Module 04

Detecting Live Systems

Scanning Techniques

Detecting Live Systems

Detecting Live Systems uses scanning techniques, and refers to a set of procedures for identifying hosts, ports, and services running in a target.

ICON KEY


Important Information

Quiz


CPTE Labs


Course Review

Once a target is identified and researched from reconnaissance efforts, the next step is to evaluate the target for vulnerabilities. At this point, the Penetration Tester should know enough about a target to select how to analyze for possible vulnerabilities or weaknesses.

Vulnerability Assessments and Security Audits typically conclude after this phase of the target evaluation process.

Lab Objectives

The objective of these labs is to help students in conducting network scanning, analyzing network vulnerabilities, and maintaining a secure network.
The lab objectives:

- Detect 'live' systems on target network
- Perform banner grabbing and OS fingerprinting
- Map and Draw network diagrams of vulnerable hosts
- Discover services running/listening on target systems
- Understand port scanning techniques
- Identify TCP and UDP services running on target network
- Discover the operating system
- Understand active and passive fingerprinting
- Automated discovery tools
- Identify network vulnerabilities

Lab Scenario

Vulnerability scanning determines the possibility of network security attacks. It evaluates the organization's systems and network for vulnerabilities such as missing patches, unnecessary services, weak authentication, and weak encryption.

Vulnerability scanning is a critical component of any penetration testing assignment.

You need to conduct a penetration test and list the threats and vulnerabilities found in an organization's network and perform port scanning, network scanning, and vulnerability scanning to identify IP/hostname, live hosts, and vulnerabilities.

Lab Environment

This lab requires:

- **Windows 7**
- **Kali Linux VM**
- A Firefox web browser with Internet connection
- Administrative privileges to run tools

Lab Duration

Time: 50 Minutes

Overview of Network Scanning

Network scanning is a procedure for identifying active hosts on a network, either for the purpose of attacking them or for network security assessments. Scanning procedures, such as ping sweeps and port scans, return information about which IP addresses map to live hosts that are active on the Internet and what services they offer. Another scanning method, inverse mapping, returns information about what IP addresses do not map to live hosts; this enables an attacker to make assumptions about viable addresses.

Scanning is one of three components of intelligence gathering for an attacker. In the footprinting phase (Information Gathering), the attacker creates a profile of the target organization, with information such as its domain name system (DNS) and e-mail servers, and its IP address range. Most of this information is available online. In the scanning phase, the attacker finds information about the specific IP addresses that can be accessed over the Internet, their operating systems, the system architecture, and the services running on each computer. In the enumeration phase, the attacker gathers information such as network user and group names, routing tables, and Simple Network Management Protocol (SNMP) data.

Building on what we learned from our information gathering and threat modeling, we can now begin to actively query our victims for vulnerabilities that may lead to a compromise. We have narrowed down our attack surface considerably since we first began the penetration test with everything potentially in scope.

Lab Tasks

Recommended labs to assist you in Detecting Live Systems:

- **Lab 1: Exploring and Auditing a Network Using Nmap**
- **Lab 2: Exploring and Auditing a Network Using ZenNmap**
- **Lab 3: Exploring and Auditing a Network Using Hping3**
- **Lab 4: Scanning a Network Using the PBNJ (Store Nmap Scan in Database)**

Lab Analysis

Analyze and document the results related to the lab. Give your opinion on your target's security posture and exposure through public and free information.

Lab 1

Scanning a Target Using nmap Tools

Nmap Overview

Nmap is designed to allow system administrators and curious individuals to scan large networks to determine which hosts are up and what services they are offering. Nmap supports a large number of scanning techniques such as: UDP, TCP connect(), TCP SYN (half open), ftp proxy (bounce attack), ICMP (ping sweep), FIN, ACK sweep, Xmas Tree, SYN sweep, IP Protocol, and Null scan. See the Scan Types section for more details. Nmap also offers a number of advanced features such as remote OS detection via TCP/IP fingerprinting, stealth scanning, dynamic delay and retransmission calculations, parallel scanning, detection of down hosts via parallel pings, decoy scanning, port filtering detection, direct (non-portmapper) RPC scanning, fragmentation scanning, and flexible target and port specification.

Lab Scenario

As an administrator it is very important for you to patch those systems after you have determined all the vulnerabilities in a network, before the attacker audits the network to gain vulnerable information.

Also, as a penetration tester and network administrator for your company, your job is to carry out daily security tasks, such as network inventory, service upgrade schedules, and the monitoring of host or service uptime. You will be guided in this lab to use Nmap to explore and audit a network.

The objective of this lab is to help students learn and understand how to perform a network inventory, manage services and upgrades, schedule network tasks, and monitor host or service uptime and downtime.
In this lab, you will need to:

- Scan TCP and UDP ports
- Analyze host details and their topology
- Determine the types of packet filters
- Record and save all scan reports
- Compare saved results for suspicious ports

Lab Resources

To run this lab, you will need the following:

- A **Kali Linux VM** as a host machine

- A Firefox web browser with Internet access
- Root privileges to run the Nmap tool

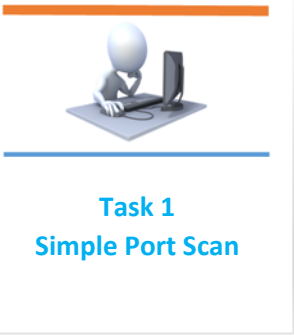
Lab Duration

Time: 20 Minutes

Nmap Syntax

Use Kali Linux VM

```
nmap [Scan Type(s)] [Options] <host or net #1 ... [#N]>
```



Task 1
Simple Port Scan

Lab Tasks

1. Log into Kali Linux using username=root/password=toor

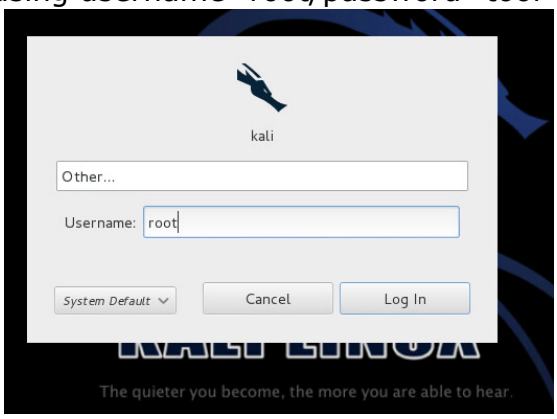


Figure: 1.1- Kali Linux – Login Screen

2. Start NMAP using: Applications > Kali Linux > Top 10 Security Tools > nmap

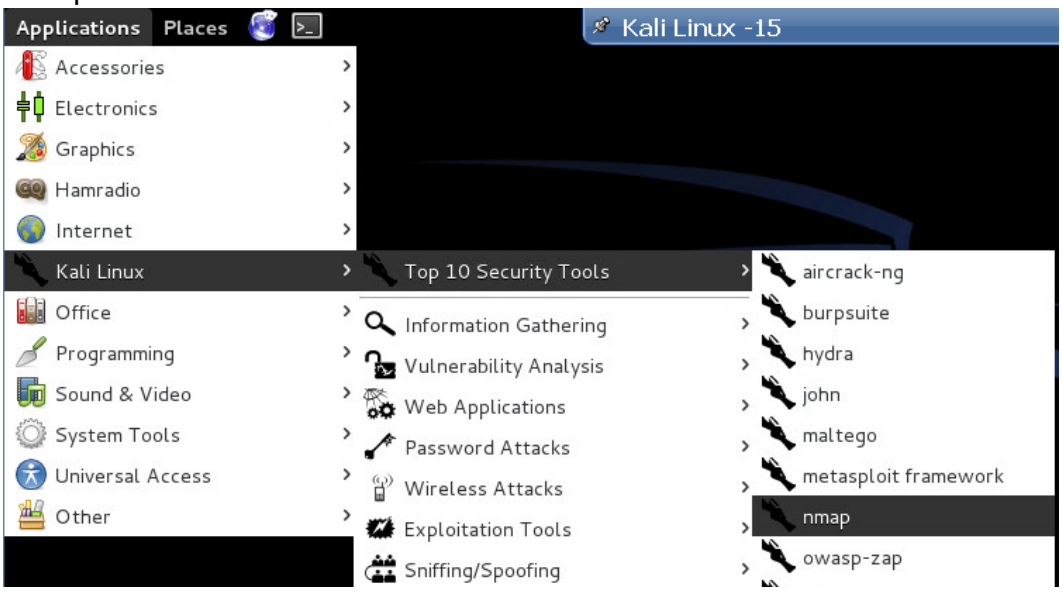


Figure: 1.2- Kali Linux – Top 10 Security Tools - Nmap

3. The Nmap window appears.

Most of the following scans are directed toward the Windows 2012 VM.

In a port scan technique, only one method may be used at time, except that UDP scan (-sU) and any one of the SCTP scan types (-sY, -sZ) may be combined with any one of the TCP scan types.

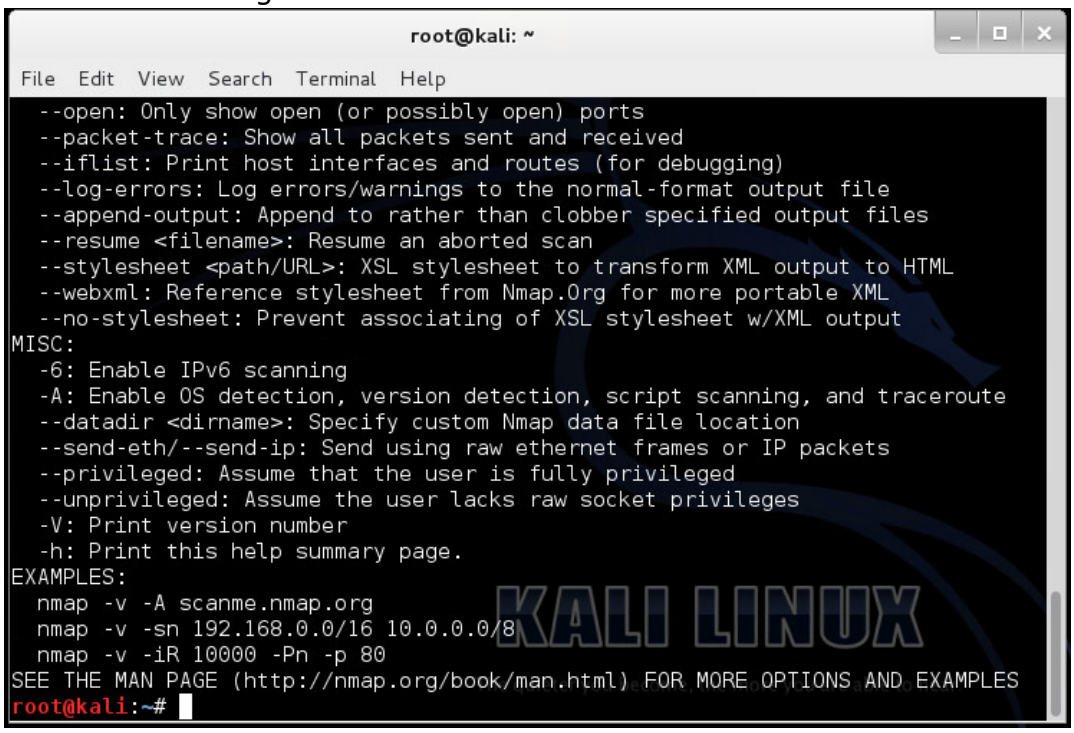


Figure: 1.3- Nmap syntax and usage Screen

4. Enter the IP address of the Windows Server 2012 virtual machine (192.168.2.42 as in our example, please use your IP address for the Windows Server 2012) using the following command:

Figure: 1.4- Nmap Simple Scan as root

5. Note that running this scan as a root user is actually equivalent to running nmap -sS 192.168.2.42:

While Nmap attempts to produce accurate results, keep in mind that all of its insights are based on packets returned by the target machines or the firewalls in front of them.

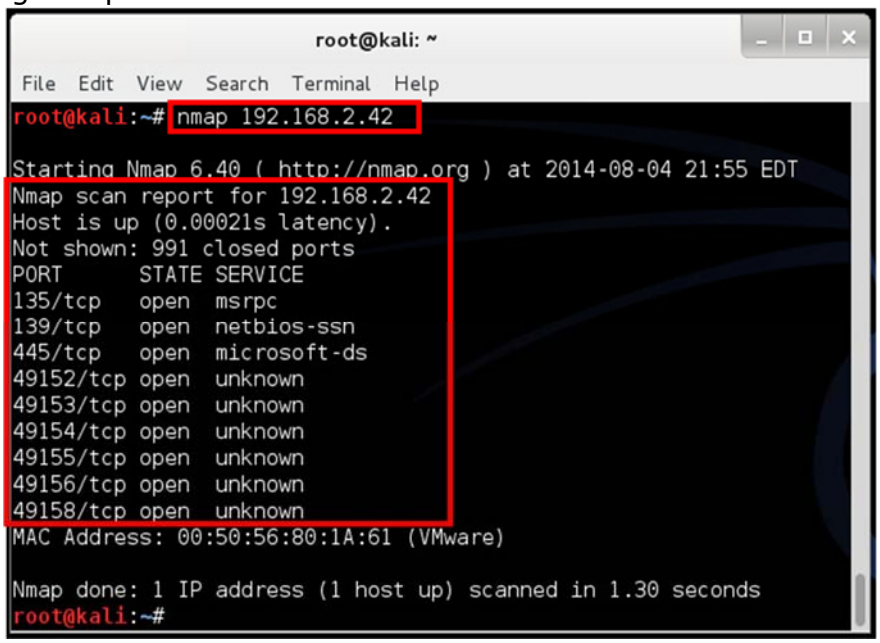


Figure: 1.5- Nmap Simple Scan Results

Nmap accepts multiple host specifications on the command line, and they don't need to be of the same type.

Task 2
Scan a Targed by explicitly specify the ports

The options available to control target selection:

- -iL <inputfilename>
- -iR <num hosts>
- -exclude <host 1 > [,<host2> [,...]]
- -excludefile <exclude file>

Task 3

Network Sweeping

6. The scan identified many open ports on 192.168.2.42, but are these all the open ports on this machine?
7. Next, try port scanning all of the available ports directed toward the Windows 2012 vm by explicitly specifying the ports to be scanned:

```
root@kali:~# nmap -p 1-65535 192.168.2.42
```

Figure: 1.6- Nmap Scan by explicitly specifying the ports

8. Notice how you've discovered some open ports that were not initially scanned because they are not present in the Nmap default port configuration file (/usr/share/nmap/nmap-services).

```
root@kali:~# nmap -p 1-65535 192.168.2.42

Starting Nmap 6.40 ( http://nmap.org ) at 2014-08-04 22:19 EDT
Nmap scan report for 192.168.2.42
Host is up (0.00018s latency).
Not shown: 65524 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5985/tcp  open  wsman
47001/tcp open  unknown
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49158/tcp open  unknown
MAC Address: 00:50:56:80:1A:61 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 28.11 seconds
root@kali:~#
```

Figure: 1.7- Nmap Scan Results by explicitly specifying the ports

9. **Network Sweeping:** Rather than scanning a single machine for all ports, scan all the machines for one port (139). This example could be useful for identifying all the computers running NetBIOS/SMB services:

```
root@kali:~# nmap -p 139 192.168.2.*
```

Figure: 1.8- Nmap Scan by explicitly specifying on port (139)

10. The scan is complete, but you see that the output is not script-friendly. Nmap supports several output formats.

By default, Nmap performs a host discovery and then a port scan against each host it determines to be on-line.

```
root@kali:~# nmap -p 139 192.168.2.*

Starting Nmap 6.40 ( http://nmap.org ) at 2014-08-04 22:43 EDT
Nmap scan report for 192.168.2.25
Host is up (0.00051s latency).
PORT    STATE SERVICE
139/tcp open  netbios-ssn
MAC Address: 00:50:56:8D:53:7B (VMware)

Nmap scan report for 192.168.2.28
Host is up (0.00046s latency).
PORT    STATE SERVICE
139/tcp open  netbios-ssn
MAC Address: 00:50:56:8D:76:8B (VMware)

Nmap scan report for 192.168.2.41
Host is up (0.0010s latency).
PORT    STATE  SERVICE
```

Figure: 1.9- Nmap Scan Results by explicitly specifying on port (139)

Task 4

Scan with greppable format -oG

11. One of my favorites is the greppable format (-oG output text file):

```
root@kali:~# nmap -p 139 192.168.2.* -oG 139.txt
root@kali:~# cat 139.txt
# Nmap 6.40 scan initiated Mon Aug  4 22:53:05 2014 as: nmap -p 13
9 -oG 139.txt 192.168.2.*
Host: 192.168.2.25 ()   Status: Up
Host: 192.168.2.25 ()   Ports: 139/open/tcp//netbios-ssn///
Host: 192.168.2.28 ()   Status: Up
Host: 192.168.2.28 ()   Ports: 139/open/tcp//netbios-ssn///
Host: 192.168.2.41 ()   Status: Up
Host: 192.168.2.41 ()   Ports: 139/closed/tcp//netbios-ssn///
Host: 192.168.2.42 ()   Status: Up
Host: 192.168.2.42 ()   Ports: 139/open/tcp//netbios-ssn///
Host: 192.168.2.55 ()   Status: Up
Host: 192.168.2.55 ()   Ports: 139/closed/tcp//netbios-ssn///
Host: 192.168.2.65 ()   Status: Up
Host: 192.168.2.65 ()   Ports: 139/closed/tcp//netbios-ssn///
Host: 192.168.2.69 ()   Status: Up
Host: 192.168.2.69 ()   Ports: 139/closed/tcp//netbios-ssn///
Host: 192.168.2.70 ()   Status: Up
Host: 192.168.2.70 ()   Ports: 139/closed/tcp//netbios-ssn///
Host: 192.168.2.71 ()   Status: Up
root@kali:~# cat 139.txt | grep open | cut -d" " -f2
192.168.2.25
192.168.2.28
192.168.2.42
root@kali:~#
```

Figure: 1.9- Nmap Scan with Greppable format -oG

Task 5

OS Fingerprinting

Unfortunately, this feature is still a bit buggy over VPN tunnels and does not work as expected in the labs.

You see that 192.168.2.42 is most probably running Windows - possibly Windows 7 or Windows 2012.

12. You've found several IP addresses with open port 139. You still do not know, however, which operating systems are present on these IPs.
nmap has a wonderful feature called OS fingerprinting (-O). This feature attempts to guess the underlying operating system by inspecting the packets received from the machine. As it turns out, each vendor implements the TCP/IP stack slightly differently (default TTL values, windows size), and these differences create an almost unique fingerprint:

```
root@kali:~# nmap -O 192.168.2.42

Starting Nmap 6.40 ( http://nmap.org ) at 2014-08-04 23:16 EDT
Nmap scan report for 192.168.2.42
Host is up (0.00036s latency).
Not shown: 991 closed ports
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
49158/tcp open  unknown
MAC Address: 00:50:56:80:1A:61 (VMware)
Device type: general purpose
Running: Microsoft Windows 7|2012
OS CPE: cpe:/o:microsoft:windows_7:::ultimate cpe:/o:microsoft:win
dows_2012
OS details: Microsoft Windows 7 or Windows Server 2012
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at htt
p://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 3.35 seconds
root@kali:~#
```

Figure: 1.10- Nmap OS Fingerprinting -O

13. The Nmap Scripting Engine (NSE) is a recent addition to Nmap that allows users to write simple scripts to automate a wide variety of networking tasks. The scripts include a wide variety of utilities, from DNS enumeration scripts, brute force attack scripts, and even vulnerability identification scripts. A list of these scripts can be found in the /usr/local/share/nmap/scripts directory:

Task 6

Nmap Scripting Engine

```
root@kali:~# locate *.nse
/usr/share/exploitdb/platforms/hardware/webapps/31527.nse
/usr/share/golismero/wordlist/fingerprint/httprecon/httprecon.nse
/usr/share/nmap/scripts/acarsd-info.nse
/usr/share/nmap/scripts/address-info.nse
/usr/share/nmap/scripts/afp-brute.nse
/usr/share/nmap/scripts/afp-ls.nse
/usr/share/nmap/scripts/afp-path-vuln.nse
/usr/share/nmap/scripts/upnp-info.nse
/usr/share/nmap/scripts/url-snarf.nse
/usr/share/nmap/scripts/ventrilo-info.nse
/usr/share/nmap/scripts/versant-info.nse
/usr/share/nmap/scripts/vmauthd-brute.nse
/usr/share/nmap/scripts/vnc-brute.nse
/usr/share/nmap/scripts/vnc-info.nse
/usr/share/nmap/scripts/voldemort-info.nse
/usr/share/nmap/scripts/vuze-dht-info.nse
/usr/share/nmap/scripts/wdb-version.nse
/usr/share/nmap/scripts/whois.nse
/usr/share/nmap/scripts/wsdd-discover.nse
/usr/share/nmap/scripts/x11-access.nse
/usr/share/nmap/scripts/xdmcp-discover.nse
/usr/share/nmap/scripts/xmpp-brute.nse
/usr/share/nmap/scripts/xmpp-info.nse
root@kali:~#
```

Figure: 1.11- Nmap Scripting Engine – Sample output

14. The scripts contain descriptions in their source code, which also has usage examples: (Use Metasploitable VM as the target)

```
root@kali:~# nmap 192.168.2.28 --script smb-enum-users.nse
```

```
root@kali:~# nmap 192.168.2.28 --script smb-enum-users.nse

Starting Nmap 6.40 ( http://nmap.org ) at 2014-08-06 11:15 EDT
Nmap scan report for 192.168.2.28
Host is up (0.00035s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:50:56:8D:76:8B (VMware)

Host script results:
| smb-enum-users:
|   METASPLOITABLE\backup (RID: 1068)
|     Full name:   backup
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\bin (RID: 1004)
|     Full name:   bin
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\bind (RID: 1210)
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\daemon (RID: 1002)
|     Full name:   daemon
|     Flags:       Normal user account, Account disabled
|   METASPLOITABLE\dhcp (RID: 1202)
```

Figure: 1.12- Nmap Scripting Engine Scan – Sample output

Lab 2

Scanning a Target Using Zenmap Tools

ZeNmap Overview

Zenmap is the official Nmap Security Scanner GUI. It is a multi-platform (Linux, Windows, Mac OS X, BSD, etc.) free and open source application which aims to make Nmap easy for beginners to use while providing advanced features for experienced Nmap users. Frequently used scans can be saved as profiles to make them easy to run repeatedly. A command creator allows interactive creation of Nmap command lines. Scan results can be saved and viewed later. Saved scan results can be compared with one another to see how they differ. The results of recent scans are stored in a searchable database.

Lab Scenario

In the previous lab, you learned to use the Nmap tool to scan a network to find out the vulnerability level, system patching status, details for open and closed ports, vulnerable computers, etc. As an administrator and an attacker, you can use the same tools to fix or exploit a system. If an attacker gets to know all the information about vulnerable computers, they will immediately act to compromise those systems using reconnaissance techniques.

Therefore, as an administrator it is very important for you to patch those systems after you have determined all the vulnerabilities in a network, before the attacker audits the network to gain vulnerable information.

Also, as a penetration tester and network administrator for your company, your job is to carry out daily security tasks, such as network inventory, service upgrade schedules, and the monitoring of host or service uptime. You will be guided in the lab to use ZeNmap to explore and audit a network.

The objective of this lab is to help students learn and understand how to perform a network inventory, manage services and upgrades, schedule network tasks, and monitor host or service uptime and downtime.

In this lab, you will need to:

- Scan TCP and UDP ports
- Analyze host details and their topology
- Determine the types of packet filters
- Record and save all scan reports
- Compare saved results for suspicious ports

Lab Resources

To run this lab, you will need the following:

- Nmap located on Desktop

Zenmap works on Windows versions after and including Windows 7, and Server 2003/2008.

ICON KEY

- A computer running Windows 7 VM as host machine
- Windows Server 2008 running on a virtual machine as a guest
- A Firefox web browser with Internet access
- Administrative privileges to run the ZeNmap tool

Lab Duration

Time: 20 Minutes

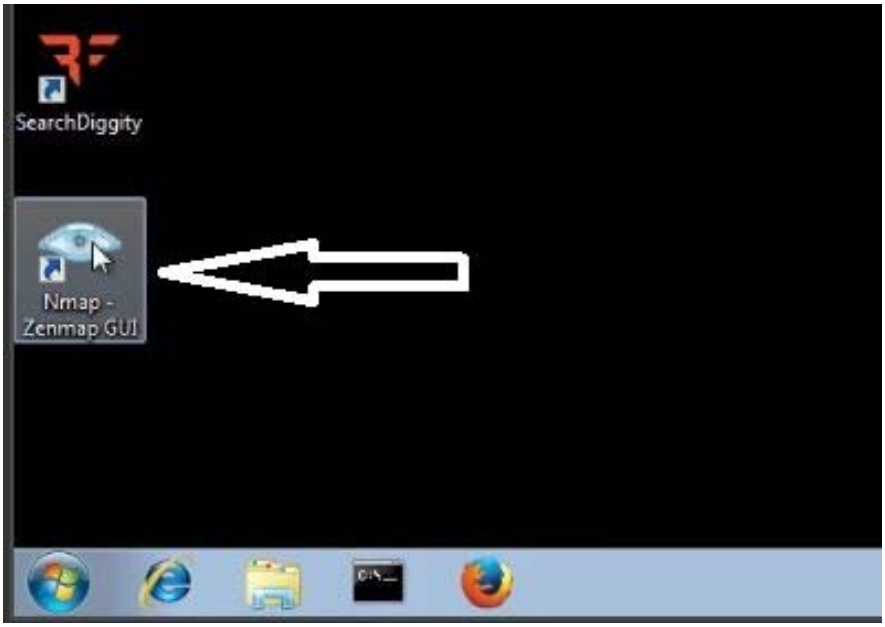
Lab Tasks

Follow the wizard-driven installation steps and install Nmap (Zenmap) scanner on the host machine (Use Windows 7 VM).

Task 1

Intense Scan

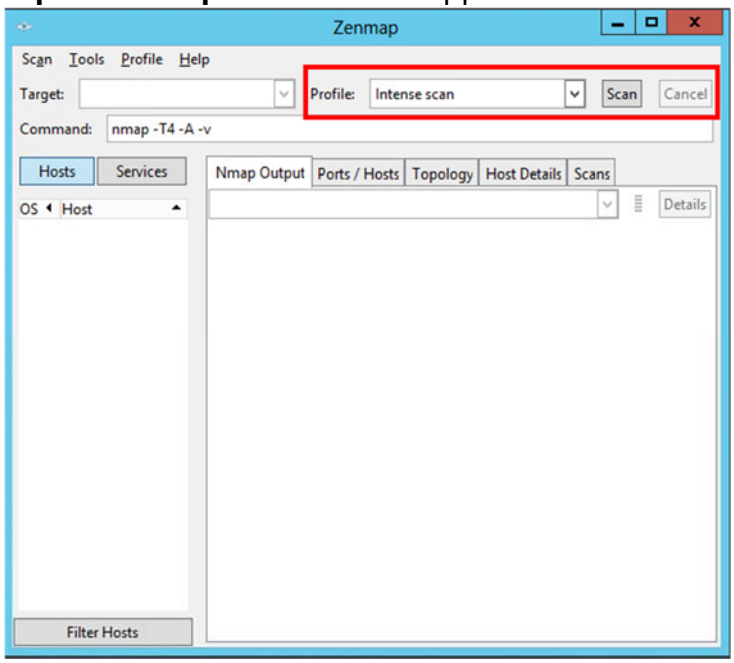
1. Click the Nmap-Zenmap GUI icon on the Desktop to open the Zenmap window



Zenmap file installs the following files:

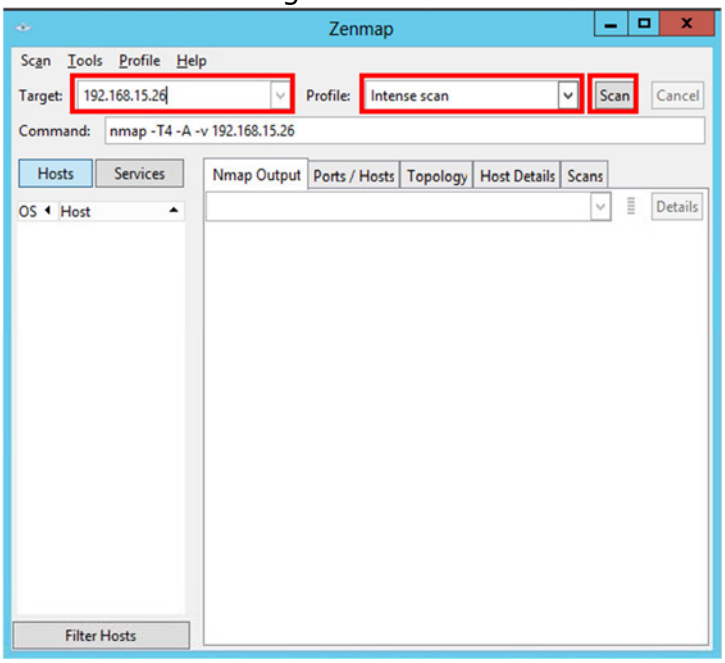
- Nmap Core Files
- Nmap Path
- WinPcap 4.1.1
- Network Interface
- Import
- Zenmap (GUI frontend)
- Neat (Modem Netcat)
- Ndiff

2. The **Nmap - Zenmap GUI** window appears.



Zenmap GUI main window

3. Enter the virtual machine IP address for your Windows Server 2008 (192.168.15.26, in our example) in the target: text field. You are performing a network inventory for the virtual VM.
4. In the Profile: text field, select, from the drop-down list, the type of profile you want to scan. In this lab, select Intense Scan.
5. Click Scan to start scanning the virtual machine.



Zenmap GUI main window with Target and Profile Entered

The six port states recognized by Nmap:

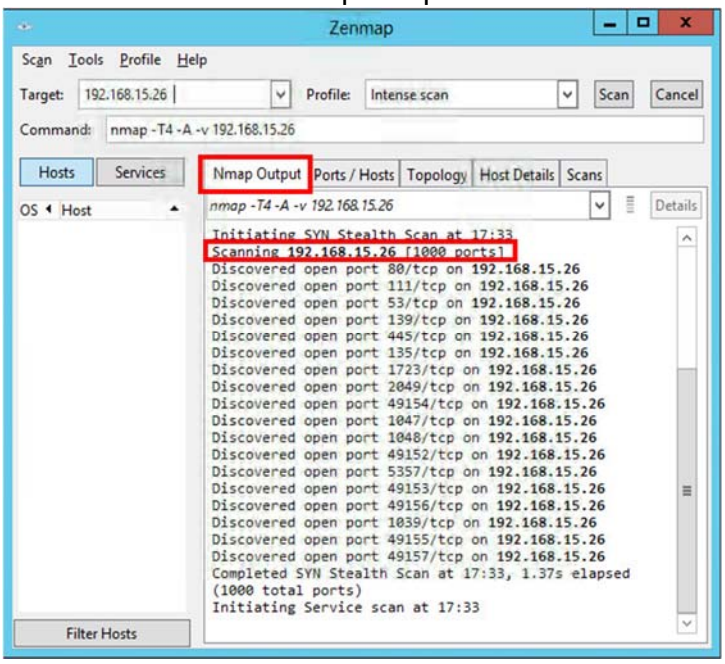
- Open
- Closed
- Filtered
- Unfiltered
- Open|Filtered
- Closed|Unfiltered

Nmap accepts multiple host specifications on the command line, and they don't need to be of the same type.

The options available to control target selection:

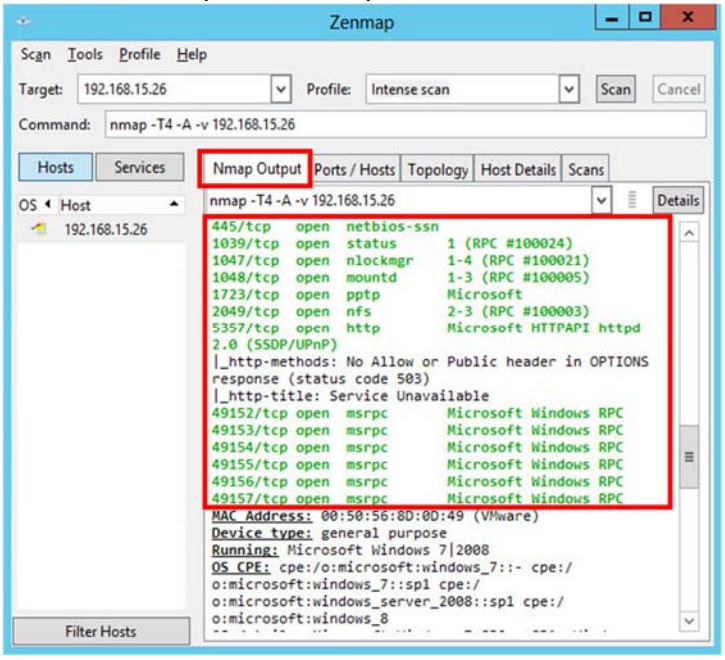
- -iL <inputfilename>
- -1R <num hosts>
- --exclude <host 1 > [,<host2> [,...]]
- --excludefile <exclude file>

6. Nmap scans the provided IP address with **Intense scan** and displays the scan result below the Nmap Output tab.



Zenmap window with the Nmap Output tab for Intense Scan

7. After the scan is complete, Nmap shows the scanned results.

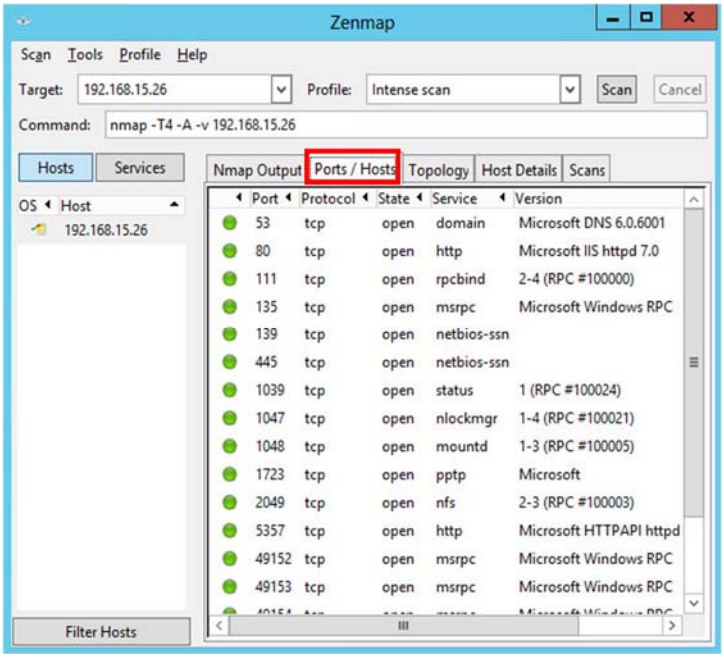


Zenmap window with the Nmap Output tab for Intense Scan

8. Click the **Ports / Hosts** tab to display more information on the scan results.
9. Nmap also displays the **Port, Protocol, State, Service,** and **Version** of the scan.

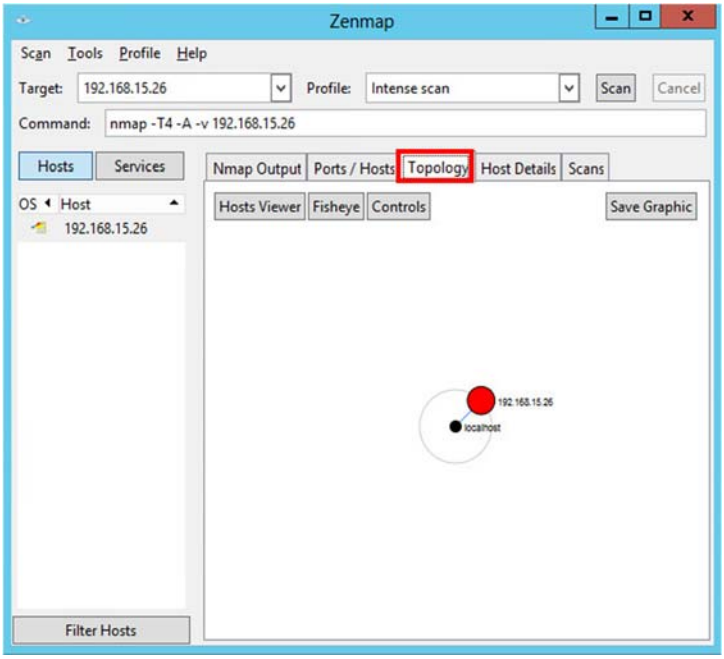
The following options control host discovery:

- -sL (list Scan)
- -sn (No port scan)
- -Pn (No ping)
- -PS <port list> (TCP SYN Ping)
- -PA <port list> (TCP ACK Ping)
- -PU <port list> (UDP Ping)
- -PY <port list> (SCTP INTT Ping)
- -PE;-PP;-PM (ICMP Ping Types)
- -PO <protocol list> (IP Protocol Ping)
- -PR (ARP Ping)
- --traceroute (Trace path to host)
- -n (No DNS resolution)
- -R (DNS resolution for all targets)
- -system-dns (Use system DNS resolver)
- -dns-servers < server 1 > [,< server 2 > [,...]] (Servers to use for reverse DNS queries)



Zenmap window with the Ports/Hosts tab for Intense Scan

10. Click the Topology tab to view Nmap's topology for the provided IP address in the Intense scan Profile.

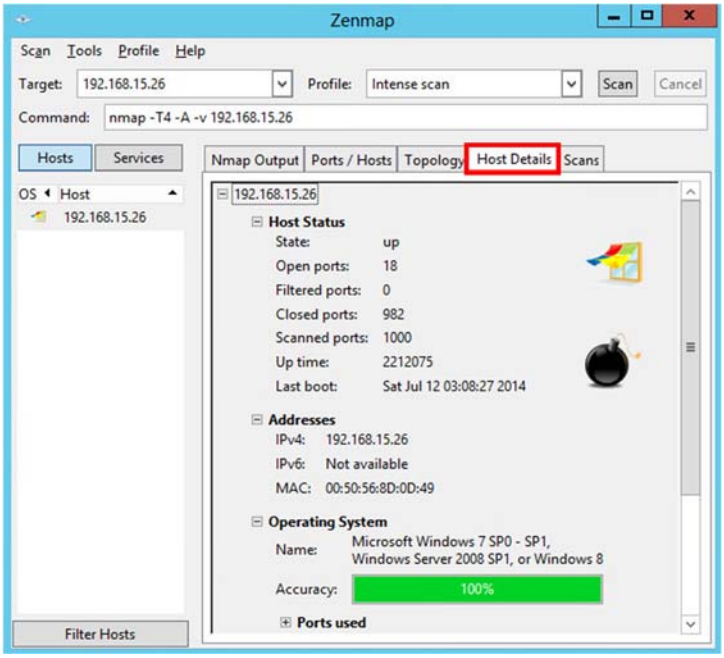


Zenmap window with the Ports/Hosts tab for Intense Scan

11. Click the Host Details tab to see the details of all hosts discovered during the intense scan profile.

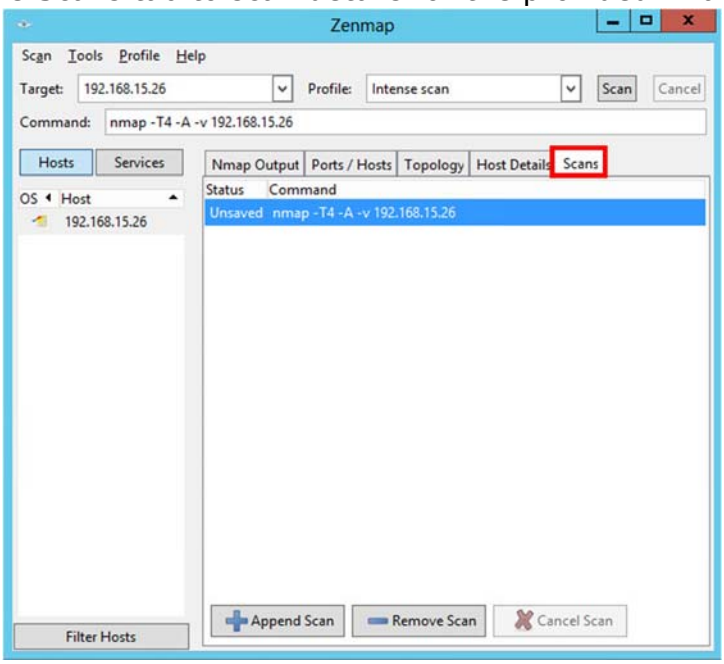
By default, Nmap determines your DNS servers (for DNS resolution) from your resolv.conf file (UNIX) or the Registry (Win32).

Nmap offers options for specifying which ports are scanned and whether the scan order is randomized or sequential.



Zenmap window with Host Details tab for Intense Scan

12. Click the Scans tab to scan details for the provided IP addresses.



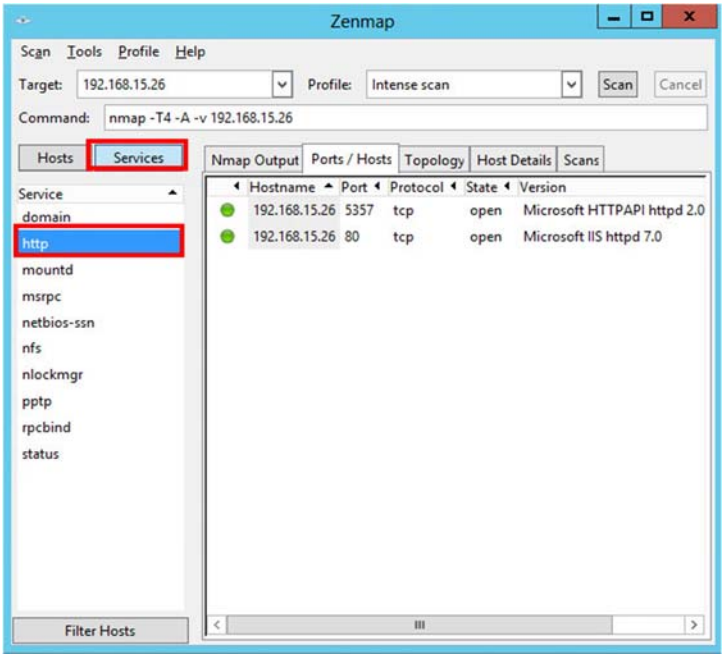
Zenmap window with Scan tab for Intense Scan

13. Now, click the Services tab located in the right pane of the window. This tab displays the list of services.
14. Click the http service to list all the HTTP Hostnames/IP addresses, ports, and their states (Open/Closed).

In Nmap, option -p <port ranges> means scan only specified ports.

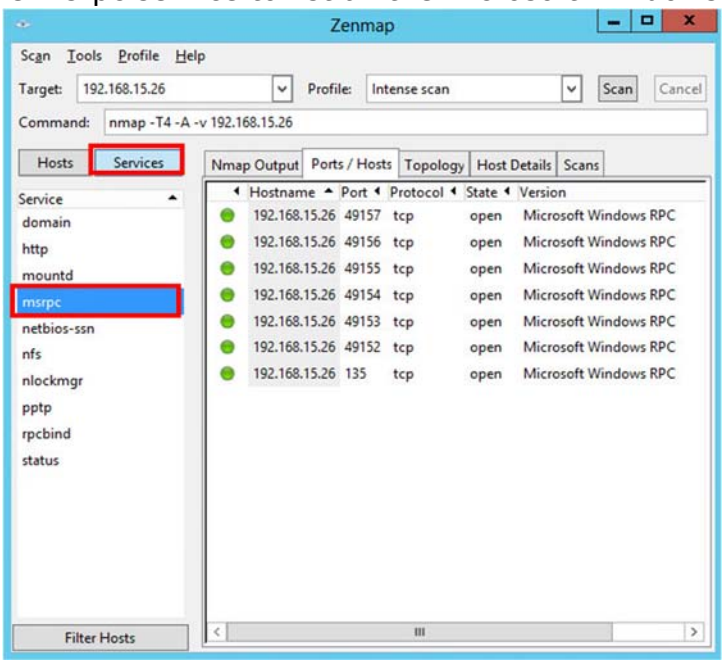
In Nmap, option -F means fast (limited port)

In Nmap, Option - port-ratio <ratio><decimal number between 0 and 1> means Scans all ports in nmap-services file with a ratio greater than the one given. <ratio> must be between 0.0 and 1.1



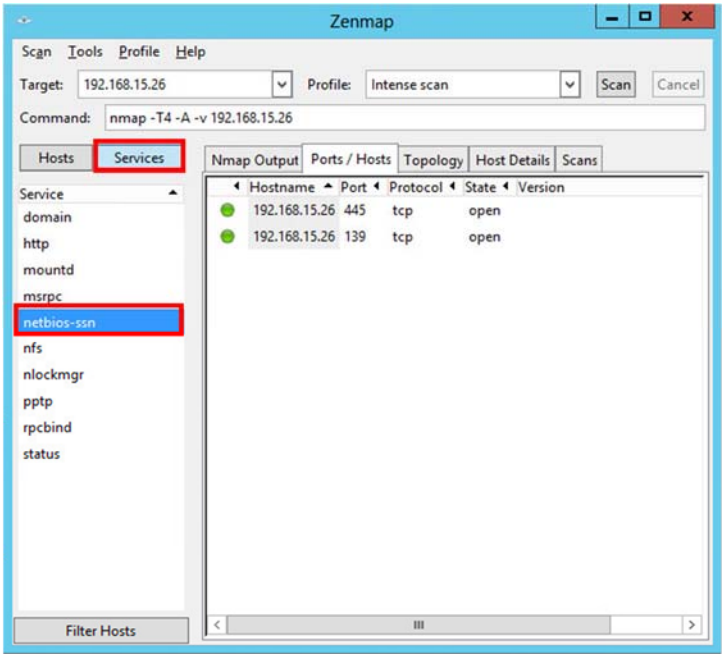
Zenmap window with Services option for Intense Scan

15. Click the msrpc service to list all the Microsoft Windows RPC.



Zenmap window with msrpc Service for Intense Scan

16. Click the netbios-ssn service to list all NetBIOS hostnames.

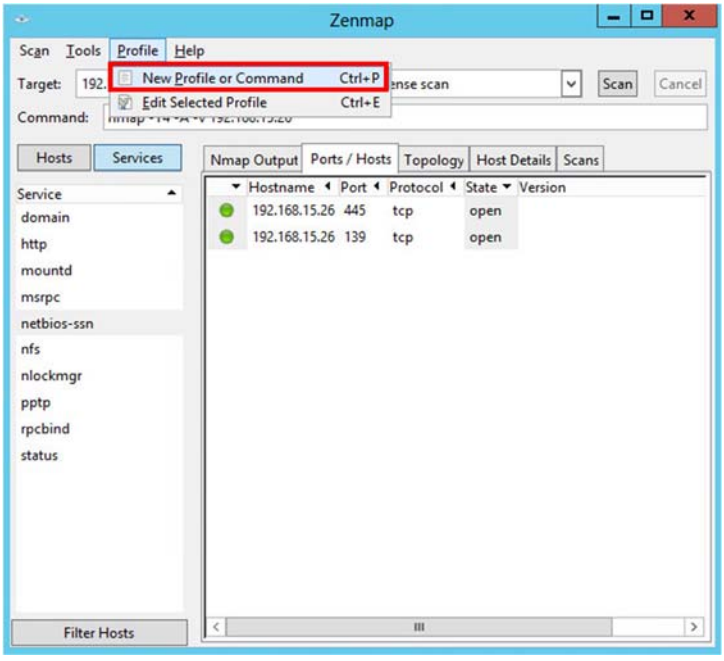


Zenmap window with netbios-ssn option for Intense Scan

Task 2

Xmas Scan

17. Xmas scan sends a TCP frame to a remote device with URG, ACK, RST, SYN, and FIN flags set. FIN scans only with OS TCP/IP developed according to RFC 793. The current version of Microsoft Windows is not supported.
18. Now, to perform a Xmas Scan, you need to create a new profile. Click Profile → New Profile or Command Ctrl+P

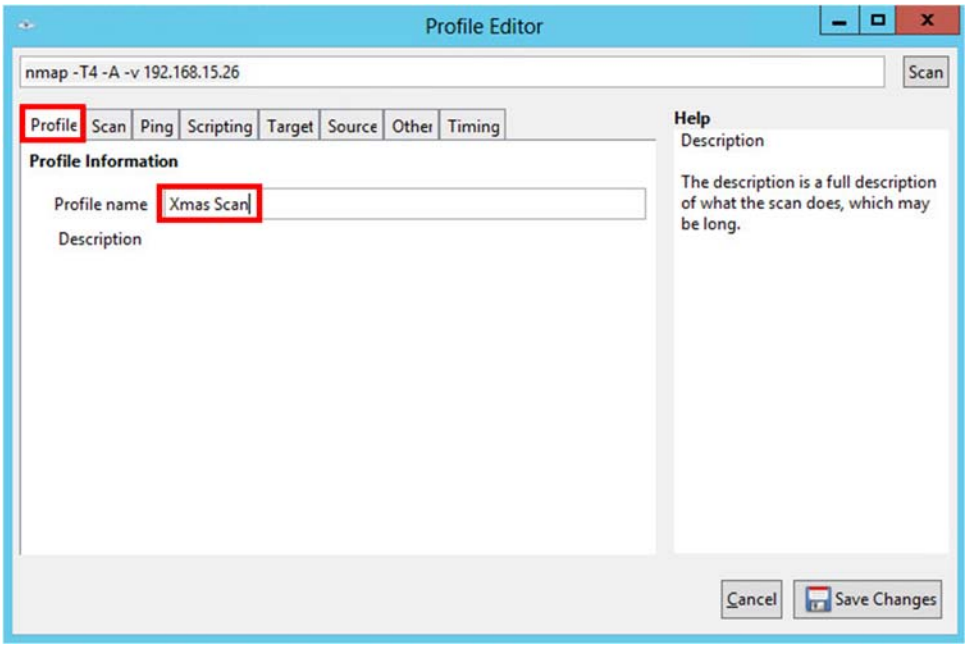


Zenmap window with New Profile or Command menu option

Xmas scan (-sX) sets the FIN, PSH, and URG flags, lighting the packet up like a Christmas tree.

19. On the Profile tab, enter Xmas Scan in the Profile name text field.

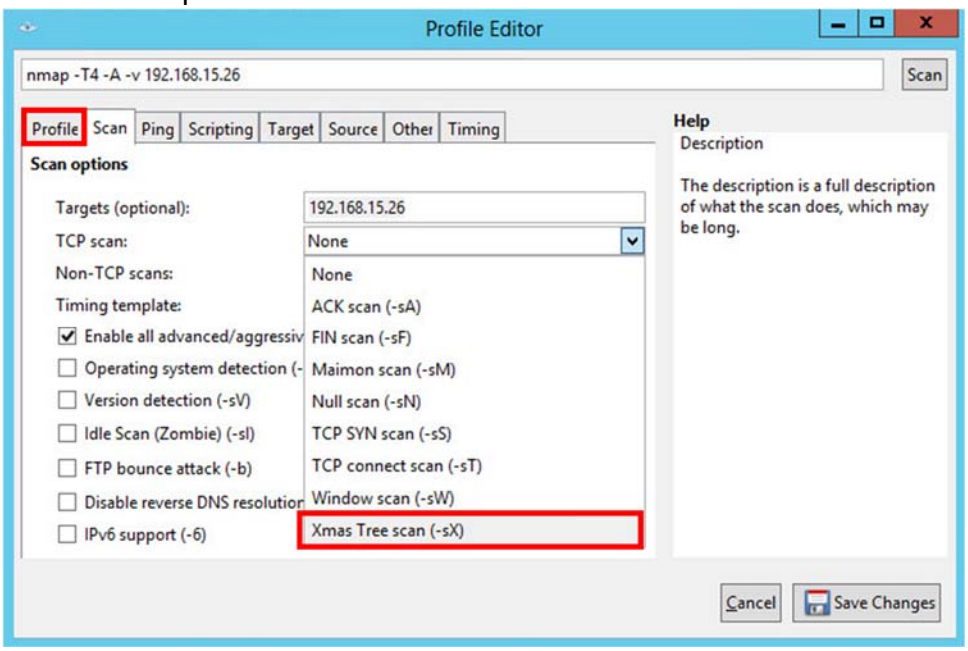
UDP scan is activated with the -sU option. It can be combined with a TCP scan type such as SYN scan (-sS) to check both protocols during the same



Zenmap Profile Editor window with the Profile tab

Nmap detects rate limiting and slows down accordingly to avoid flooding the network with useless packets that the target machine drops.

20. Click the Scan tab, and select Xmas Tree scan (-sX) from the TCP scans: drop-down list.



Zenmap Profile Editor window with the Scan tab

21. Select *None* in the Non-TCP scans: drop-down list and Aggressive (-T4) in the Timing template: list and click **Save Changes**.

Nmap detects rate limiting and slows down accordingly to avoid flooding the network with useless packets that the target machine drops.

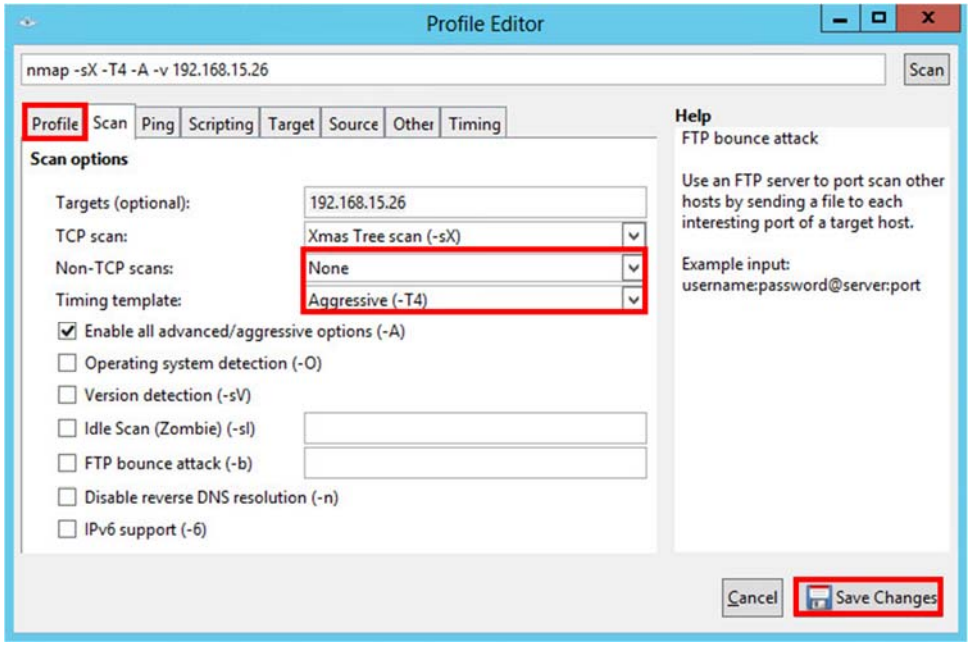


Figure: 2.17- Zenmap Profile Editor window with the Profile tab

When scanning systems, compliant with this RFC text, any packet not containing SYN, RST, or ACK bits results in a returned RST, if the port is closed, and no response at all, if the port is open.

22. Enter the IP address in the **Target:** field, select the Xmas scan option from the **Profile:** hold and click *Scan*.

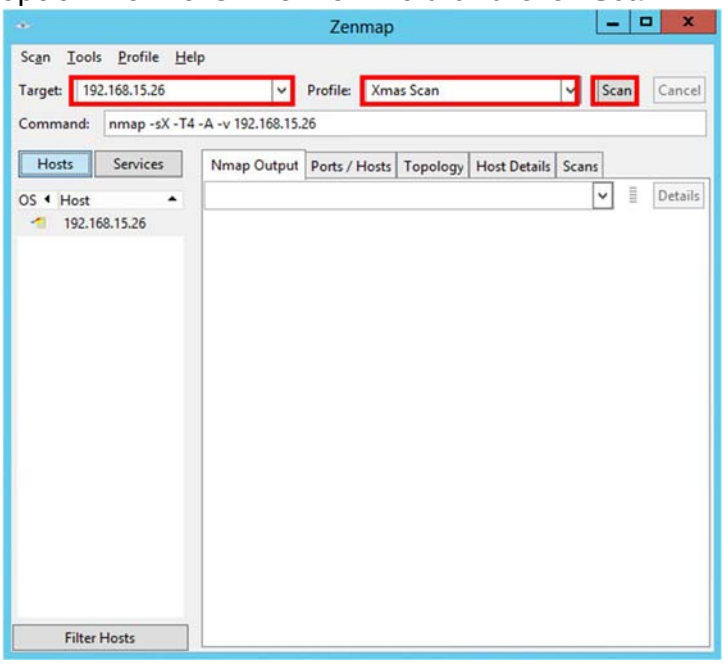


Figure: 2.18- Zenmap Profile Editor window with Profile and Profile Entered

23. Nmap scans the target IP address provided and displays results on the Nmap Output tab.

The option, -sA (TCP ACK scan) is used to map out firewall rulesets, determining whether they are stateful or not and which ports are filtered.

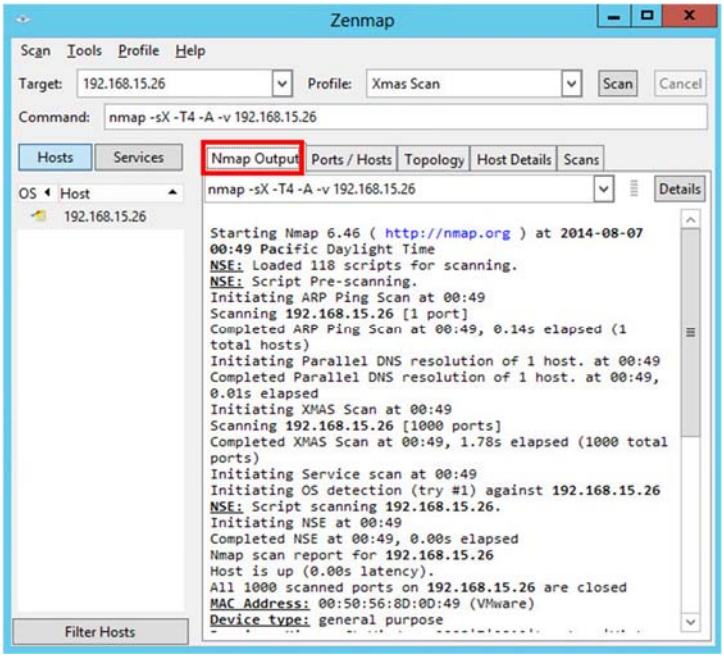


Figure: 2.19- Zenmap main Windows with Output Tab

24. Click the Services tab located on the right side of the pane. It displays all the services of that host.

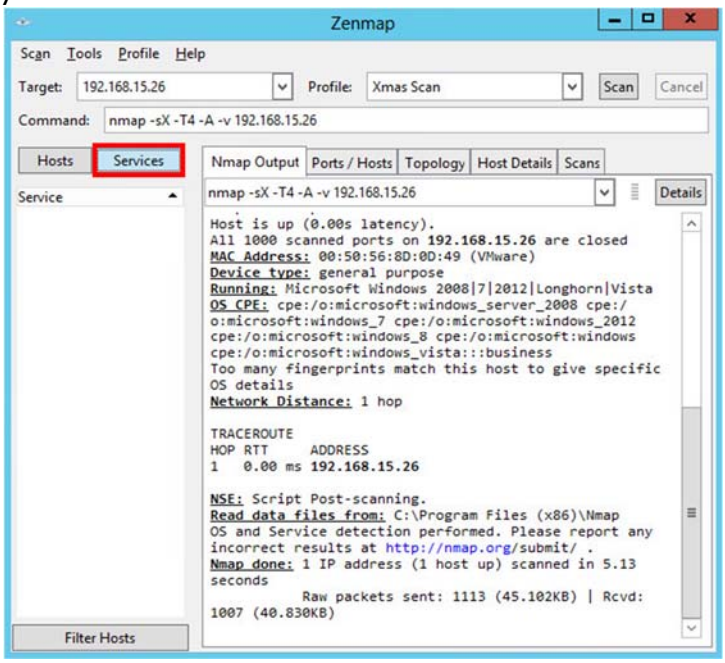


Figure: 2.20- Zenmap main Windows with Services Tab

25. Null scan works only if the operating system's TCP/IP implementation is developed according to RFC 793. In a null scan, attackers send a TCP frame to a remote host with NO Flags.

Task 3

Null Scan

26. Click the Services tab located on the right side of the pane. It displays all the services of that host.

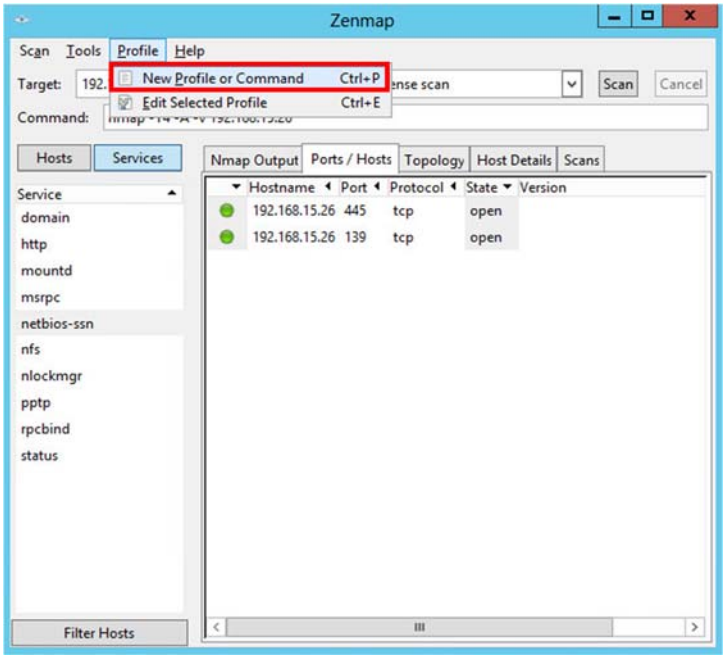


Figure: 2.21- Zenmap window with New Profile or Command menu option

The option Null Scan (-sN) does not set any bits (TCP flag header is 0).

27. On the Profile tab, input a profile name *Null Scan* in the **Profile name** text field.

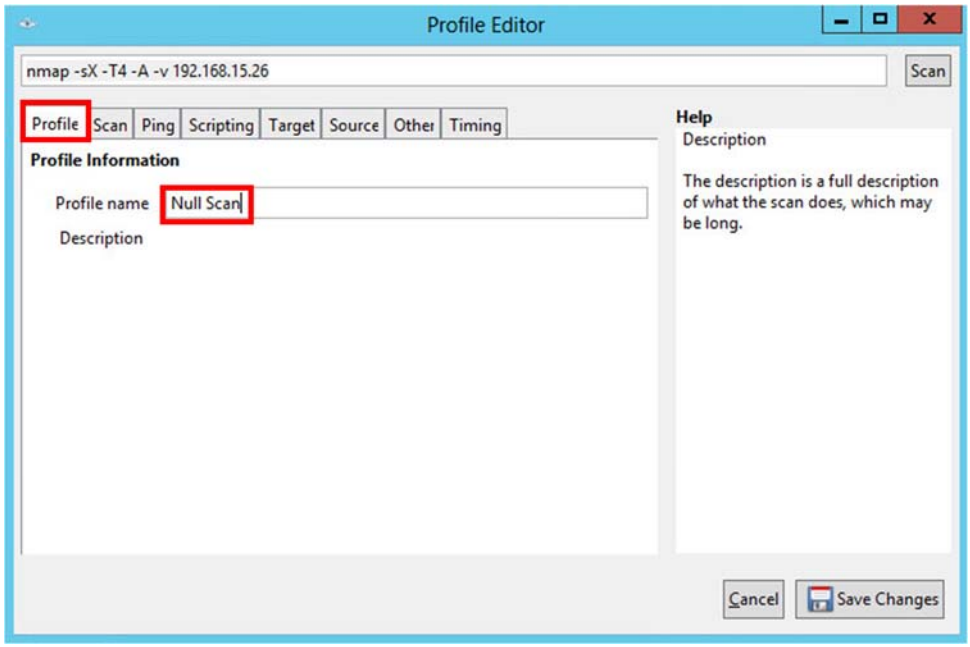


Figure: 2.22- Zenmap Profile Editor with the Profile tab

The option, -sZ (SCTP COOKIE ECHO scan) is an advance SCTP COOKIE ECHO scan. It takes advantage of the fact that SCTP implementations should silently drop packets containing COOKIE ECHO chunks on open ports but send an ABORT if the port is closed.

28. Click the Scan tab in the Profile Editor window. Now select the Null Scan (-sN) option from the TCP scan: drop-down list.

The option, -sI <zombie host>[:<probeport>] (idle scan) is an advanced scan method that allows for a truly blind TCP port scan of the target (meaning no packets are sent to the target from your real IP address). Instead, a unique side-channel attack exploits predictable IP fragmentation ID sequence generation on the zombie host to glean information about the open ports on the target.

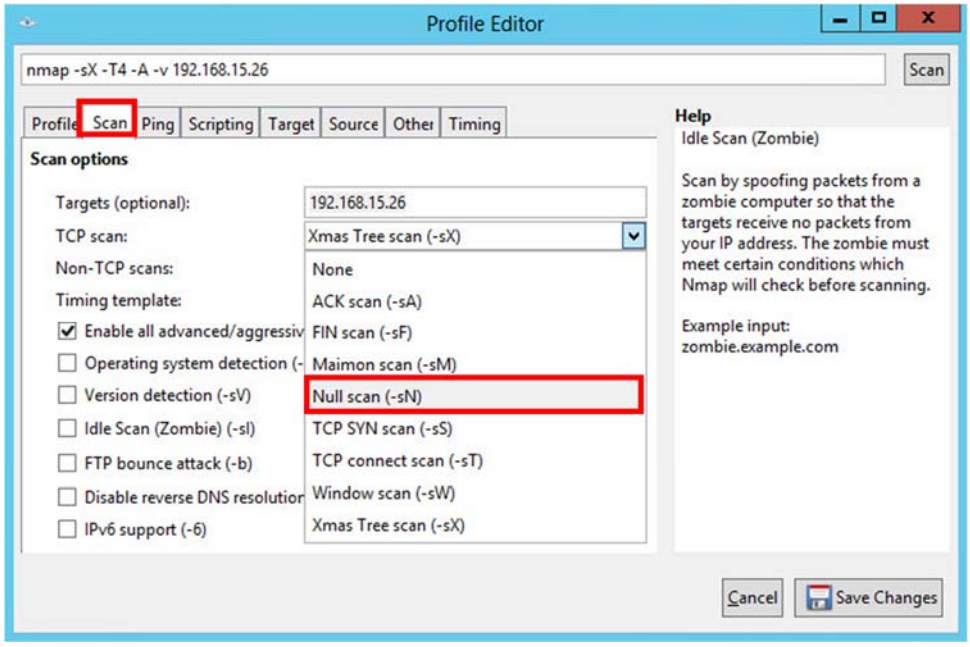


Figure: 2.23- Zenmap Profile Editor with the Scan tab

29. Select None from the **Non-TCP scans:** drop-down field and select Aggressive (-T4) from the **Timing template:** drop-down field.
30. Click Save Changes to save the newly created profile.

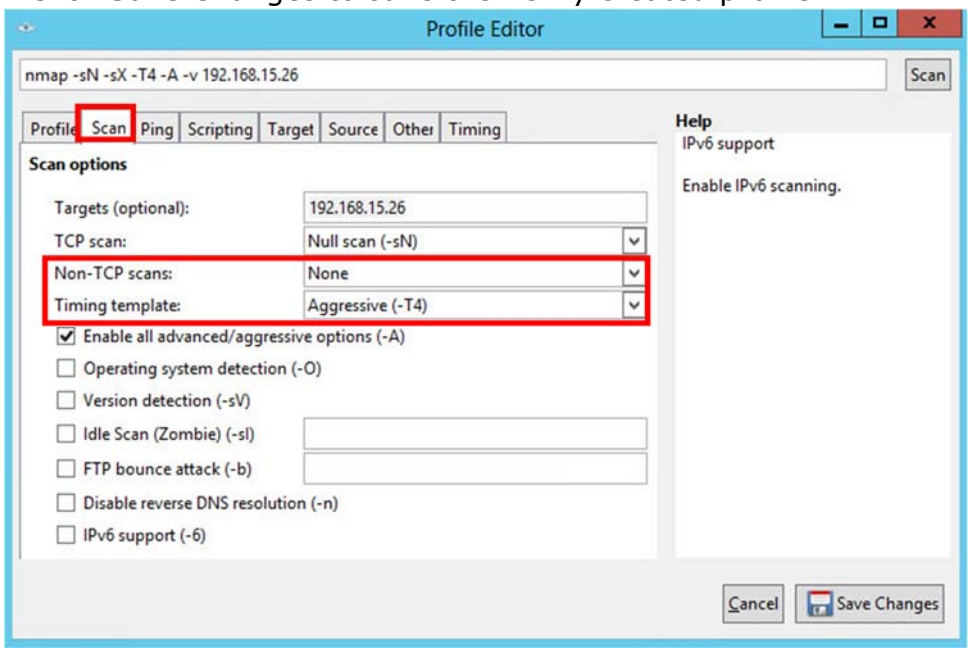


Figure: 2.24- Zenmap Profile Editor with the Scan tab

The option, -b < FTP relay host> (FTP bounce scan) allows a user to connect to one FTP server, and then ask that files be sent to a third-party server. Such a feature is ripe for abuse on many levels, so most servers have ceased supporting it.

31. In the main window of Zenmap, enter the target IP address to scan, select the Null Scan profile from the Profile drop-down list, and then click **Scan**.

The option, -r (Don't randomize ports): By default, Nmap randomizes the scanned port order (except that certain commonly accessible ports are moved near the beginning for efficiency reasons). This randomization is normally desirable, but you can specify -r for sequential (sorted from lowest to highest) port scanning instead.

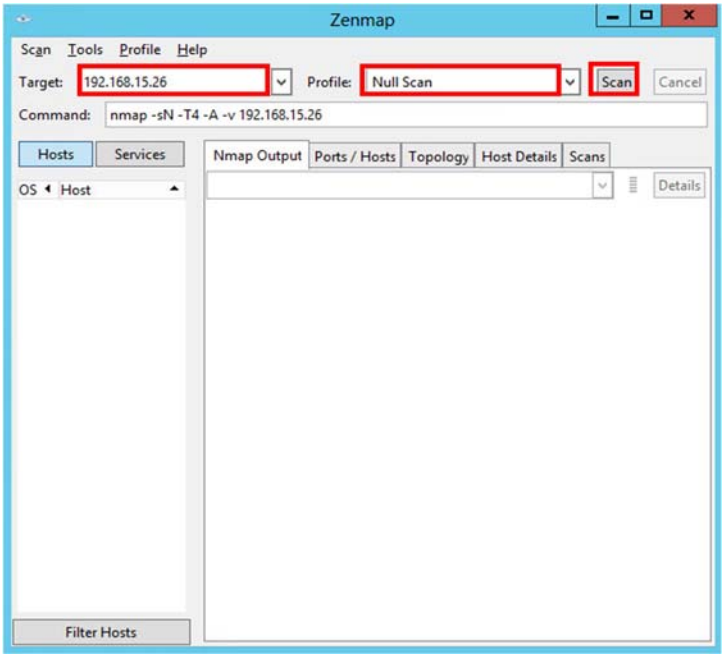


Figure: 2.25- Zenmap main window with Target and Profile entered

32. Nmap scans the target IP address provided and displays results in Nmap Output tab.

In Nmap, option --version-all (Try every single probe) is an alias for - version-intensity 9, ensuring that every single probe is attempted against each port.

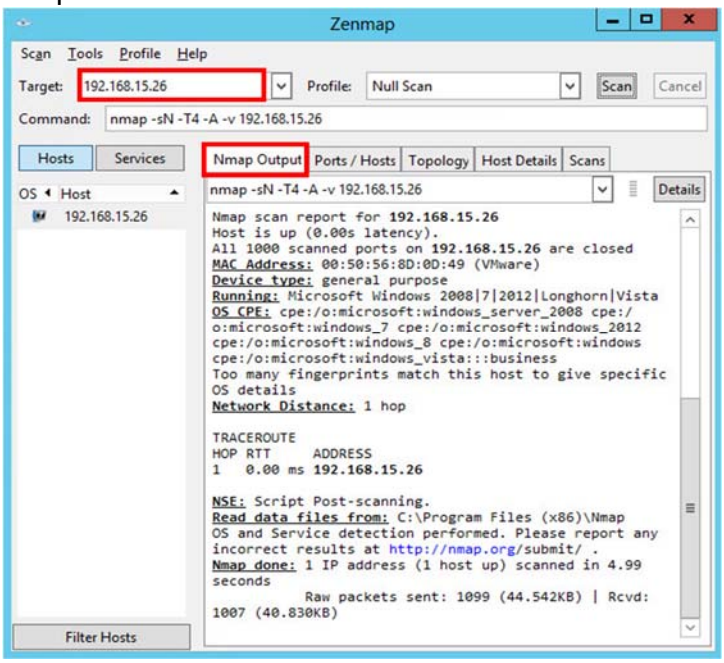


Figure: 2.26- Zenmap main window with the Nmap Output tab

33. Click the Host Details tab to view the details of hosts, such as Host Status, Addresses, Open Ports, and Closed Ports.

The option,--top- ports <n> scans the <n> highest-ratio ports found in the nmap-services file. <n> must be 1 or greater.

The option -sR (RPC scan), method works in conjunction with the various port scan methods of Nmap. It takes all the TCP/UDP ports found open and floods them with SunRPC program NULL commands in an attempt to determine whether they are RPC ports, and if so, what program and version number they serve up.

Task 4

Ack Flag Scan

The option --version-trace (Trace version scan activity) causes Nmap to print out extensive debugging info about what version scanning is doing.
It is a subset of what you get with --packet-trace,

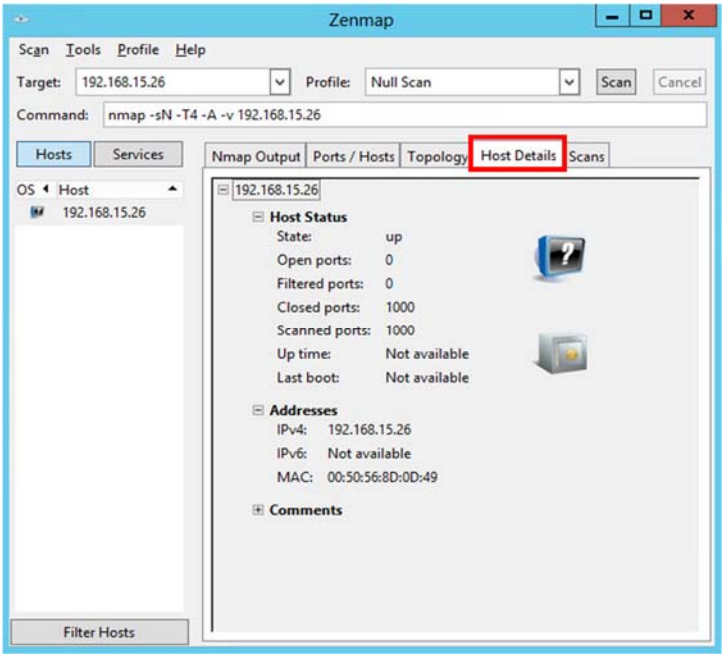


Figure: 2.27- Zenmap main window with the Host Details tab

Lab Analysis

Document all the IP addresses, open and closed ports, services, and protocols you discovered during the lab.

Tool / Utility	Information Collected / Objectives Achieved

Nmap	Types of Scan used:
	• Intense scan
	• Xmas scan
	• Null scan
	• ACK Flag scan
	
	Intense Scan - Nmap Output
	• ARP Ping Scan - 1 host
	• Parallel DNS resolution of 1 host
	• SYN Stealth Scan
	  ◦ Discovered open port on 192.168.15.26
	  ▪ 135/tcp, 139/tcp, 445/tcp, ...
	• MAC Address
	• Operating System Details
	• Uptime Guess
	• Network Distance
	• TCP Sequence Prediction
	• IP ID Sequence Generation
	• Service Info

Quiz

1. Analyze and evaluate the results by scanning a target network using;
   a. Stealth Scan (Half-open Scan)
   b. nmap -P
2. Perform Inverse TCP Flag Scanning and analyze hosts and services for a target machine in the network.

Lab 3

Scanning a Target Using hping3 Utility

Hping3 Overview

Hping is a command-line oriented TCP/IP packet crafter. HPING can be used to create IP packets containing TCP, UDP or ICMP payloads. All header fields can be modified and controlled using the command line. A good understanding of IP and TCP/UDP is mandatory to use and understand the utility.

Hping3's implementation makes the actual construction and transmission of a crafted packet transparent to the user. The tool easily assembles and sends custom ICMP/UDP/TCP packets and displays target replies in the same way ping does with ICMP replies. It handles fragmentation and arbitrary packet body and size, and it can be used to transfer files under the above supported protocols. Hping3 is command-line oriented, and employs a large number of extensions.

ICON KEY


Important Information

Quiz


CPTE Labs


Course Review

Please use all the examples in this lab environment with care. Some examples may actually slow down or crash firewalls or end systems.

Tools demonstrated in this lab are available in

Kali Linux Virtual Machine

Lab Scenario

All known NMAP scanning techniques can be easily reproduced (except a CONNECT scan), but a finer (don't get me wrong, not a bad word about NMAP!!!) control on the packets can be obtained.

The objective of this lab is to help students learn and understand how to build TCP/IP packet using Hping3 to audit a network, a firewall and more.

In this lab, you will need to:

- Scan TCP and UDP ports
- Build a TCP/IP packet with all options
- Modify header files of a TCP/IP packet
- Analyze host details and their topology
- Determine the types of packet filters
- Packet Crafting and Firewall mapping
- Idle Scanning

Lab Resources

To run this lab, you will need the following:

- Hping3 (integrated within Kali)
- A **Kali Linux VM** as a host machine
- A Firefox web browser with Internet access
- Root privileges to run the Hping3 tool
- Target is the Windows 2008 VM

Lab Duration

Time: 40 Minutes

Hping3 Syntax

```
usage: hping3 host [options]
-h --help show this help
-v --version show version
-c --count packet count
-i --interval wait (uX for X microseconds, for example -i u1000)
--fast alias for -i u10000 (10 packets for second)
--faster alias for -i u1000 (100 packets for second)
--flood sent packets as fast as possible. Don't show replies.
-n --numeric numeric output
-q --quiet quiet
-I --interface interface name (otherwise default routing interface)
-V --verbose verbose mode
-D --debug debugging info
-z --bind bind ctrl+z to ttl (default to dst port)
-Z --unbind unbind ctrl+z
--beep beep for every matching packet received
```

Lab Tasks

Note: To stop scans press Ctrl C

1. Log into Kali Linux using username=root/password=toor

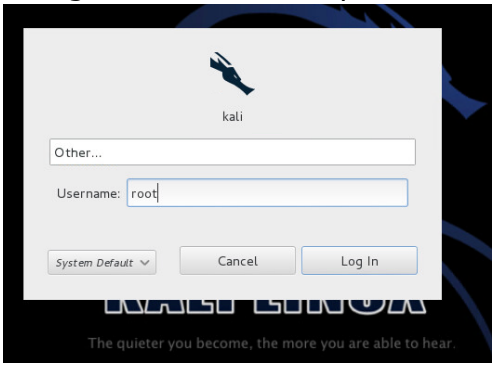


Figure: 3.1- Kali Linux – Login Screen

Task 1

Start Hping3

2. Start Hping3 using: Applications → Kali Linux → Information Gathering → Live Host Identification → hping3

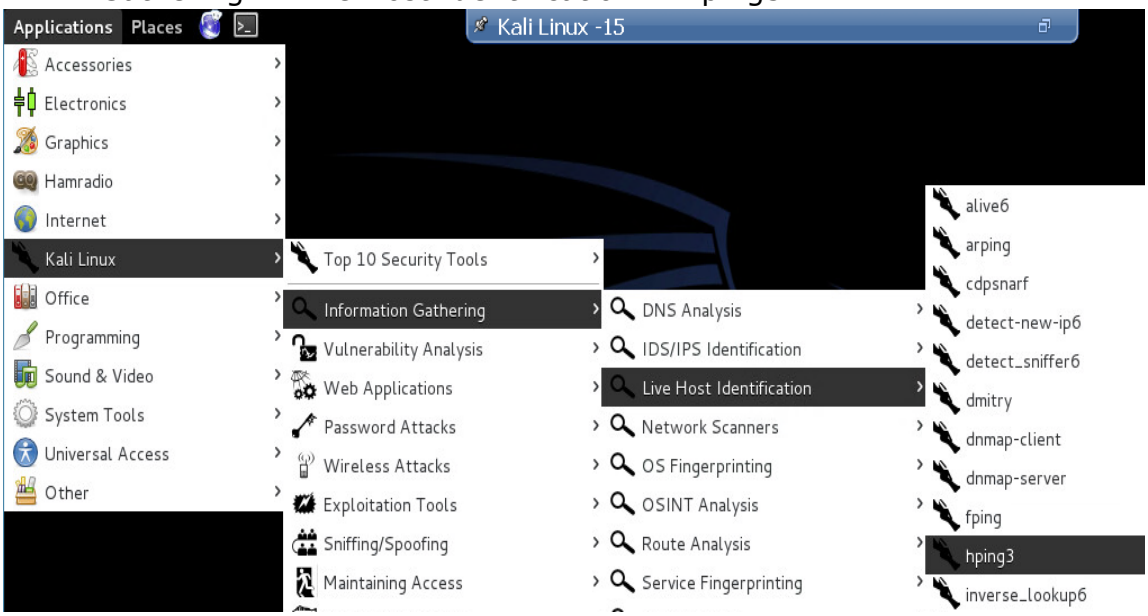


Figure: 3.2- Hping3 in Kali Linux

3. The Hping3 help window will appear:

```
  -R  --rst          set RST flag
  -P  --push         set PUSH flag
  -A  --ack          set ACK flag
  -U  --urg          set URG flag
  -X  --xmas         set X unused flag (0x40)
  -Y  --ymas         set Y unused flag (0x80)
  --tcpexitcode      use last tcp->th_flags as exit code
  --tcp-mss          enable the TCP MSS option with the given value
  --tcp-timestamp    enable the TCP timestamp option to guess the HZ/uptime
Common
  -d  --data         data size                  (default is 0)
  -E  --file         data from file
  -e  --sign         add 'signature'
  -j  --dump         dump packets in hex
  -J  --print        dump printable characters
  -B  --safe         enable 'safe' protocol
  -u  --end          tell you when --file reached EOF and prevent rewind
  -T  --traceroute   traceroute mode            (implies --bind and --ttl 1)
  --tr-stop          Exit when receive the first not ICMP in traceroute mode
  --tr-keep-ttl      Keep the source TTL fixed, useful to monitor just one hop
  --tr-no-rtt        Don't calculate/show RTT information in traceroute mode
ARS packet description (new, unstable)
  --apd-send         Send the packet described with APD (see docs/APD.txt)
root@kali:~#
```

Figure: 3.3- Hping3 print usage

Task 2
Hping3 as a Port Scanner

4. Hping3 as a port scanner.
   Crafting TCP packets is the default behavior of HPING.
   By specifying the TCP flags, a destination port and a target IP address, one can easily construct TCP packets.

```
  -F  --fin             set FIN flag
  -S  --syn             set SYN flag
  -R  --rst             set RST flag
  -P  --push            set PUSH flag
  -A  --ack             set ACK flag
  -U  --urg             set URG flag
  -X  --xmas            set X unused flag (0x40)
  -Y  --ymas            set Y unused flag (0x80)
```

An **open port** is indicated by an **SA return packet**, **closed ports** are indicated by **RA packets**.

Remember the TCP 3-way handshake!

This is similar to a very known way of scanning, called a SYN scan or Stealth scan.

```
root@kali:~# hping3 -I eth0 192.168.15.26 -p 139
HPING 192.168.15.26 (eth0 192.168.15.26): NO FLAGS are set, 40 headers + 0 data
bytes
len=46 ip=192.168.15.26 ttl=128 DF id=6067 sport=139 flags=RA seq=0 win=0 rtt=1.
2 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6068 sport=139 flags=RA seq=1 win=0 rtt=0.
5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6069 sport=139 flags=RA seq=2 win=0 rtt=0.
5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6070 sport=139 flags=RA seq=3 win=0 rtt=0.
6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6071 sport=139 flags=RA seq=4 win=0 rtt=0.
6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6072 sport=139 flags=RA seq=5 win=0 rtt=0.
4 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6073 sport=139 flags=RA seq=6 win=0 rtt=0.
5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6074 sport=139 flags=RA seq=7 win=0 rtt=0.
5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6075 sport=139 flags=RA seq=8 win=0 rtt=0.
4 ms
```

hping3 -I eth0 <IP address of your target> -p 139

5. A nice built-in feature is the ++, which will increase the destination port in the packets by one.
   You can also press 'ctrl+z', instead of using ++, to increase the port number during the scan.

All known NMAP scanning techniques can be easily reproduced (except a CONNECT scan), but a finer control on the packets can be obtained. Take a look at the following options that can be set.

```
root@kali:~# hping3 -I eth0 192.168.15.26 -p ++130
HPING 192.168.15.26 (eth0 192.168.15.26): NO FLAGS are set, 40 headers + 0 dat
a bytes
len=46 ip=192.168.15.26 ttl=128 DF id=6107 sport=130 flags=RA seq=0 win=0 rtt=
1.0 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6108 sport=131 flags=RA seq=1 win=0 rtt=
0.4 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6109 sport=132 flags=RA seq=2 win=0 rtt=
0.6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6110 sport=133 flags=RA seq=3 win=0 rtt=
0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6111 sport=134 flags=RA seq=4 win=0 rtt=
0.4 ms
```

hping3 -I eth0 <IP address of your target> -p ++130

6. You can easily combine flags and other parameters as follows:

```
root@kali:~# hping3 -I eth0 -M 3000 -SA 192.168.15.26 -p 139
HPING 192.168.15.26 (eth0 192.168.15.26): SA set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=6116 sport=139 flags=R seq=0 win=0 rtt=1
.6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6117 sport=139 flags=R seq=1 win=0 rtt=0
.6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6118 sport=139 flags=R seq=2 win=0 rtt=0
.4 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6119 sport=139 flags=R seq=3 win=0 rtt=0
.7 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6120 sport=139 flags=R seq=4 win=0 rtt=0
.7 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6121 sport=139 flags=R seq=5 win=0 rtt=0
.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6122 sport=139 flags=R seq=6 win=0 rtt=0
.4 ms
```

hping3 -I eth0 -M 3000 -SA <IP address of your target> -p 139

Task 3

Idle scanning

7. **Idle scanning** is a technique to port scan a remote system fully anonymously.

The idle scan is a TCP port scan method that you can use to send a spoofed source address to a computer to find out what services are available and offers complete blind scanning of a remote host. This is accomplished by impersonating another computer. No packet is sent from your own IP address; instead, another host is used, often called a "zombie," to scan the remote host and determine the open ports. This is done by expecting the sequence numbers of the zombie host and if the remote host checks the IP of the scanning party, the IP of the zombie machine will show up.

To IDLE scan a system, some requirements must be met.
The Attacker in our example is the machine running two sessions of hping.

Port is considered open if an application is listening on the port. One way to determine whether a port is open is to send a "SYN" (session establishment) packet to the port. The target machine will send back a "SYN | ACK" and a Reset "RST" packet if the port is closed.

```
root@kali:~# hping3 -I eth0 -SA 192.168.15.26
HPING 192.168.15.26 (eth0 192.168.15.26): SA set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=6127 sport=0 flags=R seq=0 win=0 rtt=1.1
 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6128 sport=0 flags=R seq=1 win=0 rtt=0.5
 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6129 sport=0 flags=R seq=2 win=0 rtt=0.4
 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6130 sport=0 flags=R seq=3 win=0 rtt=0.4
 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6131 sport=0 flags=R seq=4 win=0 rtt=0.4
 ms
len=46 ip=192.168.15.26 ttl=128 DF id=6132 sport=0 flags=R seq=5 win=0 rtt=0.5
 ms
```

hping3 -I eth0 -SA <IP address of your target>

Most network servers listen on TCP ports, such as web servers on port 80 and mail servers on port 25.

```
root@kali:~# hping3 -I eth0 -SA -r 192.168.15.26
HPING 192.168.15.26 (eth0 192.168.15.26): SA set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=9839 sport=0 flags=R seq=0 win=0 rtt=0.9 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=0 flags=R seq=1 win=0 rtt=0.7 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=0 flags=R seq=2 win=0 rtt=0.4 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=0 flags=R seq=3 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=0 flags=R seq=4 win=0 rtt=0.4 ms
```

hping3 -I eth0 -SA -r <IP address of your target>

If we run a continuous probe against the silent host and the attacker scans the server, spoofed with the IP address of the silent host, the following will happen.
If we scan an open port with a SYN packet, the server will respond with a SYN/ACK packet (to the Silent host). The Silent host will react by sending a RESET packet to the server, and will, of course, increase the IP_ID by one (because he sends a packet). The next probe we send will have the next IP_ID in return. (2 units higher than the previous probe).
If we send a SYN packet to a closed port, a RST is sent to the Silent host, which does not imply sending any packet from the silent host.

(IP_ID is not increased), since it will be discarded.
It is indeed a good idea to run a tcpdump to see what actually happens.

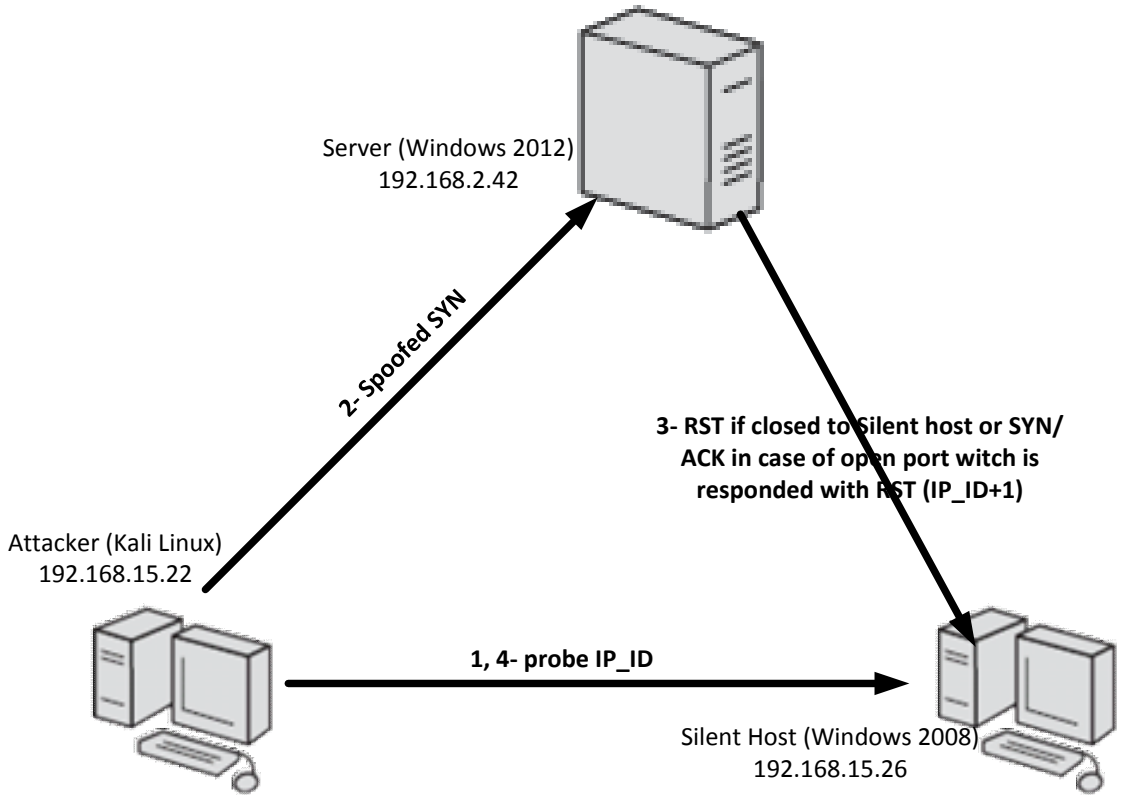


Figure: 3.9- Idle Scanning Architecture

Use 2 windows or tabs(one for each session)

Session 1, a spoofed scan of the server by the attacker:
Type: **hping3 -I eth0 -a <IP address of your 2008 Server> -S <IP address of your 2012 Server> -p ++20**

```
root@kali:~# hping3 -I eth0 -a 192.168.15.26 -S 192.168.2.42 -p ++20
HPING 192.168.2.42 (eth0 192.168.2.42): S set, 40 headers + 0 data bytes
```

Figure: 3.10- Idle Scanning - Session 1: Spoofed Scan

Session 2, a continuous probe from the attacker to the Silent host to monitor the IP IDENTIFICATION field:

hping3 -I eth0 -r -S < **IP address of your 2008 Server** > -p 2000

```
root@kali:~# hping3 -I eth0 -r -S 192.168.15.26 -p 2000
HPING 192.168.15.26 (eth0 192.168.15.26): S set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=9863 sport=2000 flags=RA seq=0 win=0 rtt=0.4
ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=1 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=2 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=3 win=0 rtt=0.4 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=4 win=0 rtt=0.3 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=5 win=0 rtt=0.6 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=6 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=7 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=8 win=0 rtt=0.5 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+2 sport=2000 flags=RA seq=9 win=0 rtt=0.3 ms
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=10 win=0 rtt=0.3 m
s
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=11 win=0 rtt=0.4 m
s
len=46 ip=192.168.15.26 ttl=128 DF id=+1 sport=2000 flags=RA seq=12 win=0 rtt=0.3 m
```

Figure: 3.11- Idle Scanning - Session 2: Continuous Probe

Server 2008 is Silent Host(zombie)
Server 2012 is main Victim(target)

Task 5

Half Open Scan
UDP Scan
FIN Scan

8. Other Hping3 Scan Techniques
Windows 2008 is the target. Port is 80.

To run Half Open Scan Type: hping3 -I eth0 -S <target> -p <port> -c 1

```
root@kali:~# hping3 -I eth0 -S 192.168.15.26 -p 80 -c 1
HPING 192.168.15.26 (eth0 192.168.15.26): S set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=9897 sport=80 flags=SA seq=0 win=8192 rtt=0.7
 ms

--- 192.168.15.26 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 0.7/0.7/0.7 ms
```

Figure: 3.12- Half Open Scan

To run UDP Scan Type: hping3 -I eth0 -2 <target> -p 139 -c 1

```
root@kali:~# hping3 -I eth0 -2 192.168.15.26 -p 139 -c 1
HPING 192.168.15.26 (eth0 192.168.15.26): udp mode set, 28 headers + 0 data bytes
ICMP Port Unreachable from ip=192.168.15.26 name=UNKNOWN
status=0 port=1098 seq=0

--- 192.168.15.26 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 10.9/10.9/10.9 ms
```

Figure: 3.13- UDP Scan

To run FIN Scan Type: hping3 -I eth0 -F <target> -p 6000 -c 1

```
root@kali:~# hping3 -I eth0 -F 192.168.15.26 -p 6000 -c 1
HPING 192.168.15.26 (eth0 192.168.15.26): F set, 40 headers + 0 data bytes
len=46 ip=192.168.15.26 ttl=128 DF id=10022 sport=6000 flags=RA seq=0 win=0 rtt=1.2
 ms

--- 192.168.15.26 hping statistic ---
1 packets transmitted, 1 packets received, 0% packet loss
round-trip min/avg/max = 1.2/1.2/1.2 ms
```

Figure: 3.14- FIN Scan

Lab Analysis

Document all the IP addresses, open and closed ports, services, and protocols you discovered during the lab.

Tool / Utility	Information Collected / Objectives Achieved

Hping3	Types of Scan used:
	• Intense scan
	• Xmas scan
	• Null scan
	• ACK Flag scan
	• FIN Scan
	• UDP Scan
	
	Hping3 Output
	• ARP Ping Scan - 1 host
	• Parallel DNS resolution of 1 host
	• SYN Stealth Scan
	  • Discovered open port on 192.168.15.26
	  ▪ 135/tcp, 139/tcp, 445/tcp, ...
	• MAC Address
	• Operating System Details
	• Uptime Guess
	• Network Distance
	• TCP Sequence Prediction
	• IP ID Sequence Generation
	• Service Info

Quiz

1. Analyze and evaluate the results by scanning a target network using hping3:
   a. SYN/ACK scan
   b. NULL
2. Perform Inverse TCP Flag Scanning and analyze hosts and services for a target machine on the network.
3. How do you use hping3 as DOS tools? Give the command syntax. Try this command on the Metasploitable Virtual Machine.

Lab 4

Scanning a Target Using PBNJ Utility

PBNJ Overview

ScanPBNJ performs an Nmap scan and then stores the results in a database. The ScanPBNJ stores information about the machine that has been scanned. ScanPBNJ stores the IP Address, Operating System, Hostname and a localhost bit. The localhost bit is simply a single bit which is 1 when the target machine is localhost, otherwise it is 0. It also stores two timestamps for the machine table. The first is a human readable version and the second is the Unix time. Both of these timestamps correspond to the first time that the machine was scanned.

ICON KEY


Important Information

Quiz


CPTE Labs


Course Review

Lab Scenario

As described by its authors, PBNJ is a suite of tools to monitor changes on a network over time. PBNJ monitors changes by checking for changes on the target machines, which includes the details about the services running on them as well as the service state. PBNJ parses the data from Nmap scans and stores it in a MySQL database.
Logging Nmap results into a MySQL database has several advantages, especially when the number of hosts scanned is large.

Lab Resources

To run this lab, you will need the following:

- PBNJ
- If you decide to download the latest version, the screenshots shown in the lab might differ
- A **Kali Linux VM** as a host machine
- A Firefox web browser with Internet access
- Root privileges to run the PBNJ tools

Lab Duration

Time: 20 Minutes


Tools demonstrated in this lab are available in

Kali Linux Virtual Machine

Lab Tasks

1. Log into Kali Linux using username=root/password=toor

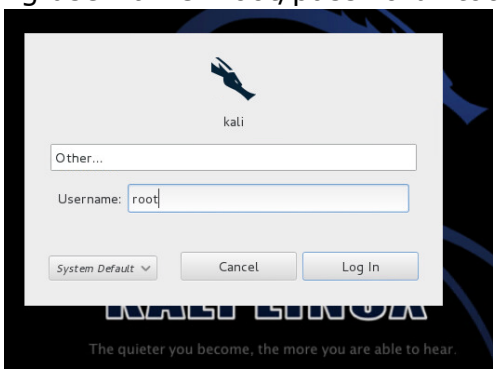


Figure: 4.1- Kali Linux – Login Screen

Task 1

Start MySQL Database

2. Quickly set up the MySQL database and get started with a logged scan:
   /etc/init.d/mysql start

```
root@kali:~# /etc/init.d/mysql start
[....] Starting MySQL database server: mysqld
[ .k
[info] Checking for tables which need an upgrade, are corrupt or were
not closed cleanly..
```

Figure: 4.2- Start MySQL Database

3. Check that port TCP/3306 of the MySQL database was open:
   netstat –naptul | grep 3306

```
root@kali:~# netstat -naptul|grep 3306
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      27440/mysqld
```

Figure: 4.3- MySQL Database TCP/3306 port

4. Create PBNJ database to store Nmap Scan Results:
   mysql -u root
   CREATE DATABASE pbnj;
   exit

Task 2

Create PBNJ Nmap Scan Result Database

```
root@kali:~# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 45
Server version: 5.5.35-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> CREATE DATABASE pbnj;
Query OK, 1 row affected (0.00 sec)

mysql> exit
Bye
```

Figure: 4.4- Create MySQL Database called PBNJ

Task 3

Install PBNJ Package in Kali

5. Install PBNJ package in Kali:
   First type in <apt-get remove pbnj> to remove any residual files.
   apt-get install pbnj
   Y (for Yes)

```
root@kali:~# apt-get install pbnj
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  libmozjs22d python-apsw xulrunner-22.0
Use 'apt-get autoremove' to remove them.
The following extra packages will be installed:
  libfile-homedir-perl libfile-which-perl libnmap-parser-perl libtext-csv-perl
  libtext-csv-xs-perl libyaml-perl
Suggested packages:
  libyaml-shell-perl
The following NEW packages will be installed:
  libfile-homedir-perl libfile-which-perl libnmap-parser-perl libtext-csv-perl
  libtext-csv-xs-perl libyaml-perl pbnj
0 upgraded, 7 newly installed, 0 to remove and 230 not upgraded.
Need to get 351 kB of archives.
After this operation. 1,164 kB of additional disk space will be used.
Do you want to continue [Y/n]?
```

Figure: 4.5- Kali Linux – PBNJ Package Installation

Task 4

Configure PBNJ yaml file

6. Configure the PBNJ yaml file with the database details:

mkdir -p /root/.pbnj-2.0
cd /root/.pbnj-2.0
cp /usr/share/doc/pbnj/examples/mysql.yaml config.yaml
nano config.yaml

database: pbnj
user: root
passwd: ""
host: localhost
port: 3306

Ctrl O (Write out) or Ctrl X (save), Yes to save, Enter to accept filename.

Task 5

Start a simple ping sweep

```
root@kali:/# mkdir -p /root/.pbnj-2.0
root@kali:/# cd /root/.pbnj-2.0
root@kali:~/.pbnj-2.0# cp /usr/share/doc/pbnj/examples/mysql.yaml config.yaml
root@kali:~/.pbnj-2.0# nano config.yaml
# YAML:1.0
# Config for connecting to a DBI database
# SQLite, mysql etc
db: mysql
# for SQLite the name of the file. For mysql the name of the database
database: pbnj
# Username for the database. For SQLite no username is needed.
user: root
# Password for the database. For SQLite no password is needed.
passwd: ""
# Password for the database. For SQLite no host is needed.
host: localhost
# Port for the database. For SQLite no port is needed.
port: 3306
```

7. Start with a simple ping sweep:

scanpbnj -a "-sP" <IP address of your targets range>

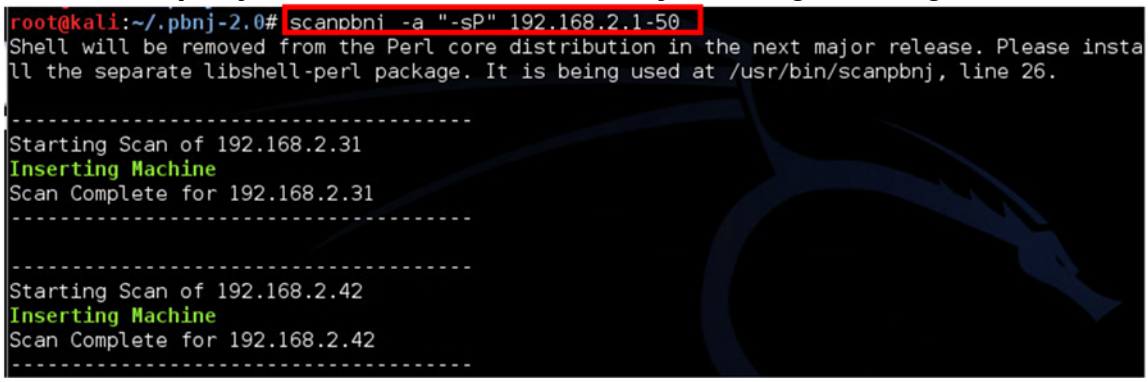


Figure: 4.7- PBNJ ping sweep for 192.168.2.1-50

Adjust your target ip address range to match correct ip addresses for VMs.

8. Query the MySQL database for the found machines:

mysql -u root
use pbnj;
show tables;
select * from services;
select * from machines;

Task 6

Query MySQL PBNJ database for scan results

You discover that the database has two tables: **machines** and **services**. Because you only ran a ping sweep, no services were recorded for any of the machines.

```
root@kali:~/.pbnj-2.0# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 50
Server version: 5.5.35-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use pbnj;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+---------------+
| Tables_in_pbnj |
+---------------+
| machines       |
| services       |
+---------------+
2 rows in set (0.00 sec)

mysql> select * from sevices;
ERROR 1146 (42S02): Table 'pbnj.sevices' doesn't exist
mysql> select * from services;
Empty set (0.00 sec)

mysql> select * from machines;
+-----+--------------+------+--------+------------+----------------+----------------------
--+
| mid | ip           | host | localh | os         | machine_created | created_on
  |
+-----+--------------+------+--------+------------+----------------+----------------------
--+
|   1 | 192.168.2.31 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|   2 | 192.168.2.42 | 0    |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|   3 | 192.168.2.47 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|   4 | 192.168.2.46 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|   5 | 192.168.2.22 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  42 | 192.168.2.16 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  43 | 192.168.2.41 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  44 | 192.168.2.37 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  45 | 192.168.2.50 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  46 | 192.168.2.25 | 0    |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  47 | 192.168.2.8  |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  48 | 192.168.2.10 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  49 | 192.168.2.19 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
|  50 | 192.168.2.32 |      |      0 | unknown os | 1407940691      | Wed Aug 13 10:38:11 201
4 |
+-----+--------------+------+--------+------------+----------------+----------------------
--+
50 rows in set (0.00 sec)

mysql>
```

9. Now, try to sweep TCP port 139:
   (If needed from bash prompt menu; file → new tab)

scanpbnj -a "-p 139" <IP address of your targets range>

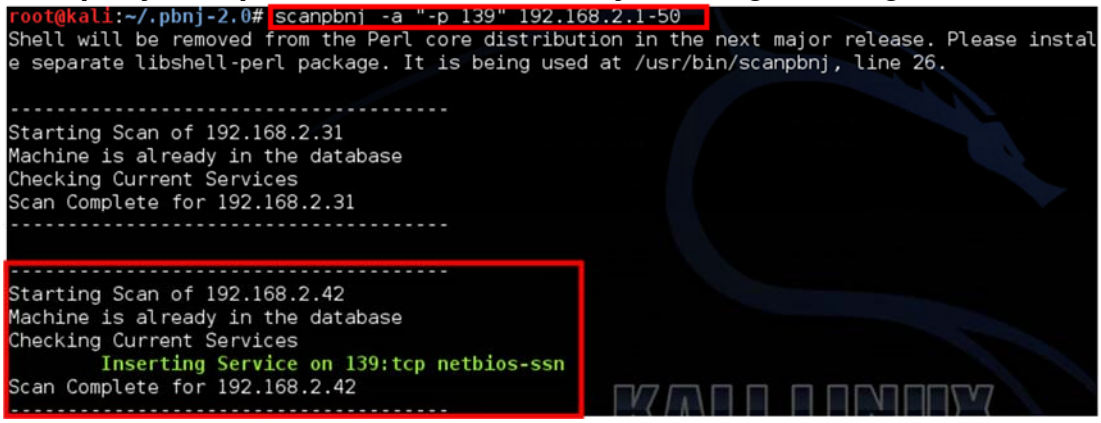


Figure: 4.9- PBNJ Query the PBNJ database

(if needed
mysql –u root
use pbnj;)

10. And, once again, inspect the database:
    select * from services;

Figure: 4.10- PBNJ Query the PBNJ database – Services Tables

As more information is gathered about a machine (such as banners, OS versions, and so forth), it is added to the relevant fields in the database.

Because PBNJ is a wrapper for Nmap, it is not recommended to run large or heavy scans; rather, build the database slowly using shorter, more specific scans.

Lab Analysis

Document all the IP addresses, open and closed ports, services, and protocols you discovered during the lab.

Tool / Utility	Information Collected / Objectives Achieved

PBNJ tools	Types of Scan used:
	• Intense scan
	• Xmas scan
	• Null scan
	• FIN Scan
	• UDP Scan
	
	PBNJ Output
	• ARP Ping Scan - 1 host
	• Parallel DNS resolution of 1 host
	• SYN Stealth Scan
	  ◦ Discovered open port on 192.168.15.26
	  ▪ 135/tcp, 139/tcp, 445/tcp, ...
	• MAC Address
	• Operating System Details
	• Uptime Guess
	• Network Distance
	• TCP Sequence Prediction
	• IP ID Sequence Generation
	• Service Info

Quiz

1. Analyze and evaluate the results by scanning a target network using PBNJ:
   a. SYN/ACK scan
   b. NULL
2. Perform Inverse TCP Flag Scanning and analyze hosts and services for a target machine in the network.
3. Query PBNJ database using all query keywords:

```
a. possiblevuln
b. sshmachines
c. allservices
d. services
e. unknown_version_up
f. unknown_banner_up
g. machines
h. mdump
i. servicesup
j. service_audit
```