Dynamische Malware-Analyse

Die statische Analyse verdächtiger Dateien führt zu Hypothesen, die Forensiker handfest prüfen müssen. Sie beobachten den Schadcode in voller Aktion – und zeichnen seine Aktivitäten auf.

Von Nadia Meichtry

Digitale Forensiker – oder auch Reverse Engineers – analysieren Schadsoftware in vier Phasen. In der vierten und letzten Phase, der dynamischen Analyse, wird die Malware in Aktion beobachtet. Der vierte Teil unserer Artikelserie zur Malware-Analyse erklärt also, welche Werkzeuge dafür notwendig sind und wie Forensiker in dieser Phase vorgehen. Eine vorausgehende OSINT-Analyse (Open Source Intelligence) schaffte zuerst einen Überblick darüber, womit man es überhaupt zu tun hat [1], danach folgte eine vollautomatische Analyse mit entsprechender Software in einer Sandbox-Umgebung [2]. Im dritten Teil ging es an die statischen Eigenschaften der Schadsoftware [3]. Alle drei vorherigen Schritte dienten dazu, Hypothesen über die Art und Wirkungsweise der Malware aufzustellen. Der letzte Teil der Artikelreihe befasst sich nun mit der dynamischen Analyse – dem Beobachten der Malware in Aktion –, um die formulierten Hypothesen zu überprüfen.

Infektion zulassen

Während der dynamischen Untersuchung führt man die verdächtige Datei in einer virtuellen Maschine aus und zeichnet ihre Aktivitäten mit diversen Werkzeugen auf. Alle im Artikel erwähnten Tools sind auch unter ix.de/z48a zu finden. Analysten sollten jedoch nur die notwendigen Anwendungen starten, um keine unnötigen Interaktionen zu erzeugen. Sie überwachen unter anderem gestartete Prozesse, aufgebaute Netzwerkverbindungen, abgelegte Dateien sowie neu erstellte oder geänderte Registrierungsschlüssel und Dienste. Dabei läuft die virtuelle Maschine ohne Schutzmechanismen, sodass die Malware freies Spiel hat und nicht blockiert oder entfernt wird.

Dadurch will man Hinweise auf das Verhalten der zu analysierenden Datei gewinnen, um ihre Natur und die Bedrohung, die sie für das Unternehmen darstellen kann, zu verstehen. Die Beobachtungen können die vorherigen Hypothesen weiterentwickeln, die Liste der gefundenen Indicators of Compromise (IOCs) erweitern, die bei der Analyse der statischen Eigenschaften festgestellt wurden, oder zuvor identifizierte IOCs bestätigen. Es ist natürlich absolut notwendig, die Operation in einer virtuellen Maschine durchzuführen – handelt es sich tatsächlich um Schadsoftware, infiziert sie schließlich das System. Vorher erstellt man am besten einen sauberen Speicherpunkt, um das System gegebenenfalls zurückzuspielen und eine erneute Analyse durchführen zu können. Das Starten der Schadsoftware kann die Auf-

iX-TRACT

- Bei der dynamischen Malware-Analyse überwachen Forensiker Prozesse, Netzwerkaktivitäten und Änderungen an der Registry von Windows-Systemen. Dafür müssen sie genau wissen, was zu den normalen Systemaktivitäten gehört, um Auffälligkeiten zu erkennen.
- Schadsoftware verfügt über unterschiedliche Fähigkeiten, zum Beispiel Prozessinjektion oder die Kommunikation mit einem C2-Server (Command and Control). Diese Fähigkeiten wollen Forensiker aufdecken.
- Die Analyseumgebung muss akribisch vorbereitet sein: Weder dürfen eigene Prozesse die Aufzeichnung stören, noch darf die Umgebung zu leer sein – denn manche Malware erkennt, ob sie sich in einer Laborumgebung befindet.

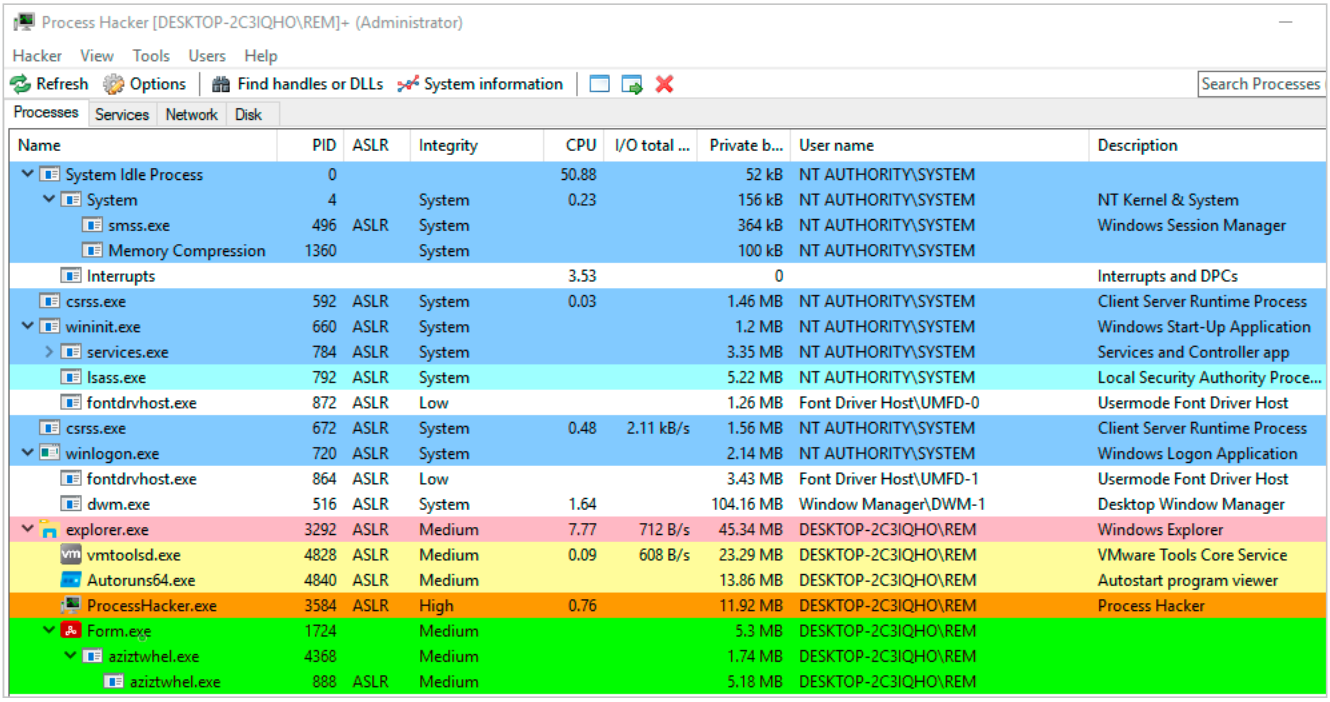
Malware-Analyse in vier Schritten

Teil 1: Einstieg – Grundlagen und Ablauf der Malware-Analyse

Teil 2: OSINT – Abfragen von öffentlichen Informationen, vollautomatische Analyse in der Sandbox

Teil 3: Statische Eigenschaften – Hypothesen anhand der Beschaffenheit von Malware aufstellen

Teil 4: Dynamische Eigenschaften – Aktivitäten der Malware aufzeichnen und auswerten

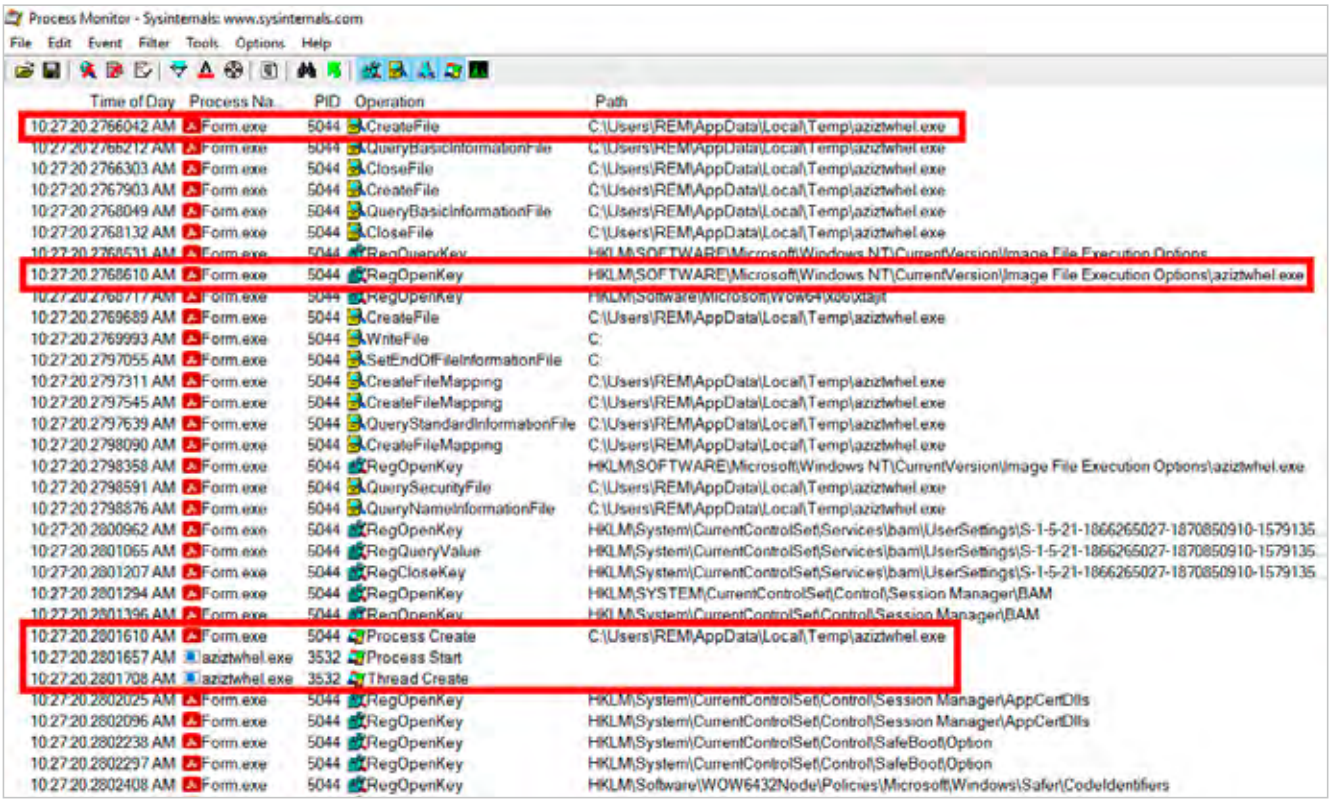


Process Hacker zeigt bereits, dass Form.exe verdächtige Prozesse startet (Abb. 1).

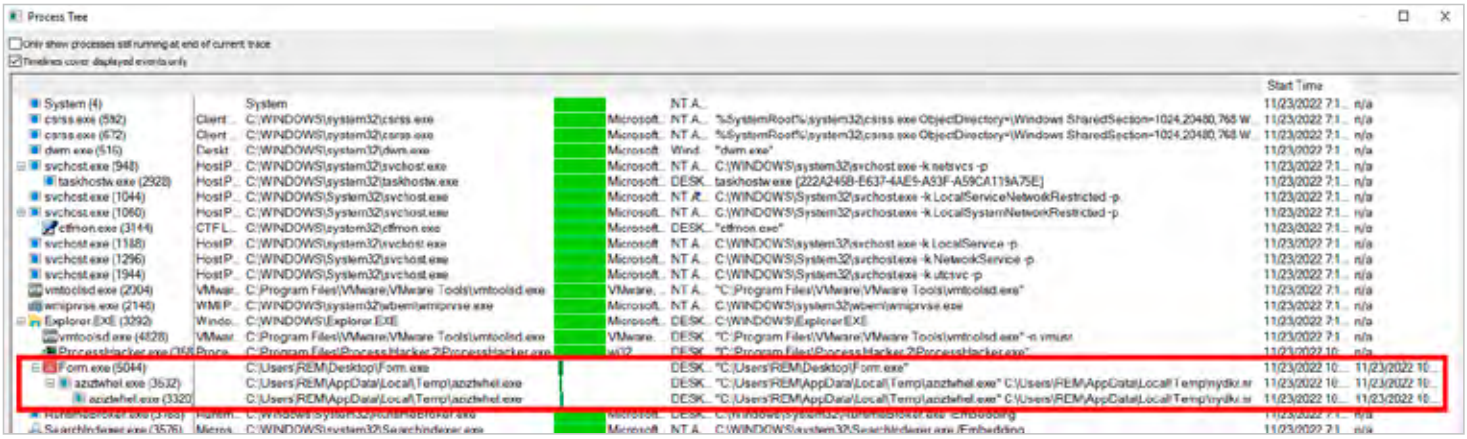
merksamkeit der Angreifer wecken und ihnen mitteilen, dass sie untersucht werden. Grund dafür ist die Kommunikation mit ihrem Server, beispielsweise über den Aufruf ihrer Domänen oder IP-Adressen. Um das zu verhindern, sollte man entweder die Netzwerkverbindung der virtuellen Maschine unterbrechen oder die Verbindung in eine andere virtuelle Maschine weiterleiten, die einen Internetzugang nur simuliert. Das beeinflusst allerdings auch, was gemessen wird: Will die Schadsoftware etwa zusätzliche Dateien herunterladen und ausführen, so ist dies bei deaktivierter Netzwerkverbindung nicht mehr möglich und wird auch nicht erfasst. Der Mittelweg ist die Nutzung eines VPN oder des TOR-Netzwerks, um die eigene Herkunft zu verbergen und sich so zu schützen.

Malware-Fähigkeiten

Malware hat bestimmte Fähigkeiten, die man mit der dynamischen Analyse ermitteln kann. Sie könnte beispielsweise mit einem Command-and-Control-Server (C2-Server) kommunizieren oder Dateien herunterladen und ablegen. Außerdem setzt Malware verschiedene Techniken ein, um sich der Erkennung zu entziehen – zum Beispiel Prozessinjektion. Ein C2-Server ist ein Server, der von den Angreifern kontrolliert wird, um die Kommunikation mit dem kompromittierten System aufrechtzuerhalten und es im schlimmsten Fall fernzusteuern. Meist wird HTTPS als Protokoll für C2-Server benutzt. Dass andere Protokolle wie DNS



In Process Monitor kann man das Verhalten der untersuchten Malware genauer nachverfolgen (Abb. 2).



Im aus Logs erstellten Prozessbaum zeigen sich die verdächtigen Prozesse von ihrer flüchtigen Seite – der grüne Balken illustriert ihre Lebensdauer (Abb. 3).

und sogar die Webanwendungen und Schnittstellen sozialer Medien zum Einsatz kommen, ist ebenfalls möglich. Die Schadsoftware sendet in regelmäßigen Abständen kurze Nachrichten mit grundlegenden Informationen über den Zustand des Schadprogramms und des infizierten Hosts an den Angreifer, was als Beaconing bezeichnet wird. Sie kann vom Angreifer auch Befehle einholen, zum Beispiel um sich selbst zu aktualisieren oder einen Rechner zu scannen. Über diesen Kanal schafft die Malware letztlich auch gestohlene Daten nach draußen.

Wenn die Malware ein Downloader ist, lädt sie Inhalte wie weitere Schadsoftware, bösartige Befehle oder ihre Konfiguration aus dem Internet herunter. Wenn es sich hingegen um einen Dropper handelt, wird die Malware andere Dateien aus ihrem eigenen Code extrahieren; sie kann etwa ausführbare Dateien in ihrer Ressourcensektion verstecken. Dies lässt sich aber auch bereits in der vorhergegangenen Analyse statischer Eigenschaften herausfinden.

Um Erkennungswerkzeuge zu umgehen und Analysten zu verwirren, statten Angreifer ihre Schadsoftware gerne mit einer Technik namens Prozess- oder Codeinjektion aus. Sie schleust dann Code in andere, eigentlich harmlose Prozesse ein. Eine Variante davon ist die DLL-Injektion (Dynamic Link Library). Weil DLLs Code und Daten enthalten, die mehr als ein Programm gleichzeitig verwendet, sind sie ein beliebtes Ziel für Injektionen. Der bösartige Code wird dann von den Programmen ausgeführt, die die DLL nutzen.

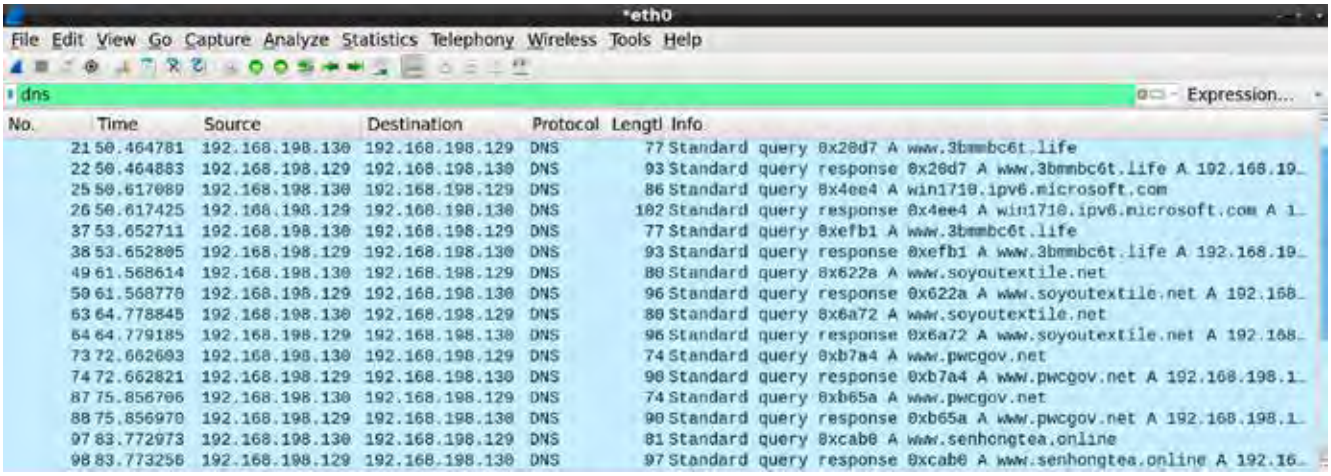
Eine weitere Technik ist das Hooking. Hier fängt die Malware die Ausführung von bestimmtem Code innerhalb eines Opferprozesses ab und stört den infizierten Prozess. In Kombination mit Codeinjektionstechniken können sie so Rootkits implementieren, die bösartige Artefakte vor dem Benutzer des infizierten Systems verbergen. Indem es sich gezielt in Funktionen in legitimen Programmen einhakt, unterbricht ein Rootkit den normalen Informationsfluss innerhalb des Opferprozesses. So spioniert Malware zum Beispiel verarbeitete Daten aus.

Darüber hinaus kann Malware Process Hollowing oder Process Replacement betreiben. Die Schadsoftware startet dafür einen legitimen Prozess, hält ihn an, gibt den Arbeitsspeicher frei, der den Programmcode enthält, und ersetzt ihn durch ein bösartiges Programm. Der gestartete Prozess führt dann den neu eingeschleusten Code aus. Auch hier wird wieder bösartiger Code in Prozessen versteckt, die legitim erscheinen. Häufig geschieht das beim Entpacken von verpackter Malware.

Malware versucht oft zu erkennen, ob sie in einer Analyseumgebung läuft oder auf dem System eines echten Opfers. Sie hält dafür nach Hinweisen Ausschau, beispielsweise nach fehlenden Anwendungen wie Microsoft Office in der Laborumgebung, sehr kurzen Browserverläufen oder nur wenigen Dateien auf der Festplatte. Bestimmte Hardwarekomponenten oder Dateien und Registrierungsschlüssel, die damit verbunden sind, geben ihr zusätzliche Informationen über die Umgebung. In der Konfiguration des infizierten Computers könnte sie außerdem nach Artefakten fahnden, die mit automatisierten Malware-Sandboxes und nicht mit einem wirklich benutzten System in Verbindung gebracht werden. Das kann eine leere Zwischenablage sein, eine CPU mit nur einem Kern, ein stillstehender Mauszeiger, eine recht kleine Festplatte oder der Hinweis auf ein frisch gestartetes System. Und auch Analysewerkzeuge wie Process Hacker, Wireshark oder Ghidra können allein durch ihre Anwesenheit auf eine Analyseumgebung

```
remnux@remnux:~$ fakedns
pyminifakeDNS:: dom.query. 60 IN A 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: v20.vortex-win.data.microsoft.com. -> 192.168.198.129
Respuesta: www.msftconnecttest.com. -> 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: v20.vortex-win.data.microsoft.com. -> 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: www.msftconnecttest.com. -> 192.168.198.129
Respuesta: www.3bmmbc6t.life. -> 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: www.3bmmbc6t.life. -> 192.168.198.129
Respuesta: www.soyoutextile.net. -> 192.168.198.129
Respuesta: www.soyoutextile.net. -> 192.168.198.129
Respuesta: www.pwcgov.net. -> 192.168.198.129
Respuesta: www.pwcgov.net. -> 192.168.198.129
Respuesta: www.senhongtea.online. -> 192.168.198.129
Respuesta: www.senhongtea.online. -> 192.168.198.129
Respuesta: watson.telemetry.microsoft.com. -> 192.168.198.129
Respuesta: www.sinavouv.online. -> 192.168.198.129
Respuesta: www.sinavouv.online. -> 192.168.198.129
Respuesta: www.timebasecorruption.com. -> 192.168.198.129
Respuesta: www.timebasecorruption.com. -> 192.168.198.129
Respuesta: www.gitedecamat.com. -> 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: www.gitedecamat.com. -> 192.168.198.129
Respuesta: www.easy005.xyz. -> 192.168.198.129
Respuesta: www.easy005.xyz. -> 192.168.198.129
Respuesta: www.wehavepros.com. -> 192.168.198.129
Respuesta: www.wehavepros.com. -> 192.168.198.129
Respuesta: www.95788.biz. -> 192.168.198.129
Respuesta: www.95788.biz. -> 192.168.198.129
Respuesta: www.emiliayenrique.com. -> 192.168.198.129
Respuesta: www.emiliayenrique.com. -> 192.168.198.129
Respuesta: www.autowallet.top. -> 192.168.198.129
Respuesta: www.autowallet.top. -> 192.168.198.129
Respuesta: www.maturistic.com. -> 192.168.198.129
Respuesta: win1710.ipv6.microsoft.com. -> 192.168.198.129
Respuesta: www.maturistic.com. -> 192.168.198.129
Respuesta: www.baltic-freya.com. -> 192.168.198.129
Respuesta: www.baltic-freya.com. -> 192.168.198.129
Respuesta: www.elwex.com. -> 192.168.198.129
Respuesta: www.elwex.com. -> 192.168.198.129
Respuesta: www.rbqxjyrb.shop. -> 192.168.198.129
```

FakeDNS beantwortet alle Anfragen der Malware mit der IP-Adresse der Linux-Maschine, auf der die infizierte Windows-Umgebung virtualisiert wird (Abb. 4).



Wireshark stellt die Kommunikationen von Form.exe zwischen der infizierten virtuellen Maschine und dem Linux-System dar (Abb. 5).

hinweisen, weil sie auf dem System eines typischen Opfers nicht installiert sind.

Prozessanalyse

Um der Malware nun endgültig auf die Schliche zu kommen und ihre bösartigen Prozesse zu erkennen, müssen Analysten zunächst wissen, wie der Prozessbaum auf einem System im Normalfall aussieht. Dazu hat bereits das SANS-Institut ein Poster für Windows-Systeme veröffentlicht (siehe ix.de/z48a). Unter dem Titel „Evil Hunt 1" erklärt es, welche Elternprozesse welche Kindprozesse starten. Beispielsweise sollte es nur eine Instanz von lsass.exe geben, mit wininit.exe als Elternprozess. Bei der Analyse ist es daher wichtig, die Beziehungen zwischen den Prozessen und ihre Prozesspfade zu überprüfen. Dafür kann man verschiedene freie Werkzeuge für Windows verwenden, wie Process Hacker und Sysinternals Process Explorer von Microsoft. Sie überwachen unter anderem die Systemressourcen sowie den Prozessbaum, die Prozessnamen und -kennungen (PID). Die Tools zeigen die von den Prozessen geöffneten oder geladenen Handles und DLLs an. Darüber hinaus listet Process Hacker Dienste und aktive Netzwerkverbindungen auf, ähnlich wie Sysinternals TCPView4. Prozesse werden auch farblich hervorgehoben: Grün steht für gestartete Prozesse, Rot für beendete und Blau für System- und Dienstprozesse.

Neben diesen beiden Werkzeugen gibt es Sysinternals Process Monitor. Er überwacht in Echtzeit die Prozessaktivitäten und Interaktionen mit anderen Prozessen, der Registry, dem Dateisystem und dem Netzwerk und protokolliert sie in einer Datei. Process Monitor erfasst Prozessdetails, einschließlich Befehlszeile und Operationen. Man kann Filter definieren, um sich beispielsweise auf einen Prozess zu konzentrieren oder normale Aktivitäten auszuschließen. Das Tool verfügt auch über eine Funktion zum Zeichnen des Prozessbaums, der alle im Protokoll referenzierten Prozesse in einer Eltern-Kind-Hierarchie anzeigt. Prozesse mit demselben Elternteil sortiert es nach ihrer Startzeit. Process Monitor gibt dabei auch einige Daten aus: den Pfad der ausführbaren Datei, das Benutzerkonto und die Startzeit. Solche Protokolle können anschließend mit ProcDOT visualisiert werden, indem man ein Aktivitätsdiagramm für den gewählten Prozess erstellt. Dadurch lassen sich Anomalien leicht aufdecken.

Um die vorgestellten Werkzeuge zu veranschaulichen, kann man innerhalb einer virtuellen Windows-Maschine die Malware-Datei Form.exe ausführen, die bereits in Teil 2 der Reihe vorgestellt wurde [2]. Der Prozessbaum in Process Hacker zeigt, dass der Prozess Form.exe unterhalb von explorer.exe läuft (siehe Abbildung 1). Das bedeutet, dass dieses Programm durch den Benutzer per Doppelklick ausgeführt wurde, in diesem Fall vom Desktop aus. Form.exe verwendet außerdem das Logo von Adobe, um legitim zu wirken. Ebenso kann man sehen, dass Form.exe aziztwhel.exe aufruft, das dann eine weitere Instanz von sich selbst startet (in Abbildung 1 grün hervorgehoben). Das deutet auf eine Prozessinjektion hin. Ein solcher Prozessname aus zufälligen Charakteren ist besonders auffällig und legt nahe, dass es sich um einen bösartigen Prozess handelt – ein Eindruck, der sich dadurch verstärkt, dass sich alle drei Prozesse in weniger als einer Sekunde beendeten, um unentdeckt zu bleiben.

In Process Monitor ist das noch einmal detaillierter zu sehen, denn das Tool zeigt an, welche Operationen konkret durchgeführt wurden. Wie in Abbildung 2 zu sehen, legt Form.exe zuerst die Datei aziztwhel.exe unter C:\Users\REM\AppData\Local\Temp mit der Operation `CreateFile` an (zu sehen in der ersten roten Box in Abbildung 2). Der Pfad App Data\Local\Temp ist als Ablageort von Malware beliebt, da er zuverlässig vorhanden ist und keine erhöhten Rechte zum Schreiben benötigt. Außerdem ist er ein unauffälliger Bereich, den viele Opfer von Malware-Angriffen übersehen, da er nicht zwingend von Antivirenlösungen gescannt wird.

Anschließend öffnet die Schadsoftware den Registrierungsschlüssel HKLM\SOFTWARE\Microsoft\WindowsNT\CurrentVersion\Image File Execution Options\aziztwhel.exe mit der Operation `RegOpenKey` (siehe zweite rote Box in Abbildung 2), um den Wert abzufragen.

Mit so einem Vorgehen versucht Malware üblicherweise, Persistenz herzustellen und Privilegien zu erhöhen. MITRE ATT&CK beschreibt, dass es die Image File Execution Options (IFEO) ermöglichen, einen Debugger an eine Anwendung anzuhängen. Beim Erstellen eines Prozesses wird ein Debugger, der in den IFEO einer Anwendung konfiguriert ist, dem Namen der Anwendung vorangestellt – der neue Prozess startet effektiv unter dem Debugger. Diese Optionen können direkt über die Registry unter HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<executable> eingestellt werden, wobei <executable> die Binärdatei ist, an die der Debugger angehängt ist. In diesem Fall handelt es sich um die Datei aziztwhel.exe. Die Malware verschafft sich so einen Hintertürzugang als „Debugger". Nachdem die Malware die Registry geändert hat, wird das Debugger-Programm

mit Systemrechten ausgeführt. Form.exe startet dann mit der Operation `Process Create` aziztwhel.exe (siehe dritte rote Box auf Abbildung 2).

Der aus den Logs von Process Monitor erstellte Prozessbaum (siehe Abbildung 3) sieht genauso aus wie der von Process Hacker. Man kann feststellen, dass die Lebensdauer (mit grüner Markierung) der Prozesse Form.exe und azizthwel.exe recht kurz war. Beide azizthwel.exe-Prozesse rufen außerdem die Datei nydkr.nr im Temp-Ordner auf.

Mit ProcDOT ist das Starten der beiden azizthwel.exe-Prozesse ebenfalls deutlich sichtbar. Es zeigt im Ereignisbaum an, dass temporäre Dateien erstellt wurden: nydkr.nr, gdeinuzidby.smf, syD0B8.tmp und syD0B9.tmp. Die zwei tmp-Dateien wurden dann gelöscht – bei tmp-Dateien ist das zu erwarten, da sie in der Regel entfernt werden, wenn der Prozess sich beendet.

Die Gesamtheit der Resultate stimmt mit den Informationen im Behavior Graph von Joe Sandbox und mit dem beschriebenen Verhalten in VirusTotal überein (siehe ix.de/z48a). Das stärkt natürlich das Vertrauen in die vorangegangenen Untersuchungsergebnisse. Und auch die von der Malware abgelegten Dateien findet man bei Joe Sandbox und VirusTotal unter „Dropped files“. Joe Sandbox berichtet, dass die nr-Datei eine COM-Datei ist – ausführbare Dateien, die in der Regel für das Ausführen einer Reihe von Befehlen verwendet werden.

Analyse von Netzwerkverbindungen

Nachdem man beobachtet hat, was die Malware im System anstellt, kann man sich der Kommunikation widmen, die sie nach draußen aufbaut. Um Netzwerkverbindungen zu untersuchen, empfiehlt es sich, die Internetressource, auf die die Datei zugreifen möchte, im Labor zu simulieren. Indem man die bösartige Netzwerkverbindung auf einen kontrollierten Server und Dienst umleitet, gibt man der Malware die Möglichkeit, Daten mit dem Server auszutauschen. Dadurch lassen sich mehr Informationen über die netzwerkbezogenen Aktivitäten des Programms gewinnen, wie die Kommunikation zu C2-Servern, Datenexfiltration und Versuche, andere Software herunterzuladen. Verdächtig ist zum Beispiel der Verbindungsaufbau zu IP-Adressen oder Domänen, die bereits als bösartig bekannt sind, zu IP-Adressen aus unüblichen Ländern, zu auffälligen Ports oder über ungewöhnliche Protokolle.

Zum Überwachen der Netzwerkverbindungen eignen sich Tools wie Wireshark, TCPLogView oder Sysinternals TCPView (siehe ix.de/z48a). Der Network-Tab von Process Hacker kann auch weiterhelfen. Das Tool zeigt allerdings nur aktive Verbindungen an, eine Historie gibt es nicht. Die Überwachungswerkzeuge sollte man prinzipiell auf einem anderen System als dem infizierten ausführen, denn so ist es für die Malware schwieriger zu entdecken, dass sie beobachtet wird – sie könnte versuchen, die Analysetools zu stören. Ein Sniffer zur Netzwerkanalyse kann auf jedem Rechner innerhalb des Labornetzwerks eingesetzt werden, vorausgesetzt, er arbeitet im promiskuitiven Modus. Dafür stellt man die Netzwerkschnittstelle so ein, dass sie alle Pakete im zugewiesenen Netzsegment erfasst und jedes empfangene Paket im Detail aufzeichnet.

Wireshark eignet sich gut zum Überwachen des Labornetzwerks. Der Netzwerk-Sniffer erfasst Pakete und speichert sie. Dabei gibt er die Quell- und die Ziel-IP-Adresse, das verwendete Protokoll, die Paketlänge und den Inhalt der Pakete an. Die erfassten Pakete kann man dann nach jeder Komponente filtern. Sie werden auch eingefärbt, in der Regel entsprechend dem verwendeten Protokoll. Wireshark gibt jedoch nicht an, welcher lokale Prozess an der Verbindung beteiligt war – an dieser Stelle können aber TCPLogView oder TCPView helfen, da sie ein Protokoll der lokalen TCP-Verbindungen speichern. Das zeigt, welcher Prozess welche Verbindung hergestellt hat.

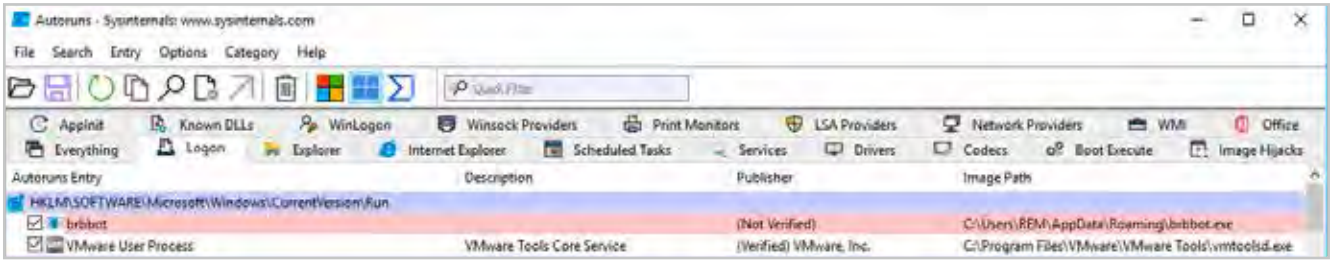
Damit die Malware auf die benötigten Ressourcen zugreifen kann, startet man eine interaktive dynamische Analyse mit Werkzeugen wie FakeDNS. FakeDNS löst alle Domänen auf, die von der Malware kontaktiert werden, wenn sie das System infiziert. Es beantwortet alle Anfragen nach Domänennamen mit einer konfigurierbaren IP-Adresse im Antwortdatensatz. Bei der Analyse enthält der Datensatz von FakeDNS die IP-Adresse der virtuellen Maschine, auf der er läuft.

FakeDNS und Wireshark zeigen beide deutlich, welche Domänen von Form.exe kontaktiert wurden, darunter www.3bmmbc6t.life und www.pwcgov.net (siehe Abbildung 4 und 5). In Abbildung 4 ist auch erkennbar, dass FakeDNS mit der IP-Adresse der Linux-Maschine antwortet (192.168.198.129). In Abbildung 5 ist zu sehen, dass die Resultate von Wireshark gefiltert wurden, um nur DNS-Anfragen anzuzeigen, bei denen die Kommunikation zwischen der infizierten Windows-Maschine (IP-Adresse 192.168.198.130) und der Linux-Maschine sichtbar ist.

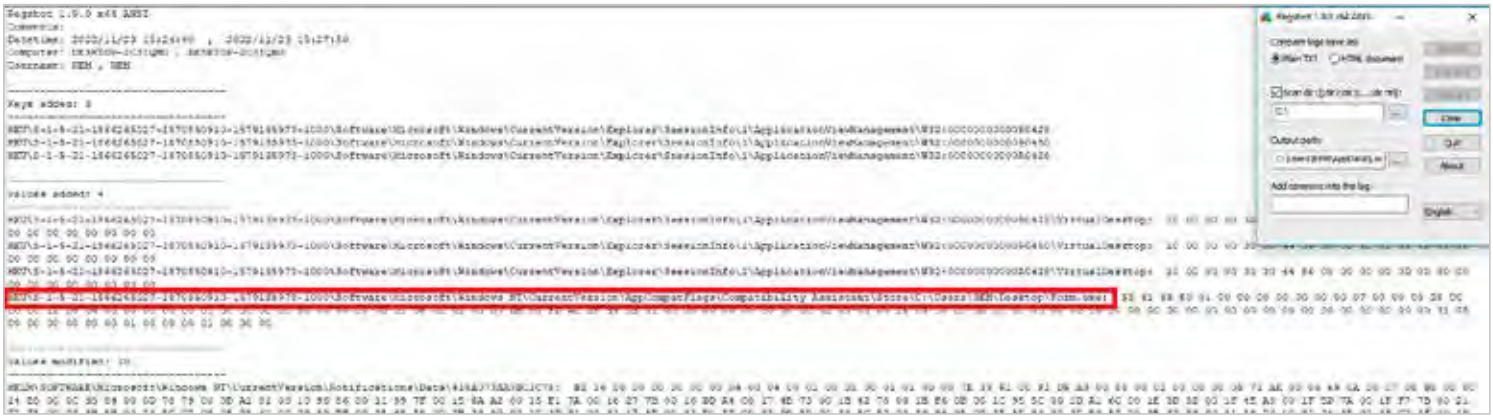
Hierbei handelt es sich um die Kommunikation mit einem C2-Server, schließlich hat Joe Sandbox www.pwcgov.net bereits als C2 aufgelistet. Die anderen Domänen sind auch unter „Domains and IPs“ in Joe Sandbox und unter „Relations“ auf VirusTotal zu finden. Die Anfragen an die Microsoft-Domänen sind hingegen legitim und entsprechen dem normalen Betrieb des Systems.

Systemänderungen überwachen

Schadsoftware nimmt häufig Änderungen am System vor, um einen Systemneustart zu überleben – oder unbemerkt auf dem System zu bleiben, indem sie beispielsweise Antivirenprogramme deaktiviert. Auf solche Maßnahmen können auch Änderungen an den Autostart-Speicherorten wie den Run-Registrierungsschlüsseln und im %AppData%\Roaming-Ordner hinweisen, die es der Malware ermöglichen, sich beim Anmelden des Benutzers oder beim Systemstart auszuführen.



Um persistent zu bleiben, kann Malware sich in den Autostart schleichen. Autoruns zeigt, dass brbbot.exe dafür einen Registrierungsschlüssel modifiziert (Abb. 6).



Regshot analysiert das Verhalten von Form.exe – die Malware nistet sich im Programmkompatibilitätsassistenten ein (Abb. 7).

Schadsoftware kann außerdem Dateien ablegen, wie nachgeladene Malware, Konfigurationsdateien oder Lösegeldforderungen von Ransomware als ReadMe-Textdateien.

Solche Änderungen kann man mit Werkzeugen wie Sysinternals Autoruns und Regshot überwachen. Autoruns zeigt Konfigurationen auf, die es erlauben, Malware persistent zu machen. Dazu gehören RunOnce-Registrierungsschlüssel, geplante Aufgaben und Dienste. Bei vielen schädlichen Programmen fehlen oft Angaben zum Herausgeber, eine Beschreibung oder die Version. Autoruns prüft die Signatur der Programme und kann auch Einträge mit VirusTotal abgleichen, wobei das Tool eine bestimmte Farbcodierung benutzt: Rosa bedeutet, dass keine Informationen zum Herausgeber gefunden wurden oder dass die digitale Signatur nicht existiert, bei gelber Färbung ist der Starteintrag vorhanden, aber die Datei, auf die er verweist, gibt es nicht mehr, und Grün zeigt Elemente an, die beim letzten Mal nicht im Autostart vorhanden waren.

Bei der Analyse von Form.exe fand Autoruns keine Persistenzeinträge. Um sie dennoch zu veranschaulichen, kommt an dieser Stelle eine andere beispielhafte Schadsoftware zum Einsatz: brbbot.exe (siehe Abbildung 6). brbbot.exe erstellt einen Eintrag im Registrierungsschlüssel Microsoft\Windows\CurrentVersion\Run. Das ermöglicht es der Malware,

```
---------------------------------
Files added: 12
---------------------------------
C:\ProgramData\Microsoft\Windows\WER\Temp\WER8D33.tmp.WERInternalMetadata.xml
C:\ProgramData\Microsoft\Windows\WER\Temp\WER8E1C.tmp.csv
C:\ProgramData\Microsoft\Windows\WER\Temp\WER8E5C.tmp.txt
C:\Users\All Users\Microsoft\Windows\WER\Temp\WER8D33.tmp.WERInternalMetadata.xml
C:\Users\All Users\Microsoft\Windows\WER\Temp\WER8E1C.tmp.csv
C:\Users\All Users\Microsoft\Windows\WER\Temp\WER8E5C.tmp.txt
C:\Users\REM\AppData\Local\Temp\aziztwhel.exe
C:\Users\REM\AppData\Local\Temp\gdeinuzidby.smf
C:\Users\REM\AppData\Local\Temp\nydkr.nr
C:\Users\REM\AppData\Local\Temp\RDR8D32.tmp\empty.txt
C:\Windows\Prefetch\AZIZTWHEL.EXE-72B7C94A.pf
C:\Windows\Prefetch\FORM.EXE-CAA82D32.pf
```

Mehrere Dateien wurden von der Malware neu angelegt. Dank Regshot können Forensiker sehen, wie sie an unverdächtigen Stellen auftauchen (Abb. 8).

dauerhaft auf dem System zu bleiben und bei jeder Anmeldung gestartet zu werden. Weil es weder eine Beschreibung noch einen Herausgeber gibt, hebt Autoruns ihn in Rosa hervor.

Regshot vergleicht den Zustand des Systems vor und nach der Infektion, indem es Schnappschüsse erstellt und einen Bericht generiert, der die Änderungen am Dateisystem und der Registrierung wie neue Dateien oder Einträge in der Registrierung aufzeigt, die während der Infektion entstanden. Das liefert Hinweise auf Persistenzmechanismen der Malware. Regshot kann jedoch keine Informationen über die spezifische Abfolge von Ereignissen liefern oder angeben, welcher Prozess für die Änderungen verantwortlich ist. Dafür ist Process Monitor besser geeignet.

Im Beispiel von Form.exe zeigt Regshot aber, dass der Eintrag C:\Users\REM\Desktop\Form.exe dem Registrierungsschlüssel Software\Microsoft\Windows NT\CurrentVersion\AppCompatFlags\Compatibility Assistant\Store hinzugefügt wurde (siehe rote Box in Abbildung 7). In diesem Registrierungsschlüssel erscheinen die Programme, bei denen der Programmkompatibilitätsassistent verwendet wird. Er verfolgt ein Programm, wenn es ausgeführt wird, und identifiziert alle Anzeichen bekannter Kompatibilitätsprobleme – Programme, die in diesem Registrierungsschlüssel auftauchen, sind also erst einmal unverdächtig.

Regshot macht aber auch sichtbar, dass mehrere Dateien durch das Ausführen von Form.exe geändert oder hinzugefügt wurden (in der oberen roten Box in Abbildung 8 zu sehen). aziztwhel.exe, nydkr.nr und gdeinuzidby.smf sind alle im Temp-Ordner gelandet, außerdem hat die Malware pf-Dateien (Prefetch) für Form.exe und aziztwhel.exe erstellt (siehe untere rote Box in Abbildung 8). Prefetch ist ein Prozess, bei dem das Betriebssystem wichtigen Code und Daten von der Festplatte in den Speicher lädt, bevor sie benötigt werden. Der Prefetch-Ordner wird nach der ersten Ausführung eines Programmes mit einer .pf-Datei gefüllt. Auch das ist normales Systemverhalten.

Die Dateien im Temp-Ordner sind auch unter „Dropped files“ im Bericht von Joe Sandbox zu sehen, die Registrierungsschlüssel und andere Änderungen am Dateisystem findet man dort ebenfalls unter „Behavior“. Joe Sandbox und VirusTotal ließen bereits vermuten, dass es sich um eine Schadsoftware, höchstwahrscheinlich Formbook, handelt, und die Überprüfung der IOCs auf Onlinediensten wie Malpedia deutet ebenfalls darauf hin. Statische und dynamische Analyse haben also vermitteln können, wie die Malware arbeitet – und erlauben es dadurch auch, Gegenmaßnahmen zu ergreifen.

Fazit

Durch das Überwachen von Prozessen, Netzwerkaktivitäten und Systemänderungen kann eine dynamische Analyse die Fähigkeiten einer Schadsoftware bestimmen, wie die Kommunikation mit einem Command-and-Control-Server oder die Injektion in andere Prozesse, um unentdeckt zu bleiben. Diese Aktivitäten können aber nur dann aufgedeckt werden, wenn bereits bekannt ist, wie das System normalerweise funktioniert. Die dynamische Analyse vervollständigt am Schluss die Liste der IOCs, die Forensiker in den vorherigen Analysephasen identifizierten.

Allerdings ist es kaum möglich, alles, was die Malware anrichten kann, nur mit einer dynamischen Analyse herauszufinden – insbesondere, wenn die Malware fähig ist zu erkennen, dass sie analysiert wird. Man kann zusätzlich noch eine Arbeitsspeicheranalyse durchführen, beispielsweise mit dem Speicherforensik-Framework Volatility. Codeanalyse (Reverse Engineering) führt ebenfalls zu einem tiefgreifenden Verständnis der Funktionsweise der Schadsoftware, ist jedoch zeitaufwendig. Beide Techniken bereiten zwar den Weg für weiter gehende Analysen – die vier in der Artikelreihe genannten Schritte liefern aber bereits ein ziemlich klares Bild von der fraglichen Schadsoftware.

Malware stellt ein tägliches Risiko für Unternehmen dar und verschiedene Arten von Angreifern setzen verschiedene Arten von Schadsoftware ein. Hat man eine verdächtige Datei in den Händen, ist Vorsicht geboten – selbst Onlineplattformen und Sandboxes bergen Risiken. Angreifer setzen außerdem natürlich alles daran, ihre Software so gut wie möglich zu verbergen – eine statische Analyse kann zum Beispiel die Tarnung als Makro auffliegen lassen, mit der Schadsoftware in PDF- und Office-Dokumenten versteckt ist. Informationen, die man durch solche Analysen gewinnt, kann man in der dynamischen Analyse dann noch einmal erhärten und genau hinschauen: Wie verhält sich die Malware, wie kommuniziert sie? Sämtliche Erkenntnisse hängen aber von einem gut sortierten Werkzeugkasten ab – verschiedene Werkzeuge sind in unterschiedlichen Phasen der Analyse hilfreich.

Schlussendlich dürfen Forensiker aber bei aller Detektivarbeit nicht vergessen, dass Schadsoftware auch während der Analyse Risiken birgt. Oberste Maxime sollte daher sein, sich selbst entsprechend zu schützen. (kki@ix.de)

Quellen

[1] Nadia Meichtry, Fabian Murer, Tabea Nordieker; Einstieg in die Malware-Analyse; iX 2/2023, S. 98

[2] Nadia Meichtry, Fabian Murer, Tabea Nordieker; Malware Analyse per OSINT und Sandbox; iX 3/2023, S. 122

[3] Nadia Meichtry, Statische Malware-Analyse; iX 5/2023, S. 132

[4] Die Vorgestellten Tools und weitere Informationen finden sich unter ix.de/z48a.

NADIA MEICHTRY

ist Digital-Forensics- und Incident-Response-Spezialistin bei der Oneconsult AG. Sie unterstützt bei der Bewältigung und Untersuchung von Cybervorfällen.