KAPE-Einführung, Teil 4:
Was wurde wann von wem ausgeführt?

Ein Doppelklick und … ups

Gregor Wegberg

Cyberangriffe durchlaufen mehrere Phasen. Früher oder später wird der Angreifer auf dem angegriffenen System vorhandene Programme oder eigene Schadsoftware ausführen. Das hinterlässt Spuren, die man mit KAPE aufdecken kann.

Bei der Bewältigung eines Cyberangriffs stellt sich die Frage, wie es zu einer Infektion des Systems kommen konnte. Und auch, was schon alles vor dem jetzt sichtbaren Schaden passiert ist. Beim Beantworten geht es nicht um reine Neugierde oder um Schuldzuweisungen, vielmehr ist es essenzieller Bestandteil einer gelungenen Vorfallsbewältigung. Je besser man den Ablauf eines Angriffs vom initialen Zugriff bis zur sichtbaren Auswirkung versteht, desto wahrscheinlicher ist die erfolgreiche und nachhaltige Behandlung des Vorfalls. Jeder aufgedeckte Schritt liefert wichtige Hinweise. Damit können kompromittierte Systeme isoliert und gefährdete geschützt werden.

Erst das Aufdecken dieser Abläufe ermöglicht zielgerichtete Schutzmaßnahmen. Diese wirken idealerweise bereits vor dem Eintritt des Schadens. Im Nachgang kann die Erkennung zukünftiger Angriffe anhand der entdeckten Abläufe ausgebaut und verbessert werden. Erfolgreiche Cyberangriffe zeigen Schwächen in der eigenen IT-Sicherheit auf. Sie sollten als Chance zur dauerhaften Verbesserung begriffen werden. Besonders eignen sie sich als konkrete Fallbeispiele für die Benutzersensibilisierung.

Tutorialinhalt

Wie im vorhergehenden Teil dieses Tutorials gezeigt, müssen sich Angreifer zunächst Zutritt zu einem ersten System verschaffen, entweder durch den Fernzugriff mit einem kompromittierten Nutzerkonto, durch Infizieren des Computersystems mit einer Schadsoftware oder durch Ausnutzen einer Sicherheitsschwachstelle. Dieses Vorgehen gilt nicht nur für die Erstinfektion. Auch die weitere Ausbreitung im Unternehmensnetzwerk basiert darauf.

Falls der Angreifer den Fernzugriff oder eine Schwachstelle nutzt, wird er Schadsoftware nachladen, sie dort festsetzen (persistieren) und ausführen oder versuchen, mit bereits installierten Anwendungen mehr über das System und seine Umgebung zu erfahren. Geschieht die Infektion mithilfe eines vermeintlich vertrauenswürdigen Dokuments, muss dieses zuerst mit einer Applikation geöffnet oder, falls es selbst ein Programm ist, gestartet werden. In all diesen Fällen wird Software ausgeführt, was auf einem Windows-System diverse Spuren hinterlässt. Mit KAPE kann man diese Spuren aufdecken und damit mehr über den Angriffsablauf erfahren.

Beweise einsammeln

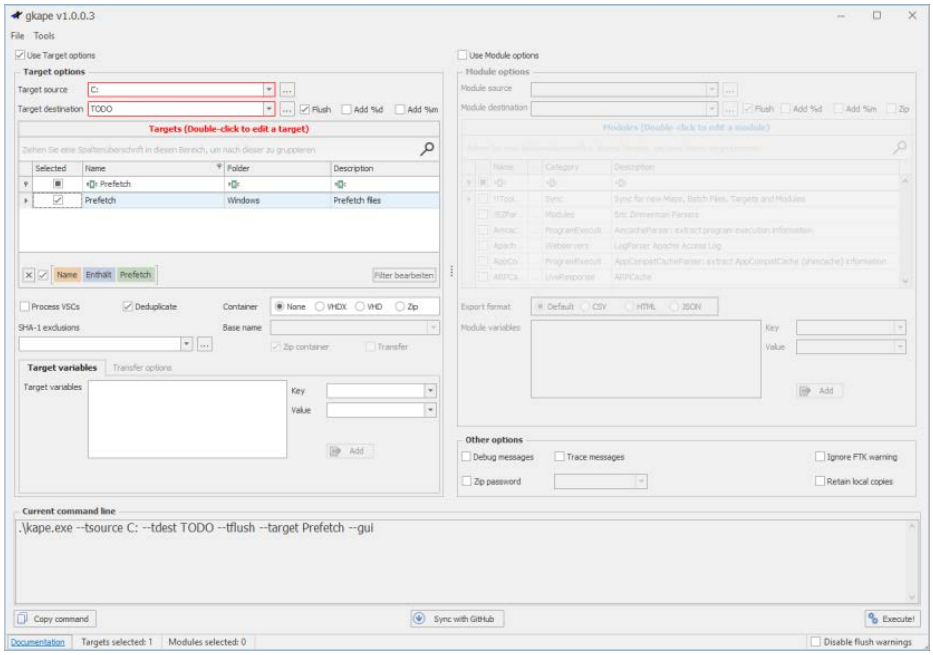
Das Windows Forensic Analysis Poster des SANS Institute (siehe ix.de/zb7e) verzeichnet unter Program Execution neun Typen forensischer Artefakte, die Hinweise auf Programmausführungen geben. Einer von ihnen, die Prefetch-Dateien, sollen nachfolgend einer vertieften Betrachtung unterzogen werden. Sie liefern besonders ausführliche Informationen zu ausgeführten Applikationen.

Zum Einsammeln der relevanten Dateien steht das Prefetch-Target in KAPE bereit. Alternativ kann man das Compound-Target EvidenceOfExecution einsetzen, das weitere Artefakte sammelt, die auf das Ausführen von Anwendungen hindeuten.

Wie in den vorangegangenen Teilen des Tutorials beschrieben, ist KAPE zuerst auf einem Analysesystem vorzubereiten: Es wird auf einen externen Datenträger kopiert und aktualisiert. Für die Targets werden keine Drittanwendungen benötigt, wodurch das Herunterladen und Einrichten entfällt. Der Kommandozeilenbefehl für das Sammeln der Daten wird anschließend auf dem Analysesystem zusammengestellt.

Dazu startet man die grafische Oberfläche von KAPE, klickt „Use Target options“ an, setzt „Target source“ auf die Systempartition des zu untersuchenden Systems – also sehr wahrscheinlich C: –, trägt bei „Target destination“ einen Platzhalter ein und wählt das Target aus (siehe Abbil-

Beispielkonfiguration für das Sammeln der forensisch relevanten Dateien, mit denen sich die Applikationsausführung nachverfolgen lässt (Abb. 1)



dung 1). Den Kommandozeilenbefehl unter „Current command line“ speichert man für die Ausführung auf dem zu untersuchenden System und schließt KAPE.

Das externe Laufwerk wird nun an das zu untersuchende System angeschlossen, eine administrative Konsole gestartet, zum Beispiel PowerShell, und damit in den KAPE-Ordner auf dem externen Datenträger gewechselt. Danach fügt man den vorbereiteten Kommandozeilenaufruf in die Kommandozeile ein und ersetzt den Platzhalter für die „Target destination“. Im vorliegenden Beispiel wurde der externen Festplatte der Laufwerksbuchstabe F zugeordnet. Entsprechend wird der Platzhalter durch einen gültigen Ordnerpfad für die zu sammelnden Dateien ersetzt:

```
.\kape.exe --tsource C: --tdest F:\TargetDestination\ --tflush --target Prefetch -gui
```

Jetzt kann das Sammeln durch die Ausführung des Befehls starten. Anschließend finden sich im Zielordner (F:\TargetDestination\) die zusammenkopierten Daten sowie drei Protokolldateien.

Für die nun anstehende Verarbeitung und Analyse der gesammelten Dateien wird der externe Datenträger wieder an das Analysesystem angeschlossen.

Softwarebeschleunigung als forensisches Artefakt

Prefetch-Artefakte entstehen durch die Aktivität des Windows Cache Managers. Während des Boot-Prozesses oder bei der Ausführung einer Anwendung überwacht er, welche Daten von der Festplatte in den Zwischenspeicher geladen werden und auf welche Ordner die Anwendung zugreift. Daraus erzeugt er pro Applikation eine Prefetch-Datei mit der Endung .pf im Ordner C:\Windows\Prefetch. Jede dieser Dateien hält fest, auf welche Ordner und Dateien eine Anwendung bis zu zehn Sekunden nach ihrem Start zugreift.

Da sie diese Ordner und Dateien mit hoher Wahrscheinlichkeit bei der nächsten Ausführung wieder benötigt, können sie dann bei einem erneuten Start gleich zu Beginn in den Zwischenspeicher geladen werden, bevor die Applikation sie anfordert. Das beschleunigt den Aufruf der Anwendung signifikant. Ein Großteil der benötigten Informationen ist zum Zeitpunkt des Zugriffs bereits im Zwischenspeicher geladen. Auf Endnutzersystemen ist diese Optimierung seit Windows XP aktiviert. Auf Windows-Server-Systemen hingegen ist sie im Normalfall deaktiviert. Seit Windows 8 verwaltet der Cache Manager bis zu 1024 Prefetch-Dateien.

Der Dateiname der Prefetch-Dateien besteht aus dem Namen der Binärdatei, einem Bindestrich und dem Hash des Dateipfads zur ausgeführten Anwendung. In spezifischen Fällen berechnet sich der Hash aus den Kommandozeilenoptionen anstelle des Dateipfads. Dieser Sonderfall ist beispielsweise für die Beschleunigung von Windows-Diensten (svchost.exe) und anderen Host-Applikationen (dllhost.exe, rundll32.exe, mmc.exe und backgroundtaskhost.exe) notwendig: In diesen Fällen ist nicht die Anwendung selbst, sondern deren jeweilige Kommandozeilenoption entscheidend. Sie beeinflusst, was ausgeführt wird und was es zu beschleunigen gilt.

Prefetch-Dateien sind für forensische Untersuchungen aufgrund ihres hohen Informationsgehalts besonders interessant. Die Prefetch-Datei wird ungefähr 10 Sekunden nach der ersten Ausführung der jeweiligen Applikation erstellt. Der Zeitpunkt der letzten Änderung dieser Datei liegt also bis zu 10 Sekunden nach dem letzten Anwendungsstart. Handelt es sich um eine größere Anwendung, beträgt die Abweichung circa 10 Sekunden. Bei einer kleinen Anwendung, zum Beispiel der cmd.exe, die keine 10 Sekunden läuft, weicht der tatsächliche Start von der Prefetch-Dateierzeugung und -Dateiänderung um weniger als 10 Sekunden ab.

Der Inhalt der Prefetch-Datei hält den präzisen Startzeitpunkt der Anwendung ebenfalls fest. Seit Windows 8 enthält sie die Zeitpunkte der letzten acht Ausführungen, wie häufig die Anwendung gestartet wurde und die bereits angesprochene Liste von Dateien und Ordnern, auf die das Programm in den ersten circa 10 Sekunden zugegriffen hat. Damit lassen sich bis zu neun Ausführungszeitpunkte feststellen: die erste bekannte Ausführung (Erzeugungszeitpunkt der Prefetch-Datei minus circa 10 Sekunden) und die acht präzisen Ausführungszeitpunkte in der Prefetch-Datei.

Durch die Begrenzung auf 1024 solche Dateien kann es sein, dass die Prefetch-Datei der ersten Ausführung inzwi-

X-TRACT

- Die zielgerichtete und erfolgreiche Bekämpfung eines Cybervorfalls erfordert ein möglichst genaues Verständnis der Angriffsschritte.
- Im Verlauf eines Angriffs führen Cyberkriminelle sowohl eigene als auch Standardapplikationen aus.
- Software hinterlässt bei der Ausführung forensisch auswertbare Spuren. Sie geben Einblick in den Ablauf eines Angriffs und dienen damit der Aufklärung des Vorfalls, aber auch der Prävention.

schen gelöscht wurde und man eine später erzeugte untersucht. Daher sprechen Forensiker beispielsweise vom ersten bekannten Start und nicht von der ersten Ausführung.

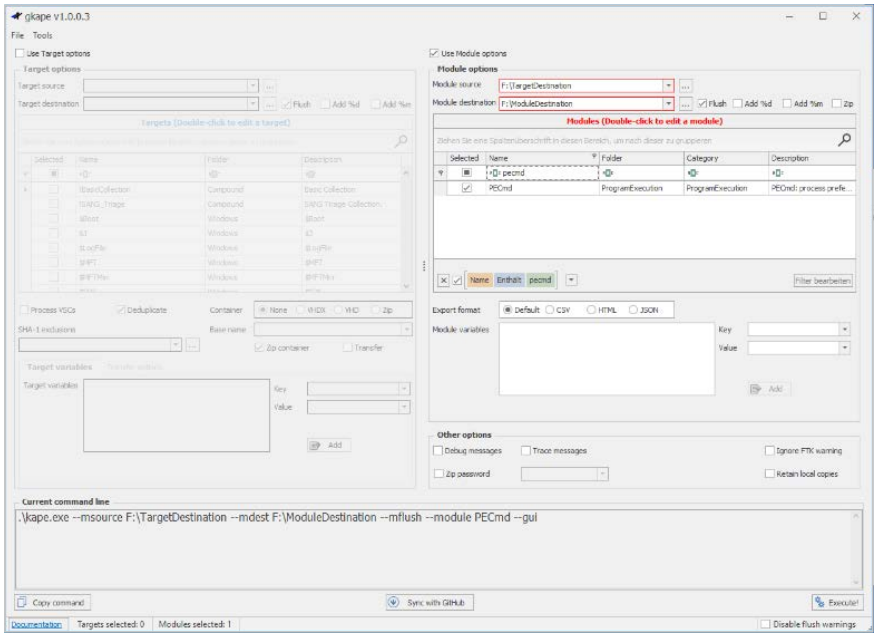
Prefetch lesbar machen

Wie gewohnt müssen die mit KAPE gesammelten Prefetch-Dateien zuerst durch ein Modul in ein für den Forensiker lesbares Format überführt werden. Dazu startet man die grafische Oberfläche von KAPE und wählt „Use Module options" aus. Der Zielordner der vorherigen Dateisammlung (entspricht „Target destination") wird als „Module source" konfiguriert. Bei „Module destination" wird ein neuer Ordner für die Resultate der Auswertung gewählt. Das von Eric Zimmerman geschriebene Werkzeug PECmd ruft das gleichnamige Modul auf, das die Verarbeitung durchführt (siehe ix.de/zb7e). Es kommt bereits mit KAPE mit und muss nicht getrennt heruntergeladen werden. Nach dem Auswählen dieses Moduls sind alle Einstellungen vorgenommen (siehe Abbildung 2) und die Auswertung wird mit dem Execute-Knopf gestartet.

Das Resultat der Auswertung liegt abschließend als PECmd_Output.csv und PECmd_Output_Timeline.csv im Unterordner ProgramExecution des Module-destination-Ordners. Beide CSV-Dateien werden, wie bei den vorhergehenden Analysen dieser Tutorialreihe, für die Untersuchung im Timeline Explorer geöffnet.

Auffälliges im Anwendungszeitstrahl

Die PECmd_Output_Timeline.csv gibt einen Überblick, wann welche Applikation ausgeführt wurde. Für eine bessere Übersicht wählt man im Kontextmenü einer beliebigen Spaltenüberschrift den Eintrag „Optimale Breite (alle Spalten)" aus. Anschließend sortiert man die Einträge in zeitlich absteigender Folge durch zwei Klicks auf die „Run Time"-Spalte oder durch



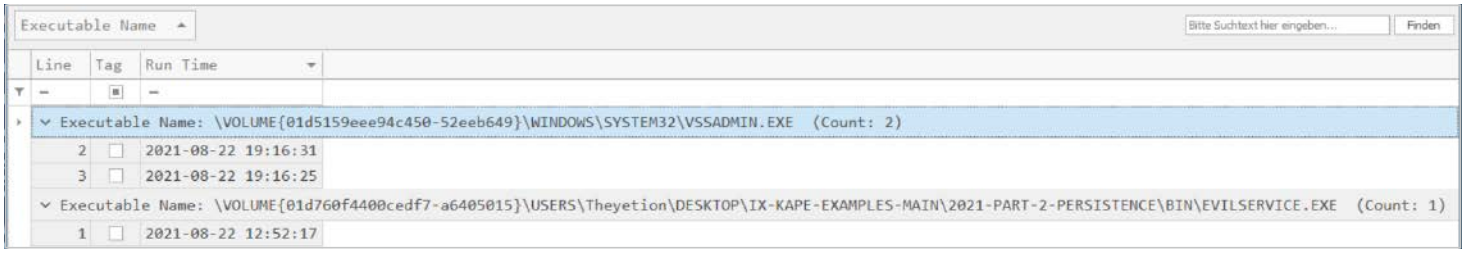
Mit Eric Zimmermans PECmd kann man die gesammelten Prefetch-Dateien in ein für Menschen analysierbares Format überführen (Abb. 2).

Wahl des entsprechenden Kontextmenüeintrags.

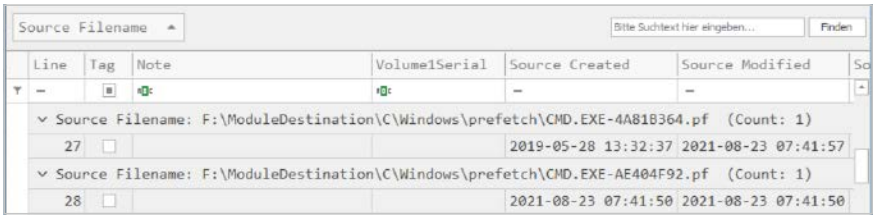
Für einen ersten Überblick gruppiert man die Zeilen nach der „Executable Name"-Spalte. Im Kontextmenü der Spaltenüberschrift wählt man hierzu den Eintrag „Nach dieser Spalte gruppieren" aus. Danach geht der Analyst alle Applikationen (Executable Name) durch und hält Ausschau nach verdächtigen Einträgen. Verdächtig sind beispielsweise Anwendungen, die einen unüblichen Dateipfad haben, und Kommandozeilenanwendungen, die von den Nutzern selten bis nie aufgerufen werden sollten oder ein privilegiertes Nutzerkonto benötigen. Interessant ist auch der Start von Kommandozeileninterpretern (cmd.exe, powershell.exe), Skriptlaufzeitumgebungen (cscript.exe, wscript.exe), Applikationen im Zusammenhang mit Persistenzmechanismen (siehe Teil 2 dieser Tutorialreihe [1]) und der Aufruf sogenannter Living Off The Land Binaries and Scripts (LOLBAS) vom Typ Binaries (siehe ix.de/zb7e).

LOLBAS ist eine Sammlung von Applikationen und Skripten, die Teil des Windows-Betriebssystems sind oder von Microsoft heruntergeladen werden können. Sie sind von Microsoft signiert und verfügen neben ihrer eigentlichen Bestimmung auch über Funktionen, die für Angreifer von Nutzen sind. Eine klassische LOLBAS-Anwendung ist certutil.exe, mit der unter anderem Dateien aus dem Internet heruntergeladen und Base64-decodiert werden können (siehe ix.de/zb7e). Weil die Anwendungen Teil des Betriebssystems oder zumindest von Microsoft signiert sind, fallen sie deutlich weniger auf als Programme, die ein Angreifer auf das System überträgt und dort ausführt. Ihre Signatur ermöglicht häufig das Umgehen von Schutzmechanismen, allen voran das Anwendungs-Whitelisting.

Auf dem hier untersuchten System fallen besonders zwei Applikationen auf (siehe Abbildung 3): Auf dem Gerät wurde das Persistenz-Beispielszenario aus Teil 2 dieser Tutorialreihe ausgeführt. In der Liste der ausgeführten Anwendungen sieht man den Start des bösartigen Dienstes (EvilService.exe). Auffällig ist auch die Ausführung von vssadmin.exe, einer Komman-



Zwei besonders auffallende Applikationen wurden ausgeführt: eine selten genutzte Systemanwendung, die häufig im Zusammenhang mit Ransomware-Angriffen steht, und eine auffällige schädliche Applikation, die sich festgesetzt hat (Abb. 3).



Zweimal wurde eine Anwendung gleichen Namens aufgerufen. Die Hashwerte im Prefetch-Dateinamen deuten auf zwei cmd.exe-Dateien in unterschiedlichen Ordnern hin (Abb. 4).

dozeilenanwendung zum Verwalten der Volume-Schattenkopien (siehe ix.de/zb7e). Sie wird selten oder nie auf einem Endnutzersystem genutzt und benötigt ein privilegiertes Nutzerkonto. Schadsoftware, vor allem im Zusammenhang mit Ransomware-Angriffen, wird hingegen diese Applikation nahezu immer aufrufen. Mit ihr werden vor dem Verschlüsseln der Daten alle Volume-Schattenkopien gelöscht (siehe ix.de/zb7e). Beide Ausführungen müssen zwingend näher untersucht werden.

Bei der Betrachtung des Zeitstrahls können Aufrufe mehrerer aufeinanderfolgender Standardanwendungen ein Hinweis auf unerwünschte Aktivität sein. Angreifer nutzen diese Programme auch im Sinne ihrer eigentlichen Funktion: Mit net.exe, ipconfig.exe, netstat.exe et cetera sammeln sie Informationen zum Windows-Netzwerk. ver.exe, whoami.exe, systeminfo.exe, tasklist.exe und weitere liefern wiederum Hinweise zum Windows-System. Eine zeitliche Häufung der Ausführung solcher Anwendungen sollte genauer analysiert werden. Ihre Ursache kann ein IT-Administrator sein, der einem Problem nachgeht, aber auch ein Angreifer, der mehr über die Umgebung erfahren möchte.

Vertiefter Blick in die Ausführung

Nachdem erste auffällige Anwendungsstarts entdeckt wurden, geht es als Nächstes an die detaillierte Auswertung der Prefetch-Daten. Dazu öffnet man PECmd_Output.csv im Timeline Explorer. Für eine bessere Übersicht wählt man im Tools-Menü den Eintrag „Reset column widths“ (Strg + R), der alle Spaltenbreiten auf einen vordefinierten Maximalwert setzt.

Bei der Suche nach verdächtigen Anwendungsaufrufen beginnt man beispielsweise in der „Source Filename“-Spalte. Sie gibt an, aus welcher Prefetch-Datei die Informationen in der jeweiligen Zeile stammen. Wie einleitend beschrieben besteht der Dateiname aus dem Namen der aufgerufenen Anwendung und einem Hash. Taucht der gleiche Anwendungsname mit unterschiedlichen Hashwerten auf, ist das ein mögliches Zeichen dafür, dass die Anwendungen in unterschiedlichen Ordnern liegen. Das wiederum kann auf den Versuch hindeuten, eine Schadsoftware unter dem Deckmantel einer Standardanwendung laufen zu lassen.

Durch das Gruppieren und Sortieren der „Source Filename“-Spalte lassen sich solche Fälle schnell identifizieren. Auf dem hier untersuchten System fallen zwei cmd.exe-Aufrufe mit unterschiedlichen Hashes auf (siehe Abbildung 4). Hat man einen solchen Fall entdeckt, ist der nächste Schritt, den Inhalt in der „Files Loaded“-Spalte der verdächtigen Zeilen näher zu betrachten. In dieser Spalte sind die vom Cache Manager beobachteten Dateizugriffe festgehalten – darunter auch der Dateipfad zur gestarteten cmd.exe. Mithilfe dieser Informationen lässt sich sofort erkennen, dass eine der aufgerufenen cmd.exe nicht die bekannte Windows-Kommandozeile ist, sondern eine andere, vermutlich unerwünschte Anwendung (siehe Abbildung 5). Dieses Programm sollte einer vertieften Untersuchung unterzogen werden.

Zellen mit langem Inhalt lassen sich in der Tabellenansicht des Timeline Explorer nur schwer lesen. Dies gilt im Besonderen in den Spalten „Files Loaded“ und „Directories“. Um den Text besser lesbar zu machen, klickt man zweimal auf die gewünschte Zelle: Es öffnet sich das „Cell content“-Fenster. Am unteren Rand dieses Fensters stehen Optionen zur automatischen Formatierung des angezeigten Inhalts zur Verfügung. In beiden Spalten werden die Datei- und Ordnerpfade durch Kommata getrennt. Für bessere Lesbarkeit wählt man daher den „Comma“-Eintrag und betätigt anschließend den Format-Knopf. Dies führt dazu, dass jeder Datei- und Ordnerpfad in einer eigenen Zeile steht. Damit wird der gesamte Inhalt einfacher lesbar.

Verdächtige Dateien aufspüren

Hat man noch sehr wenige Informationen zum Angriff oder ist auf der Suche nach einer bestimmten Datei, kommt erneut die „Files Loaded“-Spalte ins Spiel. Darin kann man nach geöffneten Office-Dokumenten und ausgeführten Skripten suchen. Zur Beschleunigung der Suche öffnet man das Kontextmenü der „Files Loaded“-Spalte und wählt den Menüeintrag „Filter bearbeiten“ aus. Im sich öffnenden Fenster kann man in einem Arbeitsgang nach mehreren typischen Dateiendungen oder -namen filtern. Der folgende Filter sucht beispielsweise nach drei häufig auftretenden Skriptdateien (JavaScript, PowerShell und Visual Basic Script) anhand ihrer Endungen:

```
Contains([Files Loaded], '.js') ⅀
          Or Contains([Files Loaded], '.ps1') ⅀
            Or Contains([Files Loaded], '.vbs')
```

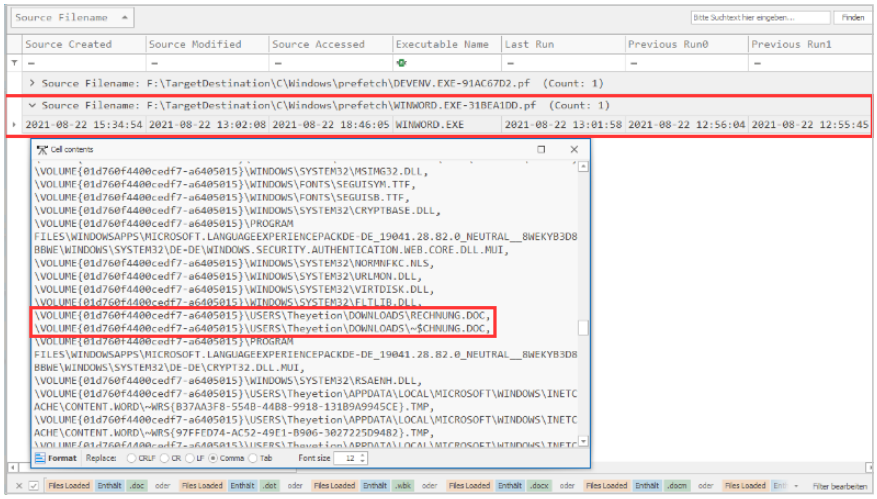
Ein ähnlicher Filter lässt sich auch für Microsoft-Word-Dokumente schreiben:

```
Contains([Files Loaded], '.doc') ⅀
          Or Contains([Files Loaded], '.dot') ⅀
          Or Contains([Files Loaded], '.wbk') ⅀
         Or Contains([Files Loaded], '.docx') ⅀
         Or Contains([Files Loaded], '.docm') ⅀
         Or Contains([Files Loaded], '.dotx') ⅀
         Or Contains([Files Loaded], '.dotm') ⅀
           Or Contains([Files Loaded], '.docb')
```

Mit solchen Filtern kommt man verdächtigen Dateien schnell näher. Die auf der Webseite filesec.io des gleichnamigen Projekts gelisteten Dateiendungen der Kategorien Executable oder Phishing sind für eine solche Suche besonders interessant. Nach dem Aktivieren des Filters empfiehlt es sich, die verbleibenden Einträge

Der Inhalt der Prefetch-Datei des linken cmd.exe-Aufrufs führte den Standardbefehlsinterpreter von Windows aus. Dagegen wurde beim rechten Aufruf eine Anwendung mit gleichem Namen gestartet, die aber einen untypischen Dateipfad hat. Der Aufruf auf der rechten Seite muss genauer untersucht werden (Abb. 5).

Die Vermutung bestätigt sich, dass das heruntergeladene Word-Dokument mit Microsoft Word geöffnet wurde (Abb. 6).



nach der „Executable Name“-Spalte zu gruppieren. Hält man Ausschau nach unerwünschten Skripten, so werden vor allem Aufrufe von cmd.exe, powershell.exe, wscript.exe et cetera von Interesse sein. Interessiert man sich hingegen für das Öffnen von Office-Dokumenten, sucht man nach Ausführung der jeweiligen Office-Anwendungen, zum Beispiel winword.exe für Word-Dokumente.

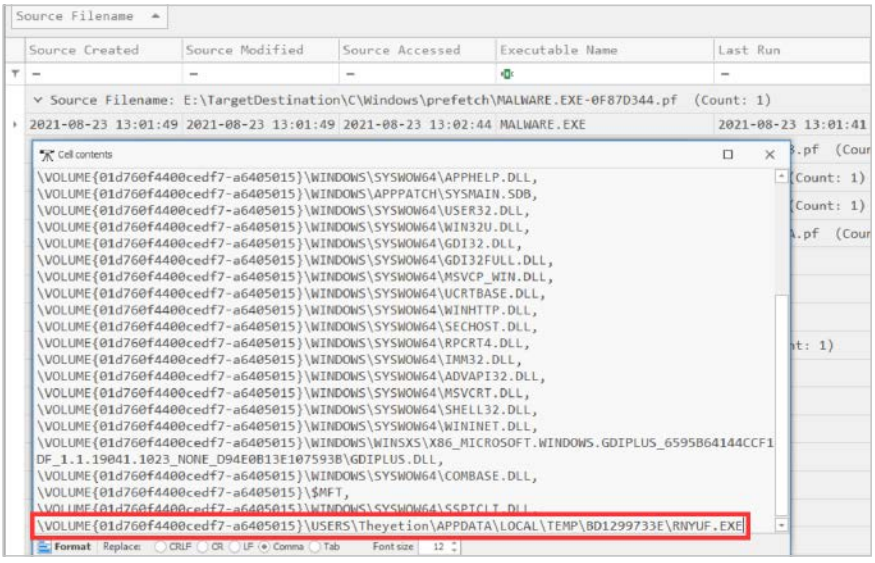
Auf dem untersuchten System wurde ein Word-Dokument heruntergeladen. Das konnte mithilfe der Browserhistorie bereits festgestellt werden. Anhand der Prefetch-Dateien möchte man nun versuchen herauszufinden, ob und wann dieses Dokument geöffnet wurde. Im vorliegenden Beispiel wurde der vorherige Word-Dokumente-Filter genutzt. Alternativ hätte man auch direkt nach dem aus der Browserhistorie bekannten Namen suchen können.

Nun wird der Inhalt der „Files Loaded“-Zelle der ausgeführten winword.exe-Anwendung betrachtet. Sie zeigt, dass das heruntergeladene Dokument mit sehr hoher Wahrscheinlichkeit tatsächlich geöffnet wurde: zuletzt und mindestens einmal am 22. August 2021 um circa 13:01 Uhr (siehe Abbildung 6). Nicht beantworten lässt sich die Frage, ob die Datei bereits zu einem früheren Zeitpunkt geöffnet wurde, zum Beispiel am selben Tag um circa 12:56 Uhr oder 12:55 Uhr, als Word ebenfalls lief.

Was die Malware tat

Wurde eine Schadsoftware gestartet und die dazugehörige Prefetch-Datei sichergestellt, können die vom Cache Manager aufgezeichneten Ordner- und Dateizugriffe wertvolle Hinweise liefern. Auf dem hier untersuchten System wurde ein Exemplar der Amadey-Schadsoftware ausgeführt (malware.exe) (siehe ix.de/zb7e). Wie man dem „Files Loaded“-Inhalt entnehmen kann, hat die ausgeführte Datei auf die Anwendung rnyuf.exe im Temp-Ordner eines Nutzers zugegriffen (siehe Abbildung 7). Es ist wahrscheinlich, dass die Schadsoftware selbst sie dort abgelegt und möglicherweise ausgeführt hat.

Falls es zu einer Ausführung kam, müsste eine Prefetch-Datei für rnyuf.exe existieren. Filtert man nach diesem Wert in der „Executable Name“-Spalte, findet sich tatsächlich eine solche Datei. Diese Anwendung greift wiederum auf cmd.exe und schtasks.exe zu (siehe Abbildung 8). Letzteres ist die Kommandozeilenanwendung zum Verwalten der Windows-Aufgabenplanung. Man kann also vermuten, dass rnyuf.exe sie als Persistenzmechanismus nutzt. Der Speicherort der rnyuf.exe und die Aufrufe von cmd.exe und schtasks.exe sind starke Hinweise auf eine unerwünschte Anwendung. Sie sollten vertieft analysiert werden. Auch lohnt sich die Untersuchung der Aufgabenplanung, wie im zweiten Teil dieser Tutorialreihe vorgestellt. Parallel kann bereits auf allen Systemen im Unternehmen nach der rnyuf.exe gesucht werden. Geräte, auf denen man diese Anwendung findet, sind als infiziert zu betrachten.



Wer wars?

Der Cache Manager hält leider nicht fest, welches Nutzerkonto eine Anwendung gestartet hat. Damit gibt es keine direkte, weitgehend vertrauenswürdige Information in den Prefetch-Dateien zum ausführenden Konto. In den meisten Fällen lässt sich trotzdem vermuten, welches Nutzerkonto das letzte Mal eine Anwendung gestartet hat. In der Liste der Ordner und Dateien, auf die zugegriffen wurde, finden sich häufig Ordnerpfade, die den Nutzernamen der betroffenen Person enthalten: Im vorherigen Beispiel mit der verdächtigen cmd.exe (siehe Abbildung 5) lag die Anwendung im Temp-Ordner eines Nutzerkontos (Theyetion). Ein vergleichbares Verhalten hat die ausgeführte Schadsoftware gezeigt: Sie platzierte die rnyuf.exe im selben Ordner (siehe Abbildung 7). Das verdächtige Word-Dokument wurde aus dem Internet heruntergeladen und lag im Standard-Downloads-Ordner eines Nutzerkontos (siehe Abbildung 6).

In vielen Fällen lässt sich also allein anhand der Prefetch-Daten herleiten, welches Nutzerkonto die letzte Ausführung wahrscheinlich gestartet hat. Nutzt man

Das Ausführen einer Schadsoftwareprobe auf dem untersuchten System hat mit hoher Wahrscheinlichkeit die rnyuf.exe hinterlassen, die möglicherweise gestartet wurde (Abb. 7).

weitere forensische Daten, zum Beispiel die restlichen Program-Execution-Artefakte auf dem Windows Forensic Analysis Poster des SANS Institute, lässt sich diese Vermutung noch untermauern.

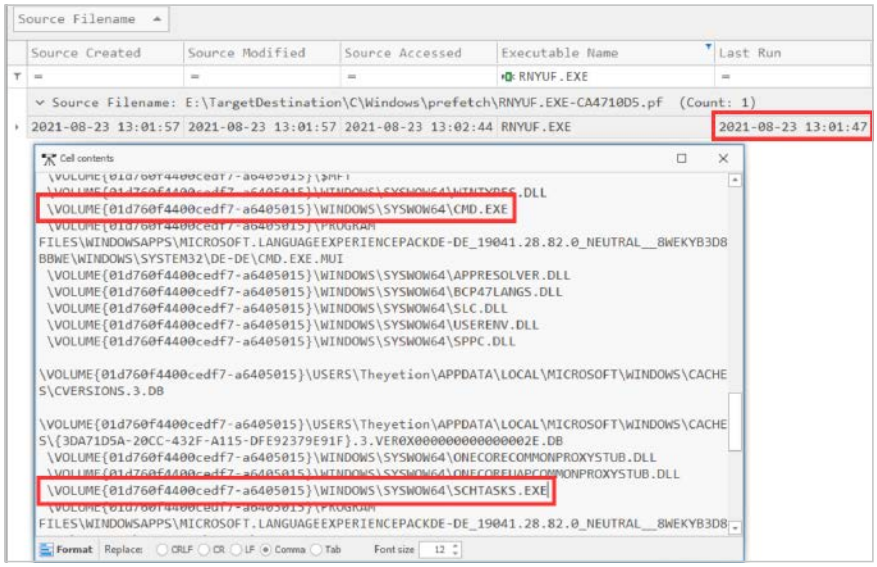
Risiko Live-Forensics

Im ersten Teil dieser Tutorialreihe wurde bereits auf die Risiken einer Live-Forensics-Untersuchung hingewiesen. Diese führt zu Änderungen am untersuchten System und kann so versehentlich Spuren verwischen. Aus diesem Grund führen Forensiker soweit möglich jegliche Vorbereitung und Untersuchung auf einem Analysesystem durch. Beim Prefetch-Artefakt wird klar, warum dies so wichtig ist: Die Ausführung von KAPE erzeugt, wie alle anderen Anwendungen auch, mindestens eine Prefetch-Datei. Sind auf dem System bereits 1024 dieser Dateien vorhanden, so wird durch den Start von KAPE eine andere, potenziell wichtige Prefetch-Datei gelöscht. Die Nutzung weiterer Module der „LiveResponse"-Kategorie erzeugt weitere Prefetch-Dateien – für jede aufgerufene Drittanwendung mindestens eine.

Die gelöschten Prefetch-Dateien sind aber nicht zwingend verloren. Sie können häufig mithilfe von Carving, einer Form der Datenrettung, wiederhergestellt werden. Dazu wird im Speicher, den das Dateisystem nicht verwendet, nach den gelöschten Dateien gesucht. Diesen Umstand gilt es vor allem bei forensischen Untersuchungen außerhalb eines Cybervorfalls zu berücksichtigen.

Vertiefte Untersuchung

Während der Auswertung der Prefetch-Dateien erzeugt der Analyst eine Liste verdächtiger Anwendungen und Dateien. Sie sind alle tiefergehend zu untersuchen, um weitere Schlüsse zum Ablauf des Angriffs zu ziehen.



Eine verdächtige Anwendung ruft cmd.exe und schtasks.exe auf. Sie lief zuletzt, kurz nachdem malware.exe gestartet wurde (Abb. 8).

Mutmaßliche Schadsoftware, hier die Dateien EvilService.exe, cmd.exe im Temp-Ordner und die rnyuf.exe, sollte unter Anwendung von Schutzmaßnahmen gesichert und zum Beispiel in einer Schadsoftware-analyse-Sandbox untersucht werden. Der zweite Teil dieser Tutorialreihe geht vertieft darauf ein. Gleiches gilt für die verdächtigen Skriptdateien und Dokumente (hier Rechnung.doc).

Auffällige Aufrufe von Standardanwendungen, zum Beispiel schtasks.exe und cmd.exe, müssen häufig mithilfe anderer forensischer Artefakte untersucht werden. In diesen Fällen stützt sich der Analyst auf die Windows-Ereignisprotokolle, anwendungsspezifische Hinweise (beispielsweise die Befehlshistorie der cmd.exe) und die Untersuchung des sichergestellten Zwischenspeichers (Memory-Forensik). Bei der Nachverfolgung von Cybervorfällen wird nie nur ein einziges forensisches Artefakt oder nur eine Untersuchungsmethode verwendet. So weit wie möglich werden immer mehrere kombiniert. Das erlaubt ein tieferes Verständnis des Ereignisses. Außerdem lassen sich so die Resultate der Analyse anderer Artefakte bestätigen. (ur@ix.de)

Quellen

[1] Gregor Wegberg; KAPE-Einführung; Teil 2: Autoruns-Artefakte auswerten und verstehen; *iX* 8/2021, S. 100

[2] Die im Text angesprochenen Werkzeuge und Artikel sind über ix.de/zb7e zu finden.

Gregor Wegberg

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