KAPE-Einführung, Teil 3: Browserhistorie auswerten und verstehen

Auf der Suche nach der Quelle

Gregor Wegberg

Viele Angriffe beginnen mit einem Link zu einer manipulierten Website. Die Auswertung der Browserhistorie hilft nachzuvollziehen, wann und wo der Angriff startete.

Angreifer nutzen gerne die Gutgläubigkeit von Menschen aus. Damit verleiten sie sie zur Preisgabe vertraulicher Informationen oder zu unerwünschten Handlungen. Solche Social-Engineering-Angriffe sind häufig der Beginn eines Cybervorfalls. Unter einem Vorwand wird der Benutzer zum Besuch einer Phishingwebseite oder zum Ausführen von Schadsoftware motiviert. Der dritte Teil des Tutorials zeigt an einem realen Ransomware-Fall, wie man mit KAPE und NirSofts BrowsingHistoryView den Besuch solcher Internetseiten und das Herunterladen von Schadsoftware nachvollziehen kann.

Für einen erfolgreichen Angriff müssen sich Angreifer Zutritt zu einem ersten System verschaffen. Von diesem aus können sie sich in der Umgebung weiter ausbreiten. Diese initiale Kompromittierung kann auf drei Wegen erfolgen: durch Ausnutzen eines kompromittierten Nutzerkontos für die Anmeldung an einem Fernzugriffsdienst, durch Infizieren eines Endnutzergeräts mit Schadsoftware oder durch Ausnutzen einer Sicherheitsschwachstelle in einem aus dem Internet erreichbaren System. Social-Engineering-Angriffe eignen sich besonders gut für die ersten beiden Eintrittswege.

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?

Auf die falsche Seite gelockt

Unter einem Vorwand werden beispielsweise Mitarbeiter dazu bewogen, ihre Nutzerdaten preiszugeben. Der Angreifer erzeugt hierzu einen Klon eines aus dem Internet erreichbaren und vertrauten Anmeldeformulars des Unternehmens, zum Beispiel des Webmail-Log-ins. Anschließend gibt er sich als Mitarbeiter der IT-Abteilung aus und bittet die Kollegen, sich zum Anstoßen einer Datenmigration beim neuen Webmail-Dienst auf der vermeintlich vertrauenswürdigen Webseite anzumelden. Mit einer Phishing-E-Mail kann dieser Angriff eine große Masse von Mitarbeitenden ansprechen. Zur Steigerung der Erfolgsaussichten können mit der gleichen Geschichte Mitarbeiter telefonisch kontaktiert werden. Diesen Angriff bezeichnet man als Vishing (Voice Phishing).

Analog dazu verlaufen viele Schadsoftwareinfektionen. Der Mitarbeiter wird zum Beispiel dazu verleitet, ein vermeintliches PDF-Dokument herunterzuladen. Die heruntergeladene Datei ist aber eine Anwendung. Beim Öffnen führt der Nutzer unbewusst die Schadsoftware aus und infiziert damit das eigene System.

Beide Angriffsformen sind in vielen Fällen der Beginn einer Krise. Es kann zum Diebstahl vertraulicher Daten oder zur Verschlüsselung aller Systeme kommen. In beiden Szenarien wird eine vom Angreifer kontrollierte Adresse mit einem Webbrowser besucht. Dies hinterlässt, ebenso wie das „normale“ Surfen im Internet, eine Fülle forensischer Daten auf dem System. Allem voran generieren die Aktivitäten eine Reihe von Artefakten, die sich mit KAPE sammeln und auswerten lassen.

Browserartefakte einsammeln

Targets der Kategorie Browser (im KAPE-Ordner unter Targets/Browsers) sammeln die forensisch interessanten Dateien unterschiedlicher Browser. Das im ersten Teil des Tutorials angesprochene Compound-Target WebBrowsers fasst sieben dieser Targets zusammen: Damit werden die forensischen Artefakte von Google Chrome, Microsoft Edge Legacy, Microsoft Edge (Chromium-basiert), Microsoft Internet Explorer, Mozilla Firefox, Opera und Puffin Secure Browser vom untersuchten System kopiert.

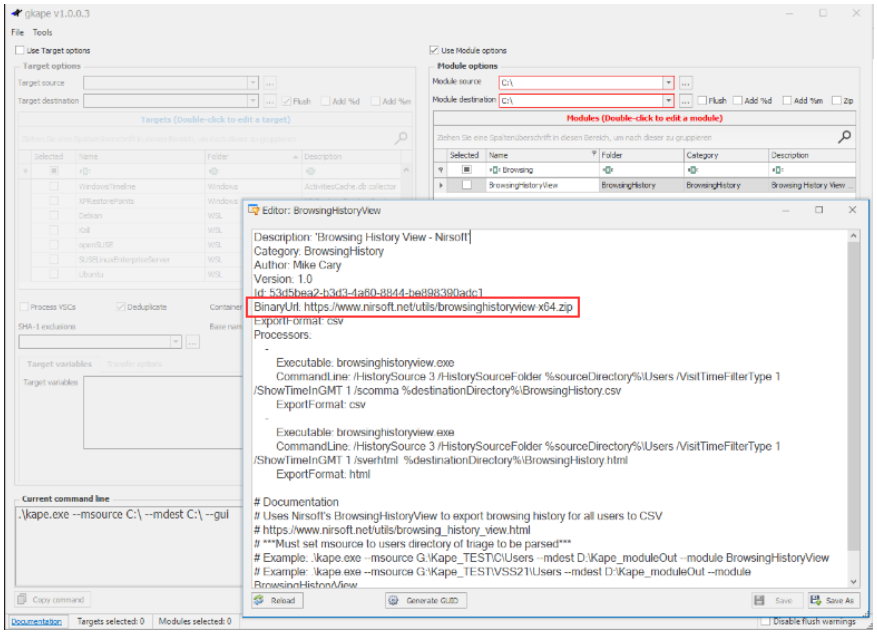
Unter den eingesammelten Dateien sind auch solche, die Aufschluss über die besuchten Webseiten und heruntergeladenen Dateien geben. Diese Historie kann beim Aufdecken der beschriebenen Social-Engineering-Angriffe weiterhelfen. Je nach Webbrowser gibt es für die Auswertung unterschiedliche Werkzeuge. Für die am häufigsten verwendeten Browser hat NirSoft das Programm BrowsingHistoryView (siehe ix.de/z6js) entwickelt und bietet es als Freeware an. Das BrowsingHistoryView-Modul von KAPE erlaubt sein automatisches Ausführen.

Wie in den vergangenen beiden Teilen des Tutorials beschrieben, ist KAPE zuerst auf einem Analysesystem vorzubereiten: Es wird auf einen externen Datenträger kopiert und aktualisiert. Anschließend wird die Drittanwendung von der im BrowsingHistoryView-Modul angegebenen BinaryUrl heruntergeladen (siehe Abbildung 1) und im KAPE-Unterordner Modules/bin entpackt.

Der Kommandozeilenbefehl für das Sammeln der Daten wird anschließend auf dem Analysesystem zusammengestellt. Hierzu 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 WebBrowsers-Compound-Target aus (siehe Abbildung 2). Der Kommandozeilenbefehl unter „Current command line“ wird daraufhin für die Ausführung auf dem zu untersuchenden System gespeichert und KAPE geschlossen.

Untersuchen des betroffenen Systems

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. Nun fügt man den vorbereiteten Kommandozeilenaufruf in die Kommandozeile ein und ersetzt den Platzhalter für



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

die „Target destination“: Im vorliegenden Beispiel wurde der externen Festplatte der Laufwerksbuchstabe F zugeordnet. Entsprechend wird der Platzhalter durch einen gültigen Ordnerpfad für das Sammeln der Webbrowserdateien ersetzt:

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

Mit dieser Änderung kann das Sammeln der Dateien durch die Ausführung des Befehls starten. Anschließend finden sich im Zielordner (hier F:\TargetDestination\) die zusammenkopierten Daten sowie drei Protokolldateien.

Als Nächstes sind die gesammelten Dateien mithilfe von NiroSofts BrowsingHistoryView in ein für den Analysten lesbares Format zu überführen. Hierzu wird der externe Datenträger wieder an das Analysesystem angeschlossen, die grafische Oberfläche von KAPE gestartet und „Use Module options“ ausgewählt. Als „Module source“ wird der Unterordner mit dem Namen der untersuchten Systempartition (hier C) im zuvor als „Target destination“ gewählten Pfad konfiguriert.

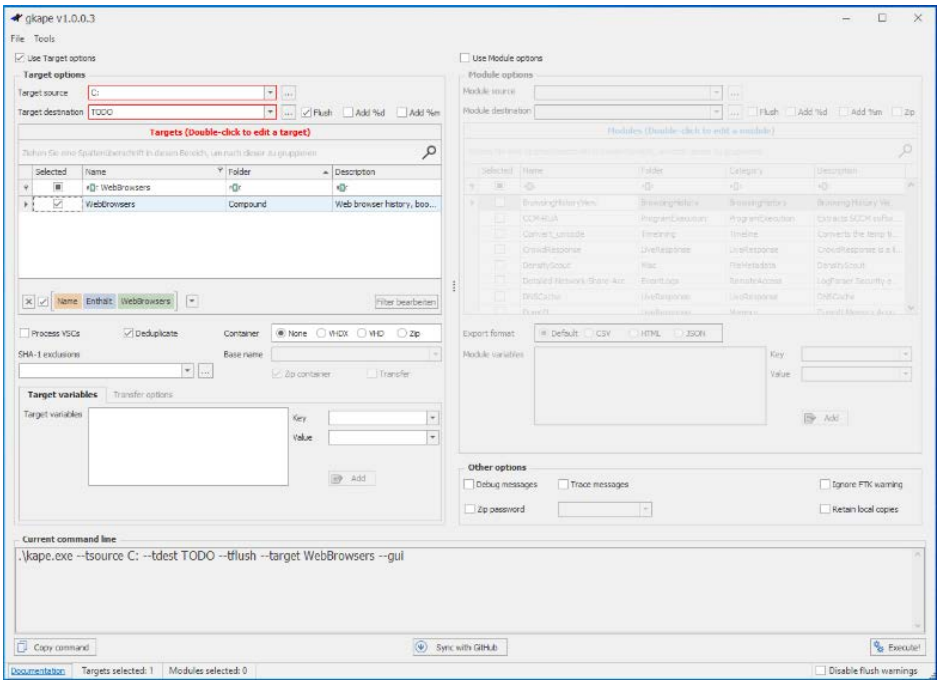
Dies weicht von der Konfiguration in den vorhergehenden Tutorialteilen ab. Wenn man sich den CommandLine-Aufruf im BrowsingHistoryView-Modul ansieht (siehe Abbildung 1), wird klar, weshalb der Unterordner gewählt werden muss: Das Modul ruft die Drittanwendung auf und übergibt ihr als Untersuchungsquelle den Users-Unterordner von „Module source“ (%sourceDirectory%). Dieser befindet sich im Ordner, der den Inhalt der Systempartition enthält, und nicht im als „Target destination“ definierten Ordner. Bei „Module destination“ wird ein neuer Ordner für die Resultate der Auswertung gewählt und das BrowsingHistoryView-Modul selektiert. Nach diesen Einstellungen (siehe Abbildung 3) erfolgt die Auswertung über den Execute-Knopf.

Schadsoftware-Download nachvollziehen

Das Resultat der Auswertung liegt abschließend als BrowsingHistory.csv im Unterordner BrowserHistory des Module-destination-Ordners. Die CSV-Datei wird für die Untersuchung wie in den vorhergehenden Untersuchungen dieser Tutorialreihe im Timeline Explorer geöffnet. Für eine bessere Übersicht wählt man im Tools-Menü den Eintrag „Reset column widths“ (Strg + R). Dieser setzt alle Spaltenbreiten auf einen vordefinierten Maximalwert. Das erleichtert die Arbeit mit den

X-TRACT

- Ein Großteil der Cybervorfälle beginnt mit einem Social-Engineering-Angriff.
- Durch Ausspähen persönlicher Nutzerdaten oder Ausführen von Schadsoftware erhält der Angreifer Zutritt ins Unternehmensnetzwerk. Damit startet die Kompromittierung der gesamten IT-Landschaft.
- KAPE wertet die Browserhistorie aus und bereitet die Daten für Forensiker auf. So lässt sich der Angriff nachvollziehen.

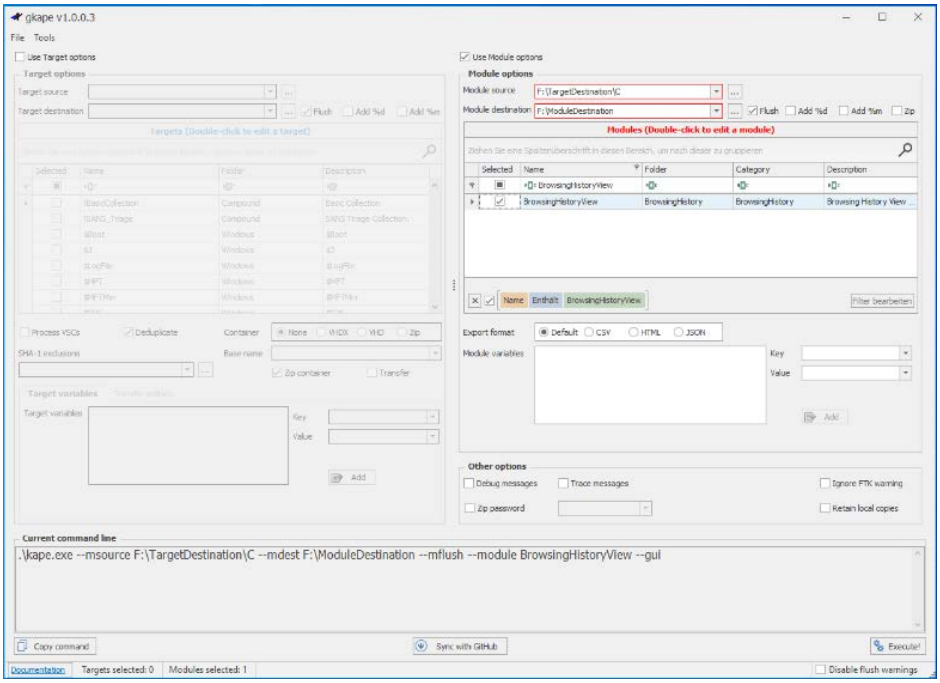


Beispielkonfiguration für das Sammeln der forensisch relevanten Webbrowserdateien (Abb. 2)

Daten, gerade auch wegen langer URLs und Webseitentitel.

In diesem Tutorialteil folgen wir einer forensischen Untersuchung im Rahmen eines Ransomware-Angriffs, der Ende 2020 bis Anfang 2021 stattgefunden hat. Der Angriff war Teil einer großen Malware-Kampagne, die neben Ransomware auch E-Banking-Trojaner und weitere Schadsoftware auslieferte (siehe ix.de/z6js).

Zu Beginn solcher Untersuchungen lohnt es sich, zuerst die Downloads näher zu betrachten. Dazu fährt man mit der Maus über den Titel der Spalte „Visit Type“, klickt auf das erscheinende Filtersymbol und wählt den Download-Eintrag aus (siehe Abbildung 4). Alternativ gibt man selbst das Wort (Download) als Filter ein. Die danach sichtbaren Zeilen repräsentieren Downloads, die mit einem



So kann man mit NirSofts BrowsingHistoryView die Browserhistorie und die heruntergeladenen Dateien auswerten (Abb. 3).

Webbrowser auf dem untersuchten System erfolgten.

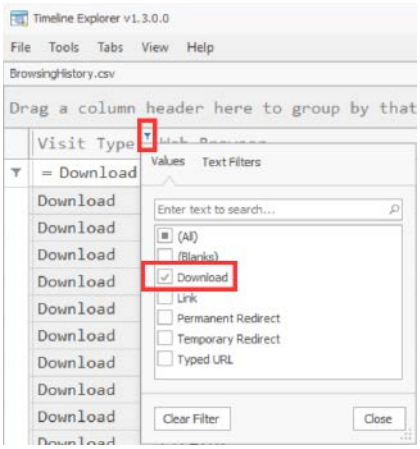
Als Nächstes prüft man in der Spalte „Title“ die Namen der heruntergeladenen Dateien. Hier hält man Ausschau nach Dateiendungen, die auf eine Anwendung, ein Skript oder ein Office-Dokument mit Makros hindeuten. Es könnte sich dabei um heruntergeladene Schadsoftware handeln. Im Fallbeispiel fällt der Name sendung_N311025_25112020.com auf (siehe Abbildung 5). Die Dateiendung deutet auf eine unter Windows ausführbare Anwendung hin. Auch fällt auf, dass der Rest des Dateinamens sich eher nach einem Dokument als nach einer Applikation anhört – ein weiteres Anzeichen für einen Social-Engineering-Angriff.

Der Verdacht, dass es sich um heruntergeladene Schadsoftware handelt, lässt sich auf mehrere Arten erhärten. Falls die Datei auf dem untersuchten System existiert, kann man sie direkt untersuchen. Oft befindet sie sich im Standard-Download-Ordner oder im Papierkorb des Nutzerkontos. Komplett gelöschte Dokumente lassen sich häufig mit einem Datenrettungswerkzeug aus dem Dateisystem wiederherstellen und mit einer Schadsoftwareanalyse-Sandbox, also in einem abgeschotteten System, untersuchen. Kann sie nicht mehr gefunden oder mit einer Sandbox analysiert werden, lohnt sich die Untersuchung der Download-Quelle. Diese ist in der Spalte „Url“ festgehalten.

Vorsichtiges Surfen

Vorsicht, keine der aufgeführten Adressen sollte jemals direkt vom Analysesystem besucht werden. Die Webseite könnte eine Schwachstelle des Browsers ausnutzen und das System infizieren. Auch kann aus Versehen erneut ein Schädling heruntergeladen und ausgeführt werden. Das Analysesystem und seine Integrität sind hier in Gefahr. Aus diesem Grund werden alle Internetadressen entweder aus einer für solche Zwecke geeigneten abgeschotteten und abgesicherten virtuellen Maschine besucht oder noch einfacher mit einem Onlinedienst wie urlscan.io.

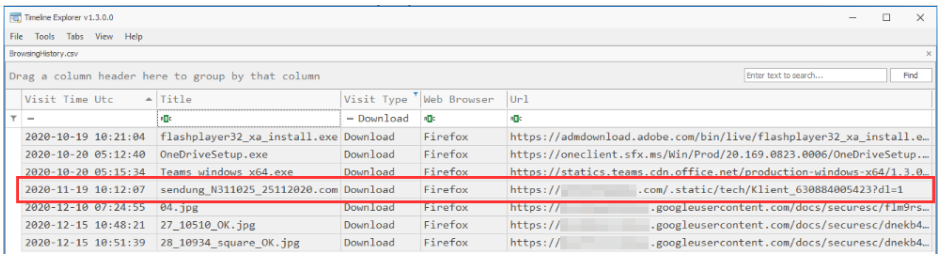
Dieser Dienst nimmt eine URL entgegen, besucht die Webseite und sammelt wertvolle Informationen ein. Allem voran erstellt er einen Screenshot der Seite, gleicht sie mit der Datenbank des Google-Safe-Browsing-Projekts (siehe ix.de/z6js) ab und warnt, wenn die Webseite sich für eine bekannte Marke ausgibt. Das reicht meist zur Erkennung einer Phishingwebseite aus. Für tiefer gehende Untersuchun-



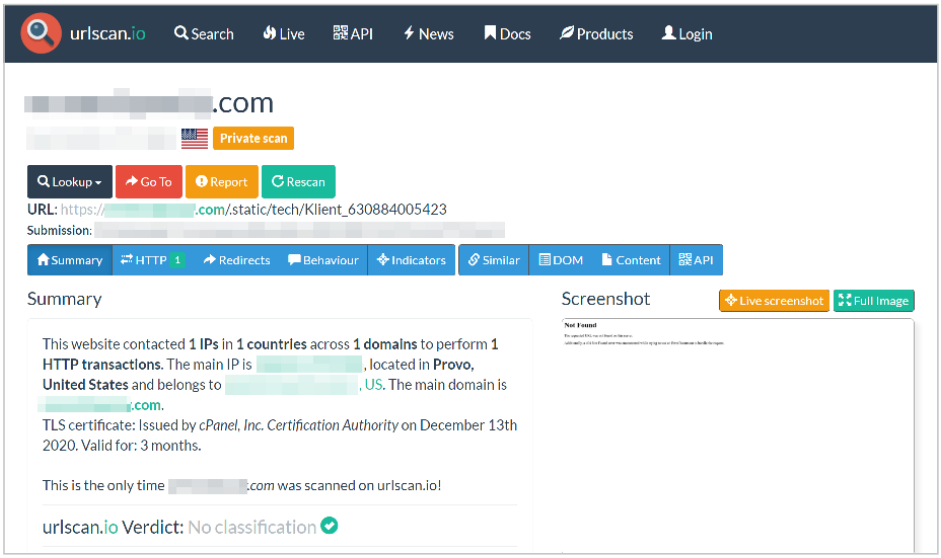
Mit dem Timeline-Explorer-Filter kann man die Anzeige auf erfolgte Downloads beschränken (Abb. 4).

gen sammelt urlscan.io weitere Daten: den Quelltext der Seite, die nachgeladenen Ressourcen, Fakten rund um die Domain und den Hoster et cetera. All dies, ohne das eigene System in Gefahr zu bringen und dem Angreifer potenziell wertvolle Informationen preiszugeben, zum Beispiel die Firmen-IP-Adresse.

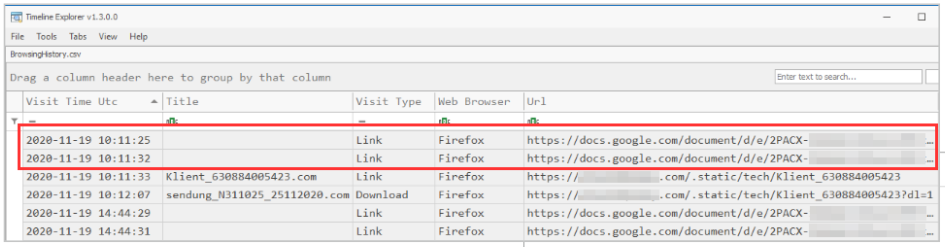
Zum Zeitpunkt der Untersuchung war die Download-Quelle nicht mehr erreichbar (siehe Abbildung 6). Bei einem Social-Engineering-Angriff wird man in vielen



Ein Download fällt auf: Der Dateiname klingt zwar wie der eines Dokuments, die Endung deutet aber auf eine ausführbare Anwendung hin (Abb. 5).



Die Quelle des verdächtigen Downloads steht nicht mehr zur Verfügung (Abb. 6).



Google Docs wird kurz vor dem verdächtigen Download besucht (Abb. 7).

Vor dem verdächtigen Download besuchtes Google-Docs-Dokument (Abb. 8)

Fällen zuerst auf eine Webseite gelockt, die sich erneut auf den konstruierten Vorwand bezieht und erst danach zum Download einer Datei führt. Die kurz vor dem Download besuchten Webseiten können daher aufschlussreich sein.

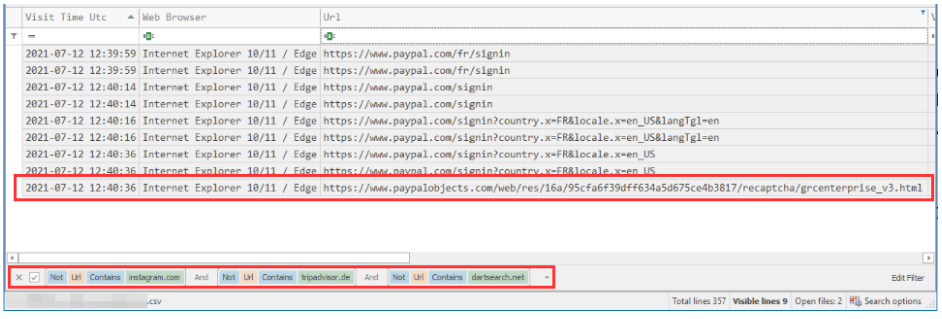
Im Timeline Explorer sortiert man hierzu die Spalte „Visit Time Utc“ und prüft die zuvor besuchten Webseiten. Im Fallbeispiel wurde kurz vor dem Herunterladen ein Google-Docs-Dokument besucht (siehe Abbildung 7). Angreifer nutzen häufig vertraute Cloud-Dienste wie Google Docs, das weckt seltener Zweifel an der Vertrauenswürdigkeit. Gleichzeitig führt ein solcher Dienst diverse technische Schutzmaßnahmen hinters Licht.

Aufschlussreicher Quelltext

Zum Zeitpunkt der Untersuchung war das Google-Docs-Dokument noch erreichbar und konnte mit urlscan.io besucht werden. Der von urlscan.io erzeugte Screenshot lässt keine Zweifel offen, dass es sich hier um einen unerwünschten Download und einen Social-Engineering-Angriff handelt (siehe Abbildung 8): Im Quelltext (DOM-Knopf auf urlscan.io) sah man, dass alle Links im Google-Docs-Dokument auf die Download-URL zeigten. Dass ein vertrauenswürdiges Unternehmen aus Google Docs heraus nur auf eine Webseite Dritter weiterleitet, ist sehr unwahrscheinlich. Im Normalfall würde man erwarten, gleich das Dokument in Google Docs zu sehen.

Wenn keine auffällige Datei aus dem Internet heruntergeladen worden wäre, müsste man alle besuchten URLs einzeln untersuchen. Vor einem systematischen Abarbeiten der Adressen lohnt sich die Suche nach Ausreißern. Besonders häufig besuchte und vertrauenswürdige Adressen können mit dem Filter Editor („Filter Editor …“ im Kontextmenü eines beliebigen Spaltentitels) herausgefiltert werden. Die verbleibenden Adressen werden einer näheren Untersuchung unterzogen. Solche Filter erlaubten es bei einem anderen Cybervorfall, den Besuch einer PayPal-Phishing-Webseite aufzudecken (siehe Abbildung 9).

Abhängig vom Nutzerverhalten lohnt sich ein Blick auf die besonders selten besuchten Webseiten. Hierzu werden die Einträge gemäß ihrem Visit Count sortiert.



Besonders häufig besuchte und mit hoher Wahrscheinlichkeit legitime URLs wurden herausgefiltert. Damit wird der Besuch einer bekannten PayPal-Phishing-Webseite schnell erkennbar (Abb. 9).

Gerade in Kombination mit den erwähnten Filtern oder durch Einschränkung des Zeitraums ist dies ein in der Praxis erfolgversprechendes Vorgehen.

Ausblick

Die KAPE-Targets sammeln deutlich mehr Dateien, als BrowsingHistoryView auswerten kann. Zum Beispiel werden auch Cookies, der Browsercache, die offenen Browsertabs und vieles mehr mitgesichert. All diese Informationen können von hohem Wert für forensische Untersuchungen sein. Vor allem, um das Verhalten von Nutzern nachzuvollziehen. Es lohnt sich also, sich in einer freien Minute näher mit all diesen Artefakten auseinanderzusetzen.

Entdeckt das beschriebene Vorgehen keine verdächtige Surfaktivität in der Browserhistorie, lohnt es sich, sich anderen forensischen Artefakten zuzuwenden – zum Beispiel verdächtigen Applikationsausführungen auf dem Computer. Im vierten und letzten Teil der Tutorialreihe geht es um sogenannte Prefetch-Artefakte und die Suche nach solchen verdächtigen Ausführungen. (ur@ix.de)

Quellen

Die im Text genannten Werkzeuge und Dienste sind über ix.de/z6js zu finden.

Gregor Wegberg

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