Erklärung der Bedrohungen

Verständliche Erklärungen der Angriffstechniken, gegen die Cyber Crucible entwickelt wurde – dateilose Malware, In-Memory-Angriffe, Hacker-Automatisierung, Infostealer und Diebstahl von Sitzungs-Token.

Kann Cyber Crucible die laterale Bewegung von Angreifern stoppen?

Die Verhaltensanalyse-Engine des Produkts und die Zero-Trust-Methoden sammeln effektiv Nachweise für laterale Bewegung.

Dies können Nachweise für laterale Bewegung von Prozess zu Prozess auf dem Rechner selbst sein, oder es kann sich um die Entdeckung des Einstiegspunkts eines Angreifers in das System von einem entfernten System aus handeln, im Rahmen einer Root-Cause-Analyse.

Programme und ihre Argumente, die mit Methoden zur Prozessinjektion und mit „Living off the Land“-Prozessen verbunden sind, die andere Prozesse öffnen, werden erfasst und an Cyber Crucible übermittelt.

Aufgrund der kernelbasierten Verhaltensanalyse von Prozessen gibt es keine Prozesse, die für die Beobachtung und Berichterstattung durch Cyber Crucible zu privilegiert sind (einschließlich anderer Kernel-Treiber oder sogar anderer Endpoint-Security-Tools).

Automatisierte Gegenmaßnahmen gegen Erpressung erfolgen, sobald entweder Datenerpressung oder Ransomware-Verschlüsselungsaktionen begonnen werden.

Hier ist beispielsweise ein Video, das laterale Bewegung über einen Rechner hinweg zeigt, unter Verwendung von In-Memory-Techniken und dem Log4J-Exploit. Bitte beachten Sie, dass die automatisierte Suspendierung der betroffenen Prozesse erfolgte, nachdem Datendiebstahl- und Ransomware-Verhalten begonnen hatte.

rr-log4j-procinj-3mins.mp4

Ein weiteres Beispiel dieser Dynamik zeigt die Nutzung durch einen Angreifer, um eine Windows-Shell zu öffnen. Die Beobachtung erfasste alle Nachweise, aber die Aktivität wurde suspendiert, sobald die Windows-Shell versuchte, Datenerpressungsaktivitäten durchzuführen (in diesem Fall Datendiebstahl).

rr-com-dll.mp4

Schließlich wird auch die Migration von einem kompromittierten Browser aus beobachtet, und in diesem Fall gelang es dem Angreifer aufgrund der Überwachung des Speicherzustands des Browsers nie, sich erfolgreich vom kompromittierten Chrome-Browser aus zu bewegen:

rr-untrusted-chrome.mp4

Funktioniert Application Whitelisting?

Was ist Application Whitelisting?

Application Whitelisting ist ein Konzept, bei dem eine Liste von Programmen existiert, die ausgeführt werden dürfen, während alles andere nicht funktionieren darf.

Die fortschrittlichsten Application-Whitelist-Technologien kombinieren eine Art Quell- und Ziellogik. Zum Beispiel könnte eine Application Whitelist besagen: „Windows Notepad darf auf den Ordner Steuern auf meinem Desktop zugreifen, Microsoft Word jedoch nicht.“

Wir werden dies weiter untersuchen, aber generell verlassen sich Verteidigungsstrategien mittels Application Whitelisting stark auf diese beiden Variablen:

1. Die Angreifer stimmen zu, sich an die Regeln zu halten, die Sie im Hinblick auf Ihr Application-Whitelisting-Cybersicherheitstool festgelegt haben.
2. Ihre Umgebung ist sehr klein, mit wenigen Anwendungen, die relativ stabil bleiben und keine Exploits aufweisen.

Nachteile von Application-Whitelisting-Technologien

Komplexität

IT-Netzwerke sind wie komplexe Organismen

Selbst kleine IT-Netzwerke sind sich ständig verändernde Gebilde mit einer Vielzahl von Verhaltensweisen. Anwendungen werden von Benutzern ständig hinzugefügt und entfernt. Module innerhalb von Anwendungen werden von den Programmherstellern ständig hinzugefügt und entfernt, insbesondere bei Updates. Schließlich werden Anwendungsverhalten – zusätzlich zu Codeänderungen, die zu Verhaltensänderungen führen – von verschiedenen Benutzern zu

unterschiedlichen Zeiten auf unterschiedliche Weise genutzt. Stellen Sie sich nun das Team vor, das diese ständige Verfolgung unterstützen müsste, sowie die IT-Helpdesk-Tickets, die dabei fortlaufend eingehen würden.

Realistisch betrachtet ist die einzige stabile IT-Umgebung eine, die durch Kioske angetrieben wird, bei denen Programme installiert werden, ohne dass sie aktualisiert, hinzugefügt oder entfernt werden können, und bei denen Benutzer nur eng begrenzt eine oder zwei Funktionen ausführen dürfen. Vielleicht ein Kiosk für den Fotodruck oder Ähnliches. In diesem Fall müsste die Application Whitelist nur jedes Mal aktualisiert werden, wenn das Kiosk-Terminal zum Beispiel zweimal jährlich ein Upgrade erhält.

Fortschrittlichere Application Whitelists bedeuten mehr Komplexität

Die einfachsten Application Whitelists bestehen lediglich aus Listen von Programmen, die im System eines Benutzers existieren dürfen.

Dies allein kann bereits eine Herausforderung im Management darstellen, führt aber zu einer Vielzahl möglicher Erweiterungen.

Zum Beispiel: Nur weil die Personalabteilung (Payroll) eine Anwendung zur Interaktion mit dem Gehaltsabrechnungssystem des Unternehmens hat, sollte es dann nicht ein separates Regelwerk für das Lagerteam geben, das andere (nicht mit der Gehaltsabrechnung zusammenhängende) Anwendungen nutzt?

Erweiterungen dahingehend, welche Abteilungen oder Benutzer welche Anwendungen ausführen dürfen, sind also eine naheliegende Erweiterung. Eine Erweiterung, die nun zusätzliche Komplexität schafft, zusätzlich zur bloßen Verfolgung der Anzahl, Art und Versionen der Programme in Ihrem Netzwerk.

Nehmen wir nun an, wir benötigen einen weiteren Satz von Erweiterungen – wir müssen entscheiden, welcher Benutzer Zugriff auf welche Anwendung für welchen Datensatz hat.

Das ergibt ebenfalls vollkommen Sinn, nicht wahr? Sie möchten nicht, dass Ihr Lagerteam Zugriff auf die HR-Daten hat, und das HR-Team benötigt vermutlich keinen Zugriff auf die Betriebsdaten des Lieferteams im Lager.

Das ist eine geradezu absurd komplexe Menge an Kombinationen, die es zu verfolgen gilt, jetzt, da Datenspeicherorte als zusätzliche Variable hinzugekommen sind.

Mit jeder notwendigen Erweiterung, um Application-Whitelisting-Technologien wirksam zu machen, wird also die Verwaltung, Konfiguration und der laufende (man könnte sagen nie endende, stets eskalierende) Support, der zur Aufrechterhaltung der Funktionsfähigkeit erforderlich ist, die

Technologie unbrauchbar machen. Das Ergebnis ist normalerweise, dass großen Benutzergruppen Zugriff auf große Gruppen von Anwendungen gewährt wird – und wahrscheinlich auf nahezu alle Daten. Damit entsteht eine nahezu nutzlose Konfiguration der Technologie, da kein Unternehmen über die Ressourcen verfügt, um sie ordnungsgemäß zu verwalten.

Anwendungen interagieren miteinander

In modernen Betriebssystemen ist es üblich, dass Anwendungen mit anderen Anwendungen interagieren. Manchmal beruht diese Interaktion auf einer Integration zwischen zwei Anwendungen, die das Benutzererlebnis verbessert – etwa wenn ein Word-Dokument Microsoft Photos öffnet. In anderen Fällen (dies kommt sehr häufig vor) gibt es Aufrufe zwischen Programmen, die auf eine Weise erfolgen, die für den Endbenutzer in der Regel unsichtbar ist.

Was ist Application Whitelisting?

Einführung

Application Whitelisting bedeutet im Kontext von Cyber Crucible, einer Anwendung die Erlaubnis zu erteilen, eine Aktion auszuführen, die andernfalls als verdächtig oder bösartig eingestuft werden könnte.

Application-Whitelisting-Technologie findet sich typischerweise in defensiven Fachpraktiken der Verhaltensanalyse, obwohl einige Angreifer sie ebenfalls (hier nicht behandelt) in ihren Operationen einsetzen.

Man kann es sich vorstellen wie in der Schule: Normalerweise durften Schüler während des Unterrichts nicht durch die Flure laufen. Wenn ihnen jedoch von einer Autoritätsperson eine Art „Hallenausweis“ ausgestellt wurde, wurden sie nicht in ihr Klassenzimmer zurückgebracht oder ins Büro des Schulleiters geschickt.

Arten von Application Whitelisting - eine nicht-technische Beschreibung

Bleiben wir bei der Analogie des „Hallenausweises“, so gibt es verschiedene Techniken, die Schulverwaltungen im Laufe der Zeit entwickelt haben, um die Verhaltensweisen zu formalisieren, mit denen manche Schüler (ganz bestimmt nicht unser CEO Dennis) davonzukommen versuchten. All diese Techniken haben eine direkte Entsprechung im Sicherheitsbetrieb.

Der offensichtliche & umständliche Hallenausweis (eine gute Sache)

21987331.jpg?width=278
Erinnern Sie sich noch an einen Hallenausweis oder einen Schlüssel zu einem Raum, der unglaublich groß und auffällig war?

Dafür gab es ein paar gute Gründe:

1. Damit er nicht verloren ging.
2. Um den Zeit- und Energieaufwand einer Autoritätsperson bei der schnellen Überprüfung der genehmigten Aktivitäten des Schülers zu verringern.

Gute Software „verliert“ keine Whitelist, aber eine Application-Whitelist-Software sollte eine schnelle, ressourceneffiziente Überprüfung ermöglichen, dass die Anwendung geprüft wurde.

Der generische Hallenausweis (wahrscheinlich keine gute Sache)

Dies ist wahrscheinlich sowohl für die Anwendungsüberwachung als auch für Schulen am wenigsten nützlich, erfordert aber auch das geringste Management oder Fachwissen.

In diesem Fall darf der Schüler seine Besorgung im Flur erledigen. Der Lehrer weiß vielleicht Bescheid, aber es gibt keine Überprüfung durch Aufsichtspersonen, was der Schüler tatsächlich tut. Selbst wenn eine Überprüfung versucht wird, entstehen zwei Ebenen erheblichen Ressourcenaufwands:

1. der Aufwand, den Schüler anzusprechen und zu seinem Verhalten und seiner Absicht zu befragen
2. der Aufwand, sicherzustellen, dass der Schüler ehrlich ist, um seine Erklärung zu überprüfen

Bei Anwendungen zeigt sich dies meist dann, wenn ein Programm zu einer Whitelist hinzugefügt wird, ohne dass eine weitere Überprüfung der Verhaltensweisen oder erlaubten Aktivitäten erfolgt.

Leider steht die Möglichkeit, Kontext zu den Verhaltensweisen einer Anwendung zu erhalten, in der Regel nicht zur Verfügung, wenn dies nicht extern durch ein anderes Framework nachverfolgt wird.

Der generisch-spezifische Hallenausweis

Nicht zu verwechseln mit „Los, Anwendung, mach, was du willst!“

Kehren wir zum Toilettenausweis zurück, denn er passt hier besonders gut. Angesichts der Sichtbarkeit des Ausweises ist es relativ einfach zu beurteilen, was der Schüler tun sollte. Grobe Verhaltensweisen, die nicht mit dem „Toilettengang“ übereinstimmen, lassen sich ohne großen Aufwand erkennen.

Application Whitelisting mit generischen Grenzen der Aktivitäten ist am häufigsten anzutreffen, obwohl diese Grenzen in vielen Fällen dem Äquivalent entsprechen, einem Schüler mit Toilettenausweis das Spielen auf dem Schulhof zu erlauben.

Der spezifisch-spezifische Hallenausweis

Nicht zu verwechseln mit dem „generisch-spezifischen“ Hallenausweis!

22642689.jpg?width=340
Während wir beim generisch-spezifischen Hallenausweis wissen, dass der Schüler die Toilette benutzt, wissen wir in einer großen Schule nicht:
1. Wer dem Schüler die Erlaubnis erteilt hat

2. Wo er gerade herkommt.

3. Welche Toilette die Autoritätsperson bei ihrer Genehmigung im Sinn hatte.

4. Wie lange der Schüler schon „unterwegs“ ist.

Dies erforderte mehr Arbeit am Autoritätsrahmen der Schule.

Um zum Application Whitelisting zurückzukehren: Die Verhaltensanalyse wird genauer, je mehr Variablen in eine Entscheidungsfindungsfähigkeit einfließen.

Wenn man außerdem einen ausgestellten Hallenausweis überprüft, muss man normalerweise keine Kette von 5 Lehrern zurückverfolgen, um zur Quelle zu gelangen. Man kann auch fast immer darauf

vertrauen, dass der Lehrer keine Art „schlechter Lehrer“ war, der Schülern falsches Verhalten beibringt. Application Whitelisting erfordert im Beispiel einer spezifisch-spezifischen Whitelist ein Rückwärtsarbeiten, um jede übergeordnete oder vorangehende Anwendung zu untersuchen, die die nächste Anwendung geöffnet hat. Zum Beispiel könnte ein Stück Malware ein legitimes Windows-Dienstprogramm geöffnet haben.

Die folgenden Fragen entsprechen ungefähr dem oben genannten „Schüler-Hallenausweis“:

1. Wofür wurde die Anwendung gestartet, oder was wurde ihr aufgetragen? Wir müssen das beabsichtigte Verhalten kennen.
2. Wer hat die Anwendung gestartet? Dies ist typischerweise ein anderes Programm.
3. Welche „übergeordneten“ oder „vorangehenden“ Programme führten zum Öffnen dieses Programms? Was taten oder beabsichtigten diese jeweils zu tun?

Die kompromittierte Anwendung

Erinnern Sie sich an verschiedene Science-Fiction- oder Horrorliteratur und -filme, in denen ein Betrüger auf irgendeine Weise die

22413315.jpg?width=226
Kontrolle über das Verhalten einer Figur übernommen hat? In Schulen besteht kaum die Gefahr, dass eine Lehrkraft von Außerirdischen übernommen wird.

Die Übernahme von Anwendungen ist eine beliebte Technik von Angreifern, denn wie in den Filmen bemerken die meisten Beobachter nichts Ungewöhnliches, bis es zu spät ist.

In diesem Fall übernimmt ein Angreifer ein laufendes Programm, das gerade in den Speicher geladen ist. Das vollständig intakte Programm im Dateisystem wird dabei ignoriert. Das Programm im Speicher wird bearbeitet, um den Code des Angreifers hinzuzufügen - ähnlich wie ein Außerirdischer im Körper eines Menschen. Der Angreifer führt seine kriminellen Handlungen aus, und wenn das Programm fertig ausgeführt wurde, ist der größte Teil der Beweise verschwunden. Wenn das legitime Programm erneut läuft und der Angreifer den Code nicht erneut bearbeitet, verhält es sich genau so, wie es die ursprünglichen legitimen Programmierer beabsichtigt hatten. Vielleicht wäre in diesem Fall ein Werwolf die bessere Analogie?

Eine Application-Whitelist muss daher zusätzlichen Aufwand betreiben, um sicherzustellen, dass das Programm sowie alle vorangehenden oder übergeordneten Programme intakt sind und nicht von Hackern bearbeitet wurden, während sie ausgeführt werden.

Zusammenfassung - Fragen, die bei der Untersuchung einer Application-Whitelist-Funktion gestellt werden sollten

Stellen Sie sicher, dass Sie die Grenzen dessen verstehen, was eine Whitelist erlaubt, wenn Sie mit Ihrem Technologieteam oder Anbieter über Application Whitelisting sprechen.

1. Untersucht und verfolgt diese Funktion, was ein Programm tun soll, und vergleicht sie erwartetes mit beobachtetem Verhalten?
2. Wie viele Details erfasst die Whitelist-Funktion bei ihrer Entscheidungsfindung? Mehr oder weniger als ein detaillierter Schüler-Hallenausweis?
3. Verfolgt und untersucht die Whitelist-Funktion übergeordnete und vorangehende Programme auf Bösartigkeit? (Programm A öffnet B, öffnet C, öffnet D - sehr häufig)
4. Überprüft die Whitelist-Funktion, ob das laufende Programm nicht manipuliert wurde?

Vertiefende Lektüre des National Institute of Standards & Technology (NIST)

Eine maßgebliche Publikation, die nicht zur beiläufigen Lektüre gedacht ist, aber sehr anschaulich beschreibt.

Obwohl sie nicht jedes Szenario abdeckt und die Details nicht häufig genug aktualisiert werden, um mit jedem Szenario Schritt zu halten, sind die Beschreibungen strategisch genug, um in den meisten Fällen wertvoll zu sein.

Das Team von Cyber Crucible schätzt Erkenntnisse aus NIST-Publikationen stets sehr und nimmt sich entsprechend Zeit dafür.

https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-167.pdf

Wie kann ich erkennen, was ohne das angepasste Verhalten unterbrochen worden wäre?

Benutzer können erkennen, welche Erpressungsreaktionen unterbrochen worden wären, wenn ein angepasstes Verhalten nicht existiert hätte, indem sie zunächst zur Seite Erpressungsreaktionen navigieren, die im Bereich Operationen in der Seitenleiste zu finden ist.

Die Spalte Ausgeschlossen zeigt an, ob eine Erpressungsreaktion ausgeschlossen wurde oder nicht, und der Standardfilter der Spalte zeigt nicht ausgeschlossene, nicht stille Reaktionen an.

ExcludedColumn.png
Um ausgeschlossene Reaktionen anzuzeigen, öffnen Sie den Filter der Spalte Ausgeschlossen und wählen Sie nur Ausgeschlossen aus.

ExcludedFilter.png
Nach dem Klicken auf Anwenden zeigt das Raster ausgeschlossene Erpressungsreaktionen an, die unterbrochen worden wären, wenn das angepasste Verhalten nicht existiert hätte.

Um zu sehen, welches angepasste Verhalten auf eine Reaktion angewendet wurde, klicken Sie zunächst auf den Pfeil in der Spalte Anzahl der Vorfälle für die jeweilige Zeile. Im inneren Raster, das sich nach dem Klicken auf den Pfeil öffnet, zeigt die Spalte Ausblendregel den Namen des angepassten Verhaltens an, das auf die Reaktion angewendet wurde.

ExcludedResponses.png

Verwalten von individuellen Verhaltensregeln

Die Seite „Tailored Behaviors“ finden Sie unter dem Reiter „Operations“ in der Seitenleiste. Beachten Sie, dass Benutzer über die Berechtigung „View Tailored Behaviors“ für eine Gruppe verfügen müssen, um die individuellen Verhaltensregeln dieser Gruppe auf dieser Seite einsehen zu können.

Wie man eine individuelle Verhaltensregel erstellt

Klicken Sie nach dem Aufrufen der Seite „Tailored Behaviors“ auf das Symbol „Prozess ausschließen“ oberhalb dieser Tabelle.

Durch Klicken auf dieses Symbol öffnet sich ein Modal zum Erstellen der individuellen Verhaltensregel.

CreateTailoredBehaviorIcon.png
Create Exclusion.png

Benutzer haben die Möglichkeit, bei der Ausnahme ein übergeordnetes Programm anzugeben

94732291.png?width=504
Geben Sie zusätzliche anzuwendende Einschränkungen ein

CreateTailoredBehaviorAdditionalRestrictions.png
Benutzer haben jetzt die Möglichkeit, die Ausnahme auf bestimmte Dateizugriffsauslöser zu beschränken; standardmäßig sind alle Dateizugriffsauslöser eingeschlossen. Beachten Sie, dass nur Agenten ab Version 4.4.6.2 diese Funktion für Dateizugriffsauslöser unterstützen.

Benutzer haben außerdem die Möglichkeit, temporäre Verhaltensregeln zu erstellen; mehr dazu im nächsten Abschnitt

Wie man temporäre individuelle Verhaltensregeln erstellt und bearbeitet

Benutzer können eine temporäre individuelle Verhaltensregel erstellen, indem sie den normalen Erstellungsprozess wie oben beschrieben durchführen und dabei den Schalter „Make This Tailored Behavior Temporary“ aktivieren. Anschließend können Benutzer die Anzahl der Stunden eingeben, nach denen die temporäre Verhaltensregel ablaufen soll (maximal eine Woche).

Temporary Exclusion.png
Das Ablaufdatum der Verhaltensregel ist in der Spalte „Expiration Date“ der Tabelle sichtbar.

Um bei einer temporären Verhaltensregel das Ablaufdatum zu verlängern, eine temporäre Verhaltensregel dauerhaft zu machen oder eine dauerhafte Verhaltensregel temporär zu machen, klicken Sie auf das Bearbeitungssymbol bei der gewünschten Verhaltensregel in der Spalte „Expiration Date“.

Edit Temp Behavior Icon.png

Durch Klicken auf dieses Symbol öffnet sich ein Modal, in dem Benutzer das Ablaufdatum der Verhaltensregel nach Bedarf bearbeiten können

Edit behavior modal.pngScreenshot from 2024-09-09 15-21-58.png

Wie man individuelle Verhaltensregeln löscht

Um eine oder mehrere Verhaltensregeln gleichzeitig zu löschen, wählen Sie zunächst die gewünschte(n) Verhaltensregel(n) in der Tabelle aus und klicken Sie auf das Papierkorbsymbol oberhalb der Tabelle.

Geben Sie anschließend „delete“ ein und klicken Sie auf die Schaltfläche „Delete“.

RemoveTailoredBehaviorIcon.pngRemoveTailoredBehaviorModal.png

Wie man individuelle Verhaltensregeln in eine andere Gruppe kopiert

Um eine oder mehrere Verhaltensregeln gleichzeitig in eine andere Gruppe zu kopieren, wählen Sie die gewünschte(n) Verhaltensregel(n) in der Tabelle aus und klicken Sie auf das Kopiersymbol oberhalb der Tabelle.

Durch Klicken auf dieses Symbol öffnet sich ein Modal, in dem Sie die Gruppe auswählen können, in die die Verhaltensregel(n) kopiert werden sollen.

Dieses Modal bietet außerdem die Option, die bestehenden Verhaltensregeln in der ausgewählten Gruppe zu löschen (mit Ausnahme automatisch generierter Verhaltensregeln), sodass nach dem Absenden der Anfrage nur die kopierten Verhaltensregeln in der ausgewählten Gruppe verbleiben.

CopyTailoredBehaviorIcon.pngCopyTailoredBehaviorModal.png
Nach dem Absenden der Anfrage werden die kopierten Verhaltensregeln nun für die ausgewählte Gruppe angezeigt.

ResultGrid.png

Erstellen einer individuellen Verhaltensregel über die Seite „Extortion Response“

Benutzer haben die Möglichkeit, individuelle Verhaltensregeln über die Seite „Extortion Responses“ zu erstellen.

Klicken Sie zunächst auf den Pfeil in der gewünschten Zeile der Extortion Response in der Spalte „Number of Incidents“, um die innere Tabelle anzuzeigen.

Klicken Sie anschließend auf das Symbol „Response ausschließen“ in der Spalte „Response Name“, wodurch sich ein Modal zum Erstellen der individuellen Verhaltensregel öffnet.

ExcludeResponseIcon.png

In diesem Modal können Sie Verhaltensregeln anhand der vorgeschlagenen ausführbaren Pfade erstellen oder die Response zur Überprüfung einreichen, falls die vorgeschlagenen Pfade nicht das Gewünschte enthalten.

Exclude Modal.png

Sie können auch auf die Schaltfläche „Take me to Tailored Behaviors Page“ klicken, wodurch Sie zur Seite „Tailored Behaviors“ weitergeleitet werden und das Modal zum Erstellen der Verhaltensregel automatisch mit derselben Gruppe, demselben Pfad und denselben Programmargumenten aus der Response vorausgefüllt wird.

AutofilledModal.png

Beachten Sie, dass die Einstellung „Limit Exception to Program + Parent Program“ standardmäßig deaktiviert ist. Der Pfad und die Argumente des übergeordneten Programms aus der Extortion Response werden ebenfalls automatisch ausgefüllt, wenn Sie diese Einstellung aktivieren.

PrefilledArgs.png

Der Tailored Behavior Exclusion-Eintrag nach Klicken auf „Create“ bei aktivierter Einstellung „Limit Exception to Program + Parent Program“:

ExlusionResult.png

Was ist ein dateiloser (speicherinterner) Angriff?

Kurze Antwort: Ein dateiloser Angriff ist ein Angriff, bei dem niemals ein Programm auf die Festplatte geschrieben wird. Der Angreifer injiziert bösartigen Code direkt in den Speicher eines legitimen, vertrauenswürdigen Prozesses und baut dort seine Payload auf. Da keine Datei erstellt wird, haben Sicherheitstools, die Dateien scannen — einschließlich der meisten EDR-Lösungen — nichts zu finden.

Warum dateibasierte Tools versagen

Herkömmlicher Endpunktschutz basiert auf einer einfachen Annahme: Malware ist eine Datei, also müssen Dateien überprüft werden. Dateilose Angriffstechniken entfernen die Datei vollständig aus der Gleichung.

In einem realen Einsatz betrieb ein Finanzdienstleistungsunternehmen drei branchenführende EDR-Produkte (Microsoft, CrowdStrike und Sophos) zusammen mit einem Top-50-Managed-SOC. Angreifer kompromittierten ein vertrauenswürdiges Remote-Monitoring-Tool, injizierten Code in die Verwaltungsprozesse von Microsoft SQL Server und kompilierten ihre Ransomware sowie ihre Datendiebstahl-Malware direkt im Serverspeicher. Da nichts auf die Festplatte gelangte, waren alle drei EDR-Lösungen, der MDR-Dienst und das SOC vollständig blind. Cyber Crucible fing eigenständig fast 10.000 bösartige Prozesse ab — 98 % davon auf der SQL-Server-Farm — ohne einen einzigen Alarm des Legacy-Stacks.

Wie Cyber Crucible dies erkennt

Cyber Crucible überwacht Verhalten und Absicht auf Kernel-Ebene, anstatt Dateien zu scannen. Ein Prozess, der sich bösartig verhält, wird in weniger als 200 Millisekunden gestoppt — unabhängig davon, ob jemals eine Datei geschrieben wurde.

Was ist Hacker-Automatisierung, und warum durchbricht sie traditionelle Verteidigungsmaßnahmen?

Kurze Antwort: Hacker-Automatisierung ist der Einsatz von Skripten, KI und robotischer Prozessautomatisierung, um jede Phase eines Angriffs mit Maschinengeschwindigkeit auszuführen. Ein automatisierter Angriff kann einen Rechner infiltrieren, Daten stehlen und seine eigenen speicherresidenten Tools innerhalb von Sekunden löschen — weitaus schneller, als jeder menschliche Analyst reagieren könnte.

Was automatisiert wird

- **Aufklärung (Reconnaissance):** Tools durchsuchen das Internet nach anfälligen Systemen und kartieren ein Zielnetzwerk innerhalb von Sekunden und erzeugen überzeugendes Phishing in großem Maßstab.
- **Ausnutzung (Exploitation):** Sobald eine Schwachstelle gefunden wird, setzen Frameworks den Exploit sofort ein — ein Zero-Day kann auf Millionen von Rechnern genutzt werden, bevor ein Mensch den Alarm registriert.
- **„Smash and Grab“:** Moderne Malware bewertet nicht, welche Dateien wertvoll sind. Sie verschlüsselt oder exfiltriert alles, was sie erreichen kann, so schnell wie möglich.

Warum Menschen dieses Rennen nicht gewinnen können

Die Grenze liegt nicht an der Qualität des Teams oder der Personalausstattung. Es ist die Biologie. Die Zeit, die eine Person benötigt, um einen Alarm zu sehen, ihn zu verarbeiten und zu handeln, ist länger als die Zeit, die der Angriff zur Ausführung benötigt. Das Weiterleiten von Telemetriedaten an einen Cloud-Server zur Analyse und zurück verursacht zusätzliche Verzögerungen.

Die einzige praktikable Antwort ist eine Verteidigung, die autonom, auf dem Endpunkt, innerhalb von Millisekunden entscheidet und handelt.

Warum zielen automatisierte Angriffe auf jedem Computer auf dieselben Speicherorte ab?

Kurze Antwort: Angreifer wissen nichts über Ihre spezifische Umgebung, daher ist ihre Automatisierung so geschrieben, dass sie auf Speicherorte abzielt, die praktisch auf jedem Rechner existieren – Benutzerprofilordner, Laufwerksbuchstaben und Anwendungsdatenverzeichnisse. Diese Vorhersehbarkeit ist eine Schwachstelle, die Verteidiger nutzen können.

Die vorhersehbaren Ziele

- **Benutzerprofile und Desktops:** Unter Windows enthält `C:\Users\[Username]` zuverlässig wertvolle Daten. Ein Skript durchläuft jedes Benutzerverzeichnis, ohne das System verstehen zu müssen.
- **Laufwerksbuchstaben:** Automatisierte Tools zählen `C:\`, `D:\` und weitere Laufwerke auf, um Netzwerkfreigaben, externe Laufwerke und eingebundene Partitionen zur Verschlüsselung zu finden.
- **Anwendungsdatenordner:** Jedes Betriebssystem speichert Zugangsdaten, Cookies und Konfigurationen an bekannten Pfaden – bevorzugte Ziele für Identitätsdiebstahl.

Die Vorhersehbarkeit gegen den Angreifer wenden

Das Angriffsmuster ist nicht durchdacht; es ist ein plumpes Durchkämmen bekannter Speicherorte. Cyber Crucible überwacht genau diese bekannten Einstiegspunkte für Identitäts- und Datendiebstahl. Bei jedem Programm, das an diesen Speicherorten auf Daten zugreift – zusammen mit seinen übergeordneten und untergeordneten Prozessen sowie den zugehörigen Bibliotheken – wird die Absicht in Bruchteilen einer Sekunde bewertet und im Falle einer böswilligen Absicht unterbunden.

Was ist ein Infostealer, und warum ist der Diebstahl von Sitzungstoken so gefährlich?

Kurze Antwort: Ein Infostealer ist Schadsoftware, die darauf ausgelegt ist, digitale Identitätsdaten – Passwörter, Sitzungstoken, API-Schlüssel und VPN-Anmeldedaten – abzugreifen und schnell zu exfiltrieren. Ein gestohlenes Sitzungstoken ist gefährlich, weil es Passwörter und Multi-Faktor-Authentifizierung oft vollständig umgeht und einem Angreifer erlaubt, sich als legitimer Benutzer anzumelden.

Warum Angreifer heute Identitäten stehlen, statt im Netzwerk zu verweilen

Moderne Angreifer vermeiden es zunehmend, sich dauerhaft in einem Netzwerk aufzuhalten. Ein Verbleib in der Umgebung erhöht die Wahrscheinlichkeit einer Entdeckung. Stattdessen entwenden sie die Identitätsdaten und verschwinden wieder, was ihnen zwei Vorteile verschafft:

1. **Geringeres Risiko** – Sie analysieren das Gestohlene in einer sicheren Umgebung, in ihrem eigenen Tempo.
2. **Einfache Rückkehr** – Mit einem gültigen Sitzungstoken, Passwort oder VPN-Schlüssel können sie jederzeit zurückkehren und erscheinen dem Authentifizierungssystem gegenüber vollkommen legitim.

Warum Geschwindigkeit das eigentliche Problem ist

Diese Tools bewegen sich innerhalb von Sekunden von der Infiltration zur Selbstlöschung – schneller, als eine Cloud-Warnung das Gebäude verlassen kann. Cyber Crucible erkennt die Absicht im Moment des Zugriffs auf Identitätsdaten und unterbricht den Prozess in unter 200 Millisekunden, bevor die Exfiltration beginnt.

Was sind Living-off-the-Land-(LotL)-Angriffe, und warum versagt Anwendungs-Whitelisting gegen sie?

Kurze Antwort: Living-off-the-Land-Angriffe nutzen Software, die bereits installiert und bereits vertrauenswürdig ist – Systemhilfsprogramme, Administrationstools, Browser – anstatt neue schädliche Dateien einzuschleusen. Anwendungs-Whitelisting versagt, weil der Angreifer innerhalb von Programmen agiert, die die Whitelist ausdrücklich zulässt.

Die Whitelist wird zur Zielliste

Anwendungs-Whitelisting beantwortet eine einzige Frage: „Darf diese Datei ausgeführt werden?“ Moderne Angriffstechniken machen diese Frage bedeutungslos. Wenn Angreifer schädlichen Code im Arbeitsspeicher eines zugelassenen Prozesses verstecken, hat die Whitelist bereits zugestimmt. In der Praxis verrät eine Whitelist einem Angreifer genau, in welchen Prozessen er sich am sichersten verstecken kann.

Die bessere Frage

Wirksame Verteidigung muss sich von *„Ist diese Datei erlaubt?“* zu *„Was tut dieser Code gerade tatsächlich?“* weiterentwickeln. Das erfordert die Untersuchung von Verhalten und Arbeitsspeicher auf Kernel-Ebene, wo die tatsächliche Aktivität sichtbar ist – nicht auf Dateiebene, die der Angreifer bereits umgangen hat.

Was ist ein System32-DLL-In-Memory-Angriff?

Kurze Antwort: Es handelt sich um einen Angriff, der die zentralen Windows-Codebibliotheken (System32-DLLs) infiziert, während diese im Arbeitsspeicher ausgeführt werden. Da nahezu jede Anwendung und jedes Sicherheitstool zur Funktion auf diese Bibliotheken angewiesen ist, verschafft deren Kompromittierung einem Angreifer nahezu vollständige Kontrolle über den Endpunkt – bei minimalen Spuren.

Warum dies den Höhepunkt der Endpoint-Angriffstechniken darstellt

System32-DLLs stellen die grundlegenden Operationen bereit, auf die sich Windows und seine Anwendungen verlassen – Speicherzuweisung, Kryptografie, Netzwerkkommunikation. Sie wurden für Geschwindigkeit und Zuverlässigkeit entwickelt, niemals dafür, im laufenden Produktivbetrieb gepatcht oder gehookt zu werden.

Ein Angreifer, der sie im Arbeitsspeicher verändern kann, erlangt die Fähigkeit, Daten und Programme auf dem System zu untersuchen, zu erstellen, zu verändern oder zu zerstören – mit nahezu nicht nachweisbarer Unauffälligkeit.

Warum die meisten Tools dies nicht erkennen können

Sicherheitsprodukte, die auf von Microsoft bereitgestellte Sensordaten angewiesen sind, verlassen sich per Definition auf genau die Bibliotheken, die angegriffen werden. Die für die Erkennung dieser Angriffsklasse erforderlichen Sensoren existieren in Standard-Toolsets schlicht nicht – sodass das Tool verstummt, während es meldet, dass alles in Ordnung sei.

Cyber Crucible wurde bewusst so entwickelt, dass es unabhängig von Windows-Bibliotheken funktioniert, damit ein kompromittiertes Betriebssystem es nicht blenden oder deaktivieren kann.

Warum aktiviert sich Malware nicht immer in einer Sicherheitstest-Umgebung?

Kurze Antwort: Angreifer konzipieren Malware so, dass sie inaktiv bleibt, bis sie ein Validierungssignal von einem lebenden Command-and-Control-Server (C2) erhält, den sie kontrollieren. Bis eine Probe in eine Malware-Datenbank gelangt, hat sich die C2-Infrastruktur in der Regel bereits verlagert – sodass Forscher häufig eine verwaiste Probe testen, die nichts tut.

Warum Angreifer Inaktivität einbauen

- **Minimierung der Informationsgewinnung:** Inaktive Malware führt keine beobachtbaren bösartigen Aktionen aus, wodurch Analysten weniger dokumentieren können und Anbieter weniger Grundlage für die Erstellung von Signaturen haben.
- **Verhinderung einer Übernahme:** Durch das Erfordernis eines geheimen Erkennungssignals wird sichergestellt, dass ein Konkurrent oder Forscher, der den C2-Server übernimmt, die Malware nicht einfach steuern und die gestohlenen Daten abgreifen kann.

Was das für die Bewertung von Sicherheitstools bedeutet

Ein Labortest gegen eine getrennte Probe misst sehr wenig. Die entscheidende Frage ist nicht, ob ein Tool auf inaktive, reglos vorliegende Malware reagiert, sondern ob es lebende, schnell ablaufende Angriffe stoppt, während sie geschehen. Cyber Crucible wurde für Letzteres entwickelt: Es überwacht die Einstiegspunkte für Identitäts- und Datendiebstahl, auf die reale Angriffe abzielen, bewertet die Absicht in unter 200 Millisekunden und stoppt den Diebstahl, bevor er erfolgreich ist.