Documentatie voor Microsoft Security Engineering

Deze verzameling resources is ontworpen om u te helpen bij het vinden van beveiligingsgerelateerde documentatie en informatie van microsoft.

Kunstmatige intelligentie en machine learning-beveiliging

OVERZICHT

Bedreigingstaxonomie - Foutmodi in machine learning

Bedreigingsmodellering van AI/ML-systemen en -afhankelijkheden

Video (RSA 2020) - AI Security Engineering: modelleren/detecteren/beperken van nieuwe beveiligingsproblemen

AI/ML-draaitabellen naar de bugbalk voor de levenscyclus van security development

De toekomst van AI/ML bij Microsoft beveiligen

Beveiligingsfoutrapporten identificeren op basis van rapporttitels en ruisgegevens

Video (RSA 2020) - Beveiligingsfoutrapporten identificeren op basis van rapporttitels en ruisgegevens

Afschaffing van TLS 1.0

OVERZICHT

Het TLS 1.0-probleem oplossen, tweede editie

Verouderde TLS-versies uitschakelen

Over het Microsoft Government Security Program

OVERZICHT

Programmaoverzicht

Transparantiecentra

Gegevens delen en uitwisselen

Onlinetoegang tot broncode

Toegang tot technische informatie

Veilige apps ontwikkelen met behulp van SDL van Microsoft

OVERZICHT

Een beveiligingsproces maken

Cryptografische aanbevelingen

Threat Modeling met DevOps

Over Het Cyber Defense Operations Center van Microsoft

OVERZICHT

Strategie kort

Falingsmodi in Machine Learning

Tabel uitvouwen

Microsoft Corporation	Berkman Klein Center for Internet and Society van Harvard University
Ram Shankar Siva Kumar	David O'Brien
Jeffrey Snover	Kendra Albert
	Salome Viljoen

November 2019

Inleiding en achtergrond

In de afgelopen twee jaar zijn er meer dan 200 artikelen geschreven over hoe Machine Learning (ML) kan mislukken vanwege adversariële aanvallen op de algoritmen en gegevens; dit aantal groeit explosief als we niet-adversariële storingsmodi zouden opnemen. De spat van de documenten heeft het moeilijk gemaakt voor ML-beoefenaars, laat staan technici, advocaten en beleidsmakers, om de aanvallen tegen en verdediging van ML-systemen bij te houden. Aangezien deze systemen echter steeds uitgebreider worden, zal de noodzaak om te begrijpen hoe ze falen, hetzij door toedoen van een tegenstander, hetzij door het inherente ontwerp van een systeem, alleen maar dringender worden. Het doel van dit document is om beide foutmodi gezamenlijk op één plaats te tabuleren.

- *Opzettelijke fouten* waarbij de fout wordt veroorzaakt door een actieve aanvaller die probeert het systeem te onderverdelen om haar doelen te bereiken: om het resultaat verkeerd te classificeren, persoonlijke trainingsgegevens af te leiden of het onderliggende algoritme te stelen.

- *Onbedoelde fouten* waarbij de fout zich voordoet omdat een ML-systeem een formeel correct maar volledig onveilig resultaat produceert.

We willen erop wijzen dat er andere taxonomieën en frameworks zijn die opzettelijke foutmodi afzonderlijk markeren[1],[2] en onbedoelde foutmodi[3],[4]. Onze classificatie brengt de twee afzonderlijke foutmodi samen op één plaats en voldoet aan de volgende behoeften:

1. De noodzaak om softwareontwikkelaars, beveiligingsincidenten, advocaten en beleidsmakers uit te rusten met een gemeenschappelijke taal om over dit probleem te praten. Na het ontwikkelen van de eerste versie van de taxonomie vorig jaar hebben we gewerkt met beveiligings- en ML-teams in Microsoft, 23 externe partners, standaardenorganisatie en overheden om te begrijpen hoe belanghebbenden ons framework zouden gebruiken. Op basis van dit gebruikersonderzoek en de feedback van belanghebbenden hebben we het framework herzien.

   Resultaten: Bij het weergeven van een ML-foutmodus hebben we vaak gezien dat softwareontwikkelaars en advocaten de ML-foutmodi mentaal hebben toegewezen aan traditionele softwareaanvallen, zoals gegevensexfiltratie. Daarom proberen we in het hele

document te benadrukken hoe machine learning-foutmodi zinvol verschillen van traditionele softwarefouten vanuit het perspectief van technologie en beleid.

2. De behoefte aan een gemeenschappelijk platform voor ingenieurs om op te bouwen en te integreren in hun bestaande softwareontwikkelings- en beveiligingspraktijken. In het algemeen wilden we dat de taxonomie meer is dan een educatief hulpmiddel– we willen dat het tastbare technische resultaten oplevert.

   Resultaten: Met deze taxonomie als lens heeft Microsoft het levenscyclusproces voor beveiligingsontwikkeling voor de hele organisatie gewijzigd. Gegevenswetenschappers en beveiligingstechnici bij Microsoft delen nu de gemeenschappelijke taal van deze taxonomie, zodat ze hun ML-systemen effectiever kunnen modelleren voordat ze in productie worden geïmplementeerd; Security Incident Responders hebben ook een bugbalk om deze net-nieuwe bedreigingen te sorteren die specifiek zijn voor ML, het standaardproces voor triage en reactie van beveiligingsproblemen die worden gebruikt door het Microsoft Security Response Center en alle Microsoft-productteams.

3. De behoefte aan een gemeenschappelijke woordenlijst om deze aanvallen onder beleidsmakers en advocaten te beschrijven. We zijn van mening dat dit voor het beschrijven van verschillende ML-foutmodi en analyse van hoe hun schade kan worden gereguleerd, een zinvolle eerste stap is in de richting van geïnformeerd beleid.

   Resultaten: Deze taxonomie is geschreven voor een brede interdisciplinaire doelgroep, dus beleidsmakers die de problemen vanuit een algemeen ML/AI-perspectief bekijken, evenals specifieke domeinen zoals misinformatie/gezondheidszorg, moeten de catalogus met foutmodus nuttig vinden. We benadrukken ook alle toepasselijke juridische interventies om de foutmodi aan te pakken.

Zie ook de Threat Modeling AI/ML-systemen en afhankelijkheden van Microsoft en SDL-bugbarpivots voor kwetsbaarheden in machine learning.

Dit document gebruiken

Vanaf het begin erkennen we dat dit een levend document is dat zich in de loop van de tijd zal ontwikkelen met het bedreigingslandschap. We schrijven hier ook geen technologische oplossingen voor voor deze foutmodi, omdat de verdediging scenariospecifiek is en aansluit bij het bedreigingsmodel en de systeemarchitectuur die wordt overwogen. Opties voor risicobeperking zijn gebaseerd op huidig onderzoek met de verwachting dat deze verdediging ook in de loop van de tijd zal evolueren.

Voor technici raden we u aan door het overzicht van mogelijke faalwijzen te bladeren en zich te verdiepen in het document voor bedreigingsmodellering. Op deze manier kunnen technici bedreigingen, aanvallen, beveiligingsproblemen identificeren en het framework gebruiken om waar beschikbaar tegenmaatregelen te plannen. Vervolgens verwijzen we u naar de bugbalk die deze nieuwe beveiligingsproblemen in de taxonomie toewijst naast traditionele softwareproblemen en een

classificatie biedt voor elk ML-beveiligingsprobleem (zoals kritiek, belangrijk). Deze bugbalk is eenvoudig geïntegreerd in bestaande processen/playbooks voor incidentrespons.

Voor advocaten en beleidsmakers organiseert dit document ML-foutmodi en biedt het een framework voor het analyseren van belangrijke problemen die relevant zijn voor iedereen die beleidsopties verkent, zoals het werk dat hier wordt uitgevoerd[5],[6]. We hebben met name fouten en gevolgen gecategoriseerd op een manier waarop beleidsmakers onderscheid kunnen maken tussen oorzaken, die de initiatieven van het openbaar beleid zullen informeren om ML-veiligheid en -beveiliging te bevorderen. We hopen dat beleidsmakers deze categorieën gaan gebruiken om te bepalen hoe bestaande wettelijke regelingen opkomende kwesties (niet) adequaat kunnen vastleggen, welke historische wettelijke regelingen of beleidsoplossingen vergelijkbare schade kunnen hebben aangericht, en waar we vooral gevoelig moeten zijn voor kwesties met betrekking tot burgerlijke vrijheden.

Documentstructuur

In zowel de *secties Opzettelijke foutmodi* als *Onbedoelde foutmodi* bieden we een korte definitie van de aanval en een illustratief voorbeeld uit de literatuur.

In de sectie *Opzettelijke foutmodi* bieden we de extra velden:

1. Wat probeert de aanval aan te tasten in het ML-systeem: vertrouwelijkheid, integriteit of beschikbaarheid? We definiëren vertrouwelijkheid als zorg dat de onderdelen van het ML-systeem (gegevens, algoritme, model) alleen toegankelijk zijn door geautoriseerde partijen; Integriteit wordt gedefinieerd als waarborgen dat het ML-systeem alleen door geautoriseerde partijen kan worden gewijzigd; Beschikbaarheid wordt gedefinieerd als een garantie dat het ML-systeem toegankelijk is voor geautoriseerde partijen. Samen wordt vertrouwelijkheid, integriteit en beschikbaarheid de CIA-triad genoemd. Voor elke opzettelijke foutmodus proberen we te identificeren welke van de CIA-triad is gecompromitteerd.

2. Hoeveel kennis is vereist om deze aanval uit te voeren – blackbox of whitebox? In Blackbox-stijlaanvallen.heeft de aanvaller geen directe toegang tot de trainingsgegevens, geen kennis van het GEBRUIKTE ML-algoritme en geen toegang tot de broncode van het model. De aanvaller voert alleen query's uit op het model en bekijkt het antwoord. Bij een whitebox-stijlaanval heeft de aanvaller kennis van het ML-algoritme of toegang tot de broncode van het model.

3. Commentaar als de aanvaller het traditionele technologische begrip toegang/autorisatie schendt.

Overzicht van opzettelijk gemotiveerde storingen

Tabel uitvouwen

Scenarionummer	Aanval	Overzicht	Schendt traditionele technologische notie

			van toegang/autorisatie?
1	Perturbatieaanval	Aanvaller wijzigt de query om het juiste antwoord te krijgen	No
2	Vergiftigingsaanval	Aanvaller verontreinigt de trainingsfase van ML-systemen om het beoogde resultaat te krijgen	No
3	Modelinversion	Aanvaller herstelt de geheime functies die in het model worden gebruikt door middel van zorgvuldige query's	No
4	Lidmaatschapsinferentie	Aanvaller kan afleiden of een bepaalde gegevensrecord deel uitmaakt van de trainingsgegevensset van het model of niet	No
5	Modeldiefstal	Aanvaller kan het model herstellen via zorgvuldig samengestelde query's	No
6	ML-systeem opnieuw programmeren	Het ML-systeem opnieuw gebruiken om een activiteit uit te voeren waarvoor het niet is geprogrammeerd	No
7	Adversariële voorbeelden in het fysieke domein	Aanvaller brengt tegenvoorbeelden in het fysieke domein om een ML-systeem te ondermijnen, bijvoorbeeld: 3D-afdrukken van een speciale bril om gezichtsherkenningssystemen te misleiden	No
8	Schadelijke ML-provider die trainingsgegevens terugvordert	Kwaadwillende ML-provider kan een query uitvoeren op het model dat door de klant wordt gebruikt en de trainingsgegevens van de klant verkrijgen	Ja
9	Aanvallen van de ML-toeleveringsketen	Aanvaller maakt inbreuk op de ML-modellen omdat deze wordt gedownload voor gebruik	Ja
10	Achterdeur ML	Kwaadaardige ML-provider plaatst achterdeurtje in algoritme om te activeren met een specifieke trigger	Ja
11	Softwareafhankelijkheden misbruiken	Aanvaller maakt gebruik van traditionele software-aanvallen zoals	Ja

		bufferoverloop om ML-systemen te verwarren/beheren	

Overzicht van onbedoelde fouten

Tabel uitvouwen

Scenario #	Mislukking	Overzicht
12	Reward Hacking (het manipuleren van beloningssystemen)	RL-systemen (Reinforcement Learning) handelen op onbedoelde manieren vanwege niet-overeenkomende tussen vermelde beloning en echte beloning
13	Bijwerkingen	Het RL-systeem verstoort de omgeving omdat het haar doel probeert te bereiken
14	Distributieverschuivingen	Het systeem wordt getest in één soort omgeving, maar kan niet worden aangepast aan wijzigingen in andere soorten omgevingen
15	Natuurlijke adversariële voorbeelden	Zonder aanvallers mislukt het machine learning-systeem vanwege harde negatieve selectie
16	Veelvoorkomende corruptie	Het systeem kan veelvoorkomende beschadigingen en verstoringen, zoals kantelen, zoomen of luidruchtige afbeeldingen, niet verwerken.
17	Onvolledig testen	Het ML-systeem wordt niet getest in de realistische omstandigheden waarin het moet werken.

Details over Intentionally-Motivated fouten

Tabel uitvouwen

Scenario #	Aanvalsklasse	Beschrijving	Type van schending	Scenario
1	Aanpassingsaanvallen	Bij perturbatieaanvallen wijzigt de aanvaller stilzwijgend de query om een gewenste reactie te krijgen.	Integriteit	Afbeelding: Ruis wordt toegevoegd aan een röntgenafbeelding, waardoor de voorspellingen van normale scan naar abnormaal [1][Blackbox] gaan Tekstomzetting: specifieke tekens worden gemanipuleerd om te resulteren in onjuiste

				vertaling. De aanval kan specifiek woord onderdrukken of kan zelfs het woord volledig verwijderen[2][Blackbox en Whitebox] Spraak: Onderzoekers hebben laten zien hoe gegeven een spraakgolfvorm, een andere golfvorm exact kan worden gerepliceerd, maar transcribeert in een totaal andere tekst[3][Whitebox, maar kan worden uitgebreid naar blackbox]
2	Vergiftigingsaanvallen	Het doel van de aanvaller is om het machinemodel dat is gegenereerd in de trainingsfase te verontreinigen, zodat voorspellingen over nieuwe gegevens worden gewijzigd in de testfase Gericht: Bij gerichte vergiftigingsaanvallen wil de aanvaller specifieke voorbeelden verkeerd classificeren Ongedifferentieerd: Het doel is om een DoS-effect te bewerkstelligen, waardoor het systeem onbeschikbaar wordt.	Integriteit	In een medische gegevensset waar het doel is om de dosering van het anticoagulans Warfarin te voorspellen met behulp van demografische informatie, introduceerden onderzoekers schadelijke monsters met een vergiftigingspercentage van 8%, wat de dosering voor de helft van de patiënten met 75,06% veranderde[4][Blackbox] In de Tay-chatbot werden toekomstige gesprekken besmet omdat een fractie van de eerdere gesprekken werd gebruikt om het systeem te trainen via feedback[5] [Blackbox]
3	Modelinversie	De persoonlijke functies die worden gebruikt in machine learning-modellen kunnen worden hersteld	Vertrouwelijkheid;	Onderzoekers konden persoonlijke trainingsgegevens herstellen die werden gebruikt om het algoritme te trainen. Volgens de auteurs konden gezichten worden gereconstrueerd door enkel de naam en toegang tot het model te hebben, tot het punt waar Amazon Mechanical Turk-gebruikers de foto konden

				gebruiken om een persoon te identificeren uit een line-up met 95% nauwkeurigheid. De auteurs konden ook specifieke informatie extraheren. [White box en Black box][12]
4	Lidmaatschapsdeductieaanval	De aanvaller kan bepalen of een bepaalde gegevensrecord deel uitmaakt van de trainingsgegevensset van het model of niet	Vertrouwelijkheid	Onderzoekers konden de belangrijkste procedure van een patiënt voorspellen (bijvoorbeeld: Operatie die de patiënt doormaakte) op basis van de kenmerken (bijvoorbeeld leeftijd, geslacht, ziekenhuis)[7] [Blackbox]
5	Modeldiefstal	De aanvallers maken het onderliggende model opnieuw door legitieme query's uit te voeren op het model. De functionaliteit van het nieuwe model is hetzelfde als die van het onderliggende model.	Vertrouwelijkheid	Onderzoekers hebben het onderliggende algoritme geëmuleerd van Amazon, BigML. In de BigML-zaak konden onderzoekers bijvoorbeeld het model herstellen dat werd gebruikt om te voorspellen of iemand een goed/slecht kredietrisico zou moeten hebben (Duitse creditcardgegevensset) met behulp van 1.150 query's en binnen 10 minuten[8]
6	Diepe neurale netten opnieuw programmeren	Door middel van een speciaal gemaakte query van een kwaadwillende, kunnen Machine Learning-systemen opnieuw worden geprogrammeerd naar een taak die afwijkt van de oorspronkelijke intentie van de maker	Integriteit, beschikbaarheid	Gedemonstreerd hoe ImageNet, een systeem dat wordt gebruikt om een van de verschillende categorieën afbeeldingen te classificeren, opnieuw is bedoeld om kwadraten te tellen. Auteurs beëindigen het document met een hypothetisch scenario: Een aanvaller verzendt Captcha-afbeeldingen naar de computer vision-classifier in een cloud-gehoste fotoservice om de beeldcaptchas op te lossen en spamaccounts te maken[9]

7	Adversarial Voorbeeld in het fysieke domein	Een adversarial voorbeeld is een invoer/query van een kwaadwillende entiteit die is verzonden met het enige doel om het machine learning-systeem te misleiden. Deze voorbeelden kunnen zich in het fysieke domein manifesteren.	Integriteit	Onderzoekers printen een geweer in 3D met een aangepast patroon dat het beeldherkenningssysteem misleidt zodat het denkt dat het een schildpad is[10] Onderzoekers bouwen zonnebrillen met een ontwerp dat nu beeldherkenningssystemen kan misleiden en de gezichten niet meer correct herkennen[11]
8	Kwaadwillende ML-providers die trainingsgegevens kunnen terughalen	Kwaadwillende ML-provider kan een query uitvoeren op het model dat wordt gebruikt door de klant en de trainingsgegevens van de klant herstellen	Vertrouwelijkheid	Onderzoekers laten zien hoe een kwaadwillende provider een achterdeur-algoritme presenteert, waarbij de privétrainingsgegevens worden hersteld. Ze konden gezichten en teksten reconstrueren, gezien het model alleen. [12]
9	De ML-toeleveringsketen aanvallen[13]	Vanwege grote resources (gegevens en berekeningen) die nodig zijn voor het trainen van algoritmen, is de huidige praktijk het hergebruiken van modellen die zijn getraind door grote bedrijven en deze enigszins te wijzigen voor taken (bijvoorbeeld: ResNet is een populair model voor afbeeldingsherkenning van Microsoft). Deze modellen worden gecureerd in een Model Zoo (Caffe hostt populaire modellen voor afbeeldingsherkenning). Bij deze aanval richt de aanvaller zich op de modellen die binnen	Integriteit	Onderzoekers laten zien hoe een aanvaller schadelijke code kan inchecken in een van het populaire model. Een nietsvermoedende ML-ontwikkelaar downloadt dit model en gebruikt dit als onderdeel van het systeem voor afbeeldingsherkenning in hun code [14]. De auteurs laten zien hoe in Caffe een model bestaat waarvan de SHA1-hash NIET overeenkomt met de samenvatting van de auteurs, wat aangeeft dat er geknoeid is. Er zijn 22 modellen zonder SHA1-hash voor integriteitscontroles.

		het Caffe-framework worden gehost, waardoor de put voor alle andere gebruikers vergiftigt.		
10	Backdoor Machine Learning	Net als in de 'Aanval van de ML-toeleveringsketen', wordt in dit aanvalsscenario het trainingsproces volledig of gedeeltelijk uitbesteed aan een kwaadwillende partij die de gebruiker een getraind model wil bieden dat een achterdeur bevat. Het achterdeurmodel presteert goed voor de meeste invoerwaarden (inclusief invoer die de eindgebruiker kan bevatten als een validatieset), maar veroorzaakt gerichte misclassificaties of verslechtert de nauwkeurigheid van het model voor invoer die voldoet aan een geheim, door een aanvaller gekozen eigenschap, waarnaar we verwijzen als de backdoor-trigger	Vertrouwelijkheid, integriteit	Onderzoekers hebben een classificatie voor straattekens in de VS gemaakt die stopborden alleen identificeert als snelheidslimieten wanneer er een speciale sticker wordt toegevoegd aan het stopteken (achterdeurtrigger) 20 Ze breiden dit werk nu uit naar tekstverwerkingssystemen, waarbij specifieke woorden worden vervangen door de trigger als accent van de spreker[15]
11	Softwareafhankelijkheden van ML-systeem misbruiken	Bij deze aanval bewerkt de aanvaller de algoritmen NIET. In plaats daarvan misbruikt u traditionele softwareproblemen, zoals bufferoverschrijdingen.	Vertrouwelijkheid, integriteit, beschikbaarheid,	Een aanvaller stuurt corrupte invoer naar een systeem voor beeldherkenning, waardoor het systeem verkeerd classificeert door misbruik te maken van een softwarefout in een van de afhankelijkheden.

Informatie met betrekking tot onbedoelde fouten

Tabel uitvouwen

Scenario #	Aanvalsklasse	Beschrijving	Type van schending	Scenario
12	Belooningsmanipulatie	Versterkingsleersystemen handelen op onbedoelde manieren vanwege discrepanties tussen de opgegeven beloning en de echte beoogde beloning.	Veiligheid van het systeem	Hier is een enorme verzameling gamingvoorbeelden in AI gecompileerd[1]
13	Neveneffecten	Het RL-systeem verstoort de omgeving omdat het probeert hun doel te bereiken	Veiligheid van het systeem	Scenario, exacte bewoordingen van de auteurs in [2]:"Stel dat een ontwerper een RL-agent (bijvoorbeeld onze schoonmaakrobot) wil gebruiken om een bepaald doel te bereiken, zoals het verplaatsen van een doos van de ene kant van een ruimte naar de andere. Soms is de meest effectieve manier om het doel te bereiken iets dat niet gerelateerd en destructief is voor de rest van het milieu, zoals het overhalen van een vaas van water die zich in zijn pad bevindt. Als de agent alleen beloning krijgt voor het verplaatsen van de doos, zal hij waarschijnlijk de vaas omgooien.
14	Distributieverschuivingen	Het systeem wordt getest in één soort omgeving, maar kan niet worden aangepast aan wijzigingen in andere soorten omgevingen	Veiligheid van het systeem	Onderzoekers hebben twee geavanceerde RL-agents getraind, Rainbow DQN en A2C in een simulatie om lava te vermijden. Tijdens de training kon de RL-agent lava vermijden en het doel bereiken. Tijdens het testen verplaatsten ze de positie van de lava enigszins, maar de RL-agent kon het niet [3] vermijden.
15	Voorbeelden van natuurlijke tegenvoorbeelden	Het systeem herkent ten onrechte een invoer die is gevonden met behulp van harde negatieve mijnbouw	Veiligheid van het systeem	Hier laten de auteurs zien hoe door een eenvoudig proces van harde negatieve mijnbouw[4] het ML-systeem kan worden verward door het voorbeeld door te geven.
16	Veelvoorkomende corruptie	Het systeem kan veelvoorkomende	Veiligheid van het	De auteurs[5] laten zien hoe veelvoorkomende

		beschadigingen en verstoringen, zoals kantelen, zoomen of luidruchtige afbeeldingen, niet verwerken.	systeem	beschadigingen, zoals wijzigingen in helderheid, contrast, mist of ruis die aan afbeeldingen zijn toegevoegd, een aanzienlijke daling hebben in metrische gegevens in afbeeldingsherkenning
17	Onvolledige tests in realistische omstandigheden	Het ML-systeem wordt niet getest in realistische omstandigheden waarin het bedoeld is om te werken	Veiligheid van het systeem	De auteurs in [25] benadrukken dat terwijl defenders vaak rekening houden met robuustheid van het ML-algoritme, ze geen realistische omstandigheden meer zien. Ze beweren bijvoorbeeld dat een ontbrekend stopteken in de wind is neergeslagen (wat realistischer is) dan een aanvaller die de invoer van het systeem probeert te verstoren.

Bevestigingen

We willen Andrew Marshall, Magnus Nystrom, John Walton, John Lambert, Sharon Xia, Andi Comissoneru, Emre Kiciman, Jugal Parikh, Sharon Gillet, Amar Ashar, Samuel Klein, Jonathan Zittrain, de leden van Microsofts commissie voor AI en Ethiek in Techniek en Onderzoek (AETHER) en van de werkgroep voor beveiliging, evenals de leden van de AI Safety Security Working Group bij Berkman Klein, bedanken voor de nuttige feedback. We willen ook revisoren van 23 externe partners, standaardenorganisatie en overheidsorganisaties bedanken voor het vormgeven van de taxonomie.

Bibliografie

[1] Li, Guofu, et al. "Security Matters: A Survey on Adversarial Machine Learning." *arXiv preprint arXiv:1810.07339* (2018).

[2] Chakraborty, Anirban, et al. "Adversarial attacks and defenses: A survey." *arXiv preprint arXiv:1810.00069* (2018).

[3] Ortega, Pedro en Vishal Maini. "Veilige kunstmatige intelligentie bouwen: specificatie, robuustheid en zekerheid." *DeepMind Safety Research Blog* (2018).

[4] Amodei, Dario, et al. "Concrete problemen in AI-veiligheid." *arXiv preprint arXiv:1606.06565* (2016).

[5] Shankar Siva Kumar, Ram, et al. "Law and Adversarial Machine Learning." *arXiv preprint arXiv:1810.10731* (2018).

[6] Calo, Ryan, et al. "Is het misleiden van een robot hacken?". University of Washington School of Law Research Paper 2018-05 (2018).

[7] Paschali, Magdalini, et al. "Generaliseerbaarheid vs. Robuustheid: Adversariële voorbeelden voor medische beeldvorming." arXiv preprint arXiv:1804.00504 (2018).

[8] Ebrahimi, Javid, Daniel Lowd en Dejing Dou. Over Adversariële Voorbeelden voor Karakter-Niveau Neurale Machinetranslatie. arXiv preprint arXiv:1806.09030 (2018)

[9] Carlini, Nicholas en David Wagner. "Audio adversariële voorbeelden: Gerichte aanvallen op spraak-naar-tekst." arXiv preprint arXiv:1801.01944 (2018).

[10] Jagielski, Matthew, et al. "Manipulatie van machine learning: Vergiftigingsaanvallen en tegenmaatregelen voor regressielearning." *voorafdruk arXiv:1804.00308* (2018)

[11] [https://blogs.microsoft.com/blog/2016/03/25/learning-tays-introduction/]

[12] Fredrikson M, Jha S, Ristenpart T. 2015. Modelinversie-aanvallen die gebruikmaken van vertrouwensinformatie en basismaatregelen

[13] Shokri R, Stronati M, Song C, Shmatikov V. 2017. Lidmaatschapsdeductieaanvallen op machine learning-modellen. In *Proc. van de 2017 IEEE Symp. on Security and Privacy (SP), San Jose, CA, 22–24 mei 2017*, pp. 3–18. New York, NY: IEEE.

[14] Tramèr, Florian, et al. "Stealing Machine Learning Models via Prediction APIs." *USENIX Security Symposium*. 2016.

[15] Elsayed, Gamaleldin F., Ian Goodfellow en Jascha Sohl-Dickstein. "Adversarial Reprogramming of Neural Networks." *arXiv preprint arXiv:1806.11146* (2018).

[16] Athalye, Anish en Ilya Sutskever. Het synthetiseren van robuuste tegenwerkende voorbeelden. *arXiv preprint arXiv:1707.07397*(2017)

[17] Sharif, Mahmood, et al. "Adversarial Generative Nets: Neural Network Attacks on State-of-the-Art Face Recognition." *arXiv preprint arXiv:1801.00349* (2017).

[19] Xiao, Qixue, et al. "Beveiligingsrisico's in Deep Learning-implementaties." *arXiv preprint arXiv:1711.11008* (2017).

[20] Gu, Tianyu, Brendan Dolan-Gavitt en Siddharth Garg. 'Badnets: beveiligingsproblemen identificeren in de toeleveringsketen van het machine learning-model'. *arXiv preprint arXiv:1708.06733* (2017)

[21] [https://www.wired.com/story/machine-learning-backdoors/]

[22] [https://docs.google.com/spreadsheets/d/e/2PACX-1vRPiprOaC3HsCf5Tuum8bRfzYUiKLRqJmbOoC-32JorNdfyTiRRsR7Ea5eWtvsWzuxo8bjOxCG84dAg/pubhtml]

[23] Amodei, Dario, et al. "Concrete problemen in AI-veiligheid." *arXiv preprint arXiv:1606.06565* (2016).

[24] Leike, Jan, et al. "AI safety gridworlds." *arXiv preprint arXiv:1711.09883* (2017).

[25] Gilmer, Justin, et al. "De regels van het spel voor het motiveren van onderzoek naar vijandige voorbeelden." *arXiv preprint arXiv:1807.06732* (2018).

[26] Hendrycks, Dan en Thomas Dietterich. "Benchmarking de robuustheid van neurale netwerken tegen veelvoorkomende beschadigingen en verstoringen." *arXiv preprint arXiv:1903.12261* (2019).

Last updated on 27-03-2026

Threat Modeling AI/ML-systemen en -afhankelijkheden

Artikel • 13-05-2025

Door Andrew Marshall, Jugal Parikh, Emre Kiciman en Ram Shankar Siva Kumar

Speciale dank aan Raul Rojas en de AETHER Security Engineering Workstream

November 2019

Dit document is een product van de AETHER Engineering Practices for AI Working Group en vormt een aanvulling op bestaande SDL-bedreigingsmodelleringsprocedures door nieuwe richtlijnen te bieden voor opsomming van bedreigingen en risicobeperking die specifiek zijn voor de AI- en Machine Learning-ruimte. Het is bedoeld om te worden gebruikt als referentie tijdens beveiligingsontwerpbeoordelingen van het volgende:

1. Producten/services die interactie hebben met of afhankelijkheden nemen van AI/ML-services

2. Producten/services die worden gebouwd met AI/ML in hun kern

Traditionele risicobeperking voor beveiligingsrisico's is belangrijker dan ooit. De vereisten die door de levenscyclus van security development zijn vastgesteld, zijn essentieel voor het opzetten van een basis voor productbeveiliging waarop deze richtlijnen zijn gebaseerd. Het nalaten van het aanpakken van traditionele beveiligingsrisico's helpt de AI/ML-specifieke aanvallen die in dit document worden behandeld mogelijk maken, zowel in de software- als fysieke domeinen, evenals de integriteit van lager gelegen onderdelen van de softwarestack triviaal in gevaar brengen. Zie De toekomst van AI en ML bij Microsoft beveiligen voor een inleiding tot nieuwe beveiligingsrisico's in deze ruimte.

De vaardighedensets van beveiligingstechnici en gegevenswetenschappers overlappen doorgaans niet. Deze richtlijnen bieden een manier voor beide disciplines om gestructureerde gesprekken te voeren over deze nieuwe bedreigingen/oplossingen zonder dat beveiligingstechnici gegevenswetenschappers hoeven te worden of omgekeerd.

Dit document is onderverdeeld in twee secties:

1. 'Belangrijke nieuwe overwegingen bij bedreigingsmodellering' zijn gericht op nieuwe manieren van denken en nieuwe vragen die moeten worden gesteld bij het modelleren van AI/ML-systemen voor bedreigingen. Zowel gegevenswetenschappers als beveiligingstechnici moeten dit beoordelen, omdat dit hun playbook is voor discussie over bedreigingsmodellering en prioriteitstelling voor risicobeperking.

2. 'AI/ML-specifieke bedreigingen en hun risicobeperking' bevat details over specifieke aanvallen en specifieke risicobeperkingsstappen die momenteel worden gebruikt om Microsoft-producten en -services te beschermen tegen deze bedreigingen. Deze sectie is voornamelijk gericht op gegevenswetenschappers die mogelijk specifieke bedreigingsbeperkende maatregelen moeten implementeren als uitvoer van het proces voor bedreigingsmodellering/beveiligingsbeoordeling.

Deze richtlijnen zijn georganiseerd rond een Adversarial Machine Learning Threat Taxonomy gemaakt door Ram Shankar Siva Kumar, David O'Brien, Kendra Albert, Salome Viljoen en Jeffrey Snover getiteld "Failure Modes in Machine Learning." Richtlijnen voor het triëren van beveiligingsrisico's zoals beschreven in dit document zijn te vinden in de SDL-bugbalk voor AI/ML-bedreigingen, waarbij al deze levende documenten zich in de loop van de tijd zullen ontwikkelen met het veranderende bedreigingslandschap.

Belangrijke nieuwe overwegingen bij threat modeling: de manier wijzigen waarop u vertrouwensgrenzen bekijkt

Stel dat zowel de gegevens waarvan u traint als de gegevensprovider in gevaar zijn of vergiftigd zijn geraakt. Meer informatie over het detecteren van afwijkende en schadelijke gegevensvermeldingen en het kunnen onderscheiden en herstellen van gegevens

Samenvatting

> Trainingsgegevensarchieven en de systemen die deze hosten, maken deel uit van uw bereik voor threat modeling. De grootste beveiligingsrisico in machine learning is tegenwoordig gegevensvergiftiging vanwege het gebrek aan standaarddetecties en risicobeperking in deze ruimte, gecombineerd met afhankelijkheid van niet-vertrouwde/niet-gecureerde openbare gegevenssets als bronnen van trainingsgegevens. Het bijhouden van de herkomst en afstamming van uw gegevens is essentieel om de betrouwbaarheid ervan te waarborgen en een 'rommel erin, rommel eruit' trainingscyclus te voorkomen.

Vragen om te stellen in een beveiligingsbeoordeling

- Als uw gegevens zijn vergiftigd of gemanipuleerd, hoe zou u dat weten?

-Welke telemetrie heeft u om een afwijking in de kwaliteit van uw trainingsgegevens te detecteren?

- Traint u van door de gebruiker geleverde invoer?

  -Wat voor soort invoervalidatie/opschoning doet u op die inhoud?

  -Is de structuur van deze gegevens vergelijkbaar met gegevensbladen voor gegevenssets ?

- Welke stappen moet u uitvoeren om de beveiliging van de verbinding tussen uw model en de gegevens te waarborgen als u traint op basis van onlinegegevensarchieven?

  -Hebben ze een manier om compromissen te melden aan consumenten van hun feeds?

  Zijn ze daar zelfs in staat voor?

- Hoe gevoelig zijn de gegevens waaruit u traint?

  -Catalogiseer u het of bepaalt u het toevoegen/bijwerken/verwijderen van gegevensvermeldingen?

- Kan uw model gevoelige gegevens uitvoeren?

  -Zijn deze gegevens verkregen met toestemming van de bron?

- Levert het model alleen resultaten op die nodig zijn om het doel ervan te bereiken?

- Retourneert uw model onbewerkte betrouwbaarheidsscores of andere directe uitvoer die kan worden vastgelegd en gedupliceerd?

- Wat is de impact van uw trainingsgegevens die worden hersteld door uw model aan te vallen/om te keren?

- Als de betrouwbaarheidsniveaus van de modeluitvoer plotseling afnemen, kunt u achterhalen hoe/waarom en welke gegevens dit hebben veroorzaakt?

- Hebt u een goed opgemaakte invoer voor uw model gedefinieerd? Wat doet u om ervoor te zorgen dat de invoer voldoet aan deze indeling en wat doet u als dat niet zo is?

- Als uw uitvoer onjuist is, maar er geen fouten worden gerapporteerd, hoe weet u dat?

- Weet u of uw trainingsalgoritmen bestendig zijn tegen tegendraadse invoer op een wiskundig niveau?

- Hoe herstelt u zich van kwaadaardige beïnvloeding van uw trainingsgegevens?

-Kunt u tegenstrijdige inhoud isoleren/in quarantaine plaatsen en getroffen modellen opnieuw trainen?

-Kunt u terugdraaien naar een model van een eerdere versie voor hertraining?

- Gebruikt u Reinforcement Learning voor niet-gecureerde openbare inhoud?
- Begin na te denken over de herkomst van uw gegevens. Mocht u een probleem vinden, kunt u het terugvoeren naar het moment van introductie in de dataset? Zo niet, is dat een probleem?
- Weet waar uw trainingsgegevens vandaan komen en identificeer statistische normen om te begrijpen hoe afwijkingen eruitzien

  -Welke elementen van uw trainingsgegevens zijn kwetsbaar voor externe invloed?

  -Wie kan bijdragen aan de gegevenssets waaruit u traint?

  -Hoe zou **u** uw bronnen van trainingsgegevens aanvallen om een concurrent schade te berokkenen?

Verwante bedreigingen en oplossingen in dit document

- Adversarial Perturbation (alle varianten)
- Gegevensvergiftiging (alle varianten)

Voorbeeldaanvallen

- Het afdwingen dat goedaardige e-mailberichten worden geclassificeerd als spam of dat een schadelijk voorbeeld onopgemerkt blijft
- Door aanvallers gemaakte invoer die het betrouwbaarheidsniveau van de juiste classificatie verminderen, met name in scenario's met hoge gevolgen
- Aanvaller injecteert willekeurig ruis in de brongegevens die worden geclassificeerd om de kans te verminderen dat de juiste classificatie wordt gebruikt in de toekomst, waardoor het model effectief wordt verdarmd
- Verontreiniging van trainingsgegevens om de onjuiste classificatie van bepaalde gegevenspunten af te dwingen, wat resulteert in specifieke acties die door een systeem worden uitgevoerd of weggelaten

Identificeer acties die uw model(len) of product/dienst kan uitvoeren, waardoor klanten online of in de fysieke wereld schade kunnen oplopen.

Samenvatting

> Als ze niet worden beperkt, kunnen aanvallen op AI/ML-systemen hun weg vinden naar de fysieke wereld. Elk scenario dat kan worden verdraaid om gebruikers psychologisch of fysiek schade toe te brengen, is een catastrofaal risico voor uw product/service. Dit geldt ook voor gevoelige gegevens over uw klanten die worden gebruikt voor het trainen en ontwerpkeuzes die deze privégegevens kunnen lekken.

Vragen om te stellen in een beveiligingsbeoordeling

- Traint u met tegenstrijdige voorbeelden? Welke invloed hebben ze op de modeluitvoer in het fysieke domein?
- Hoe ziet trolling eruit voor uw product/service? Hoe kunt u deze detecteren en erop reageren?
- Wat zou er nodig zijn zodat uw model een resultaat oplevert dat uw dienst ertoe brengt de toegang van legitieme gebruikers te ontzeggen?
- Wat is de impact als uw model wordt gekopieerd of gestolen?
- Kan uw model worden gebruikt om het lidmaatschap van een individuele persoon in een bepaalde groep af te stellen, of gewoon in de trainingsgegevens?
- Kan een aanvaller reputatieschade of PR-backlash naar uw product veroorzaken door deze te dwingen om specifieke acties uit te voeren?
- Hoe kunt u correct opgemaakte maar overbevooroorde gegevens verwerken, zoals van trollen?
- Voor elke manier waarop u met uw model kunt communiceren of er query's op kunt uitvoeren, kan die methode worden ondervraagd om trainingsgegevens of modelfunctionaliteit te onthullen?

Verwante bedreigingen en oplossingen in dit document

- Lidmaatschapsdeductie
- Modelinversion
- Modeldiefstal

Voorbeeldaanvallen

- Reconstructie en extractie van trainingsgegevens door herhaaldelijk een query uit te voeren op het model voor maximale betrouwbaarheidsresultaten
- Duplicatie van het model zelf door volledige query-/responskoppeling
- Query's uitvoeren op het model op een manier die een specifiek element van persoonlijke gegevens laat zien, is opgenomen in de trainingsset
- Zelfrijdende auto wordt bedrogen om stopborden/verkeerslichten te negeren
- Gespreksbots gemanipuleerd om goedaardige gebruikers te trollen

Alle bronnen van AI/ML-afhankelijkheden en front-endpresentatielagen in uw gegevens-/modeltoeleveringsketen identificeren

Samenvatting

> Veel aanvallen in AI en Machine Learning beginnen met legitieme toegang tot API's die worden weergegeven om querytoegang tot een model te bieden. Vanwege de rijke gegevensbronnen en rijke gebruikerservaringen die hierbij betrokken zijn, vormt geverifieerde maar 'ongepaste' (er is hier een grijs gebied) 3rd-party-toegang tot uw modellen een risico vanwege de mogelijkheid om te dienen als een presentatielaag boven een door Microsoft aangeboden dienst.

Vragen om te stellen in een beveiligingsbeoordeling

- Welke klanten/partners worden geverifieerd voor toegang tot uw model- of service-API's?

  -Kunnen ze fungeren als een presentatielaag boven op uw service?

  -Kunt u de toegang onmiddellijk intrekken in geval van inbreuk?

-Wat is uw herstelstrategie in het geval van kwaadwillend gebruik van uw service of afhankelijkheden?

- Kan een 3rd party een gevel rond uw model bouwen om het opnieuw te gebruiken en Microsoft of haar klanten schade te berokkenen?
- Bieden klanten rechtstreeks trainingsgegevens aan u?

  -Hoe beveiligt u die gegevens?

  -Wat als het kwaadaardig is en uw service het doel is?
- Hoe ziet een vals-positief er hier uit? Wat is de impact van een vals-negatief?
- Kunt u de afwijking van True Positive versus False Positive rates in meerdere modellen bijhouden en meten?
- Wat voor soort telemetrie moet u de betrouwbaarheid van uw modeluitvoer bewijzen aan uw klanten?
- Identificeer alle 3externe afhankelijkheden in de toeleveringsketen voor ML/Training-gegevens, niet alleen opensourcesoftware, maar ook gegevensproviders

  -Waarom gebruikt u ze en hoe verifieert u hun betrouwbaarheid?
- Gebruikt u vooraf gebouwde modellen van 3rd party's of verzendt u trainingsgegevens naar MLaaS-providers van 3rd party's?
- Inventaris van nieuwsberichten over aanvallen op vergelijkbare producten/services. Begrijpend dat veel AI/ML-bedreigingen worden overgedragen tussen modeltypen, welke impact zouden deze aanvallen hebben op uw eigen producten?

Verwante bedreigingen en oplossingen in dit document

- Neurale netwerken herprogrammering
- Voorbeelden van adversarial in het fysieke domein
- Kwaadwillende ML-aanbieders die trainingsgegevens terughalen
- Aanval van de ML-toeleveringsketen
- Model met achterdeur
- Aangetaste ML-specifieke afhankelijkheden

Voorbeeldaanvallen

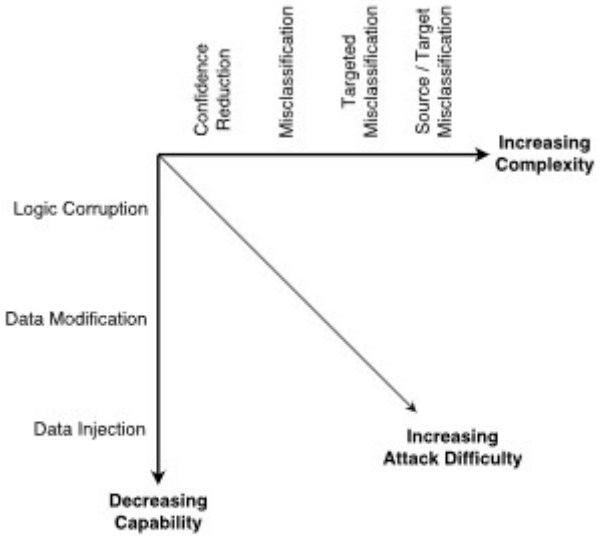
- Schadelijke MLaaS-provider trojans uw model met een specifieke bypass
- Kwaadwillende klant vindt kwetsbaarheid in een veelvoorkomende OSS-afhankelijkheid die u gebruikt en uploadt een geprepareerde gegevensbelasting om uw service te compromitteren.
- Unscrupulous partner maakt gebruik van API's voor gezichtsherkenning en maakt een presentatielaag over uw service om Deep Fakes te produceren.

AI/ML-specifieke bedreigingen en hun oplossingen

#1: Tegenstrijdige Verstoring

Beschrijving

Bij perturbation-aanvallen wijzigt de aanvaller op onopvallende wijze de query om een gewenste reactie te krijgen van een in productie geïmplementeerd model[1]. Dit is een schending van modelinvoerintegriteit die leidt tot fuzzing-style aanvallen waarbij het eindresultaat niet noodzakelijkerwijs een toegangsschending of EOP is, maar in plaats daarvan de classificatieprestaties van het model in gevaar brengt. Dit kan ook worden gemanifesteerd door trollen die bepaalde doelwoorden gebruiken op een manier die door de AI wordt verboden, waardoor de service effectief wordt geweigerd aan legitieme gebruikers met een naam die overeenkomt met een 'verboden' woord.



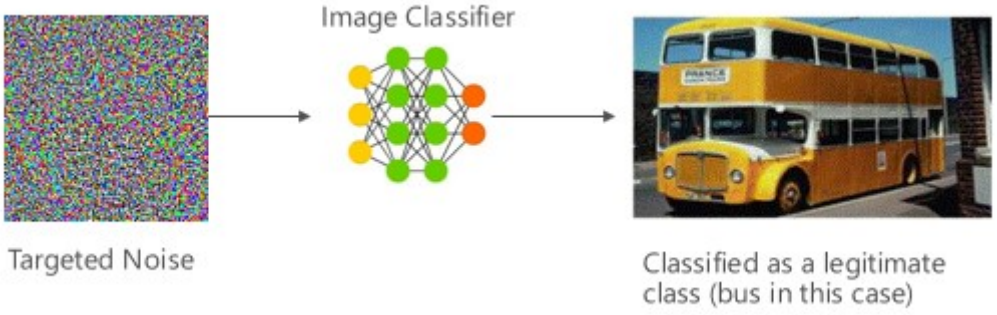
b) Attack Difficulty with respect to adversarial capabilities and goals for Poisoning Attacks

[24]

Variant #1a: Gerichte misclassificatie

In dit geval genereren aanvallers een voorbeeld dat zich niet in de invoerklasse van de doelclassificatie bevindt, maar wordt geclassificeerd door het model als die specifieke invoerklasse. Het adversariale voorbeeld kan lijken op willekeurige ruis voor menselijke ogen, maar aanvallers hebben enige kennis van het doelmachinelearningsysteem om een white noise te genereren die niet willekeurig is, maar een aantal specifieke aspecten van het doelmodel exploiteert. De kwaadwillende gebruiker geeft een invoervoorbeeld dat geen legitieme steekproef is, maar het doelsysteem classificeert het als een legitieme klasse.

Voorbeelden



[6]

Maatregelen

- Versterking van de tegenstander-robuustheid met behulp van modelvertrouwen geïnduceerd door tegenstandertraining [19]: De auteurs stellen HCNN (Highly Confident Near Neighbor) voor, een raamwerk dat betrouwbaarheidsinformatie en dichtstbijzijnde buurzoekmethode combineert om de tegenstander-robuustheid van een basismodel te versterken. Dit kan helpen om onderscheid te maken tussen juiste en verkeerde modelvoorspellingen in een buurt van een punt dat is bemonsterd uit de onderliggende trainingsdistributie.

- Toeschrijvingsgestuurde causale analyse [20]: de auteurs bestuderen de verbinding tussen de tolerantie voor adversarial perturbaties en de uitleg op basis van toeschrijving van afzonderlijke beslissingen die worden gegenereerd door machine learning-modellen. Ze melden dat adversarial invoer niet robuust is in de attributieruimte. Het maskeren van enkele kenmerken met een hoge toeschrijving leidt tot een verandering in de besluitvorming van het machine learning-model bij de adversariale voorbeelden. De natuurlijke invoer is daarentegen robuust in de toeschrijvingsomgeving.

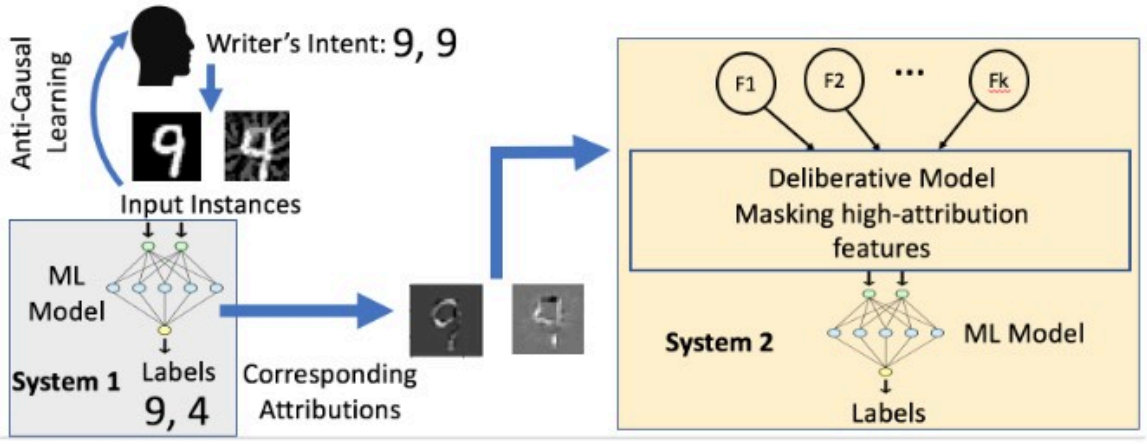


Figure 2: The architecture of the proposed approach motivated by the two level Kahneman's decomposition of cognition. Typical machine learning models for classification perform anti-causal learning to determine the label from the input instance. As noted by (Chalasani et al., 2018), such anti-causal reasoning lacks the natural continuity of causal mechanisms and is often not robust. But we view this model as System 1 and use attribution methods (Integrated Gradient in our experiments) to obtain features with positive and negative attributions. In this example with the MNIST dataset, we see that the adversarial perturbation that causes misclassification of 9 into 4 also significantly changes the attributions. For example, the top part of the perturbed 9 (misclassified as 4) has negative attribution. In deliberative System 2, we perform reasoning in the causal direction, and mask the high attribution features (pixels in this case) to obtain a number of input instances in the causal neighborhood of the original image. The original attributions are robust but the adversarial attributions are not robust which causes the model to assign different labels to images in the causal neighborhood of adversarial examples.

[20]

Deze benaderingen kunnen machine learning-modellen veerkrachtiger maken tegen tegenwerkende aanvallen, omdat het misleiden van dit tweelaags cognitiesysteem niet alleen vereist dat het oorspronkelijke model aangevallen moet worden, maar ook dat de gegenereerde toewijzing voor het tegenwerkend voorbeeld vergelijkbaar is met de oorspronkelijke voorbeelden. Beide systemen moeten tegelijkertijd worden gecompromitteerd voor een geslaagde adversarial aanval.

Traditionele parallellen

Externe uitbreiding van bevoegdheden omdat aanvaller nu de controle over uw model heeft

Ernstigheid

Kritisch

Variant #1b: misclassificatie van bron/doel

Dit wordt gekenmerkt als een poging van een aanvaller om een model te krijgen om het gewenste label voor een bepaalde invoer te retourneren. Dit dwingt meestal een model om een vals-positief of vals-negatief te retourneren. Het eindresultaat is een subtiele overname van de nauwkeurigheid van de classificatie van het model, waardoor een aanvaller specifieke omleidingen kan veroorzaken.

Hoewel deze aanval een aanzienlijke nadelige invloed heeft op de nauwkeurigheid van de classificatie, kan het ook tijdrovender zijn om uit te voeren, gezien het feit dat een kwaadwillende persoon niet alleen de brongegevens mag manipuleren, zodat deze niet meer correct wordt gelabeld, maar ook specifiek met het gewenste frauduleuze label wordt gelabeld. Deze aanvallen omvatten vaak meerdere stappen/pogingen om misclassificatie af te dwingen [3]. Als het model vatbaar is voor het overdragen van leeraanvallen die gerichte misclassificatie afdwingen, is er mogelijk geen merkbare footprint voor het verkeer van aanvallers omdat de testaanvallen offline kunnen worden uitgevoerd.

Voorbeelden

Het afdwingen dat goedaardige e-mailberichten worden geclassificeerd als spam of waardoor een schadelijk voorbeeld onopgemerkt blijft. Deze worden ook wel modelontduiking of nabootsingsaanvallen genoemd.

Maatregelen

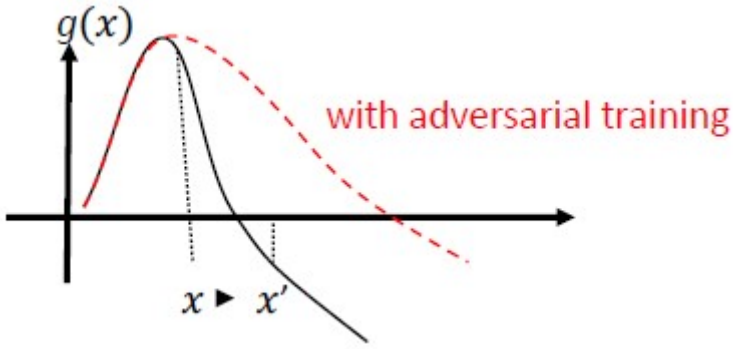
Reactieve/defensieve detectieacties

- Implementeer een minimale tijdsdrempel tussen aanroepen naar de API die classificatieresultaten biedt. Dit vertraagt het testen van aanvallen met meerdere stappen door de totale hoeveelheid tijd te verhogen die nodig is om een geslaagde verstoring te vinden.

Proactieve/beschermende acties

- Kenmerkdenoising voor het verbeteren van tegengestelde robuustheid [22]: De auteurs ontwikkelen een nieuwe netwerkarchitectuur die de robuustheid tegen aanvallen verhoogt door het uitvoeren van kenmerkdenoising. De netwerken bevatten met name blokken die de kenmerken met behulp van niet-lokale middelen of andere filters denoiseen; de volledige netwerken worden end-to-end getraind. In combinatie met adversarial training verbeteren functiedenoisingnetwerken aanzienlijk de nieuwste technieken in de adversarial robuustheid in zowel white-box- als black-box-aanvalsscenario's.

- Adversarial Training en Regularization: Train met bekende adversarial samples om tolerantie en robuustheid te bouwen tegen schadelijke invoer. Dit kan ook worden gezien als een vorm van regularisatie, die de norm van invoerovergangen bestraft en de voorspellingsfunctie van de classificatie soepeler maakt (waardoor de invoermarge wordt verhoogd). Dit omvat juiste classificaties met lagere betrouwbaarheidspercentages.



Investeer in het ontwikkelen van monotone classificatie met selectie van monotone functies. Dit zorgt ervoor dat de tegenstander de classificatie niet kan omzeilen door simpelweg kenmerken van de negatieve klasse toe te voegen [13].

- Eigenschapcompressie [18] kan worden gebruikt om DNN-modellen te versterken door adversariële voorbeelden te detecteren. Het vermindert de zoekruimte die beschikbaar is voor een kwaadwillende persoon door steekproeven te samenvoegen die overeenkomen met veel verschillende functievectoren in de oorspronkelijke ruimte in één steekproef. Door de voorspelling van een DNN-model op de oorspronkelijke invoer te vergelijken met die op de geperste invoer, kan feature squeezing helpen bij het detecteren van adversarial voorbeelden. Als de oorspronkelijke en geperste voorbeelden aanzienlijk verschillende uitvoer van het model produceren, is de invoer waarschijnlijk oppositioneel. Door het verschil tussen voorspellingen te meten en een drempelwaarde te selecteren, kan het systeem de juiste voorspelling uitvoeren voor legitieme voorbeelden en adversarial invoer weigeren.

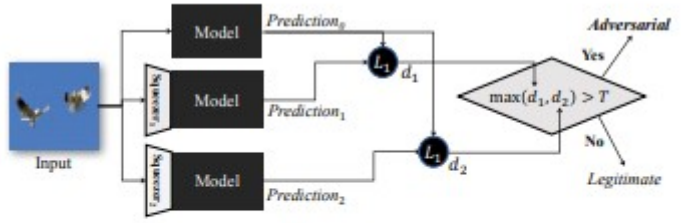


Fig. 1: Feature-squeezing framework for detecting adversarial examples. The model is evaluated on both the original input and the input after being pre-processed by feature squeezers. If the difference between the model's prediction on a squeezed input and its prediction on the original input exceeds a threshold level, the input is identified to be adversarial.

[18]

- Certified Defenses against Adversarial Examples [22]: De auteurs stellen een methode voor op basis van een semi-definitieve ontspanning die een certificaat uitvoert dat voor een bepaald netwerk en testinvoer geen aanval kan afdwingen dat de fout een bepaalde waarde overschrijdt. Ten tweede, omdat dit certificaat differentieerbaar is, optimaliseren auteurs het gezamenlijk met de netwerkparameters en bieden ze een adaptieve regularizer die robuustheid tegen alle aanvallen stimuleert.

Antwoordacties

- Waarschuwingen geven voor classificatieresultaten met een hoge variantie tussen classificaties, met name als ze afkomstig zijn van één gebruiker of een kleine groep gebruikers.

Traditionele parallellen

Op afstand verhoging van rechten

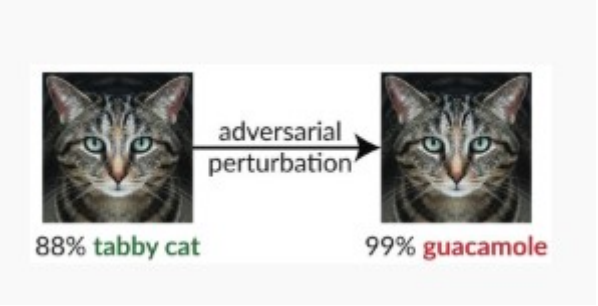
Ernstigheid

Kritisch

Variant #1c: Willekeurige misclassificatie

Dit is een speciale variatie waarbij de doelclassificatie van de aanvaller iets anders kan zijn dan de legitieme bronclassificatie. De aanval omvat over het algemeen het willekeurig injecteren van ruis in de brongegevens die worden geclassificeerd om de kans te verminderen dat de juiste classificatie in de toekomst wordt gebruikt [3].

Voorbeelden



Maatregelen

Hetzelfde als variant 1a.

Traditionele parallellen

Tijdelijke weigering van service

Ernstigheid

Belangrijk

Variant #1d: Betrouwbaarheidsvermindering

Een aanvaller kan invoer maken om het betrouwbaarheidsniveau van de juiste classificatie te verminderen, met name in scenario's met hoge gevolgen. Dit kan ook de vorm aannemen van een groot aantal false positives die bedoeld zijn om beheerders of monitoringsystemen te overbelasten met frauduleuze waarschuwingen die niet te onderscheiden zijn van legitieme waarschuwingen [3].

Voorbeelden

Maatregelen

- Naast de acties die in variant #1a worden behandeld, kan gebeurtenisbeperking worden gebruikt om het aantal waarschuwingen van één bron te verminderen.

Traditionele parallellen

Niet-permanente Denial of Service

Ernstigheid

Belangrijk

#2a Gerichte gegevensvergiftiging

Beschrijving

Het doel van de aanvaller is om het machinemodel dat ***is gegenereerd in de trainingsfase*** te besmetten, zodat voorspellingen over nieuwe gegevens worden gewijzigd in de testfase[1]. Bij gerichte vergiftigingsaanvallen wil de aanvaller specifieke voorbeelden verkeerd classificeren om ervoor te zorgen dat specifieke acties worden uitgevoerd of weggelaten.

Voorbeelden

Av-software indienen als malware om de misclassificatie als schadelijk af te dwingen en het gebruik van gerichte AV-software op clientsystemen te elimineren.

Maatregelen

- Anomaliesensoren instellen om dagelijks de gegevensdistributie te monitoren en waarschuwen bij variaties.

  -Meet trainingsgegevensvariatie dagelijks, telemetrie voor scheefheid/drift

- Invoervalidatie, zowel opschoning als integriteitscontrole

- Vergiftiging beïnvloedt afwijkende trainingsvoorbeelden. Twee belangrijke strategieën voor het counteren van deze bedreiging:

  -Gegevens opschonen/valideren: vergiftigingsmonsters verwijderen uit trainingsgegevens
  -Bagging voor het bestrijden van vergiftigingsaanvallen [14]

  -Weigeren-op-Negative-Impact (RONI) bescherming [15]

  -Robuust leren: Kies leeralgoritmen die robuust zijn in aanwezigheid van vergiftigingsmonsters.

  -Een dergelijke benadering wordt beschreven in [21] waarbij auteurs het probleem van gegevensvergiftiging in twee stappen aanpakken: 1) een nieuwe robuuste matrixfactorisatiemethode introduceren om de echte subruimte te herstellen, en 2) nieuwe robuuste principecomponentregressie om adversarial instanties te verwijderen op basis van de basis die in stap (1) is hersteld. Ze beschrijven de noodzakelijke en voldoende voorwaarden voor het succesvol herstellen van de ware subruimte en presenteren een grens voor het verwachte voorspellingsverlies in vergelijking met de grondwaarheid.

Traditionele parallellen

Trojaned host waarbij aanvaller zich op het netwerk blijft bevinden. Training- of configuratiegegevens zijn gecompromitteerd en worden gebruikt/vertrouwd voor het maken van modellen.

Ernstigheid

Kritisch

#2b Gegevensvergiftiging ondiscrimineren

Beschrijving

Het doel is om de kwaliteit/integriteit van de gegevensset die wordt aangevallen te ruïneren. Veel gegevenssets zijn openbaar/niet-vertrouwd/niet-gecureerd, dus dit zorgt voor extra zorgen over de mogelijkheid om dergelijke schendingen van gegevensintegriteit in de eerste plaats te herkennen. Training van onbewust gecompromitteerde gegevens is een rommel erin, rommel eruit situatie. Zodra dit is gedetecteerd, moet de prioritering de omvang bepalen van gegevens die zijn geschonden en deze in quarantaine plaatsen en opnieuw trainen.

Voorbeelden

Een bedrijf scant een bekende en gerenommeerde website om olietermijncontractgegevens te verkrijgen voor het trainen van hun modellen. De website van de gegevensprovider wordt vervolgens gecompromitteerd via EEN SQL-injectieaanval. De aanvaller kan de gegevensset op elk gewenst moment vergiftigen en het model dat wordt getraind, heeft geen idee dat de gegevens zijn besmet.

Maatregelen

Hetzelfde als variant 2a.

Traditionele parallellen

Geverifieerde authenticatie van een denial-of-service-aanval tegen een waardevol systeem

Ernstigheid

Belangrijk

#3 Modelinversie-aanvallen

Beschrijving

De persoonlijke functies die worden gebruikt in machine learning-modellen kunnen worden hersteld [1]. Dit omvat het reconstrueren van persoonlijke trainingsgegevens waartoe de aanvaller geen toegang heeft. Dit wordt ook wel hill climbing-aanvallen genoemd in de biometrische community [16, 17]. Dit wordt bereikt door de invoer te vinden die het geretourneerde betrouwbaarheidsniveau maximaliseert, waarbij de classificatie moet overeenkomen met het doel [4].

Voorbeelden

Figure 1: An image recovered using a new model inversion attack (left) and a training set image of the victim (right). The attacker is given only the person's name and access to a facial recognition system that returns a class confidence score.

[4]

Maatregelen

- Interfaces voor modellen die zijn getraind op basis van gevoelige gegevens hebben sterk toegangsbeheer nodig.
- Frequentielimietquery's die zijn toegestaan per model
- Implementeer poorten tussen gebruikers/bellers en het werkelijke model door invoervalidatie uit te voeren voor alle voorgestelde query's, waarbij niets wordt geweigerd dat niet voldoet aan de definitie van de invoer correctheid van het model en alleen de minimale hoeveelheid informatie die nodig is om nuttig te zijn.

Traditionele parallellen

Gerichte, geheime openbaarmaking van informatie

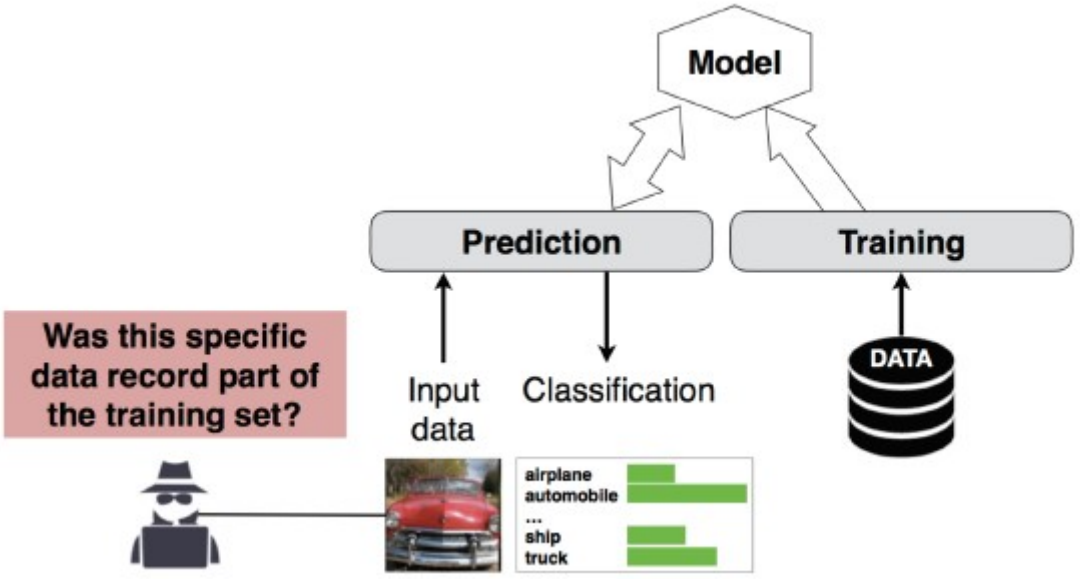
Ernstigheid

Dit is standaard belangrijk volgens de SDL-standaard bugbar, maar gevoelige of persoonlijk identificeerbare gegevens die worden geëxtraheerd, zouden dit tot een kritisch niveau verhogen.

#4 Lidmaatschapsdeductieaanval

Beschrijving

De aanvaller kan bepalen of een bepaalde gegevensrecord deel uitmaakt van de trainingsgegevensset van het model of niet[1]. Onderzoekers konden de belangrijkste procedure van een patiënt voorspellen (bijvoorbeeld operatie die de patiënt doormaakte) op basis van de kenmerken (bijvoorbeeld leeftijd, geslacht, ziekenhuis) [1].



[12]

Maatregelen

Onderzoeksdocumenten die aantonen dat deze aanval levensvatbaar is, geven aan dat Differentiële privacy [4, 9] een effectieve beperking zou zijn. Dit is nog steeds een opkomende plek bij Microsoft en AETHER Security Engineering raadt het bouwen van expertise aan met onderzoeksinvesteringen in deze ruimte. Dit onderzoek moet differentiële privacymogelijkheden opsommen en hun praktische effectiviteit evalueren als oplossingen, en vervolgens manieren ontwerpen om deze verdedigingsmechanismen transparant over te nemen op onze platformen voor onlineservices, vergelijkbaar met hoe het compileren van code

in Visual Studio u on-by-standaardbeveiligingsbeveiligingen biedt die transparant zijn voor de ontwikkelaar en gebruikers.

Het gebruik van neuron dropout en model stacking kan tot op zekere hoogte effectieve mitigaties vormen. Het gebruik van neuron dropout verhoogt niet alleen de tolerantie van een neuraal net voor deze aanval, maar verhoogt ook de modelprestaties [4].

Traditionele parallellen

Gegevensprivacy. Er worden deducties gemaakt over de opname van een gegevenspunt in de trainingsset, maar de trainingsgegevens zelf worden niet openbaar gemaakt

Ernstigheid

Dit is een privacyprobleem, geen beveiligingsprobleem. Het wordt behandeld in richtlijnen voor bedreigingsmodellering omdat de domeinen overlappen, maar elk antwoord hier wordt aangestuurd door privacy, niet door beveiliging.

#5 Model stelen

Beschrijving

De aanvallers maken het onderliggende model opnieuw door legitieme query's uit te voeren op het model. De functionaliteit van het nieuwe model is hetzelfde als die van het onderliggende model[1]. Zodra het model opnieuw is gemaakt, kan het worden omgekeerd om functiegegevens te herstellen of deducties te maken op trainingsgegevens.

- Vergelijking oplossen: voor een model dat klassekansen retourneert via API-uitvoer, kan een aanvaller query's maken om onbekende variabelen in een model te bepalen.
- Padzoeken: een aanval die gebruikmaakt van API-bijzonderheden om de 'beslissingen' te extraheren die door een boom zijn genomen bij het classificeren van een invoer [7].
- Overdrachtsaanval: Een tegenstander kan een lokaal model trainen, bijvoorbeeld door voorspellingsaanvragen te doen aan het doelmodel, en dit gebruiken om kwaadaardige voorbeelden te maken die overdragen naar het doelmodel [8]. Als uw model werd geëxtraheerd en kwetsbaar blijkt voor een soort adversarial invoer, kunnen nieuwe aanvallen op uw productiemodel volledig offline worden ontwikkeld door de aanvaller die een kopie van uw model heeft geëxtraheerd.

Voorbeelden

In instellingen waarin een ML-model dienst doet om adversarial gedrag te detecteren, zoals het identificeren van spam, malwareclassificatie en detectie van netwerkafwijkingen, kan modelextractie aanvallen mogelijk maken [7].

Maatregelen

Proactieve/beschermende acties

- Minimaliseer of verdoezel de details die worden geretourneerd in voorspellings-API's, terwijl ze nog steeds nuttig blijven voor 'eerlijke' toepassingen [7].
- Definieer een goed opgemaakte query voor uw modelinvoer en retourneer alleen resultaten als reactie op voltooide, goed gevormde invoer die overeenkomt met die indeling.
- Geef afgeronde betrouwbaarheidswaarden terug. De meeste legitieme bellers hebben niet meerdere decimalen met precisie nodig.

Traditionele parallellen

Niet-geverifieerde, alleen-lezen manipulatie van systeemgegevens, gerichte openbaarmaking van informatie met hoge waarde?

Ernstigheid

Belangrijk in beveiligingsgevoelige modellen, anders gemiddeld

#6 Herprogrammering van Neurale Netwerken

Beschrijving

Door middel van een speciaal gemaakte query van een kwaadwillende, kunnen Machine Learning-systemen opnieuw worden geprogrammeerd naar een taak die afwijkt van de oorspronkelijke intentie van de maker [1].

Voorbeelden

Zwakke toegangscontroles van een API voor gezichtsherkenning waardoor derde partijen apps kunnen maken die zijn ontworpen om Microsoft-klanten te schaden, zoals een deepfakegenerator.

Maatregelen

- Sterke client-server<> wederzijdse verificatie en toegangsbeheer voor modelinterfaces
- Het verwijderen van de aanstootgevende accounts.
- Een service level agreement voor uw API's identificeren en afdwingen. Bepaal de acceptabele time-to-fix voor een probleem dat eenmaal is gerapporteerd en zorg ervoor dat het probleem niet meer opnieuw wordt weergegeven zodra de SLA is verlopen.

Traditionele parallellen

Dit is een misbruikscenario. U hebt minder kans om een beveiligingsincident op dit te openen dan u gewoon het account van de dader uitschakelt.

Ernstigheid

Van belangrijk tot kritiek

#7 adversarial voorbeeld in het fysieke domein (bits->atomen)

Beschrijving

Een adversarieel voorbeeld is een invoer/query van een schadelijke entiteit, bedoeld om het machine learning systeem te misleiden [1]

Voorbeelden

Deze voorbeelden kunnen zich in het fysieke domein manifesteren, zoals een zelfrijdende auto die wordt misleid om een stopteken uit te voeren vanwege een bepaalde kleur van licht (de adversarial invoer) die op het stopteken wordt weergegeven, waardoor het systeem voor afbeeldingsherkenning het stopteken niet meer als stopteken kan zien.

Traditionele parallellen

Privilege-escalatie, externe code-uitvoering

Maatregelen

Deze aanvallen komen tot uiting omdat problemen in de machine learning-laag (de gegevens- en algoritmelaag onder ai-gestuurde besluitvorming) niet zijn verzacht. Net als bij andere software *of* fysieke systemen, kan de laag onder het doel altijd worden aangevallen via traditionele vectoren. Daarom zijn traditionele beveiligingsprocedures belangrijker dan ooit, met name met de laag van ongemitteerde beveiligingsproblemen (de gegevens/algo-laag) die worden gebruikt tussen AI en traditionele software.

Ernstigheid

Kritisch

#8 Schadelijke ML-providers die trainingsgegevens kunnen herstellen

Beschrijving

Een kwaadwillende provider presenteert een algoritme met een achterdeur, waarbij de persoonlijke trainingsgegevens worden hersteld. Ze konden gezichten en teksten reconstrueren, gezien het model alleen.

Traditionele parallellen

Gerichte openbaarmaking van informatie

Maatregelen

Onderzoeksdocumenten die de levensvatbaarheid van deze aanval demonstreren, geven aan dat homomorfe versleuteling een effectieve beperking zou zijn. Dit is een gebied met weinig huidige investeringen bij Microsoft en AETHER Security Engineering raadt het bouwen van expertise aan met onderzoeksinvesteringen in deze ruimte. Dit onderzoek moet de principes van homomorfe encryptie opsommen en hun praktische effectiviteit evalueren als maatregelen tegenover kwaadwillende ML-as-a-Service-providers.

Ernstigheid

Belangrijk als gegevens PII zijn, anders gemiddeld

#9 Aanval op de ML-toeleveringsketen

Beschrijving

Vanwege grote resources (gegevens en berekeningen) die nodig zijn voor het trainen van algoritmen, is de huidige praktijk het hergebruiken van modellen die zijn getraind door grote bedrijven en deze enigszins wijzigen voor taken (bijvoorbeeld: ResNet is een populair model voor afbeeldingsherkenning van Microsoft). Deze modellen worden samengesteld in een Model Zoo (Caffe host populaire modellen voor afbeeldingsherkenning). Bij deze aanval valt de aanvaller de modellen aan die in Caffe worden gehost, waardoor de bron voor iedereen wordt vergiftigd. [1]

Traditionele parallellen

- Inbreuk op niet-beveiligingsafhankelijkheid van derden
- App Store host onbewust malware.

Maatregelen

- Minimaliseer waar mogelijk afhankelijkheden van derden voor modellen en gegevens.
- Neem deze afhankelijkheden op in uw threat modeling-proces.
- Maak gebruik van sterke verificatie, toegangsbeheer en versleuteling tussen 1st/3rd-party systemen.

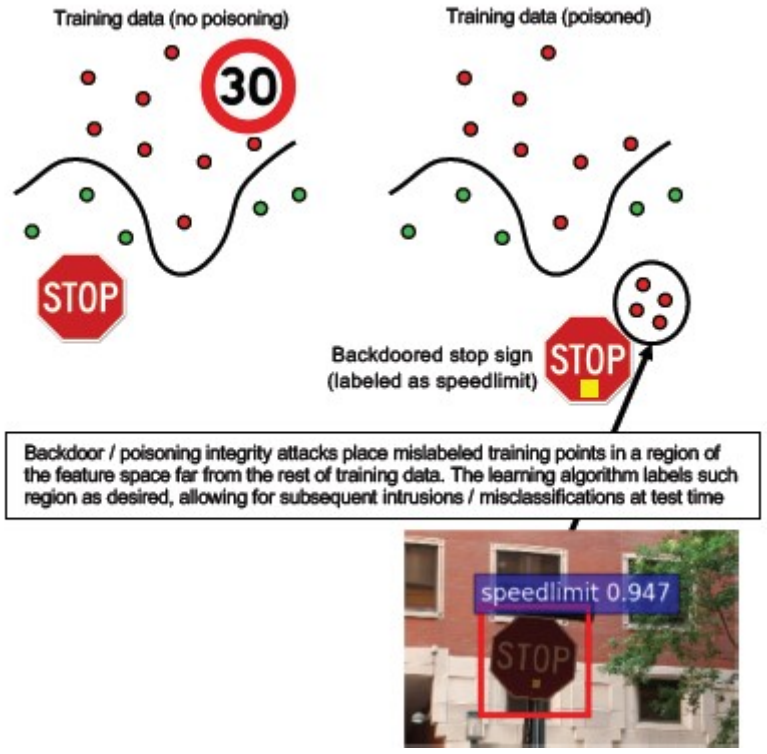
Ernstigheid

Kritisch

#10 Backdoor Machine Learning

Beschrijving

Het trainingsproces wordt uitbesteed aan een kwaadwillende derde partij die knoeit met trainingsgegevens en een trojaans model levert dat gerichte misclassificaties dwingt, zoals het classificeren van een bepaald virus als niet-schadelijk[1]. Dit is een risico in ML-as-a-Service-modelgeneratiescenario's.



[12]

Traditionele parallellen

- Inbreuk op beveiligingsafhankelijkheid van derden
- Gecompromitteerd mechanisme voor software-updates
- Inbreuk op certificeringsinstantie

Maatregelen

Reactieve/defensieve detectieacties

- De schade is al aangericht zodra deze bedreiging is ontdekt, dus het model en eventuele trainingsgegevens die door de kwaadwillende provider worden geleverd, zijn niet te vertrouwen.

Proactieve/beschermende acties

- Alle gevoelige modellen intern trainen
- Trainingsgegevens catalogiseren of ervoor zorgen dat deze afkomstig zijn van een vertrouwde derde partij met sterke beveiligingsprocedures
- Bedreigingsmodel voor de interactie tussen de MLaaS-provider en uw eigen systemen

Antwoordacties

- Hetzelfde als voor inbreuk op externe afhankelijkheid

Ernstigheid

Kritisch

#11 Softwareafhankelijkheden van het ML-systeem misbruiken

Beschrijving

Bij deze aanval bewerkt de aanvaller de algoritmen NIET. In plaats daarvan misbruikt u softwareproblemen, zoals bufferoverloop of cross-site scripting[1]. Het is nog steeds eenvoudiger om softwarelagen onder AI/ML in gevaar te krijgen dan de leerlaag rechtstreeks aan te vallen, dus traditionele beveiligingsrisicobeperkingsprocedures die in de levenscyclus van beveiligingsontwikkeling worden beschreven, zijn essentieel.

Traditionele parallellen

- Gecompromitteerde opensource-softwareafhankelijkheid
- Beveiligingsprobleem met webservers (XSS, CSRF, API-invoervalidatiefout)

Maatregelen

Werk samen met uw beveiligingsteam om de toepasselijke best practices voor security development lifecycle/operational security assurance te volgen.

Ernstigheid

Veranderlijk; Tot kritiek, afhankelijk van het type traditionele softwareprobleem.

Bibliografie

[1] Foutenmodi in Machine Learning, Ram Shankar Siva Kumar, David O'Brien, Kendra Albert, Salome Viljoen en Jeffrey Snover, https://learn.microsoft.com/security/failure-modes-in-machine-learning

[2] AETHER Security Engineering Workstream, Data Provenance/Lineage v-team

[3] Adversariële Voorbeelden in Deep Learning: Characterisatie en Divergentie, Wei, et al, https://arxiv.org/pdf/1807.00051.pdf

[4] ML-Leaks: Model and Data Independent Membership Inference Attacks and Defenses on Machine Learning Models, Salem, et al, https://arxiv.org/pdf/1806.01246v2.pdf

[5] M. Fredrikson, S. Jha en T. Ristenpart, "Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures," in Proceedings van de 2015 ACM SIGSAC Conferentie over Computer- en Communicatiebeveiliging (CCS).

[6] Nicolas Papernot & Patrick McDaniel- Adversariële Voorbeelden in Machinaal Leren AIWTB 2017

[7] Machine Learning-modellen stelen via voorspellings-API's, Florian Tramèr, École Polytechnique Fédérale de Lausanne (EPFL); Fan Zhang, Cornell University; Ari Juels, Cornell Tech; Michael K. Reiter, The University of North Carolina at Chapel Hill; Thomas Ristenpart, Cornell Tech

[8] De ruimte van overdraagbare adversarial voorbeelden, Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh en Patrick McDaniel

[9] Begrijpen van Lidmaatschap Inferences op Well-Generalized Leer Modellen Yunhui Long1, Vincent Bindschaedler1, Lei Wang2, Diyue Bu2, Xiaofeng Wang2, Haixu Tang2, Carl A. Gunter1 en Kai Chen3,4

[10] Simon-Gabriel et al., Adversariële kwetsbaarheid van neurale netwerken neemt toe met de invoerdimensie, ArXiv 2018;

[11] Lyu et al., Een uniforme gradiënt regularisatie familie voor adversariale voorbeelden, ICDM 2015

[12] Wilde patronen: Tien jaar na de opkomst van adversarial Machine Learning - NeCS 2019 Battista Biggioa, Fabio Roli

[13] Adversarial Robuuste Malwaredetectie met Monotone Classificatie Inigo Incer et al.

[14] Battista Biggio, Igino Corona, Giorgio Fumera, Giorgio Giacinto, en Fabio Roli. Bagging-classificaties voor het bestrijden van vergiftigingsaanvallen in tegenstander-classificatietaken.

[15] Een Verbeterde Afwijzing van Negatieve Invloed Verdediging Hongjiang Li en Patrick P.K. Chan

[16] Adler. Beveiligingsproblemen in biometrische versleutelingssystemen. 5e Internationale Conferentie AVBPA, 2005

[17] Galbally, McCool, Fierrez, Marcel, Ortega-Garcia. Op de kwetsbaarheid van gezichtsverificatiesystemen voor hill-climbing aanvallen. Patt. Rec., 2010

[18] Weilin Xu, David Evans, Yanjun Qi. Functiesamenknijpen: Het detecteren van vijandige voorbeelden in diepe neurale netwerken. 2018 Network and Distributed System Security Symposium. 18-21 februari.

[19] Versterking van adversarial robuustheid met behulp van door adversarial training geïnduceerd modelvertrouwen - Xi Wu, Uyeong Jang, Jiefeng Chen, Lingjiao Chen, Somesh Jha

[20] Attributiegestuurde causale analyse voor detectie van adversariële voorbeelden, Susmit Jha, Sunny Raj, Steven Fernandes, Sumit Kumar Jha, Somesh Jha, Gunjan Verma, Brian Jalaian, Ananthram Swami

[21] Robuuste lineaire regressie tegen trainingsgegevensvergiftiging – Chang Liu et al.

[22] Kenmerkruisvervaging voor verbetering van tegenstrijdige robuustheid, Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, Kaiming He

[23] Gecertificeerde verdediging tegen Adversarial Examples - Aditi Raghunathan, Jacob Steinhardt, Percy Liang

AI-/ML-pivot naar de Buggrens van de Beveiliging Ontwikkelingscyclus

Door Andrew Marshall, Jugal Parikh, Emre Kiciman en Ram Shankar Siva Kumar

November 2019

Dit artikel is een product van de Microsoft AETHER Engineering Practices for AI Working Group. Dit artikel fungeert als aanvulling op de bestaande SDL-bugbalk die wordt gebruikt voor het opsporen van traditionele beveiligingsproblemen. Het is bedoeld om te worden gebruikt als referentie voor het triage van beveiligingsproblemen met betrekking tot AI/ML. De classificatie ernst van beveiligingsproblemen voor AI-systemen (gepubliceerd door Microsoft Security Response Center), definieert algemene typen beveiligingsproblemen en ernstniveaus voor systemen met AI.

Deze richtlijnen zijn georganiseerd rond de adversarial Machine Learning Threat Taxonomy, gemaakt door Ram Shankar Siva Kumar, David O'Brien, Kendra Albert, Salome Viljoen en Jeffrey Snover, en getiteld Failure Modes in Machine Learning. Hoewel het onderzoek naar deze inhoud is gebaseerd op zowel opzettelijk/schadelijk als onopzettelijk gedrag in ML-foutmodi, richt deze foutbalk zich volledig op opzettelijk/schadelijk gedrag dat zou leiden tot een beveiligingsincident en/of implementatie van een oplossing.

Tabel uitvouwen

Bedreiging	Beschrijving/bedrijfsrisico's/voorbeelden
Datavergiftiging	Beschadigde trainingsgegevens: het einddoel van de aanvaller is om het machinemodel dat is gegenereerd *in de trainingsfase* te besmetten, zodat voorspellingen over nieuwe gegevens in de testfase worden gewijzigd. Bij gerichte verontreinigingsaanvallen wil de aanvaller specifieke voorbeelden verkeerd classificeren om ervoor te zorgen dat specifieke acties worden uitgevoerd of nagelaten. AV-software als malware indienen om de software onjuist te laten classificeren als schadelijke software om zo het gebruik van gerichte AV-software op clientsystemen te elimineren. Een bedrijf scrapet een bekende en vertrouwde website op toekomstige gegevens om zijn modellen te trainen. De website van de gegevensprovider wordt vervolgens gecompromitteerd via EEN SQL-injectieaanval. De aanvaller kan de gegevensset naar believen vervuilen en het model dat wordt getraind heeft geen besef dat de gegevens zijn besmet.
Modeldiefstal	Recreatie van het onderliggende model door legitiem query's erop uit te voeren. De functionaliteit van het nieuwe model is hetzelfde als die van het

Bedreiging	Beschrijving/bedrijfsrisico's/voorbeelden
	onderliggende model. Zodra het model opnieuw is gemaakt, kan het worden omgekeerd om informatie over het kenmerk te herstellen of om trainingsgegevens te deduceren. Vergelijkingen oplossen – Voor een model dat klassewaarschijnlijkheden retourneert via API-uitvoer, kan een aanvaller query's maken om onbekende variabelen in een model te bepalen. Path Finding: een aanval die gebruikmaakt van API-bijzonderheden om de 'beslissingen' te extraheren die door een boomstructuur worden genomen bij het classificeren van een invoer. Overdrachtsaanval – Een kwaadwillend iemand kan een lokaal model trainen, mogelijk door voorspellingsquery's naar het doelmodel te verzenden en dit te gebruiken om schadelijke voorbeelden te maken die worden overgedragen op het doelmodel. Als uw model geëxtraheerd is en kwetsbaar blijkt te zijn voor een bepaald type schadelijke invoer, kunnen nieuwe aanvallen tegen uw productie-implementatiemodel volledig offline worden ontwikkeld door de aanvaller die een kopie van uw model heeft geëxtraheerd. In instellingen waarbij een ML-model wordt gebruikt om schadelijk gedrag te detecteren, zoals het identificeren van spam, het classificeren van malware en het detecteren van netwerkafwijkingen, kan modelextractie leiden tot fraudeaanvallen
Modelinversie	De privéfuncties die in machine learning-modellen worden gebruikt, kunnen worden hersteld. Dit omvat het reconstrueren van persoonlijke trainingsgegevens waartoe de aanvaller geen toegang heeft. Dit wordt bereikt door de invoer te vinden die het geretourneerde betrouwbaarheidsniveau maximaliseert, afhankelijk van de classificatie die met het doel overeenkomt. Voorbeeld: Reconstructie van gezichtsherkenningsgegevens van geraden of bekende namen en API-toegang om een query uit te voeren op het model.
Adversarieel voorbeeld in fysieke domein	Deze voorbeelden kunnen zich in het fysieke domein bevinden, zoals een zelfrijdende auto die wordt misleid om een stopteken uit te voeren vanwege een bepaalde kleur van licht (de indringerinvoer) die op het stopteken wordt weergegeven, waardoor het systeem voor afbeeldingsherkenning het stopteken niet meer als stopteken kan zien.
ML-toeleveringsketen aanvallen	Vanwege grote resources (gegevens en berekeningen) die nodig zijn voor het trainen van algoritmen, is de huidige praktijk het hergebruiken van modellen die zijn getraind door grote bedrijven en deze enigszins wijzigen voor taken (bijvoorbeeld: ResNet is een populair model voor afbeeldingsherkenning van Microsoft). Deze modellen zijn ondergebracht in een modelzoo (Caffe hostt populaire modellen voor afbeeldingsherkenning).

Bedreiging	Beschrijving/bedrijfsrisico's/voorbeelden
	Bij deze aanval valt de kwaadwillende persoon de in Caffe gehoste modellen aan, waardoor de bron voor anderen wordt verontreinigd.
Gemanipuleerd algoritme van schadelijke ML-provider	Het onderliggende algoritme in gevaar brengen Een schadelijke ML-as-a-service-provider stelt een gemanipuleerd algoritme voor waarin de privé-trainingsgegevens worden hersteld. Dit biedt de aanvaller de mogelijkheid om slechts op basis van het model gevoelige gegevens, zoals gezichten en teksten, te reconstrueren.
Neural Net herprogrammeren	Met een speciaal gemaakte query van een aanvaller kunnen ML-systemen opnieuw worden geprogrammeerd naar een taak die afwijkt van de oorspronkelijke intentie van de maker Zwake toegangscontroles op een gezichtsherkennings-API die derden in staat stellen te integreren in apps die zijn ontworpen om gebruikers te schaden, zoals een deepfakegenerator. Dit is een scenario voor misbruik/accountverwijdering
Adversarial perturbatie	Bij verstorende aanvallen wijzigt de aanvaller heimelijk de query om een gewenste reactie te krijgen van een *productie-implementatiemodel*. Dit is een schending van modelinvoerintegriteit die leidt tot fuzzing-style aanvallen waarbij het eindresultaat niet noodzakelijkerwijs een toegangsschending of EOP is. In plaats daarvan worden de classificatieprestaties van het model aangetast. Dit kan worden gemanifesteerd door trollen die bepaalde doelwoorden gebruiken op een manier die de AI hen verbiedt, waardoor de service effectief wordt geweigerd aan legitieme gebruikers met een naam die overeenkomt met een 'verboden' woord. Afdwingen dat goedaardige e-mails worden geclassificeerd als spam of ervoor zorgen dat een schadelijk voorbeeld onopgemerkt blijft. Deze aanvallen worden ook wel modelontwijkings- of imitatieaanvallen genoemd. De aanvaller kan ingangen maken om het betrouwbaarheidsniveau van de juiste classificatie te verminderen, vooral in scenario's met grote gevolgen. Dit kan ook de vorm aannemen van een groot aantal false positives dat is bedoeld om beheerders of bewakingssystemen te overbelasten met frauduleuze waarschuwingen die niet te onderscheiden zijn van legitieme waarschuwingen.
Lidmaatschapsinferentie	Het lidmaatschap van een individu afleiden uit een groep die wordt gebruikt om een model te trainen Bijvoorbeeld: voorspelling van chirurgische ingrepen op basis van leeftijd/geslacht/ziekenhuis

Last updated on 26-03-2026

De toekomst van kunstmatige intelligentie en Machine Learning veilig stellen bij Microsoft

Artikel • 12-03-2025

Door Andrew Marshall, Raul Rojas, Jay Stokes en Donald Brinkman

Met speciale dank aan Mark Cartwright en Graham Calladine

Samenvatting

Kunstmatige intelligentie (AI) en Machine Learning (ML) hebben al een grote invloed op hoe mensen werken, socialiseren en hun leven leiden. Naarmate het verbruik van producten en services die zijn gebouwd rond AI/ML toeneemt, moeten gespecialiseerde acties worden ondernomen om niet alleen uw klanten en hun gegevens te beveiligen, maar ook om uw AI en algoritmen te beschermen tegen misbruik, trolling en extractie. In dit document wordt aandacht besteed aan enkele van de beveiligingslessen die Microsoft heeft getrokken uit het ontwerpen van producten en het exploiteren van onlineservices die zijn gebouwd op AI. Hoewel het moeilijk is om te voorspellen hoe dit gebied zich ontwikkelt, hebben we geconcludeerd dat er actie-bare problemen zijn om nu op te lossen. Daarnaast hebben we geconstateerd dat er strategische problemen zijn die de tech-industrie moet aanpakken om de beveiliging van klanten op de lange termijn te garanderen, evenals de beveiliging van hun gegevens.

Dit document gaat niet over op AI gebaseerde aanvallen of zelfs AI die wordt gebruikt door menselijke aanvallers. In plaats daarvan richten we ons op problemen die microsoft- en branchepartners moeten aanpakken om ai-producten en -services te beschermen tegen zeer geavanceerde, creatieve en kwaadaardige aanvallen, ongeacht of ze worden uitgevoerd door afzonderlijke trollen of hele wolfspakketten.

Dit document richt zich volledig op beveiligingstechnische problemen die uniek zijn voor de AI/ML-ruimte, maar vanwege de uitgebreide aard van het InfoSec-domein wordt begrepen dat problemen en bevindingen die hier worden besproken, elkaar overlappen met de domeinen van privacy en ethiek. Aangezien dit document ingaat op uitdagingen van strategisch belang voor de technische bedrijfstak, is de doelgroep voor dit document leadership voor beveiligingstechniek binnen alle branches.

Onze vroege bevindingen suggereren dat:

- AI/ML-specifieke aanpassingen van bestaande beveiligingsprocedures vereist zijn om de typen beveiligingsproblemen te verhelpen die in dit document worden besproken.

- Machine Learning-modellen in hoofdlijnen niet in staat zijn om onderscheid te maken tussen kwaadwillende invoer en onschadelijke gegevens die afwijken van de norm. Een belangrijke bron van trainingsgegevens is afgeleid van niet-gecureerde, niet-gemodereerde openbare gegevenssets, die open zijn voor 3[externe] bijdragen. Aanvallers hoeven geen inbreuk te maken op gegevenssets wanneer ze er gratis aan kunnen bijdragen. Na verloop van tijd worden schadelijke gegevens met een lage betrouwbaarheid vertrouwde gegevens, als de gegevensstructuur/opmaak correct blijft.

- Gezien het grote aantal lagen van verborgen classificaties/neuronen die kunnen worden gebruikt in een Deep Learning-model, wordt te veel vertrouwen gelegd op de uitvoer van AI/ML-besluitvormingsprocessen en -algoritmen zonder kritisch inzicht in hoe deze beslissingen zijn bereikt. Deze vertroebeling maakt het onmogelijk om 'uw werk te laten zien' en maakt het lastig om bevindingen van AI/ML aantoonbaar te verdedigen wanneer deze in twijfel worden geroepen.

- AI/ML wordt steeds vaker gebruikt ter ondersteuning van belangrijke besluitvormingsprocessen in de geneeskunde en andere branches, waarbij de verkeerde beslissing kan leiden tot ernstige letsel of zelfs de dood. Een gebrek aan forensische rapportagemogelijkheden in AI/ML voorkomt dat deze belangrijke conclusies verdedigbaar zijn in zowel de rechtszaal als in de publieke opinie.

De doelstellingen van dit document zijn om (1) beveiligingstechnische problemen te benadrukken, die uniek zijn voor de AI/ML-ruimte, (2) geven enkele eerste gedachten en waarnemingen over opkomende bedreigingen aan en (3) delen vroege gedachten over mogelijke herstel. Enkele van de uitdagingen in dit document zijn problemen die de branche in de komende twee jaar moet oplossen, andere zijn problemen die we zo snel mogelijk achter ons moeten laten. Zonder dieper onderzoek naar de gebieden die in dit document worden behandeld, lopen we het risico dat AI een zwarte doos wordt door ons onvermogen om AI-besluitvormingsprocessen op wiskundig niveau te vertrouwen of te begrijpen (en indien nodig te wijzigen). Vanuit een veiligheidsperspectief betekent dit effectief verlies van controle en een vertrek uit de leidende principes van Microsoft op het gebied van kunstmatige intelligentie [3, 7].

Nieuwe uitdagingen voor beveiligingstechniek

Traditionele aanvalsvectoren van software zijn nog steeds essentieel om aan te pakken, maar ze bieden niet voldoende dekking in het AI/ML-bedreigingslandschap. De tech-

industrie moet problemen van de volgende generatie niet bestrijden met oplossingen van de vorige generatie door nieuwe frameworks te bouwen en nieuwe benaderingen te omarmen die zich richten op hiaten in het ontwerp en de werking van services die op AI/ML zijn gebaseerd:

1. Zoals hieronder wordt beschreven, moeten veilige fundamenten voor ontwikkeling en uitvoering gebruikmaken van de concepten van tolerantie en discretie bij het beveiligen van AI en de gegevens onder controle van AI. AI-specifieke aanpassingen zijn vereist op het gebied van verificatie, scheiding van plichten, invoervalidatie en Denial of Service-mitigatie. Zonder investeringen op deze gebieden blijven AI/ML-services vechten tegen kwaadwillende tegenstanders van alle vaardigheidsniveaus.

2. AI moet in staat zijn om vooroordelen bij anderen te herkennen, zonder zelf vooroordelen te hanteren in de interactie met mensen. Hiervoor is een collectieve en zich steeds ontwikkelende kennis nodig van vooroordelen, stereotypen, specifiek taalgebruik en andere culturele concepten. Een dergelijk begrip helpt AI te beschermen tegen aanvallen van social engineering en manipulatie van datasets. Een correct geïmplementeerd systeem wordt sterker van dergelijke aanvallen en kan zijn uitgebreide kennis delen met andere AIS's.

3. Machine Learning-algoritmen moeten in staat zijn om kwaadwillende geïntroduceerde gegevens te onderscheiden van goedaardige 'Black Swan'-gebeurtenissen [1] door trainingsgegevens met negatieve gevolgen voor de resultaten te weigeren. Anders zijn leermodellen altijd vatbaar voor gaming door aanvallers en trollen.

4. AI moet over ingebouwde forensische mogelijkheden beschikken. Hierdoor kunnen ondernemingen klanten transparantie en verantwoordelijkheid bieden voor hun AI, zodat hun acties niet alleen verifieerbaar correct zijn, maar ook juridisch verdedigbaar zijn. Deze mogelijkheden functioneren ook als een vroege vorm van 'detectie van AI-indringing', zodat technici het exacte tijdstip kunnen bepalen dat een beslissing is genomen door een classificatie, welke gegevens hierop van invloed zijn geweest en of die gegevens betrouwbaar zijn. De mogelijkheden voor gegevensvisualisatie op dit gebied zijn snel vooruit en tonen de belofte om technici te helpen de hoofdoorzaken voor deze complexe problemen te identificeren en op te lossen [10].

5. AI moet gevoelige gegevens herkennen en beveiligen, zelfs als mensen dit niet zien. Rijke gebruikerservaringen in AI vereisen grote hoeveelheden onbewerkte gegevens nodig om op te trainen, dus moet er rekening worden gehouden met 'over-delen' door klanten.

Elk van deze gebieden, met inbegrip van bedreigingen en mogelijke oplossingen, wordt hieronder uitvoerig besproken.

AI vereist nieuwe aanpassingen voor traditionele modellen voor veilig ontwerpen/veilige exploitatie: de introductie van tolerantie en discretie

AI-ontwerpers moeten de vertrouwelijkheid, integriteit en beschikbaarheid van gevoelige gegevens garanderen, dat het AI-systeem vrij is van bekende beveiligingsproblemen en controles bieden voor de beveiliging, detectie en reactie op schadelijk gedrag tegen het systeem of de gegevens van de gebruiker.

De traditionele manieren om te beschermen tegen schadelijke aanvallen bieden niet dezelfde dekking in dit nieuwe paradigma, waarbij aanvallen op basis van spraak/video/afbeeldingen huidige filters en verdediging kunnen omzeilen. Nieuwe aspecten van bedreigingsmodellering moeten worden verkend om te voorkomen dat er nieuwe beveiligingsproblemen ontstaan door misbruik van onze AI. Dit gaat veel verder dan het identificeren van het traditionele aanvalsoppervlak door middel van fuzzing of invoermanipulatie (deze aanvallen hebben ook hun eigen AI-specifieke varianten). Het vereist het integreren van scenario's die uniek zijn voor het AI/ML-domein. Belangrijk hierbij zijn AI-gebruikerservaringen zoals spraak, video en gebaren. De bedreigingen die aan deze ervaringen zijn gekoppeld, zijn niet traditioneel gemodelleerd. Video-inhoud wordt nu bijvoorbeeld aangepast om fysieke effecten op te roepen. Daarnaast laat onderzoek zien dat op audio gebaseerde aanvalsopdrachten kunnen worden gemaakt [9].

De onvoorspelbaarheid, creativiteit en schadelijke intenties van criminelen, vastberaden aanvallers en trollen vereisen dat wij onze AI's uitbreiden met de waarden van **tolerantie** en **discretie**:

Tolerantie: Het systeem moet abnormaal gedrag kunnen identificeren en manipulatie of dwang buiten de normale grenzen van acceptabel gedrag met betrekking tot het AI-systeem en de specifieke taak kunnen voorkomen. Dit zijn nieuwe typen aanvallen die specifiek zijn voor het AI/ML-domein. Systemen moeten zo worden ontworpen dat ze invoer weigeren die anderszins een conflict zou opleveren met lokale wetgeving, ethiek en waarden en normen die leven binnen de gemeenschap en bij de makers. Dit betekent dat AI moet beschikken over de mogelijkheid om vast te stellen wanneer een interactie 'off-script' gaat. Dit kan worden bereikt met de volgende methoden:

1. Stel individuele gebruikers vast die afwijken van de normen die zijn ingesteld door de verschillende grote clusters van vergelijkbare gebruikers, bijvoorbeeld gebruikers die te snel lijken te typen, te snel reageren, niet in slaapstand of onderdelen van het systeem activeren die andere gebruikers niet gebruiken.

2. Patronen van gedrag opsporen waarvan bekend is dat ze indicatoren zijn van kwaadaardige aanvallen en het begin van de Network Intrusion Kill Chain.

3. Herkennen wanneer meerdere gebruikers op een gecoördineerde manier handelen; Meerdere gebruikers geven bijvoorbeeld allemaal dezelfde onverklaarbare maar opzettelijk gemaakte query, plotselinge pieken in het aantal gebruikers of plotselinge pieken in de activering van specifieke onderdelen van een AI-systeem.

Aanvallen van dit type moeten worden overwogen in combinatie met Denial of Service-aanvallen, omdat de AI mogelijk bugfixes en opnieuw trainen vereist om niet opnieuw te vallen voor dezelfde trucs. Van cruciaal belang is de mogelijkheid om schadelijke intenties te identificeren in de aanwezigheid van tegenmaatregelen, zoals maatregelen die worden gebruikt om sentimentanalyse-API's te verslaan [4].

Discretie: AI moet een verantwoordelijke en betrouwbare beheerder zijn van *alle* informatie waar hij toegang toe heeft. Als mens wijzen we ongetwijfeld een bepaald vertrouwensniveau toe aan onze AI-relaties. Op een bepaald moment zullen deze agents namens ons communiceren met andere agents of andere mensen. We moeten erop kunnen vertrouwen dat een AI-systeem voldoende discreet is om alleen in beperkte vorm te delen wat over ons moet worden gedeeld, zodat andere agents namens het systeem taken kunnen uitvoeren. Bovendien mogen meerdere agenten namens ons interactie hebben met persoonlijke gegevens, niet elke persoon globale toegang tot deze gegevens nodig heeft. Scenario's voor gegevenstoegang waarbij meerdere AI's of bot-agents betrokken zijn, moeten de levensduur van de toegang tot de vereiste minimale duur beperken. Gebruikers moeten ook gegevens kunnen weigeren en de verificatie van agents van specifieke bedrijven of landinstellingen weigeren, net zoals webbrowsers siteblokkering vandaag toestaan. Het oplossen van dit probleem vereist een nieuwe benadering van verificatie tussen agents en bevoegdheden voor gegevenstoegang, zoals de investeringen in cloudverificatie van gebruikers in de vroege jaren van cloud-computing.

AI moet in staat zijn om vooroordelen bij anderen te herkennen, zonder zelf bevooroordeeld te zijn.

Hoewel AI eerlijk en inclusief moet zijn zonder discriminerend te zijn ten aanzien van een bepaalde groep personen of geldige uitkomsten, is een aangeboren begrip van vooroordelen noodzakelijk om dit mogelijk te maken. Als AI niet is getraind om vooringenomenheid, trolling of sarcasme te herkennen, kan het worden bedrogen door degenen die op zoek zijn naar goedkope lachen op zijn best, of, in het ergste geval, schade toebrengen aan klanten.

Het bereiken van dit niveau van bewustzijn is alleen mogelijk als 'goede mensen AI slechte dingen leren', omdat hiervoor een uitgebreide en meegroeiende kennis van culturele vooroordelen vereist is. AI moet een gebruiker kunnen herkennen met wie het in het verleden negatieve interacties had en wees voorzichtig, vergelijkbaar met hoe ouders hun kinderen leren om voorzichtig te zijn met vreemden. De beste manier om dit aan te pakken, is door het AI-systeem voorzichtig bloot te stellen aan trolls op een gecontroleerde/bewaakte/beperkte manier. Op deze manier kan AI het verschil leren tussen een goedaardige gebruiker die wat wil uitproberen en daadwerkelijk kwaadwillende intenties/trolling. Trolls bieden een waardevolle stroom trainingsgegevens voor AI, waardoor het systeem beter bestand wordt tegen toekomstige aanvallen.

AI moet ook in staat zijn om vooroordelen te herkennen in gegevenssets waarop ze traint. Deze kunnen cultureel of regionaal van aard zijn, met taal die wordt gebruikt door een bepaalde groep mensen, of onderwerpen/meningen die van speciale interesse zijn voor een groep. Net als bij opzettelijk schadelijke trainingsgegevens moet AI bestand zijn tegen de effecten van deze gegevens op zijn eigen inferenties en redeneringen. In essentie is dit een complex probleem van invoervalidatie met overeenkomsten met bereikcontrole. In plaats van om te gaan met bufferlengtes en offsets, zijn buffer- en bereikcontroles woorden met een rode vlag uit een breed aanbod van bronnen. De gespreksgeschiedenis en de context waarin woorden worden gebruikt, zijn ook belangrijk. Net zoals defense-in-depth-practices worden gebruikt om lagen van beveiliging over de front-end van een traditionele webservice-API te leggen, moeten er meerdere beveiligingslagen worden gebruikt voor de herkenning en het voorkomen van vooroordelen.

Machine Learning-algoritmen moeten in staat zijn om kwaadwillende geïntroduceerde gegevens te onderscheiden van goedaardige 'Black Swan'-gebeurtenissen

Er worden talloze whitepapers gepubliceerd over het theoretische potentieel van manipulatie van ML-modellen/classificaties en extractie/diefstal van services waar

aanvallers toegang hebben tot zowel de trainingsgegevensset als een geïnformeerd begrip van het model in gebruik [2, 3, 6, 7]. Het overkoepelende probleem hier is dat alle ML-classificatoren kunnen worden misleid door een aanvaller die controle heeft over de gegevens van de trainingsset. Aanvallers hoeven niet eens de mogelijkheid te hebben om bestaande trainingsgegevens aan te passen; ze hoeven alleen maar gegevens aan de set toe te voegen en ervoor te zorgen dat hun invoer na verloop van tijd 'vertrouwd' wordt, dankzij het onvermogen van de ML-classificator om kwaadaardige gegevens te onderscheiden van echte afwijkende gegevens.

Dit probleem met de toeleveringsketen van de trainingsgegevens brengt ons bij het concept van 'beslissingsintegriteit'; de mogelijkheid om kwaadwillend geïntroduceerde trainingsgegevens of gebruikersinvoer te identificeren en weigeren voordat deze een negatieve invloed hebben op het classificatiegedrag. De reden hiervoor is dat betrouwbare trainingsgegevens een hogere kans hebben om betrouwbare resultaten/beslissingen te genereren. Hoewel het nog steeds van cruciaal belang is om te trainen en bestendig te zijn tegen niet-vertrouwde gegevens, moet de kwaadaardige aard van die gegevens worden geanalyseerd voordat ze deel uitmaken van een corpus van trainingsgegevens met hoge betrouwbaarheid. Zonder dergelijke maatregelen kan AI ertoe worden verleid om overtrokken te reageren op trolling en de toegang tot de service te weigeren aan legitieme gebruikers.

Dit is met name een probleem wanneer leeralgoritmen zonder toezicht worden getraind met behulp van niet-gecureerde of niet-vertrouwde gegevenssets. Dit betekent dat aanvallers alle gegevens die ze willen kunnen introduceren, op voorwaarde dat de indeling correct is en het algoritme hierop is getraind. Feitelijk worden die datapunten evenveel vertrouwd als de rest van de trainingsset. Met genoeg zorgvuldig samengestelde invoer door de aanvaller verliest het trainingsalgoritme het vermogen om ruis en afwijkingen te onderscheiden van zeer vertrouwenswaardige data.

Als een voorbeeld van deze bedreiging nemen we een database van stopborden van over de hele wereld, in elke mogelijke taal. Een dergelijke gegevensset is zeer lastig te cureren vanwege het aantal betrokken afbeeldingen en talen. Kwaadwillende bijdragen aan die gegevensset zouden waarschijnlijk niet worden opgemerkt tot het moment dat zelfrijdende auto's stopborden niet meer herkennen. Gegevensweerbaarheid en integriteit van beslissingen moeten hier hand in hand werken om de schade door schadelijke gegevens te identificeren en te elimineren, zodat het geen kernonderdeel van het leerproces wordt.

AI moet beschikken over ingebouwde forensische gegevens en registratie van

beveiligingsgebeurtenissen om transparantie en aansprakelijkheid te bieden

AI zal uiteindelijk in staat zijn om in een professionele hoedanigheid als een agent namens ons op te treden en ons te helpen bij het nemen van belangrijke beslissingen. Een voorbeeld hiervan kan een AI zijn waarmee financiële transacties kunnen worden verwerkt. Als de AI wordt misbruikt en transacties op een of andere manier worden gemanipuleerd, kunnen de gevolgen variëren van het individu tot het systeem. In hoogwaardige scenario's heeft AI passende forensische en beveiligingslogboekregistratie nodig om integriteit, transparantie, verantwoordelijkheid en in sommige gevallen bewijs te leveren waarbij civiele of strafrechtelijke aansprakelijkheid kan ontstaan.

Essentiële AI-services hebben controle-/gebeurtenistraceringsfaciliteiten nodig op algoritmeniveau, waarbij ontwikkelaars de geregistreerde status van specifieke classificaties kunnen onderzoeken, wat kan hebben geleid tot een onnauwkeurige beslissing. Deze mogelijkheid is nodig in de hele branche om de juistheid en transparantie van door AI gegenereerde beslissingen te bewijzen wanneer deze in twijfel worden getrokken.

Mogelijkheden voor het traceren van gebeurtenissen kunnen beginnen met de correlatie van basisgegevens voor het nemen van beslissingen, zoals:

1. Het tijdsblok waarin de laatste trainingsgebeurtenis zich heeft voorgedaan
2. Het tijdstempel van de meest recente invoer in de gegevensset waarop is getraind
3. Wegingen en vertrouwensniveaus van belangrijke classificaties die worden gebruikt voor het nemen van belangrijke beslissingen
4. De classificaties of onderdelen die betrokken zijn bij de beslissing
5. De uiteindelijke invloedrijke beslissing die is genomen door het algoritme

Dergelijke tracering is overkill voor de meeste door algoritme ondersteunde besluitvorming. Het is echter mogelijk om de gegevenspunten en metagegevens van het algoritme te identificeren die leiden tot specifieke resultaten, zijn van groot belang bij het nemen van beslissingen met een hoge waarde. Dergelijke mogelijkheden tonen niet alleen betrouwbaarheid en integriteit aan door middel van de mogelijkheid van het algoritme om zijn werk te laten zien, maar deze gegevens kunnen ook worden gebruikt voor het verfijnen van gegevens.

Een andere forensische functie die nodig is in AI/ML is de mogelijkheid om manipulatie te detecteren. Net zo als we willen dat ons AI-systeem vooroordelen herkent en hier niet vatbaar voor is, hebben we forensische mogelijkheden nodig die onze technici kunnen helpen bij het detecteren en reageren op dergelijke aanvallen. Dergelijke forensische mogelijkheden zijn van enorme waarde wanneer ze zijn gekoppeld aan technieken voor gegevensvisualisatie [10] waardoor de algoritmen kunnen worden gecontroleerd, foutopsporing en afstemming van algoritmen voor effectievere resultaten.

AI moet gevoelige gegevens beveiligen, zelfs als mensen dat niet doen

Rijke ervaringen vereisen rijke gegevens. Mensen leveren al enorme hoeveelheden gegevens aan waarop machine learning (ML) kan worden getraind. Deze variëren van de inhoud van wachtrijen voor het streamen van alledaagse videobeelden tot trends in creditcardaankopen/transactiegeschiedenissen die worden gebruikt om fraude op te sporen. AI moet een ingesleten gevoel van discretie hebben als het gaat om het verwerken van gebruikersgegevens, altijd handelend om deze te beschermen, zelfs wanneer ze vrijelijk worden gedeeld door een publiek dat te veel informatie prijsgeeft.

Aangezien een AI-systeem een geverifieerde groep van 'peers' kan hebben waarmee wordt gesproken om complexe taken uit te voeren, moet ook het belang worden onderkend van het beperken van de gegevens die worden gedeeld met deze peers.

Vroege waarnemingen met betrekking tot AI en beveiligingsproblemen

Ondanks de beginfase van dit project geloven we dat het tot nu toe verzamelde bewijs aantoont dat grondiger onderzoek naar elk van de onderstaande gebieden essentieel is om onze branche te bewegen naar betrouwbaardere en veiligere AI/ML-producten en -diensten. Hieronder ziet u onze vroege waarnemingen en gedachten over wat we graag zien gebeuren in dit domein.

1. Het ontwikkelen van op AI/ML gerichte penetratietests en een instantie voor beveiligingsevaluatie om ervoor te zorgen dat onze toekomstige AI aansluit bij onze normen en waarden en voldoet aan de Asilomar AI Principles.
    a. Een dergelijke instantie kan ook tools en frameworks ontwikkelen die binnen de gehele branche worden ingezet om services op basis van AI/ML te beveiligen.
    b. Na verloop van tijd zal deze expertise organisch groeien binnen de engineering-groepen, net zoals dat het geval was bij de expertise in traditionele beveiliging gedurende de afgelopen tien jaar.

2. Er kan training worden ontwikkeld die ondernemingen helpt bij het realiseren van doelen zoals het democratiseren van AI en het tegelijkertijd oplossen van de problemen die in dit document aan bod komen.
    a. Specifieke beveiligingstraining voor AI betekent dat technici zich bewust zijn van de risico's **voor** hun AI-systeem en de resources die ze tot hun beschikking hebben. Dit materiaal moet worden geleverd met de huidige training voor het beveiligen van klantgegevens.
    b. Dit is mogelijk zonder dat elke gegevenswetenschapper verplicht moet worden omgeschoold tot beveiligingsexpert. De focus moet liggen op het benadrukken aan ontwikkelaars van het belang van de concepten van tolerantie en discretie zoals deze gelden voor hun gebruiksscenario's van AI.
    c. Ontwikkelaars moeten inzicht krijgen in de veilige 'bouwstenen' van AI-services die opnieuw worden gebruikt in hun onderneming. Er moet nadruk worden gelegd op fouttolerant ontwerp met subsystemen, die eenvoudig kunnen worden uitgeschakeld (bijvoorbeeld afbeeldingsprocessors, tekstparsers).

3. ML-classificaties en hun onderliggende algoritmen kunnen worden beveiligd en geschikt worden gemaakt voor het detecteren van kwaadaardige trainingsgegevens zonder dat deze geldige trainingsgegevens verontreinigen die in gebruik zijn of de resultaten scheeftrekken.

    a. Technieken zoals Weigeren op negatieve invoer [5] hebben onderzoeksrondes nodig.

    b. Dit werk vereist wiskundige verificatie, proof-of-concept in code en testen op zowel kwaadwillende als onschadelijke gegevens die afwijken van de norm.

    c. Spotchecks/controle door mensen kan hier zinvol zijn, met name als er sprake is van statistische afwijkingen.

    d. Er kunnen 'toezichthoudende classificaties' worden ontwikkeld om een meer universeel begrip van bedreigingen te hebben tussen verschillende AI-systemen. Hierdoor wordt de beveiliging van het systeem enorm verbeterd omdat de aanvaller niet meer de mogelijkheid heeft om één bepaald model uit te filteren.

    e. AI-systemen kunnen worden gekoppeld om bedreigingen in gekoppelde systemen te identificeren.

4. Er kan een centrale bibliotheek met controleactiviteiten/forensische gegevens voor ML worden opgezet die als norm fungeert voor de transparantie en betrouwbaarheid van AI.

    a. Er kunnen ook querymogelijkheden worden gebouwd voor het controleren en reconstrueren van beslissingen door AI met grote bedrijfsimpact.

5. Het taalgebruik van bepaalde tegenstanders in verschillende culturele groepen en op social media kan continu worden geïnventariseerd en geanalyseerd door AI om trolling, sarcasme, etc. te kunnen detecteren en verwerken.

    a. AI-systemen moeten tolerant zijn ten aanzien van allerlei soorten taalgebruik, of dit nu technisch of regionaal is of specifiek geldt voor een bepaald forum.

    b. Deze verzameling van kennis kan ook worden gebruikt voor de automatisering van inhoudsfiltering, labeling en blokkeren om schaalbaarheidsproblemen van moderators aan te pakken.

    c. Deze algemene database van termen kan worden gehost in ontwikkelbibliotheken of zelfs beschikbaar worden gesteld via cloudservice-API's voor hergebruik door verschillende AI-systemen, zodat nieuwe AI-systemen baat hebben bij de gecombineerde wijsheid van oudere systemen.

6. Er kan een 'Machine Learning Fuzzing Framework' worden opgezet waarmee technici verschillende soorten aanvallen kunnen injecteren in trainingssets voor testdoeleinden om deze te laten evalueren door AI.
    a. Dit kan niet alleen gericht zijn op teksttaal, maar op afbeeldings-, spraak- en gebarengegevens en permutaties van deze gegevenstypen.

Conclusie

De Asilomar AI Principles illustreren de complexiteit van het leveren van AI op een manier die uitsluitend positief is voor de mensheid. Toekomstige API's moeten communiceren met andere API's om rijke, aantrekkelijke gebruikerservaringen te bieden. Dat betekent dat het simpelweg niet goed genoeg is voor Microsoft om AI goed te doen vanuit een beveiligingsperspectief – de *wereld* ook. We hebben behoefte aan uitlijning van de industrie en samenwerking met een grotere zichtbaarheid van de problemen in dit document op een manier die vergelijkbaar is met onze wereldwijde push voor een Digital Geneva Convention [8]. Door oplossingen te zoeken voor de hier beschreven problemen, kunnen we een begin maken om samen met onze klanten en branchepartners een traject te bewandelen waar AI echt is gedemocratiseerd en een bijdrage levert aan de intelligentie van de mensheid als geheel.

Bibliografie

[1] *Taleb, Nassim Nicholas (2007),* The Black Swan: The Impact of the Highly Improbable, Random House, ISBN 978-1400063512

[2] *Florian Tramèr, Fan Zhang, Ari Juels, Michael K. Reiter, Thomas Ristenpart,*Stealing Machine Learning Models via Prediction APIs

[3] *Satya Nadella:*Het partnerschap van de toekomst

[4] *Claburn, Thomas:*Google's troll-destroying AI kan niet omgaan met typfouten

[5] *Marco Barreno, Blaine Nelson, Anthony D. Joseph, J.D. Tygar:*De beveiliging van machine learning

[6] *Wolchover, Natalie:*Deze pionier op het gebied van kunstmatige intelligentie heeft een paar zorgen

[7] *Conn, Ariel:*Hoe stemmen we kunstmatige intelligentie af op menselijke waarden?

[8] *Smith, Brad:*De noodzaak van dringende collectieve actie om mensen online te houden: Lessen van de cyberaanval van vorige week

[9] *Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, Wenchao Zhou:*Hidden Voice Commands

[10] *Fernanda Viégas, Martin Wattenberg, Daniel Smilkov, James Wexler, Jimbo Wilson, Nikhil Thorat, Charles Nicholson, Google Research:*Big Picture

Feedback

Is deze pagina nuttig? Yes No

Beveiligingsfoutrapporten identificeren op basis van rapporttitels en ruisgegevens

Artikel • 18-03-2025

Tabel uitvouwen

Mayana Pereira	**Scott Christiansen**
CELA Data Science	Beveiliging en vertrouwen van klanten
Microsoft	Microsoft

Abstract: het identificeren van beveiligingsfoutrapporten (SBR's) is een essentiële stap in de levenscyclus van softwareontwikkeling. Bij benaderingen op basis van machine learning onder supervisie is het gebruikelijk om ervan uit te gaan dat volledige foutrapporten beschikbaar zijn voor training en dat hun labels ruisvrij zijn. Naar onze beste kennis is dit de eerste studie om te laten zien dat nauwkeurige labelvoorspelling mogelijk is voor SDR's, zelfs wanneer alleen de titel beschikbaar is en in aanwezigheid van labelruis.

Indextermen: Machine Learning, onjuiste etikettering, ruis, beveiligingsfoutrapport, foutenopslagplaatsen

Ik. INTRODUCTIE

Het identificeren van beveiligingsproblemen tussen gerapporteerde bugs is een dringende behoefte aan softwareontwikkelingsteams, zoals problemen, vragen om snellere oplossingen om te voldoen aan de nalevingsvereisten en de integriteit van de software- en klantgegevens te waarborgen.

Machine learning en hulpprogramma's voor kunstmatige intelligentie beloven de softwareontwikkeling sneller, flexibel en correct te maken. Verschillende onderzoekers hebben machine learning toegepast op het probleem van het identificeren van beveiligingsfouten [2], [7], [8], [18]. Uit eerdere gepubliceerde studies is ervan uitgegaan dat het hele bugrapport beschikbaar is voor het trainen en scoren van een machine learning-model. Dit is niet noodzakelijkerwijs het geval. Er zijn situaties waarin het hele foutenrapport niet beschikbaar kan worden gesteld. Het foutenrapport kan bijvoorbeeld wachtwoorden bevatten, persoonlijke identificatiegegevens (PII) of andere soorten gevoelige gegevens. Dit is een geval dat we momenteel bij Microsoft tegenkomen. Het

is daarom belangrijk om vast te stellen hoe goed de identificatie van beveiligingsfouten kan worden uitgevoerd met minder informatie, bijvoorbeeld wanneer alleen de titel van het foutenrapport beschikbaar is.

Daarnaast bevatten foutopslagplaatsen vaak verkeerd gelabelde vermeldingen [7]: niet-beveiligingsfoutrapporten die zijn geclassificeerd als beveiligingsgerelateerd en omgekeerd. Er zijn verschillende redenen voor het optreden van verkeerd labelen, variërend van het gebrek aan expertise van het ontwikkelingsteam op het gebied van beveiliging, tot de scherpte van bepaalde problemen, bijvoorbeeld dat er niet-beveiligingsfouten op een indirecte manier worden misbruikt om een beveiligingsimplicatie te veroorzaken. Dit is een ernstig probleem omdat het verkeerd labelen van SBR's resulteert in beveiligingsexperts die handmatig een foutdatabase moeten controleren in een dure en tijdrovende inspanning. Begrijpen hoe ruis van invloed is op verschillende classificaties en hoe robuuste (of kwetsbare) verschillende machine learning-technieken aanwezig zijn in aanwezigheid van gegevenssets die besmet zijn met verschillende soorten ruis, is een probleem dat moet worden aangepakt om automatische classificatie toe te passen aan de praktijk van software-engineering.

Voorlopig werk betoogt dat foutopslagplaatsen intrinsiek luidruchtig zijn en dat de ruis een nadelig effect kan hebben op de prestatie-machine learning-classificaties [7]. Er ontbreekt echter een systematische en kwantitatieve studie van de manier waarop verschillende niveaus en soorten ruis van invloed zijn op de prestaties van verschillende machine learning-algoritmen onder supervisie voor het probleem van het identificeren van beveiligingsfoutrapporten (SRB's).

In deze studie laten we zien dat de classificatie van foutrapporten kan worden uitgevoerd, zelfs wanneer alleen de titel beschikbaar is voor training en scoren. Voor zover wij weten, is dit het allereerste werk dat dit doet. Daarnaast bieden we de eerste systematische studie van het effect van ruis in de classificatie van foutenrapporten. We maken een vergelijkende studie van de robuustheid van drie machine learning-technieken (logistieke regressie, naïve Bayes en AdaBoost) tegen klasse-onafhankelijke ruis.

Hoewel er enkele analytische modellen zijn die de algemene invloed van ruis voor enkele eenvoudige classificaties vastleggen [5], [6], bieden deze resultaten geen strikte grenzen aan het effect van de ruis op precisie en zijn ze alleen geldig voor een bepaalde machine learning-techniek. Een nauwkeurige analyse van het effect van ruis in machine learning-modellen wordt meestal uitgevoerd door rekenkundige experimenten uit te voeren. Dergelijke analyses zijn uitgevoerd voor verschillende scenario's, variërend van softwaremetingsgegevens [4], tot satellietbeeldclassificatie [13] en medische gegevens [12]. Deze resultaten kunnen echter niet worden vertaald naar ons specifieke probleem, vanwege de hoge afhankelijkheid van de aard van de gegevenssets en het

onderliggende classificatieprobleem. Voor zover wij weten, zijn er geen gepubliceerde resultaten over het effect van gegevenssets met ruis op de classificatie van rapporten over beveiligingsfouten in het bijzonder.

ONZE ONDERZOEKSBIJDRAGEN:

- We trainen classificaties voor de identificatie van beveiligingsfoutrapporten (SBR's) uitsluitend op basis van de titel van de rapporten. Voor zover wij weten is dit de eerste keer dat dit wordt gedaan. Eerdere werken gebruikten het volledige foutrapport of verbeterden het foutrapport met aanvullende functies. Het classificeren van bugs op basis van de tegel is met name relevant wanneer de volledige foutrapporten niet beschikbaar kunnen worden gesteld vanwege privacyproblemen. Het is bijvoorbeeld berucht dat foutenrapporten wachtwoorden en andere gevoelige gegevens bevatten.
- We bieden ook de eerste systematische studie van de labelruistolerantie van verschillende machine learning-modellen en technieken die worden gebruikt voor de automatische classificatie van SBR's. We maken een vergelijkende studie van robuustheid van drie afzonderlijke machine learning-technieken (logistieke regressie, naïve Bayes en AdaBoost) tegen klasseafhankelijke en klasse-onafhankelijke ruis.

De rest van het document wordt als volgt gepresenteerd: in sectie II presenteren we enkele van de vorige werken in de literatuur. In sectie III beschrijven we de gegevensset en hoe gegevens vooraf worden verwerkt. De methodologie wordt beschreven in sectie IV en de resultaten van onze experimenten die in sectie V worden geanalyseerd. Ten slotte worden onze conclusies en toekomstige werken gepresenteerd in VI.

II. VORIGE WERKEN

Toepassingen van machine learning voor bugrepositories.

Er bestaat uitgebreide literatuur over het toepassen van tekstanalyse, verwerking van natuurlijke taal en machine learning op foutopslagplaatsen in een poging om arbeidsintensische taken zoals detectie van beveiligingsfouten [2], [7], [8], [18], dubbele identificatie van fouten [3], foutsorteerd [1], [11] te automatiseren om een paar toepassingen te noemen. Idealiter vermindert het huwelijk van machine learning (ML) en verwerking van natuurlijke taal mogelijk het handmatige werk dat nodig is voor het cureren van foutdatabases, verkort de vereiste tijd voor het uitvoeren van deze taken en kan de betrouwbaarheid van de resultaten verhogen.

In [7] stellen de auteurs een model voor natuurlijke taal voor om de classificatie van SBR's te automatiseren op basis van de beschrijving van de fout. De auteurs halen een woordenlijst op uit alle foutbeschrijvingen in de trainingsgegevensset en cureren deze handmatig in drie lijsten met woorden: relevante woorden, stopwoorden (veelvoorkomende woorden die niet relevant lijken voor classificatie) en synoniemen. Ze vergelijken de prestaties van beveiligingsfoutclassificatie die is getraind op gegevens die allemaal worden geëvalueerd door beveiligingstechnici en een classificatie die is getraind op gegevens die zijn gelabeld door foutrapporteurs in het algemeen. Hoewel hun model duidelijk effectiever is bij het trainen van gegevens die door beveiligingstechnici worden beoordeeld, is het voorgestelde model gebaseerd op een handmatig afgeleide vocabulaire, waardoor het afhankelijk is van menselijke curatie. Bovendien is er geen analyse van hoe verschillende niveaus van ruis van invloed zijn op hun model, hoe verschillende classificaties reageren op ruis en of ruis in beide klassen de prestaties verschillend beïnvloedt.

Zou et. al [18] maken gebruik van meerdere soorten informatie in een foutrapport waarbij de niet-tekstuele velden van een bugrapport worden betrokken (metafuncties, bijvoorbeeld tijd, ernst en prioriteit) en de tekstuele inhoud van een foutrapport (tekstuele functies, de tekst in samenvattingsvelden). Op basis van deze functies bouwen ze een model om de SBR's automatisch te identificeren via verwerking van natuurlijke taal en machine learning-technieken. In [8] voeren de auteurs een vergelijkbare analyse uit, maar daarnaast vergelijken ze de prestaties van machine learning-technieken onder supervisie en zonder supervisie en bestuderen ze hoeveel gegevens er nodig zijn om hun modellen te trainen.

In [2] verkennen de auteurs ook verschillende machine learning-technieken om bugs te classificeren als SBR's of NSBRs (Non-Security Bug Report) op basis van hun beschrijvingen. Ze stellen een pijplijn voor gegevensverwerking en modeltraining voor op basis van TFIDF. Ze vergelijken de voorgestelde pijplijn met een model op basis van bag-of-words en naïef Bayes. Wijayasekara et al. [16] gebruikte ook tekstanalysetechnieken om de functievector van elk bugrapport te genereren op basis van frequente woorden om Verborgen impact bugs (HIBs) te identificeren. Yang et al. [17] beweerde dat ze foutenrapporten met hoge impact (bijvoorbeeld SDR's) met behulp van Term Frequency (TF) en naïve Bayes identificeerden. In [9] stellen de auteurs een model voor om de ernst van een bug te voorspellen.

LABELGELUID

Het probleem van het omgaan met gegevenssets met labelruis is uitgebreid bestudeerd. Frenay en Verleysen stellen in [6] een labelruistaxonomie voor om verschillende soorten ruislabels te onderscheiden. De auteurs stellen drie verschillende soorten ruis voor:

labelruis die onafhankelijk van de werkelijke klasse plaatsvindt en van de waarden van de instantiefuncties; labelruis die alleen afhankelijk is van het ware label; en labelruis waarbij de kans op verkeerd labelen ook afhankelijk is van de functiewaarden. In ons werk bestuderen we de eerste twee soorten ruis. Vanuit theoretisch oogpunt vermindert labelruis meestal de prestaties van een model [10], behalve in sommige specifieke gevallen [14]. Over het algemeen zijn robuuste methoden afhankelijk van het vermijden van overfitting om labelgeluid te verwerken [15]. De studie van geluidseffecten in classificatie is eerder gedaan op veel gebieden zoals satellietafbeeldingsclassificatie [13], classificatie van softwarekwaliteit [4] en classificatie van medische domeinen [12]. Naar onze beste kennis zijn er geen gepubliceerde werken die de precieze kwantificering bestuderen van de effecten van ruislabels in het probleem van classificatie van SBR's. In dit scenario is de precieze relatie tussen ruisniveaus, ruistypen en prestatievermindering niet vastgesteld. Bovendien is het de moeite waard om te begrijpen hoe verschillende classificaties zich gedragen in aanwezigheid van ruis. Over het algemeen zijn we niet op de hoogte van werk dat systematisch het effect van ruisgegevenssets onderzoekt op de prestaties van verschillende machine learning-algoritmen in de context van softwarefoutrapporten.

III. BESCHRIJVING VAN DATASET

Onze gegevensset bestaat uit 1.073.149 fouttitels, waarvan 552.073 overeenkomen met SBR's en 521.076 aan NSBR's. De gegevens zijn verzameld van verschillende teams in Microsoft in de jaren 2015, 2016, 2017 en 2018. Alle labels zijn verkregen door bugs te verifiëren met systemen op basis van handtekeningen of zijn door mensen gelabeld. Fouttitels in onze gegevensset zijn zeer korte teksten, met ongeveer 10 woorden, met een overzicht van het probleem.

Een. Gegevens vooraf verwerken We parseren elke bugtitel door de lege spaties, wat resulteert in een lijst met tokens. We verwerken elke lijst met tokens als volgt:

- Alle tokens verwijderen die bestandspaden zijn
- Gesplitste tokens waarin de volgende symbolen aanwezig zijn: { , (,), -, }, {, [,], }
- Verwijder stopwoorden, tokens die bestaan uit alleen numerieke tekens en tokens die minder dan 5 keer voorkomen in het hele corpus.

IV. METHODOLOGIE

Het proces van het trainen van onze machine learning-modellen bestaat uit twee hoofdstappen: het coderen van de gegevens in functievectoren en het trainen van

machine learning-classificaties onder supervisie.

A. Functievectoren en Machine Learning-technieken

Het eerste deel omvat het coderen van gegevens in functievectoren met behulp van het term frequencyinverse documentfrequentie-algoritme (TF-IDF), zoals gebruikt in [2]. TF-IDF is een techniek voor het ophalen van gegevens die een termenfrequentie (TF) en de inverse documentfrequentie (IDF) wegen. Elk woord of elke term heeft zijn respectieve TF- en IDF-score. Het TF-IDF algoritme wijst het belang van dat woord toe op basis van het aantal keren dat het in het document wordt weergegeven, en belangrijker nog, het controleert hoe relevant het trefwoord is in de verzameling titels in de gegevensset. We hebben drie classificatietechnieken getraind en vergeleken: naïve Bayes (NB), versterkte beslissingsstructuren (AdaBoost) en logistieke regressie (LR). We hebben deze technieken gekozen omdat ze goed zijn gebleken voor de gerelateerde taak om beveiligingsfoutrapporten te identificeren op basis van het hele rapport in de literatuur. Deze resultaten werden bevestigd in een voorlopige analyse waarbij deze drie classificaties beter presteerden dan ondersteuningsvectormachines en willekeurige forests. In onze experimenten gebruiken we de scikit-learn-bibliotheek voor codering en modeltraining.

B. Soorten ruis

Het geluid dat in dit werk wordt bestudeerd, verwijst naar ruis in het klasselabel in de trainingsgegevens. In de aanwezigheid van zulk geluid worden als gevolg daarvan het leerproces en het resulterende model aangetast door verkeerd gelabelde voorbeelden. We analyseren de impact van verschillende ruisniveaus die zijn toegepast op de klasse-informatie. Typen labelruis zijn eerder in de literatuur besproken met behulp van verschillende terminologie. In ons werk analyseren we de effecten van twee verschillende labelruis in onze classificaties: klasse-onafhankelijke labelruis, die wordt geïntroduceerd door willekeurig exemplaren te kiezen en hun label te spiegelen; en klasseafhankelijke ruis, waarbij klassen een andere kans hebben om lawaaierig te zijn.

a) *klasse-onafhankelijke ruis*: klasse-onafhankelijke ruis verwijst naar de ruis die onafhankelijk van de werkelijke klasse van de exemplaren plaatsvindt. In dit type ruis is de kans op verkeerd labelen p_{br} hetzelfde voor alle exemplaren in de gegevensset. We introduceren klasse-onafhankelijke ruis in onze databestanden door elk label in ons databestand willekeurig om te draaien met waarschijnlijkheid p_{br}.

b) *klasseafhankelijke ruis*: klasseafhankelijke ruis verwijst naar de ruis die afhankelijk is van de werkelijke klasse van de exemplaren. In dit type ruis is de kans op verkeerd labelen in klasse SBR, p_{sbr}, en de kans op verkeerd labelen in klasse NSBR p_{nsbr}. We

introduceren klasseafhankelijke ruis in onze gegevensset door elke invoer in de gegevensset te veranderen waarvoor het werkelijke label SBR is met waarschijnlijkheid p_{sbr}. Op een vergelijkbare manier spiegelen wij het klasselabel van NSBR-exemplaren met waarschijnlijkheid p_{nsbr}.

c) *Single-class ruis*: Single-class ruis is een specifiek geval van klasseafhankelijke ruis, waarbij $p_{nsbr} = 0$ en $p_{sbr} > 0$. Let op: voor klasse-onafhankelijke ruis hebben we $p_{sbr} = p_{nsbr} = p_{br}$.

C. Ruis genereren

Onze experimenten onderzoeken de impact van verschillende ruistypen en niveaus bij de training van SBR-classificatoren. In onze experimenten stellen we 25% van de gegevensset in als testgegevens, 10% als validatie en 65% als trainingsgegevens.

We voegen ruis toe aan de trainings- en validatiegegevenssets voor verschillende niveaus p_{br}, p_{sbr} en p_{nsbr} . We brengen geen wijzigingen aan in de testgegevensset. De verschillende gebruikte ruisniveaus zijn $P = \{0{,}05 \times i | 0 < i < 10\}$.

In klasse-onafhankelijke ruisexperimenten, voor $p_{br} \in P$ doen we het volgende:

- Ruis toevoegen aan trainings- en validatiedatasets.
- Train de logistieke regressie, naïef Bayes- en AdaBoost-modellen met behulp van de trainingsgegevensset (met ruis). Stem de modellen af met de validatiegegevensset (met ruis).
- Test modellen met behulp van een testgegevensset (ruisloos).

In klasseafhankelijke ruisexperimenten, voor $p_{sbr} \in P$ en $p_{nsbr} \in P$ doen we het volgende voor alle combinaties van p_{sbr} en p_{nsbr}:

- Ruis genereren voor trainings- en validatiedataverzamelingen.
- Trainen van logistische regressie-, naïeve Bayes- en AdaBoost-modellen met behulp van trainingsdataset (met ruis);
- Modellen afstemmen met behulp van validatiegegevensset (met ruis);
- Test modellen met behulp van een testgegevensset (ruisloos).

V. EXPERIMENTELE RESULTATEN

In deze sectie analyseert u de resultaten van experimenten die worden uitgevoerd volgens de methodologie die in sectie IV is beschreven.

a) Modelprestaties zonder ruis in de trainingsgegevensset: een van de bijdragen van dit document is het voorstel van een machine learning-model om beveiligingsfouten te identificeren door alleen de titel van de bug te gebruiken als gegevens voor besluitvorming. Dit maakt het mogelijk om machine learning-modellen te trainen, zelfs als ontwikkelteams geen foutenrapporten volledig willen delen vanwege aanwezigheid van gevoelige gegevens. We vergelijken de prestaties van drie machine learning-modellen wanneer ze worden getraind met alleen fouttitels.

Het logistieke regressiemodel is de best presterende classificatie. Het is de classifier met de hoogste AUC-waarde van 0,9826, en een recall van 0,9353 voor een FPR-waarde van 0,0735. De naïve Bayes-classificatie presenteert iets lagere prestaties dan de logistieke regressieclassificatie, met een AUC van 0,9779 en een terugroeping van 0,9189 voor een FPR van 0,0769. De AdaBoost-classificator heeft een inferieure prestatie in vergelijking met de twee eerder genoemde classificatoren. Het bereikt een AUC van 0,9143 en een recallwaarde van 0,7018 bij een FPR van 0,0774. Het gebied onder de ROC-curve (AUC) is een goede meetwaarde voor het vergelijken van de prestaties van verschillende modellen, zoals deze samenvat in één waarde de TPR versus FPR-relatie. In de volgende analyse beperken we onze vergelijkende analyse tot AUC-waarden.

TABLE I
PERFORMANCE OF DIFFERENT ML TECHNIQUES, LOGISTIC REGRESSION(LR), NAIVE BAYES(NB) AND ADABOOST(AB), IN SECURITY BUG CLASSIFICATION.

ML Model	Metric			
	Acc	*TPR*	*FPR*	*AUC*
LR	0.9318	0.9353	0.0735	0.9831
NB	0.8977	0.9189	0.0769	0.9770
AB	0.8257	0.7018	0.0774	0.9143

A. Ruis: één klasse

U kunt zich een scenario voorstellen waarin alle fouten standaard worden toegewezen aan klasse NSBR en een fout alleen wordt toegewezen aan klasse SBR als er een beveiligingsexpert is die de foutopslagplaats controleert. Dit scenario wordt weergegeven in de experimentele instelling van een enkele klasse, waarbij we ervan uitgaan dat $p_{nsbr} = 0$ en $0 < p_{sbr} < 0,5$.

TABLE II
DROP IN AUC FOR VARIOUS p_{sbr}

p_{sbr} value	Machine Learning Model		
	logistic regression	*naive Bayes*	*AdaBoost*
0.0	0.983	0.974	0.923
0.05	0.982	0.972	0.921
0.10	0.982	0.970	0.920
0.15	0.981	0.969	0.919
0.20	0.981	0.968	0.917
0.25	0.980	0.968	0.917
0.30	0.980	0.967	0.916
0.35	0.979	0.966	0.916
0.40	0.978	0.965	0.914
0.45	0.977	0.964	0.914
0.50	0.976	0.963	0.913

In tabel II zien we een zeer kleine impact in de AUC voor alle drie de classificaties. De AUC-ROC van een model dat is getraind op p_{sbr} = 0 in vergelijking met een AUC-ROC van het model waarbij p_{sbr} = 0,25 verschilt met 0,003 voor logistieke regressie, 0,006 voor naïve Bayes en 0,006 voor AdaBoost. In het geval van p_{sbr} = 0,50 verschilt de AUC voor elk van de modellen van het model dat is getraind met p_{sbr} = 0 bij 0,007 voor logistieke regressie, 0,011 voor naïve Bayes en 0,010 voor AdaBoost. Logistieke regressieclassificatie die is getraind in aanwezigheid van ruis van één klasse geeft de kleinste variatie in de AUC-meetwaarde, d.w.z. een robuuster gedrag, in vergelijking met onze naïve Bayes- en AdaBoost-classificaties.

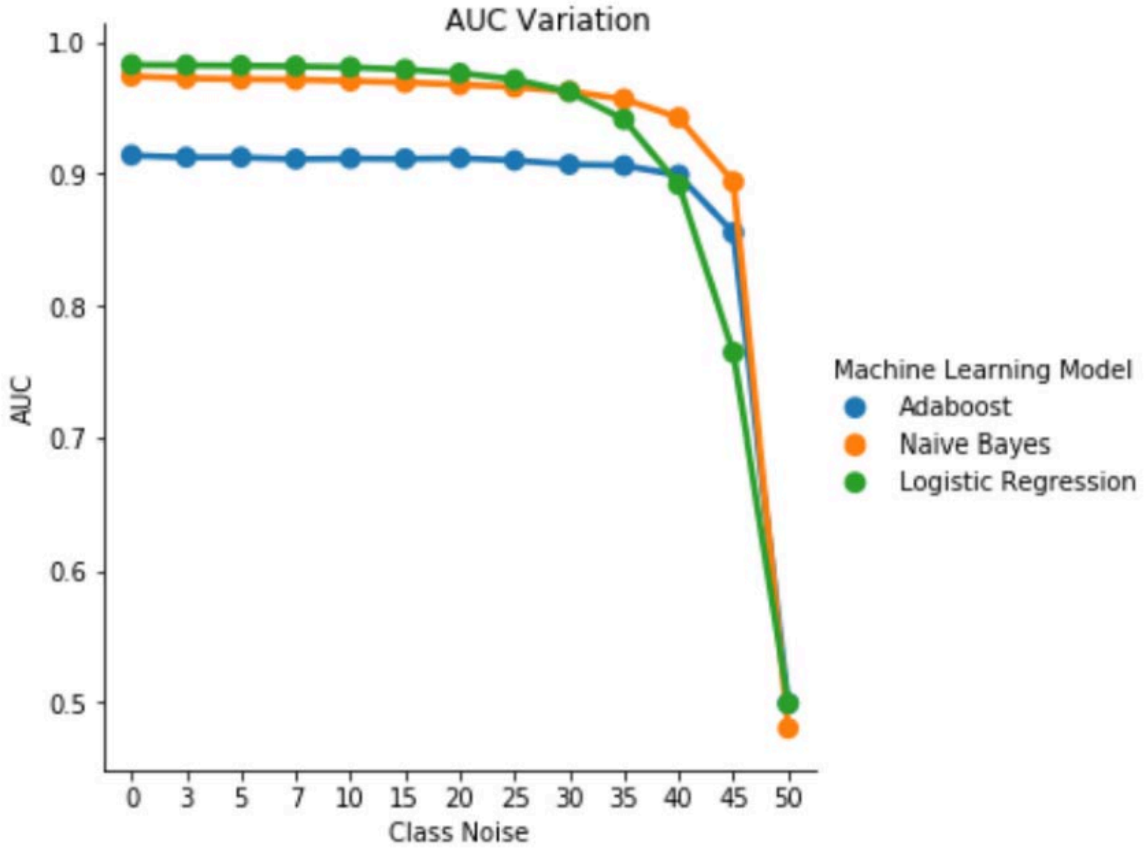
B. Klasseruis: klasse-onafhankelijk

We vergelijken de prestaties van onze drie classificatoren voor het geval dat de trainingsset beschadigd is door klasse-onafhankelijke ruis. We meten de AUC voor elk model dat is getraind met verschillende niveaus p_{br} in de trainingsgegevens.

TABLE III
DROP IN AUC FOR VARIOUS p_{br}

p_{sbr} value	Machine Learning Model		
	logistic regression	*naive Bayes*	*AdaBoost*
0.0	0.9827	0.9739	0.9140
0.05	0.9820	0.9716	0.9125
0.10	0.9808	0.9703	0.9116
0.15	0.9792	0.9692	0.9113
0.20	0.9763	0.9676	0.9120
0.25	0.9714	0.9658	0.9102
0.30	0.9621	0.9626	0.9071
0.35	0.9412	0.9566	0.9062
0.40	0.8917	0.9425	0.8989
0.45	0.7645	0.8939	0.8553
0.50	0.4994	0.4806	0.4996

In tabel III zien we een afname in de AUC-ROC voor elke toename van ruis in het experiment. De AUC-ROC gemeten van een model dat is getraind op ruisloze gegevens vergeleken met een AUC-ROC van het model dat is getraind met klasse-onafhankelijke ruis met p_{br} = 0,25 verschilt met 0,011 voor logistieke regressie, 0,008 voor naïve Bayes en 0,0038 voor AdaBoost. We zien dat labelruis geen invloed heeft op de AUC van naïve Bayes- en AdaBoost-classificaties wanneer de ruisniveaus lager zijn dan 40%. Aan de andere kant ondervindt de logistieke regressieclassificator een impact op de AUC-meting bij labelruisniveaus boven de 30%.



Afb. 1. Variatie van AUC-ROC in klasse-onafhankelijke ruis. Voor een ruisniveau p_{br} =0,5 fungeert de classificatie als een willekeurige classificatie, d.w.w.v. AUC≈0,5. We kunnen echter zien dat voor lagere ruisniveaus (p_{br} ≤0,30) de logistieke regressie-leerling beter presteert in vergelijking met de andere twee modellen. Voor 0,35≤ p_{br} ≤0,45 presenteert het naïve Bayes-model echter betere AUCROC-metrieken.

C. Geluidsstoring per klasse: klasse-afhankelijk

In de laatste reeks experimenten beschouwen we een scenario waarin verschillende klassen verschillende ruisniveaus bevatten, d.w.z. $p_{sbr} \neq p_{nsbr-}$. We verhogen p_{sbr} en p_{nsbr} onafhankelijk met 0,05 in de trainingsgegevens en observeren de verandering in het gedrag van de drie classificatoren.

TABLE V
NAIVE BAYES: DROP IN AUC FOR CLASS-DEPENDENT NOISE

p_{sbr} value	p_{nsbr} values										
	0.0	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**
0.0	0.9743	0.9721	0.9708	0.9699	0.9689	0.9680	0.9673	0.9665	0.9654	0.9644	0.9633
0.05	0.9736	0.9712	0.9703	0.9690	0.9684	0.9676	0.9667	0.9660	0.9647	0.9643	0.9624
0.10	0.9731	0.9710	0.9697	0.9688	0.9677	0.9664	0.9662	0.9654	0.9639	0.9620	0.9611
0.15	0.9729	0.9706	0.9693	0.9683	0.9675	0.9665	0.9652	0.9645	0.9630	0.9607	0.9584
0.20	0.9723	0.9701	0.9692	0.9682	0.9670	0.9660	0.9647	0.9639	0.9612	0.9590	0.9563
0.25	0.9720	0.9697	0.9683	0.9673	0.9661	0.9654	0.9626	0.9622	0.9602	0.9565	0.9494
0.30	0.9717	0.9695	0.9680	0.9672	0.9652	0.9645	0.9621	0.9596	0.9556	0.9499	0.9428
0.35	0.9711	0.9687	0.9675	0.9665	0.9643	0.9624	0.9608	0.9573	0.9490	0.9429	0.9267
0.40	0.9706	0.9685	0.9666	0.9650	0.9623	0.9614	0.9567	0.9530	0.9428	0.9228	0.8941
0.45	0.9701	0.9677	0.9660	0.9643	0.9621	0.9583	0.9522	0.9437	0.9264	0.8954	0.8087
0.50	0.9699	0.9673	0.9657	0.9624	0.9588	0.9541	0.9447	0.9288	0.8937	0.8085	0.5056

TABLE V
NAIVE BAYES: DROP IN AUC FOR CLASS-DEPENDENT NOISE

p_{sbr} value	p_{nsbr} values										
	0.0	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**
0.0	0.9743	0.9721	0.9708	0.9699	0.9689	0.9680	0.9673	0.9665	0.9654	0.9644	0.9633
0.05	0.9736	0.9712	0.9703	0.9690	0.9684	0.9676	0.9667	0.9660	0.9647	0.9643	0.9624
0.10	0.9731	0.9710	0.9697	0.9688	0.9677	0.9664	0.9662	0.9654	0.9639	0.9620	0.9611
0.15	0.9729	0.9706	0.9693	0.9683	0.9675	0.9665	0.9652	0.9645	0.9630	0.9607	0.9584
0.20	0.9723	0.9701	0.9692	0.9682	0.9670	0.9660	0.9647	0.9639	0.9612	0.9590	0.9563
0.25	0.9720	0.9697	0.9683	0.9673	0.9661	0.9654	0.9626	0.9622	0.9602	0.9565	0.9494
0.30	0.9717	0.9695	0.9680	0.9672	0.9652	0.9645	0.9621	0.9596	0.9556	0.9499	0.9428
0.35	0.9711	0.9687	0.9675	0.9665	0.9643	0.9624	0.9608	0.9573	0.9490	0.9429	0.9267
0.40	0.9706	0.9685	0.9666	0.9650	0.9623	0.9614	0.9567	0.9530	0.9428	0.9228	0.8941
0.45	0.9701	0.9677	0.9660	0.9643	0.9621	0.9583	0.9522	0.9437	0.9264	0.8954	0.8087
0.50	0.9699	0.9673	0.9657	0.9624	0.9588	0.9541	0.9447	0.9288	0.8937	0.8085	0.5056

TABLE VI
ADABOOST: DROP IN AUC-ROC FOR CLASS-DEPENDENT NOISE

p_{sbr} value	p_{nsbr} values										
	0.0	**0.05**	**0.10**	**0.15**	**0.20**	**0.25**	**0.30**	**0.35**	**0.40**	**0.45**	**0.50**
0.0	0.9239	0.9219	0.9208	0.9195	0.9176	0.9174	0.9160	0.9160	0.9148	0.9149	0.9139
0.05	0.9218	0.9217	0.9213	0.9209	0.9210	0.9200	0.9179	0.9185	0.9167	0.9170	0.9147
0.10	0.9209	0.9197	0.9206	0.9202	0.9202	0.9181	0.9199	0.9182	0.9172	0.9170	0.91451
0.15	0.9210	0.9208	0.9202	0.9207	0.9187	0.9181	0.9180	0.9187	0.9164	0.9172	0.9164
0.20	0.9189	0.9187	0.9190	0.9188	0.9195	0.9185	0.9174	0.9181	0.9180	0.9152	0.9162
0.25	0.9191	0.9194	0.9184	0.9188	0.9188	0.9177	0.9178	0.9166	0.9149	0.9153	0.9151
0.30	0.9184	0.9195	0.9195	0.9173	0.9170	0.9165	0.9171	0.9175	0.9151	0.9131	0.9087
0.35	0.9183	0.9186	0.9184	0.9157	0.9167	0.9169	0.9146	0.9158	0.9135	0.9117	0.8978
0.40	0.9186	0.9186	0.9175	, 0.9171	0.9146	0.9171	0.9150	0.9133	0.9066	0.9009	0.8726
0.45	0.9160	0.9174	0.9194	0.9177	0.9164	0.9158	0.9159	0.9094	0.8975	0.8673	0.7995
0.50	0.9157	0.9177	0.9174	0.9162	0.9146	0.9135	0.9108	0.8969	0.8692	0.7966	0.4980

Tabellen IV, V en VI tonen de variatie van de AUC naarmate ruis in verschillende niveaus in elke klasse wordt verhoogd: in Tabel IV voor logistieke regressie, in Tabel V voor naïeve Bayes en in Tabel VI voor AdaBoost. Voor alle classificaties zien we een impact in de AUC-meetwaarde wanneer beide klassen een ruisniveau van meer dan 30%bevatten. Naïve Bayes gedraagt zich het meest robuust. De impact op AUC is erg klein, zelfs wanneer de 50% van het label in de positieve klasse worden gespiegeld, mits de negatieve klasse 30% van ruislabels of minder bevat. In dit geval is de daling in AUC 0,03. AdaBoost heeft het meest robuuste gedrag van alle drie de classificaties gepresenteerd. Een aanzienlijke wijziging in AUC vindt alleen plaats voor ruisniveaus die groter zijn dan 45% in beide klassen. In dat geval beginnen we met het observeren van een AUC-verval groter dan 0,02.

D. Over de aanwezigheid van restruis in de oorspronkelijke gegevensset

Onze gegevensset is gelabeld door geautomatiseerde systemen op basis van handtekeningen en door menselijke experts. Bovendien zijn alle bugsrapporten verder beoordeeld en gesloten door menselijke experts. Hoewel we verwachten dat de hoeveelheid ruis in onze gegevensset minimaal en niet statistisch significant is, maakt de aanwezigheid van restruis onze conclusies niet ongeldig. Inderdaad, ter illustratie gaan we ervan uit dat de oorspronkelijke gegevensset is beschadigd door klasse-onafhankelijke ruis, die gelijk is aan $0 < p < 1/2$ en onafhankelijk en identiek verdeeld (i.i.d) is voor elke invoer.

Als we boven op de oorspronkelijke ruis een klasse-onafhankelijke ruis met waarschijnlijkheid p_{br} i.i.d toevoegen, zal de resulterende ruis per vermelding zijn $p* = p(1 - p_{br}) + (1 - p)p_{br}$. Voor $0 < p, p_{br} < 1/2$ hebben we dat de werkelijke ruis per label $p*$ strikt groter is dan de ruis die we kunstmatig toevoegen aan de gegevensset p_{br} . De prestaties van onze classificaties zouden dus nog beter zijn als ze werden getraind met een volledig ruisloze gegevensset ($p = 0$) in de eerste plaats. Samengevat betekent het bestaan van restruis in de gegevensset dat de tolerantie tegen ruis van onze classificaties beter is dan de resultaten die hier worden gepresenteerd. Als de restruis in onze gegevensset statistisch relevant was, zou de AUC van onze classificaties bovendien 0,5 (een willekeurige schatting) worden voor een niveau van ruis dat strikt minder dan 0,5 is. We observeren dit gedrag niet in onze resultaten.

VI. CONCLUSIES EN TOEKOMSTIGE WERKEN

Onze bijdrage in dit document is tweeledig.

Eerst hebben we de haalbaarheid van classificatie van beveiligingsfoutenrapporten laten zien op basis van de titel van het foutrapport. Dit is met name relevant in scenario's waarin het hele bugrapport niet beschikbaar is vanwege privacybeperkingen. In ons

geval bevatten de foutrapporten bijvoorbeeld persoonlijke gegevens, zoals wachtwoorden en cryptografische sleutels, en waren ze niet beschikbaar voor het trainen van de classificaties. Ons resultaat laat zien dat SBR-identificatie met hoge nauwkeurigheid kan worden uitgevoerd, zelfs wanneer alleen rapporttitels beschikbaar zijn. Ons classificatiemodel dat gebruikmaakt van een combinatie van TF-IDF en logistieke regressie voert uit op een AUC van 0,9831.

Ten tweede hebben we het effect van verkeerd gelabelde training en validatiegegevens geanalyseerd. We hebben drie bekende machine learning-classificatietechnieken (naïve Bayes, logistieke regressie en AdaBoost) vergeleken in termen van hun robuustheid tegen verschillende geluidstypen en ruisniveaus. Alle drie de classificaties zijn robuust voor ruis van één klasse. Ruis in de trainingsgegevens heeft geen significant effect in de resulterende classificatie. De afname in AUC is zeer klein (0,01) voor een niveau van ruis van 50%. Voor ruis die in beide klassen aanwezig is en klasse-onafhankelijk is, tonen naïve Bayes- en AdaBoost-modellen alleen significante variaties in de AUC wanneer ze zijn getraind met een dataverzameling met niveaus van ruis die groter zijn dan 40%.

Ten slotte heeft klasseafhankelijke ruis een aanzienlijke invloed op de AUC alleen als er meer dan 35% ruis in beide klassen is. AdaBoost liet de meest robuustheid zien. De impact op de AUC is zeer klein, zelfs wanneer de positieve klasse 50% van zijn labels vervuild heeft, op voorwaarde dat de negatieve klasse 45% vervuilde labels of minder bevat. In dit geval is de daling in AUC kleiner dan 0,03. Naar onze beste kennis is dit de eerste systematische studie over het effect van lawaaierige gegevenssets voor identificatie van beveiligingsfoutenrapporten.

TOEKOMSTIGE WERKEN

In dit document zijn we begonnen met de systematische studie van de effecten van ruis in de prestaties van machine learning-classificaties voor de identificatie van beveiligingsfouten. Er zijn verschillende interessante vervolgen op dit werk, waaronder: het onderzoeken van het effect van ruisgegevenssets bij het bepalen van het ernstniveau van een beveiligingsfout; inzicht krijgen in het effect van klasseonbalans op de tolerantie van de getrainde modellen tegen ruis; inzicht krijgen in het effect van ruis dat adversarial in de gegevensset wordt geïntroduceerd.

VERWIJZINGEN

John Anvik, Lyndon Hiew en Gail C Murphy. Wie moet deze fout oplossen? *In de Proceedings van de 28e internationale conferentie over Software Engineering*, pagina's 361-370. ACM, 2006.

[2] Diksha Behl, Sahil Handa en Anuja Arora. Een hulpmiddel voor het opsporen en analyseren van beveiligingsfouten met behulp van naïeve bayes en tf-idf. In *Optimalisatie, Betrouwbaarheid en Informatietechnologie (ICROIT)*, Internationale Conferentie van 2014 over, pagina's 294–299. IEEE, 2014.

[3] Nicolas Bettenburg, Rahul Premraj, Thomas Opgegevenmann en Sunghun Kim. Dubbele foutrapporten beschouwd als schadelijk echt? In *Softwareonderhoud, 2008. ICSM 2008. Internationale IEEE-conferentie over*, pagina's 337-345. IEEE, 2008.

[4] Andres Folleco, Taghi M Khoshgoftaar, Jason Van Hulse en Lofton Bullard. Het identificeren van cursisten die robuust zijn voor gegevens van lage kwaliteit. In *Gegevens hergebruik en integratie, 2008. IRI 2008. IEEE International Conference on*, pagina's 190-195. IEEE, 2008.

[5] Benoît Frenay. *Onzekerheid en labelruis in machine learning*. PhD-scriptie, Katholieke Universiteit van Leuven, Leuven-la-Neuve, België, 2013.

[6] Benoˆıt Frenay en Michel Verleysen. Classificatie in aanwezigheid van labelruis: een enquête. *IEEE-transacties op neurale netwerken en leersystemen*, 25(5):845-869, 2014.

[7] Michael Gegick, Pete Rotella en Tao Xie. Beveiligingsfoutrapporten identificeren via tekstanalyse: een industriële casestudy. In *Mining softwarearchieven (MSR), tijdens de 7e IEEE-werkconferentie van 2010 over*, pagina's 11–20. IEEE, 2010.

Katerina Goseva-Popstojanova en Jacob Tyo. Identificatie van beveiligingsgerelateerde foutrapporten via tekstanalyse met behulp van classificatie onder supervisie en zonder supervisie. Tijdens de *2018 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, pagina's 344–355, 2018.

[9] Ahmed Lamkanfi, Serge Demeyer, Emanuel Giger en Bart Goethals. De ernst van een gerapporteerde fout voorspellen. In *Mining Software Repositorys (MSR), 2010 7e IEEE Working Conference on*, pagina's 1-10. IEEE, 2010.

[10] Naresh Manwani en PS Sastry. Ruistolerantie onder risicominimalisatie. *IEEE-transacties over cybernetica*, 43(3):1146–1151, 2013.

[11] G Murphy en D Cubranic. Automatische foutortering met behulp van tekstcategorisatie. In *De Zestiende Internationale Conferentie over Software Engineering & Kennisengineering*. Citeseer, 2004.

[12] Mykola Pechenizkiy, Alexey Tsymbal, Seppo Puuronen en Oleksandr Pechenizkiy. Klasruis en leren onder supervisie in medische domeinen: Het effect van functieextractie. In *null*, pagina's 708-713. IEEE, 2006.

[13] Charlotte Pelletier, Silvia Valero, Jordi Inglada, Nicolas Champion, Claire Marais Sicre en Gerard Dedieu. Effect van ruis in trainingsklasselabels op de classificatieprestaties voor het karteriseren van landbedekking met behulp van tijdreeksen van satellietbeelden. *Remote Sensing*, 9(2):173, 2017.

[14] PS Sastry, GD Nagendra en Naresh Manwani. Een team van continuousaction learning automata voor ruistolerant leren van halve ruimten. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 40(1):19–28, 2010.

[15] Choh-Man Teng. Een vergelijking van technieken voor het verwerken van ruis. In *FLAIRS Conference*, pagina's 269-273, 2001.

[16] Dumidu Wijayasekara, Milos Manic en Miles McQueen. Identificatie en classificatie van beveiligingsproblemen via bugdatabases voor tekstanalyse. In *Industrial Electronics Society, IECON 2014-40e Jaarlijkse Conferentie van de IEEE, pagina's 3612-3618. IEEE*, 2014.

[17] Xinli Yang, David Lo, Qiao Huang, Xin Xia en Jianling Sun. Geautomatiseerde identificatie van foutenrapporten met hoge impact die gebruikmaken van onevenwichtige leerstrategieën. In *Computer Software and Applications Conference (COMPSAC), 2016 IEEE 40e jaarlijkse conferentie*, volume 1, pagina's 227-232. IEEE, 2016.

[18] Deqing Zou, Zhijun Deng, Zhen Li en Hai Jin. Automatisch beveiligingsbugrapporten identificeren via analyse van multidimensionale kenmerken. In *Australasian Conference on Information Security and Privacy*, pagina's 619-633. Springer, 2018.

Feedback

Het TLS 1.0-probleem oplossen, tweede editie

In dit document vindt u de meest recente richtlijnen voor het snel identificeren en verwijderen van TLS-protocolversie 1.0-afhankelijkheden in software die is gebouwd op basis van Microsoft-besturingssystemen, met informatie over productwijzigingen en nieuwe functies die door Microsoft worden geleverd om uw eigen klanten en onlineservices te beschermen. Het is bedoeld om te worden gebruikt als uitgangspunt voor het bouwen van een migratieplan naar een TLS 1.2+ netwerkomgeving. Hoewel de oplossingen die hier worden besproken kunnen helpen het gebruik van TLS 1.0 in niet-Microsoft-besturingssystemen of cryptobibliotheken te verwijderen, zijn ze niet het onderwerp van dit document.

TLS 1.0 is een beveiligingsprotocol dat voor het eerst in 1999 is gedefinieerd voor het tot stand brengen van versleutelingskanalen via computernetwerken. Microsoft heeft dit protocol ondersteund sinds Windows XP/Server 2003. Hoewel TLS 1.0 niet langer het standaardbeveiligingsprotocol is dat door moderne besturingssystemen wordt gebruikt, wordt het nog steeds ondersteund voor achterwaartse compatibiliteit. Veranderende wettelijke vereisten en nieuwe beveiligingsproblemen in TLS 1.0 bieden bedrijven de stimulans om TLS 1.0 volledig uit te schakelen.

Microsoft raadt klanten aan dit probleem voor te gaan door WAAR mogelijk TLS 1.0-afhankelijkheden in hun omgevingen te verwijderen en TLS 1.0 uit te schakelen op besturingssysteemniveau. Gezien de tijdsduur dat TLS 1.0 wordt ondersteund door de software-industrie, wordt het ten zeerste aanbevolen dat een TLS 1.0-afschaffingsplan het volgende omvat:

- Codeanalyse voor het vinden/herstellen van in code vastgelegde exemplaren van TLS 1.0 of oudere beveiligingsprotocollen.
- Scannen van netwerkeindpunten en verkeersanalyse om besturingssystemen te identificeren met behulp van TLS 1.0 of oudere protocollen.
- Volledige regressietests via de volledige toepassingsstack waarbij TLS 1.0 is uitgeschakeld.
- Migratie van verouderde besturingssystemen en ontwikkelingsbibliotheken/frameworks naar versies die standaard kunnen onderhandelen over TLS 1.2.
- Compatibiliteitstests in besturingssystemen die door uw bedrijf worden gebruikt om ondersteuningsproblemen met TLS 1.2 te identificeren.
- Coördinatie met uw eigen zakelijke partners en klanten om hen op de hoogte te stellen van uw overstap om TLS 1.0 te verwijderen.
- Begrijpen welke clients mogelijk geen verbinding meer kunnen maken met uw servers zodra TLS 1.0 is uitgeschakeld.

Het doel van dit document is aanbevelingen te bieden waarmee technische blokkeringen kunnen worden verwijderd om TLS 1.0 uit te schakelen, terwijl tegelijkertijd de zichtbaarheid van de gevolgen van deze wijziging voor uw eigen klanten wordt vergroot. Het voltooien van dergelijke onderzoeken kan helpen de bedrijfsimpact van het volgende beveiligingsprobleem in TLS 1.0 te verminderen. Voor de doeleinden van dit document bevatten verwijzingen naar de afschaffing van TLS 1.0 ook TLS 1.1.

Bedrijfssoftwareontwikkelaars hebben een strategische behoefte aan meer toekomstveilige en flexibele oplossingen (ook wel cryptoflexibiliteit genoemd) om toekomstige inbreuk op beveiligingsprotocollen te kunnen aanpakken. Hoewel dit document flexibele oplossingen voorstelt voor het verwijderen van TLS-hardcoding, vallen bredere cryptoflexibiliteitsoplossingen buiten het bereik van dit document.

De huidige status van de TLS 1.0-implementatie van Microsoft

De TLS 1.0-implementatie van Microsoft is vrij van bekende beveiligingsproblemen. Vanwege het potentieel voor toekomstige protocol downgradeaanvallen en andere TLS 1.0-beveiligingsproblemen die niet specifiek zijn voor de implementatie van Microsoft, wordt aanbevolen dat afhankelijkheden van alle beveiligingsprotocollen die ouder zijn dan TLS 1.2, waar mogelijk worden verwijderd (TLS 1.1/1.0/SSLv3/SSLv2).

Bij het plannen van deze migratie naar TLS 1.2+ moeten ontwikkelaars en systeembeheerders rekening houden met het potentieel voor hardcodering van protocolversies in toepassingen die zijn ontwikkeld door hun werknemers en partners. Hardcoding betekent hier dat de TLS-versie is vastgezet op een versie die verouderd en minder veilig is dan nieuwere versies. TLS-versies die hoger zijn dan de in code vastgelegde versie, kunnen niet worden gebruikt zonder het betreffende programma te wijzigen. Deze klasse van probleem kan niet worden opgelost zonder broncodewijzigingen en implementatie van software-updates. Hardcodering van protocolversies was in het verleden gebruikelijk voor test- en ondersteuningsdoeleinden, aangezien veel verschillende browsers en besturingssystemen verschillende niveaus van TLS-ondersteuning hadden.

Ondersteunde versies van TLS in Windows

Veel besturingssystemen hebben verouderde TLS-versiestandaarden of ondersteuningsdrempels waarvoor rekening moet worden gehouden.

Afbeelding 1: Ondersteuning van beveiligingsprotocollen per OS-versie

Tabel uitvouwen

Windows besturingssysteem	SSLv2	SSLv3	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
Windows Vista	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund	Niet ondersteund	Niet ondersteund
Windows Server 2008	Ingeschakeld	Ingeschakeld	Ingeschakeld	Uitgeschakeld*	Uitgeschakeld*	Niet ondersteund
Windows 7 (WS2008 R2)	Ingeschakeld	Ingeschakeld	Ingeschakeld	Uitgeschakeld*	Uitgeschakeld*	Niet ondersteund
Windows 8	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet

Windows besturingssysteem	SSLv2	SSLv3	TLS 1.0	TLS 1.1	TLS 1.2	TLS 1.3
(WS2012)						ondersteund
Windows 8.1 (WS2012 R2)	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund
Windows 10	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund
Windows 11	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld	Ingeschakeld
Windows Server 2016	Niet ondersteund	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund
Windows Server 2016	Niet ondersteund	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund
Windows Server 2019	Niet ondersteund	Disabled	Ingeschakeld	Ingeschakeld	Ingeschakeld	Niet ondersteund
Windows Server 2019 GS-editie	Niet ondersteund	Disabled	Disabled	Disabled	Ingeschakeld	Niet ondersteund
Windows Server 2022	Niet ondersteund	Disabled	Disabled	Disabled	Ingeschakeld	Ingeschakeld

Windows Server 2019 GS-editie is compatibel met Microsoft SDL, TLS 1.2 alleen met een beperkte set coderingssuites.

Windows Server 2022-editie is compatibel met Microsoft SDL, TLS 1.2 en TLS 1.3 alleen met een beperkte set coderingssuites.

TLS 1.1/1.2 kan worden ingeschakeld op Windows Server 2008 via dit optionele Windows Update-pakket.

Zie VOOR meer informatie over afschaffing van TLS 1.0/1.1 in IE/Edge TLS-verbindingen moderniseren in Microsoft Edge en Internet Explorer 11, wijzigingen die van invloed zijn op sitecompatibiliteit naar Microsoft Edge en tls/1.0 en TLS/1.1 uitschakelen in de nieuwe Edge-browser

Een snelle manier om te bepalen welke TLS-versie wordt aangevraagd door verschillende clients wanneer u verbinding maakt met uw onlineservices, is door te verwijzen naar de Handshake-simulatie bij Qualys SSL Labs. Deze simulatie heeft betrekking op combinaties van clientbesturingssystemen/browsers voor alle fabrikanten. Zie bijlage A aan het einde van dit document voor een gedetailleerd voorbeeld met de tls-protocolversies die zijn onderhandeld door verschillende combinaties van gesimuleerd client-besturingssysteem/browser bij het maken van verbinding met www.microsoft.com.

Als dit nog niet is voltooid, wordt het ten zeerste aanbevolen om een inventarisatie uit te voeren van besturingssystemen die worden gebruikt door uw onderneming, klanten en partners (de laatste twee via bereik/communicatie of ten minste HTTP User-Agent tekenreeksverzameling). Deze inventaris kan verder worden aangevuld door verkeersanalyse aan de rand van uw bedrijfsnetwerk. In een dergelijke situatie levert een verkeersanalyse de TLS-versies op die zijn onderhandeld door klanten/partners die verbinding maken met uw services, maar het verkeer zelf blijft versleuteld.

Technische verbeteringen van Microsoft om TLS 1.0-afhankelijkheden te elimineren

Sinds de v1-versie van dit document heeft Microsoft een aantal software-updates en nieuwe functies verzonden ter ondersteuning van afschaffing van TLS 1.0. Deze omvatten:

- Aangepaste IIS-logboekregistratie om client-IP-/gebruikersagenttekenreeks, service-URI, TLS-protocolversie en coderingssuite te correleren.
  - Met deze logboekregistratie kunnen beheerders eindelijk de blootstelling van hun klanten aan zwakke TLS kwantificeren.
- SecureScore - Om beheerders van Office 365-tenants te helpen bij het identificeren van hun eigen zwakke TLS-gebruik, is de SecureScore-portal gebouwd om deze informatie te delen, aangezien de ondersteuning van TLS 1.0 werd beëindigd in Office 365 in oktober 2018.
  - Deze portal biedt Office 365-tenantbeheerders de waardevolle informatie die ze nodig hebben om contact op te leggen met hun eigen klanten die zich mogelijk niet bewust zijn van hun eigen TLS 1.0-afhankelijkheden.
  - Ga naar https://securescore.microsoft.com/ voor meer informatie.
- .Net Framework-updates om hardcodering op app-niveau te elimineren en framework-overgenomen TLS 1.0-afhankelijkheden te voorkomen.
- Richtlijnen voor ontwikkelaars en software-updates zijn uitgebracht om klanten te helpen .Net-afhankelijkheden op zwakke TLS te identificeren en te elimineren: Best practices voor Transport Layer Security (TLS) met .NET Framework
  - Ter informatie: Alle apps die zijn gericht op .NET 4.5 of lager, moeten waarschijnlijk worden gewijzigd om TLS 1.2 te ondersteunen.
- TLS 1.2 is teruggezet naar Windows Server 2008 SP2 en XP POSReady 2009 om klanten te helpen met verouderde verplichtingen.
- Begin 2019 worden er meer aankondigingen gedaan en in volgende updates van dit document gecommuniceerd.

TLS 1.0-afhankelijkheden zoeken en herstellen in code

Voor producten die gebruikmaken van de door het Windows-besturingssysteem geleverde cryptografiebibliotheken en beveiligingsprotocollen, moeten de volgende stappen helpen bij het identificeren van eventuele in code vastgelegde TLS 1.0-gebruik in uw toepassingen:

1. Identificeer alle exemplaren van AcquireCredentialsHandle (). Dit helpt revisoren dichter bij codeblokken te komen waar TLS mogelijk is vastgelegd.
2. Bekijk alle exemplaren van de SecPkgContext_SupportedProtocols en SecPkgContext_ConnectionInfo structuren voor in code vastgelegde TLS.

3. Stel in systeemeigen code alle niet-nul toewijzingen van grbitEnabledProtocols in op nul. Hierdoor kan het besturingssysteem de standaard TLS-versie gebruiken.

4. Schakel de FIPS-modus uit als deze is ingeschakeld vanwege het potentieel voor conflict met instellingen die zijn vereist voor het expliciet uitschakelen van TLS 1.0/1.1 in dit document. Zie bijlage B voor meer informatie.

5. Werk toepassingen bij en compileer deze opnieuw met WinHTTP die worden gehost op Server 2012 of ouder.

   a. Beheerde apps: herbouwen en opnieuw bepalen op basis van de nieuwste .NET Framework-versie

   b. Toepassingen moeten code toevoegen om TLS 1.2 te ondersteunen via WinHttpSetOption

6. Als u alle bases wilt behandelen, scant u broncode en onlineserviceconfiguratiebestanden voor de onderstaande patronen die overeenkomen met geïnventareerde typewaarden die vaak worden gebruikt in TLS-hardcoding:

   a. SecurityProtocolType

   b. SSLv2, SSLv23, SSLv3, TLS1, TLS 10, TLS11

   c. WINHTTP_FLAG_SECURE_PROTOCOL_

   d. SP_PROT_

   e. NSStreamSocketSecurityLevel

   f. PROTOCOL_SSL of PROTOCOL_TLS

De aanbevolen oplossing in alle bovenstaande gevallen is om de vastgelegde protocolversieselectie te verwijderen en uit te stellen op de standaardinstelling van het besturingssysteem. Als u DevSkim gebruikt, klikt u hier om regels te zien die betrekking hebben op de bovenstaande controles die u met uw eigen code kunt gebruiken.

Windows PowerShell-scripts of gerelateerde registerinstellingen bijwerken

Windows PowerShell maakt gebruik van .NET Framework 4.5, die TLS 1.2 niet als beschikbaar protocol bevat. Er zijn twee oplossingen beschikbaar om dit te omzeilen:

1. Wijzig het betreffende script om het volgende op te nemen:

   Powershell

   ```
   [System.Net.ServicePointManager]::SecurityProtocol =
   [System.Net.SecurityProtocolType]::Tls12;
   ```

2. Voeg een systeembrede registersleutel (bijvoorbeeld via groepsbeleid) toe aan elke computer die TLS 1.2-verbindingen vanuit een .NET-app moet maken. Dit zorgt ervoor dat .NET gebruikmaakt van

de TLS-versies 'Systeemstandaard' die TLS 1.2 als een beschikbaar protocol toevoegt. Hierdoor kunnen de scripts toekomstige TLS-versies gebruiken wanneer het besturingssysteem deze ondersteunt. (bijvoorbeeld TLS 1.3)

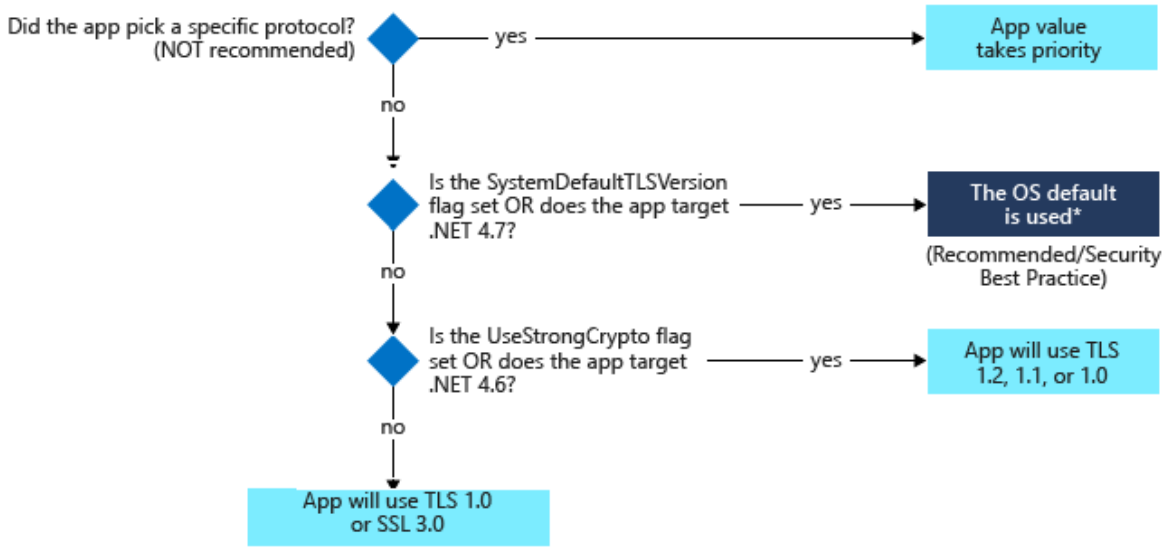
reg add HKLM\SOFTWARE\Microsoft.NETFramework\v4.0.30319 /v SystemDefaultTlsVersions /t REG_DWORD /d 1 /f /reg:64

reg add HKLM\SOFTWARE\Microsoft.NETFramework\v4.0.30319 /v SystemDefaultTlsVersions /t REG_DWORD /d 1 /f /reg:32

Oplossingen (1) en (2) sluiten elkaar wederzijds uit, wat betekent dat ze niet samen hoeven te worden geïmplementeerd.

Beheerde toepassingen opnieuw bouwen/opnieuw samenstellen met behulp van de nieuwste .Net Framework-versie

Toepassingen die gebruikmaken van .NET Framework-versies vóór 4.7, hebben mogelijk beperkingen waardoor ondersteuning voor TLS 1.0 wordt beperkt, ongeacht de onderliggende standaardinstellingen voor het besturingssysteem. Raadpleeg het onderstaande diagram en de Transport Layer Security (TLS)-best practices met het .NET Framework voor meer informatie.



SystemDefaultTLSVersion heeft voorrang boven het app-niveau waarmee specifieke TLS-versies worden gericht. De aanbevolen werkwijze is om altijd de standaard TLS-versie van het besturingssysteem te gebruiken. Het is ook de enige crypto agile oplossing waarmee uw apps kunnen profiteren van toekomstige TLS 1.3-ondersteuning.

Als u zich richt op oudere versies van .NET Framework, zoals 4.5.2 of 3.5, gebruikt uw toepassing standaard de oudere en niet aanbevolen protocollen zoals SSL 3.0 of TLS 1.0. Het wordt ten zeerste aanbevolen om een upgrade uit te voeren naar nieuwere versies van .NET Framework, zoals .NET Framework 4.6 of de juiste registersleutels in te stellen voor UseStrongCrypto.

Testen met TLS 1.2+

Na de oplossingen die in de bovenstaande sectie worden aanbevolen, moeten producten worden getest op regressiefouten en compatibiliteit met andere besturingssystemen in uw onderneming.

- Het meest voorkomende probleem in deze regressietest is een TLS-onderhandelingsfout vanwege een clientverbindingspoging van een besturingssysteem of browser die TLS 1.2 niet ondersteunt.
  - Een Vista-client onderhandelt bijvoorbeeld niet over TLS met een server die is geconfigureerd voor TLS 1.2+ omdat de maximale ondersteunde TLS-versie van Vista 1.0 is. Deze client moet worden bijgewerkt of buiten gebruik gesteld in een TLS 1.2+-omgeving.

- Voor producten die gebruikmaken van wederzijdse TLS-verificatie op basis van certificaten, is mogelijk extra regressietests vereist omdat de certificaatselectiecode die is gekoppeld aan TLS 1.0 minder expressief was dan die voor TLS 1.2.
  - Als een product onderhandelt over MTLS met een certificaat van een niet-standaardlocatie (buiten de standaardcertificaatarchieven in Windows), moet die code mogelijk worden bijgewerkt om ervoor te zorgen dat het certificaat correct wordt verkregen.

- Service-interafhankelijkheden moeten worden gecontroleerd op problemen.

  - Alle diensten die samenwerken met diensten van derde partijen, moeten aanvullende interoperabiliteitstests uitvoeren met die derden.

  - Voor niet-Windows-toepassingen of serverbesturingssystemen die in gebruik zijn, is onderzoek/bevestiging vereist dat ze TLS 1.2 ondersteunen. Scannen is de eenvoudigste manier om dit te bepalen.

Een eenvoudige blauwdruk voor het testen van deze wijzigingen in een onlineservice bestaat uit het volgende:

1. Voer een scan uit van productieomgevingssystemen om besturingssystemen te identificeren die TLS 1.2 niet ondersteunen.

2. Scan broncode- en onlineserviceconfiguratiebestanden voor in code vastgelegde TLS, zoals beschreven in 'TLS 1.0-afhankelijkheden zoeken en herstellen in code'

3. Toepassingen bijwerken/opnieuw compileren zoals vereist:

   a. Beheerde apps

      i. Bouw opnieuw op basis van de nieuwste .NET Framework-versie.

      ii. Controleer of het gebruik van de SSLProtocols-enumeratie is ingesteld op SSLProtocols.None om de standaardinstellingen van het besturingssysteem te gebruiken.

   b. WinHTTP-apps : herbouwen met WinHttpSetOption ter ondersteuning van TLS 1.2

4. Begin met testen in een preproductie- of faseringsomgeving, waarbij alle beveiligingsprotocollen ouder dan TLS 1.2 via het register zijn uitgeschakeld.

5. Corrigeer eventuele resterende gevallen van TLS-hardcoding zoals ze worden aangetroffen tijdens het testen. Implementeer de software opnieuw en voer een nieuwe regressietest uit.

Partners op de hoogte stellen van uw TLS 1.0-afschaffingsplannen

Nadat TLS-hardcoding is aangepakt en updates voor het besturingssysteem en ontwikkelingsframework zijn voltooid, moet u, als u ervoor kiest om TLS 1.0 af te schaffen, dit coördineren met klanten en partners.

- Vroege partner-/klantactiviteiten zijn essentieel voor een succesvolle afschaffing van TLS 1.0. Dit moet minimaal bestaan uit blogberichten, whitepapers of andere webinhoud.
- Partners moeten elk hun eigen TLS 1.2-gereedheid evalueren via het besturingssysteem/codescan-/regressietestinitiatieven die in bovenstaande secties worden beschreven.

Conclusion

Het verwijderen van TLS 1.0-afhankelijkheden is een ingewikkeld probleem om end-to-end uit te voeren. Microsoft- en branchepartners nemen hier vandaag actie op om ervoor te zorgen dat onze volledige productstack standaard veiliger is, van onze onderdelen van het besturingssysteem en de ontwikkelingsframeworks tot de toepassingen/services die erop zijn gebouwd. Door de aanbevelingen in dit document te volgen, kunt u de juiste koers in uw onderneming in kaart brengen en weten welke uitdagingen u kunt verwachten. Het helpt uw eigen klanten ook meer voorbereid te worden op de overgang.

Bijlage A: Handshake Simulatie voor verschillende clients die verbinding maken met www.microsoft.com, met dank aan SSLLabs.com

Handshake Simulation			
Android 2.3.7 No SNI [2]	RSA 2048 (SHA256)	TLS 1.0	TLS_RSA_WITH_AES_128_CBC_SHA No FS
Android 4.0.4	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Android 4.1.1	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Android 4.2.2	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Android 4.3	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Android 4.4.2	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Android 5.0.0	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 6.0	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Android 7.0	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256 ECDH secp256r1 FS
Baidu Jan 2015	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
BingPreview Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Chrome 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Chrome 69 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Chrome 70 / Win 10	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Firefox 31.3.0 ESR / Win 7	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 47 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 ECDH secp256r1 FS
Firefox 49 / XP SP3	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Firefox 62 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Googlebot Feb 2018	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
IE 7 / Vista	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
IE 8 / XP No FS [1] No SNI [2]	Server sent fatal alert: handshake_failure		
IE 8-10 / Win 7 R	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
IE 11 / Win 7 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 11 / Win 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 10 / Win Phone 8.0	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
IE 11 / Win Phone 8.1 R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 ECDH secp256r1 FS
IE 11 / Win Phone 8.1 Update R	RSA 2048 (SHA256)	TLS 1.2 > http/1.1	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
IE 11 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Edge 15 / Win 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Edge 13 / Win Phone 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Java 6u45 No SNI [2]	RSA 2048 (SHA256)	TLS 1.0	TLS_RSA_WITH_AES_128_CBC_SHA No FS
Java 7u25	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA ECDH secp256r1 FS
Java 8u161	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
OpenSSL 0.9.8y	RSA 2048 (SHA256)	TLS 1.0	TLS_RSA_WITH_AES_256_CBC_SHA No FS
OpenSSL 1.0.1l R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
OpenSSL 1.0.2e R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 5.1.9 / OS X 10.6.8	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Safari 6 / iOS 6.0.1	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 6.0.4 / OS X 10.8.4 R	RSA 2048 (SHA256)	TLS 1.0	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA ECDH secp256r1 FS
Safari 7 / iOS 7.1 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 7 / OS X 10.9 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 8 / iOS 8.4 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 8 / OS X 10.10 R	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 ECDH secp256r1 FS
Safari 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 9 / OS X 10.11 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 10 / iOS 10 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Safari 10 / OS X 10.12 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Apple ATS 9 / iOS 9 R	RSA 2048 (SHA256)	TLS 1.2 > h2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
Yahoo Slurp Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS
YandexBot Jan 2015	RSA 2048 (SHA256)	TLS 1.2	TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 ECDH secp256r1 FS

Bijlage B: TLS 1.0/1.1 wordt afgeschaft terwijl de FIPS-modus behouden blijft

Volg de onderstaande stappen als voor uw netwerk de FIPS-modus is vereist, maar u ook TLS 1.0/1.1 wilt verwijderen:

1. Configureer TLS-versies via het register door 'Ingeschakeld' in te stellen op nul voor de ongewenste TLS-versies.
2. Schakel Curve 25519 (alleen Server 2016) uit via Groepsbeleid.
3. Schakel coderingssuites uit met behulp van algoritmen die niet zijn toegestaan door de relevante FIPS-publicatie. Voor Server 2016 (ervan uitgaande dat de standaardinstellingen van kracht zijn) betekent dit het uitschakelen van RC4-, PSK- en NULL-coderingen.

Inzenders/dankzij

Mark Cartwright
Bryan Sullivan
Patrick Jungles
Michael Scovetta
Tony Rice
David LeBlanc
Mortimer Cook
Daniel Sommerfeld
Andrei Popov
Michiko Short
Justin Burke
Gov Maharaj
Brad Turner
Sean Stevenson

Last updated on 26-03-2026

Mogelijkheden voor het afdwingen van TLS-versies zijn nu beschikbaar per certificaatbinding in Windows Server 2019

Artikel • 22-03-2025

Dit bericht is geschreven door

Andrew Marshall, Hoofdbeveiligingsprogramma Manager, Klantenbeveiliging en Vertrouwen

Gabriel Montenegro, Principal Program Manager, Core Networking

Niranjan Inamdar, Senior Software Engineer, Core Networking

Michael Brown, Senior Software Engineer, Internet Information Services

Ivan Pashov, Principal Software Engineering Lead, Core Networking

Augustus 2019

Als technici wereldwijd hun eigen afhankelijkheden van TLS 1.0 elimineren, lopen ze de complexe uitdaging aan om hun eigen beveiligingsbehoeften te verdelen met de migratiegereedheid van hun klanten. Tot op heden hebben we klanten geholpen deze problemen op te lossen door TLS 1.2-ondersteuning toe te voegen aan oudere besturingssystemen, door nieuwe indelingen voor logboekregistratie in IIS te verzenden voor het detecteren van zwak TLS-gebruik door clients en de meest recente technische richtlijnen voor het elimineren van TLS 1.0-afhankelijkheden.

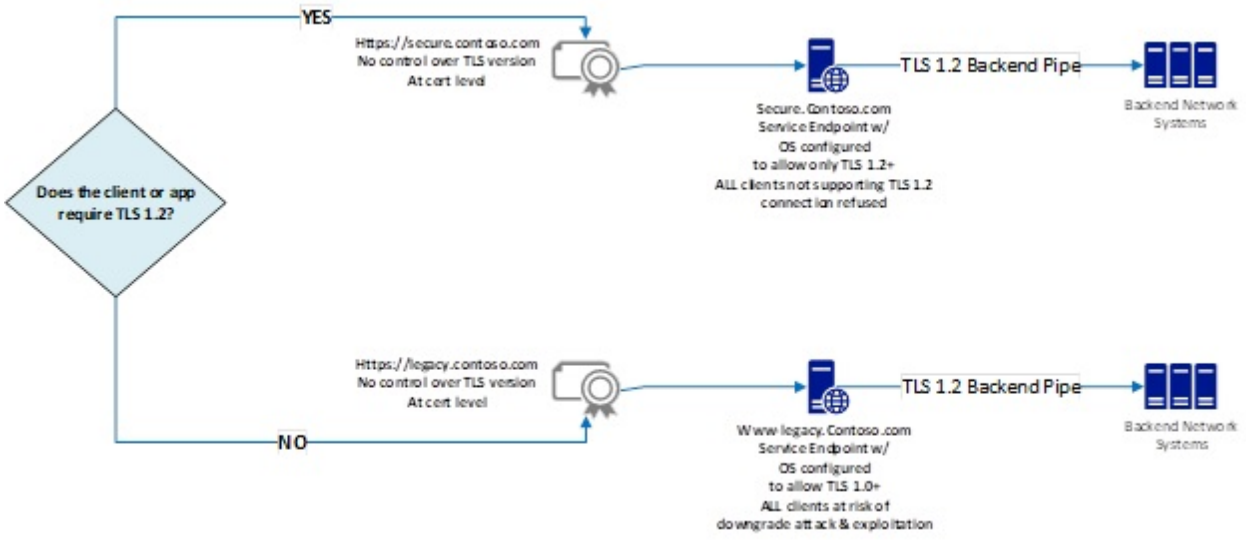
Microsoft kondigt nu een krachtige nieuwe functie in Windows aan om uw overgang naar een TLS 1.2+ wereld eenvoudiger te maken. Vanaf KB4490481 kunt u met Windows Server 2019 nu voorkomen dat zwakke TLS-versies worden gebruikt met afzonderlijke certificaten die u aanwijst. We noemen deze functie 'Verouderde TLS uitschakelen' en dwingt effectief een TLS-versie en coderingssuitevloer af op elk certificaat dat u selecteert.

Als u verouderde TLS uitschakelt, kan een onlineservice ook twee verschillende groeperingen van eindpunten op dezelfde hardware aanbieden: één die alleen TLS 1.2+ verkeer toestaat en een andere service die geschikt is voor verouderd TLS 1.0-verkeer. De wijzigingen worden geïmplementeerd in HTTP.sysen met de uitgifte van extra certificaten kunnen verkeer worden gerouteerd naar het nieuwe eindpunt met de juiste

TLS-versie. Vóór deze wijziging zou het implementeren van dergelijke mogelijkheden extra hardware-investeringen vereisen, omdat dergelijke instellingen alleen via het register kunnen worden geconfigureerd.

Details van functiescenario's

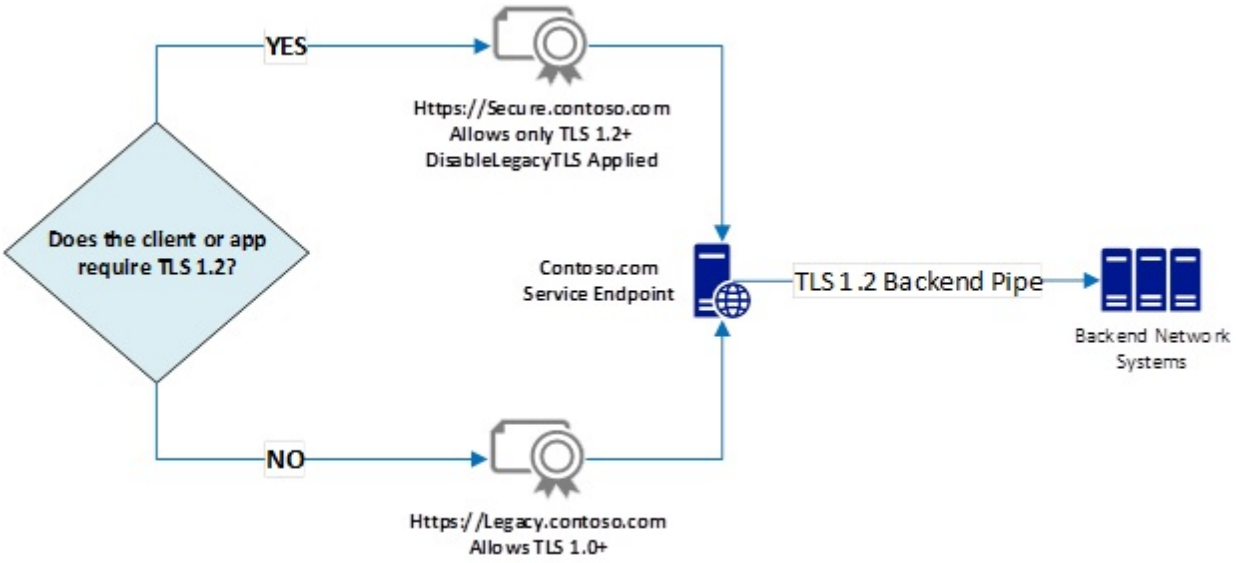
Een algemeen implementatiescenario bevat één set hardware in een datacenter met klanten met gemengde behoeften: sommigen hebben TLS 1.2 nodig als een afgedwongen minimum op dit moment en anderen zijn niet klaar met het verwijderen van TLS 1.0-afhankelijkheden. Afbeelding 1 illustreert het selecteren van TLS-versies en certificaatbinding als afzonderlijke acties. Dit is de standaardfunctionaliteit:



Afbeelding 1: Standaardselectie van TLS-versie en functionaliteit voor certificaatbinding

- secure.contoso.com uw klanten doorsturen naar een service-eindpunt dat alleen TLS 1.2 en hoger ondersteunt.
- Legacy.contoso.com verwijst klanten met verouderde TLS 1.0-behoeften (zoals degenen die nog steeds migreren naar TLS 1.2) naar een eindpunt dat TLS 1.0 gedurende beperkte tijd ondersteunt. Hierdoor kunnen klanten gereedheidstests voltooien voor TLS 1.2 zonder serviceonderbreking en zonder dat andere klanten die gereed zijn voor TLS 1.2 worden geblokkeerd.

Normaal gesproken hebt u twee fysiek afzonderlijke hosts nodig om al het verkeer af te handelen en tls-versie af tedwingen, omdat het afdwingen van TLS-aanvragen met een minimale protocolversie vereist dat zwakkere protocollen worden uitgeschakeld via registerinstellingen voor het hele systeem. We hebben deze functionaliteit hoger beschikbaar gemaakt op de stack, waarbij de TLS-sessie is gebonden aan het certificaat, zodat een specifieke minimale TLS-versie kan worden toegewezen, zoals beschreven in afbeelding 2 hieronder.



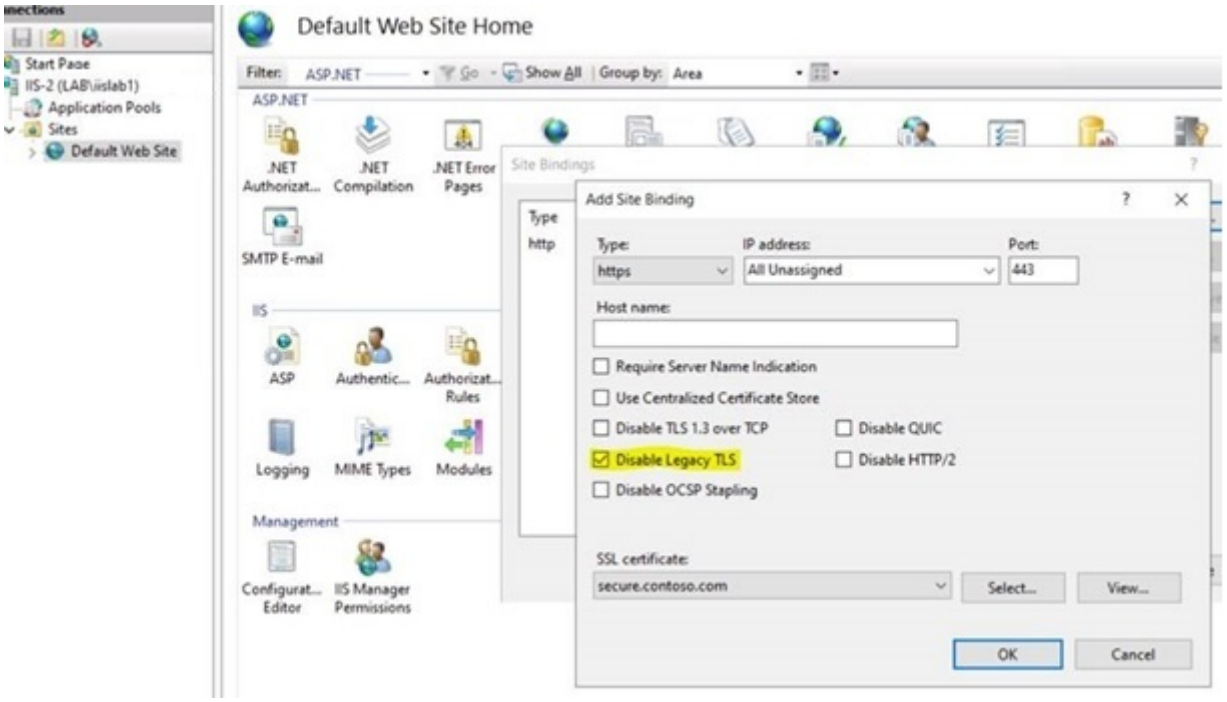
Afbeelding 2: Verouderde TLS-functie uitschakelen die minimale TLS-versie afdwingt voor een geselecteerd certificaat, Secure.contoso.com.

Richtlijnen voor functie-implementatie

De functie Verouderde TLS uitschakelen kan worden geïmplementeerd via de IIS-serverinterface (Internet Information Services), via PowerShell-opdrachten of C++ HTTP.sys API's.

Optie 1: IIS UI-configuratie (beschikbaar in Windows 10 versie 2004 en Windows Server versie 2004 en hoger)

Maak een sitebinding voor het SSL-certificaat 'secure.contoso.com' zoals hieronder wordt weergegeven, schakel 'Verouderde TLS uitschakelen' in en klik op OK.



Optie 2: PowerShell (beschikbaar in Windows 10 versie 2004 en Windows Server versie 2004 en hoger)

In PowerShell kunt u als volgt verwijzen naar SSL-vlaggen:

PowerShell

```
[Microsoft.Web.Administration.SslFlags]::DisableLegacyTLS
```

Het is handig om kortere benoemde variabelen voor deze variabelen te maken:

PowerShell

```
$Sni = [Microsoft.Web.Administration.SslFlags]::Sni

$Sni\_CCS = [Microsoft.Web.Administration.SslFlags]::Sni +
[Microsoft.Web.Administration.SslFlags]::CentralCertStore

$CCS = [Microsoft.Web.Administration.SslFlags]::CentralCertStore

$DisableLegacyTLS =
[Microsoft.Web.Administration.SslFlags]::DisableLegacyTLS

 $storeLocation = "Cert:\\LocalMachine\\My"
```

Een voorbeeld van het maken van een sitebinding naar een nieuwe site en het uitschakelen van verouderde TLS:

PowerShell

```
$BindingInformation = "\*:443:"

$siteName = "contoso"

$Thumbprint = $certificate.ThumbPrint
```

New-IISSite met sslflag DisableLegacyTLS-eigenschapswaarde:

PowerShell

```
New-IISSite $siteName "$env:systemdrive\\inetpub\\wwwroot"
"\*:443:secure.contoso.com" https $certificate.Thumbprint $DisableLegacyTLS
$storeLocation -passthru
```

Een voorbeeld van het toevoegen van een sitebinding aan een bestaande site en het uitschakelen van verouderde TLS:

PowerShell

```
New-IISSiteBinding -Name "Default Web Site" -BindingInformation
$BindingInformation -CertificateThumbPrint $certificate.Thumbprint -Protocol
https -SslFlag $DisableLegacyTLS, $CCS -Force -verbose
```

Daarnaast kunt u problemen met deze functie oplossen en testen met Netsh:

- Een nieuwe binding toevoegen:

  netsh http add sslcert <standaardparameters> disablelegacytls=enable

- Een bestaande binding bijwerken:

  netsh http update sslcert <reguliere parameters> disablelegacytls=enable

- Controleer of deze is ingesteld op een koppeling:

  netsh http show sslcert <standaardparameters>

  Let op voor Verouderde TLS-versies uitschakelen: Ingesteld/niet ingesteld

Optie 3: C++ HTTP.sys API's (nu beschikbaar)

Naast Verouderde TLS uitschakelen zijn de volgende toevoegingen aangebracht aan HTTP.sys:

- HTTP_SERVICE_CONFIG_SSL_PARAM. DefaultFlags ondersteunt nu de volgende nieuwe waarden:

- HTTP_SERVICE_CONFIG_SSL_FLAG_ENABLE_SESSION_TICKET: Sessieticket in- of uitschakelen voor een bepaald SSL-eindpunt.
- HTTP_SERVICE_CONFIG_SSL_FLAG_LOG_EXTENDED_EVENTS: Schakel uitgebreide gebeurtenislogboeken in of uit voor een specifiek SSL-eindpunt. Aanvullende gebeurtenissen worden vastgelegd in het Windows-gebeurtenislogboek. Er wordt vanaf nu slechts één gebeurtenis ondersteund die wordt geregistreerd wanneer de SSL-handshake mislukt.
- HTTP_SERVICE_CONFIG_SSL_FLAG_DISABLE_LEGACY_TLS: verouderde TLS-versies in- of uitschakelen voor een bepaald SSL-eindpunt. Als u deze vlag instelt, wordt TLS1.0/1.1 voor dat eindpunt uitgeschakeld en worden coderingssuites beperkt die kunnen worden gebruikt voor HTTP2-coderingssuites.
- HTTP_SERVICE_CONFIG_SSL_FLAG_DISABLE_TLS12: TLS1.2 inschakelen/uitschakelen voor een bepaald SSL-eindpunt.
- HTTP_SERVICE_CONFIG_SSL_FLAG_DISABLE_HTTP2: HTTP/2 inschakelen/uitschakelen voor een bepaald SSL-eindpunt.

De eenvoudigste manier om deze functionaliteit per certificaat in C++ in of uit te schakelen, is met de HTTP_SERVICE_CONFIG_SSL_FLAG_DISABLE_LEGACY_TLS vlag die wordt geleverd door de HttpSetServiceConfiguration HTTP.sys-API.

Wanneer Verouderde TLS uitschakelen is ingesteld, worden de volgende beperkingen afgedwongen:

- Schakel de protocollen SSL2, SSL3, TLS1.0 en TLS1.1 uit.
- Schakel versleutelingscoderingen DES, 3DES en RC4 uit (dus wordt alleen AES gebruikt).
- Versleutelingscodering AES uitschakelen met CBC-ketenmodus (dus alleen AES GCM wordt gebruikt).
- Schakel RSA-sleuteluitwisseling uit.
- Schakel DH-sleuteluitwisseling uit met sleutelgrootte kleiner dan 2048.
- Schakel ECDH-sleuteluitwisselingen uit met een sleutelgrootte kleiner dan 224.

Officiële documentatie over deze wijzigingen over docs.Microsoft.com komt binnenkort.

Volgende stappen voor het afdwingen van versies van TLS

Uitschakeling van Verouderde TLS biedt krachtige nieuwe mogelijkheden voor het afdwingen van minimumniveaus van TLS-versies en coderingsuites op specifieke certificaat- en eindpuntbindingen. U moet ook de naamgeving van de certificaten plannen die zijn uitgegeven met deze functionaliteit. Enkele van de overwegingen zijn:

- Wil ik dat het standaardpad naar mijn service-eindpunt vandaag TLS 1.2 afdwingt en een ander certificaat opgeeft als een verouderd back-uptoegangspunt voor gebruikers die TLS 1.0 nodig hebben?
- Moet mijn standaard, al in gebruik zijnde Contoso-certificaat Verouderde TLS uitschakelen? Zo ja, dan moet ik mogelijk een legacy.contoso.com certificaat opgeven en verbinden met een eindpunt dat TLS 1.0 toestaat.
- Hoe kan ik het aanbevolen gebruik van deze certificaten het beste doorgeven aan mijn klanten?

U kunt deze functie gebruiken om te voldoen aan de behoeften van grote groepen klanten, degenen met een verplichting om TLS 1.2+ te gebruiken en degenen die nog steeds aan de migratie werken, weg van TLS 1.0, allemaal zonder extra hardwareuitgaven. Naast de beschikbaarheid van TLS-versiebinding per certificaat in Windows Server 2019, die vanaf vandaag beschikbaar is, zal Microsoft, op basis van klantvraag, de optie om verouderde TLS uit te schakelen beschikbaar maken voor zijn onlineservices.

Feedback

Programmaoverzicht

Artikel • 07-02-2024

Onze toewijding aan vertrouwen en transparantie

De missie van microsoft's Government Security Program (GSP) is het opbouwen van vertrouwen door middel van transparantie. Microsoft erkent dat mensen alleen technologie zullen gebruiken die ze vertrouwen en we streven ernaar om onze toewijding aan het opbouwen van dit vertrouwen te demonstreren via onze transparantie, privacy, naleving en beveiligingsprincipes. Sinds het begin van het programma in 2003 heeft Microsoft inzicht gegeven in onze technologie die overheden en internationale organisaties kunnen gebruiken om zichzelf en hun burgers te beschermen.

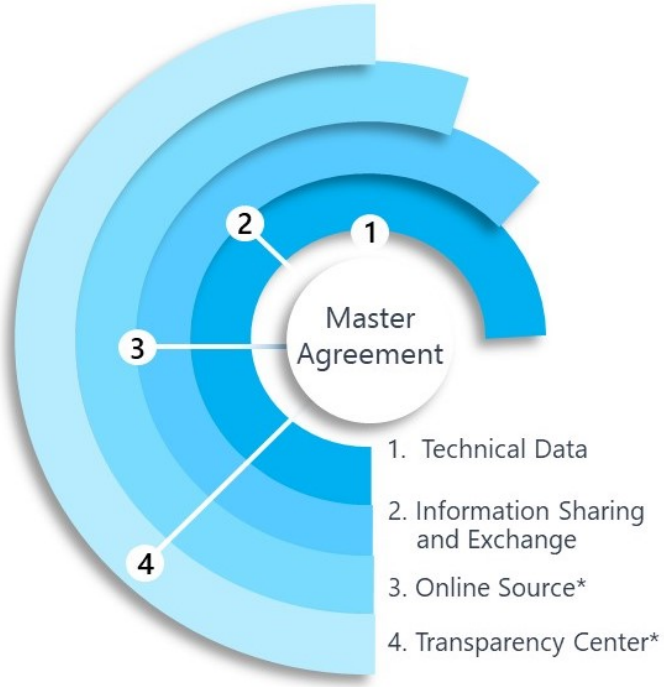
Het GSP is ontworpen om deelnemers de vertrouwelijke beveiligingsinformatie en bronnen te bieden die ze nodig hebben om de producten en services van Microsoft te vertrouwen. Deelnemers omvatten momenteel meer dan 40 landen en internationale organisaties die worden vertegenwoordigd door meer dan 100 agentschappen. Deelname maakt gecontroleerde toegang tot broncode mogelijk, uitwisseling van informatie over bedreigingen en beveiligingsproblemen, betrokkenheid bij technische inhoud over producten en services van Microsoft en toegang tot vijf wereldwijd gedistribueerde Transparantiecentra, die zich in de Verenigde Staten, Singapore, Brazilië, China en Ierland bevinden.

Het doel van de GSP is om regeringen te helpen zichzelf en hun burgers te beschermen door

- Vertrouwen en transparantie inschakelen
- Toegang verlenen tot beveiligingsinformatie over Microsoft-producten en -services
- Gegevens bieden om de bescherming van overheidsinformatietechnologie tegen cyberbedreigingen te verbeteren
- Samenwerking tussen Microsoft-beveiligingsteams en experts op het gebied van cyberbeveiliging van de overheid bevorderen

GSP-aanbiedingen

Oorspronkelijk ontwikkeld om het vertrouwen van de overheid in de stabiliteit en integriteit van Windows te waarborgen, is de GSP uitgebreid in zowel diepte als breedte om het veranderende cyberbeveiligingslandschap aan te pakken, samen met veranderende technologie. Het GSP bestaat uit vier afzonderlijke servicecategorieën of aanbiedingen die elk betrekking hebben op verschillende behoeften en prioriteiten. Deze aanbiedingen omvatten toegang tot technische informatie en documentatie over onze producten en services (zoals cloudservices en algemene criteriacertificeringartefacten); toegang tot informatie over internetveiligheid, cyberbeveiligingsbedreigingen, beveiligingsproblemen en richtlijnen; toegang tot gecontroleerde, online beoordeling van broncode; en toegang tot vijf wereldwijd gedistribueerde Transparantiecentra in de Verenigde Staten, Singapore, Brazilië, China en Ierland. Agentschappen werken samen met lokale Microsoft-vertegenwoordigers en het GSP-team om ervoor te zorgen dat hun doelstellingen het beste worden ondersteund door de juiste combinatie van GSP-aanbiedingen.



Tabel uitvouwen

GSP-autorisatie	Biedt
Technische gegevens	Het aanbod biedt toegang tot informatie over producten en services. Dit omvat technische documentatie over Microsoft-producten'en cloudservices, mogelijkheden voor toegang tot Microsoft-technici om specifieke onderwerpen aan te pakken en beveiligingsspecifieke technische reizen naar Microsoft-faciliteiten voor diepgaande face-to-face-gesprekken.
Gegevens delen en uitwisselen	De aanbieding biedt gegevens over bedreigingen en beveiligingsproblemen en een communicatiekanaal met Microsoft-beveiligings- en responsteams.
Onlinebron*	Met de aanbieding kan online toegang worden geboden om broncode weer te geven. Toegang wordt geboden via een beveiligde webportal die geselecteerde broncode in een alleen-lezen indeling biedt voor Microsoft-producten zoals Windows 11 en Office.
Transparantiecentrum*	Het aanbod biedt agentschappen de mogelijkheid om een veilige faciliteit te bezoeken om diepe niveaus van broncode-inspectie en -analyse uit te voeren. De vijf Transparantiecentra van Microsoft'bevinden zich in de Verenigde Staten, Singapore, Brazilië, China en Ierland.

**Specifieke geschiktheidsvereisten*

Deelnamecriteria

In aanmerking komende instanties nemen gratis deel. GSP-deelnemers moeten:

- Een nationale of federale overheidsinstantie
- Een overeenkomst namens hun regering ondertekenen
- Gebruik de resources die worden aangevraagd
- Kan intellectuele eigendom en vertrouwelijke informatie adequaat beschermen

Deelnameprofiel

GSP-deelnemers hebben doorgaans een missie die zich richt op informatiebeveiliging en -zekerheid. Deelnemers kunnen het volgende omvatten:

- National Computer Emergency Readiness Teams of incident response authorities
- Beveiliging/informatiecontrole en nationale defensiebureaus die bestaan uit:
  - Ontwikkelaars en testers
  - Toepassings- en beveiligingsspecialisten
  - Cryptografiespecialisten
- Openbare veiligheidsbureaus

GSP-lidmaatschap

Lokale Microsoft-vertegenwoordigers werken samen met overheid en internationale organisaties om te bepalen of deelname aan het GSP via een van meer aanbiedingen overeenkomt met de behoeften van dat agentschap, ervan uitgaande dat lokale praktijken en wetten met betrekking tot vertrouwelijkheid, delen van informatie, bescherming van intellectuele eigendom, exportcontrole en gerelateerde gebieden zijn afgestemd op het programma.

Als een land meer dan één agentschap wil laten deelnemen aan het programma, kunnen de lokale Microsoft-vertegenwoordigers en de agentschappen samenwerken om te bepalen of elk agentschap een eigen GSP-aanbieding moet hebben of één bureau moet hebben als primaire GSP-deelnemer die extra instanties sponsort.

Sponsoring

Voor sommige agentschappen is de toegang tot de voordelen van het GSP mogelijk haalbaar via Sponsorship. Er wordt een Sponsorship-relatie gecreëerd wanneer een deelnemende GSP-instantie (het "Sponsorbureau") en een andere overheidsinstantie uit hetzelfde land (het "Gesponsorde agentschap") dezelfde GSP-overeenkomst ondertekent. Een GSP-deelnemer (Sponsorbureau) kan microsoft toestemming vragen om andere overheidsinstanties (gesponsorde instanties) toe te voegen als deelnemers aan de GSP. Het Sponsorbureau neemt de verantwoordelijkheid voor de naleving van alle contractuele voorwaarden van het Gesponsorde Agentschap.

Contact opnemen

Neem contact op met uw lokale Microsoft-vertegenwoordiger voor meer informatie over het Government Security Program.

Technische gegevens

De missie van het Microsoft Government Security Program (GSP) is het opbouwen van vertrouwen door middel van transparantie en het verstrekken van agentschappen de beveiligingsinformatie om hun natie en burgers te beschermen. Sinds het begin van het programma in 2003 heeft Microsoft inzicht gegeven in onze technologie- en beveiligingsartefacten die overheden en internationale organisaties kunnen gebruiken om zichzelf en hun burgers te beschermen tegen beveiligingsrisico's. De aanbieding technische gegevens biedt bewijs van beveiliging via toegang tot een breed scala aan vertrouwelijke technische informatie (exclusief broncode), waardoor deelnemende instanties de betrouwbaarheid kunnen evalueren en een zekere mate van vertrouwen in Microsoft-producten en -services kunnen vaststellen.

Toegang inschakelen

Hebt u vragen over zekerheid? Hebt u toegang nodig? GSP-deelnemers kunnen de aanbieding technische gegevens gebruiken om informatie aan te vragen, resources te onderzoeken en vragen te stellen over de producten en services van Microsoft. Toegang tot een breed scala aan vertrouwelijke informatie omvat:

- **Geschreven materialen** zoals documentatie voor interne technische documenten, architectuurdocumenten en nalevingsrapporten
- **Directe dialoog** met Microsoft-technici of beveiligingsexperts via gecoördineerde vergaderingen door het GSP-team
- **Vroege toegang tot** documentatie over producten en services van Microsoft

Technische reizen

Persoonlijke gesprekken met Microsoft-technici en beveiligingsexperts om beveiligingsvragen te behandelen en deelnemers meer vertrouwen te bieden in Microsoft-producten en -services.

Technische reizen zijn diepgaande technische gesprekken (gehouden in een vergaderingsinstelling) met Microsoft-productgroepen over specifieke beveiligingsonderwerpen die van belang zijn. Om deze te plannen en te vergemakkelijken, werkt het GSP-team nauw samen met het agentschap om inzicht te krijgen in de doelstellingen voor beveiligingscontrole en om te helpen bij het selecteren van de beste Technische Teams van Microsoft om met het agentschap te vergaderen. Technische reizen worden ondergebracht op basis van de beschikbaarheid van het technische team van Micrsoft.

Gebruiksvoorbeelden

- Exemplaren van documentatie over de producten en services van Microsoft die niet op grote schaal worden vrijgegeven. Beveiligingsinhoud met betrekking tot de cloudservices van Microsoft controleren
- Samenwerkingen met ingenieurs in veel verschillende formaten, waaronder Technical Adoption Programs (TAP)
- Documentatie over cloudservicebeveiliging, privacy en naleving, zoals auditrapporten en risicoanalyses
- Algemene criteria-evaluatiedocumenten voor Windows
- Mogelijkheden voor gerichte gesprekken, zoals technische reizen en executive briefings. Voorbeeldonderwerpen over reizen:
  - Cyber Bedreigingsinformatie (DCU)
  - Proces voor reactie op beveiligingsproblemen/incidenten
  - Productbeveiligingsfuncties
  - Telemetrie

Contact opnemen

Neem contact op met uw lokale Microsoft-vertegenwoordiger voor meer informatie over het Government Security Program

---

Last updated on 20-02-2026

Gegevens delen en uitwisselen

De missie van microsoft's Government Security Program (GSP) is het opbouwen van vertrouwen door middel van transparantie. Sinds het begin van het programma in 2003 heeft Microsoft inzicht gegeven in onze technologie- en beveiligingsartefacten, die overheden en internationale organisaties kunnen gebruiken om zichzelf en hun burgers te beschermen. Met de aanbieding Voor het delen van gegevens en Exchange kan Microsoft materialen delen en uitwisselen over beveiligingsrisico's, beveiligingsproblemen, afwijkend gedrag, informatie over malware en beveiligingsproblemen tegen of gerelateerd aan Microsoft-producten en -services.

Dit aanbod brengt groepen en resources samen in de Microsoft-omgeving om overheden te helpen burgers, infrastructuur en organisaties te beschermen.

Het isE-aanbod (Information Sharing and Exchange) biedt

Tabel uitvouwen

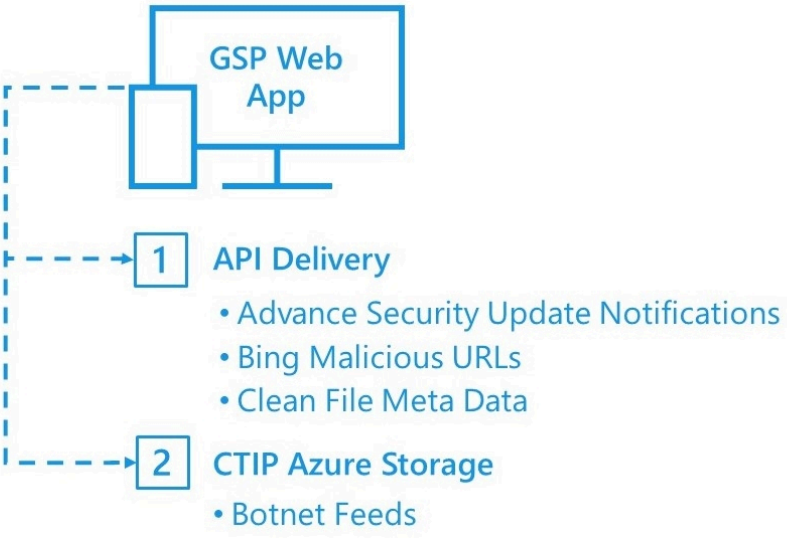
Naam	Het detail
Geavanceerde kennisgeving van beveiligingsproblemen	• 5-daagse geavanceerde kennisgeving van beveiligingsproblemen met releaseopmerkingen en betrokken softwaretabellen • Geavanceerde kennisgeving van 24 uur, inclusief exploitabiliteitsindex
Schadelijke URL's	• Feed van mogelijk kwaadwillende openbaar gerichte servers en services gedetecteerd door Bing-crawlers • Elke drie uur wordt de gegevenscyclus van 5 dagen bijgewerkt
CTIP-botnetfeeds	• Geleverd door de Digital Crimes Unit (DCU) Cyber Threat Intelligence Program (CTIP) • Botnet-gegevens zijn afgestemd op het agentschap (of het domein op het hoogste niveau van landcode in het geval van CERT's) • 4 feeds: Geïnfecteerd apparaat, Command & Control, IoT en domeinen • Bijna realtime geleverd, elk uur of dagelijks (ontdubbeld).
Metagegevens van schone bestanden	• Schoon bestandshash-gegevens worden vaak gebruikt voor toestemmingslijst en forensische analyses. • Elke 3 uur bijgewerkt • Behandelt alle binaire Microsoft-bestanden in het Microsoft Downloadcentrum
Partnerschap	• Informatie-uitwisseling via verschillende forums • Toegang tot de DCU-portal (Digital Crimes Community)

Naam	Het detail
	• Gegevens over bedreigingsinformatie delen met de Digital Crimes Unit (DCU) • Directe betrokkenheid met technische groepen en andere Microsoft-teams, waaronder het Microsoft Security Response Center (MSRC) en Windows Defender Security Intelligence

Levering van gegevensfeeds

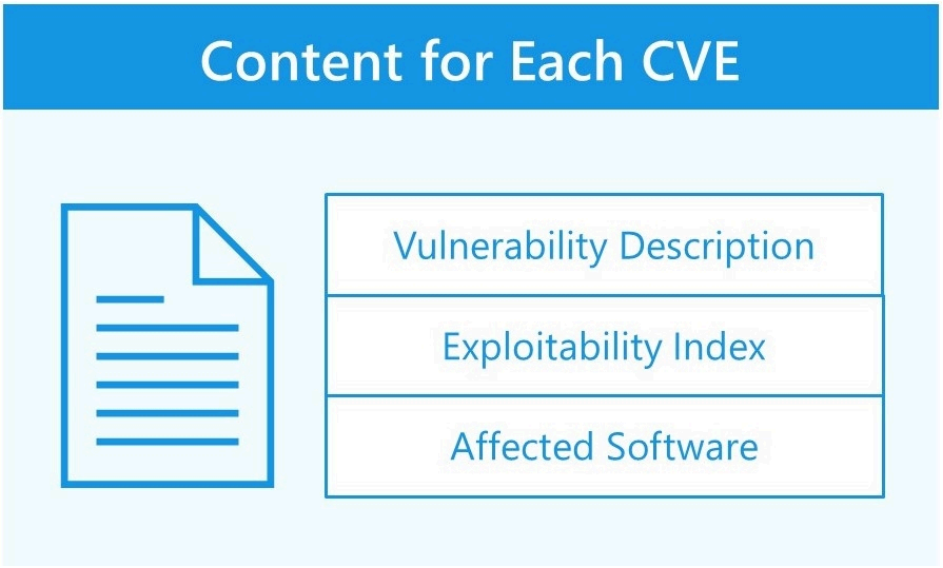
De feeds die worden aangeboden onder de ISE-autorisatie bevinden zich in verschillende groepen, waaronder het **Microsoft Security Response Center (MSRC)**, de **Digital Crimes Unit (DCU)**, **Bing** en **Product Release and Security Services (PRSS).**

Het **GSP-team biedt een webtoepassing waarmee GSP-instanties** toegang hebben tot de ISE-gegevensfeeds vanuit één interface. Alle communicatie met gevoelige gegevens wordt versleuteld.

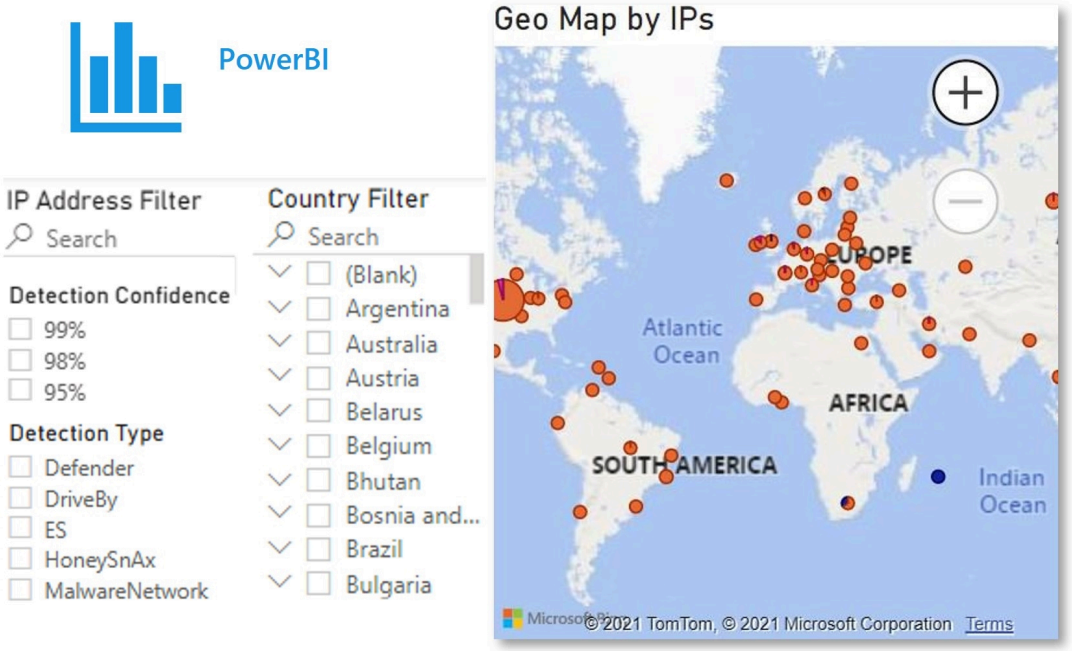


Beschrijvingen van gegevensgebruik

Geavanceerde beveiligingsupdatemelding In het kennisgevingspakket worden alle CVEs (Common Vulnerabilities and Exposures) vermeld die in de release worden behandeld. Elke CVE bevat een set informatie, waaronder de beschrijving van beveiligingsproblemen (inclusief metrische gegevens), exploitabiliteitsindex en beïnvloede software.



Bing Schadelijke URL's De Bing Schadelijke URL-feed bevat openbaar gerichte servers of services die zijn geïdentificeerd als mogelijk schadelijk. Nieuwe bestanden worden elke drie uur geüpload; volledige gegevenssets worden binnen 5 dagen gegenereerd. Veel agentschappen importeren de JSON-bestanden rechtstreeks in hun bestaande hulpprogramma's voor bedreigingsinformatieanalyse.



Threat	Detection Type	Detection Confidence	Description
Defender	SUSPICIOUS JAVASCRIPT or HTML	98%	Potentially compromised URLs that have been observed hosting Malware by Windows Defender or other anti-virus providers - Documents indexed in Bing are scanned by Windows Defender and any URL flagged by Windows Defender is recorded.
DriveBy	DRIVE-BY-DOWNLOAD	99%	Potentially compromised URLs that have been observed hosting DriveBy attacks - Instrumented browsers in a sandbox environment solicit attacks from malicious players and any URL showing malicious interactions is recorded.
ES	EXPLOIT SERVER	95%	Potentially compromised URLs that have been observed directing traffic to a known malicious server hosting exploit code - Through detailed analysis of network traces derived from crawl data, and correlating them with malware detection data we have, a set of malicious server hosting are inferred and any URL directing traffic to them is recorded.
HoneySnAx	SUSPICIOUS ACTIVE-X	99%	Potentially compromised URLs that have been observed as hosting malicious ActiveX code - Browsers with ActiveX instrumentation layer in a sandbox environment solicit attacks from malicious players and any URL showing malicious interactions with ActiveX is recorded.
MalwareNetwork	MALWARE DISTRIBUTION NETWORK	95%	Potentially compromised URLs that have been identified as being a part of a malware distribution network - Through detailed analysis of network traces derived from crawl data, and correlating them with malware detection data we have , URL patterns distributing malware are inferred as Malware Distribution Network and any URL directing traffic to this network is recorded.

Clean File Meta Data (CFMD)

De CFMD-feed (Clean File Meta Data) bevat cryptografische handtekeningen (SHA256 hashes) voor de bestanden in Microsoft-producten. Deze worden vaak gebruikt bij forensisch onderzoek van mogelijk aangetaste apparaten en voor het toestaan/weigeren van bestandsuitvoering in kritieke systemen.

filename	size	releasedate	sha256
2C871FF86C5829833C3D17374F9A37ADB1E6B68B.exe	295360	10/30/2021 1:00:34 PM	A92A09C
93AE425B46935755C21D56B53AB49070096823EA.exe	532912	10/30/2021 1:00:59 PM	2FF1587[
A768EB3A2B77AEC37088C06586F3567EC431E4D3.exe	291248	10/30/2021 1:00:59 PM	4AB9E17
B1372CAAAA63CE6F5D01A7341437F33D061F73F9.exe	291248	10/30/2021 1:00:59 PM	F1FA016
C371279332145D107CE2DF7E370D130336DE312F.exe	303560	10/30/2021 1:01:38 PM	7374F04
EE27FDAA91DB919F4B2789109FCF08561C7D17D5.exe	295344	10/30/2021 1:01:38 PM	83DDD2
07F1C5B804624A7E63A00DCF1207335DE77167FF.exe	311728	10/30/2021 1:02:07 PM	C95079B
ae6cbbf5-a285-44cb-8aae-73b1b8e91646	186	10/30/2021 1:17:11 PM	B6C523D

CTIP Botnet-feeds: Geïnfecteerde gegevensstroom

De DCU biedt gecompromitteerde botnetgegevens van slachtoffers via de CTIP threat intelligence-service Geïnfecteerde apparaatgegevensfeed van de DCU, om netwerkbeveiligingsscenario's voor CTIP-abonnees mogelijk te maken en om het herstel van de aangetaste systemen te vergemakkelijken met het doel het aantal geïnfecteerde systemen op internet te verminderen. Andere feeds zijn onder andere de lijsten Command and Control (C2), IoT en Domains die vaak worden gebruikt om de verkeersstroom te beperken tot bekende malwarenetwerken via firewalls en beschermende DNS.

SourcePort	AsnOrg Name	Country Code	City	Date Time Received UTC	Malware	ThreatCode
63601	UUNET	US	Paterson	11/13/2021 11:19:23 PM	Gamarue	B66-SS-Gamarue
63613	UUNET	US	Paterson	11/13/2021 11:20:08 PM	Avalanche	B67-SS-Gamarue
42096	UUNET	US	Queens	11/14/2021 12:15:38 AM	Bladabindi+Jenxcus	B106-Mobibez
61873	UUNET	US	Lanham	11/15/2021 12:33:13 AM	Bladabindi+Jenxcus	B106-Spybot
63167	UUNET	US	Lanham	11/14/2021 12:58:38 AM	Bladabindi+Jenxcus	B106-Knowlog
63487	UUNET	US	Lanham	11/14/2021 12:59:14 AM	Bladabindi+Jenxcus	B106-Spybot
51058	UUNET	US	Severna	11/13/2021 11:34:38 PM	Avalanche	B67-SS-GENERIC

Domain	Country	Malware	ThreatCode	Threat Confidence
aaaelscmh.ac		Necurs	B80-Domains-DGA	High
aaanhbpukn.in	India	Necurs	B80-Domains-DGA	High
aaaoyyxqfffw.in	India	Necurs	B80-Domains-DGA	High
aaasdvokjknth.net	United States of America (the)	Necurs	B80-Domains-DGA	High
aaduubgfsmkjoyhuifa.in	India	Necurs	B80-Domains-DGA	High
aaexxhyv.net	United States of America (the)	Necurs	B80-Domains-DGA	High
aagcqrqnbdhyvapgrsyd.pr o	United States of America (the)	Necurs	B80-Domains-DGA	High

Neem contact met ons op

Neem contact op met uw lokale Microsoft-vertegenwoordiger voor meer informatie over het Government Security Program.

Last updated on 27-03-2026

Onlinetoegang tot broncode

Artikel • 07-02-2024

Sinds de lancering in 2003 heeft het Government Security Program (GSP) overheden en internationale organisaties de mogelijkheid geboden om toegang te krijgen tot en de broncode te inspecteren voor verschillende Microsoft-producten. Het onlinebronaanbod biedt meer bewijs van beveiliging door toegang tot productbroncode mogelijk te maken met behulp van Code Center Premium (CCP), een beveiligde webportal. Beschikbare producten zijn Onder andere **Windows**, **Office**, **SharePoint**, **Exchange** en **SQL Server**. Personen van het agentschap die Microsoft Entra ID gebruiken, krijgen alleen-lezentoegang tot CCP om te bladeren, te zoeken en te verwijzen naar de broncode van het product.

Code Center Premium

Met de onlinebronaanbiedingen en de CTP-site kunnen GSP-deelnemers afzonderlijke functies van systeemonderdelen, interactie met onderdelen en mogelijkheden voor beveiliging en betrouwbaarheid evalueren. Agentschappen kunnen CCP gebruiken om broncodestructuren te doorzoeken met behulp van uitgebreide zoekfunctionaliteit. Om de waarde van persoonlijke bezoeken te verbeteren, moeten agentschappen doorgaans de broncode op afstand controleren in CCP voordat ze naar een Transparantiecentrum gaan.

Voorbeeldvoorbeelden van onlinebron:

- Begrijpen hoe specifieke functies werken
- Microsoft-coderingsprocedures controleren
- Beveiligingsproblemen zoeken of valideren

Contact opnemen

Neem contact op met uw lokale Microsoft-vertegenwoordiger voor meer informatie over het Government Security Program.

Transparantiecentra

Artikel • 30-10-2024

Microsoft streeft ernaar een ongekend transparantieniveau te bieden via het Government Security Program (GSP), gericht op het helpen van klanten om vertrouwen te krijgen in de integriteit en zekerheid van de producten en services waarop ze vertrouwen. Transparency Centers (TCs) zijn een uitstekende showcase om de inzet van Microsoft voor beveiliging en transparantie te demonstreren.

Een veilige faciliteit voor inspectie en analyse

Transparantiecentra bieden GSP-deelnemers de mogelijkheid om een veilige faciliteit te bezoeken om diepgaande niveaus van document- en broncodeinspectie en -analyse uit te voeren. Om deze inspanningen te ondersteunen, heeft Microsoft vier Transparantiecentra over de hele wereld: Verenigde Staten, Ierland, Singapore en Brazilië. Deelnemers hebben toegang tot documenten en broncode en een omgeving voor diepgaande inspectie met industriestandaardhulpprogramma's. Momenteel bieden de Transparency Centers broncode voor producten zoals **Windows, Windows Server, Office, Exchange Server, SQL Server en SharePoint Server**.

Bezoeken aan het centrum

Elk bezoek aan een Transparantiecentrum is afgestemd op de unieke doelstellingen van wat een agentschap wil bereiken. Bezoeken kunnen van één dag tot twee weken duren, afhankelijk van de behoeften van een bureau en worden gepland op basis van de beschikbaarheid van de faciliteit. Face-to-face- of teleconference-uitwisselingen met Microsoft-technici zijn mogelijk ook beschikbaar en kunnen nuttig zijn tijdens transparantiecentrumbezoeken voor agentschappen die ook deelnemen aan het gedeelte Technische gegevens van het programma.

Omgeving en hulpprogramma's

De omgeving en hulpprogramma's voor broncode-evaluatie omvatten:

- Privénetwerk met toegewezen servers en clients
- OpenGrok open source zoeken en kruisverwijzingen
- PowerShell, Visual Studio
- HeyRays IDA Disassembler en Decompiler
- Hulpprogramma's van de deelnemer en goedgekeurd door Microsoft
- SysInternals

Gebruiksvoorbeelden

- Evaluatie van de implementatie van de volgende generatie cryptografie en voorbereiding op de nationale cryptografie-implementatie
- Controle van SSL- en TCP/IP-implementatie
- Inspectie van de bron van generatoren voor willekeurige getallen
- Overzicht van het buildproces van Microsoft
- Beoordeling van afzonderlijke binaire bestanden om te vergelijken met verzonden binaire bestanden

Contact opnemen

Neem contact op met uw lokale Microsoft-vertegenwoordiger voor meer informatie over het Government Security Program.

Feedback

SDL-beveiligingsfoutbalk (voorbeeld)

Artikel • 22-03-2025

Opmerking: dit voorbeelddocument is alleen bedoeld voor illustratiedoeleinden. De onderstaande inhoud bevat basiscriteria die u moet overwegen bij het maken van beveiligingsprocessen. Het is geen volledige lijst van activiteiten of criteria en mag niet als zodanig worden behandeld.

Raadpleeg de definities van termen in deze sectie.

Server

Raadpleeg de Denial of Service Matrix voor een volledige matrix met doS-serverscenario's.

De serverbalk is meestal niet geschikt wanneer gebruikersinteractie deel uitmaakt van het exploitatieproces. Als er alleen een kritiek beveiligingsprobleem bestaat op serverproducten en wordt misbruikt op een manier die interactie van de gebruiker vereist en leidt tot inbreuk op de server, kan de ernst worden verminderd van Kritiek naar Belangrijk in overeenstemming met de NETTE/gegevensdefinitie van uitgebreide gebruikersinteractie die aan het begin van de ernst van de client wordt gepresenteerd.

Tabel uitvouwen

Server	

Cruciaal / Kritisch	Serveroverzicht: netwerkwormen of *onvermijdelijke* gevallen waarin de server 'eigendom' is.
	• Uitbreiding van bevoegdheden: de mogelijkheid om willekeurige code *uit te voeren of* meer bevoegdheden te verkrijgen dan geautoriseerd
	  ◦ Anonieme gebruiker op afstand
	  ◦ Voorbeelden:
	  ◦ Onbevoegde bestandssysteemtoegang: willekeurig schrijven naar het bestandssysteem
	  ◦ Uitvoering van willekeurige code
	  ◦ SQL-injectie (waarmee code kan worden uitgevoerd)
	  ◦ Alle schendingen van schrijftoegang (AV), misbruikbare lees AV's of overloop van gehele getallen in op afstand anoniem aanroepbare code

Belangrijk	Serversamenvatting: niet-standaard kritieke scenario's of gevallen waarin er maatregelen bestaan die kunnen helpen *kritieke scenario's te voorkomen*.

Server

• Denial of service: Moet 'eenvoudig te exploiteren' zijn door een kleine hoeveelheid gegevens te verzenden of anderszins snel te worden geïnduceerd
  ◦ Anoniem
  ◦ Voortdurende Denial of Service
  ◦ Voorbeelden:
  ◦ Het verzenden van één schadelijk TCP-pakket resulteert in een Blue Screen of Death (BSoD)
  ◦ Een klein aantal pakketten verzenden dat een servicefout veroorzaakt
  ◦ Tijdelijke Denial-of-Service met versterking
  ◦ Voorbeelden:
  ◦ Een klein aantal pakketten verzenden waardoor het systeem gedurende een bepaalde periode onbruikbaar is
  ◦ Een webserver (zoals IIS) is een minuut of langer niet beschikbaar
  ◦ Eén *externe client* die alle beschikbare resources (sessies, geheugen) op een server verbruikt door sessies tot stand te brengen en open te houden
  ◦ Geverifieerd
  ◦ Permanente DoS ***tegen een waardevol bezit***
  ◦ Voorbeeld:
  ◦ Een klein aantal pakketten verzenden dat een servicefout veroorzaakt voor een ***asset*** met een hoge waarde in serverfuncties (certificaatserver, Kerberos-server, domeincontroller), bijvoorbeeld wanneer een door een domein geverifieerde gebruiker een DoS op een domeincontroller kan uitvoeren
 	
• Uitbreiding van bevoegdheden: de mogelijkheid om willekeurige code *uit te voeren of* om meer bevoegdheden te verkrijgen dan bedoeld
  ◦ Geverifieerde gebruiker op afstand
  ◦ Lokale geverifieerde gebruiker (Terminal Server)
  ◦ Voorbeelden:
  ◦ Onbevoegde bestandssysteemtoegang: willekeurig schrijven naar het bestandssysteem
  ◦ Uitvoering van willekeurige code
  ◦ Alle schrijf-AV's, exploiteerbare lees-AV's of overloop van gehele getallen in code die toegankelijk zijn voor externe of lokale geverifieerde gebruikers die geen beheerders zijn (beheerdersscenario's hebben geen beveiligingsproblemen per definitie, maar zijn nog steeds betrouwbaarheidsproblemen.)
• Openbaarmaking van informatie (gericht)
  ◦ Gevallen waarin de aanvaller informatie *kan vinden en lezen vanaf elke locatie* op het systeem, inclusief systeeminformatie die niet is bedoeld of ontworpen om te worden blootgesteld
  ◦ Voorbeelden:
  ◦ Openbaarmaking van persoonsgegevens (PII)
  ◦ Openbaarmaking van PII (e-mailadressen, telefoonnummers, creditcardgegevens)

Server	
	○ Aanvaller kan PII verzamelen zonder toestemming van de gebruiker of op een geheime manier • Adresvervalsing (spoofing) ○ Een entiteit (computer, server, gebruiker, proces) kan zich voordoen ***als een* specifieke entiteit** (gebruiker of computer) van zijn/haar keuze. ○ Voorbeelden: ○ Webserver gebruikt clientcertificaatverificatie (SSL) onjuist om een aanvaller te laten identificeren als een gebruiker van zijn/haar keuze ○ Nieuw protocol is ontworpen om externe clientverificatie te bieden, maar er bestaat een fout in het protocol waarmee een kwaadwillende externe gebruiker kan worden gezien als een andere gebruiker van zijn of haar keuze • Manipulatie ○ Wijziging van gegevens ***met een hoge waarde*** in ***een gemeenschappelijk of standaardscenario*** waarbij de wijziging zich blijft voordoen na het opnieuw opstarten van de betreffende software ○ Permanente of aanhoudende wijziging van gebruikers- of systeemgegevens die ***in een gemeenschappelijk of standaardscenario*** worden gebruikt. ○ Voorbeelden: ○ Wijziging van toepassingsgegevensbestanden of -databases in een gemeenschappelijk of standaardscenario, zoals geverifieerde SQL-injectie ○ Proxycachevergiftiging in een veelvoorkomend of standaardscenario ○ Wijziging van besturingssysteem- of toepassingsinstellingen zonder gebruikerstoestemming in een algemeen of standaardscenario • Beveiligingsfuncties: Het breken of omzeilen van beveiligingsvoorzieningen. Houd er rekening mee dat een beveiligingsprobleem in een beveiligingsfunctie standaard 'Belangrijk' is, maar de classificatie kan worden aangepast op basis van andere overwegingen, zoals beschreven in de SDL-bugbalk. ○ Voorbeelden: ○ Een firewall uitschakelen of omzeilen zonder gebruikers te informeren of toestemming te krijgen ○ Een firewall opnieuw configureren en verbindingen met andere processen toestaan
Matig	• Dienstweigering ○ Anoniem ○ Tijdelijke DoS zonder versterking in een standaard/algemene installatie. ○ Voorbeeld: ○ ***Meerdere externe clients*** die alle beschikbare resources (sessies, geheugen) op een server gebruiken door sessies tot stand te brengen en ze open te houden ○ Geverifieerd ○ Permanente DoS ○ Voorbeeld: ○ Een aangemelde Exchange-gebruiker kan een specifiek e-mailbericht verzenden en daarmee de Exchange-server laten

Server
crashen; de crash is **niet** te wijten aan een schrijf-AV, een exploiteerbare lees-AV of een overloop van gehele getallen. ◦ Tijdelijke DoS met amplification in een standaard/algemene installatie ◦ Voorbeeld: ◦ Gewone SQL Server-gebruiker voert een opgeslagen procedure uit die door een bepaald product is geïnstalleerd en verbruikt een paar minuten 100% van de CPU • Openbaarmaking van informatie (gericht) ◦ Gevallen waarin de aanvaller eenvoudig informatie over het systeem *kan lezen vanaf* ***specifieke locaties***, inclusief systeeminformatie, die niet bedoeld of ontworpen is om te worden blootgesteld. ◦ Voorbeeld: ◦ Gerichte openbaarmaking van anonieme gegevens ◦ Gerichte openbaarmaking van het bestaan van een bestand ◦ Gerichte openbaarmaking van een versienummer van een bestand • Adresvervalsing (spoofing) ◦ Een entiteit (computer, server, gebruiker, proces) kan zich voordoen als een andere, willekeurige entiteit die niet specifiek kan worden geselecteerd. ◦ Voorbeeld: ◦ De client wordt correct geverifieerd bij de server, maar server stuurt een sessie terug van een andere willekeurige gebruiker die op hetzelfde moment met de server is verbonden • Manipulatie ◦ Permanente of aanhoudende wijziging van gebruikers- of systeemgegevens ***in een specifiek scenario*** ◦ Voorbeelden: ◦ Wijziging van toepassingsgegevensbestanden of -databases ***in een specifiek scenario*** ◦ Proxycachevergiftiging ***in een specifiek scenario*** ◦ Wijziging van besturingssysteem-/toepassingsinstellingen zonder toestemming van de gebruiker ***in een specifiek scenario*** ◦ Tijdelijke wijziging van gegevens in een algemeen of standaardscenario dat niet behouden blijft na het opnieuw opstarten van het besturingssysteem/de toepassing-/sessie • Beveiligingsgaranties: ◦ Een beveiligingsgarantie is een beveiligingsfunctie of een andere productfunctie die klanten verwachten beveiliging te bieden. Communicaties hebben (expliciet of impliciet) gemeld dat klanten kunnen vertrouwen op de integriteit van de functie, en dat is wat het een beveiligingsgarantie biedt. Beveiligingsbulletins worden vrijgegeven voor een tekortkoming in een zekerheidsgarantie die het vertrouwen of de afhankelijkheid van de klant aantasten. ◦ Voorbeelden: ◦ Processen die worden uitgevoerd met normale 'gebruikersbevoegdheden' kunnen geen beheerdersbevoegdheden krijgen, tenzij beheerderswachtwoorden/-referenties zijn opgegeven via opzettelijk geautoriseerde methoden.

Server	
	◦ JavaScript op internet dat wordt uitgevoerd in Internet Explorer kan niets beheren van het hostbesturingssysteem, tenzij de gebruiker de standaardbeveiligingsinstellingen expliciet heeft gewijzigd.
Laag	• Openbaarmaking van informatie (ongericht) ◦ Runtime-informatie ◦ Voorbeeld: ◦ Lek van willekeurig heap-geheugen • Manipulatie ◦ Tijdelijke wijziging van gegevens *in een specifiek scenario* dat niet behouden blijft na het opnieuw opstarten van het besturingssysteem/de toepassing

Klant

Uitgebreide gebruikersactie wordt gedefinieerd als:

- Gebruikersinteractie kan alleen plaatsvinden in clientgestuurd scenario.
- Normale, eenvoudige gebruikersacties, zoals het bekijken van een voorbeeld van e-mail, het weergeven van lokale mappen of bestandsshares, zijn geen uitgebreide gebruikersinteractie.
- 'Uitgebreid' omvat gebruikers die handmatig naar een bepaalde website navigeren (bijvoorbeeld typen in een URL) of door te klikken op een ja/nee-beslissing.
- Gebruikers die op links in e-mails klikken, vallen onder "Niet uitgebreid".
- **NEAT-kwalificatie** (*alleen* van toepassing op waarschuwingen). Overduidelijk is de UX:
  - **N**ecessary (is het echt nodig dat de gebruiker een beslissing moet nemen?)
  - **E**xplained (Bevat de UX alle informatie die de gebruiker nodig heeft om deze beslissing te nemen?)
  - **Actiegericht** (Zijn er stappen die gebruikers kunnen nemen voor goede beslissingen in zowel goedaardige als kwaadwillende scenario's?)
  - **Getest** (Is de waarschuwing beoordeeld door meerdere personen om ervoor te zorgen dat mensen begrijpen hoe ze moeten reageren op de waarschuwing?)
- **Verduidelijking:** Houd er rekening mee dat het effect van uitgebreide gebruikersinteractie niet één niveauvermindering is in ernst, maar is en is een

vermindering van de ernst in bepaalde omstandigheden waarbij de woordgroep uitgebreide gebruikersinteractie op de bugbalk wordt weergegeven. Het doel is om klanten te helpen bij het onderscheiden van snel verspreidende en wormbare aanvallen van die, waarbij door interactie van de gebruiker, de aanval wordt vertraagd. Met deze bugbalk kunt u de uitbreiding van bevoegdheden onder Belangrijk niet verminderen vanwege gebruikersinteractie.

Tabel uitvouwen

Klant	

Cruciaal	Clientoverzicht:
	• Netwerkwormen of onvermijdelijke browse-/gebruiksscenario's waarbij de client zonder waarschuwingen of prompts 'overgenomen' is.
	• Uitbreiding van bevoegdheden (extern): de mogelijkheid om willekeurige code *uit te voeren of* om meer bevoegdheden te verkrijgen dan bedoeld
	  ◦ Voorbeelden:
	  ◦ Onbevoegde toegang tot het bestandssysteem: schrijven naar het bestandssysteem
	  ◦ Uitvoering van willekeurige code zonder uitgebreide gebruikersactie
	  ◦ Alle schrijf-AV's, exploiteerbare lees-AV's, stack-overloop of gehele getallen in extern aanroepbare code (***zonder*** uitgebreide gebruikersactie)

Belangrijk	Clientoverzicht:
	• Veelvoorkomende browse-/gebruiksscenario's waarin de client wordt beheerst **met** waarschuwingen of prompts, of door uitgebreide acties zonder prompts. Houd er rekening mee dat dit geen onderscheid maakt tussen de kwaliteit/bruikbaarheid van een prompt en waarschijnlijkheid dat een gebruiker door de prompt kan klikken, maar alleen dat er een prompt van een bepaald formulier bestaat.
	• Verhoging van rechten (op afstand)
	  ◦ Uitvoering van willekeurige code *met* uitgebreide gebruikersactie
	  ◦ Alle schrijf-AV's, exploiteerbare lees-AV's of gehele getallen in ***externe*** aanroepbare code (***met*** uitgebreide gebruikersactie)
	• Uitbreiding van bevoegdheden (lokaal)
	  ◦ Lokale gebruiker met lage bevoegdheden kan zichzelf uitbreiden naar een andere gebruiker, beheerder of lokaal systeem.
	  ◦ Alle schrijf-AV's, exploiteerbare lees-AV's of gehele getallen in **lokale** aanroepbare code
	• Openbaarmaking van informatie (gericht)
	  ◦ Gevallen waarin de aanvaller informatie op het systeem kan vinden en lezen, inclusief systeeminformatie die niet is bedoeld of ontworpen om te worden blootgesteld.
	  ◦ Voorbeeld:
	  ◦ Onbevoegde toegang tot het bestandssysteem: lezen vanuit het bestandssysteem

Klant
- Openbaarmaking van PII   - Openbaarmaking van PII (e-mailadressen, telefoonnummers)   - Telefoonscenario's voor thuisgebruik   - Dienstweigering   - Een DoS-aanval door systeemcorruptie vereist het opnieuw installeren van het systeem en/of de componenten.   - Voorbeeld: - Een webpagina bezoeken veroorzaakt beschadiging van het register waardoor de computer niet kan worden gestart   - DoS-aanval zonder directe interactie   - Criteria: - Niet-geauthenticeerde systeem DoS   - Standaardblootstelling   - Geen standaardbeveiligingsfuncties of grensbeperking (firewalls)   - Geen gebruikersinteractie   - Geen audit- en strafproces   - Voorbeeld: - Drive-by Bluetooth-systeem DoS of SMS op een mobiele telefoon   - Adresvervalsing (spoofing)   - De mogelijkheid voor aanvaller om een gebruikersinterface te presenteren die verschilt van maar visueel identiek is aan de gebruikersinterface waarop gebruikers *moeten vertrouwen om geldige vertrouwensbeslissingen* te nemen in een *standaard/gemeenschappelijk scenario*. Een vertrouwensbeslissing wordt gedefinieerd wanneer de gebruiker een actie onderneemt die denkt dat bepaalde informatie wordt gepresenteerd door een bepaalde entiteit: het systeem of een specifieke lokale of externe bron.   - Voorbeelden: - Een andere URL weergeven in de adresbalk van de browser van de URL van de site die de browser daadwerkelijk weergeeft in een ***standaard/gemeenschappelijk scenario***   - Een venster weergeven op de adresbalk van de browser die identiek is aan een adresbalk, maar valse gegevens in een ***standaard-/gemeenschappelijk scenario weergeven***   - Een andere bestandsnaam weergeven in 'Wilt u dit programma uitvoeren?' dialoogvenster dan dat van het bestand dat daadwerkelijk wordt geladen in een ***standaard/gebruikelijk scenario***   - Een 'vals' aanmeldingsprompt weergeven om gebruikers- of accountreferenties te verzamelen   - Manipulatie   - Permanente wijziging van alle gebruikersgegevens of gegevens die worden gebruikt om vertrouwensbeslissingen te nemen in een algemeen of standaardscenario dat zich blijft voordoen na het opnieuw opstarten van het besturingssysteem/de toepassing.   - Voorbeelden: - Vergiftiging van webbrowsercache   - Wijziging van belangrijke instellingen voor het besturingssysteem/de toepassing zonder toestemming van de gebruiker   - Wijziging van gebruikersgegevens

Klant	

	• Beveiligingsfuncties: Het doorbreken of omzeilen van enige beschikbare beveiligingsfunctie.
	  ◦ Voorbeelden:
	  ◦ Een firewall uitschakelen of overslaan met het informeren van de gebruiker of het verkrijgen van toestemming
	  ◦ Een firewall opnieuw configureren en verbinding met andere processen toestaan
	  ◦ Zwakke versleuteling gebruiken of de sleutels opslaan in tekst zonder opmaak
	  ◦ Omzeiling van AccessCheck
	  ◦ Bitlocker omzeilen; bijvoorbeeld een deel van de harde schijf niet versleutelen
	  ◦ Syskey bypass, een manier om de syskey te decoderen zonder het wachtwoord

Matig	• Dienstweigering
	  ◦ Permanente DoS vereist een cold reboot of veroorzaakt een Blue Screen/bugcontrole.
	  ◦ Voorbeeld:
	  ◦ Als u een Word-document opent, krijgt de computer een blauwscherm/foutcontrole.
	• Openbaarmaking van informatie (gericht)
	  ◦ Gevallen waarin de aanvaller informatie op het systeem *van bekende locaties* kan lezen, inclusief systeeminformatie die niet bedoeld is of die is ontworpen om te worden blootgesteld.
	  ◦ Voorbeelden:
	  ◦ Gericht bestaan van bestand
	  ◦ Versienummer van doelbestand
	• Adresvervalsing (spoofing)
	  ◦ De mogelijkheid voor aanvallers om een gebruikersinterface te presenteren die verschilt van maar visueel identiek is aan de gebruikersinterface die gebruikers *gewend zijn om te vertrouwen* in *een specifiek scenario*. 'Gewend aan vertrouwen' wordt gedefinieerd als alles waarmee een gebruiker vaak bekend is op basis van normale interactie met het besturingssysteem of de toepassing, maar wordt meestal niet gezien als een 'vertrouwensbeslissing'.
	  ◦ Voorbeelden:
	  ◦ Vergiftiging van webbrowsercache
	  ◦ Wijziging van belangrijke instellingen voor het besturingssysteem/de toepassing zonder toestemming van de gebruiker
	  ◦ Wijziging van gebruikersgegevens

Laag	• Dienstweigering
	  ◦ Tijdelijke DoS vereist opnieuw opstarten van de toepassing.
	  ◦ Voorbeeld:
	  ◦ Als u een HTML-document opent, loopt Internet Explorer vast
	• Adresvervalsing (spoofing)

Klant
◦ De mogelijkheid voor aanvaller om een gebruikersinterface te presenteren die verschilt van maar visueel identiek is aan de gebruikersinterface *die één deel van een groter aanvalsscenario* is. ◦ Voorbeeld: ◦ Gebruiker moet naar een 'schadelijke' website gaan, op een knop in een vervalst dialoogvenster klikken en is vervolgens vatbaar voor een beveiligingsprobleem door een andere browserfout. • Manipulatie ◦ Tijdelijke wijziging van gegevens die niet behouden blijven na het opnieuw opstarten van het besturingssysteem/de toepassing. ◦ Openbaarmaking van informatie (niet-gericht) ◦ Voorbeeld: ◦ Lek van willekeurig heap-geheugen

Definitie van termen

Geverifieerde
Elke aanval die verificatie door het netwerk moet omvatten. Dit betekent dat logboekregistratie van een bepaald type moet kunnen plaatsvinden, zodat de aanvaller kan worden geïdentificeerd.

anoniem
Elke aanval die niet hoeft te worden geverifieerd om te worden voltooid.

client
Software die lokaal wordt uitgevoerd op één computer of software die toegang heeft tot gedeelde resources die door een server via een netwerk worden geleverd.

standaard/algemeen
Alle functies die out-of-the-box actief zijn of die meer dan 10 procent van de gebruikers bereiken.

scenario
Alle functies waarvoor speciale aanpassingen of use cases nodig zijn om deze in te schakelen, waardoor minder dan 10 procent van de gebruikers wordt bereikt.

server
Computer die is geconfigureerd voor het uitvoeren van software die wacht op en voldoet aan aanvragen van clientprocessen die worden uitgevoerd op andere computers.

Kritiek
Een beveiligingsprobleem dat als het hoogste potentieel voor schade wordt beoordeeld.

Belangrijk
Een beveiligingsprobleem dat als aanzienlijk potentieel voor schade wordt beoordeeld, maar minder dan Kritiek.

Gematigd
Een beveiligingsprobleem dat als gemiddelde kans op schade wordt beoordeeld, maar minder dan Belangrijk.

Laag
Een beveiligingsprobleem dat zou worden beoordeeld als een laag potentieel voor schade.

gerichte openbaarmaking van informatie
Mogelijkheid om opzettelijk gewenste informatie te selecteren (doel).

tijdelijke DoS-aanval
Een tijdelijke DoS is een situatie waarin aan de volgende criteria wordt voldaan:

- Het doel kan geen normale bewerkingen uitvoeren vanwege een aanval.
- De reactie op een aanval is ongeveer dezelfde grootte als de grootte van de aanval.
- Het doel keert terug naar het normale functionaliteitsniveau kort nadat de aanval is voltooid. De exacte definitie van 'binnenkort' moet voor elk product worden geëvalueerd.

Een server reageert bijvoorbeeld niet terwijl een aanvaller voortdurend een stroom pakketten via een netwerk verzendt en de server een paar seconden na het stoppen van de pakketstroom weer normaal wordt.

tijdelijke DoS met amplificatie

Een tijdelijke DoS met amplificatie is een situatie waarin aan de volgende criteria wordt voldaan:

- Het doel kan geen normale bewerkingen uitvoeren vanwege een aanval.
- De reactie op een aanval is een grootte die groter is dan de grootte van de aanval.
- Het doel keert terug naar het normale functionaliteitsniveau nadat de aanval is voltooid, maar het duurt enige tijd (misschien een paar minuten).

Als u bijvoorbeeld een schadelijk pakket van 10 byte kunt verzenden en een antwoord van 2048k op het netwerk kunt veroorzaken, voert u een DoS-aanval uit door onze aanvalsinspanning te versterken.

permanente DoS

Een permanente DoS is een doS die een beheerder nodig heeft om alle of onderdelen van het systeem te starten, opnieuw te starten of opnieuw te installeren. Elk beveiligingsprobleem dat automatisch opnieuw wordt opgestart, is ook een permanente DoS.

Matrix voor Denial of Service (Server)

Tabel uitvouwen

Geverifieerd versus anonieme aanval	Standaard/gemeenschappelijk versus scenario	Tijdelijke DoS versus Permanent	Beoordeling
Geverifieerd	Standaard/algemeen	Permanente	Matig
Geverifieerd	Standaard/algemeen	Tijdelijke DoS met versterking	Matig
Geverifieerd	Standaard/algemeen	Tijdelijke DoS-aanval	Laag
Geverifieerd	Scenario	Permanent	Matig
Geverifieerd	Scenario	Tijdelijke DoS met versterking	Laag
Geverifieerd	Scenario	Tijdelijke DoS-aanval	Laag
Anoniem	Standaard/algemeen	permanent	Belangrijk
Anoniem	Standaard/algemeen	Tijdelijke DoS-aanval met versterking	Belangrijk
Anoniem	Standaard/algemeen	Tijdelijke Denial-of-Service (DoS)aanval	Matig
Anoniem	Scenario	Permanente	Belangrijk
Anoniem	Scenario	Tijdelijke DoS-aanval met versterking	Belangrijk

Geverifieerd versus anonieme aanval	Standaard/gemeenschappelijk versus scenario	Tijdelijke DoS versus Permanent	Beoordeling
Anoniem	Scenario	Tijdelijke DoS-aanval	Laag

Inhouds disclaimer

Tabel uitvouwen

Feedback

Is deze pagina nuttig? Yes No

Cryptografische aanbevelingen voor Microsoft SDL

18-08-2025

Gebruik deze informatie als referentie bij het ontwerpen van producten om dezelfde API's, algoritmen, protocollen en sleutellengten te gebruiken die Microsoft nodig heeft voor eigen producten en services. Veel van de inhoud is gebaseerd op de eigen interne beveiligingsstandaarden van Microsoft die worden gebruikt om de levenscyclus voor beveiligingsontwikkeling te maken.

Ontwikkelaars op niet-Windows-platforms kunnen profiteren van deze aanbevelingen. Hoewel de API- en bibliotheeknamen mogelijk verschillen, zijn de aanbevolen procedures met betrekking tot de keuze van het algoritme, de sleutellengte en de gegevensbescherming op verschillende platforms vergelijkbaar.

Aanbevelingen voor beveiligingsprotocol, algoritme en sleutellengte

TLS/SSL-versies

Producten en services moeten cryptografisch beveiligde versies van TLS/SSL gebruiken:

- TLS 1.3 moet zijn ingeschakeld
- TLS 1.2 kan worden ingeschakeld om de compatibiliteit met oudere clients te verbeteren.
- TLS 1.1, TLS 1.0, SSL 3 en SSL 2 moeten zijn uitgeschakeld

Symmetrische blokcijfers, coderingsmodi en initialisatievectoren

Coderingen blokkeren

Voor producten die symmetrische blok-coderingen gebruiken:

- Advanced Encryption Standard (AES) is vereist.
- Alle andere blokcoderingen, waaronder 3DES (Triple DES/TDEA), en RC4 moeten worden vervangen als ze worden gebruikt voor versleuteling.

Voor symmetrische blokversleutelingsalgoritmen wordt u aangeraden 256-bits sleutels te ondersteunen, maar maximaal 128 bits toe te staan. Het enige blokversleutelingsalgoritmen dat wordt aanbevolen voor nieuwe code is AES (AES-128, AES-192 en AES-256 zijn allemaal acceptabel, omdat AES-192 geen optimalisatie op sommige processors heeft).

Coderingsmodi

Symmetrische algoritmen kunnen worden uitgevoerd in verschillende modi, waarvan de meeste de versleutelingsbewerkingen aan opeenvolgende blokken met tekst zonder opmaak en codering koppelen.

Symmetrische blok-coderingen moeten worden gebruikt met een van de volgende coderingsmodi:

- Cipher Block Chaining (CBC)
- Ciphertext Stealing (CTS)
- XEX-Based Tweaked-Codebook met XTS- (Ciphertext Stealing)

Sommige andere coderingsmodi, zoals die volgen, hebben valkuilen in de implementatie waardoor ze waarschijnlijker onjuist worden gebruikt. In het bijzonder moet de werkingsmodus van het Electronic Code Book (ECB) worden vermeden. Als u dezelfde initialisatievector (IV) hergebruikt met blokcoderingen in 'modi voor streaming-coderingen', zoals CTR, kunnen versleutelde gegevens worden onthuld. Extra beveiligingsbeoordeling wordt aanbevolen als een van de onderstaande modi wordt gebruikt:

- Uitvoerfeedback (OFB)
- Cijferfeedback (CFB)
- Teller (CTR)
- Niets anders dan in de bovenstaande lijst met aanbevolen items

Initialisatievectoren (IV)

Alle symmetrische blok-coderingen moeten ook worden gebruikt met een cryptografisch sterk willekeurig getal als initialisatievector. Initialisatievectors mogen nooit een constante of prediceerbare waarde zijn. Zie Generatoren voor willekeurige getallen voor aanbevelingen voor het genereren van cryptografische sterke willekeurige getallen.

Initialisatievectors mogen nooit opnieuw worden gebruikt bij het uitvoeren van meerdere versleutelingsbewerkingen. Hergebruik kan informatie onthullen over de gegevens die worden

versleuteld, met name bij het gebruik van streaming-coderingsmodi zoals Uitvoerfeedback (OFB) of Teller (CTR).

aanbevelingen voor AES-GCM en AES-CCM

AES-GCM (Galois/tellermodus) en AES-CCM (Teller met CBC-MAC) worden veelgebruikte geverifieerde versleutelingsmodi gebruikt. Ze combineren vertrouwelijkheids- en integriteitsbescherming, waardoor ze nuttig zijn voor veilige communicatie. Hun fragiliteit ligt echter in niet-hergebruik. Wanneer dezelfde nonce (initialisatievector) tweemaal wordt gebruikt, kan dit leiden tot catastrofale gevolgen.

We raden u aan de niet-ce-richtlijnen te volgen, zoals beschreven in NIST SP 800-38D, aanbeveling voor blokcoderingsmodi van werking: Galois/tellermodus (GCM) en GMAC, waarbij speciale aandacht wordt besteed aan secties 8.1, 8.2 en 8.3 met betrekking tot de uniekheidsvereisten voor de sleutel en nonce/IV.

Een andere optie is om unieke AES-GCM/CCM-sleutels te genereren voor elk bericht dat wordt versleuteld, waardoor het maximum aantal aanroepen wordt beperkt tot 1. Deze methode wordt aanbevolen voor het versleutelen van gegevens in rusttoestand, waarbij het gebruik van een teller of het bijhouden van het maximum aantal aanroepen voor een bepaalde sleutel niet praktisch is.

Voor het versleutelen van gegevens in ruste kunt u ook overwegen om AES-CBC te gebruiken met een berichtverificatiecode (MAC) als alternatief met behulp van een Encrypt-then-MAC-schema, zodat u ervoor zorgt dat u afzonderlijke sleutels gebruikt voor versleuteling en voor de MAC.

Integriteitsverificatie

Het is een veelvoorkomende misvatting dat versleuteling standaard zowel vertrouwelijkheid als integriteitsgarantie biedt. Veel versleutelingsalgoritmen bieden geen integriteitscontrole en zijn mogelijk kwetsbaar voor manipulatieaanvallen. Er moeten extra stappen worden ondernomen om de integriteit van gegevens te waarborgen voordat ze worden verzonden en na ontvangst.

Als u geen geverifieerd versleutelingsalgoritme kunt gebruiken met gekoppelde gegevens (AEAD), zoals AES-GCM, kunt u ook de integriteit valideren met een berichtverificatiecode (MAC) met behulp van een versleutelings- en MAC-schema, zodat u afzonderlijke sleutels gebruikt voor versleuteling en voor de MAC.

Het gebruik van een afzonderlijke sleutel voor versleuteling en voor de MAC is essentieel. Als het niet mogelijk is om de twee sleutels op te slaan, is een geldig alternatief om twee sleutels af te leiden van de hoofdsleutel met behulp van een geschikte functie voor sleutelafleiden

(KDF), één voor versleutelingsdoeleinden en één voor MAC. Zie [SP 800-108 Rev. 1, Aanbeveling voor Key Derivation Using Pseudorandom Functions | CSRC (nist.gov)] (https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-38d.pdf.

Asymmetrische algoritmen, sleutellengten en opvullingsmodi

RSA

- RSA kan worden gebruikt voor sleuteltransport, sleuteluitwisseling en handtekeningen.
- RSA-versleuteling moet gebruikmaken van de OAEP- of RSA-PSS opvullingsmodi.
- Bestaande code kan PKCS #1 v1.5 opvullingsmodus gebruiken voor ondertekening.
- Het gebruik van PKCS#1 v1.5 voor versleuteling is niet toegestaan.
- Het gebruik van null-opvulling wordt niet aanbevolen.
- Een 2048-bits sleutellengte is het minimum, maar we raden u aan een 3072-bits sleutellengte te ondersteunen.

ECDSA en ECDH

- Op ECDH gebaseerde sleuteluitwisseling en op ECDSA gebaseerde handtekeningen moeten gebruikmaken van een van de drie NIST-goedgekeurde curven (P-256, P-384 of P521).
- Ondersteuning voor P-256 moet als minimum worden beschouwd, maar we raden u aan P-384 te ondersteunen.

ML-DSA

- Moet de FIPS 204-standaard gebruiken. Gebruik de versies niet in de conceptstandaard.
- Het wordt aanbevolen om ML-DSA te gebruiken in combinatie met een klassiek handtekeningalgoritmen (bijvoorbeeld ECDSA of RSA). Het gebruik van combinaties die zijn gedefinieerd door IETF (concept) of andere standaarden heeft de voorkeur voor interoperabiliteit.
- Voor ML-DSA is een geldig hybride (samengesteld) mechanisme om dezelfde gegevens te ondertekenen met zowel een klassieke als ML-DSA en beide te verifiëren (als een van beide verificaties mislukt, mislukt de verificatie).

ML-KEM

- Moet de FIPS 203-standaard gebruiken. Gebruik de versies niet in de conceptstandaard.
- Het wordt aanbevolen om een combinatie of hybride cryptosysteem te gebruiken om ML-KEM en een klassiek KEM-algoritme (dat wil bijvoorbeeld ECDH) te combineren. Het gebruik van combinaties die zijn gedefinieerd door IETF (concept) of andere standaarden heeft de voorkeur voor interoperabiliteit.

SLH-DSA

- Moet de FIPS 205-standaard gebruiken. Gebruik de versies niet in de conceptstandaard.
- Alle SLH-DSA parametersets zijn toegestaan, maar de aanbevolen parameterset is afhankelijk van de use-case.
- Geen bekend beveiligingsverschil tussen SHA-2- en SHAKE-versies (SHA-3)

LMS en XMSS

- Het is veilig om LMS of XMSS te ondersteunen voor handtekeningverificatie.
- Het wordt sterk aanbevolen om contact op te vragen met een deskundige voordat u infrastructuur implementeert/implementeert voor ondertekening en het genereren van sleutels. Er is geen zwakheid bekend, maar door mensen geïnduceerde fouten zijn zeer mogelijk en gemakkelijk te maken omdat het essentieel is om de toestand goed te beheren.

Gehele Diffie-Hellman

- Hoewel Integer Diffie-Hellman (DH) is goedgekeurd voor sleuteluitwisseling, is het niet de meest efficiënte volgens moderne standaarden. Het wordt sterk aanbevolen om in plaats daarvan ECDH te gebruiken.
- Sleutellengte >= 2048 bits wordt aanbevolen0
- De groepsparameters moeten een bekende benoemde groep zijn (bijvoorbeeld RFC 7919) of worden gegenereerd door een vertrouwde partij en geverifieerd voor gebruik.

Sleutellevensduur

- Definieer een cryptoperiod voor alle sleutels.
  - Bijvoorbeeld: Een symmetrische sleutel voor gegevensversleuteling, ook wel gegevensversleutelingssleutel of DEK genoemd, kan een gebruiksperiode van

maximaal twee jaar hebben voor het versleutelen van gegevens, ook wel bekend als de gebruiksperiode van de oorsprongsteller. U kunt definiëren dat het voor ontsleuteling een geldige gebruiksperiode heeft van drie jaar, ook wel bekend als de gebruiksperiode voor ontvangers.

- U moet een mechanisme opgeven of een proces hebben voor het vervangen van sleutels om de beperkte actieve levensduur te bereiken. Na het einde van de actieve levensduur mag een sleutel niet worden gebruikt om nieuwe gegevens te produceren (bijvoorbeeld voor versleuteling of ondertekening), maar kan nog steeds worden gebruikt om gegevens te lezen (bijvoorbeeld voor ontsleuteling of verificatie).

Generatoren voor willekeurige getallen

Alle producten en services moeten cryptografisch veilige generatoren voor willekeurige getallen gebruiken wanneer willekeurigheid vereist is.

CNG

- Gebruik BCryptGenRandom met de vlag BCRYPT_USE_SYSTEM_PREFERRED_RNG.

Win32/64

- Verouderde code kan RtlGenRandom- gebruiken in de kernelmodus.
- Nieuwe code moet BCryptGenRandom- of CryptGenRandom gebruiken.
- De C-functie Rand_s() wordt ook aanbevolen (die in Windows CryptGenRandom aanroept).
- Rand_s() is een veilige en goed presterende vervanging voor Rand().
- Rand() mag niet worden gebruikt voor cryptografische toepassingen.

.NET

- Gebruik RandomNumberGenerator.

PowerShell

- Gebruik Get-SecureRandom (PowerShell).

Windows Store-apps

- Windows Store-apps kunnen CryptographicBuffer.GenerateRandom of CryptographicBuffer.GenerateRandomNumber gebruiken.

Niet aanbevolen

- Onveilige functies met betrekking tot het genereren van willekeurige getallen zijn onder andere: rand, System.Random (.NET), GetTickCount, GetTickCount64 en Get-Random (PowerShell-cmdlet).
- Het gebruik van het algoritme voor willekeurige getalgenerator met dubbele elliptische curven ('DUAL_E_DRBG') is niet toegestaan.

Door het Windows-platform ondersteunde cryptobibliotheken

Op het Windows-platform raadt Microsoft aan de crypto-API's te gebruiken die zijn ingebouwd in het besturingssysteem. Op andere platforms kunnen ontwikkelaars ervoor kiezen om cryptobibliotheken zonder platform te evalueren voor gebruik. Over het algemeen worden cryptobibliotheken van het platform vaker bijgewerkt omdat ze worden verzonden als onderdeel van een besturingssysteem in plaats van te worden gebundeld met een toepassing.

Elke gebruiksbeslissing met betrekking tot platform versus niet-platform crypto moet worden geleid door de volgende vereisten:

- De bibliotheek moet een ondersteunde versie zijn die vrij is van bekende veiligheidskwetsbaarheden.
- De meest recente beveiligingsprotocollen, algoritmen en sleutellengten moeten worden ondersteund.
- (Optioneel) De bibliotheek moet alleen oudere beveiligingsprotocollen/algoritmen kunnen ondersteunen voor compatibiliteit met eerdere versies.

Systeemeigen code

- Crypto Primitives: Als uw release zich in Windows bevindt, gebruikt u CNG, indien mogelijk.
- Verificatie van codehandtekening: WinVerifyTrust is de ondersteunde API voor het verifiëren van codehandtekeningen op Windows-platforms.

- Certificaatvalidatie (zoals gebruikt in validatie van beperkte certificaten voor ondertekening van code of TLS/DTLS): CAPI2-API; Bijvoorbeeld CertGetCertificateChain en CertVerifyCertificateChainPolicy.

Belangrijke afleidingsfuncties

Sleutelontduivatie is het proces van het afleiden van cryptografisch sleutelmateriaal uit een gedeeld geheim of een bestaande cryptografische sleutel. Producten moeten aanbevolen belangrijke afleidingsfuncties gebruiken. Het afleiden van sleutels van door de gebruiker gekozen wachtwoorden of hash-wachtwoorden voor opslag in een verificatiesysteem is een speciaal geval dat niet wordt gedekt door deze richtlijnen; ontwikkelaars moeten een expert raadplegen.

De volgende standaarden geven KDF-functies op die worden aanbevolen voor gebruik:

- NIST SP 800-108 (Revisie 1): Aanbeveling voor sleutelafleiding door pseudowillekeurige functies. Met name de KDF in tegenmodus, met HMAC als een pseudowillekeurige functie
- NIST SP 800-56A (revisie 3): aanbeveling voor Pair-Wise sleuteluitwisselingsschema's met discrete logaritmecryptografie.

Als u sleutels wilt afleiden van bestaande sleutels, gebruikt u de BCryptKeyDerivation -API met een van de algoritmen:

- BCRYPT_SP800108_CTR_HMAC_ALGORITHM
- BCRYPT_SP80056A_CONCAT_ALGORITHM

Als u sleutels wilt afleiden van een gedeeld geheim (de uitvoer van een sleutelovereenkomst), gebruikt u de BCryptDeriveKey -API met een van de volgende algoritmen:

- BCRYPT_KDF_SP80056A_CONCAT
- BCRYPT_KDF_HMAC

Certificaatvalidatie

Producten die gebruikmaken van TLS of DTLS moeten de X.509-certificaten van de entiteiten waarmee ze verbinding maken, volledig verifiëren. Dit proces omvat verificatie van de volgende onderdelen van het certificaat:

- Domeinnaam.
- Geldigheidsdatums (zowel begin- als vervaldatums).

- Intrekkingsstatus.
- Gebruik (bijvoorbeeld 'Serververificatie' voor servers, 'Clientverificatie' voor clients).
- Vertrouwensketen. Certificaten moeten worden gekoppeld aan een basiscertificeringsinstantie (CA) die wordt vertrouwd door het platform of expliciet is geconfigureerd door de beheerder.

Als een van deze verificatietests mislukt, moet het product de verbinding met de entiteit beëindigen.

Gebruik geen zelfondertekende certificaten. Zelfondertekende certificaten wekken niet inherent vertrouwen, ondersteunen intrekking, of het vernieuwen van sleutels.

Cryptografische hashfuncties

Producten moeten gebruikmaken van de SHA-2-serie hash-algoritmen (SHA-256, SHA-384 en SHA-512). Het afkappen van cryptografische hashes voor beveiligingsdoeleinden tot minder dan 128 bits is niet toegestaan. Hoewel het gebruik van SHA-256 het minimum is, raden we u aan SHA-384 te ondersteunen.

MAC/HMAC/keyed hash-algoritmen

Een berichtverificatiecode (MAC) is een stukje informatie dat is gekoppeld aan een bericht waarmee de ontvanger zowel de echtheid van de afzender als de integriteit van het bericht kan verifiëren met behulp van een geheime sleutel.

Het gebruik van een op hash-gebaseerde MAC (↗) of een op blokcodering-gebaseerde MAC (↗) wordt aanbevolen, zolang alle onderliggende hash- of symmetrische versleutelingsalgoritmen ook worden aanbevolen voor gebruik; momenteel omvat dit de HMAC-SHA2-functies (HMAC-SHA256, HMAC-SHA384 en HMAC-SHA512). Hoewel het gebruik van HMAC-SHA256 het minimum is, raden we u aan HMAC-SHA384 te ondersteunen.

Afkapping van HMAC's tot minder dan 128 bits wordt niet aanbevolen.

Ontwerp- en operationele overwegingen

- U moet zo nodig een mechanisme opgeven voor het vervangen van cryptografische sleutels. Sleutels moeten worden vervangen zodra ze het einde van hun actieve levensduur hebben bereikt of als de cryptografische sleutel is aangetast.
  - Wanneer u een certificaat verlengt, moet u het vernieuwen met een nieuwe sleutel (opnieuw versleutelen).

- Producten die cryptografische algoritmen gebruiken om gegevens te beveiligen, moeten voldoende metagegevens bevatten, samen met die inhoud ter ondersteuning van migratie naar verschillende algoritmen in de toekomst. Deze metagegevens moeten het gebruikte algoritme, sleutelgrootten en opvullingsmodi bevatten.
  - Zie het artikel Cryptografische flexibiliteit voor meer informatie over cryptografische flexibiliteit.

- Waar beschikbaar moeten producten gebruikmaken van gevestigde, door het platform geleverde cryptografische protocollen in plaats van ze opnieuw te implementeren, inclusief ondertekeningsindelingen (bijvoorbeeld een standaard, bestaande indeling).

- Rapporteer geen cryptografische bewerkingsfouten aan eindgebruikers. Wanneer u een fout retourneert naar een externe beller (bijvoorbeeld een webclient of client in een clientserverscenario), gebruikt u alleen een algemeen foutbericht.
  - Vermijd onnodige informatie, zoals het rechtstreeks rapporteren van fouten buiten het bereik of ongeldige lengtefouten. Alleen gedetailleerde fouten op de server vastleggen, en dat alleen doen als gedetailleerde logboekregistratie is ingeschakeld.

- Extra beveiligingsbeoordeling wordt ten zeerste aanbevolen voor elk ontwerp met de volgende items:

  - Een nieuw protocol dat voornamelijk gericht is op beveiliging (zoals een verificatie- of autorisatieprotocol)

  - Een nieuw protocol dat cryptografie op een nieuwe of niet-standaard manier gebruikt. Voorbeelden van overwegingen zijn:

    - Worden er als onderdeel van de protocolimplementatie cryptografische API's of methoden aangeroepen door een product dat het protocol implementeert?

    - Is het protocol afhankelijk van een ander protocol dat wordt gebruikt voor verificatie of autorisatie?

    - Definieert het protocol opslagindelingen voor cryptografische elementen, zoals sleutels?

- Zelfondertekende certificaten worden niet aanbevolen. Het gebruik van een zelfondertekend certificaat, zoals het gebruik van een onbewerkte cryptografische sleutel, biedt gebruikers of beheerders geen basis voor het nemen van een vertrouwensbeslissing.
  - Het gebruik van een certificaat dat is geroot in een vertrouwde certificeringsinstantie maakt daarentegen duidelijk de basis voor het vertrouwen op de bijbehorende persoonlijke sleutel en maakt intrekking en updates mogelijk als er sprake is van een beveiligingsfout.

Bedreigingsmodellering integreren met DevOps

Artikel • 20-10-2023

Dit bericht is geschreven door Britna Curzi, Anthony Nevico, Jonathan Davis, Raphael Pazos Rodriguez en Ben Hanson

Introductie

Threat modeling is een belangrijke beveiligingsmethode die helpt bij het identificeren en prioriteren van de belangrijkste risicobeperking voor een toepassing of systeem. Dit document bevat enkele reflecties over hoe het mogelijk is om bedreigingsmodellering effectiever en efficiënter te gebruiken, deze te integreren met moderne DevOps-methodologieën en -hulpprogramma's, en zich te concentreren op de waarde die is verstrekt aan alle verschillende actoren die betrokken zijn bij de levenscyclus van softwareontwikkeling.

Is dit papier voor jou?

Dit document is het resultaat van het werk van een klein team beveiligings- en bedreigingsmodelleringsexperts van Microsoft en bevat invoer en ideeën van enkele van de meest prominente experts van buiten Microsoft. Er wordt geprobeerd een eenvoudige maar dringende vraag aan te pakken: wat moeten we doen om ervoor te zorgen dat het bedreigingsmodelleringsproces dat we gebruiken, wordt bijgewerkt naar de moderne vereisten die worden opgelegd door Agile-methodologieën en DevOps, zodat we de vereiste waarde tegen de laagste kosten bieden?

Als u een producteigenaar bent, het lid van een beveiligingsteam of gewoon een ontwikkelaar die overweegt om bedreigingsmodellering te gebruiken als onderdeel van uw ontwikkelingslevenscyclus, is dit document voor u.

Als u al bedreigingsmodellering hebt gebruikt, kunt u nog steeds praktische ideeën vinden om uw proces te verbeteren.

Toch is het document ontworpen om ideeën te introduceren om de huidige processen te verbeteren of om bedreigingsmodellering in te voeren als onderdeel van uw DevOps-proces. Het introduceert geen specifieke hulpprogramma's of producten, zelfs als het onze hoop is om deze ideeën te zien die in de toekomst door sommige

hulpprogramma's of producten zijn geïmplementeerd. Daarom vindt u hier geen aankondigingen van nieuwe hulpprogramma's of previews van toekomstige functies.

Waarom is threat modeling belangrijk?

Threat Modeling is een van de belangrijkste benaderingen voor het veilig ontwerpen van softwareoplossingen. Met threat Modeling analyseert u een systeem aanvalsvectoren en ontwikkelt u acties voor het beperken van risico's die door deze aanvallen worden veroorzaakt. Op de juiste wijze is threat modeling een uitstekend onderdeel van elk risicobeheerproces. Het kan ook helpen om kosten te verlagen door ontwerpproblemen vroeg te identificeren en op te lossen. Een oud onderzoek van NIST schatte de kosten voor het oplossen van een ontwerpprobleem in productiecode ongeveer 40 keer hoger dan het herstellen ervan tijdens de ontwerpfase. Het bespaart ook kosten als gevolg van beveiligingsincidenten voor de uiteindelijke ontwerpproblemen. Houd er rekening mee dat in het rapport kosten van gegevenslekken van IBM Security en het Ponemon Instituut de gemiddelde kosten van een gegevenslek worden geschat op $ 4,35 miljoen. Voor de zogenaamde Mega-schendingen, waarbij het compromis van meer dan 50 miljoen records is betrokken, bereikt de gemiddelde kosten $ 387M!

Threat modeling is de eerste activiteit die u kunt uitvoeren om uw oplossing te beveiligen, omdat deze werkt in het oplossingsontwerp. Dit kenmerk maakt het de meest effectieve beveiligingspraktijk die u op uw SDLC kunt toepassen.

Microsoft heeft een lange geschiedenis met threat modeling. In 1999 schreef twee (toen) Microsoft-werknemers, Loren Kohnfelder en Praerit Garg, een document, De bedreigingen voor onze producten. In dit document is de STRIDE-benadering geïntroduceerd, een synoniem voor het Microsoft Threat Modeling-proces.

Bedreigingsmodellering is een evolutief proces

Threat modeling is geen statisch proces; het ontwikkelt zich naarmate de behoeften veranderen en technologieën veranderen.

- Supply Chain-aanvallen zoals de recente aanval op SolarWinds laten zien dat er meer scenario's nodig zijn voor Threat Modeling dan de oplossing zelf, inclusief ontwikkeling en implementatie.
- Open Source-beveiligingsproblemen zoals de recente voor Log4j hebben aangetoond dat de huidige aanpak moet worden aangevuld op basis van de acceptatie van hulpprogramma's voor softwaresamenstellingsanalyse om te

scannen op kwetsbare onderdelen door de oplossing defensief te ontwerpen om de blootstelling ervan te beperken.

- De toepassing van nieuwe technologieën zoals Machine Learning introduceert nieuwe aanvalsvectoren die moeten worden begrepen en gecontroleerd. Denk bijvoorbeeld aan de mogelijkheid om kwaadwillende geluiden door menselijke oren te spelen om de uitvoering van opdrachten door AI-services te veroorzaken, zoals besproken in https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/carlini.

Bij Microsoft oefenen verschillende productgroepen verschillende varianten van threat modeling uit op basis van hun specifieke beveiligingsvereisten. Elke variant is erop gericht een adequaat beveiligingsniveau te garanderen voor de scenario's waarop deze wordt toegepast, maar wat 'voldoende' betekent dat wijzigingen worden aangebracht, afhankelijk van de specifieke context.

Het beveiligen van Windows verschilt bijvoorbeeld van het beveiligen van Azure Cognitive Services, omdat deze systemen zeer verschillende grootten en kenmerken hebben. Een belangrijk aspect van threat modeling is het verdelen van de kosten ten opzichte van de risicotolerantie voor een toepassing. Hoewel dit kan leiden tot de beslissing om bedreigingsmodellering helemaal te voorkomen voor sommige scenario's, is het zo effectief wanneer dit op de juiste manier wordt gedaan, dat we het alleen kunnen aanbevelen voor elk IT-initiatief, inclusief projecten voor softwareontwikkeling en infrastructuurimplementatie.

Het belang van het focussen op de ROI

De afgelopen jaren is er een constante toename in het belang van threat modeling gezien als een belangrijk softwareontwikkelingsproces. Dit belang is te wijten aan de exponentiële toename van aanvallen op infrastructuren en oplossingen. Initiatieven zoals de aanbevolen minimumstandaard van NIST voor de verificatie van code voor leveranciers of ontwikkelaars en het manifest voor bedreigingsmodellen hebben de vraag verder verhoogd tot het punt dat de huidige benaderingen enkele limieten hebben getoond. De resultaten van bedreigingsmodellering zijn bijvoorbeeld sterk afhankelijk van het aangenomen proces en van wie het bedreigingsmodel uitvoert. Er is dus een zorg om consistent hogere kwaliteit uit de ervaring te halen.

Maar wat betekent kwaliteit voor bedreigingsmodellering? Voor ons moet een kwaliteitsrisicomodel de volgende kenmerken hebben:

- Het moet bruikbare oplossingen identificeren, activiteiten die u kunt doen om de potentiële verliezen te verminderen die het gevolg zijn van aanvallen. Uitvoerbaar

betekent dat deze oplossingen goed moeten zijn gedefinieerd, wat betekent dat u voldoende informatie krijgt om ze te implementeren en vervolgens de implementatie te testen. Dit betekent ook dat ze moeten worden verstrekt om eenvoudig verbruik van het ontwikkelingsteam mogelijk te maken. Met DevOps en Agile betekent dit dat er een eenvoudig pad is om de oplossingen in de achterstand te importeren.

- Voor elke beperking moet deze de status identificeren. Sommige oplossingen zijn nieuw, terwijl andere al bestaan. Het bedreigingsmodel moet herkennen wat er al is en zich richten op het huidige risico om te bepalen hoe de situatie moet worden verbeterd.

- Het moet duidelijk aangeven waarom elke beperking is vereist door deze te koppelen aan de respectieve bedreigingen.

- Bovendien hebben risicobeperkende oplossingen een relatieve sterkte voor elke bedreiging. TLS-versleuteling kan bijvoorbeeld een sterke beperking zijn voor het risico dat gegevens tijdens overdracht openbaar worden gemaakt, en tegelijkertijd kan het een bijna volledige beperking zijn voor het risico dat de server is vervalst.

- De bedreigingen moeten geloofwaardig, goed gedefinieerd en specifiek zijn voor de oplossing.

- De bedreigingen moeten een bijbehorende ernst hebben, die rekening moet houden met zowel de waarschijnlijkheid als de impact. De ernst moet redelijk en idealiter onbevoordeeld zijn.

- Het moet mogelijk zijn om een uitgebreid overzicht te krijgen van de risico's en hoe deze kunnen worden aangepakt. Deze weergave zou nuttig zijn bij het stimuleren van zinvolle gesprekken met het beveiligingsteam en met besluitvormers voor bedrijven, en het zou ons in staat stellen om de onnodige complexiteiten te verbergen.

Deze lijst bevat al een belangrijk concept: threat modeling kan waarde bieden aan veel rollen die betrokken zijn tijdens de levenscyclus van de software, maar elke rol heeft verschillende behoeften en vereisten. Ontwikkelaars moeten bijvoorbeeld duidelijke informatie ontvangen over wat ze moeten implementeren en hoe ze kunnen controleren of wat ze hebben geïmplementeerd zich gedragen zoals verwacht. Aan de andere kant houdt het beveiligingsteam zich doorgaans bezig met de algehele beveiliging van het ecosysteem van infrastructuur en toepassingen die eigendom zijn van de organisatie; daarom moeten ze informatie ontvangen waarmee kan worden bepaald of het systeem binnen het bereik veilig genoeg is en voldoet aan de nalevingsvereisten. Ten slotte

moeten producteigenaren en besluitvormers begrijpen wat nodig is om het risico aanvaardbaar te maken voor de organisatie.

Dergelijke verschillende behoeften moeten verschillende weergaven bieden voor het bedreigingsmodel, die elk zijn gericht op een specifiek gebruiksscenario.

Een typisch probleem met bedreigingsmodellering is dat hoe meer het succesvol is, hoe moeilijker het is voor de weinige beschikbare experts om de vraag te dekken en tegelijkertijd de hoge kwaliteit te bieden die op basis van deze ervaring wordt verwacht. Als gevolg hiervan kan de kwaliteit in sommige gevallen negatief worden beïnvloed. Alles is goed totdat threat modeling stopt met het leveren van een aanzienlijke waarde in vergelijking met de investering. Meer dan een paar organisaties worden beïnvloed door dit probleem. Er zijn al een aantal rapporten van zakelijke besluitvormers begonnen met het vragen over bedreigingsmodellering, omdat het geen aanzienlijke waarde zou opleveren voor de kosten.

Met waarde verwijzen we naar de bedrijfswaarde. Dit is de mogelijkheid om de informatie te verstrekken die nodig is om inzicht te krijgen in de risico's die het systeem binnen het bereik vertegenwoordigt en een zinvol beslissingsproces aan te sturen voor het selecteren van de juiste oplossingen die moeten worden geïmplementeerd. Bovendien is de waarde ook gerelateerd aan het verstrekken van de juiste informatie aan de ontwikkelaars en de testers. Ten slotte is de waarde gerelateerd aan de communicatie van het restrisico met alle betrokken partijen. We kunnen bijvoorbeeld de waarde meten door de impact van het threat modeling-proces te meten. Stel dat we het totale risico voor de oplossing meten door een getal toe te wijzen aan de ernst die aan elke bedreiging is geïdentificeerd. In dat geval verwachten we dat het totale risico in de loop van de tijd afneemt per effect van het bedreigingsmodel. Als het totale risico constant blijft of toeneemt, kunnen we een probleem hebben. Hoe steiler de afname, hoe hoger de impact van het bedreigingsmodel. Natuurlijk zou het bedreigingsmodel geen controle hebben over de geïmplementeerde oplossingen. Het is de verantwoordelijkheid van de producteigenaar om te bepalen wat er moet worden geïmplementeerd. Maar het voordeel van het koppelen van de effectiviteit van het bedreigingsmodel aan de daadwerkelijke implementatie van de risico's is dat het de impact op de werkelijke beveiliging van de oplossing verhoogt, waardoor het risico dat het bedreigingsmodel een theoretische oefening blijft.

In plaats daarvan zijn de kosten gerelateerd aan de activiteiten die nodig zijn om het bedreigingsmodel zelf uit te voeren. Dit is de tijd die alle betrokken partijen nodig hebben om het bedreigingsmodel te produceren en te bespreken.

Dit stelt de vraag: kunnen we een threat modeling-proces definiëren dat is gericht op het maximaliseren van de bedrijfswaarde en het minimaliseren van de kosten?

Het belang van DevOps

We hebben al uitgelicht hoe belangrijk het is om ervoor te zorgen dat threat modeling een waardevolle praktijk is die is geïntegreerd met het DevOps-proces. Dit betekent dat het proces beschikbaar moet zijn voor alle teamleden, meestal door het te vereenvoudigen en te automatiseren. Het belangrijkste is dat we ons richten op threat modeling voor DevOps, dat we ervoor moeten zorgen dat de ervaring diep is geïntegreerd met de bestaande DevOps-processen.

Bedreigingsmodellering mag niet nog een andere last worden, maar in plaatsdaarvan moet het een asset zijn om de beveiligingsvereisten te vereenvoudigen, het ontwerp van veilige oplossingen, het opnemen van activiteiten in het & hulpprogramma voor het bijhouden van taakfouten en de evaluatie van het resterende risico gezien de huidige en toekomstige status van de oplossing.

Uitlijning met DevOps

We kunnen verschillende technieken gebruiken om bedreigingsmodellering af te stemmen op de huidige DevOps-praktijk.

Bedreigingen en oplossingen

Eerst moeten we ons richten op het threat modeling-proces op wat er moet gebeuren. Bedreigingen, die de aanvalspatronen zijn en hoe ze kunnen optreden, zijn nodig om uit te leggen waarom het team een beveiligingscontrole moet implementeren. Ze zijn ook een factor bij het bepalen wanneer oplossingen moeten worden geïmplementeerd. Toch is het echte doel om te bepalen wat er moet worden gedaan: de oplossingen. Daarom moet de aanpak leiden tot een snelle identificatie van de vereiste oplossingen en moet het besluitvormingsproces worden geïnformeerd, zodat het gemakkelijker is om te bepalen wat er moet worden uitgevoerd en wanneer. Het belangrijkste product van dit beslissingsproces is het hebben van de geselecteerde oplossingen in de achterstand om ze deel te laten uitmaken van het standaardproces. In het ideale geval moeten het hulpprogramma voor bedreigingsmodellering en het hulpprogramma voor het bijhouden van taakfouten & worden gesynchroniseerd met de updates voor de beperkingsstatus in het bedreigingsmodel. Hierdoor zou het restrisico dynamisch en automatisch kunnen worden herzien, wat essentieel is voor de ondersteuning van geïnformeerde beslissingen als onderdeel van de gebruikelijke choreografieën van de aangenomen Agile-methodologie, zoals de sprintplanningsvergadering.

> **Wat kun je vandaag doen?**

> Als expert op het gebied van bedreigingsmodellering moet u ervoor zorgen dat u een proces voor bedreigingsmodellering implementeert dat acties duidelijk kan identificeren en opnemen in het & bijhouden van taakfouten naar keuze. Een manier is om een van de vele hulpprogramma's voor threat modeling te gebruiken om dit proces te automatiseren.
>
> Als ontwikkelaar moet u zich richten op de beveiligingscontroles die als nodig zijn geïdentificeerd. Het proces moet zo zijn ontworpen dat u ze op dezelfde manier ontvangt als u verwacht andere activiteiten te ontvangen.

Functies, gebruikersverhalen en taken

We hebben al aangegeven dat de oplossingen het belangrijkste artefact vertegenwoordigen dat is geproduceerd door het bedreigingsmodel met betrekking tot DevOps-integratie. Daarom is het belangrijk om het type objecten dat is gemaakt op basis van deze oplossingen duidelijk te definiëren op het hulpprogramma Voor & het bijhouden van taakfouten naar keuze. Sommige oplossingen kunnen meer duren dan een Sprint. Daarom is het misschien het beste om ze als functies te maken. Maar veel zijn eenvoudiger en kunnen in één Sprint worden geïmplementeerd; Het zou dus mogelijk zijn om ze te vertegenwoordigen als gebruikersverhalen of -taken. Hoewel het genereren van verschillende typen werkitems mogelijk is, kan dit leiden tot een gecompliceerd proces dat tot fouten en verwarring kan leiden. Daarom lijkt het praktischer om één type werkitem te gebruiken. Gezien het feit dat risicobeperking kan worden beschouwd als kinderen van gebruikersverhalen, kunt u overwegen deze te vertegenwoordigen als taken, wat inhoudt dat de vereiste voor het uitvoeren van het genoemde werkitemtype in één Sprint wordt versoepeld.

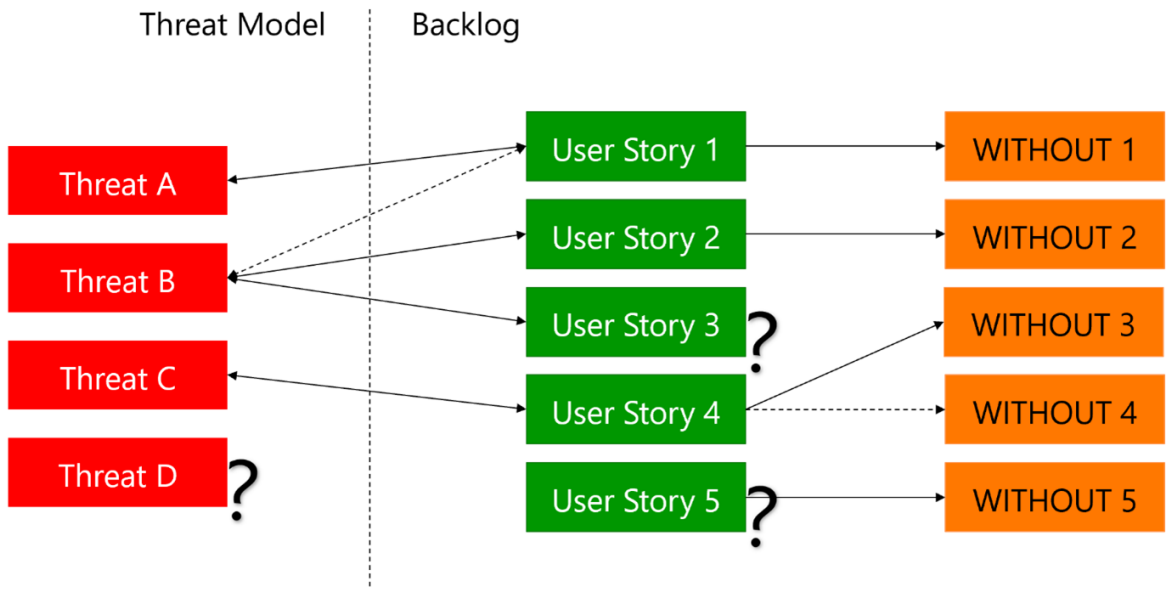
> **Wat kun je vandaag doen?**
>
> Zorg ervoor dat oplossingen die worden geïdentificeerd door het bedreigingsmodel, worden weergegeven in de achterstand. Identificeer een manier om ze duidelijk weer te geven.

Gebruikersverhalen

De oplossingen zijn niet de enige artefacten die deel uitmaken van een bedreigingsmodel. Dit kan en moet worden afgestemd op wat u hebt in het hulpprogramma Voor & het bijhouden van taakfouten. U kunt bijvoorbeeld ook bedreigingen voorstellen. Dit doel kan worden bereikt door de gebruikersverhalen uit te breiden door de toevoeging van een ZONDER-component aan de gebruikelijke 'Als een [wie ben ik] ik wil [wat ik wil] zodat ik [iets kan doen]'. Bijvoorbeeld: "Als gebruiker wil ik

betalen met mijn creditcard, zodat ik bepaalde goederen kan kopen, ZONDER dat mijn creditcardgegevens worden gestolen". De WITHOUT-component kan worden toegewezen aan een of meer bedreigingen en kan soms beveiligingsvereisten uitdrukken. Door ervoor te zorgen dat deze afstemming tussen bedreigingen en ZONDER-componenten expliciet wordt gemaakt binnen het bedreigingsmodel, kunnen we ervoor zorgen dat mogelijke risico's door het team worden weerspiegeld en verzorgd, omdat ze worden opgenomen als onderdeel van de gebruikersverhalen. U kunt deze relatie ook gebruiken om elke beveiligingsvereiste die is geïdentificeerd als onderdeel van de gebruikersverhalen toe te wijzen aan ten minste een bedreiging.

> **Leuk om te weten**
>
> De WITHOUT-component is geen origineel idee van het team dat deze pagina heeft geproduceerd. We weten niet zeker wie het voor het eerst heeft geïntroduceerd, maar we zijn dankbaar voor wie dit idee heeft gekregen.



Afbeelding 1: Vereisten uitlijnen

In de vorige afbeelding ziet u bijvoorbeeld de volgende situaties:

- Threat A is gekoppeld aan User Story 1 via component WITHOUT 1.
- Threat B is gekoppeld aan User Story 2 via component WITHOUT 2.
- Bedreiging B is ook gekoppeld aan User Story 3. Maar User Story 3 is niet toegewezen aan een WITHOUT-component. Waarom? Het vertegenwoordigt een mogelijke anomalie die u moet onderzoeken.
- Bedreiging B is ook gekoppeld aan User Story 1. Het is nog niet duidelijk of we gebruikersverhalen moeten toestaan die zijn gekoppeld aan meer dan één

bedreiging.

- Threat C is gekoppeld aan User Story 4, dat is gekoppeld aan WITHOUT 3 en 4. Het is nog niet duidelijk of we meer dan één ZONDER component mogen hebben.
- Threat D is niet gekoppeld aan een gebruikersverhaal. Ontbreken we een gebruikersverhaal of een WITHOUT-component?
- User Story 5 is gekoppeld aan een WITHOUT-component, maar heeft geen bijbehorende bedreiging. Missen we een bedreiging of gewoon een koppeling tussen een gebruikersverhaal en een bedreiging?

We identificeren zelden beveiligingsvereisten als onderdeel van het bedreigingsmodel. Daarom introduceert de WITHOUT-component de mogelijkheid om de ervaring verder te integreren door de bedreigingsmodellen uit te breiden met beveiligingsvereisten en deze te koppelen aan de gerelateerde gebruikersverhalen. Deze benadering speelt een belangrijke rol bij het ontwikkelen van de bedreigingsmodelleringservaring van een evaluatie die na verloop van tijd wordt herhaald om het hulpprogramma voor beveiligingsontwerp voor DevOps te worden.

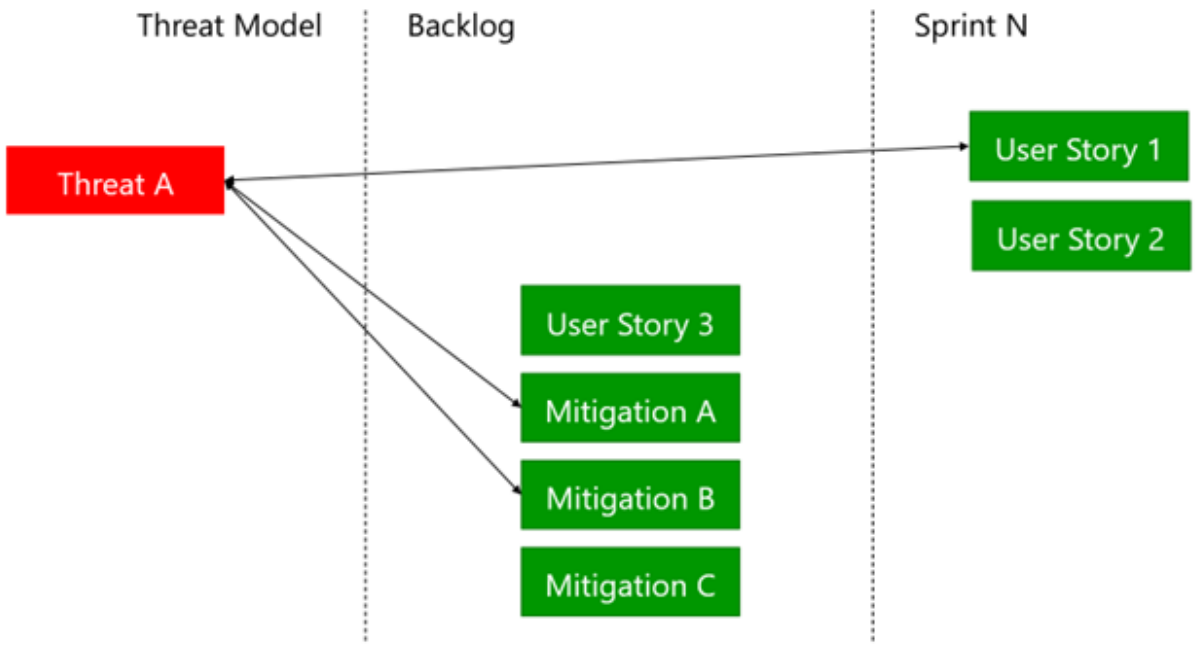
> **Wat kun je vandaag doen?**
>
> Begin met het gebruik van de WITHOUT-component in uw gebruikersverhalen.
>
> Wijs de bedreigingen die u identificeert toe aan gebruikersverhalen met WITHOUT-componenten en vice versa.

Een geïntegreerde ervaring

U kunt hetzelfde idee toepassen op andere scenario's. Het bedreigingsmodel kan bijvoorbeeld de beveiligingsvereisten koppelen aan artefacten in het bedreigingsmodel zelf, zoals bedreigingen en oplossingen, en die in het & hulpprogramma Traceren van fouten bijhouden. De vereiste voor het implementeren van bewaking voor het identificeren van aanvallen die worden uitgevoerd, moet bijvoorbeeld worden toegewezen aan al deze oplossingen die betrekking hebben op bewaking en vervolgens op de bijbehorende artefacten in het hulpprogramma Taakfouttracking & . Als gevolg hiervan zou het gemakkelijk zijn om situaties te identificeren waarbij een beveiligingsvereiste niet wordt gerealiseerd: in feite zou deze niet aan iets worden gekoppeld.

U kunt dezelfde koppelingen gebruiken tussen de artefacten in het hulpprogramma Voor & het bijhouden van taakfouten en de bedreigingen en oplossingen die zijn geïdentificeerd door het bedreigingsmodel om de prioriteit van de

beveiligingsactiviteiten te vergemakkelijken. Beveiliging wordt meestal als laatste geïmplementeerd, soms om reactieve beveiligingsproblemen aan te pakken die worden geïdentificeerd door een hulpprogramma of een penetratietest. Integendeel, het zou het meest effectief zijn om de oplossingen samen met de gerelateerde gebruikersverhalen of -functies te implementeren. Waarom moet u wachten met het implementeren van de besturingselementen om de creditcardgegevens te beveiligen wanneer u deze samen met de gerelateerde betalingsfuncties moet implementeren? Het bedreigingsmodel moet deze relaties markeren en een eenvoudige manier bieden om te bepalen wanneer een bepaalde functie tijdens een Sprint wordt geïmplementeerd, vereist de implementatie van een gerelateerde beveiligingsfunctie. Deze informatie kan bijvoorbeeld worden gebruikt tijdens de sprintplanningsvergadering om een zinvolle discussie te voeren en een geïnformeerde prioriteitsaanduiding te stimuleren. Het mechanisme is eenvoudig. Stel dat de producteigenaar voor een project waaraan we werken besluit om een gebruikersverhaal voor de volgende Sprint te plannen. Het genoemde gebruikersverhaal heeft een WITHOUT-component die is gekoppeld aan een bedreiging. Het bedreigingsmodel identificeert verschillende oplossingen voor dezelfde bedreiging. Daarom kunnen we onmiddellijk afleiden dat we prioriteit moeten geven aan een of meer van de geïdentificeerde oplossingen.



Afbeelding 2: Prioriteit geven aan beveiliging

In de bovenstaande afbeelding zien we bijvoorbeeld dat User Story 1 is gekoppeld aan bedreiging 1, die op zijn beurt is gekoppeld aan risicobeperking A en B. Daarom moeten we ook overwegen om een of beide van deze oplossingen te implementeren.

> **Wat kun je vandaag doen?**

> Koppel gebruikersverhalen met WITHOUT-componenten aan de werkitems die overeenkomen met de geselecteerde oplossingen met behulp van het bedreigingsmodel als referentie. Zorg er bij het plannen van de volgende Sprint voor dat u prioriteit geeft aan de gekoppelde beveiligingsactiviteiten wanneer u een van deze gebruikersverhalen implementeert met WITHOUT-componenten.

Integratie om onjuiste uitlijningen te markeren

Zodra we beginnen na te denken over hoe we de artefacten die het bedreigingsmodel opstellen, kunnen koppelen aan de artefacten in het hulpprogramma Voor & het bijhouden van taakfouten, wordt het gemakkelijker om mogelijkheden te identificeren voor het verbeteren van de kwaliteit van beide. De sleutel is om hun relaties te gebruiken om verschillen te markeren en de informatie te gebruiken die aanwezig is in één om de aanwezige informatie aan te vullen, te integreren en te interpreteren wat er in de andere aanwezig is. Zoals hierboven is besproken, kunt u dit doen zonder dat dit aanzienlijk van invloed is op de werking van het team. Dat komt doordat de benadering afhankelijk is van bestaande informatie en relaties creëert tussen de verschillende objecten in de verschillende werelden. Daarom zou het bedreigingsmodel de bron van waarheid worden voor de beveiliging van de oplossing. Tegelijkertijd wordt de achterstand continu afgestemd op de status van de oplossing.

> **Wat kun je vandaag doen?**
>
> Controleer regelmatig of er geen niet-toegewezen bedreigingen of gebruikersverhalen zijn met WITHOUT-componenten.

Bedreigingsmodellering en de bewerkingen

Al deze ideeën zijn voornamelijk gericht op de ontwikkelingszijde van DevOps. Kunnen we ook iets doen om bewerkingen te verbeteren? Dat denken we wel. Het zou bijvoorbeeld mogelijk zijn om threat modeling te gebruiken als een hulpprogramma om hoofdoorzaakanalyse mogelijk te maken, omdat het een uitgebreid overzicht van het systeem biedt vanuit een beveiligingsperspectief en zo een beter inzicht kan krijgen in de gevolgen van sommige aanvallen. Hiervoor zou het nodig zijn om het bedreigingsmodel te integreren met de bestaande feeds van de gekozen bewakingshulpprogramma's. Deze benadering kan een aanvulling vormen op de gekozen SIEM.

Een ander idee voor het integreren van bedreigingsmodellering met Operations is het gebruik van de eerste om het ontwerp te bepalen van hoe dit laatste kan gebeuren. Een voorbeeld hiervan is het ontwerp van gebeurtenissen voor de oplossing.

Bedreigingsmodellering identificeert mogelijke aanvallen en we kunnen die kennis gebruiken om gebeurtenissen te identificeren die de oplossing binnen het bereik kan veroorzaken wanneer deze aanvallen mislukken. Als u strikte invoervalidatie uitvoert, heeft een kwaadwillende aanvaller een paar pogingen nodig voordat deze slaagt. In eerste instantie mislukken de pogingen en slaagt een van deze pogingen uiteindelijk. Door gebeurtenissen voor elke fout op te halen en waarschuwingen te activeren wanneer een bepaalde drempelwaarde is bereikt, kunt u mogelijk aanvallen detecteren en acties ondernemen om deze te herstellen. Deze situaties worden zelden gedetecteerd als u zich beperkt tot het bewaken van de infrastructuur. Daarom is het noodzakelijk om aangepaste gebeurtenissen op te nemen die het team moet ontwerpen en implementeren voordat de SOC deze kan gebruiken.

Bovendien weet de laatste mogelijk niet veel over de oplossing. Daarom kan de SOC mogelijk niet bepalen hoe moet worden gereageerd wanneer de invoervalidatie mislukt. Wanneer een gegevenslek optreedt, is het helaas noodzakelijk om snel te reageren om de directe schade en de waarschijnlijkheid en entiteit van uiteindelijke boetes te verminderen.

Daarom moeten we van tevoren plannen wat er moet worden bewaakt, onder welke omstandigheden we mogelijk een probleem hebben en wat er moet gebeuren als dat gebeurt. De beste manier om deze gebeurtenissen te identificeren, is om te vertrouwen op een bedreigingsmodel. Daarom zou het nuttig zijn om het te gebruiken om gestandaardiseerde artefacten te genereren om de implementatie van de benodigde configuraties te begeleiden en te versnellen om bewaking en controle te stimuleren en incidentrespons te vergemakkelijken.

> **Wat kun je vandaag doen?**
>
> Gebruik actief bedreigingsmodel om gebeurtenissen te identificeren die u voor elke bedreiging kunt genereren. Deze gebeurtenissen kunnen worden geleverd door de infrastructuur of iets dat door de toepassing moet worden gegenereerd. Neem werkitems op in uw achterstand om ervoor te zorgen dat deze gebeurtenissen worden geïmplementeerd.
>
> Werk actief samen met uw operations- en beveiligingsteams, waaronder het SOC-team, om ervoor te zorgen dat de gebeurtenissen worden gebruikt om waarschuwingen te genereren en beveiligingsincidenten te identificeren.

De impact op de ROI

U vraagt zich misschien af waarom deze technieken de ROI van bedreigingsmodellering positief kunnen beïnvloeden. Vanuit ons oogpunt zijn ze cruciaal voor het verhogen van

de waarde van bedreigingsmodellering voor de DevOps-teams. Het probleem dat we herhaaldelijk hebben gezien, is dat deze teams de beveiliging ervaren als kosten die een beperkte waarde bieden en veel onvoorzien werk vereisen. Soms is het onduidelijk waarom ze zoveel van hun tijd moeten investeren in het herstellen van de beveiliging. Als gevolg hiervan wordt beveiliging een probleem in plaats van een kans. Threat modeling biedt de mogelijkheid om deze problemen op te lossen, omdat het de redenen biedt om beveiliging te implementeren. Bovendien kan het vroeg in het ontwikkelingsproces worden gestart en ontwerpfouten voorkomen die kostbaar kunnen zijn als ze niet snel worden gedetecteerd. De bovenstaande technieken zijn bedoeld om bedreigingsmodellering beter te integreren met DevOps. Dit zorgt ervoor dat zakelijke besluitvormers en ontwikkelaars bedreigingsmodellering als een natuurlijke aanvulling op het ontwikkelings- en operationele proces ervaren. Daarom neemt de waarde die wordt ontvangen door het aannemen van bedreigingsmodellering toe en nemen de kosten af vanwege de integratie met de verschillende hulpprogramma's die al in gebruik zijn.

Het werk voor threat modelers vereenvoudigen

Een ander belangrijk aspect dat nodig is om het RENDEMENT van bedreigingsmodellering te verbeteren, is gerelateerd aan het verlagen van de kosten en het verhogen van het aantal mensen dat het kan leveren, terwijl er meer homogene kwaliteitsniveaus worden gehandhaafd.

Er zijn veel pogingen om het tekort aan bevoegde mensen aan te pakken. Sommige hiervan zijn gebaseerd op de actieve betrokkenheid van het hele DevOps-team in de oefening voor threat modeling. Het idee is om een leider van het initiatief te identificeren, dat wil zeggen iemand met tussenliggende kennis over het proces, maar niet noodzakelijkerwijs een expert is, en haar de discussie tussen de andere teamleden laten leiden. Deze aanpak wordt actief goedgekeurd door de ondertekenaars van het Threat Modeling Manifesto.

We zijn het er wel mee eens dat deze aanpak een goede waarde biedt en een verbetering vormt ten opzichte van de huidige situatie. Het biedt ook goede inzichten en stelt het team in staat om de beveiligingscultuur te laten groeien. Toch is het niet zonder nadeel, omdat het slechts een paar problemen behandelt, waardoor er veel wegvalt. Dit creëert een consistentieprobleem omdat het te gemakkelijk is om het konijnengat te omlaag te gaan en kostbare tijd te verspillen aan secundaire problemen, ontbrekende belangrijke problemen. De ervaring van de leider speelt een belangrijke rol bij het voorkomen van deze situaties. Bovendien vereist deze aanpak veel tijd van alle teamleden om elk probleem te bespreken.

Daarom kan zelfs het uitgeven van een paar uur per Sprint voor deze oefening een aanzienlijke investering vertegenwoordigen. Iedereen weet dat de meeste teams vaak tijd verspillen aan grote vergaderingen waarbij iedereen betrokken is, en die bedreigingsmodelleringssessies zouden geen uitzondering maken. Toch is deze aanpak uitstekend voor kleine producten, waarbij het team een handvol senioren omvat.

Een andere benadering

Gezien de beperkingen van de vorige benadering, geven we er de voorkeur aan om het aantal vergaderingen, hun lengte en het aantal deelnemers te beperken. Daarom zou de verantwoordelijkheid van de bedreigingsmodeller belangrijker zijn: niet alleen om de interviews te leiden, maar ook om het bedreigingsmodel zelf te creëren en te onderhouden. Deze aanpak vereist meer significante competenties en expertise. Bedreigingsmodellers kunnen worden vertegenwoordigd door beveiligingsleiders of door leden van het interne beveiligingsteam. De meeste organisaties gaan voor het eerst omdat het beveiligingsteam doorgaans volledig is geboekt.

Beveiligingsleiders zijn lid van de DevOps-teams met een bepaald belang in beveiliging. Het zijn geen experts, maar ze hebben een basiskennis en de bereidheid om de beveiligingspostuur van hun team te verbeteren. Het idee is om een bevoorrechte verbinding te maken tussen de beveiligingsleiders en het interne beveiligingsteam, zodat de eersten in staat zijn om hun teams te helpen bij het doen van het juiste, terwijl het beveiligingsteam de werkbelasting kan verminderen. Met threat modeling zouden de beveiligingsleiders fungeren als bedreigingsmodellers en zou het interne beveiligingsteam de verantwoordelijkheid hebben om hen te begeleiden en hun werk te beoordelen.

> **Wat kun je vandaag doen?**
>
> Onderzoek de mogelijkheid om een Beveiligingskampioen-programma te gebruiken en gebruik te maken van het programma om uw levenscyclus voor secure softwareontwikkeling verder te versterken.

De rol van knowledge bases

Een belangrijk probleem met threat modeling is ervoor te zorgen dat de kwaliteit van de ervaring en de waarde voor het DevOps-team hoog is, ongeacht wie het bedreigingsmodel uitvoert. Met Beveiligingsleiders wordt dit probleem nog urgenter. Een idee om dit aan te pakken is om knowledge bases te bieden om het bedreigingsmodel te maken. Knowledge Bases voor threat modeling zijn pakketten met informatie over een specifieke context: ze bevatten een definitie van de entiteiten die

zijn gerelateerd aan die context, de mogelijke aanvalspatronen voor die entiteiten en de standaardbeperking die kan worden toegepast. Met Knowledge Bases kan de organisatie betere en consistentere resultaten krijgen, omdat ze referentiemateriaal vertegenwoordigen dat de threat modelers op een prescriptieve manier begeleidt. Knowledge bases moeten regels hebben waarmee we automatisch bedreigingen en oplossingen voor een systeem kunnen toepassen. Met deze automatisering kunnen we het feit overwinnen dat sommige bedreigingsmodelleerders mogelijk niet over de ervaring beschikken die nodig is om te bepalen of een bedreiging moet worden toegepast of als een oplossing effectief is.

Knowledge bases zijn geen nieuw idee: veel huidige hulpprogramma's voor threat modeling ondersteunen ze al in een bepaalde vorm. Maar veel huidige implementaties hebben aanzienlijke nadelen. U moet bijvoorbeeld eenvoudig knowledge bases kunnen onderhouden. Hun onderhoud is een probleem dat nog steeds onopgeloste is. Het is bijvoorbeeld niet eenvoudig om de beste informatiebronnen te identificeren die u kunt gebruiken om ze te bouwen. Bovendien is onderhoud doorgaans handmatig. Het maken en onderhouden van de knowledge bases moet de verantwoordelijkheid van het interne beveiligingsteam van de organisatie zijn. We hopen dat bedrijven in de toekomst kennisdatabases gaan bieden voor de meest voorkomende hulpprogramma's voor threat modeling om een aantal van de lasten van hun klanten op te heffen. Deze knowledge bases moeten flexibel zijn om hun acceptatie te ondersteunen en te vergemakkelijken, zelfs door de meest volwassen organisaties, die de genoemde knowledge bases moeten aanpassen aan hun praktijken, beleid en voorschriften.

> **Wat kun je vandaag doen?**
>
> Overweeg de mogelijkheid om deel uit te maken van de inspanningen van het gecentraliseerde beveiligingsteam voor het ontwikkelen van knowledge bases die door de verschillende ontwikkelteams kunnen worden gebruikt om threat modeling te versnellen.

Knowledge bases gebruiken

Een ander probleem met knowledge bases is dat ze soms te complex zijn om te gebruiken. Veel van hen proberen uitgebreid te zijn door essentiële en minder kritieke problemen op te slaan. Helaas zijn niet alle systemen vereist. U wilt een eenvoudigere benadering gebruiken wanneer het systeem dat u analyseert klein is en geen gevoelige gegevens verwerkt. Integendeel, u zou gedetailleerder willen gaan als het systeem complexer is en PII- en hoogwaardige gegevens verwerkt. Daarom moet het mogelijk zijn om verschillende versies van de kennis toe te passen, afhankelijk van de context of om bepaalde aanvalspatronen en bijbehorende risicobeperking als 'TOP' te markeren.

Als gevolg hiervan kunnen de bedreigingsmodelleerders beslissen of ze een uitgebreide ervaring willen of eenvoudig willen gaan en het vereiste werk minimaliseren.

Over efficiëntie gesproken, is het noodzakelijk om ervoor te zorgen dat de activiteiten zo veel mogelijk worden gestroomlijnd en geautomatiseerd om de benodigde hoeveelheid werk te verminderen. We denken dat een sweet spot voor het uitvoeren van een bedreigingsmodel van een oplossing van gemiddelde grootte 1 dag voor de bedreigingsmodeller moet zijn. Dergelijke resultaten zijn alleen mogelijk als het hulpmiddel van de keuze accelerators biedt om de benodigde tijd te besparen. Als het hulpprogramma bijvoorbeeld 20 verschillende soorten oplossingen op 100 verschillende plaatsen toepast en u wordt gevraagd om voor elk van deze locaties hun status op te geven, zou u vijf keer efficiënter zijn door u te richten op de eerste in plaats van de laatste. Het hulpprogramma van de keuze moet deze mogelijkheid bieden en tegelijkertijd de mogelijkheid verlenen om indien nodig een grondiger taak uit te voeren.

> **Wat kun je vandaag doen?**
>
> Als de knowledge bases die u vandaag gebruikt, het concept van 'TOP'-bedreigingen en -oplossingen niet ondersteunen, kunt u overwegen om te verwijderen wat zelden of nuttig is, zodat u zich alleen kunt concentreren op wat echt belangrijk is.
>
> Soms is het probleem dat de aangenomen knowledge bases algemeen proberen te zijn en meerdere scenario's behandelen. U kunt de situatie verbeteren door ze te specialiseren.

De juiste vragen stellen

Tijdens onze analyse hebben we gekeken naar de mogelijkheid om een hulpprogramma te gebruiken ter ondersteuning van een vragenframework om de eerste fasen van de analyse te stimuleren. We hebben gemerkt dat de meeste onervaren threat modelers niet de juiste vragen kunnen stellen om de informatie op te halen die nodig is voor hun analyse. Sommige van onze experts hebben aangetoond dat het mogelijk is om een aantal cruciale vragen te bepalen op basis van een systeemdiagram binnen het bereik. Deze vragen kunnen zelfs automatisch worden toegepast, met enkele generatieregels. Het probleem is dat deze benadering mogelijk niet de waarde biedt die het lijkt te beloven. Dat komt doordat u de logica achter elke vraag moet begrijpen. Anders zou u het antwoord niet kunnen evalueren en bepalen of het bevredigend is. Het genereren van geautomatiseerde vragen kan echter een aanzienlijke waarde opleveren voor de minder deskundige bedreigingsmodelleerders, waardoor het inzicht in de systemen binnen het bereik wordt verbeterd.

> **Wat kun je vandaag doen?**
>
> Gebruik een gestructureerde benadering om vragen te stellen. Ons team heeft bijvoorbeeld goede resultaten behaald door Microsoft STRIDE als referentie te gebruiken. U kunt dit doen door elk onderdeel van de oplossingsvragen te vragen, zoals:
>
> - **Adresvervalsing**: hoe verifieert het onderdeel zich bij de services en resources die het gebruikt?
> - **Manipulatie: valideert** het onderdeel de berichten die het ontvangt? Is de validatie los of strikt?
> - **Repudiation**: wordt het onderdeel waarin de interacties in een auditlogboek worden vastgelegd?
> - **Openbaarmaking van** informatie: is het verkeer dat binnenkomt en uitgaand is het onderdeel versleuteld? Welke protocollen en algoritmen zijn toegestaan?
> - **Denial of Service**: is het onderdeel geconfigureerd in hoge beschikbaarheid? Is het beveiligd tegen DDoS-aanvallen?
> - **Uitbreiding van bevoegdheden**: zijn gebruikers de minste bevoegdheden toegewezen die vereist zijn? Is de oplossingsmixcode gericht op normale gebruikers met die voor gebruikers met hoge bevoegdheden?
>
> Technieken zoals deze kunnen worden geleerd en kunnen worden verbeterd met ervaring. Daarom is het belangrijk om een continue leerbenadering te implementeren die is ontworpen om leermateriaal te verzamelen en te verspreiden binnen de organisatie.

De impact op de ROI

Het is mogelijk om veel ideeën te identificeren om de efficiëntie van de bedreigingsmodelleringservaring, de kwaliteit ervan te verbeteren en uiteindelijk de ROI te verhogen. Deze inspanning moet echter worden beschouwd als een doorlopend proces, dat moet worden gericht op de continue verbetering van de praktijk.

Conclusies

Threat modeling is een uitstekende methodologie voor het verbeteren van de beveiliging van uw organisatie. Indien correct gedaan, kan het waarde bieden voor een zeer redelijke kosten. We hebben al verschillende technieken geïdentificeerd die

essentieel kunnen zijn voor het verbeteren van de waarde van bedreigingsmodellering voor het beveiligen van moderne oplossingen, waaronder:

- Het bedreigingsmodel afstemmen met uw DevOps-oefening door
    - Gericht op de risicobeperking
    - Oplossingen koppelen aan gebruikersverhalen
    - Verschillen tussen het bedreigingsmodel en de achterstand markeren
    - Het bedreigingsmodel gebruiken om een uitgebreidere bewaking en controle voor beveiliging te stimuleren
- Vereenvoudig het maken van bedreigingsmodellen en verhoog de consistentie van de resultaten
    - Vertrouwen op beveiligingskampioenen
    - Knowledge bases gebruiken om de identificatie van bedreigingen en risicobeperking te automatiseren
    - Betere knowledge bases maken
    - Een vraagframework bieden dat wordt ondersteund door automatisering

Hopelijk zijn sommige hiervan al te vinden in uw hulpprogramma voor bedreigingsmodellering van uw keuze. Anderen zullen in de toekomst worden opgenomen. We weten dat het maximaliseren van de ROI voor threat modeling een langetermijninspanning is die antwoorden vereist die we nog niet hebben. We weten ook dat sommige vragen nog steeds onbekend zijn. In dit document moet u goed nadenken en hopelijk kunt u helpen bij het verbeteren van hoe u bedreigingsmodellering uitvoert. We hopen dat het een vuurtoren voor u en ons kan zijn en dat het nuttig zal zijn om onze inspanningen voor de komende jaren te leiden.

Microsoft Cybersecurity Defense Operations Center

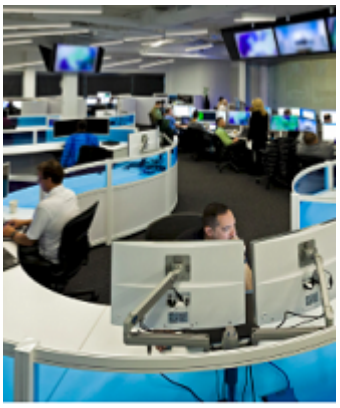
Artikel • 12-03-2025

Cybersecurity is een gedeelde verantwoordelijkheid die ons allemaal beïnvloedt. Vandaag de dag kan één inbreuk, fysiek of virtueel, miljoenen dollars schade aan een organisatie en mogelijk miljarden financiële verliezen aan de mondiale economie veroorzaken. Elke dag zien we rapporten over cybercriminelen die gericht zijn op bedrijven en individuen voor financiële winst of sociaal-gemotiveerde doeleinden. Voeg aan deze bedreigingen toe die door nationale staatsactoren die activiteiten willen verstoren, spionage willen uitvoeren of over het algemeen vertrouwen ondermijnen.

In dit kort delen we de status van onlinebeveiliging, bedreigingsactoren en de geavanceerde tactieken die ze gebruiken om hun doelen te verbeteren, en hoe het Cyber Defense Operations Center van Microsoft deze bedreigingen bestrijdt en klanten helpt hun gevoelige toepassingen en gegevens te beschermen.

Tabel uitvouwen

Het Microsoft Cyber Defense Operations Center



Microsoft doet er alles aan om de online wereld veiliger te maken voor iedereen. De cyberbeveiligingsstrategieën van ons bedrijf zijn ontwikkeld op basis van de unieke zichtbaarheid die we hebben in het snel veranderende cyberdreigingslandschap.

Innovatie in de aanvalsruimte tussen mensen, plaatsen en processen is een noodzakelijke en voortdurende investering die we allemaal moeten maken, aangezien kwaadwillende personen zich blijven ontwikkelen in zowel vastberadenheid als verfijning. In reactie op meer investeringen in verdedigingsstrategieën door veel organisaties, passen aanvallers zich aan en verbeteren ze tactieken op breakneck snelheid. Gelukkig zijn cyberdefenders zoals de wereldwijde teams voor informatiebeveiliging van Microsoft ook bezig met het innoveren en verstoren van lang betrouwbare aanvalsmethoden met doorlopende, geavanceerde training en moderne beveiligingstechnologieën, hulpprogramma's en processen.

Het Microsoft Cyber Defense Operations Center (CDOC) is een voorbeeld van de meer dan $ 1 miljard die we elk jaar investeren in beveiliging, gegevensbescherming en risicobeheer. De CDOC brengt cyberbeveiligingsspecialisten en gegevenswetenschappers samen in een faciliteit van 24x7 om bedreigingen in realtime te bestrijden. We zijn wereldwijd verbonden met meer dan 3500 beveiligingsprofessionals in onze productontwikkelingsteams, informatiebeveiligingsgroepen en juridische teams om onze cloudinfrastructuur en -services, producten en apparaten en interne resources te beschermen.

Microsoft heeft meer dan $ 15 miljard geïnvesteerd in onze cloudinfrastructuur, met meer dan 90 procent van de Fortune 500-bedrijven die gebruikmaken van de Microsoft-cloud. Tegenwoordig bezitten en bedienen we een van de grootste cloudvoetafdrukken van de wereld met meer dan 100 geografisch gedistribueerde datacenters, 200 cloudservices, miljoenen apparaten en een miljard klanten over de hele wereld.

Bedreigingsactoren en motivaties voor cyberbeveiliging

De eerste stap bij het beveiligen van mensen, apparaten, gegevens en kritieke infrastructuur is het begrijpen van de verschillende soorten bedreigingsactoren en hun motivaties.

- Cybercriminelen omvatten verschillende subcategorieën, hoewel ze vaak gemeenschappelijke motivaties delen: financiële, intelligentie en/of sociale of politieke winst. Hun

aanpak is meestal direct, door een systeem voor financiële gegevens te infiltreren, waardoor microbedragen te klein zijn om te detecteren en af te sluiten voordat ze worden gedetecteerd. Het handhaven van een permanente, clandestine aanwezigheid is essentieel voor het voldoen aan hun doelstelling.

Hun aanpak kan een inbraak zijn die een grote financiële uitbetaling afleidt via een labyrint van accounts om het bijhouden en ingrijpen te omzeilen. Soms is het doel om intellectueel eigendom te stelen dat het doel bezit, zodat de cybercrimineel fungeert als intermediair om een productontwerp, softwarebroncode of andere eigendomsinformatie te leveren die waarde heeft voor een specifieke entiteit. Meer dan de helft van deze activiteiten wordt gepleegd door georganiseerde criminele groepen.

- **Nationale staatsactoren** werken voor een overheid om gerichte overheden, organisaties of individuen te verstoren of in gevaar te brengen om toegang te krijgen tot waardevolle gegevens of intelligentie. Zij zijn betrokken bij internationale zaken om invloed uit te oefenen en een resultaat te stimuleren dat een land of landen kan bevoordelen. Het doel van een nationale actor is om activiteiten te verstoren, spionage uit te voeren tegen bedrijven, geheimen van andere overheden te stelen of anderszins vertrouwen in instellingen te ondermijnen. Ze werken met grote middelen tot hun beschikking en zonder angst voor juridische vergelding, met een toolkit die van eenvoudig tot zeer complex is.

  Nationale staatsactoren kunnen een aantal van de meest geavanceerde cyberhackingtalent aantrekken en hun hulpprogramma's naar het punt van wapenisering kunnen bevorderen. Hun inbraakbenadering omvat vaak een geavanceerde permanente bedreiging met behulp van supercomputingkracht om beveiligingsreferenties te verbreken door miljoenen pogingen om het juiste wachtwoord te verkrijgen. Ze kunnen ook hyper-gerichte phishingaanvallen gebruiken om een insider aan te trekken om hun referenties te onthullen.

- **Insider-bedreigingen** zijn met name lastig vanwege de onvoorspelbaarheid van menselijk gedrag. De motivatie voor een insider misschien opportunistisch en voor financiële winst. Er zijn echter meerdere oorzaken voor potentiële bedreigingen van binnenuit, variërend van eenvoudige onzorgvuldigheid tot geavanceerde schema's. Veel schendingen van gegevens als gevolg van insiderbedreigingen zijn volledig onbedoeld vanwege onopzettelijke of onopzettelijke activiteiten die een organisatie in gevaar brengen zonder op de hoogte te zijn van

het beveiligingsprobleem.

- **Hacktivisten** richten zich op politieke en/of sociaal-gemotiveerde aanvallen. Ze streven ernaar zichtbaar en herkend te worden in het nieuws om aandacht te vestigen op zichzelf en hun oorzaak. Hun tactieken zijn onder andere DDoS-aanvallen (Distributed Denial-of-Service), aanvallen op beveiligingsproblemen of het afmaken van een onlineaanzondering. Een verbinding met een sociaal of politiek probleem kan elk bedrijf of elke organisatie een doel maken. Social media stelt hacktivisten in staat om hun zaak snel te evangeliseren en anderen te werven om deel te nemen.

Technieken voor bedreigingsacteur

Aanvallers zijn bedreven in het vinden van manieren om het netwerk van een organisatie binnen te dringen ondanks de beveiligingen die worden toegepast met behulp van verschillende geavanceerde technieken. Sinds de vroege dagen van internet zijn er verschillende tactieken geweest, maar anderen weerspiegelen de creativiteit en toenemende verfijning van de huidige aanvallers.

- **Social engineering** is een brede term voor een aanval die gebruikers misleidt in het handelen of openbaar maken van informatie die ze anders niet zouden doen. Social engineering speelt in op de goede bedoelingen van de meeste mensen en hun bereidheid om nuttig te zijn, om problemen te voorkomen, vertrouwde bronnen te vertrouwen of om potentieel een beloning te krijgen. Andere aanvalsvectoren kunnen onder de paraplu van social engineering vallen, maar het volgende zijn enkele van de kenmerken die social engineering tactieken gemakkelijker te herkennen en te verdedigen tegen:
  - **Phishing-e-mailberichten** zijn een effectief hulpmiddel omdat ze spelen tegen de zwakste koppeling in de beveiligingsketen: dagelijkse gebruikers die niet denken aan netwerkbeveiliging als top-of-mind. Een phishingcampagne kan een gebruiker uitnodigen of bang maken om per ongeluk hun referenties te delen door ze te misleiden om te klikken op een koppeling waarvan ze denken dat ze een legitieme site zijn of een bestand met schadelijke code downloaden. Phishing-e-mails waren vroeger slecht geschreven en gemakkelijk te herkennen. Tegenwoordig zijn kwaadwillende mensen in staat geworden om legitieme e-mailberichten en landingssites

na te bootsen die moeilijk als frauduleus kunnen worden geïdentificeerd.
    - **Bij identiteitsvervalsing** wordt een kwaadwillende gebruiker gemaskeerd als een andere legitieme gebruiker door de informatie die wordt gepresenteerd aan een toepassing of netwerkresource te vervalsen. Een voorbeeld is een e-mailbericht dat schijnbaar het adres bevat van een collega die actie aanvraagt, maar het adres verbergt de echte bron van de afzender van de e-mail. Op dezelfde manier kan een URL worden vervalst om te worden weergegeven als een legitieme site, maar het werkelijke IP-adres verwijst eigenlijk naar de site van een cybercrimineel.

- Malware is sinds de dageraad van computing bij ons geweest. Vandaag zien we een sterke up-tick in ransomware en schadelijke code die specifiek is bedoeld om apparaten en gegevens te versleutelen. Cybercriminelen vragen vervolgens betaling in cryptovaluta voor de sleutels om de controle te ontgrendelen en terug te keren naar het slachtoffer. Dit kan gebeuren op individueel niveau voor uw computer en gegevensbestanden, of nu vaker, voor een hele onderneming. Het gebruik van ransomware wordt met name uitgesproken op het gebied van gezondheidszorg, omdat de levens- of doodsgevolgen van deze organisaties hen zeer gevoelig maken voor netwerk downtime.

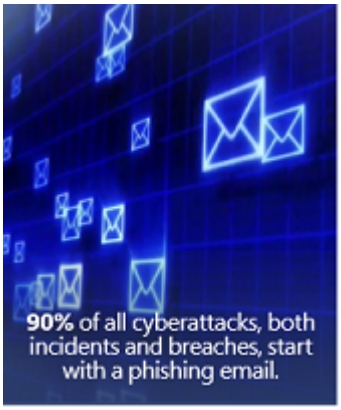
- **Supply chain insertion** is een voorbeeld van een creatieve benadering van het injecteren van malware in een netwerk. Door bijvoorbeeld een updateproces voor toepassingen te kapen, omzeilt een kwaadwillende persoon antimalwarehulpprogramma's en -beveiligingen. We zien dat deze techniek gebruikelijker wordt en deze bedreiging blijft groeien totdat uitgebreidere beveiligingsbeveiligingen door toepassingsontwikkelaars worden opgenomen in software.

- **Bij man-in-the-middle-aanvallen**moet een kwaadwillende persoon zichzelf invoegen tussen een gebruiker en een resource die ze openen, waardoor kritieke informatie wordt onderschept, zoals de aanmeldingsreferenties van een gebruiker. Een cybercrimineel in een koffiebar kan bijvoorbeeld gebruikmaken van key-logging software om de domeinreferenties van een gebruiker vast te leggen terwijl ze lid worden van het wifi-netwerk. De bedreigingsacteur kan vervolgens toegang krijgen tot de gevoelige informatie van de gebruiker, zoals bankgegevens en persoonlijke gegevens die ze op het donkere web kunnen gebruiken of verkopen.

- **DDoS-aanvallen (Distributed Denial of Service)** zijn meer dan een decennium geleden en er komen enorme aanvallen meer voor met de snelle groei van internet of things (IoT). Bij het

gebruik van deze techniek overweldigt een aanvaller een site door deze te overbelasten met schadelijk verkeer dat legitieme query's verdringt. Eerder geplaatste malware wordt vaak gebruikt om een IoT-apparaat zoals een webcam of slimme thermostaat te kapen. Bij een DDoS-aanval overspoelt binnenkomend verkeer van verschillende bronnen een netwerk met talloze aanvragen. Dit overweldigt servers en weigert de toegang van legitieme aanvragen. Veel aanvallen omvatten ook het vervalsen van IP-afzenderadressen (IP-adresvervalsing), zodat de locatie van de aanvalsmachines niet gemakkelijk kan worden geïdentificeerd en verslagen.

Vaak wordt een Denial of Service-aanval gebruikt om een meer misleidende inspanning te dekken of af te leiden om een organisatie binnen te dringen. In de meeste gevallen is het doel van de kwaadwillende persoon om toegang te krijgen tot een netwerk met behulp van gecompromitteerde referenties en zich lateraal over het netwerk te verplaatsen om toegang te krijgen tot krachtigere referenties die de sleutels zijn voor de meest gevoelige en waardevolle informatie binnen de organisatie.



De democratisering van cyberspace

De groeiende mogelijkheid van cyberoorlogse is een van de belangrijkste zorgen van regeringen en burgers vandaag. Het gaat om natiestaten die computers en netwerken in oorlogvoering gebruiken en instellen.

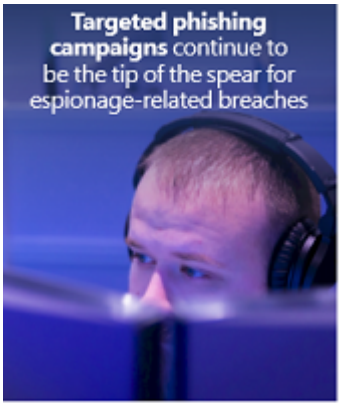
Zowel aanstootgevende als defensieve operaties worden gebruikt om cyberaanvallen, spionage en sabotage uit te voeren. Natiestaten ontwikkelen hun mogelijkheden en zijn al vele jaren betrokken bij cyberoorlogse, ofwel als agressors, verdachten of beide.

Nieuwe bedreigingstools en tactieken die zijn ontwikkeld via geavanceerde militaire investeringen kunnen ook worden geschonden en cyberdreigingen kunnen online worden gedeeld en worden gewapend door cybercriminelen voor verder gebruik.

Het microsoft-cyberbeveiligingspostuur

Hoewel beveiliging altijd een prioriteit voor Microsoft is geweest, erkennen we dat de digitale wereld continue vooruitgang vereist in onze toezegging om bedreigingen voor cyberbeveiliging te beschermen, te detecteren en erop te reageren. Deze drie toezeggingen definiëren onze benadering van cyberverdediging en fungeren als een nuttig kader voor onze bespreking van de strategieën en mogelijkheden voor cyberbeveiliging van Microsoft.

BESCHERMEN Beschermen



De eerste toezegging van Microsoft is het beschermen van de computeromgeving die door onze klanten en werknemers wordt gebruikt om de tolerantie van onze cloudinfrastructuur en -services, producten, apparaten en de interne bedrijfsresources van het bedrijf te beschermen tegen bepaalde kwaadwillende personen.

De beveiligingsmaatregelen van de CDOC-teams omvatten alle eindpunten, van sensoren en datacenters tot identiteiten en SaaS-toepassingen (Software-as-a-Service). Defensie-indepth, het toepassen van besturingselementen op meerdere lagen met overlappende beveiligings- en risicobeperkingsstrategieën, is een best practice in de hele branche en het is de benadering die we nemen om onze waardevolle klant- en bedrijfsactiva te beschermen.

De beveiligingstactieken van Microsoft zijn onder andere:

- Uitgebreide bewaking en controle over de fysieke omgeving van onze wereldwijde datacenters, waaronder camera's, personeelscontrole, hekken en barrières, en meerdere identificatiemethoden voor fysieke toegang.

- Softwaregedefinieerde netwerken die onze cloudinfrastructuur beschermen tegen inbraak en DDoS-aanvallen.

- Meervoudige verificatie wordt in onze infrastructuur gebruikt om identiteits- en toegangsbeheer te beheren. Het zorgt ervoor dat kritieke resources en gegevens worden beveiligd door ten minste twee van de volgende:
  - Iets wat u weet (wachtwoord of pincode)
  - Iets wat u bent (biometrie)
  - Iets dat u hebt (smartphone)
- Niet-permanente administratie maakt gebruik van Just-In-Time (JIT) en just-enough administratorbevoegdheden (JEA) voor technische medewerkers die infrastructuur en services beheren. Dit biedt een unieke set referenties voor verhoogde toegang die automatisch verloopt na een vooraf aangewezen duur.

- De juiste hygiëne wordt strikt onderhouden via up-to-date, antimalwaresoftware en naleving van strikte patches en configuratiebeheer.

- Microsoft Centrum voor beveiliging tegen schadelijke software team van onderzoekers identificeert, reverse-engineert en ontwikkelt malwarehandtekeningen en implementeert deze vervolgens in onze infrastructuur voor geavanceerde detectie en verdediging. Deze handtekeningen worden gedistribueerd naar onze responders, klanten en de branche via Windows Updates en meldingen om hun apparaten te beveiligen.

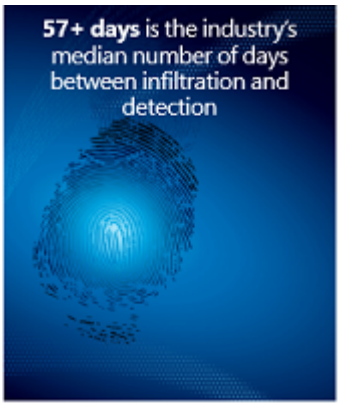
- Microsoft Security Development Lifecycle (SDL) is een softwareontwikkelingsproces dat ontwikkelaars helpt bij het bouwen van veiligere software en het voldoen aan vereisten voor beveiligingsnaleving en het verlagen van de ontwikkelingskosten. De SDL wordt gebruikt om alle toepassingen, onlineservices en producten te beveiligen en de effectiviteit ervan regelmatig te valideren door middel van penetratietests en scannen op beveiligingsproblemen.

- Bedreigingsmodellering en analyse van kwetsbaarheid voor aanvallen zorgt ervoor dat mogelijke bedreigingen worden beoordeeld, blootgestelde aspecten van de service worden geëvalueerd en het kwetsbaarheid voor aanvallen wordt geminimaliseerd door services te beperken of onnodige functies te elimineren.

- Het classificeren van gegevens op basis van de gevoeligheid en het nemen van de juiste maatregelen om deze te beveiligen, inclusief versleuteling in transit en at-rest, en het afdwingen van het principe van toegang met minimale bevoegdheden biedt extra beveiliging. • Bewustzijnstraining die een vertrouwensrelatie tussen de gebruiker en het beveiligingsteam bevordert om een omgeving te ontwikkelen waarin gebruikers incidenten en afwijkingen melden zonder angst voor gevolgen.

Een uitgebreide set besturingselementen en een diepgaande strategie voor verdediging helpt ervoor te zorgen dat elk gebied uitvalt, er compenserende controles zijn op andere gebieden om de beveiliging en privacy van onze klanten, cloudservices en onze eigen infrastructuur te behouden. Er is echter geen omgeving die echt ondoordringbaar is, omdat mensen fouten maken en bepaalde kwaadwillende personen blijven zoeken naar beveiligingsproblemen en ze misbruiken. De aanzienlijke investeringen die we blijven doen in deze beveiligingslagen en basislijnanalyse maken het mogelijk om snel te detecteren wanneer abnormale activiteit aanwezig is.

DETECTEREN Detecteren



De CDOC-teams maken gebruik van geautomatiseerde software, machine learning, gedragsanalyse en forensische technieken om een intelligente beveiligingsgrafiek van onze omgeving te maken. Dit signaal is verrijkt met contextuele metagegevens en gedragsmodellen die zijn gegenereerd op basis van bronnen zoals Active Directory, asset- en configuratiebeheersystemen en gebeurtenislogboeken.

Onze uitgebreide investeringen in beveiligingsanalyse bouwen uitgebreide gedragsprofielen en voorspellende modellen die ons in staat stellen om de puntjes te verbinden en geavanceerde bedreigingen te identificeren die anders niet zijn gedetecteerd, en vervolgens tegen sterke insluitings- en gecoördineerde herstelactiviteiten.

Microsoft maakt ook gebruik van aangepaste beveiligingssoftware, samen met hulpprogramma's voor industrieleading en machine learning. Onze bedreigingsinformatie ontwikkelt zich voortdurend, met geautomatiseerde gegevensverrijking om schadelijke activiteiten sneller te detecteren en met hoge betrouwbaarheid te rapporteren. Beveiligingsscans worden regelmatig uitgevoerd om de effectiviteit van beschermende maatregelen te testen en te verfijnen. De breedte van de investering van Microsoft in het beveiligingsecosysteem en de verscheidenheid aan signalen die door de CDOC-teams worden bewaakt, bieden een uitgebreidere bedreigingsweergave dan kan worden bereikt door de meeste serviceproviders.

De detectietactieken van Microsoft zijn onder andere:

- Bewaking van netwerk- en fysieke omgevingen 24x7x365 voor mogelijke cyberbeveiligingsevenementen. Gedragprofilering is gebaseerd op gebruikspatronen en inzicht in unieke bedreigingen voor onze services.

- Identiteits- en gedragsanalyses worden ontwikkeld om abnormale activiteit te markeren.

- Machine learning-softwarehulpprogramma's en -technieken worden regelmatig gebruikt om onregelmatigheden te detecteren en te markeren.

- Geavanceerde analytische hulpprogramma's en processen worden geïmplementeerd om afwijkende activiteiten en innovatieve correlatiemogelijkheden verder te identificeren. Hierdoor kunnen zeercontextualized detecties worden

gemaakt op basis van de enorme hoeveelheden gegevens in bijna realtime.

- Geautomatiseerde op software gebaseerde processen die continu worden gecontroleerd en ontwikkeld voor een grotere effectiviteit.

- Gegevenswetenschappers en beveiligingsexperts werken regelmatig naast elkaar om geëscaleerde gebeurtenissen aan te pakken die ongebruikelijke kenmerken vertonen die verdere analyse van doelen vereisen. Vervolgens kunnen ze potentiële reactie- en herstelinspanningen bepalen.

REAGEREN

Reageren

Wanneer Microsoft abnormale activiteiten in onze systemen detecteert, worden onze reactieteams geactiveerd om te reageren met nauwkeurige kracht. Meldingen van op software gebaseerde detectiesystemen stromen via onze geautomatiseerde responssystemen met behulp van op risico's gebaseerde algoritmen om gebeurtenissen te markeren die tussenkomst van ons antwoordteam vereisen. Mean-Time-to-Mitigation is van het grootste belang en ons automatiseringssysteem biedt beantwoorders relevante, bruikbare informatie die het triage, risicobeperking en herstel versnelt.

Om beveiligingsincidenten op zo'n enorme schaal te beheren, implementeren we een gelaagd systeem om efficiënt antwoordtaken toe te wijzen aan de juiste resource en een rationeel escalatiepad te vergemakkelijken.

De reactietactieken van Microsoft zijn onder andere:

- Geautomatiseerde responssystemen maken gebruik van op risico's gebaseerde algoritmen om gebeurtenissen te markeren die menselijke tussenkomst vereisen.

- Geautomatiseerde responssystemen maken gebruik van op risico's gebaseerde algoritmen om gebeurtenissen te markeren die menselijke tussenkomst vereisen.

- Goed gedefinieerde, gedocumenteerde en schaalbare processen voor incidentrespons binnen een model voor continue verbetering helpen ons voor te blijven door deze beschikbaar te maken voor alle responders.

- Expertise op het gebied van onderwerpen in onze teams, op meerdere beveiligingsgebieden, biedt een diverse vaardighedenset voor het aanpakken van incidenten. Beveiligingsexpertise in incidentrespons, forensische en inbraakanalyse; en een diepgaand begrip van de platforms, services en toepassingen die in onze clouddatacentra werken.

- Brede onderneming die zoekt in de cloud, hybride en on-premises gegevens en systemen om het bereik van een incident te bepalen.

- Grondige forensische analyse voor grote bedreigingen wordt uitgevoerd door specialisten om incidenten te begrijpen en te helpen bij hun insluiting en uitroeiing. • Met de hulpprogramma's voor beveiligingssoftware van Microsoft, automatisering en hyperschaalcloudinfrastructuur kunnen onze beveiligingsexperts de tijd beperken om cyberaanvallen te detecteren, onderzoeken, analyseren, reageren en herstellen.

- Penetratietests worden in alle Microsoft-producten en -services gebruikt via lopende Red Team/Blue Team-oefeningen om beveiligingsproblemen op te lossen voordat een echte kwaadwillende persoon deze zwakke punten voor een aanval kan benutten.

Cyberdefense voor onze klanten

We worden vaak gevraagd welke hulpprogramma's en processen onze klanten kunnen gebruiken voor hun eigen omgeving en hoe Microsoft kan helpen bij hun implementatie. Microsoft heeft veel van de cyberdefense-producten en -services die we in de CDOC gebruiken geconsolideerd in een reeks producten en services. De Microsoft Enterprise Cybersecurity Group en Microsoft Consulting Services-teams nemen contact op met onze klanten om de oplossingen te leveren die het meest geschikt zijn voor hun specifieke behoeften en vereisten.

Een van de eerste stappen die Microsoft ten zeerste aanbeveelt, is het opzetten van een beveiligingsbasis. Onze basisservices bieden kritieke aanvalsbeveiligingen en kernservices voor identiteitsondersteuning die u helpen om ervoor te zorgen dat assets worden beveiligd. De basis helpt u om uw digitale transformatietraject te versnellen om naar een veiliger moderne onderneming te gaan.

Op basis van deze basis kunnen klanten vervolgens gebruikmaken van oplossingen die zijn bewezen succesvol met andere Microsoft-

klanten en geïmplementeerd in de eigen IT- en cloudservicesomgevingen van Microsoft. Ga naar Microsoft.com/security en neem contact op met onze teams op cyberservices@microsoft.comvoor meer informatie over onze cyberbeveiligingsprogramma's, mogelijkheden en serviceaanbiedingen voor ondernemingen.

Aanbevolen procedures voor het beveiligen van uw omgeving

Tabel uitvouwen

Investeren in uw platform	Investeren in uw instrumentatie	Investeren in uw mensen
Flexibiliteit en schaalbaarheid vereisen planning en bouw van platform	*Zorg ervoor dat u de elementen in uw platform volledig meet*	*Ervaren analisten en gegevenswetenschappers vormen de basis van verdediging, terwijl gebruikers de nieuwe beveiligingsperimeter vormen*
Een goed gedocumenteerde inventaris van uw assets onderhouden	De hulpprogramma's verkrijgen en/of bouwen die nodig zijn om uw netwerk, hosts en logboeken volledig te bewaken	Establsih-relaties en communicatielijnen tussen het incidentresponsteam en andere groepen
Zorg voor een goed gedefinieerd beveiligingsbeleid met duidelijke standaarden en richtlijnen voor uw organisatie	Beheer en metingen proactief en test ze regelmatig op nauwkeurigheid en effectiviteit	Adopt least privilege administrator principles; permanente beheerdersrechten elimineren
Goede hygiëne handhaven: de meeste aanvallen kunnen worden voorkomen met	Houd de controle over het beleid voor wijzigingsbeheer strikt bij	Gebruik het geleerde proces om waarde te verkrijgen van elk belangrijk incident

tijdige patches en antivirus		
Meervoudige verificatie gebruiken om de beveiliging van accounts en apparaten te versterken	Controleren op abnormale account- en referentieactiviteit om misbruik te detecteren	Gebruikers inschakelen, trainen en in staat stellen om waarschijnlijke bedreigingen en hun eigen rol te herkennen bij het beveiligen van bedrijfsgegevens

Feedback

Is deze pagina nuttig? Yes No

Beveiligingsnieuws en hoogtepunten

Artikel • 27-03-2025

De site met beveiligingsdocumentatie biedt u actuele nieuws en hoogtepunten van Microsoft. Nieuws en hoogtepunten worden regelmatig bijgewerkt om u de meest relevante beveiligingsinformatie te geven.

Hier volgen eerdere nieuws en hoogtepunten die u misschien hebt gemist of die u opnieuw wilt bezoeken.

2024

- SIEM-beveiligingsbewerkingen (Beveiligingsinformatie en gebeurtenisbeheer) configureren met behulp van Microsoft Sentinel-
- Microsoft Ignite
- Wat is ransomware?
- Azure-services en -workloads beveiligen met microsoft Defender for Cloud-controles voor naleving van regelgeving
- Azure-netwerkbeveiliging

2023

- Microsoft Build 2023
- Gedeelde verantwoordelijkheid in de cloud
- Microsoft Defender XDR (uitgebreide detectie en respons) evalueren en testen
- Meer informatie over machtigingenbeheer, een CIEM-oplossing (cloudinfrastructuurrechtenbeheer)
- Veilige IaaS-services in Amazon Web Services
- snel ransomwarebeveiligingen implementeren
- Snelle moderniseringsplan voor beveiliging
- Zero Trust-implementatieplan met Microsoft 365
- Ransomware-beveiliging in Azure
- Meer informatie over de Microsoft Entra-familie van identiteits- en toegangsoplossingen voor meerdere clouds
- Veilige bestandendeling en samenwerking instellen met Microsoft Teams
- Zero Trust-principes toepassen op Azure IaaS-
- Leidende principes van Zero Trust
- cloudadaptatie-evaluaties
- De onveranderbare wetten van beveiliging

2022

- Resources voor het versnellen van uw Zero Trust-traject
- Azure-verdedigingen tegen ransomware-aanval
- Drie stappen om ransomware te voorkomen en te herstellen
- Zero Trust Essentials videoserie
- door mensen bediende ransomware-

2021

- Microsoft Pluton - CPU-beveiliging voor pc's
- Cyberaanvallen gericht op gezondheidszorg
- Microsoft Digital Defense Report
- Basislijninstellingen voor beveiligingsconfiguratie voor Windows 10 en Windows Server versie 2004
- azure-platformintegriteit en -beveiliging
- ingebouwde beveiligingsmaatregelen van Microsoft
- Lessen geleerd van de Microsoft SOC - Zen en de kunst van het opsporen van bedreigingen

2020

- cyberbeveiligingsrollen en -verantwoordelijkheden
- Netwerken (naar de cloud) - Het standpunt van één architect
- Word een Microsoft Sentinel Ninja - niveau 400 training
- Cloud Adoption Framework voor Azure : een beveiligingsstrategie definiëren
- migreren naar Microsoft Defender voor Eindpunt vanaf niet-Microsoft endpointbescherming
- Cloud Adoption Framework: beveiliging implementeren in de bedrijfsomgeving
- Beveiligingshindernissen die je kunt overwinnen — Het standpunt van een architect
- Azure Security Podcast - Netwerkisolatie en privé-eindpunten
- CISO-serie: Lessen geleerd van de Microsoft SOC - Deel: Een dag in het leven deel 2
- Azure-bedreigingsdetectie
- beveiligingsfoutrapporten identificeren op basis van rapporttitels en ruisgegevens
- het TLS 1.0-probleem oplossen, 2e editie
- Wat is Microsoft Defender voor Cloud?
- het beveiligen van de toekomst van kunstmatige intelligentie en machine learning bij Microsoft
- Microsoft Secure Score

Feedback

Is deze pagina nuttig? Yes No