Wie kann ich die Ursache eines Ereignisses untersuchen?

- Speicheranalyse
- Analyse von Prozess-Injektionen
- Analyse von Prozesserstellungen
- Alles zusammengeführt – eine durchgängige Untersuchung
  - Warum haben wir uns entschieden, dieses Beispiel zu zeigen?

Es gibt drei Analysebausteine, die Ihnen bei der Untersuchung eines Alarms zur Verfügung stehen.

Derzeit ist es wahrscheinlicher, dass ein Angreifer eine Vielzahl von Aktivitäten auf einem System durchführt, indem er eine Kombination aus speicherbasierten Techniken und Prozess-Hijacking (Injection oder Hollowing) nutzt, anstatt ein offensichtliches „hacker.exe“-Programm auszuführen.

Speicheranalyse

Der erste Baustein findet sich direkt auf der Response- bzw. Incident-Seite. Die Speicheranalyse verfolgt das Verhalten des Prozesses im Speicher während der kontinuierlichen Überprüfung auf mögliches Datenerpressungsverhalten (Diebstahl oder Verschlüsselung).

Ein Speicherzustand, der auf eine wahrscheinliche Manipulation hinweist, ist typischerweise ein Hinweis auf ein Problem.

Dies ist nicht immer bösartig an sich, da auch schlechte Programmierpraktiken oder Softwarefehler dazu führen können, dass der Speicherzustand eines laufenden Programms verändert wird.

63504385.png?width=680
Hier sehen wir einen Fall, in dem ein Adobe-Acrobat-Prozess nach dem Öffnen einer heruntergeladenen PDF-Datei mögliches Datenerpressungsverhalten zeigt, und wir sehen, dass der Speicher des Acrobat-Prozesses verändert wurde. Die Acrobat.exe-Datei auf der Festplatte wurde in keiner Weise beeinträchtigt.

Eine forensische Analyse des mit der angehaltenen Acrobat.exe verbundenen Speichers wird wahrscheinlich auf ein bedenkliches Problem hinweisen – möglicherweise auf einen genutzten

Exploit, wenn nicht sogar auf Prozess-Injection oder Prozess-Hollowing.

Bei Erpressungs-Responses mit dem Wert „True“ für „Modified Memory“ gibt es in der Spalte „Root Cause Analysis“ ein Download-Symbol, um die Memory-Diff-Datei für die Response herunterzuladen:

63602693.png?width=680
Die Memory-Diff-Datei ist der kompilierte Quellcode dessen, was in das Programm injiziert wurde. Sie enthält wahrscheinlich Malware und/oder Exploit-Code, den ein Malware- oder Sicherheitsanalyst untersuchen kann. Beachten Sie, dass diese Datei nicht das gesamte Programm enthält, sondern nur den Teil, der durch einen Exploit oder eine Angriffstechnik wie Prozess-Injection verändert wurde. Erfahren Sie mehr über diese Datei hier.

Analyse von Prozess-Injektionen

Der erste Hinweis darauf, dass ein Prozess-Injection-Ereignis für eine automatisierte Response relevant sein könnte, ist die Spalte „Untrusted Remote Threads“ für einen Incident/Response. Dies zeigt an, dass die Verhaltensanalyse auf Kernel-Ebene einen nicht vertrauenswürdigen Remote-Thread festgestellt hat, der bei einem Prozess mit Datendiebstahl- oder Ransomware-ähnlichem Verhalten beobachtet wurde.

63471635.png?width=680
Eine weitergehende Untersuchung kann auf der Seite „Process Injection“ durchgeführt werden. Der Wert „True“ bei „Untrusted Remote Threads“ ist ein guter Hinweis darauf, ob man dorthin gehen sollte.

Die Spalte „Root Cause Analysis“ verfügt außerdem über ein Symbol, mit dem Sie automatisch zur Seite „Injections“ weitergeleitet werden, um nur die zu der Response gehörenden Prozess-Injektionen anzuzeigen.

63602703.png?width=680
Nach dem Klicken auf das Injections-Symbol werden Sie zur Seite „Process Injections“ weitergeleitet, auf der Sie nur die zu der Response gehörenden Prozess-Injektionen sehen:

63569930.png?width=680
Hier auf der Seite „Process Injection Events“ sehen wir den Pfad und die Argumente des Injektor-Prozesses sowie den Pfad und die Argumente des injizierten Prozesses (Injectee). Die Argumente spielen in der Regel eine sehr wichtige Rolle bei der Speicheranalyse oder zeigen beispielsweise an, welche PowerShell-Befehle ausgeführt werden sollen. Auch die Prozess-IDs sind vorhanden, um iterative Injektionen zu verfolgen, während ein Angreifer von Prozess zu Prozess wechselt, oder um die Ursachenanalyse auf der Seite „Process Creation“ nachzuschlagen.

63438879.png?width=680
Wie wussten wir, dass rundll32.exe in Wirklichkeit ein Antivirenanbieter war? Lesen Sie weiter!

Analyse von Prozesserstellungen

Es ist sehr selten, dass ein Angreifer bei seinem Angriff nur einen einzigen Prozess verwendet. Möglicherweise weiß er nicht einmal, dass die von ihm verwendeten Tools neue Prozesse erzeugen und verschiedene Verwaltungstools auf dem infizierten System nutzen.

Die Seite „Process Creation“ verfolgt alle Prozesse, die andere Prozesse (oder sich selbst) öffnen, die dabei verwendeten Argumente sowie die Prozess-IDs (Pids) für jede Quelle und jedes Ziel.

Dies erfolgt auf Kernel-Ebene, sodass nichts übersehen wird.

Hinweis: Die Microsoft-Telemetrie enthält einige dieser Daten, jedoch nicht alle. Wir verzichten bewusst auf diese Datenquelle, da wir festgestellt haben, dass diese Telemetriequelle wichtige Ereignisse übersehen oder von Angreifern während eines Angriffs deaktiviert werden kann.

Wie gut ist unsere Sichtbarkeit?

Alles zusammengeführt – eine durchgängige Untersuchung

Wie gut ist unsere Sichtbarkeit, und wie schnell können Sie eine Ursachenanalyse (Root Cause Analysis) durchführen? Sehen Sie sich dieses Beispiel an – ein Antivirenprogramm mit möglicherweise einigen der höchsten Berechtigungen auf einem System wurde dabei beobachtet, wie es versuchte, Code in unsere Cyber Crucible-Software zu injizieren. Wir hoffen, dass dies zur Untersuchung unserer Software geschah, aber wir wechseln trotzdem in den Selbstschutzmodus (das führte also dazu, dass unsere Software die Manipulation durch das Antivirenprogramm zurückwies … es gibt keine Hintertüren, die es externen Parteien ermöglichen, unsere Software zu bearbeiten oder zu nutzen).

Beginnen wir also hier mit einer automatisierten Response:

63569937.png?width=680
Offensichtlich ist Cyber Crucible keine Ransomware. Wir sehen hier, dass ein Prozess-Injection-Versuch mit verdächtigem Remote-Thread-Verhalten in Verbindung mit datenerpressungsähnlichem Verhalten auftrat.

OK – schauen wir uns an, was in den Process Injection Events passiert, indem wir auf das Injections-Symbol klicken. 5 Sekunden später haben wir unsere Antwort!

63471644.png?width=56463438879.png?width=680

Aber Moment – rundll? Das ist doch kein Hacker, oder? Das ist sicherlich kein hacker.exe-Programm. Lassen Sie uns etwas filtern, um unsere Antwort auf der Seite „Process Creation“ zu erhalten. Wir haben hier keinen Screenshot, aber Prozess-IDs sind für alle Responses und Prozess-Injektionen verfügbar. Gehen wir zurück zur Seite „Responses“ und klicken auf das Process-Creations-Symbol, um zur Seite „Process Creations“ zu gelangen und das Raster automatisch so zu filtern, dass die zu dieser Response gehörenden Prozesserstellungen angezeigt werden.

63537184.png63438886.png?width=680

Schnell und einfach sehen wir nun, dass die „Wurzel“ (kein Wortspiel beabsichtigt) WebRoot war, ein Antivirenanbieter, der einen rundll32.exe-Prozess erzeugte, um eine der DLLs von WebRoot zu laden, die dann versuchte, *irgendetwas* mit der Software von Cyber Crucible zu tun, um sie zu zwingen, Dateien wie ein Datendiebstahl-Tool zu durchlaufen.

Warum haben wir uns entschieden, dieses Beispiel zu zeigen?

Wir verfügen über zahlreiche Beispiele für Angreifer-Taktiken mit unterschiedlichem Kompetenzniveau der Angreifer. Antivirenprogramme erreichen ein Berechtigungsniveau auf einem System, das ein sehr hohes Maß an Fähigkeiten, Technologie und Vorbereitung von Seiten der Angreifer erfordert. Dass Cyber Crucible „hackerähnliches“ Verhalten von einem etablierten Sicherheitstool erkennt und es dann automatisch stoppt, zeigt einige der höchsten Ebenen technischer und sicherheitstechnischer Exzellenz unsererseits, ohne sich in einzelnen Exploits oder Angreifer-Taktiken zu verlieren.

Revision #2
Created 2026-07-24 04:17:38 UTC by Dennis Underwood
Updated 2026-11-24 15:53:02 UTC by Dennis Underwood