account. There is a limitation in the Kerberos Network Authentication Service (V5) standard (RFC 4120), which is well-known within the industry, whereby an attacker can authenticate as a user or change that user's password if the attacker knows the user's secret key.

Possession of a user's password-derived Kerberos secret keys (RC4 and Advanced Encryption Standard [AES] by default) is validated during the Kerberos password change exchange per RFC 4757. The user's plaintext password is never provided to the Key Distribution Center (KDC), and by default, Active Directory domain controllers do not possess a copy of plaintext passwords for accounts. If the domain controller does not support a Kerberos encryption type, that secret key cannot be used to change the password.

In the Windows operating systems designated in the Applies To list at the beginning of this topic, there are three ways to block the ability to change passwords by using Kerberos with RC4 secret keys:

- Configure the user account to include the account option Smart card is required for interactive logon. This limits the user to only signing in with a valid smart card so that RC4 authentication service requests (AS-REQs) are rejected. To set the account options on an account, right-click on the account, the click Properties, and click the Account tab.

- Disable RC4 support for Kerberos on all domain controllers. This requires a minimum of a Windows Server 2008 domain functional level and an environment where all Kerberos clients, application servers, and trust relationships to and from the domain must support AES. Support for AES was introduced in Windows Server 2008 and Windows Vista.

  > ⓘ **Note**
  >
  > There is a known issue with disabling RC4 which can cause the system to restart. See the following hotfixes:
  > - Windows Server 2012 R2
  > - Windows Server 2012
  > - No hotfix is available for earlier versions of Windows Server

- Deploy domains set to Windows Server 2012 R2 domain functional level or higher, and configure users as members of the Protected Users security group. Because this feature disrupts more than just RC4 usage in the Kerberos protocol, see resources in the following See also section.

See Also

- For information about how to prevent the usage of the RC4 encryption type in Windows Server 2012 R2 domains, see Protected Users Security Group.
- For explanations about RFC 4120 and RFC 4757, see IETF Documents.

Kerberos clients allow IPv4 and IPv6 address hostnames in Service Principal Names (SPNs)

Article • 05/12/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Beginning with Windows 10 version 1507 and Windows Server 2016, Kerberos clients can be configured to support IPv4 and IPv6 hostnames in SPNs.

By default Windows will not attempt Kerberos authentication for a host if the hostname is an IP address. It will fall back to other enabled authentication protocols like NTLM. However, applications are sometimes hardcoded to use IP addresses which means the application will fall back to NTLM and not use Kerberos. This can cause compatibility issues as environments move to disable NTLM.

To reduce the impact of disabling NTLM a new capability was introduced that lets administrators use IP addresses as hostnames in Service Principal Names. This capability is enabled on the client through a registry key value.

Windows Command Prompt

```
reg add
"HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameter
s" /v TryIPSPN /t REG_DWORD /d 1 /f
```

To configure support for IP address hostnames in SPNs, create a TryIPSPN entry. This entry does not exist in the registry by default. After you have created the entry, change the DWORD value to 1. This registry value will need to be set on each client machine that needs to access Kerberos-protected resources by IP address.

Configuring a Service Principal Name as IP Address

A Service Principal Name is a unique identifier used during Kerberos authentication to identify a service on the network. An SPN is composed of a service, hostname, and optionally a port in form of `service/hostname[:port]` such as `host/fs.contoso.com`. Windows will register multiple SPNs to a computer object when a machine is joined to Active Directory.

IP addresses are not normally used in place of hostnames because IP addresses are often temporary. This can lead to conflicts and authentication failures as address leases expire and

renew. Therefore registering an IP address-based SPN is a manual process and should only be used when it's impossible to switch to a DNS-based hostname.

The recommended approach is to use the Setspn.exe tool. Note that an SPN can only be registered to a single account in Active Directory at a time so it is recommended that IP addresses have static leases if DHCP is used.

```
Setspn -s <service>/ip.address> <domain-user-account>
```

Example:

```
Setspn -s host/192.168.1.1 server01
```

NTLM overview

Article • 04/18/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional describes NTLM, any changes in functionality, and provides links to technical resources to Windows Authentication and NTLM for Windows Server.

NTLM authentication is a family of authentication protocols that are encompassed in the Windows Msv1_0.dll. The NTLM authentication protocols include LAN Manager version 1 and 2, and NTLM version 1 and 2. The NTLM authentication protocols authenticate users and computers based on a challenge response mechanism that proves to a server or domain controller that a user knows the password associated with an account. When the NTLM protocol is used, a resource server must take one of the following actions to verify the identity of a computer or user whenever a new access token is needed:

- Contact a domain authentication service on the domain controller for the computer's or user's account domain, if the account is a domain account.
- Look up the computer's or user's account in the local account database, if the account is a local account.

Current applications

NTLM authentication is still supported and must be used for Windows authentication with systems configured as a member of a workgroup. NTLM authentication is also used for local logon authentication on non-domain controllers. Kerberos version 5 authentication is the preferred authentication method for Active Directory environments, but a non-Microsoft or Microsoft application might still use NTLM.

Reducing the usage of the NTLM protocol in an IT environment requires both the knowledge of deployed application requirements on NTLM and the strategies and steps necessary to configure computing environments to use other protocols. New tools and settings have been added to help you discover how NTLM is used in order to selectively restrict NTLM traffic. For information about how to analyze and restrict NTLM usage in your environments, see Introducing the Restriction of NTLM Authentication to access the Auditing and restricting NTLM usage guide.

Related content

The following table lists relevant resources for NTLM and other Windows authentication technologies.

Expand table

Content type	References
Product evaluation	NTLM user authentication
Deployment	Extended Protection for Authentication Audit use of NTLMv1 on a Windows Server-based domain controller Ask the Directory Services Team : NTLM Blocking and You: Application Analysis and Auditing Methodologies in Windows 7 Windows Authentication Blog Configuring MaxConcurrentAPI for NTLM pass-through authentication
Development	Microsoft NTLM (Windows) [MS-NLMP]: NT LAN Manager (NTLM) Authentication Protocol Specification [MS-NNTP]: NT LAN Manager (NTLM) Authentication: Network News Transfer Protocol (NNTP) Extension [MS-NTHT]: NTLM Over HTTP Protocol Specification
Updates	New NTLM pass-through authentication protections for CVE-2022-21857

Passwords Overview

Article • 07/05/2022 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional describes passwords as used in the Windows operating systems, and links to documentation and discussions about the use of passwords in a credential management strategy.

Feature description

Operating systems and applications today are architected around passwords and even if you use smart cards or biometric systems, all accounts still have passwords and they can still be used in some circumstances. Some accounts, notably accounts used to run services, cannot even use smart cards and biometric tokens and therefore must use a password to authenticate. Windows protects passwords using cryptographic hashes.

For more information about Windows passwords, see Passwords Technical Overview.

Practical applications

In Windows and many other operating systems, the most common method for authenticating a user's identity is to use a secret passphrase or password. Securing your network environment requires that strong passwords be used by all users. This helps avoid the threat of a malicious user guessing a weak password, whether through manual methods or by using tools, to acquire the credentials of a compromised user account. This is especially true for administrative accounts. When you change a complex password regularly, it reduces the likelihood of a password attack compromising that account.

New and changed functionality

In Windows Server 2012 and Windows 8, picture passwords are new. Picture passwords are a combination of a user selected image coupled with a series of gestures. Picture password functionality is disabled on domain-joined computers. Links to more information about picture passwords are listed in See Also below.

There has been no change to password functionality in Windows Server 2012 and Windows 8. No new Group Policy settings have been added. However, improvements

and enhancements have been made in credential (and password) management, such as with picture passwords, Credential Locker and signing in to Windows 8 with a Microsoft account, formerly known as a Windows Live ID.

Deprecated functionality

No password functionality has been deprecated in Windows Server 2012 and Windows 8.

Software requirements

In enterprise environments, passwords are typically managed with Active Directory Domain Services. Passwords can also be managed on the local computer using the settings in local Security Settings, Account Policies, Password Policy.

See also

This table lists additional resources for password features, technology and credential management.

Expand table

Content type	References
Scenario documentation	Protecting your digital identity
Operations	Active Directory Users and Computers
Troubleshooting	Find out when your Password Expires - Active Directory PowerShell Blog
Security	Windows Server 2008 R2 and Windows 7 Threats and Countermeasures Guide: Account Policies Guidance to change and create strong passwords
Tools and settings	Group Policy Settings Reference for Windows and Windows Server on the Microsoft Download Center
Community resources	Protecting your digital identity Signing in to Windows 8 with a Windows Live ID Signing in with a picture password

Content type	References
	Optimizing picture password security

Feedback

Was this page helpful? Yes No

Passwords technical overview

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional explains how Windows implements passwords in versions of Windows beginning with Windows Server 2012 and Windows 8.1. It also discusses strong passwords, passphrases, and password policies.

How passwords are stored in Windows

This article provides information about the storage of passwords "at rest".

Windows represents passwords in 256-character UNICODE strings, but the logon dialog box is limited to 127 characters. Therefore, the longest possible password has 127 characters. Programs such as services can use longer passwords, but they must be set programmatically.

The Windows operating system stores passwords many different ways for different purposes.

Passwords stored as OWF

For use in Windows networking, including Active Directory domains, the password is stored two different ways by default: as the LAN Manager one-way function (LM OWF) and as the NT OWF. "One-way function" is a term that denotes a one-way mathematical transformation of data. The data that is being transformed can only be converted through encryption one way and cannot be reversed. The most common type of one-way function in use is a cryptographic hash. A hash is a small set of data that is mathematically tied to some larger set of data from which the hash is calculated. If the larger set of data is changed, the hash also changes. Hashes are useful, for example, as a checksum to verify that data has not been modified in transmission. A cryptographic hash is a hash that fulfills certain properties. A cryptographic hash must, for instance, be created in such a way that it is mathematically infeasible in a reasonable amount of time to infer the larger set of data from only the hash. Likewise, it is mathematically infeasible to find two sets of large data that generate the same hash.

There are many different types of one-way functions. All hash functions are, by definition, one-way functions. However, ordinary cryptographic functions that are typically reversible can also be used to create a one-way function. This can be done by

swapping the data and the key in a cryptographic function and encrypting the fixed value (the key) by using the data as the key. This is how the LM hash is computed. The LM hash is computed as follows:

1. The password is padded with NULL bytes to exactly 14 characters. If the password is longer than 14 characters, it is replaced with 14 NULL bytes for the remaining operations.
2. The password is converted to all uppercase.
3. The password is split into two 7-byte (56-bit) keys.
4. Each key is used to encrypt a fixed string.
5. The two results from step 4 are concatenated and stored as the LM hash.

The LM OWF algorithm is included in Windows for backward compatibility with software and hardware that cannot use newer algorithms.

The NT hash is simply a hash. The password is hashed by using the MD4 algorithm and stored. The NT OWF is used for authentication by domain members in both Windows NT 4.0 and earlier domains and in Active Directory domains.

Neither the NT hash nor the LM hash is salted. Salting is a process that combines the password with a random numeric value (the salt) before computing the one-way function.

Passwords stored in Active Directory

Passwords at rest are stored in several attributes of the Active Directory database (NTDS.DIT file). These attributes are listed in the following table:

Expand table

Active Directory Attribute	Content
unicodePwd	Encrypted NT Hash
dbcsPwd	Encrypted LM Hash
ntPwdHistory	Encrypted NT Hashes - Password History
lmPwdHistory	Encrypted LM Hashes - Password History
supplementalCredentials	Kerberos Keys, WDigest, etc.

Note

> The storage of LM hashes is disabled by default since Windows Vista and Windows Server 2008.

When stored in the DIT file, the NT hash is protected by two layers of encryption. In Windows Server 2016/Windows 10 and later versions, it is first encrypted with DES for backwards compatibility and then with CNG BCrypt AES-256 (see CNG BCRYPT_AES_ALGORITHM). Previous Windows versions encrypt NT hashes using two layers of DES + RC4 encryption.

For more information about Supplemental Credentials, see MS-SAMR: supplementalCredentials and Supplemental Credentials Structures.

Passwords stored in the local SAM

On domain members and workstations, local user account password hashes are stored in a local Security Account Manager (SAM) Database located in the registry. They are encrypted using the same encryption and hashing algorithms as Active Directory. The passwords in the supplementalCredentials attribute for local user accounts are also stored in the local SAM Database since Windows Server 2016.

Cached credentials

Windows also stores a password verifier on domain members when a domain user logs on to that domain member. This verifier can be used to authenticate a domain user if the computer is not able to access the domain controller. The password verifier is also commonly called a cached credential. It is computed by taking the NT hash, concatenating the user name to it, and then hashing the result by using the MD4 hash function.

How passwords work in Windows

In Windows and many other operating systems, one method for authenticating a user's identity is to use a secret passphrase or password.

We recommend using secure multi-factor authentication such as Smart Card, FIDO, and Windows Hello for Business. However, password authentication is still required in some scenarios.

Securing your network environment requires that strong passwords be used by all users. This helps avoid the threat of a malicious user guessing a weak password, whether through manual methods or by using tools, to acquire the credentials of a compromised

user account. This is especially true for administrative accounts. When you change a complex password regularly, it reduces the likelihood of a successful password attack.

Password policy settings control the complexity and lifetime of passwords. Password policies affect Windows passwords, not necessarily feature passwords.

Users' ability to modify their passwords is governed by the password policies and the available interfaces. For example, through the Secure Desktop, users can change their password at any time based upon the password policies administered by the system administrator or domain administrator. Features such as Windows Vault, BitLocker, and Encrypting File System allow users to modify passwords specific to that feature.

How passwords are used in Windows

When a user logs on, the password the user types is converted into both types of one-way functions and held in memory by the Local Security Authority Subsystem Service (LSASS) process. If the user using a local account for authentication, the NT OWF is compared against the locally stored NT hash, and if the two match, the user is logged on. If the user is authenticating against an Active Directory domain by using a host name to access a resource, the NT hash is used in a Kerberos logon against the Key Distribution Center (KDC), which is typically the domain controller.

Kerberos cannot be used in the following situations:

- Authenticating against a domain running only Windows NT 4.0 or earlier
- Accessing a resource on an Active Directory domain member by using an IP address rather than a host name
- Accessing a resource on a computer that is not a member of an Active Directory domain
- Accessing a resource on a computer that is a member of an Active Directory domain but not trusted by your domain
- Accessing any resource on a computer running that does not support Kerberos

In these situations, the authentication process uses two different protocols, called LAN Manager and NTLM. The process starts with the client requesting a challenge from the authentication server. After the challenge is received, the client computes a response to this challenge. This is done by first padding the two hashes of the password with null values to 168 bits. The 168 bits of each hash are then split into three 56-bit DES keys. The six DES keys are then used to encrypt the challenge. The three cipher texts produced by using the LM hash are concatenated and become the LAN Manager response. The three cipher texts produced by using the NT hash are concatenated and become the NTLM response.

The functions used to compute the response may be modified by the **LM Compatibility Level** setting in the **Network security: LAN Manager authentication level** Group Policy setting. If that value is set to 1 or lower, the client will send the original LAN Manager and NTLM responses. If it is set to 2, only the NTLM response is sent. If it is set to 3 or higher, a new version of both protocols is used. The NTLM version is called NTLMv2. The LAN Manager version is often referred to as LMv2. Both protocols use the NT hash to compute the response, and both use a client-side challenge, either instead of or in addition to the server challenge. In addition, if the **LM Compatibility Level** setting is set to 1 or higher, the NTLM response is time-stamped to help prevent replay attacks. For information about the **LM Compatibility Level** setting, see Network security: LAN Manager authentication level.

Strong passwords

Passwords provide the first line of defense against unauthorized access to your organization. Beginning with Windows Server 2003, Windows checks the complexity of the password for the Administrator account during setup of the operating system. If the password is blank or does not meet complexity requirements, the **Windows Setup** dialog box prompts you to create a strong password for the Administrator account. If you leave this password blank, you will not be able to access this account over the network.

Weak passwords provide attackers with easy access to your computers and network, while strong passwords are considerably more difficult to crack. The following table compares weak and strong passwords.

Expand table

Weak password	Strong password
Blank	Is at least seven characters long
Contains easily discoverable or known information, such as user name or domain name	Contains "secret" or random information
Is similar to previous passwords	Is significantly different from previous passwords
Contains a complete dictionary word	Contains a mix of the following characters: - Uppercase letters - Lowercase letters

Weak password	Strong password
	- Numerals - Symbols including spaces

An example of a strong password is J*p2leO4>F.

A password can meet most of the criteria of a strong password but still be rather weak. For example, Hello2U! is a relatively weak password even though it meets most of the criteria for a strong password and also meets the complexity requirements of password policy. H!elZl2o is a strong password because the dictionary word is interspersed with symbols, numbers, and other letters. It is important to educate users about the benefits of using strong passwords and to teach them how to create passwords that are actually strong.

You can create passwords that contain characters from the extended ANSI character set. Using extended ANSI characters increases the number of characters that you can choose when you create a password. As a result, it might take more time for password-cracking software to crack passwords that contain these extended ANSI characters than it does to crack other passwords. Before using extended ANSI characters in your password, test them thoroughly to make sure that passwords containing extended ANSI characters are compatible with the applications that your organization uses. Be especially cautious about using extended ANSI characters in passwords if your organization uses several different operating systems. For example, these systems may standardize in ISO-8859-15. The actual protocol implementation on Windows often use UNICODE or UTF8 rather than actual ANSI encoding.

Examples of passwords that contain characters from the extended ANSI character set are kUµ!¶0o and Wf©$0k#»g¤5ªrd.

Passphrases in Windows

A passphrase is a different form of token-based password in which the tokens are words instead of symbols from a character set. An example of a passphrase is a sentence that contains special characters, numerals, uppercase letters, and lowercase letters. The key differences between passphrases and passwords are:

- A passphrase usually has spaces; passwords do not.
- A passphrase is much longer than the vast majority of words, and, more important, longer than any random string of letters that an ordinary person could remember.

Passphrases that conform to the character limit as set in the policy are generally, more difficult to crack than passwords because they contain more characters. It is the LM and NT hash that stores the password or passphrase, and the LM hash is the weaker of the two.

There are several ways to ensure the LM hash is not stored; one of them is to use passwords or passphrases longer than 14 characters. You can also use the **Network security: Do not store LAN Manager hash value on next password change** Group Policy setting. Using this policy setting globally turns off storage LM hashes for all accounts. The change will take effect the next time the password is changed. Because the policy's effect is not immediate, you will not immediately notice any potential interoperability problems caused by not storing LM hashes.

Local password policies available in Windows

You can implement a password policy setting that enforces password complexity requirements. For more information about this policy setting, see Password must meet complexity requirements. For information about how to apply a password policy, see Apply or Modify a Password Policy. For information about all available password policy settings, see Password Policy.

Fine-grained password policy available through Active Directory Domain Services (AD DS)

Beginning with Windows Server 2008, you can use fine-grained password policies to specify multiple password policies and apply different password restrictions and account lockout policies to different sets of users within a single domain. For example, to increase the security of privileged accounts, you can apply stricter settings to the privileged accounts and then apply less strict settings to the accounts of other users. Or in some cases, you may want to apply a special password policy for accounts whose passwords are synchronized with other data sources.

To store fine-grained password policies, two new object classes exist in the AD DS schema:

- Password Settings Container
- Password Settings

For more information about these policies, see AD DS: Fine-Grained Password Policies.

Feedback

Was this page helpful? Yes No

System key utility technical overview

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional describes the system key utility (Syskey), which protects the Security Accounts Manager (SAM) database in Windows operating systems.

> **Note**
>
> Syskey utility is no longer supported in Windows 10, version 1607, Windows Server 2016, and later versions.

What is the system key utility?

Password information for user accounts is stored in the SAM database of the registry on workstations and member servers. On domain controllers, password information is stored in directory services. It is not unusual for password-cracking software to target the SAM database or directory services to access passwords for user accounts. The system key utility (Syskey) provides an extra line of defense against password-cracking software. It uses strong encryption techniques to secure account password information that is stored in the SAM database or in directory services. Cracking encrypted account passwords is more difficult and time consuming than cracking nonencrypted account passwords.

There are three system key options in the **Startup Key** dialog box that are designed to meet the needs of different environments, as described in the following table.

Expand table

System key option	Relative security level	Description
System Generated Password, Store Startup Key Locally	+	Uses a computer-generated random key as the system key and stores an encrypted version of the key on the local computer. This option provides strong encryption of password information in the registry, and it enables the user to restart the computer without the need for an administrator to enter a password or insert a disk

System key option	Relative security level	Description
Administrator generated password, Password Startup	++	Uses a computer-generated random key as the system key and stores an encrypted version of the key on the local computer. The key is also protected by an administrator-chosen password. Users are prompted for the system key password when the computer is in the initial startup sequence. The system key password is not stored anywhere on the computer.
System Generated Password, Store Startup Key on Floppy Disk	+++	Uses a computer-generated random key and stores the key on a floppy disk. The floppy disk that contains the system key is required for the system to start, and it must be inserted at a prompt during the startup sequence. The system key is not stored anywhere on the computer.

Use of the system key utility is optional. If the disk that contains the system key is lost, or if the password is forgotten, you cannot start the computer without restoring the registry to the state it was in before the system key was used.

How the system key utility works

Each time a new user is added to a computer, the Windows Data Protection API (DPAPI) generates a master key that is used to protect all other private keys used by applications and services running in that user's context, such as Encrypting File System (EFS) keys and S/MIME keys. The computer also has its own master key that protects system keys such as IPsec keys, computer keys, and SSL keys. All these master keys are then protected by a computer's startup key. When you start a computer, the startup key decrypts the master keys. The startup key also protects the local SAM database on each computer, the computer's Local Security Authority (LSA) secrets, account information stored in Active Directory Domain Services (AD DS) on domain controllers, and the administrator account password used for system recovery in Safe Mode.

The Syskey utility lets you choose where that startup key is stored. By default, the computer generates a random key and scatters it throughout the registry; a complex obfuscation algorithm ensures that the scatter pattern is different on every Windows installation. You can change this to one of two other Syskey modes: you can continue to use a computer-generated key but store it on a floppy disk, or you can have the system prompt during startup for a password that is used to derive the master key. You can always change between the three options, but if you have enabled either **System Generated Password, Store Startup Key on Floppy Disk** or **Administrator generated password, Password Startup** and you have lost your floppy disk or forgotten your

password, your only recovery option is to use a repair disk to restore the registry to the state it was in before you enabled the Syskey mode. You will lose any other changes made between then and now. To change your startup key, open a command prompt, and type *syskey* to run the Syskey utility.

Feedback

Was this page helpful? Yes No

RPC Interface Restriction

Article • 02/25/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

The Remote Procedure Call (RPC) service makes RPC interfaces secure by default to reduce attacks. The `RestrictRemoteClients` registry key enables you to modify the behavior of all RPC interfaces on the system and can be used to eliminate remote anonymous access to RPC interfaces on the system, with some exceptions. You can apply other interface controls using the `EnableAuthEpResolution` registry key covered in this article. Both RPC application developers and system administrators can configure RPC Interface Restrictions.

Prerequisites

When using `RestrictRemoteClients` on your server:

- Your RPC clients are required to use RPC security when contacting your server applications, which is the best method to mitigate security threats.
- Exempt your RPC setting from requiring authentication by setting the *RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH* flag during registration. Doing so configures RPC to allow anonymous connections to your applications.

RestrictRemoteClients threat mitigation

Enabling `RestrictRemoteClients` is useful in mitigation against worms that rely on exploitable buffer overruns that can be invoked remotely through anonymous connections. RPC applications that expect to receive call from remote anonymous RPC clients might not run correctly when using this feature. As a result, applications that use DCOM (Distributed Component Object Model) might not work correctly if this value is set.

RPC calls using connectionless protocols fail if this key is enabled. Secure RPC calls over connectionless protocols, such as UDP (User Datagram Protocol) and IPX (Internetwork Packet Exchange), use a lower level of security compared to connection-oriented protocols. Specifically, `ncadg_ip_udp` and `ncadg_ipx` are considered less secure. For the purposes of this policy, these calls are always considered nonsecure.

To allow RPC client calls using connectionless protocols, keep `RestrictRemoteClients` value set to **disabled**.

Restrictions for RPC clients

When an interface is registered using `RpcServerRegisterIf`, RPC allows the server application to restrict access to the interface through a security callback. The `RestrictRemoteClients` registry key forces RPC to perform additional security checks for all interfaces, even if the interface has no registered security callback.

RPC clients that use the named pipe protocol sequence (`ncacn_np`) are exempt from all restrictions discussed in this section. The named pipe protocol sequence can't be restricted due to significant backwards compatibility issues.

`RestrictRemoteClients` can also be controlled programmatically in the rpcdce.h API (Application Programming Interface) header.

Configuring RestrictRemoteClients

To edit these policies using the GPO (Group Policy Object) editor:

1. Select **Start** > type **gpedit.msc** > hit `Enter` to open the **Local Group Policy Editor**.
2. To enable the equivalent of `RestrictRemoteClients` settings, navigate to **Computer Configuration\Administrative Templates\System\Remote Procedure Call\Restrictions for Unauthenticated RPC Clients**, then select one of the following:

    - **Disabled** - This setting is the default value for server SKUs. Corresponds to the value *RPC_RESTRICT_REMOTE_CLIENT_NONE* in rpcdce.h and it's the responsibility of the server application to impose appropriate RPC restrictions.
    - **Authenticated** - Corresponds to the value *RPC_RESTRICT_REMOTE_CLIENT_DEFAULT* in rpcdce.h. Allows only authenticated RPC Clients to connect to RPC Servers running on the machine on which the policy setting is applied. RPC runtime rejects Anonymous calls. If an interface registers a security callback and provides the *RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH* flag, then this restriction doesn't apply to that interface.
    - **Authenticated without exceptions**[1] - Corresponds to the value *RPC_RESTRICT_REMOTE_CLIENT_HIGH* in rpcdce.h. Allows only authenticated RPC Clients to connect to RPC Servers running on the machine on which the

policy setting is applied. No exceptions are allowed when this value is set as the system can't receive remote anonymous calls using RPC.

Any changes to either of these settings requires a **system reboot** to take effect.

> ⊗ **Caution**
>
> [1]Don't use this value without significant testing. For more information, see Restrictions for Unauthenticated RPC Clients.

EnableAuthEpResolution

The `EnableAuthEpResolution` key allows the RPC client runtime to use NTLM (NT LAN Manager) to authenticate to the endpoint mapper when enabled. This authenticated query takes place only if the actual RPC client call uses RPC authentication.

RPC clients make calls to an RPC server that has a registered dynamic endpoint with RPC Endpoint Mapper Client Authentication enabled. These calls are queried on behalf of authenticated calls using NTLM authentication.

RPC clients that attempt to make a call using a dynamic endpoint queries the RPC Endpoint Mapper on the server to determine what endpoint it should connect to. This query is performed anonymously, even if the RPC client call itself is performed using RPC security. The RPC Endpoint Mapper interface isn't accessible anonymously if the `RestrictRemoteClients` setting is enabled.

Configuring EnableAuthEpResolution

To edit these policies using the GPO (Group Policy Object) editor:

1. Select **Start** > type **gpedit.msc** > hit `Enter` to open the **Local Group Policy Editor**.
2. To enable the equivalent of `EnableAuthEpResolution` settings, navigate to **Computer Configuration\Administrative Templates\System\Remote Procedure Call\Enable RPC Endpoint Mapper Client Authentication**, then select one of the two available settings:

   - **Disabled** - This setting is the default. RPC clients don't authenticate to the Endpoint Mapper Service, but they're able to communicate with the Endpoint Mapper Service on Windows NT4 Server.
   - **Enabled** - PC clients authenticate via the Endpoint Mapper Service for calls that contain authentication information. Clients making such calls aren't able

to communicate with the Windows NT4 Server Endpoint Mapper Service.

Any changes to either of these settings requires a **system reboot** to take effect.

> **ⓘ Important**
>
> The following Group Policy settings found in **Computer Configuration\Windows Settings\Security Settings\Local Policies\Security Options** can't be used with `EnableAuthEpResolution`:
>
> - Network security: Restrict NTLM: Incoming NTLM traffic – **"Deny All Accounts"**
> - Network security: Restrict NTLM: Outgoing NTLM traffic to remote servers – **"Deny All"**
>
> Moving away from NTLM to better secure your environment is encouraged. If faced with a choice between restricting NTLM and using `EnableAuthEpResolution`, the recommended approach is that you restrict NTLM in your environment.

Other RPC Interface Registration flags

These interface registration flags are created to make it easier for an application developer to secure an RPC interface:

- **RPC_IF_ALLOW_CALLBACKS_WITH_NO_AUTH**
  When this flag is registered, the RPC runtime invokes the registered security callback for all calls, regardless of the call security settings. Without this flag, RPC rejects all unauthenticated calls before they reach the security callback. This flag works only when a security callback is registered.

- **RPC_IF_SEC_NO_CACHE**
  A security callback is registered for an interface to restrict access. Typically, the security callback impersonates the client to verify if the client has sufficient rights to make a call to the interface. Once a particular client identity passes the security callback, it usually passes the same security callback on subsequent attempts.

  The RPC runtime takes advantage of this pattern by remembering when an individual client identity passes a security callback. It then skips the security callback for subsequent calls by that client to the same interface. This feature is called security callback caching and existed since the Microsoft Windows 2000 family of operating systems. You can use the *RPC_IF_SEC_NO_CACHE* flag to

disable security callback caching for a given interface. Which is useful if the security check might change, possibly rejecting a client identity that was previously permitted.

- **RPC_IF_LOCAL_ONLY**
  When an interface is registered with this flag, RPC does reject calls made by remote RPC clients. In addition, local calls over all `ncadg_`* protocol sequences and all `ncacn_`* protocol sequences (except for named pipes, using `ncacn_np`) are also rejected. If a call is made on `ncacn_np`, RPC allows the call only if it doesn't come from SRV (Service Location Protocol), which filters out all remote calls. `Ncalrpc` calls are always allowed through.

The use of these flags is at the discretion of the application developer. RPC application developers are provided with additional security tools to help secure their RPC interface as these flags doesn't change any existing application or cause them to run incorrectly.

See also

- Remote Procedure Call (RPC)
- RPC error troubleshooting

Feedback

Was this page helpful? Yes No

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

TLS (Schannel SSP) changes in Windows 10 and Windows Server 2016

Article • 02/14/2023 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Cipher Suite Changes

Windows 10, version 1511 and Windows Server 2016 add support for configuration of cipher suite order using Mobile Device Management (MDM).

For cipher suite priority order changes, see Cipher Suites in Schannel.

Added support for the following cipher suites:

- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 (RFC 5289) in Windows 10, version 1507 and Windows Server 2016
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 (RFC 5289) in Windows 10, version 1507 and Windows Server 2016

DisabledByDefault change for the following cipher suites:

- TLS_DHE_DSS_WITH_AES_256_CBC_SHA256 (RFC 5246) in Windows 10, version 1703
- TLS_DHE_DSS_WITH_AES_128_CBC_SHA256 (RFC 5246) in Windows 10, version 1703
- TLS_DHE_DSS_WITH_AES_256_CBC_SHA (RFC 5246) in Windows 10, version 1703
- TLS_DHE_DSS_WITH_AES_128_CBC_SHA (RFC 5246) in Windows 10, version 1703
- TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (RFC 5246) in Windows 10, version 1703
- TLS_RSA_WITH_RC4_128_SHA in Windows 10, version 1709
- TLS_RSA_WITH_RC4_128_MD5 in Windows 10, version 1709

Starting with Windows 10, version 1507 and Windows Server 2016, SHA 512 certificates are supported by default.

RSA key changes

Windows 10, version 1507 and Windows Server 2016 add registry configuration options for client RSA key sizes.

For more information, see KeyExchangeAlgorithm key sizes.

Diffie-Hellman key changes

Windows 10, version 1507 and Windows Server 2016 add registry configuration options for Diffie-Hellman key sizes.

For more information, see KeyExchangeAlgorithm key sizes.

SCH_USE_STRONG_CRYPTO option changes

With Windows 10, version 1507 and Windows Server 2016, SCH_USE_STRONG_CRYPTO option now disables NULL, MD5, DES, and export ciphers.

Elliptical Curve changes

Windows 10, version 1507 and Windows Server 2016 add Group Policy configuration for elliptical curves under Computer Configuration > Administrative Templates > Network > SSL Configuration Settings. The ECC Curve Order list specifies the order in which elliptical curves are preferred as well as enables supported curves which are not enabled.

Added support for the following elliptical curves:

- BrainpoolP256r1 (RFC 7027) in Windows 10, version 1507 and Windows Server 2016
- BrainpoolP384r1 (RFC 7027) in Windows 10, version 1507 and Windows Server 2016
- BrainpoolP512r1 (RFC 7027) in Windows 10, version 1507 and Windows Server 2016
- Curve25519 (RFC draft-ietf-tls-curve25519) in Windows 10, version 1607 and Windows Server 2016

Dispatch level support for SealMessage & UnsealMessage

Windows 10, version 1507 and Windows Server 2016 add support for SealMessage/UnsealMessage at dispatch level.

DTLS 1.2

Windows 10, version 1607 and Windows Server 2016 add support for DTLS 1.2 (RFC 6347).

HTTP.SYS thread pool

Windows 10, version 1607 and Windows Server 2016 add registry configuration of the size of the thread pool used to handle TLS handshakes for HTTP.SYS.

Registry path:

HKLM\SYSTEM\CurrentControlSet\Control\LSA

To specify a maximum thread pool size per CPU core, create a **MaxAsyncWorkerThreadsPerCpu** entry. This entry does not exist in the registry by default. After you have created the entry, change the DWORD value to the desired size. If not configured, then the maximum is 2 threads per CPU core.

Next Protocol Negotiation (NPN) support

Beginning with Windows 10 version 1703, Next Protocol Negotiation (NPN) has been removed and is no longer supported.

Pre-Shared Key (PSK)

Windows 10, version 1607 and Windows Server 2016 add support for PSK key exchange algorithm (RFC 4279).

Added support for the following PSK cipher suites:

- TLS_PSK_WITH_AES_128_CBC_SHA256 (RFC 5487) in Windows 10, version 1607 and Windows Server 2016
- TLS_PSK_WITH_AES_256_CBC_SHA384(RFC 5487) in Windows 10, version 1607 and Windows Server 2016
- TLS_PSK_WITH_NULL_SHA256 (RFC 5487) in Windows 10, version 1607 and Windows Server 2016
- TLS_PSK_WITH_NULL_SHA384 (RFC 5487) in Windows 10, version 1607 and Windows Server 2016
- TLS_PSK_WITH_AES_128_GCM_SHA256 (RFC 5487) in Windows 10, version 1607 and Windows Server 2016
- TLS_PSK_WITH_AES_256_GCM_SHA384 (RFC 5487) in Windows 10, version 1607 and Windows Server 2016

Session Resumption without Server-Side State server-side performance improvements

Windows 10, version 1507 and Windows Server 2016 provide 30% more session resumptions per second with session tickets compared to Windows Server 2012.

Session Hash and Extended Master Secret Extension

Windows 10, version 1507 and Windows Server 2016 add support for RFC 7627: Transport Layer Security (TLS) Session Hash and Extended Master Secret Extension.

Due to this change, Windows 10 and Windows Server 2016 requires 3rd party CNG SSL provider updates to support NCRYPT_SSL_INTERFACE_VERSION_3, and to describe this new interface.

SSL support

Beginning with Windows 10, version 1607 and Windows Server 2016, the TLS client and server SSL 3.0 is disabled by default. This means that unless the application or service specifically requests SSL 3.0 via the SSPI, the client will never offer or accept SSL 3.0 and the server will never select SSL 3.0.

Beginning with Windows 10 version 1607 and Windows Server 2016, SSL 2.0 has been removed and is no longer supported.

Changes to Windows TLS adherence to TLS 1.2 requirements for connections with non-compliant TLS clients

In TLS 1.2, the client uses the "signature_algorithms" extension to indicate to the server which signature/hash algorithm pairs may be used in digital signatures (i.e., server certificates and server key exchange). The TLS 1.2 RFC also requires that the server Certificate message honor "signature_algorithms" extension:

"If the client provided a "signature_algorithms" extension, then all certificates provided by the server MUST be signed by a hash/signature algorithm pair that appears in that extension."

In practice, some third-party TLS clients do not comply with the TLS 1.2 RFC and fail to include all the signature and hash algorithm pairs they are willing to accept in the "signature_algorithms" extension, or omit the extension altogether (the latter indicates to the server that the client only supports SHA1 with RSA, DSA or ECDSA).

A TLS server often only has one certificate configured per endpoint, which means the server can't always supply a certificate that meets the client's requirements.

Prior to Windows 10 and Windows Server 2016, the Windows TLS stack strictly adhered to the TLS 1.2 RFC requirements, resulting in connection failures with RFC non-compliant TLS clients and interoperability issues. In Windows 10 and Windows Server 2016, the constraints are relaxed and the server can send a certificate that does not comply with TLS 1.2 RFC, if that's the server's only option. The client may then continue or terminate the handshake.

When validating server and client certificates, the Windows TLS stack strictly complies with the TLS 1.2 RFC and only allows the negotiated signature and hash algorithms in the server and client certificates.

Feedback

Was this page helpful? Yes No

Manage Transport Layer Security (TLS)

Article • 04/18/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

Configure TLS cipher suite order

A cipher suite is a set of cryptographic algorithms. Different Windows versions support different TLS cipher suites and priority order. See Cipher Suites in TLS/SSL (Schannel SSP) for the default order supported by the Microsoft Schannel Provider in different Windows versions.

> ⓘ **Note**
>
> You can also modify the list of cipher suites by using CNG functions, see Prioritizing Schannel Cipher Suites for details.

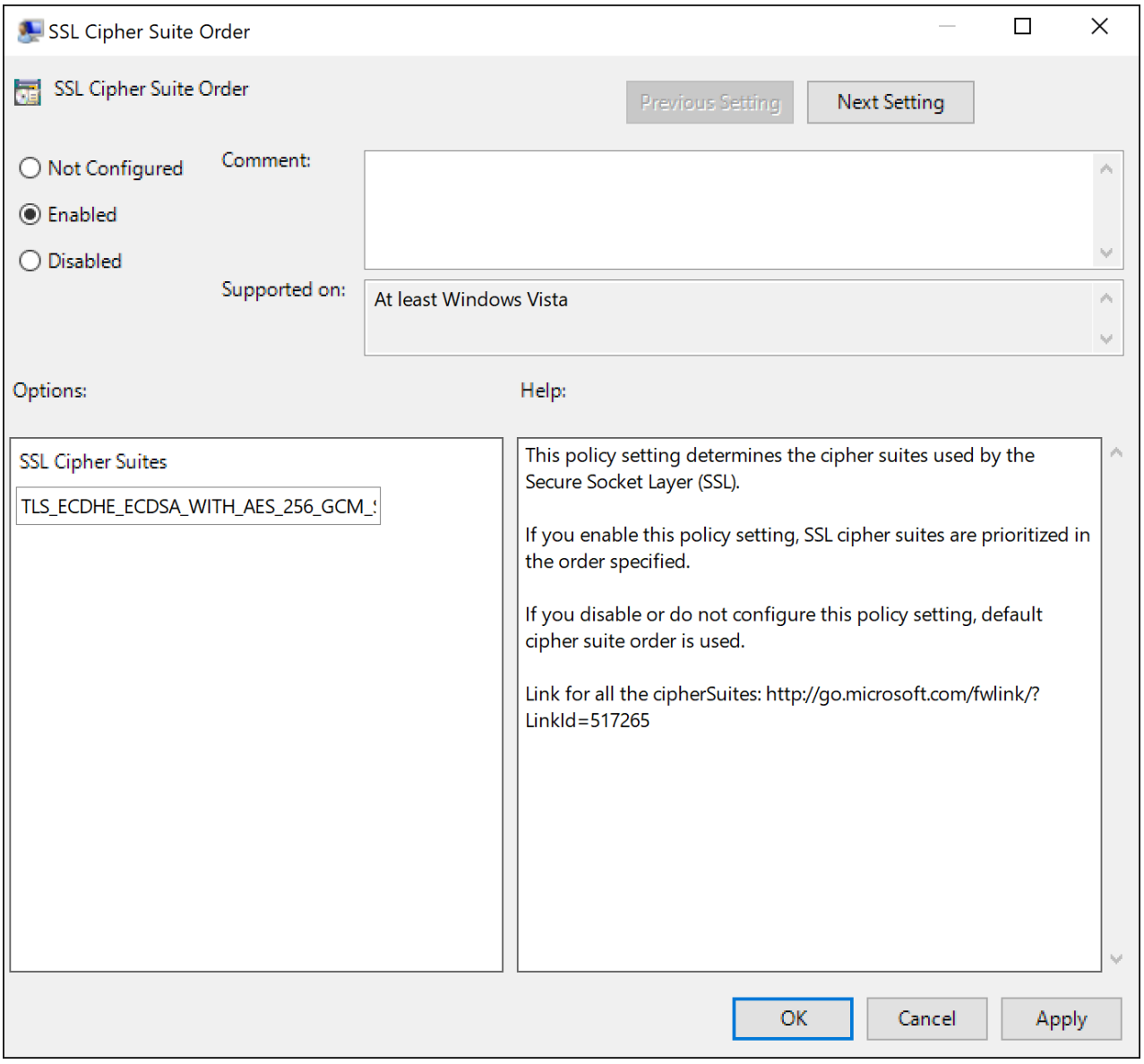
Changes to the TLS cipher suite order take effect on the next boot. Until restart or shutdown, the existing order is in effect.

> ⚠ **Warning**
>
> Updating the registry settings for the default priority ordering isn't supported and may be reset with servicing updates.

Configure TLS cipher suite order by using Group Policy

You can use the SSL cipher suite order Group Policy settings to configure the default TLS cipher suite order.

1. From the Group Policy Management Console, go to **Computer Configuration** > **Administrative Templates** > **Network** > **SSL Configuration Settings**.
2. Double-click **SSL Cipher Suite Order**, and then select the **Enabled** option.
3. Right-click **SSL Cipher Suites** box and select **Select all** from the pop-up menu.



4. Right-click the selected text, and select **copy** from the pop-up menu.

5. Paste the text into a text editor such as notepad.exe and update with the new cipher suite order list.

> **Note**
>
> The TLS cipher suite order list must be in strict comma delimited format. Each cipher suite string ends with a comma to the right side of it. Additionally, the list of cipher suites is limited to 1,023 characters.

6. Replace the list in the **SSL Cipher Suites** with the updated ordered list.

7. Select **OK** or **Apply**.

Configure TLS cipher suite order by using MDM

The Windows 10 Policy CSP supports configuration of the TLS Cipher Suites. For more information, see Cryptography/TLSCipherSuites.

Configure TLS cipher suite order by using TLS PowerShell Cmdlets

The TLS PowerShell module supports getting the ordered list of TLS cipher suites, disabling a cipher suite, and enabling a cipher suite. For more information, see TLS Module.

Configure TLS ECC curve order

Beginning with Windows 10 and Windows Server 2016, ECC curve order can be configured independent of the cipher suite order. If the TLS cipher suite order list has elliptic curve suffixes, they'll be overridden by the new elliptic curve priority order, when enabled. This allows organizations to use a Group Policy object to configure different versions of Windows Server with the same cipher suites order.

Manage ECC curves using CertUtil

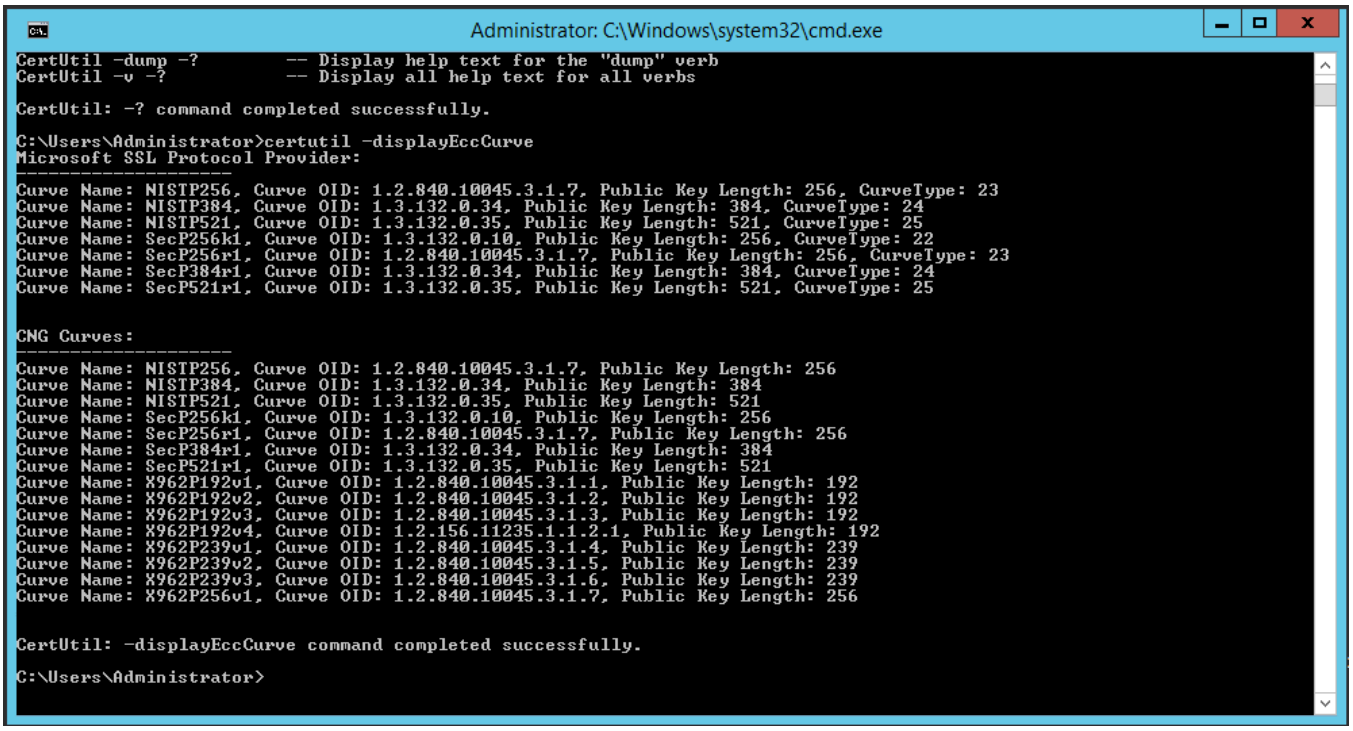
Beginning with Windows 10 and Windows Server 2016, Windows provides elliptic curve parameter management through the command line utility **certutil.exe**. Elliptic curve parameters are stored in the bcryptprimitives.dll. Administrators can add, and remove curve parameters to and from Windows Server using certutil.exe. Certutil.exe stores the curve parameters securely in the registry. Windows Server can begin using the curve parameters by the name associated with the curve.

Display registered curves

Use the following certutil.exe command to display a list of curves registered for the current computer.

PowerShell

```
certutil.exe –displayEccCurve
```



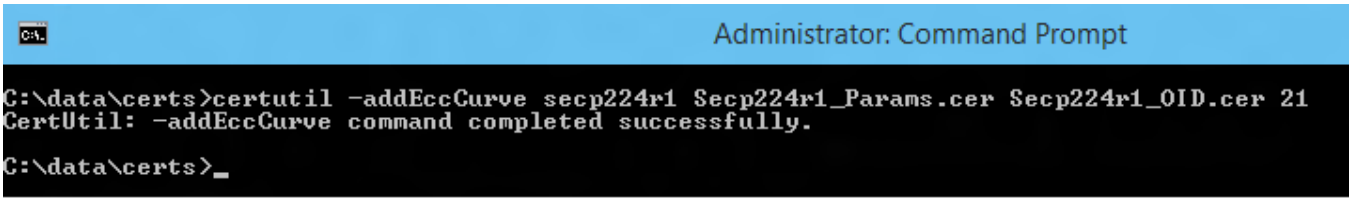
Add a new curve

Organizations can create and use curve parameters researched by other trusted entities. Administrators wanting to use these new curves in Windows must add the curve. Use the following certutil.exe command to add a curve to current computer:

PowerShell

```
Certutil –addEccCurue curveName curveParameters [curveOID] [curveType]
```

- The **curveName** argument represents the name of the curve under which the curve parameters were added.
- The **curveParameters** argument represents the filename of a certificate that contains the parameters of the curves you want to add.
- The **curveOid** argument represents a filename of a certificate that contains the OID of the curve parameters you want to add (optional).
- The **curveType** argument represents a decimal value of the named curve from the EC Named Curve Registry (optional).



Remove a previously added curve

Administrators can remove a previously added curve using the following certutil.exe command:

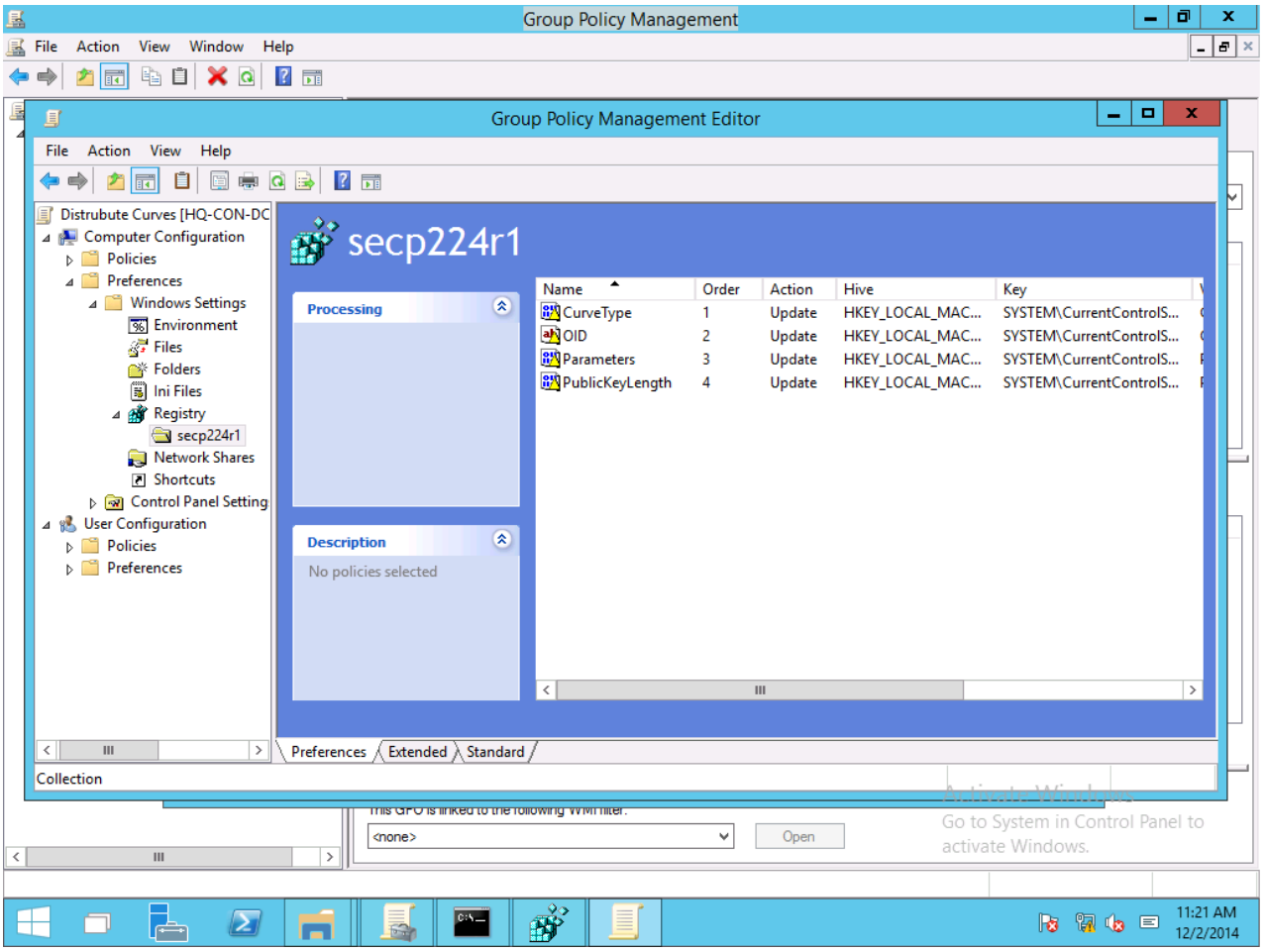
```
PowerShell

certutil.exe –deleteEccCurve curveName
```

Windows can't use a named curve after an administrator removes the curve from computer.

Manage ECC curves using Group Policy

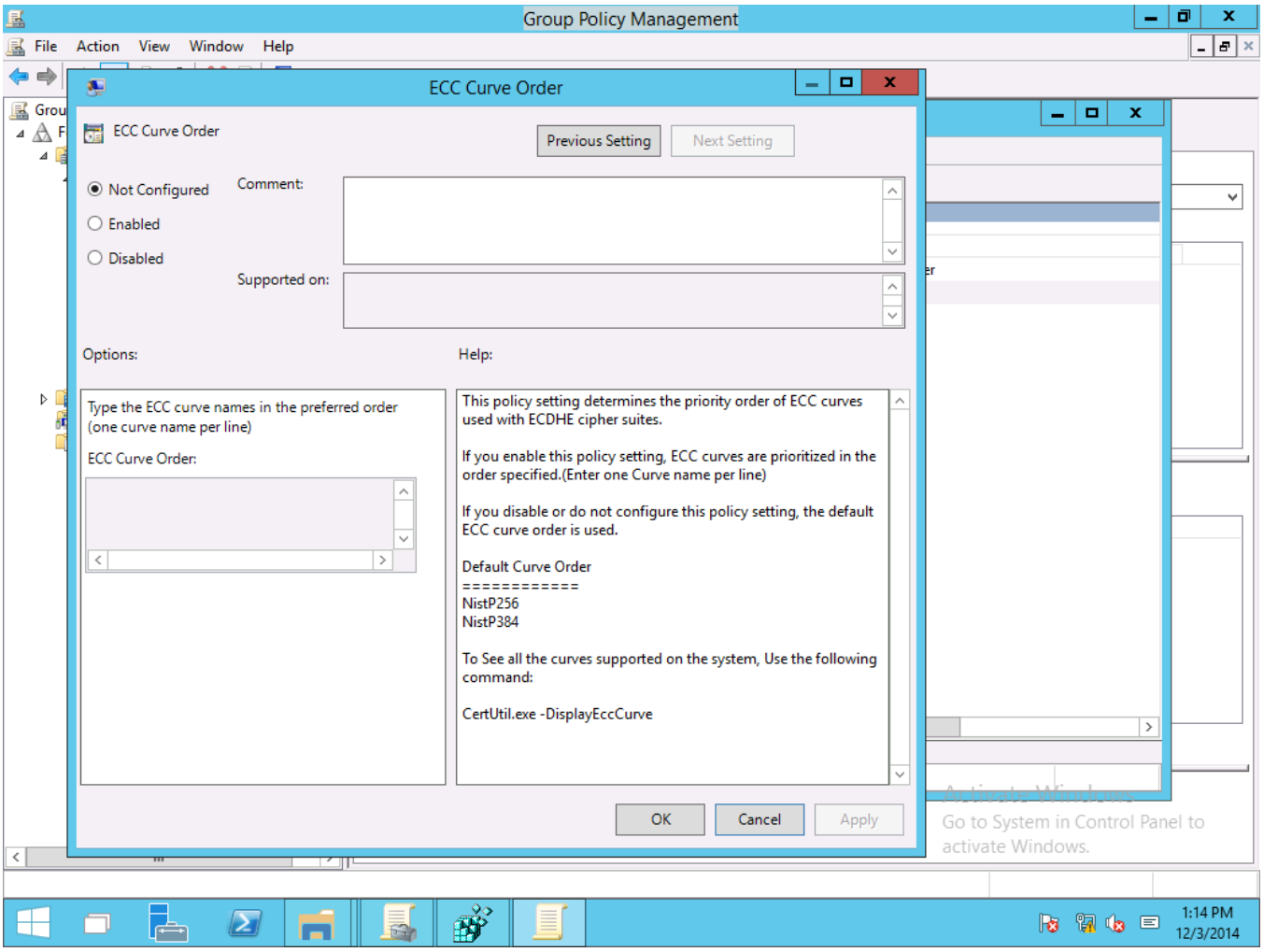
Organizations can distribute curve parameters to enterprise, domain-joined, computer using Group Policy and the Group Policy Preferences Registry extension. The process for distributing a curve is:

1. Use **certutil.exe** to add a new registered named curve.
2. From that same computer, Open the Group Policy Management Console (GPMC), create a new Group Policy object, and edit it.
3. Navigate to **Computer Configuration|Preferences|Windows Settings|Registry**. Right-click **Registry**. Hover over **New** and select **Collection Item**. Rename the collection item to match the name of the curve. You create one Registry Collection item for each registry key under *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Cryptography\ECCParameters*.
4. Configure the newly created Group Policy Preference Registry Collection by adding a new **Registry Item** for each registry value listed under *HKEY_LOCAL_MACHINE\System\CurrentControlSet\Control\Cryptography\ECCParameters[curveName]*.
5. Deploy the Group Policy object containing Group Policy Registry Collection item computers that should receive the new named curves.



Manage TLS ECC order

Beginning with Windows 10 and Windows Server 2016, ECC Curve Order group policy settings can be used to configure the default TLS ECC Curve Order. Organizations can add their own trusted named curves to the operating system, and then add those named curves to the curve priority Group Policy setting to ensure they're used in future TLS handshakes. New curve priority lists become active on the next reboot after receiving the policy settings.


Group Policy Management
File Action View Window Help
ECC Curve Order
ECC Curve Order
Previous Setting
Next Setting
Not Configured
Enabled
Disabled
Comment:
Supported on:
Options:
Help:
Type the ECC curve names in the preferred order (one curve name per line)
ECC Curve Order:
This policy setting determines the priority order of ECC curves used with ECDHE cipher suites.
If you enable this policy setting, ECC curves are prioritized in the order specified.(Enter one Curve name per line)
If you disable or do not configure this policy setting, the default ECC curve order is used.
Default Curve Order
NistP256
NistP384
To See all the curves supported on the system, Use the following command:
CertUtil.exe -DisplayEccCurve
OK
Cancel
Apply
Go to System in Control Panel to activate Windows.
1:14 PM
12/3/2014

Transport Layer Security (TLS) registry settings

Article • 04/18/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This article explains the supported registry setting information for the Windows implementation of the Transport Layer Security (TLS) protocol and the Secure Sockets Layer (SSL) protocol through the Schannel Security Support Provider (SSP). The registry subkeys and entries covered in this article help you administer and troubleshoot the Schannel SSP, specifically the TLS and SSL protocols.

> **⊗ Caution**
>
> This information is provided as a reference to use when you're troubleshooting or verifying that the required settings are applied. We recommend that you don't directly edit the registry unless there's no other alternative. Modifications to the registry aren't validated by the Registry Editor or by the Windows operating system before they're applied. As a result, incorrect values can be stored, and this can result in unrecoverable errors in the system. When possible, instead of editing the registry directly, use Group Policy or other Windows tools such as the Microsoft Management Console (MMC). If you must edit the registry, use extreme caution.

Schannel logging

There are eight logging levels for Schannel events saved to the system event log and viewable using Event Viewer. This registry path is stored in `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL` under the **EventLogging** key with a DWORD value set to *1*.

Expand table

Decimal or Hex	Schannel logging events
0	No events
1	Error events
2	Warning events
3	Error and Warning events

Decimal or Hex	Schannel logging events
4	Informational and Success events
5	Error, Informational, and Success events
6	Warning, Informational, and Success events
7	Error, Warning, Informational, and Success events

> **Note**
>
> You must reboot your device after changing the logging level.

CertificateMappingMethods

When a server application requires client authentication, Schannel automatically attempts to map the certificate that is supplied by the client computer to a user account. You can authenticate users who sign in with a client certificate by creating mappings, which relate the certificate information to a Windows user account.

After you create and enable a certificate mapping, each time a client presents a client certificate, your server application automatically associates that user with the appropriate Windows user account.

In most cases, a certificate is mapped to a user account in one of two ways:

- A single certificate is mapped to a single user account (one-to-one mapping).
- Multiple certificates are mapped to one user account (many-to-one mapping).

The Schannel provider uses four certificate mapping methods:

1. Kerberos service-for-user (S4U) mapping (enabled by default)
2. User principal name mapping
3. One-to-one mapping (also known as subject/issuer mapping)
4. Many-to-one mapping

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

Expand table

Entry name	DWORD	Enabled by default
Subject/Issuer	0x000000001	No

Entry name	DWORD	Enabled by default
Issuer	0x000000002	No
UPN	0x000000004	No
S4U2Self	0x000000008	Yes
S4U2Self Explicit	0x000000010	Yes

Ciphers

TLS/SSL ciphers should be controlled by configuring the cipher suite order. For details, see Configuring TLS Cipher Suite Order.

For information about default cipher suite orders that are used by the Schannel SSP, see Cipher Suites in TLS/SSL (Schannel SSP).

CipherSuites

Configuring TLS/SSL cipher suites should be done using group policy, MDM, or PowerShell, see Configuring TLS Cipher Suite Order for details.

For information about default cipher suite orders that are used by the Schannel SSP, see Cipher Suites in TLS/SSL (Schannel SSP).

ClientCacheTime

This entry specifies client TLS session cache item lifetime in milliseconds. Beginning with Windows Server 2008 and Windows Vista the default is 10 hours. A value of *0* turns off TLS session caching on the client.

The first time a client connects to a server through the Schannel SSP, a full TLS/SSL handshake is performed. When complete, the master secret, cipher suite, and certificates are stored in the session cache on the respective client and server.

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

EnableOcspStaplingForSni

Online Certificate Status Protocol (OCSP) stapling enables a web server, such as Internet Information Services (IIS), to provide the current revocation status of a server certificate when it

sends the server certificate to a client during the TLS handshake. This feature reduces the load on OCSP servers because the web server can cache the current OCSP status of the server certificate and send it to multiple web clients. Without this feature, each web client would try to retrieve the current OCSP status of the server certificate from the OCSP server. This would generate a high load on that OCSP server.

In addition to IIS, web services over http.sys can also benefit from this setting, including Active Directory Federation Services (AD FS) and Web Application Proxy (WAP).

By default, OCSP support is enabled for IIS websites that have a simple secure (SSL/TLS) binding. However, this support isn't enabled by default if the IIS website is using either or both of the following types of SSL/TLS bindings:

- Require Server Name Indication
- Use Centralized Certificate Store

In this case, the server hello response during the TLS handshake doesn't include an OCSP stapled status by default. This behavior improves performance: The Windows OCSP stapling implementation scales to hundreds of server certificates. However, Server Name Indication (SNI) and Central Certificate Store (CCS) enable IIS to scale to thousands of websites that potentially have thousands of server certificates, therefore enabling OCSP stapling for CCS bindings might cause performance issues.

Registry path:
`HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

Add the following key:

```
"EnableOcspStaplingForSni"=dword:00000001
```

To disable, set the DWORD value to 0:

```
"EnableOcspStaplingForSni"=dword:00000000
```

> ⓘ **Note**
>
> Enabling this registry key has potential performance impact.

Hashes

TLS/SSL hash algorithms should be controlled by configuring the cipher suite order. See Configuring TLS cipher suite order for details.

IssuerCacheSize

This entry controls the size of the issuer cache, and it's used with issuer mapping. The Schannel SSP attempts to map all of the issuers in the client's certificate chain, not just the direct issuer of the client certificate. When the issuers don't map to an account, which is the typical case, the server might attempt to map the same issuer name repeatedly, hundreds of times per second.

To prevent this, the server has a negative cache, so if an issuer name doesn't map to an account, it's added to the cache and the Schannel SSP doesn't attempt to map the issuer name again until the cache entry expires. This registry entry specifies the cache size. This entry doesn't exist in the registry by default. The default value is *100*.

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

IssuerCacheTime

This entry controls the length of the cache time-out interval in milliseconds. The Schannel SSP attempts to map all of the issuers in the client's certificate chain, not just the direct issuer of the client certificate. In the case where the issuers don't map to an account, which is the typical case, the server might attempt to map the same issuer name repeatedly, hundreds of times per second.

To prevent this, the server has a negative cache, so if an issuer name doesn't map to an account, it's added to the cache and the Schannel SSP doesn't attempt to map the issuer name again until the cache entry expires. This cache is kept for performance reasons, so that the system doesn't continue trying to map the same issuers. This entry doesn't exist in the registry by default. The default value is 10 minutes.

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

KeyExchangeAlgorithm key sizes

These following entries might not exist in the registry by default and must be manually created. Use of key exchange algorithms should be controlled by configuring the cipher suite order. To learn more about TLS/SSL cipher suite cryptographic algorithms, see Cipher suites in TLS/SSL (Schannel SSP).

Diffie-Hellman

Added in Windows 10, version 1507, and Windows Server 2016.

Registry path: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\KeyExchangeAlgorithms\Diffie-Hellman`

To specify a minimum supported range of Diffie-Hellman key bit length for the TLS client, create a `ClientMinKeyBitLength` entry. After you create the entry, change the DWORD value to the desired bit length. If not configured, 1024 bits is the minimum.

> ⓘ **Note**
>
> Configured elliptic curves determine the cryptographic strength of the ECDHE key exchange. For more information, see Manage Transport Layer Security (TLS).

MaximumCacheSize

This entry controls the maximum number of TLS sessions to cache. Setting MaximumCacheSize to `0` disables the server-side session cache to prevent session resumption. Increasing MaximumCacheSize above the default values causes Lsass.exe to consume extra memory. Each session-cache element typically requires 2 KB to 4 KB of memory. This entry doesn't exist in the registry by default. The default value is 20,000 elements.

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

Messaging – fragment parsing

This entry controls the maximum allowed size of a TLS handshake message that's accepted. Messages larger than the allowed size aren't accepted and the TLS handshake fails. These entries don't exist in the registry by default.

When you set the value to `0x0`, fragmented messages aren't processed and causes the TLS handshake to fail. This makes TLS clients or servers on the current machine noncompliant with the TLS RFCs.

The maximum allowed size can be increased up to 2^16 bytes. Allowing a client or server to read and store large amounts of unverified data from the network isn't a good idea and consumes extra memory for each security context.

Added in Windows 7 and Windows Server 2008 R2: An update that enables Internet Explorer in Windows XP, in Windows Vista, or in Windows Server 2008 to parse fragmented TLS/SSL handshake messages is available.

Registry path: `HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Messaging`

To specify a maximum allowed size of fragmented TLS handshake messages that the TLS client accepts, create a `MessageLimitClient` entry. After you create the entry, change the DWORD value to the desired bit length. If not configured, the default value is `0x8000` bytes.

To specify a maximum allowed size of fragmented TLS handshake messages that the TLS server accepts when there's no client authentication, create a `MessageLimitServer` entry. After you create the entry, change the DWORD value to the desired bit length. If not configured, the default value is *0x4000* bytes.

To specify a maximum allowed size of fragmented TLS handshake messages that the TLS server accepts when there's client authentication, create a `MessageLimitServerClientAuth` entry. After you create the entry, change the DWORD value to the desired bit length. If not configured, the default value is *0x8000* bytes.

SendTrustedIssuerList

TLS servers might send a list of the distinguished names of acceptable certificate authorities when requesting client authentication. This can help TLS clients select an appropriate TLS client certificate. Schannel-based TLS servers don't send this trusted issuer list by default because it exposes the certificate authorities trusted by the server to passive observers and also increases the amount of data exchanged in the course of the TLS handshake. Setting this value to *1* causes Schannel-based servers to send their lists of trusted issuers.

Not sending a list of trusted issuers might affect what the client sends when it's asked for a client certificate. For example, when Microsoft Edge receives a request for client authentication, it only displays the client certificates that chain up to one of the certification authorities that's sent by the server. If the server didn't send a list, Microsoft Edge displays all of the client certificates that are installed on the client.

This behavior might be desirable. For example, when PKI environments include cross certificates, the client and server certificates doesn't have the same root CA. Therefore, Microsoft Edge can't choose a certificate that chains up to one of the server's CAs. TLS clients might offer any available client certificate when a server doesn't send the trusted issuer list. This entry doesn't exist in the registry by default.

Default Send Trusted Issuer List behavior

Expand table

Windows version	Default behavior
Windows Server 2012, Windows 8, and later	FALSE

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

ServerCacheTime

This entry specifies server TLS session cache item lifetime in milliseconds. The default is 10 hours. A value of *0* turns off TLS session caching on the server and prevents session resumption. Increasing ServerCacheTime above the default values causes Lsass.exe to consume additional memory. Each session cache element typically requires 2 KB to 4 KB of memory. This entry doesn't exist in the registry by default.

Registry path: `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL`

Default server cache time: 10 hours

TLS, DTLS, and SSL protocol version settings

SChannel SSP implements versions of the TLS, DTLS, and SSL protocols. Different Windows releases support different protocol versions. The set of (D)TLS and SSL versions available system-wide can be restricted (but not expanded) by SSPI callers specifying the SCH_CREDENTIALS structure in the AcquireCredentialsHandle call. It's recommended that SSPI callers use the system defaults, rather than imposing protocol version restrictions.

A supported (D)TLS or SSL protocol version can exist in one of the following states:

- **Enabled**: Unless the SSPI caller explicitly disables this protocol version using SCH_CREDENTIALS structure, Schannel SSP might negotiate this protocol version with a supporting peer.
- **Disabled**: Schannel SSP doesn't negotiate this protocol version regardless of the settings the SSPI caller might specify.

These registry values are configured separately for the protocol client and server roles under the registry subkeys named using the following format:

`<SSL/TLS/DTLS> <major version number>.<minor version number><Client\Server>`

These version-specific subkeys can be created under the following registry path:

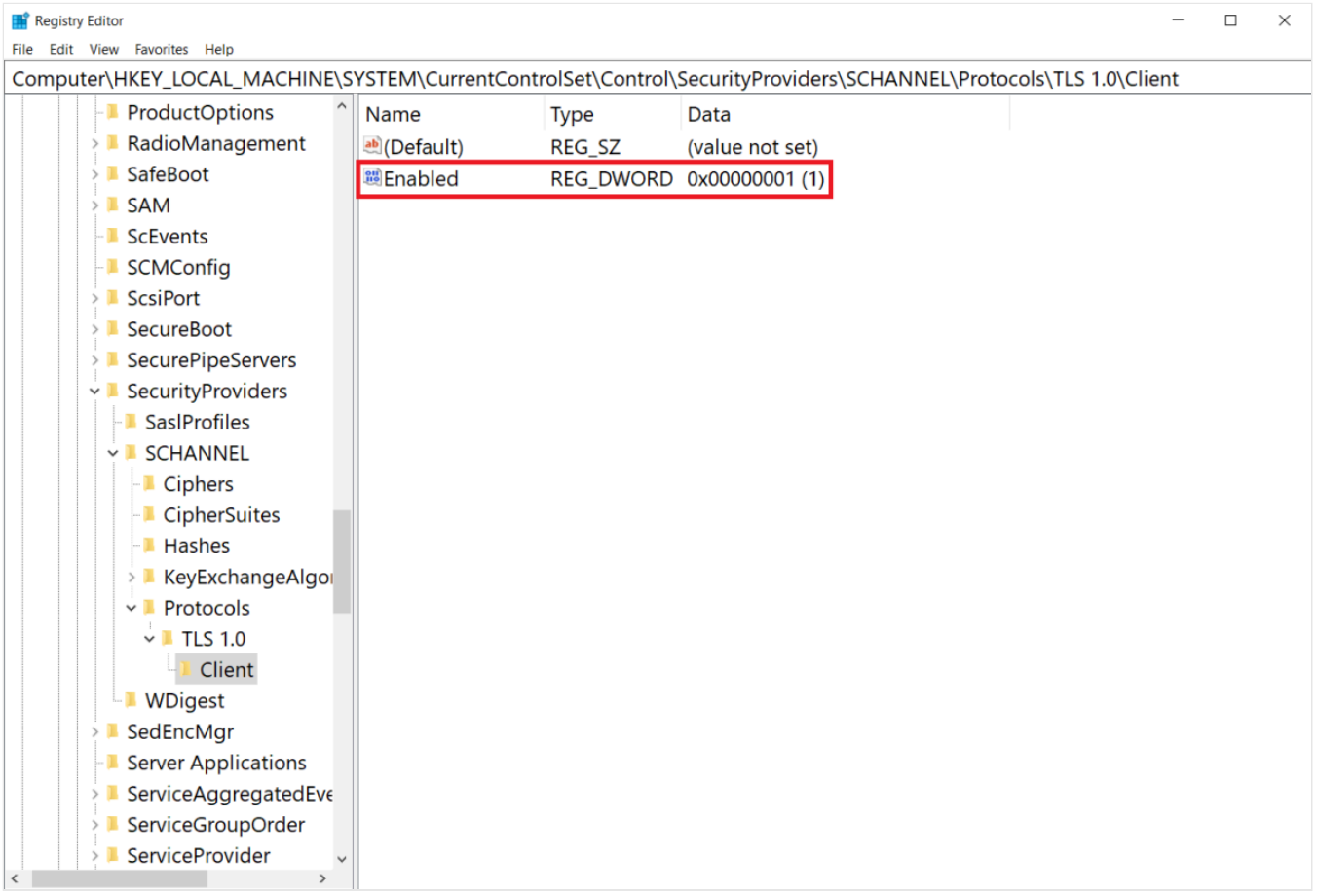
- `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols`

For example, here are some valid registry paths with version-specific subkeys:

- `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\SSL 3.0\Client`
- `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\TLS 1.2\Server`
- `HKLM SYSTEM\CurrentControlSet\Control\SecurityProviders\SCHANNEL\Protocols\DTLS 1.2\Client`

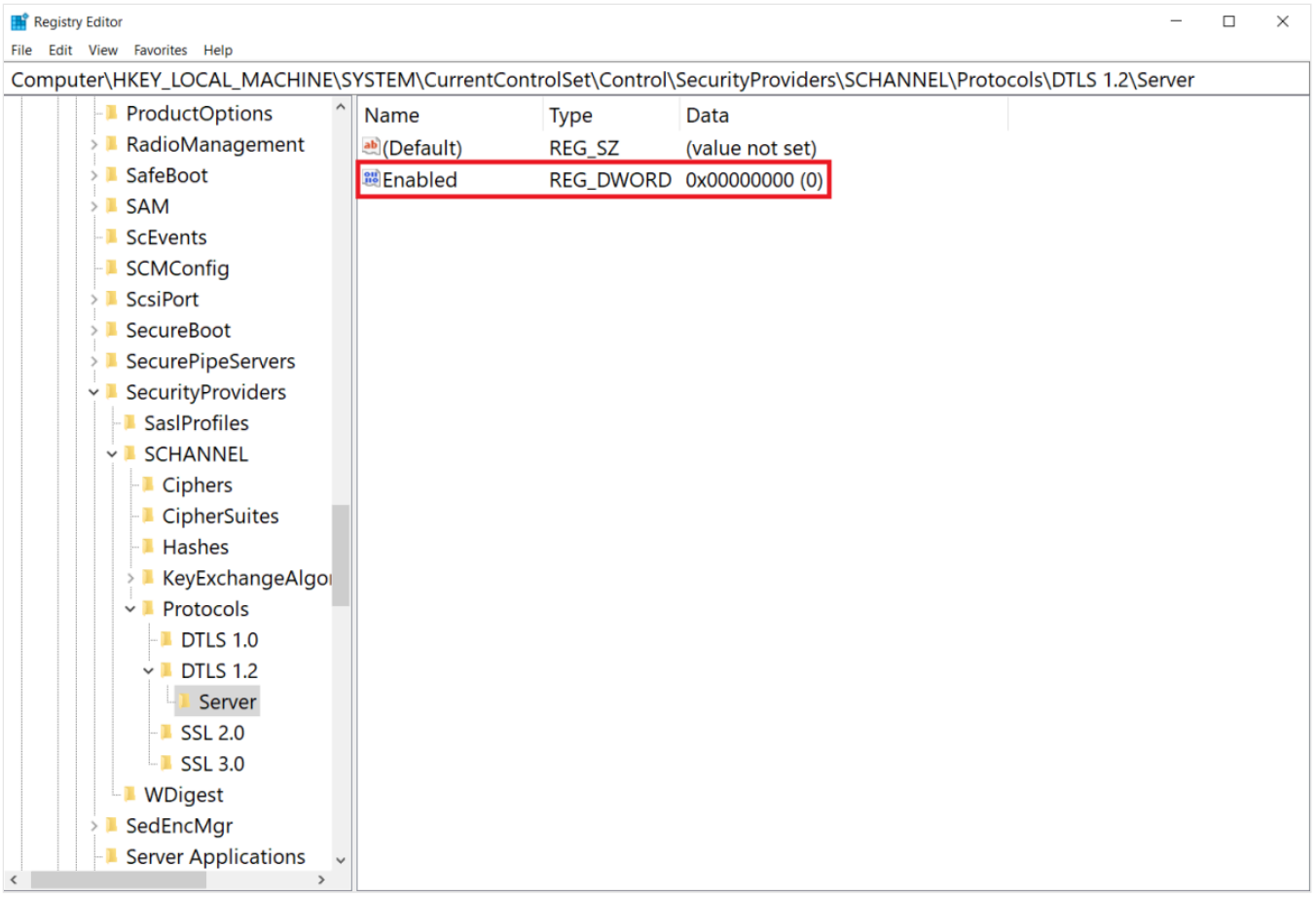
In order to override a system default and set a supported (D)TLS or SSL protocol version to the `Enabled` state, create a DWORD registry value named `Enabled` with an entry value of "**1**" under the corresponding version-specific subkey.

The following example shows TLS 1.0 client set to the **Enabled** state:



In order to override a system default and set a supported (D)TLS or SSL protocol version to the `Disabled` state, change the DWORD registry value of `Enabled` to "**0**" under the corresponding version-specific subkey.

The following example shows DTLS 1.2 disabled in the registry:



Switching a (D)TLS or SSL protocol version to `Disabled` state might cause AcquireCredentialsHandle calls to fail due to the lack of protocol versions enabled system-wide and at the same time allowed by particular SSPI callers. In addition, reducing the set of `Enabled` (D)TLS and SSL versions could break interoperability with remote peers.

Once the (D)TLS or SSL protocol version settings are modified, they take effect on connections established using credential handles opened by subsequent AcquireCredentialsHandle calls. (D)TLS and SSL client and server applications and services tend to reuse credential handles for multiple connections, for performance reasons. In order to get these applications to reacquire their credential handles, an application, or service restart might be required.

These registry settings only apply to Schannel SSP and don't affect any third-party (D)TLS and SSL implementations that might be installed on the system.

> ⚠ **Warning**
>
> Attempting to create or adjust any Schannel registry settings that aren't explicitly detailed in this article isn't recommended due to potential risks and unintended consequences that may arise from unsupported configurations.

To learn about managing the TLS cipher suite using PowerShell, see TLS command reference. If interested in managing TLS settings via Group Policy, see Configure TLS cipher suite order by

using Group Policy.

Schannel Security Support Provider Technical Reference

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This reference for IT professionals contains information about the Transport Layer Security (TLS) protocol, the Secure Sockets Layer (SSL) protocol, and the Datagram Transport Layer Security (DTLS) protocol as implemented by the Schannel Security Support Provider (SSP).

These protocols provide a means to secure data that is being sent between applications across an untrusted network by using certificate-based authentication and symmetric encryption keys.

- Transport Layer Security protocol
- Datagram Transport Layer Security protocol

Related links

Windows Authentication Kerberos Authentication

Feedback

Was this page helpful? Yes No

Transport Layer Security protocol

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016, ✅ Windows 11, ✅ Windows 10

This topic for the IT professional describes how the Transport Layer Security (TLS) protocol works and provides links to the IETF RFCs for TLS 1.0, TLS 1.1, and TLS 1.2.

> **Note**
>
> In a future release of Windows Server, TLS 1.0 and 1.1 will be disabled by default. For more information, see TLS versions 1.0 and 1.1 disablement resources.

The TLS (and SSL) protocols are located between the application protocol layer and the TCP/IP layer, where they can secure and send application data to the transport layer. Because the protocols work between the application layer and the transport layer, TLS and SSL can support multiple application layer protocols.

TLS and SSL assume that a connection-oriented transport, typically TCP, is in use. The protocol allows client and server applications to detect the following security risks:

- Message tampering
- Message interception
- Message forgery

The TLS and SSL protocols can be divided into two layers. The first layer consists of the application protocol and the three handshaking protocols: the handshake protocol, the change cipher spec protocol, and the alert protocol. The second layer is the record protocol.

TLS and SSL protocol layers

The Schannel SSP implements the TLS and SSL protocols without modification. The SSL protocol is proprietary, but the Internet Engineering Task Force produces the public TLS specifications. For information about which TLS or SSL version is supported in Windows versions, see Protocols in TLS/SSL (Schannel SSP). Each specification contains information about:

- The TLS Record Protocol
- The TLS Handshaking Protocols: - Change cipher spec protocol - Alert protocol

- Cryptographic Computations
- Mandatory Cipher Suites
- Application Data Protocol

RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2

RFC 4346 - The Transport Layer Security (TLS) Protocol Version 1.1

RFC 2246 - The TLS Protocol Version 1.0

TLS session resumption

Introduced in Windows Server 2012 R2 , the Schannel SSP implemented the server-side portion of TLS session resumption. The client-side implementation of RFC 5077 was added in Windows 8.

Devices that connect TLS to servers frequently need to reconnect. TLS session resumption reduces the cost of establishing TLS connections because resumption involves an abbreviated TLS handshake. This facilitates more resumption attempts by allowing a group of TLS servers to resume each other's TLS sessions. This modification provides the following savings for any TLS client that supports RFC 5077, including Windows Phone and Windows RT devices:

- Reduced resource usage on the server
- Reduced bandwidth, which improves the efficiency of client connections
- Reduced time spent for the TLS handshake due to resumptions of the connection

For information about stateless TLS session resumption, see the IETF document RFC 5077.

Application protocol negotiation

Windows Server 2012 R2 and Windows 8.1 introduced support that allows client-side TLS application protocol negotiation. Applications can leverage protocols as part of the HTTP 2.0 standard development, and users can access online services such as Google and Twitter by using apps running the SPDY protocol.

For information about how application protocol negotiation works, see Transport Layer Security (TLS) Application Layer Protocol Negotiation Extension.

TLS support for Server Name Indication extensions

The Server Name Indication (SNI) feature extends the SSL and TLS protocols to allow proper identification of the server when numerous virtual images are running on a single server. In a virtual hosting scenario, several domains (each with its own potentially distinct certificate) are hosted on one server. In this case, the server has no way of knowing beforehand which certificate to send to the client. SNI allows the client to inform the target domain earlier in the protocol, and this allows the server to correctly select the proper certificate.

This provides the following additional functionality:

- Allows you to host multiple SSL websites on a single Internet Protocol and port combination
- Reduces the memory usage when multiple SSL websites are hosted on a single web server
- Allows more users to connect to SSL websites simultaneously

---

Feedback

Was this page helpful? Yes No

Datagram Transport Layer Security protocol

Article • 02/14/2023

Windows Server 2016, Windows 10

This reference topic for the IT professional describes the Datagram Transport Layer Security (DTLS) protocol, which is part of the Schannel Security Support Provider (SSP).

Introduced in the Schannel SSP in Windows Server 2012 and Windows 8, the DTLS protocol provides communication privacy for datagram protocols. For information about which DTLS version is supported in Windows versions, see Protocols in TLS/SSL (Schannel SSP). The protocol allows client and server applications to communicate in a way that is designed to prevent eavesdropping, tampering, or message forgery. The DTLS protocol is based on the Transport Layer Security (TLS) protocol, and it provides equivalent security guarantees, reducing the need to use IPsec or designing a custom application layer security protocol.

Datagrams are common in streaming media, such as gaming or secured video conferencing. Developers can develop applications to use the DTLS protocol within the context of the Windows authentication Security Support Provider Interface (SSPI) model to secure the communication between clients and servers. The DTLS protocol is built on top of the User Datagram Protocol (UDP). DTLS is designed to be as similar to TLS as possible to minimize new security invention and to maximize the amount of code and infrastructure reuse.

The cipher suites that are available for configuration are patterned after those you can configure for TLS. RC4 is not permitted. Schannel continues to use Cryptography Next Generation (CNG). This takes advantage of FIPS 140 certification, which was introduced in Windows Vista.

Additional References

IETF RFC 4347 Datagram Transport Layer Security

How User Account Control Works

Article • 07/29/2021 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

User Account Control (UAC) helps prevent malicious programs (also called malware) from damaging a computer and helps organizations deploy a better-managed desktop. With UAC, applications and tasks always run in the security context of a non-administrator account, unless an administrator specifically authorizes administrator-level access to the system. UAC can block the automatic installation of unauthorized applications and prevent inadvertent changes to system settings.

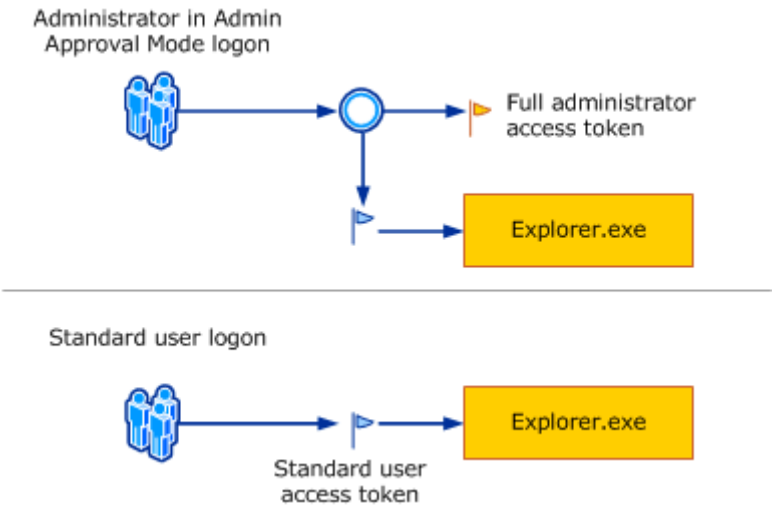
UAC Process and Interactions

Each application that requires the administrator access token must prompt the administrator for consent. The one exception is the relationship that exists between parent and child processes. Child processes inherit the user access token from the parent process. Both the parent and child processes, however, must have the same integrity level. Windows Server 2012 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, while a "low" integrity application is one that performs tasks that could potentially compromise the operating system, such as a Web browser. Applications with lower integrity levels cannot modify data in applications with higher integrity levels. When a standard user attempts to run an application that requires an administrator access token, UAC requires that the user provide valid administrator credentials.

In order to better understand how this process happens it is important to review the details of the Windows Server 2012 logon process.

Windows Server 2012 Logon Process

The following illustration demonstrates how the logon process for an administrator differs from the logon process for a standard user.



By default, standard users and administrators access resources and run applications in the security context of standard users. When a user logs on to a computer, 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. The standard user access token is used to start applications that do not perform administrative tasks (standard user applications). The standard user access token is then used to display the desktop (Explorer.exe). Explorer.exe is the parent process from which all other user-initiated processes inherit their access token. As a result, all applications run as a standard user unless a user provides consent or credentials to approve an application to use a full administrative access token.

A user that is a member of the Administrators group can log on, 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 Server 2012 automatically prompts the user for approval. This prompt is called an elevation prompt, and its behavior can be configured by using the Local Security Policy snap-in (Secpol.msc) or Group Policy.

> ⓘ **Note**
>
> The term "elevate" is used to refer to the process in Windows Server 2012 that prompts the user for consent or credentials to use a full administrator access token.

The UAC User Experience

When UAC is enabled, the user experience for standard users is different from that of administrators in Admin Approval Mode. The recommended and more secure method of running Windows Server 2012 is to make your primary user account a standard user account. 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.

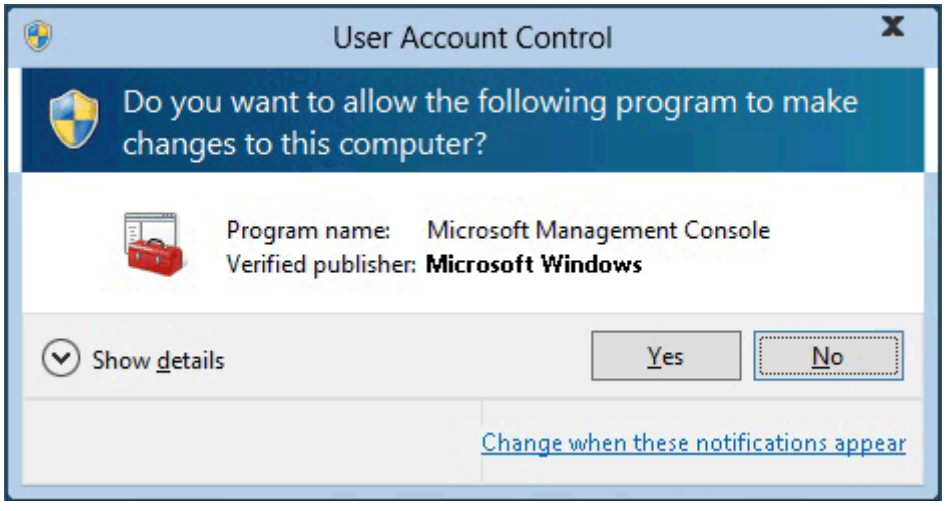
The alternative to running as a standard user is to run as an administrator in Admin Approval Mode. With the built-in UAC elevation component, members of the local Administrators group can easily perform an administrative task by providing approval. The default, built-in UAC elevation component for an administrator account in Admin Approval Mode is called the consent prompt. The UAC elevation prompting behavior can be configured by using the Local Security Policy snap-in (Secpol.msc) or Group Policy.

The consent and credential prompts

With UAC enabled, Windows Server 2012 prompts for consent or prompts for credentials of a valid local administrator account before starting a program or task that requires a full administrator access token. This prompt ensures that no malicious software can be silently installed.

The consent prompt

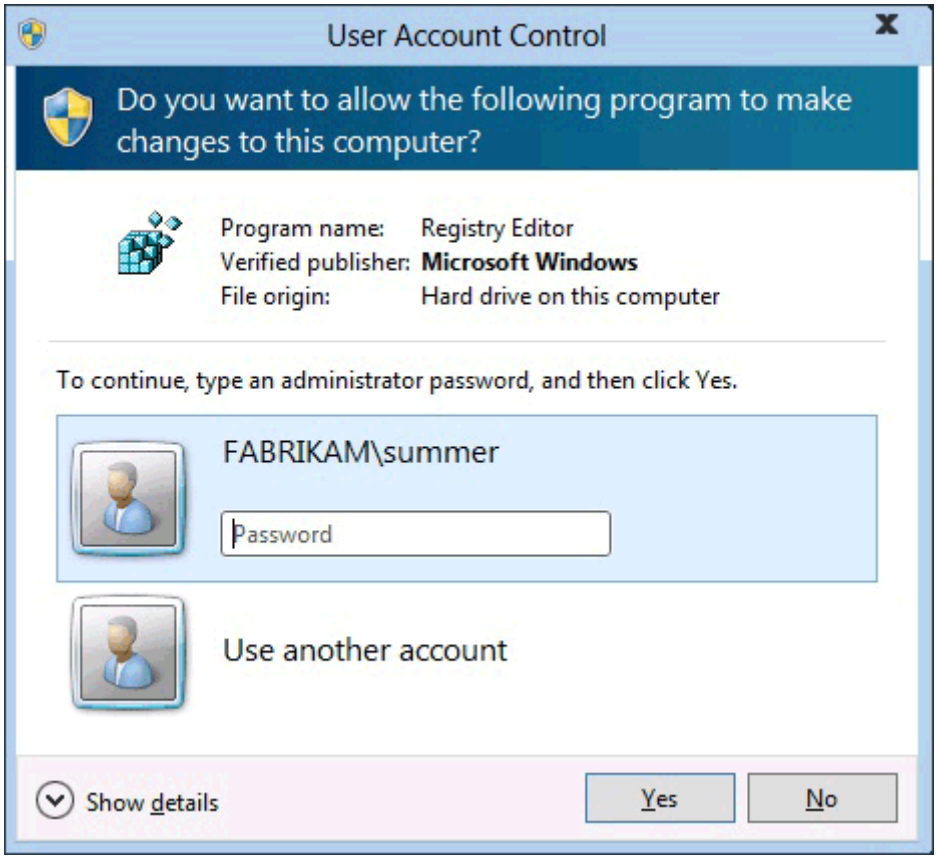
The consent prompt is presented when a user attempts to perform a task that requires a user's administrative access token. The following is a screenshot of the UAC consent prompt.



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. This standard user default prompt behavior can be configured by using the Local Security Policy snap-in (Secpol.msc) or Group Policy. 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 following screenshot is an example of the UAC credential prompt.



UAC elevation prompts

The UAC elevation prompts are color-coded to be application-specific, enabling for immediate identification of an application's potential security risk. When an application attempts to run with an administrator's full access token, Windows Server 2012 first analyzes the executable file to determine its publisher. Applications are first separated into three categories based on the executable file's publisher: Windows Server 2012 , publisher verified (signed), and publisher not verified (unsigned). The following diagram illustrates how Windows Server 2012 determines which color elevation prompt to present to the user.

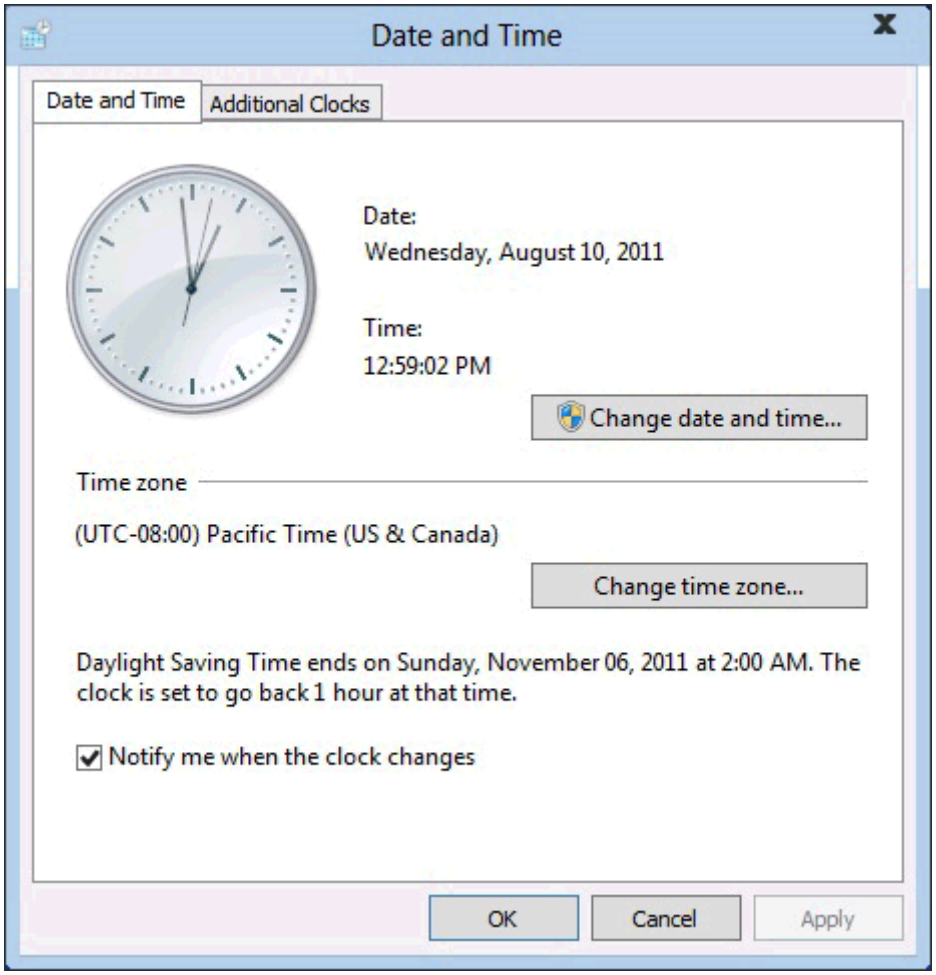
The elevation prompt color-coding is as follows:

- Red background with a red shield icon: The application is blocked by Group Policy or is from a publisher that is blocked.

- Blue background with a blue and gold shield icon: The application is a Windows Server 2012 administrative application, such as a Control Panel item.
- Blue background with a blue shield icon: The application is signed by using Authenticode and is trusted by the local computer.
- Yellow background with a yellow shield icon: The application is unsigned or signed but is not yet trusted by the local computer.

Shield icon

Some Control Panel items, such as **Date and Time Properties**, 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 Properties** Control Panel item.



The shield icon on the **Change date and time** button indicates that the process requires a full administrator access token and will display a UAC elevation prompt.

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 in Windows Server 2012 . 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.

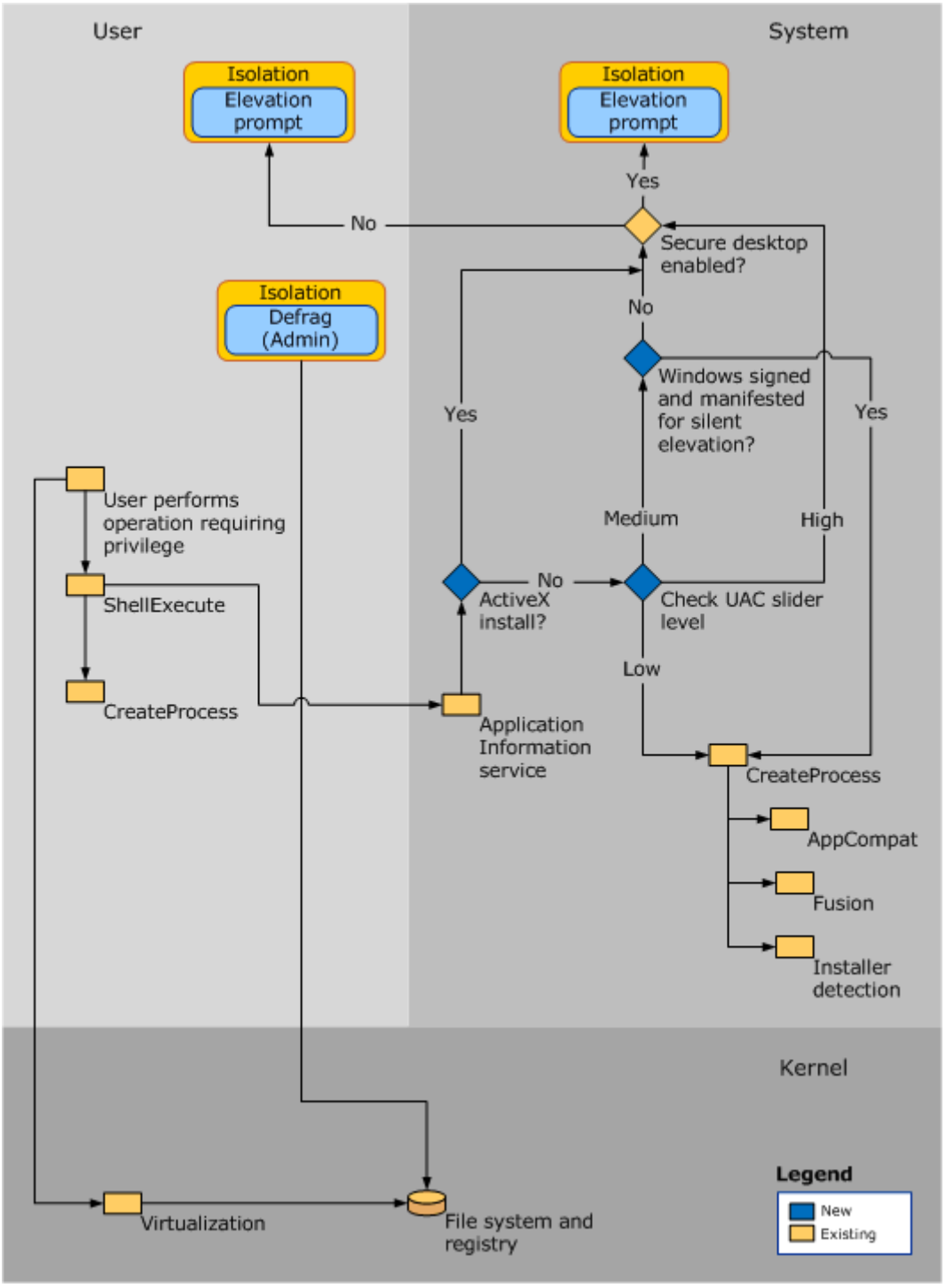
When an executable file requests elevation, the interactive desktop, also called the user desktop, is switched to the secure desktop. The secure desktop dims the user desktop and displays an elevation prompt that must be responded to before continuing. When the user clicks Yes or No, the desktop switches back to the user desktop.

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 does not gain elevation if the user clicks Yes on the imitation. If the policy setting is set to Prompt for credentials, malware imitating the credential prompt may be able to gather the credentials from the user. However, the malware does not 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 cannot occur unless a user previously installed the malware on the computer. Because processes requiring an administrator access token cannot silently install when UAC is enabled, the user must explicitly provide consent by clicking **Yes** or by providing administrator credentials. The specific behavior of the UAC elevation prompt is dependent upon Group Policy.

UAC Architecture

The following diagram details the UAC architecture.



To better understand each component, review the table below:

Expand table

Component	Description
User	
User performs operation requiring privilege	If the operation changes the file system or registry, Virtualization is called. All other operations call ShellExecute.

Component	Description
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 applications that require one or more elevated privileges or user rights to run, such as local administrative tasks, and applications that require higher integrity levels. The Application Information service helps start such applications by creating a new process for the application with an administrative user's full access token when elevation is required and (depending on Group Policy) consent is given by the user to do so.
Elevating an ActiveX install	If ActiveX is not 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 now has four levels of notification to choose from and a slider to use to select the notification level: • High If the slider is set to **Always notify**, the system checks whether the secure desktop is enabled. • Medium If the slider is set to **Default-Notify me only when programs try to make changes to my computer**, the **User Account Control: Only elevate executable files that are signed and validated** policy setting is checked: ◦ If the policy setting is enabled, the public key infrastructure (PKI) certification path validation is enforced for a given executable file before it is permitted to run. ◦ If the policy setting is not enabled (default), the PKI certification path validation is not enforced before a given executable file is permitted to run. The **User Account Control: Switch to the secure desktop when prompting for elevation** Group Policy setting is checked. • Low If the slider is set to **Notify me only when programs try to make changes to my computer (do not dim by desktop)**, the CreateProcess is called. • Never Notify If the slider is set to **Never notify me when**, UAC prompt will never notify when a program is trying to install or trying to make any change on the computer. **Important:** This setting is not recommended. This

Component	Description
	setting is the same as setting the **User Account Control: Behavior of the elevation prompt for administrators in Admin Approval Mode** policy setting to **Elevate without prompting**.
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 is not 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 application requires elevation. The executable file is then inspected to determine its requested execution level, which is stored in the application manifest for the executable file. CreateProcess fails if the requested execution level specified in the manifest does not 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 executable files, which helps prevent installations from being run without the user's knowledge and consent.
Kernel	
Virtualization	Virtualization technology ensures that non-compliant applications do not silently fail to run or fail in a way that the cause cannot 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.

There is a change on Windows Server 2012 UAC from previous Windows versions. The new slider will never turn UAC completely off. The new setting will:

- Keep the UAC service running.

- Cause all elevation request initiated by administrators to be auto-approved 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**.

> ⚠ **Warning**
>
> Tailored Applications will not work on Windows Server 2012 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, IT administrators do not need to replace the majority of applications when running Windows Server 2012 with UAC enabled.

Windows Server 2012 includes file and registry virtualization technology for applications that are not UAC compliant and that require an administrator's access token to run correctly. Virtualization ensures that even applications that are not UAC compliant are compatible with Windows Server 2012 . When an administrative application that is not UAC compliant attempts to write to a protected directory, such as Program Files, UAC gives the application its own virtualized view of the resource it is 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 non-compliant application.

Most application tasks operate properly by using virtualization features. Although virtualization allows a majority of applications to run, it is a short-term fix and not a long-term solution. Application developers should modify their applications to be compliant with the Windows Server 2012 logo program as soon as possible, rather than relying on file, folder, and registry virtualization.

Virtualization is not in option in the following scenarios:

1. Virtualization does not apply to applications that are elevated and run with a full administrative access token.

2. Virtualization supports only 32-bit applications. Non-elevated 64-bit applications simply receive an access denied message when they attempt to acquire a handle (a unique identifier) to a Windows object. Native Windows 64-bit applications are required to be compatible with UAC and to write data into the correct locations.

3. Virtualization is disabled for an application if the application includes an application manifest with a requested execution level attribute.

Request Execution Levels

An application manifest is an XML file that describes and identifies the shared and private side-by-side assemblies that an application should bind to at run time. In Windows Server 2012 , the application manifest includes entries for UAC application compatibility purposes. Administrative applications that include an entry in the application manifest prompt the user for permission to access the user's access token. Although they lack an entry in the application manifest, most administrative applications can run without modification by using application compatibility fixes. Application compatibility fixes are database entries that enable applications that are not UAC compliant to work properly with Windows Server 2012 .

All UAC-compliant applications should have a requested execution level added to the application manifest. If the application requires administrative access to the system, then marking the application with a requested execution level of "require administrator" ensures that the system identifies this program as an administrative application and performs the necessary elevation steps. Requested execution levels specify the privileges required for an application.

Installer Detection Technology

Installation programs are applications 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 do not have sufficient access to install programs. Windows Server 2012 heuristically detects installation programs and requests administrator credentials or approval from the administrator user in order to run with access privileges. Windows Server 2012 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

- 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 is an installer:

- The 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.
- There are targeted sequences of bytes within the executable file.

> **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 setting must be enabled for installer detection to detect installation programs. This setting is enabled by default and can be configured locally by using the Local Security Policy snap-in (Secpol.msc) or configured for the domain, OU, or specific groups by Group Policy (Gpedit.msc).

Feedback

Was this page helpful? Yes No

Introducing Token Binding

Article • 05/12/2025 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

The Token Binding protocol allows applications and services to cryptographically bind their security tokens to the TLS layer to mitigate token theft and replay attacks. The long-lived, uniquely identifiable TLS [RFC5246] bindings can span multiple TLS sessions and connections.

Version support:

- Windows 10, version 1507 – Off by default
  - Token Binding Protocol added [draft-ietf-tokbind-protocol-01]
  - WinInet & HTTP.SYS support of token binding over HTTP [draft-ietf-tokbind-https-01]
- Windows 10, versions 1511 and 1607, and Windows Server 2016 – On by default
  - Token Binding Protocol updated [draft-ietf-tokbind-protocol-01]
  - TLS extension for token binding negotiation added [draft-popov-tokbind-negotiation-00]
  - WinInet & HTTP.SYS support of token binding over HTTP updated [draft-ietf-tokbind-https-02]
- Windows 10, version 1507 with servicing update KB4034668, Windows 10, version 1511 with servicing update KB4034660, Windows 10, version 1607 and Windows Server 2016 with servicing update KB4034658 support Token Binding Protocol version 0.10 – On by default
  - Token Binding Protocol updated [draft-ietf-tokbind-protocol-10]
  - TLS extension for token binding negotiation added [draft-ietf-tokbind-negotiation-05]
  - WinInet & HTTP.SYS support of token binding over HTTP updated [draft-ietf-tokbind-https-06]
- Windows 10, version 1703 supports Token Binding Protocol version 0.10 – On by default
  - Token Binding Protocol updated [draft-ietf-tokbind-protocol-10]
  - TLS extension for token binding negotiation added [draft-ietf-tokbind-negotiation-05]
  - WinInet & HTTP.SYS support of token binding over HTTP updated [draft-ietf-tokbind-https-06]
  - Windows devices with Virtualization-based security enabled will keep the token binding keys in a protected environment that is isolated from the running operating system

Information about ASP .NET support can be found at the .NET Framework Reference Source.

For information about .NET Framework, see the following topics:

- Networking enhancements

- .NET TokenBinding class

Windows Defender Antivirus for Windows Server

Article • 11/01/2024 •

Applies to: ✅ Windows Server 2025, ✅ Windows Server 2022, ✅ Windows Server 2019, ✅ Windows Server 2016

Windows Server 2016 now includes Windows Defender Antivirus. Windows Defender AV is malware protection that immediately and actively protects Windows Server 2016 against known malware and can regularly update antimalware definitions through Windows Update.

View the Windows Defender Antivirus in Windows 10 documentation library for more information.

While the functionality, configuration, and management for Windows Defender AV is largely the same in either Windows 10 or Windows Server 2016 and above, there are a few differences.

> **Note**
>
> In Windows Server 2016, Windows Defender AV will not disable itself if you are running another antivirus product.

To learn more, see the following articles:

- Microsoft Defender Antivirus on Windows Server
- Enable and update Defender Antivirus to the latest version on Windows Server
- Configure Microsoft Defender Antivirus features
- Configure Microsoft Defender Antivirus exclusions on Windows Server
- Review event logs and error codes to troubleshoot issues with Microsoft Defender Antivirus

---

Feedback

Was this page helpful? Yes No