KAPE-Einführung, Teil 2:
Autoruns-Artefakte auswerten und verstehen

Am Start geschnappt

Gregor Wegberg

Angreifer und Schadsoftware streben nach erfolgreicher Kompromittierung eines IT-Systems häufig Persistenz in der Umgebung an, um trotz Neustart, Änderung von Zugangsdaten oder anderen Unterbrechungen Zugriff zu behalten. Der zweite Teil des Tutorials zeigt, wie man mit KAPE und Microsofts Autoruns-Tool einige Persistenzmechanismen aufdeckt.

X-TRACT

- Nach erfolgreichem Zugriff auf ein IT-System versuchen Angreifer und Schadsoftware, sich in der IT-Landschaft für weitere missbräuchliche oder kriminelle Handlungen dauerhaft festzusetzen.
- Sogenannte Persistenz stellt sicher, dass Schadsoftware Unterbrechungen und Veränderungen von IT-Systemen übersteht.
- Windows-Systeme bieten eine Vielzahl von Mechanismen an, die zur Erlangung von Persistenz missbraucht werden können. Mehrere bekannte und verbreitete lassen sich mit dem Autoruns-Tool finden.

Angreifern steht eine große Zahl unterschiedlicher Persistenzmechanismen zur Verfügung, mit denen sie sich dauerhaft auf einem System einnisten können. Allein in der Windows-Matrix der Wissenssammlung MITRE ATT&CK Framework (zu finden über ix.de/z3pp) sind 87 solcher Mechanismen dokumentiert. Dieses Framework ist eine systematische, kategorisierte Aufstellung von Verhaltensmustern, die bei Angreifern und Schadsoftware beobachtet wurde.

In den letzten Jahren wurde sie zu einer zentralen Referenz und zum täglichen Werkzeug für Experten sowohl offensiver als auch defensiver IT-Sicherheit. Zu jeder Angriffstechnik (Techniques und Sub-Techniques) gibt es auf der Webseite des Projekts MITRE-ATT&CK eine Kurzbeschreibung, eine Liste von Angreifern und Schadsoftware, die sie angewendet haben, Empfehlungen zu relevanten Schutzmaßnahmen und Hinweise zur Detektion.

Das Ausfindigmachen der Sicherheitslücke, über die ein Angreifer oder eine Malware einen Angriff begonnen hat und in ein IT-System eingedrungen ist, ist ein wichtiger Bestandteil bei der Bewältigung eines Informationssicherheitsvorfalls. Ihre Schließung soll sicherstellen, dass es nicht zu einer erneuten Kompromittierung kommt. Dies allein reicht aber nicht aus. Es ist wahrscheinlich, dass sich der Angreifer mithilfe von Persistenzmechanismen in der IT-Umgebung eingenistet und trotz Abdichten des ursprünglichen Eintrittspunktes weiterhin Zugang zu ihr hat. Aus diesem Grund ist das Aufspüren solcher Persistenzen ein ebenso wichtiges Element der Sicherheitsvorfallbewältigung. Mit Autoruns lässt sich ein Teil der häufig genutzten Mechanismen entdecken.

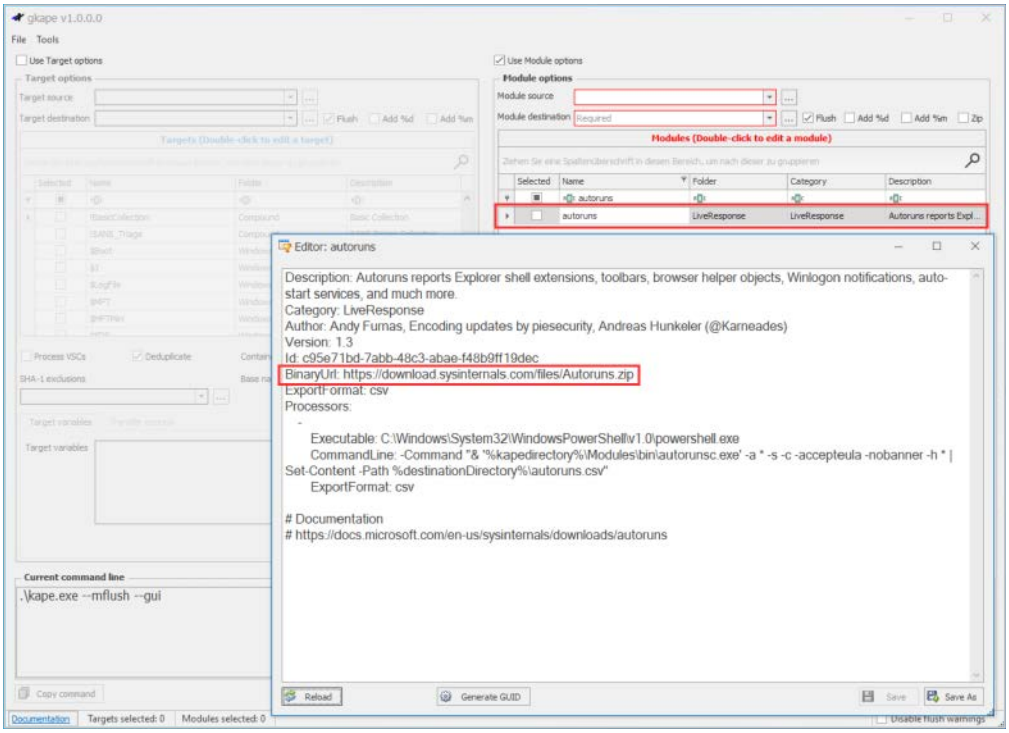
Autoruns: ein findiges Werkzeug

Autoruns wird von Microsoft Sysinternals entwickelt und als Freeware zur Verfügung gestellt. Mit diesem Tool lässt sich eine Vielzahl von Orten analysieren, die das automatische Ausführen von Skripten oder Programmen erlauben. Dazu gehört unter anderem der altbekannte Autostart-Ordner im Startmenü, aber auch weniger bekannte Registry-Schlüssel. So gibt es Schlüssel, die beliebige Software beim Starten einer Microsoft-Office-Applikation ausführen (Werkzeug und Beschreibung der verschiedenen Autostart-Varianten siehe ix.de/z3pp).

Für den täglichen IT-Betrieb ist die grafische Oberfläche von Autoruns perfekt

geeignet und einfach zu nutzen. Bei einem IT-forensischen Einsatz während eines Cybervorfalls wird hingegen die Konsolenapplikation eingesetzt, denn sie minimiert bei Live-Forensics-Untersuchungen Änderungen am untersuchten System und erlaubt die vollautomatische Ausführung mit KAPE, dem IT-forensischen Werkzeug zum Einsammeln von Daten.

KAPE verfügt über ein Autoruns-Modul als Teil der LiveResponse-Kategorie. Es führt das Autoruns-Tool auf dem zu untersuchenden System aus und sammelt das Resultat für die spätere Analyse ein. Dem Werkzeug muss diese Drittanwendung zur Ausführung bereitgestellt werden. Wie im ersten Teil des Tutorials beschrieben, ist KAPE zuerst auf einem Analysesystem vorzubereiten: Es wird auf einen externen Datenträger kopiert und aktualisiert. Eine Vielzahl von Modulen benötigen Programme, die man vor dem Ausführen herunterladen und einrichten muss. Sie enthalten jeweils einen BinaryUrl-Eintrag, der zum Download der notwendigen Drittanwendung führt. Um an die URL für das Autoruns-Modul zu kommen, startet man am einfachsten die grafische Oberfläche von KAPE, wählt „Use Module options“ aus, öffnet per Doppelklick die Beschreibung des Autoruns-Moduls und lädt es von der BinaryUrl-Adresse herunter (siehe Abbildung 1). Danach entpackt man das heruntergeladene Zip-Archiv in den Unterordner Modules/bin von KAPE. Damit ist das Autoruns-Modul für den Einsatz bereit.



In der grafischen Oberfläche lässt sich das Autoruns-Modul mit einem Doppelklick einsehen. Für die Modulausführung benötigte Dateien können unter der BinaryUrl-Adresse heruntergeladen werden (Abb. 1).

KAPE und Autoruns ausführen

Wie im ersten Teil des Tutorials beschrieben, wird der Kommandozeilenbefehl vor der Ausführung auf dem Analysesystem vorbereitet. Hierzu startet man erneut die grafische Oberfläche von KAPE, wählt „Use Module options“ aus und setzt „Module source“ auf die Systempartition des zu untersuchenden Systems, also sehr wahrscheinlich C:. Ein Teil der Module dieser Kategorie, wie Autoruns, ignorieren die „Module source“ komplett. Andere wiederum nutzen diesen Pfad analog den Targets und der Target Source, also als Quelle der zu analysierenden Daten. Aus diesem Grund sollte man sich angewöhnen, gleich die Systempartition auszuwählen.

Vorsicht, um die Veränderungen am untersuchten System minimal zu halten, sollten darauf nur LiveResponse-Module gestartet werden. Die restlichen sollten immer auf dem Analysesystem laufen. Für „Module destination“ wird ein Platzhalter gewählt, bis der Laufwerksbuchstabe des externen Datenträgers am zu untersuchenden System bekannt ist. Abschließend wird das Autoruns-Modul ausgewählt und der Kommandozeilenbefehl unter „Current command line“ für die Ausführung gespeichert (siehe Abbildung 2).

Nun schließt man das externe Laufwerk an das zu untersuchende System an, startet eine administrative Konsole wie PowerShell und wechselt in den KAPE-Ordner. Der vorbereitete Kommandozeilenaufruf wird jetzt in die Kommandozeile eingefügt und der Platzhalter für die „Module destination“ ersetzt: Im vorliegenden Beispiel wurde dem Speicher der Laufwerksbuchstabe F zugeordnet. Entsprechend wird der Platzhalter durch einen gültigen Ordnerpfad für die Sammlung der Resultate ersetzt:

```
.\kape.exe --msource C: --mdest F:\
        TargetDestination\ --mflush --module
                                    autoruns -gui
```

Nun kann das Sammeln der Autoruns-Informationen durch die Ausführung des Befehls starten. Danach befinden sich die eingesammelten Daten als CSV-Datei im Zielordner unter LiveResponse\autoruns.csv (hier F:\TargetDestination\LiveResponse\autoruns.csv).

Gut versteckt ist halb gewonnen

Für den fiktiven Angriff in diesem Teil des Tutorials wurden fünf Autostart-Orte als Persistenzmechanismen auf einem Windows-10-System ausgenutzt. Das Szenario kann mit dem Skript `Add-Persistence.ps1` nachgebildet werden (zu finden unter ix.de/z3pp). Die eingerichteten Persistenzen fügen jeweils eine Textzeile zu einer Logdatei (proofPrivileged.txt oder proofUser.txt unter C:\ProgramData\USOShared) als Beweis der Ausführung hinzu. Anstelle des Textes hätte auch beliebige Schadsoftware gestartet werden können.

Für die Analyse der CSV-Datei eignen sich die einschlägigen Tabellenkalkula-

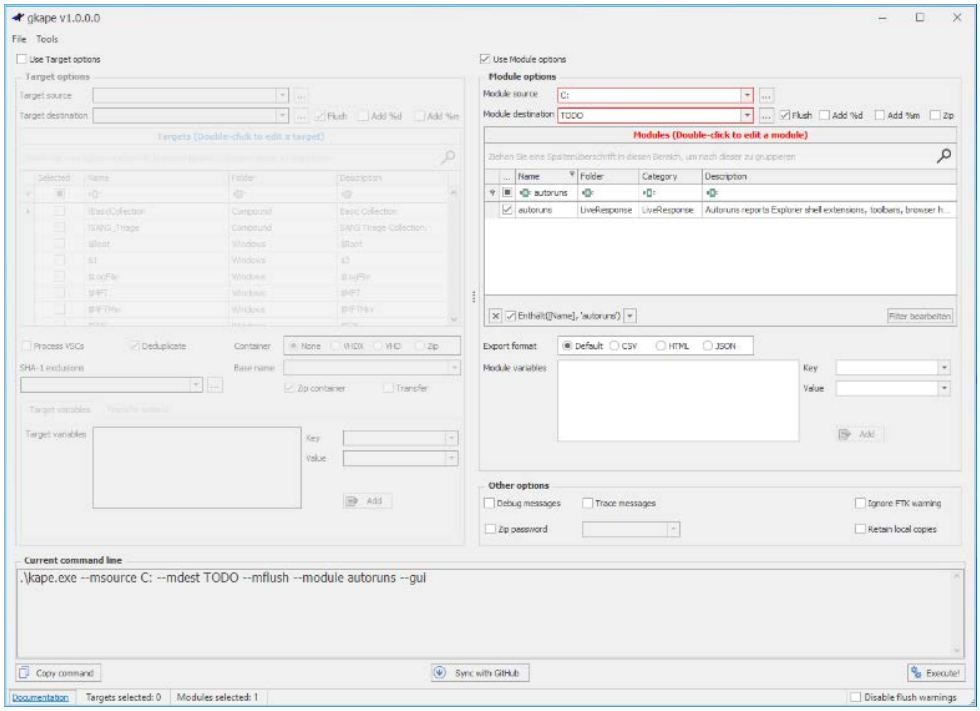
Tutorialinhalt

Teil 1: Installation, Konfiguration und Ausführung von KAPE

Teil 2: Autoruns-Artefakte auswerten und verstehen

Teil 3: Browserhistorie auswerten und verstehen

Teil 4: Was wurde von wem wann ausgeführt?



Beispielkonfiguration für die Ausführung von Autoruns: Hier ist zu nennen, von wo aus das Modul agiert und auf welchem System es Daten sammelt (Abb. 2).

tionsprogramme. Wie im ersten Teil des Tutorials wird auch in diesem die CSV-Datei mit dem Timeline Explorer analysiert. Für eine bessere Übersicht wählt man im Tools-Menü den Eintrag „Reset column widths“ (Strg + R). Dieser setzt alle Spaltenbreiten auf einen vordefinierten Maximalwert.

Der erste dieser Autostart-Orte ist die Windows-Aufgabenplanung. Mit ihr können Software, Programme oder Applikationen sowie Nutzer das Ausführen von Anwendungen zu vorgegebenen Zeitpunkten festlegen. Sie wird von legitimer Software und im IT-Betrieb häufig und gerne genutzt. Auch Angreifer setzen sie ein, was der MITRE-ATT&CK-Framework-Eintrag dokumentiert (siehe ix.de/z3pp). Damit stellen sie sicher, dass ihre Schadsoftware regelmäßig oder zu spezifischen Zeitpunkten ausgeführt wird. So überlebt der Schädling zum Beispiel einen Neustart.

Zur Analyse dieser Scheduled Tasks (Aufgabenplanung) gruppiert man die Daten zuerst nach dem Autostart-Ort: Im Kontextmenü der Spalte „Category“ wird der Eintrag „Group By This Column“ gewählt und danach die Gruppe „Category: Tasks“ aufgeklappt (siehe Abbildung 3). In dieser Kategorie finden sich die geplanten Aufgaben.

Aus IT-forensischer Perspektive müsste man nun jeder Zeile auf den Zahn fühlen. Bei Cyberangriffen fehlt dafür häufig die Zeit und Forensikexperten beschleunigen die Analyse durch die Suche nach typischen Warnsignalen. Hier hat jeder Experte sein eigenes Vorgehen. Zum Beispiel kann es helfen, nach „enabled“ in der Enabled-Spalte zu filtern. Unabhängig von vorhergehenden Filtern muss man schlussendlich die „Launch String“-Spalte der verbleibenden Einträge auf Auffälligkeiten überprüfen.

In der Praxis lohnt es sich, zuerst nach Aufrufen von powershell.exe, cmd.exe, cscript.exe und wscript.exe zu suchen. All diese Windows-Applikationen sind Laufzeitumgebungen, mit denen Angreifer schädliche Skripte ausführen können und damit versuchen, unter dem Radar von Antivirus-Scannern zu bleiben. Dies führt auch zum Fund des unerwünschten Scheduled Task im Beispielszenario (siehe Abbildung 4).

Ein so auffälliger Launch String (Aufruf) ist nicht unüblich. Einige Angreifer geben sich trotzdem Mühe, unbemerkt zu bleiben: Sie nutzen einen unauffälligen Namen in der Entry-Spalte (zum Beispiel „WinUpdate“), verschleiern die Kommandozeile oder modifizieren einen bestehenden Eintrag. Deshalb ist es besonders wichtig, sich den Aufruf genau anzusehen und sich nicht von den restlichen Informationen fehlleiten zu lassen.

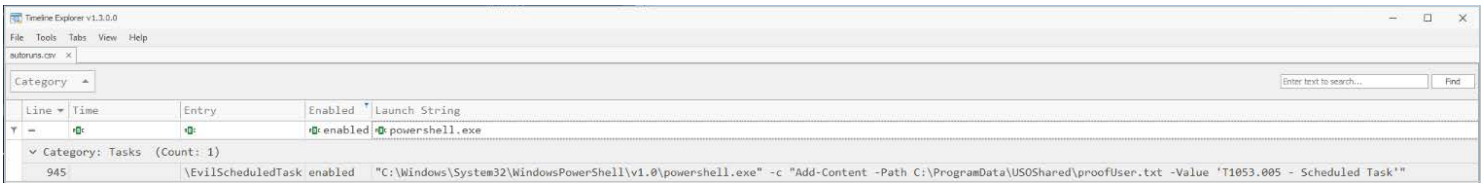
Der zweite Angriffspunkt des Autostart-Beispielszenarios betrifft die Windows-Dienste. Sie ermöglichen weitere beliebte Vorgehensweisen zur Erlangung von Persistenz. Angreifer richten dazu neue Dienste ein oder modifizieren bestehende, um die eigene Schadsoftware ausführen zu lassen (siehe ix.de/z3pp).

Unterschiedliche Techniken können die Suche nach auffälligen Diensten beschleunigen: Fehlt zum Beispiel eine ausführliche Beschreibung in der Description-Spalte, sollte der Eintrag genauer geprüft werden. Angreifer konfigurieren häufig keinen sinnvollen Text oder einen in einer anderen Sprache, etwa Englisch auf einem deutschen System. Im „Launch String“ wiederum sollte nach Auffälligkeiten im Dateipfad gesucht werden.

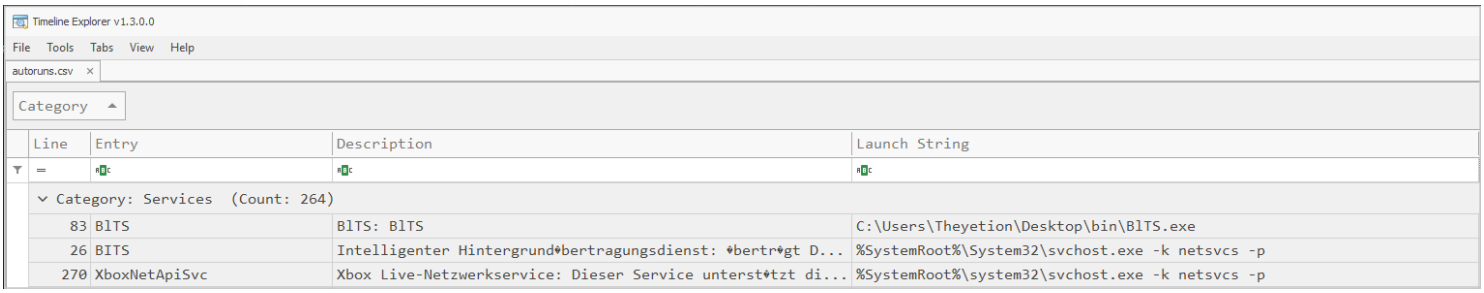
Die Schrift machts

Falls die aufgerufene Anwendung in einem Ordner eines Nutzerprofils oder einem temporären Verzeichnis liegt, ist eine tiefere

Autoruns-Resultate nach Typ gruppiert. Auf dem untersuchten System wurden 182 Scheduled Tasks gefunden (Abb. 3).



Timeline-Explorer-Filter erlauben das schnelle Auffinden auffälliger Einträge (Abb. 4).



Der erste Dienst weist mehrere Auffälligkeiten auf. Die von ihm aufgerufene Applikation muss daher zwingend vertieft analysiert werden (Abb. 5).

Prüfung notwendig. Angreifer erstellen gerne Dienste mit Namen, die einem Windows-eigenen ähneln oder die eine Applikation aufrufen, die wie eine bekannte Windows-Applikation benannt ist. Diese fallen besonders auf, wenn bei der Untersuchung eine für die Softwareentwicklung entwickelte Schrift verwendet wird. Bei solchen Fonts ist jedes Zeichen klar von allen anderen unterscheidbar. Timeline Explorer nutzt eine solche Schrift, um diese Art Untersuchungen zu unterstützen.

Dieses Vorgehen erlaubt das Auffinden des bösartigen Dienstes im Beispielszenario. Der Dienst „BlTS“ sticht hervor: Der Name BlTS (zweiter Buchstabe ist ein kleines „L“) versucht durch die Ähnlichkeit zum legitimen Windows-Dienst BITS (zweiter Buchstabe ist ein großes „i“) nicht aufzufallen. Das ist ein starkes Indiz dafür, dass es sich hierbei um einen unerwünschten Dienst handelt. Die Beschreibung (Description-Spalte) fällt ebenfalls auf. Sie enthält zweimal den Namen des Dienstes. Auffällig ist auch der Ort der aufgerufenen Applikation („Launch String“-Spalte), der Teil eines Nutzerprofils ist (siehe Abbildung 5).

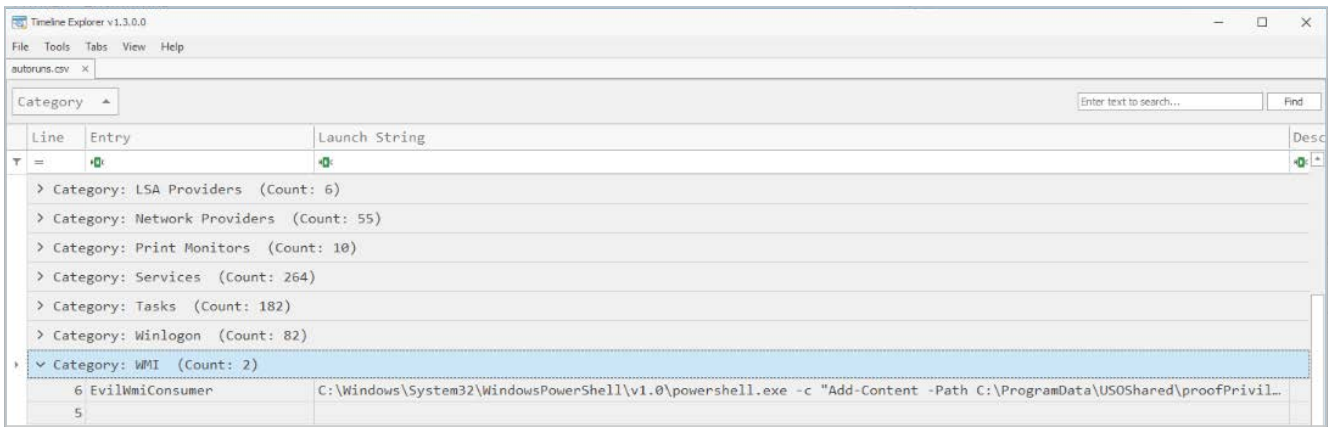
Der dritte Ansatzpunkt für ein dauerhaftes Verweilen im System ist die Windows Management Instrumentation (WMI). Sie ist eine standardisierte Schnittstelle zur Verwaltung von Windows-Systemen. Im IT-Betrieb wird sie für die Fernüberwachung und -verwaltung eingesetzt. Ihre Event-Filter und Event-Consumer können Angreifer als Persistenzmechanismus ausnutzen (siehe ix.de/z3pp): Ein Event-Consumer ruft ein Skript oder eine Applikation auf, wenn das vom Event-Filter beschriebene Ereignis eintritt. Im Beispielszenario wird ein Event-Filter verwendet, der bei Angriffen der Ragnar-Locker-Ransomware-Gruppe Anfang des Jahres 2021 beobachtet wurde. Er löste einen PowerShell-Aufruf circa 140 Sekunden nach Systemstart aus. So stellten die Angreifer sicher, dass ihre Schadsoftware nach einem Neustart wieder ausgeführt wird.

In den meisten IT-Umgebungen wird Autoruns keine oder eine nur geringe Zahl an WMI-Einträgen finden. Deshalb sollten alle unter „Category: WMI“ verzeichneten Einträge überprüft werden. Wie bei den Scheduled Tasks ist allen voran die Spalte „Launch String“ genau zu betrachten und auf auffällige Inhalte zu prüfen. Im Beispielszenario wird PowerShell direkt aufgerufen, was im IT-Betrieb sehr selten zu sehen ist (siehe Abbildung 6).

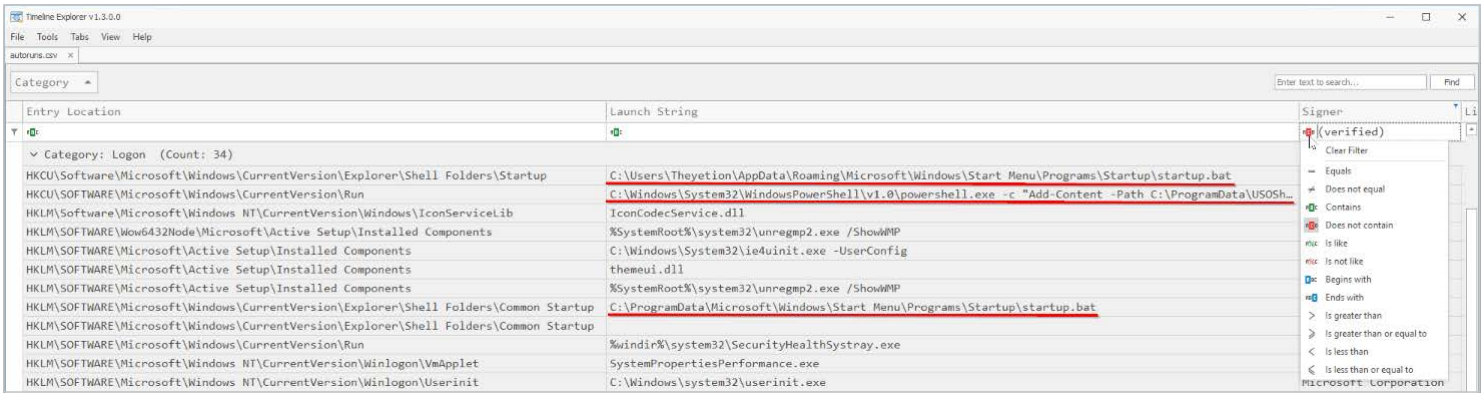
Die Einfallstore Nummer vier und fünf für um Persistenz bemühte Angreifer sind die von Autoruns als Logon-Kategorie zusammengefassten Startup-Ordner und Registry-Run-Schlüssel. Jeder Windows-Nutzer hat einen persönlichen Startup-Ordner, in dem Applikationen, Verknüpfungen, Skripte et cetera abgelegt werden können. Sie werden nach der Anmeldung des Nutzers ausgeführt. Dazu kommen ein Startup-Ordner, der für alle Nutzerkonten gilt, und mehrere Registry-Schlüssel, die denselben Effekt haben. Die Dateipfade und Registry-Schlüssel sind auf der MITRE-ATT&CK-Webseite im Detail dokumentiert (siehe ix.de/z3pp).

Startup-Ordner und Registry-Run-Schlüssel finden

Bei zeitkritischen Untersuchungen empfiehlt es sich, für den ersten Durchlauf alle Zeilen mit „(verified)“ in der Signer-Spalte



Da in den meisten IT-Systemlandschaften keine oder nur sehr wenige WMI-Autostarts zu finden sind, muss jeder Fund als potenziell schädlich eingestuft und genau untersucht werden (Abb. 6).



Diese auffälligen Startup-Einträge sollten einer näheren Untersuchung unterzogen werden (Abb. 7).

auszublenden: Dazu klickt man auf das ABC-Symbol unter dem Spaltentitel und wählt den Eintrag „Does not contain“ aus. Im benachbarten Feld trägt man „(verified)“ ein. Timeline Explorer blendet nun alle Einträge aus, auf die die Eigenschaft zutrifft.

KAPE führt Autoruns mit der Option `-s` aus, die das Tool veranlasst, die Softwaresignaturen der in der Spalte „Image Path“ aufgeführten Datei zu verifizieren. Falls eine Signatur gültig ist, kennzeichnet es sie in der besagten Spalte als verifiziert. In vielen Informationssicherheitsvorfällen hat man mit nicht oder nicht richtig signierter Schadsoftware und Skripten zu tun. Dieser Filter erleichtert und beschleunigt daher häufig die Arbeit.

Anschließend geht man wie bei den Scheduled Tasks die „Launch String“-Spalte durch und sucht nach Auffälligkeiten. Zusätzlich zu den bereits genannten, also etwa Einträgen mit powershell.exe, sollte hier auch Ausschau nach Skriptdateien gehalten werden – .bat, .vbs, .wsf und vergleichbare Dateiendungen, die in der IT-Umgebung ausführbar sind. Dieses Vorgehen offenbart die schädlichen Einträge im Beispielszenario (siehe Abbildung 7).

Findet man mit diesen Dateiformaten nichts Auffälliges, sollte der Filter entfernt und alles einer Legitimitätsprüfung unterzogen werden. Hierbei ist nach Einträgen Ausschau zu halten, bei denen Angreifer versuchen, vertrauenswürdig auszusehen – analog zum vorhergehenden Beispiel durch Verschleierung von Diensten oder durch Benutzen des Namens einer bekannten Windows-Anwendung, die aber am falschen Ort liegt.

Weitere Untersuchungen

Falls ein vergleichbares System in der IT-Umgebung existiert, kann dessen Autoruns-Resultat mit dem des untersuchten Computers verglichen werden. Unterschiede zwischen den beiden sind Kandidaten für eine nähere Untersuchung. Hier ist allerdings zu beachten, dass auch das Referenzsystem auf die gleiche Art kompromittiert sein kann.

Das Auffinden verdächtiger Autostarts benötigt vor allem viel Aufmerksamkeit und Kenntnis typischer Applikationen und ihrer Speicherorte in der eigenen IT-Landschaft. Wurde ein auffälliger Eintrag gefunden, muss er weiter analysiert werden.

Expertentricks

Falls der Fund auf eine Datei verweist, muss sie für die Analyse auf dem untersuchten Gerät lokalisiert und auf einen externen Datenträger kopiert werden. Hier ist vor allem darauf zu achten, dass diese potenzielle Schadsoftware nicht aus Versehen auf dem Analysesystem ausgeführt wird. Unter IT-Sicherheitsexperten hat es sich etabliert, solche Dateien in ein Zip-Archiv mit dem Passwort „infected“ zu packen. Das hindert die meisten Antivirus-Produkte daran, das Untersuchungsobjekt zu finden und zu löschen. Gleichzeitig sinkt die Wahrscheinlichkeit, dass die Datei durch einen Fehlklick geöffnet oder ausgeführt wird.

Kommandozeilenaufrufe und kurze Skripte lassen sich statisch analysieren. Dazu muss man in den meisten Fällen zuerst die Verschleierungsversuche der Angreifer rückgängig machen. Dazu gehört nur wenig Erfahrung in der jeweiligen Kommandozeilenumgebung oder Skriptsprache und der dazugehörigen Dokumentation. Beim Rückgängigmachen solcher Verschleierungsversuche, zum Beispiel dem Codieren von Befehlsfolgen in Base64, kann das von der britischen Behörde GCHQ entwickelte Werkzeug CyberChef (siehe ix.de/z3pp) hilfreich sein.

Alternativ oder bei längeren Skripten und Aufrufen von Applikationen lohnt sich der Einsatz einer Schadsoftwareanalyse-Sandbox. Sie führt potenzielle Malware in einer abgeschotteten Umgebung aus und beobachtet sie währenddessen. Die Sandbox zeichnet alle System- und Netzwerkaktivitäten auf und wertet sie aus. Bei den meisten kommerziellen Sandboxes offenbart die Auswertung besonders auffälliges Verhalten und liefert eine Bewertung der Schädlichkeit. Diese Informationen sind in vielen Fällen hilfreich bei der Einschätzung der ausgeführten Datei und beim Aufspüren unerwünschter Persistenzen.

Ist man sich bei einem Eintrag nicht sicher, kann das Skript oder die Applikation mit einem Antivirus-Scanner überprüft werden. Wenn man dem installierten Scanner nicht traut, ist der Microsoft Safety Scanner (zu finden über ix.de/z3pp) eine gute Alternative. Und schließlich: Führen die Resultate von Autoruns zu keinerlei Auffälligkeiten, lohnt es sich, nach Persistenzmechanismen zu suchen, die nicht auf der Autostart-Funktion basieren.

Nach diesem Ausflug in die Welt der Angriffstechniken legt der kommende Tutorialteil den Schwerpunkt auf das Verhalten von Nutzerinnen und Nutzern. Er zeigt, wie man die Browser-Historie und damit die Internetaktivitäten sammelt und analysiert. Damit geht es an den Beginn vieler Informationssicherheitsvorfälle, nachdem ein Link in einer Phishing-E-Mail angeklickt und dadurch das Ausführen von Schadsoftware angestoßen wurde.

(ur@ix.de)

Quellen

Das MITRE-ATT&CK-Framework, die Einträge zu den besprochenen Angriffen sowie die im Text genannten Skripte und Werkzeuge sind über ix.de/z3pp zu finden.

Gregor Wegberg

unterstützt mit seinem Team bei der Oneconsult AG Organisationen bei der Bewältigung von Cyberangriffen.