SOPHOS

Exploits unter der Lupe:

Umfassende Exploit Prevention

Exploits nutzen Schwachstellen in legitimen Software-Produkten wie Adobe Flash und Microsoft Office aus, um Computer für kriminelle Zwecke zu infizieren. Häufig werden sie von Cyberkriminellen genutzt, um die Abwehrmaßnahmen von Unternehmen zu überlisten. Die Beweggründe dieser Kriminellen sind vielfältig: Manche wollen Daten stehlen oder Lösegeld für die Herausgabe von Daten erpressen, andere Informationen über ihr Ziel sammeln oder einfach nur mehr gewöhnliche Malware in Umlauf bringen.

Exploits werden häufig im Rahmen von Cyberangriffen verwendet: Bei über 90 % aller gemeldeten Datenpannen kann der Einsatz eines Exploits an einem oder mehreren Punkten der Angriffskette nachgewiesen werden. Eine Exploit Prevention darf also in keiner umfassenden Sicherheitslösung fehlen.

Exploits gibt es schon seit mehr als 30 Jahren. Kein Wunder also, dass die meisten IT-Security-Anbieter die eine oder andere Form der Exploit Prevention im Angebot haben. Die Qualität dieser Exploit Prevention kann jedoch stark variieren. Für einige Anbieter ist Exploit Prevention eine reine Pflichtübung, für andere wiederum ein wichtiger strategischer Schwerpunkt. In diesem Whitepaper erfahren Sie mehr über Exploits und welche Exploit-Prevention-Funktionen die führenden Security-Produkte zu bieten haben.

Sophos-Whitepaper März 2018

Inhaltsverzeichnis

Die Exploit-Branche: Crimeware as a Service

Dank Exploit-Kits müssen sich Malware-Autoren keine Gedanken darüber machen, wie sie in Java, Silverlight oder Flash Bugs finden, wie sie aus diesen Bugs Exploits machen, wie sie unsichere Web-Server zum Hosten von Exploits finden oder wie sie potenzielle Opfer auf schädliche Webseiten locken.

Gleichzeitig müssen die Exploit-Kit-Autoren selbst keine Malware schreiben. Sie müssen keine Server betreiben, um infizierte Computer im Auge zu behalten, oder Geld von einzelnen Opfern eintreiben. Und sie müssen sich nicht mit der Exfiltration oder dem Verkauf von Daten befassen.

Cyberkriminalität hat sich zu einer milliardenschweren Branche entwickelt, die Prognosen zufolge bis 2019 Schäden in Höhe von fast 2 Trillionen USD anrichten wird.

Kriminelle befinden sich in der „glücklichen" Lage, sich auf einen oder mehrere Teile der Bedrohungslandschaft spezialisieren zu können – in einem System, das mittlerweile scherzhaft auch als Crimeware-as-a-Service oder kurz CaaS bezeichnet wird.

In dieser mittlerweile sehr lukrativen Branche treten immer häufiger sogenannte Exploit-Broker in Erscheinung: Diese Broker kaufen Exploits von Personen, die diese entdecken, und verkaufen sie anschließend an Interessierte weiter, z. B. an staatliche Stellen oder Hacker.

Die Käufer behalten ihre Motive gerne für sich. Kevin Mitnick, Gründer von Mitnick's Absolute Zero Day Exploit Exchange, erklärt Wired: „Wenn einer unserer Kunden eine Zero-Day-Schwachstelle kaufen möchte, stellen wir keine Fragen und würden, selbst wenn wir das täten, keine Antwort erhalten. Forscher finden die Schwachstellen, verkaufen sie für X an uns, wir verkaufen Sie für Y an unsere Kunden und streichen die Gewinnmarge durch den Weiterverkauf ein."

„Wenn einer unserer Kunden eine Zero-Day-Schwachstelle kaufen möchte, stellen wir keine Fragen und würden, selbst wenn wir das täten, keine Antwort erhalten. Forscher finden die Schwachstellen, verkaufen sie für X an uns, wir verkaufen Sie für Y an unsere Kunden und streichen die Gewinnmarge durch den Weiterverkauf ein."

Kevin Mitnick

Techniken zur Exploit-Abwehr

Da mittlerweile täglich mehr als 400.000 einzigartige Malware-Samples erstellt und jedes Jahr Tausende neuer Schwachstellen aufgedeckt werden, wird es immer schwieriger, Angriffe zu verhindern. Das explosive Wachstum von Malware-Varianten erfordert neue und innovative Abwehrkonzepte zum Schutz vor Cyberkriminellen.

Bei näherer Betrachtung der modernen Cybercrime-Branche wird deutlich, dass gute Voraussetzungen für eine asymmetrische Abwehr vorhanden sind. Denn trotz der endlos erscheinenden Flut neuer Angriffe gibt es insgesamt nur etwa zwei Dutzend Techniken, um Software anzugreifen.

Ein Konzept, das sich auf die Bekämpfung dieser zahlenmäßig relativ beschränkten Exploit-Techniken konzentriert, ist anderen Ansätzen klar überlegen, da nicht jeder einzelne Exploit bekämpft werden muss.

Je nach Schwachstelle müssen Angreifer oft eine Reihe von Exploit-Techniken kombinieren, bis sie in der Lage sind, ihre Malware erfolgreich zu installieren. Diese Techniken verändern sich von Jahr zu Jahr nicht sonderlich: Jedes Jahr kommen vielleicht ein bis zwei neue Exploit-Techniken zur Liste der existierenden Techniken hinzu.

Wenn man jedoch die führenden Security-Produkte unter die Lupe nimmt, fällt schnell auf, dass erstaunlich wenige leistungsstarke Verfahren zur Bekämpfung von Exploit-Techniken

anbieten. Manche der neuen selbst ernannten „Next-Gen-Technologie"-Anbieter räumen der Exploit-Abwehr zwar mehr Bedeutung ein, vernachlässigen jedoch wichtige Aspekte.

Nachstehend finden Sie eine Liste der Anti-Exploit-Funktionen, mit denen ganze Klassen von Schwachstellen eliminiert und Exploit-Techniken ausgeschaltet werden können, die von Cyberkriminellen und Nationalstaaten genutzt werden. Die Abwehrverfahren für jede dieser Techniken variieren je nach Anbieter. Anbieter, die sich die Abwehr von Exploits auf die Fahne schreiben, schützen oft nur vor einem Bruchteil der häufig verwendeten Exploit-Methoden und ihre Maßnahmen greifen für 64-Bit-Anwendungen häufig nicht. Nur Sophos bietet eine wirklich umfassende Exploit Prevention.

Enforce Data Execution Prevention (DEP)

Unter Data Execution Prevention (DEP) verbirgt sich eine Reihe von Hardware- und Software-Technologien, die zusätzliche Speicherüberprüfungen durchführen, um Pufferüberläufe zu verhindern. Ohne DEP kann ein Angreifer versuchen, eine Software-Schwachstelle auszunutzen, indem er zu Schadcode (Shellcode) an einem Speicherort springt, an dem sich vom Angreifer kontrollierte Daten befinden (z. B. Heap oder Stack). Ohne DEP werden diese Bereiche normalerweise als ausführbar markiert, sodass Schadcode ausgeführt werden kann.

DEP ist eine Opt-in-Option für Windows XP und höher, die vom Software-Anbieter beim Kompilieren einer Anwendung aktiviert werden muss. Zudem gibt es Angriffe zum Umgehen vom integriertem DEP-Schutz. Es ist daher nicht anzuraten, sich auf die Betriebssystemimplementierung zu verlassen.

Mandatory Address Space Layout Randomization (ASLR)

Einige Exploits nehmen gezielt Speicherorte ins Visier, die bekanntermaßen mit bestimmten Prozessen verknüpft sind. In älteren Versionen von Windows (auch Windows XP) wurden Hauptprozesse beim Systemstart in der Regel in vorhersagbare Speicherorte geladen. Address Space Layout Randomization (ASLR) randomisiert die von Systemdateien und anderen Programmen genutzten Speicherorte, sodass Angreifer den Speicherort eines bestimmten Prozesses nicht mehr so leicht vorhersagen können, einschließlich der Basis der ausführbaren Datei und den Positionen des Stacks, Heaps und der Libraries.

ASLR ist ausschließlich unter Windows Vista und höher verfügbar und muss wie DEP vom Software-Anbieter beim Kompilieren der Anwendung aktiviert werden. Und genau wie bei DEP gibt es auch hier Angriffe zum Umgehen von integriertem ASLR-Schutz, weshalb es wiederum nicht anzuraten ist, sich auf die Betriebssystemimplementierung zu verlassen.

Bottom-up ASLR

Bei Aktivierung verbessert Bottom-Up ASLR die Entropie oder Zufälligkeitsstufe der verbindlichen ASLR.

Der Hauptvorteil von Mandatory ASLR und Bottom-Up ASLR in Sophos Intercept X besteht darin, dass Basisadressen von Anwendungen nicht nur bei jedem Neustart, sondern auch bei jedem Start der geschützten Anwendung randomisiert werden.

Null Page (Null Dereference Protection)

Seit Windows 8 verweigert Microsoft Programmen, die „NULL Page" (Speicher unter virtuellen Adressen 0x00000000 im Adressraum) zuzuteilen und/oder zuzuordnen. Auf diese Weise verhindert Microsoft die direkte Ausnutzung einer ganzen Kategorie von Schwachstellen, die als „NULL Pointer Dereference"-Schwachstellen bezeichnet werden.

Unter Windows XP, Windows Vista und Windows 7 würde die Ausnutzung einer solchen Schwachstelle den Angreifer in die Lage versetzen, Code im Kontext des Kernels (unter der ring0-CPU-Berechtigungsstufe) auszuführen. Das Ergebnis wäre eine Berechtigungsausweitung bis auf eine der höchsten Ebenen.

Über solche Schwachstellen verschaffen sich Angreifer Zugriff auf praktisch alle Bereiche des Betriebssystems.

Heap Spray Pre-Allocation

Heap Spray ist eine Technik, die selbst keine Schwachstellen ausnutzt, sondern die Ausnutzung einer Schwachstelle erleichtert. Mit einer Technik namens Heap Feng Shui[1] ist ein Angreifer in der Lage, beabsichtigte Datenstrukturen oder Shellcode zuverlässig auf dem Heap zu positionieren und auf diese Weise eine verlässliche Ausnutzung einer Software-Schwachstelle zu erleichtern.

Bei einer typischen Heap Spray Mitigation werden allgemein genutzte Speicheradressen reserviert, damit diese nicht von Payloads beherbergt werden können. Versierte Angreifer kennen diese Adressen, weshalb Heap Spray Mitigation in der Realität wenig ausrichten kann. Heap Spray Pre-Allocation (auch bekannt als Anti-HeapSpray Enforcement oder Shellcode Preallocation) ist in der Regel effektiv gegen Standard-Exploits, die von Prüforganisationen genutzt werden.

Dynamic Heap Spray

Im Vergleich zur statischen Heap Spray Pre-Allocation wird die Dynamic Heap Spray Mitigation in der Regel durch einen plötzlichen Anstieg der Arbeitsspeichernutzung ausgelöst.

Die Dynamic Heap Spray Erkennung analysiert die Inhalte kürzlicher Speicherbelegungen, um Muster zu erkennen, die auf Heap Sprays hindeuten, die NOP Sleds, polymorphe NOP Sleds, JavaScript Arrays und andere verdächtige Sequenzen enthalten. Diese werden auf dem Heap platziert, um Exploit-Angriffe zu erleichtern.

Stack Pivot

Der Stack einer Anwendung ist ein Speicherbereich, in dem sich unter anderem eine Liste von Speicher-Adresspositionen (sogenannte Rücksprungadressen) befindet. Hier ist der Code gespeichert, den der Prozessor in der nahen Zukunft für seine Ausführung benötigt.

Stack Pivoting wird von Schwachstellen-Exploits häufig genutzt, um Schutzmaßnahmen wie DEP zu umgehen, beispielsweise durch Verkettung von ROP Gadgets in einem Return-Oriented-Programming-Angriff.

Mit Stack Pivoting können Angreifer vom echten Stack zum neuen falschen Stack umleiten. Hierbei kann es sich um einen vom Angreifer kontrollierten Puffer wie den Heap handeln, von dem aus Angreifer den zukünftigen Ablauf der Programmausführung steuern können.

1 https://cansecwest.com/slides/2014/The%20Art%20of%20Leaks%20-%20read%20version%20-%20Yoyo.pdf

Stack Exec (MemProt)

Unter normalen Umständen enthält der Stack Daten und Adressen, die auf Code für den Prozessor verweisen, der in naher Zukunft ausgeführt werden soll. Unter Verwendung eines Stack-Pufferüberlaufs[2] sind Angreifer in der Lage, den Stack mit willkürlichem Code zu überschreiben. Um diesen Code auf dem Prozessor ausführen zu können und die DEP zu überlisten, muss der Speicherbereich des Stacks ausführbar gemacht werden. Sobald der Stack-Speicher ausführbar ist, ist es für Angreifer ein Leichtes, den Programmcode einzuschleusen und auszuführen.

Stack-based ROP Mitigation (Caller)

Um Sicherheitstechnologien wie Data Execution Prevention (DEP) und Address Space Layout Randomization (ASLR) zu überlisten, übernehmen Angreifer in der Regel die Kontrolle über den Control-Flow anfälliger Anwendungen, die im Kontakt mit dem Internet stehen. Diese speicherbasierten Angriffe sind für Antivirus-Software, die meisten „Next-Gen"-Produkte und andere Cyber-Abwehrmaßnahmen unsichtbar, da keine schädlichen Dateien zum Einsatz kommen. Stattdessen wird der Angriff während der Laufzeit erstellt. Hierzu werden kurze Abschnitte von harmlosem Code kombiniert, die Teil bestehender Anwendungen wie Internet Explorer oder Adobe Flash Player sind. Man spricht auch von einem Code-Wiederverwendungs- oder „Return-Oriented Programming (ROP)"-Angriff.

Im Rahmen des normalen Control-Flows werden sensible API-Funktionen wie VirtualAlloc und CreateProcess von der CALL-Anweisung aufgerufen. Beim Aufruf einer sensiblen API stoppen klassische ROP-Abwehrfunktionen die Code-Ausführung und ermitteln, welche Adresse die API aufruft – unter Verwendung der „Absender"-Adresse, die sich ganz oben auf dem Stack befindet. Wenn die Anweisung der Adresse, die die API aufruft, kein Call ist, wird der Prozess beendet.

Da die Inhalte des Stacks beschreibbar sind, kann ein Angreifer spezifische Werte auf den Stack schreiben, um die Analyse der Stack-basierten ROP-Abwehr in die Irre zu führen. Die Stack-basierte ROP-Abwehr kann nicht feststellen, ob die Inhalte des Stacks unbedenklich sind oder von einem Angreifer manipuliert wurden.

Branch-based ROP Mitigation (Hardware Augmented Control-Flow Integrity)

Wie bereits erläutert, sind Stack-basierte Abwehrmaßnahmen gegen Return-Oriented Programming (ROP) grobkörnig und manipulationsanfällig. Um dies zu verbessern, benötigen Sicherheitsverantwortliche feinkörnigere und manipulationsresistentere Daten zur Analyse während der Laufzeit.

Sophos Intercept X führt Hardware Augmented Control-Flow Integrity (CFI) ein und nutzt hierfür eine ungenutzte Hardware-Funktion in Mainstream-Intel®-Prozessoren (ab 2008 und neuer). Die Prozessor-Hardware selbst bietet schreibgeschützte Daten, um die Erkennung komplexer Exploit-Angriffe während der Laufzeit zu verbessern. Die Anwendung Hardware-nachverfolgter (Verzweigungs-)Datensätze hat vom Sicherheitsstandpunkt betrachtet einen wesentlichen Vorteil gegenüber Stack-basierten Ansätzen. Die Verzweigungsinformationen, die aus diesen Datensätzen gewonnen werden können, identifizieren nicht nur das Ziel der Verzweigung, sondern auch die Quelle. Sie geben also Aufschluss darüber, welchen Ursprung die Änderung im Control-Flow hat. Diese spezifischen Informationen können nicht mit vergleichbarer Zuverlässigkeit über eine Stack-basierte Lösung wie Microsoft EMET oder Palo Alto Networks Traps bezogen werden.

Verzweigungsinformationen in den Hardware-verfolgten Datensätzen können nicht manipuliert werden; es besteht keine Möglichkeit, sie mit kontrollierten Daten eines Angreifers zu überschreiben. Stack-basierte Ansätze stützen sich auf Stack-Daten, die sich – insbesondere im Falle eines ROP-Angriffs – in der Kontrolle des Angreifers befinden, der wiederum den Verteidiger in die Irre führen kann. Die von Sophos Intercept X genutzten Hardware-nachverfolgten Daten sind zuverlässiger und manipulationssicherer.

2 https://en.wikipedia.org/wiki/Stack_buffer_overflow

Eine alternative Hardware-assisted Control-Flow Integrity Implementation (HA-CFI) von Endgame basiert auf dem Trainieren von regulärem Control-Flow und kann Abweichungen von dem vom Programmierer vorgesehenen Codepfad erkennen. Dieses Modell muss zum Aufbau einer Whitelist gültiger Code-Pointer-Adressen, die alle möglichen Funktionen und Versionen der geschützten Anwendung wiedergibt, kontinuierlich trainiert werden. Sophos Intercept X benötigt kein solches Training und funktioniert auch bei Thread-Kontextänderungen und dynamischer Frequenzskalierung.

Sophos Intercept X wendet automatisch Hardware Augmented Control-Flow Tracing an, wenn es einen Intel® Core™ i3-, i5-, oder i7-Prozessor (CPU) erkennt. Wird keine unterstützte Prozessor-Hardware erkannt, greift Sophos Intercept X automatisch auf Stack-basierte Integritätsprüfungen auf reiner Software-Basis zurück.

Sophos Intercept X nutzt Hardware-nachverfolgte Datensätze nicht nur, um die ROP-Erkennung zu optimieren, sondern auch zum Import Address Filtering (IAF), mit dem die Importadresstabelle geschützter Anwendungen geschützt wird.

Hinweis: Die Patches zur Behebung der Spectre-Schwachstellen in Bezug auf den Verzweigungsprädiktor innerhalb von Intel CPU Hardware haben keinen Einfluss auf die korrekte Funktionsweise von Sophos Intercept X.

Structured Exception Handler Overwrite Protection (SEHOP)

Ein Angreifer kann den Handler Pointer eines Ausnahmedatensatzes auf dem Stack mit einem kontrollierten Wert überschreiben. Sobald eine Ausnahme eintritt, durchläuft das Betriebssystem die Kette der Ausnahmedatensätze und ruft alle Handler in jedem Ausnahmedatensatz auf.

Da der Angreifer einen der Datensätze steuert, springt das Betriebssystem überall dorthin, wo der Angreifer möchte, und verschafft dem Angreifer damit die Kontrolle über den Ablauf der Ausführung.

SEHOP ist eine Opt-in-Option für Windows Vista und höher, die vom Software-Anbieter beim Kompilieren einer Anwendung aktiviert werden muss. Es gibt Angriffe zum Umgehen vom integriertem SEHOP-Schutz. Es ist daher nicht anzuraten, sich auf die Betriebssystemimplementierung zu verlassen.

Import Address Table Access Filtering (IAF)

Um Schadaktivitäten ausführen zu können, braucht ein Angreifer früher oder später die Adressen bestimmter Systemfunktionen (z. B. kernel32!VirtualProtect).

Diese Adressen können von verschiedenen Quellen abgerufen werden, beispielsweise von der Importadresstabelle (IAT) eines geladenen Moduls. Die IAT fungiert als eine Nachschlagetabelle, wenn eine Anwendung eine Funktion in einem anderen Modul aufruft. Da ein kompiliertes Programm den Speicherort der Libraries, auf die es sich stützt, nicht kennen kann, ist bei jedem API-Aufruf ein indirekter Sprung erforderlich. Der dynamische Linker lädt Module und fügt diese zusammen. Dabei schreibt er echte Adressen in die IAT Slots, sodass diese zu den Speicherorten der entsprechenden Library-Funktionen verweisen.

Sophos Intercept X führt Hardware Augmented Import Address Table Access Filtering ein und nutzt hierfür Hardware-Funktionen in Mainstream-Intel®-Prozessoren (ab 2008 und neuer). Neben den Hardware-nachverfolgten Verzweigungsdatensätzen zur Durchsetzung der Control-Flow Inregrity nutzt Intercept X auch die Vorhersage der Hardware-Verzweigung, um den Schutz der Importadresstabelle weiter zu verbessern.

Hinweis: Die Patches zur Behebung der Spectre-Schwachstellen in Bezug auf den Verzweigungsprädiktor innerhalb von Intel CPU Hardware haben keinen Einfluss auf die korrekte Funktionsweise von Sophos Intercept X.

Load Library

Angreifer können versuchen, schädliche Libraries zu laden, indem sie diese auf UNC-Pfaden platzieren. Mittels Überwachung aller Aufrufe der LoadLibrary API kann diese Art von Library Loading unterbunden werden.

Reflective DLL Injection

Wenn Sie eine DLL in Windows laden, rufen Sie normalerweise die API Funktion LoadLibrary auf. Die LoadLibrary verwendet den Dateipfad der DLL als Eingabe und lädt diese in den Speicher.

Unter Reflective DLL Loading versteht man das Laden einer DLL aus dem Speicher anstelle von der Festplatte. Windows hat keine LoadLibrary-Funktion, die diesen Vorgang unterstützt. Um die Funktion zu nutzen, müssen Sie sie also selbst schreiben. Wenn Sie Ihre eigene Funktion schreiben, besteht ein Vorteil darin, dass Sie auf einige Dinge verzichten können, die bei Windows Standard sind, z. B. Registrieren der DLL als ein geladenes Modul im Prozess, wodurch sich der Reflektive Loader schwieriger analysieren lässt. Das Tool Meterpreter nutzt beispielsweise Reflective Loading, um sich zu verstecken. Als Abwehrmaßnahme wird analysiert, ob eine DLL innerhalb des Speichers reflektierend geladen wird.

Shellcode

Bei Shellcode handelt es sich um einen kleinen Computercode-Abschnitt, der im Rahmen der Ausnutzung einer Software-Schwachstelle als Payload verwendet wird. Der Name „Shellcode" kommt daher, dass ursprünglich eine Command Shell gestartet wurde, über die der Angreifer das kompromittierte System kontrollieren kann. Allerdings kann jeder Codeabschnitt, der eine ähnliche Aktion ausführt, als Shellcode bezeichnet werden.

Ein Exploit schleust in der Regel einen Shellcode in den Zielprozess ein, bevor oder während er eine Schwachstelle ausnutzt, um die Kontrolle über den Processor Instruction Pointer (EIP/RIP) zu erlangen. Der Instruction Pointer wird angepasst, sodass er auf den Shellcode zeigt. Anschließend wird er ausgeführt und führt seine Aufgabe aus.

VBScript God Mode

Unter Windows kann VBScript in Browsern oder der lokalen Shell verwendet werden. Beim Einsatz im Browser sind die Fähigkeiten von VBScript aus Sicherheitsgründen beschränkt. Diese Beschränkung wird über das Safemode Flag gesteuert. Wird das Flag modifiziert, kann VBScript in HTML genauso wie in der lokalen Shell agieren. Folglich können Angreifer problemlos Schadcode in VBScript schreiben. Das Manipulieren des Safemode Flags auf VBScript im Web-Browser wird als God Mode bezeichnet[3].

Ein Angreifer kann beispielsweise den Wert des Safemode Flags modifizieren, indem er die Schwachstelle CVE-2014-6332[4] ausnutzt, einen durch unsachgemäße Handhabung bei der Größenanpassung eines Arrays in der Internet Explorer VBScript Engine verursachten Bug. Im God Mode kann beliebiger, in VBScript geschriebener Code aus der Browser-Sandbox ausbrechen. Dank Gode Mode greifen Schutzmaßnahmen wie Data Execution Prevention (DEP), Address Space Layout Randomization (ASLR) und Control-Flow Guard (CFG) nicht.

WoW64

Microsoft bietet Abwärtskompatibilität für 32-Bit-Software auf 64-Bit-Editionen von Windows durch den „Windows on Windows" (WoW) Layer. Bestimmte Aspekte der WoW-Implementierung liefern Angreifern interessante Methoden zum Verkomplizieren dynamischer Analysen, Entpacken von Binärdateien und zum Umgehen von Exploit-Abwehrmaßnahmen.

Das Verhalten einer 32-Bit-Anwendung in der WoW64-Umgebung unterscheidet sich in vielen Punkten von einem echten 32-Bit-System. Die Möglichkeit, während der Laufzeit zwischen Ausführungsmodi umzuschalten, kann Angreifern Methoden für Angriffe, zur Verschleierung und Anti-Emulation liefern, z. B.:

- Zusätzliche ROP Gadgets, die im 32-Bit-Code nicht vorhanden sind
- Mixed Execution Mode Payload Encoder
- Ausführungsumgebungsfunktionen, die Abwehrmaßnahmen weniger effektiv machen können
- Umgehen von Hooks, die von Sicherheitssoftware eingerichtet wurden (nur 32-Bit-Benutzerbereich)

3 https://en.wikipedia.org/wiki/Glossary_of_video_game_terms#God_mode
4 https://www.rapid7.com/db/modules/exploit/windows/browser/ms14_064_ole_code_execution

Endpoint-Protection-Software schützt meist nur sensible API-Funktionen im 32-Bit-Benutzer-Speicherbereich, wenn ein Prozess unter WoW64 ausgeführt wird. Falls ein Angreifer in der Lage ist, in den 64-Bit-Modus umzuschalten, kann er sich Zugriff auf ungeschützte 64-Bit-Versionen sensibler API-Funktionen verschaffen, die im 32-Bit-Modus gehookt sind.

In 64-Bit-Editionen von Windows verbietet Sophos Intercept X dem Programmcode, direkt vom 32-Bit- in den 64-Bit-Modus umzuschalten (z. B. unter Verwendung von ROP), ermöglicht dem WoW64 Layer jedoch weiterhin, diese Umstellung durchzuführen.

Weitere Informationen über die missbräuchliche Nutzung von WoW64 finden Sie in diesem Forschungsbeitrag von Duo Security: „WoW64 and So Can You[5] and Mitigating Wow64 Exploit Attacks[6]".

Syscall

Ein Syscall (oder System Call) ist ein programmgesteuerter Vorgang, bei dem ein Computerprogramm einen Service vom Kernel des Betriebssystems anfordert. Hierzu zählen Hardware-Services für den Zugriff auf die lokale Festplatte und für die Erstellung und Ausführung neuer Prozesse.

In der Regel verfügt das Betriebssystem über eine allgemeine Anwendungsprogrammierschnittstelle (API), die zwischen normalen Programmen und dem Betriebssystem angesiedelt ist. Unter normalen Bedingungen ruft eine Anwendung immer eine API auf, um eine spezifische Aufgabe vom Kernel anzufordern. Sicherheitssoftware platziert Hooks an sensiblen API-Funktionen, um Abfangmaßnahmen und Prüfungen wie Antivirus-Scans durchzuführen, bevor der Kernel die Anfrage bearbeiten darf.

Ein Angreifer kann sich zunutze machen, dass:

- nicht alle API-Funktionen von Sicherheitssoftware gehookt sind, sondern nur sensible API-Funktionen
- die zum Aufruf von Kernel-Funktionen genutzten Stubs sehr ähnlich sind, nur der Funktionsindex ist einmalig

Durch den Aufruf eines nicht überwachten, nicht sensiblen Function Stubs (Hilfscode) auf nicht standardisierte Weise und an einer bestimmten Position, die erforderlich ist, um die Aktion auszuführen (und um bewusst einen sensiblen Kernel-Service zu adressieren), kann ein Angreifer die meisten Sicherheitssoftware- und Sandbox-Analysen umgehen.

Sophos Intercept X wendet ein neues Verfahren an, mit dem Angreifer daran gehindert werden, sensible und ungeschützte Kernel-Funktionen für ihre Zwecke zu missbrauchen.

Nähere Informationen über die missbräuchliche Nutzung von Syscalls finden Sie in dem BreakDev.org-Blogeintrag „Defeating Antivirus Real-time Protection From The Inside"[7].

Process Hollowing

Process Hollowing ist eine Technik, bei der eine vertrauenswürdige Anwendung (z. B. explorer.exe oder svchost.exe) ausschließlich auf dem System geladen wird, um als Container für bösartigen Code zu dienen.
Ein Hollow Process wird üblicherweise in einem angehaltenen Zustand erstellt. Seine Speicherzuordnung wird anschließend aufgehoben und durch Schadcode ersetzt. Ähnlich wie bei Code Injection wird die Ausführung von schädlichem Code als legitimer Vorgang getarnt und kann so ggf. Abwehrmaßnahmen und Erkennungsanalysen täuschen.

5 https://duo.com/blog/wow64-and-so-can-you
6 https://hitmanpro.wordpress.com/2015/11/10/mitigating-wow64-exploit-attacks
7 https://breakdev.org/defeating-antivirus-real-time-protection-from-the-inside

Process Doppelgänging

Die meisten Windows-Computer nutzen das NTFS-Dateisystem. 2007 führte Microsoft eine neue Funktion namens Transactional NTFS (TxF) ein. Mit dieser Funktion können mehrere Dateioperationen als ein Ganzes gehandhabt haben: Diese können entweder in ihrer Gänze erfolgreich sein und eingecheckt werden oder in ihrer Gesamtheit fehlschlagen und rückgängig gemacht werden. So ist eine Anwendung in der Lage, viele Änderungen an diversen Dateien auf der Festplatte vorzunehmen und alle Dateien in ihren Ursprungszustand zurückzuversetzen, falls ein Fehler erkannt wird.

Am häufigsten kommt TxF bei Installationen von WIndows-Updates zum Einsatz.

Beim Process Doppelgänging wird der TxF-Mechanismus ausgenutzt, um Malware zu verbergen. Er wählt eine harmlose Datei aus, überschreibt diese und führt die Malware über eine Low-Level-API aus, um beispielsweise eine vertrauenswürdige Datei zu imitieren (ähnlich wie Process Hollowing). Kurz bevor die Malware ausgeführt werden kann, werden alle Änderungen abgelehnt oder rückgängig gemacht, sodass Antivirus-Software den tatsächlich ausgeführten Dateiinhalt nicht scannen kann. Beim Öffnen enthält die Datei auf der Festplatte keine verdächtigen Inhalte. Zudem kann es sich bei der Datei um eine bekannte, digital signierte Anwendung handeln.

DLL Hijacking

Aufgrund einer Schwachstelle, die oft als DLL Hijacking, DLL Spoofing, DLL Preloading oder Binary Planting bezeichnet wird, können viele Programme dazu gebracht werden, eine schädliche DLL auszuführen, die sich im selben Ordner befindet wie andere von diesen Programmen geöffnete Dateien.

Dynamic Data Exchange (DDE)

Windows Dynamic Data Exchange (DDE) ist ein Client-Server-Protokoll zur Inter-Process Communication (IPC) zwischen Anwendungen. Angreifer können DDE zum Ausführen beliebiger Befehle verwenden. So können beispielsweise Microsoft-Office-Dokumente mit DDEAUTO-Befehlen manipuliert und dazu missbraucht werden, PowerShell-Befehle über Spear-Phishing-Kampagnen oder gehostete Web-Inhalte auszuführen. Die Nutzung von Visual Basic for Applications (VBA) Makros wird in diesem Fall vermieden. Zudem ist es auch möglich, DDEAUTO-Befehle in den Nachrichtentext von E-Mails oder Meeting-Anfragen einzubetten, die bei ihrer Beantwortung oder Annahme in Microsoft Outlook ausgeführt werden.

Dank der Gestaltung der Application Lockdown Mitigation unterbindet Sophos Intercept X auch grundsätzlich die Ausführung von Schadcode mittels Dynamic Data Exchange.

Application Lockdown

Für den Fall, dass es einem Angreifer gelingen sollte, alle Abwehrmaßnahmen auf Speicher- und Code-Ebene auszunutzen und auszuhebeln, beschränkt Sophos Intercept X die Möglichkeiten des Angreifers, Schaden anzurichten. Dieses Feature heißt Application Lockdown und soll verhindern, dass Angreifer unerwünschten Code einschleusen.

Application Lockdown stoppt Angriffe, die sich in der Regel nicht auf Software-Bugs in Anwendungen stützen. Ein solcher Angriff kann beispielsweise der Einsatz eines speziell entwickelten (schädlichen) Makros in einem Office-Dokument sein, das an eine (Spear-)Phishing-E-Mail angehängt wird.

Makros in Dokumenten sind potenziell gefährlich, weil sie in der Programmiersprache Visual Basic for Applications (VBA) erstellt werden. Diese ermöglicht das Herunterladen und Ausführen von Binärdateien aus dem Internet und erlaubt außerdem den Einsatz von PowerShell und anderen vertrauenswürdigen Anwendungen.

Diese unerwartete Funktion (oder „Logic-Flaw Exploit") bietet Angreifern einen offensichtlichen Vorteil, da sie keinen Software-Bug ausnutzen oder Abwehrmaßnahmen auf Code- und Speicherebene aushebeln müssen, um Computer zu infizieren. Sie müssen lediglich Standardfunktionen ausnutzen, die von einer vertrauenswürdigen, weit verbreiteten Anwendung angeboten werden, und das Opfer mittels Social Engineering überzeugen, das speziell für diesen Zweck erstellte Dokument zu öffnen.

Ohne dass eine Blacklist von Ordnern gepflegt werden muss, beendet Sophos Intercept X eine geschützte Anwendung automatisch auf Basis ihres Verhaltens: Wenn beispielsweise eine Office-Anwendung genutzt wird, um PowerShell zu starten, auf den WMI zuzugreifen und ein Makro zur Installation beliebiger Skripte oder zur Manipulation kritischer Systembereiche auszuführen, blockiert Sophos Intercept X diesen schädlichen Vorgang – selbst wenn der Angriff keinen untergeordneten Prozess erstellt.

Java Lockdown

Exploit-Kits spielten in der Vergangenheit bei Drive-by-Downloads von Malware eine wichtige Rolle.

Sie nutzten Schwachstellen in der Java Runtime Environment (JRE) aus und schleusten so Windows PE Payloads ein. JRE wird in Standard-Browsern als Plug-in oder Add-on geladen.

Sophos Intercept X verhindert, dass JRE Java-fremde Anwendungen ausführt. Sophos Intercept X beendet beispielsweise eine Java-Anwendung, wenn diese versucht, eine Windows PE Binary einzuschleusen und auszuführen. Zudem können Angreifer Java nicht zur Manipulation von Autostartorten (u. a. der Ordner „Autostart", Run, RunOnce und andere Registry-Schlüssel) nutzen.

Hinweis: Seit der Einführung von Java 8 Update 20 im Jahr 2014 ist die Sicherheitsstufe für Java-Anwendungen standardmäßig auf hoch eingestellt. So ist es für Angreifer zunehmend schwierig geworden, Java Exploits mit ausreichenden Berechtigungen auszuführen und so den Endpoint zu infizieren. Daher sind Java Exploits in Exploit-Kits nicht mehr besonders beliebt und die Java Lockdown Mitigation ist demzufolge ein wenig überholt.

Code Cave

Code Cave ist eine Technik, mit der Angreifer vermutlich seriöse Software so manipulieren, dass diese eine zusätzliche Anwendung enthalten. Diese zusätzliche Anwendung wird in einen sogenannten Code Cave eingefügt – einen Abschnitt der Datei der Zielanwendung, der vom Programm nicht genutzt wird. Code Caves existieren in den meisten Anwendungen und das Hinzufügen von Code zu diesen Abschnitten sollte das Verhalten der primären Anwendung nicht beeinträchtigen.

Oft handelt es sich bei dem in einen Code Cave eingefügten Ausführungscode einfach um ein Remote-Shell-Startprogramm oder eine Backdoor; diese können sehr klein sein und gewähren dem Angreifer Zugriff auf den Endpoint, wo sie weitere Aktionen ausführen können. Bei diesem Angriffstyp muss der Angreifer eine Präsenz auf dem Endpoint etabliert haben, damit er die Backdoor-Anwendung bereitstellen oder den Benutzer dazu bringen kann, eine Anwendung herunterzuladen und zu installieren, bei der der Code Cave bereits ausgenutzt wurde.

Angreifer nutzen Code Caves hauptsächlich, um unerkannt von Benutzern und Administratoren zu bleiben. Die erwartete Anwendung funktioniert nach wie vor, aber die eingefügte Anwendung wird ebenfalls ausgeführt.

Wenn es sich bei der modifizierten Anwendung um ein seriöses Business Tool handelt, das der Administrator auf dem Gerät erwartet, stuft er diese mit geringerer Wahrscheinlichkeit als Malware ein, falls die traditionelle Antivirus-Software ein Problem erkennt. Unter Umständen setzen Administratoren die Anwendung einfach auf die Ausnahmeliste, weil sie davon ausgehen, dass die Antivirus-Engine eine False Positive generiert hat. So kann der Angreifer auf dem Endpoint Fuß fassen und den Administrator ggf. sogar dazu bringen, seine eingeschleuste Anwendung auszuführen.

Mit einem sogenannten Supply-Chain-Angriff können sich Angreifer zudem Zugriff auf die Software-Update-Server verschaffen und ein Update mit Schadcode versehen, das Kunden dann unbemerkt mit Ransomware oder Wiper Malware infiziert.

Sophos Intercept X blockiert die Ausführung von Backdoor-Anwendungen automatisch. Sogar der hinzugefügte Shellcode wird erkannt, wenn die Code-Ausführung nicht zu einem Code Cave oder zu einem hinzugefügten Abschnitt in der infizierten PE-Datei fließt. Sophos Intercept X bietet umfassenden Schutz vor Shellcode Injection Tools wie Shellter und Backdoor Factory.

Process Migration – Remote Reflective DLL Injection

Eine Prozessmigration wird von Angreifern häufig verwendet, wenn diese ihre Präsenz auf einem Gerät etablieren und zu einem anderen Prozess wechseln möchten, um mehr Berechtigungen oder dauerhaften Zugriff zu erhalten. Der Angreifer möchte nicht die Kontrolle verlieren, wenn der Enduser den Browser schließt oder den kompromittierten Prozess beendet. Deshalb wird die Migration zu einem Systemprozess angestrebt.

Ein Remote-Reflective-DLL-Angriff ist mit einer Prozessmigration vergleichbar, lässt sich jedoch schwerer bekämpfen. Der Angreifer hat bereits einen Prozess kompromittiert und manipuliert von dieser Position einen anderen Prozess, um DLLs zu laden und willkürlichen Code auszuführen.

Local Privilege Escalation (LPE)

Sophos Intercept X verhindert, dass ein Prozess mit geringfügigen Berechtigungen seine Berechtigungen ausweitet, indem er einen Token von einem Prozess mit höheren Rechten stiehlt. Diese Technik kommt oft in Kombination mit einer anderen Schwachstelle zum Einsatz, damit ein Angreifer seinen Schadecode mit Systemberechtigungen erfolgreich einschleusen und ausführen kann.

DoublePulsar Code Injection

DoublePulsar ist ein Implant Tool für Backdoor-Programme, das ursprünglich von der Equation Group der US-amerikanischen National Security Agency (NSA) entwickelt und von The Shadow Brokers Anfang 2017 geleakt wurde. Das Implant beinhaltet eine neuartige Injection-Technik, die Teil etlicher NSA Exploits ist, u. a. EternalBlue und EternalRomance. Diese Exploits wurden auch für die sich selbst ausbreitenden Wurmkomponenten bei den WannaCry- und NotPetya-Angriffen genutzt.

Die DoublePulsar-Code-Injection-Technik nutzt einen Asynchronous Procedure Call (APC), um willkürlichen Code (Shellcode) innerhalb eines regulären vertrauenswürdigen Prozesses auszuführen. Sophos Intercept X vereitelt die von DoublePulsar verwendete Methode grundsätzlich und stoppt damit auch Angriffe, die sich zur Code Injection auf die gleiche Technik stützen.

AtomBombing Code Injection

Bei einer Asynchronous Procedure Call (APC) Injection wird Schadcode an die APC-Warteschlange eines Prozess-Threads angefügt. APC-Funktionen in der Warteschlange werden ausgeführt, wenn die Bedrohung in einen veränderbaren Status eintritt. AtomBombing ist eine Variation, die APCs nutzt, um schädlichen Code aufzurufen, der zuvor auf die globale Atomtabelle geschrieben wurde.

DoubleAgent Code Injection

Der DoubleAgent nutzt das seriöse Windows-Tool „Microsoft Application Verifier" aus. Dieses Tool ist in allen Versionen von Microsoft Windows enthalten und wird als Laufzeitverifizierungstool verwendet, um Bugs in Anwendungen zu erkennen und zu beheben. Der Application Verifier kann so eingerichtet werden, dass er jede beliebige Library von der Festplatte lädt. So kann eine schädliche Library geladen werden, die die Berechtigungen des Opfer-Prozesses erhält.

DoubleAgent gilt allgemein als Schwachstelle und Zero-Day-Angriff auf Antivirus-Produkte, tatsächlich besteht jedoch die wesentliche Aufgabe des Application Verifiers darin, willkürlichen Code in eine beliebige Anwendung zu laden (einschl. vertrauenswürdiger Produktivitäts- und Windows-Prozesse).

Sophos Intercept X unterbindet Code Injection mittels Application-Verifier-Ausnutzung.

Features von Intercept X

Funktionen	
EXPLOIT PREVENTION	
Enforce Data Execution Prevention	✓
Mandatory Address Space Layout Randomization	✓
Bottom-up ASLR	✓
Null Page (Null Deference Protection)	✓
Heap Spray Allocation	✓
Dynamic Heap Spray	✓
Stack Pivot	✓
Stack Exec (MemProt)	✓
Stack-based ROP Mitigations (Caller)	✓
Branch-based ROP Mitigations (Hardware Assisted)	✓
Structured Exception Handler Overwrite (SEHOP)	✓
Import Address Table Filtering (IAF)	✓
Load Library	✓
Reflective DLL Injection	✓
Shellcode	✓
VBScript God Mode	✓
Wow64	✓
Syscall	✓
Hollow Process	✓
DLL Hijacking	✓
Squiblydoo Applocker Bypass	✓
APC Protection (Double Pulsar/AtomBombing)	✓
Process Privilege Escalation	✓
ACTIVE ADVERSARY MITIGATIONS	
Credential Theft Protection	✓
Code Cave Mitigation	✓
Man-in-the-Browser Protection (Safe Browsing)	✓
Malicious Traffic Detection	✓
Meterpreter Shell Detection	✓

Funktionen	
ANTI-RANSOMWARE PREVENTION	
Ransomware File Protection (CryptoGuard)	✓
Automatic File Recovery (CryptoGuard)	✓
Disk and Boot Record Protection (WipeGuard)	✓
APPLICATION LOCKDOWN	
Web-Browser (einschl. HTA)	✓
Web-Browser-Plugins	✓
Java	✓
Media-Anwendungen	✓
Office-Anwendungen	✓
DEEP LEARNING	
„Deep Learning"-Malware-Erkennung	✓
Deep Learning Potentially Unwanted Applications (PUA) Blocking	✓
False Positive Suppression	✓
Live Protection	✓
REAKTION, ANALYSE, BESEITIGUNG	
Ursachenanalyse	✓
Sophos Clean	✓
Synchronized Security Heartbeat	✓
BEREITSTELLUNG	
Kann als Standalone-Agent ausgeführt werden	✓
Kann mit bestehendem Antivirus ausgeführt werden	✓
Kann als Komponente von bestehendem Sophos Endpoint Agent ausgeführt werden	✓
Windows 7	✓
Windows 8	✓
Windows 8.1	✓
Windows 10	✓
macOS*	✓

* Unterstützte Funktionen: CryptoGuard, Malicious Traffic Detection, Synchronized Security Heartbeat, Root Cause Analysis

Testen Sie Sophos Intercept X kostenfrei

unter www.sophos.de/intercept-x

Sales DACH (Deutschland, Österreich, Schweiz)
Tel.: +49 611 5858 0 | +49 721 255 16 0
E-Mail: sales@sophos.de

18-03-06 WP-DE (DD)

SOPHOS