Department of Information Technology Engineering

IT321-Ethical Hacking

AY:2023-2024

Lab1 A- Internet Foot-printing

Lab1 B- Network Scanning

Name of Student : Adwait Purao UCID : 2021300101

Class : COMPS B

Objectives:

[1] Introduce the anatomy of ethical hacking

[2]To perform Network Reconnaissance Using Command Line

[3]To perform DNS Interrogation

[4]To perform Web Reconnaissance

Outcomes: After completing the lab, you will be able to:

[1] Identify and demonstrate the ethical hacking phases.

[2] Define footprinting and how it is accomplished.

[3] Identify many resources that can be used to footprint an organization

[4] Search an organization’s public Web pages and identify internal components

[5] Determine the IP address range assigned to a particular organization

[6] Identify host machines that are active within an organization

System Requirements:

[1] Kali Linux- Offensive

[2] Windows 7/ Windows XP- Client or victim

[3] Tools:
ping,fping,nslookup,traceroute,arp,host,dig,TCPDUMP/Wireshark, nmap, etherape,ping etc

Lab1 A- Internet Foot-printing

1A- **Foot-printing**

1A.1 Google Hacking Database

1A.2- Who is Database

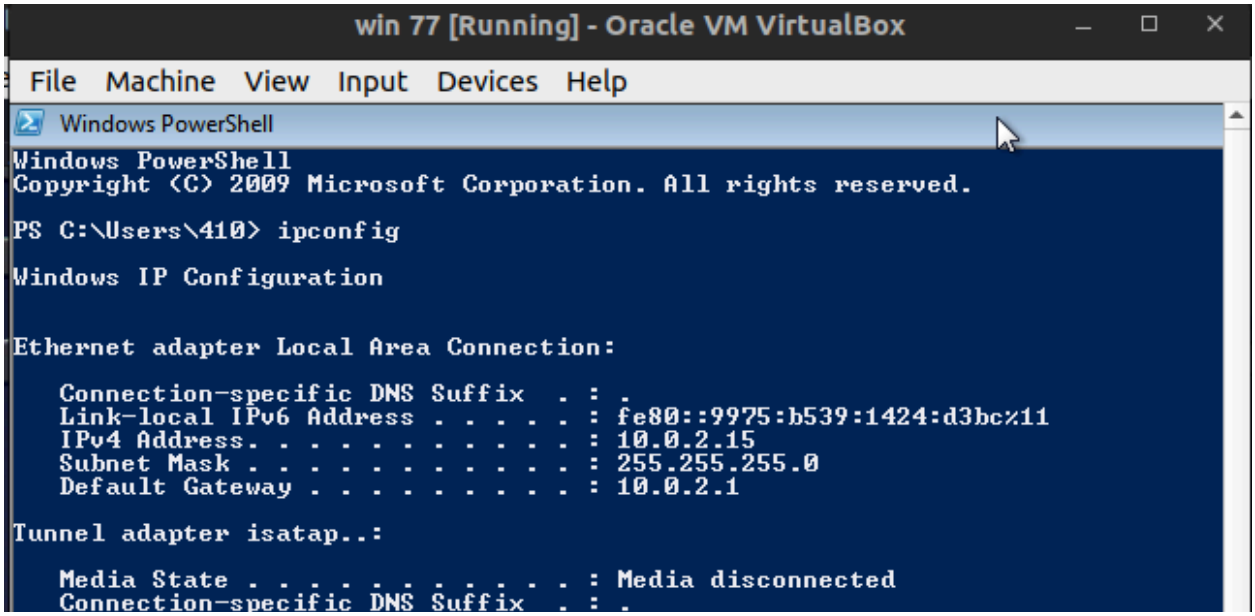
1A.3 Other available resources

1B **Network Footprinting**

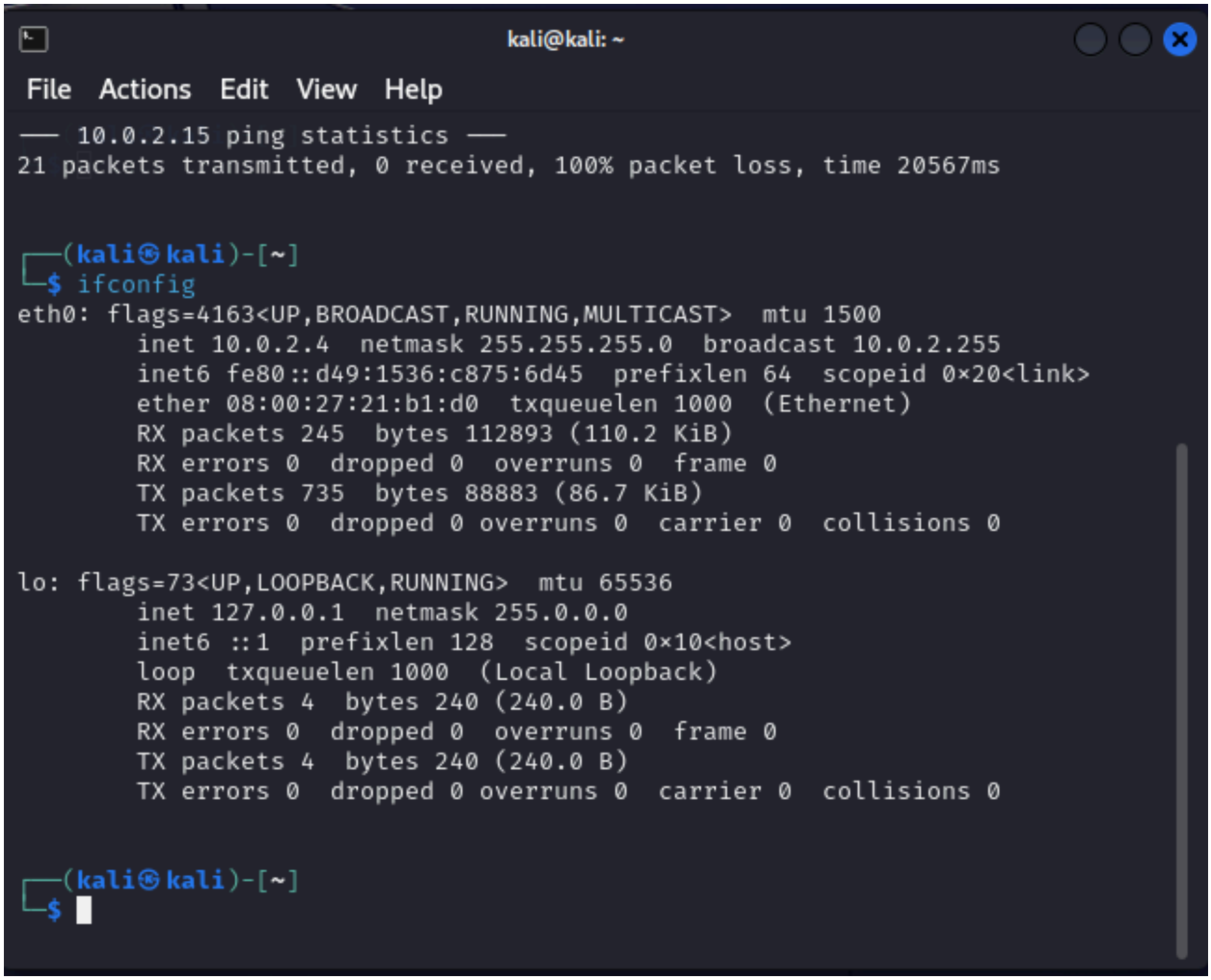
1B.1 Network Reconnaissance using command-line tools

1. **Ifconfig/ipconfig**

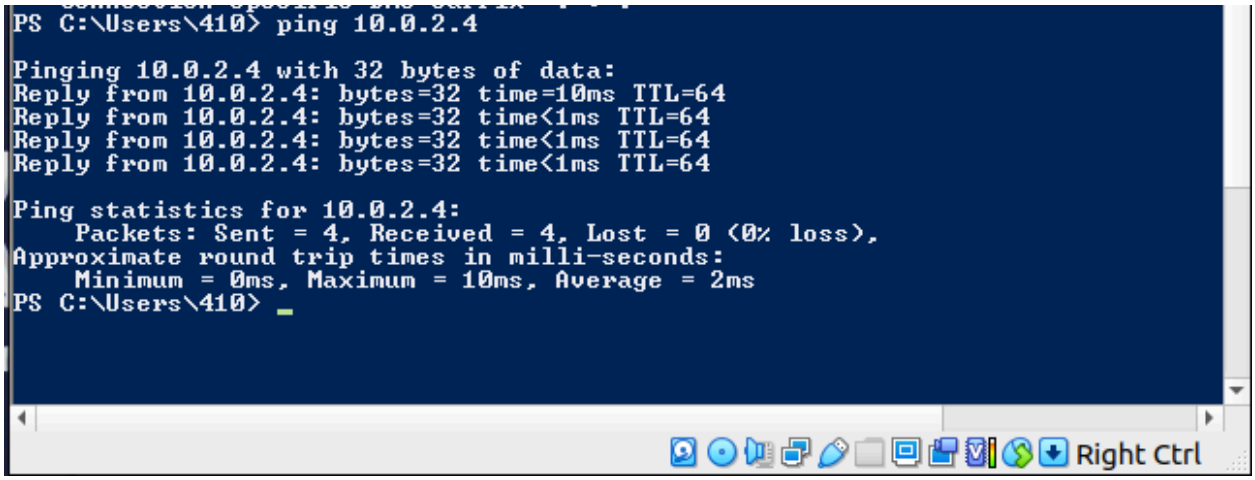
   Windows


win 77 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Windows PowerShell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.
PS C:\Users\410> ipconfig
Windows IP Configuration
Ethernet adapter Local Area Connection:
Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::9975:b539:1424:d3bc%11
IPv4 Address. . . . . : 10.0.2.15
Subnet Mask . . . . . : 255.255.255.0
Default Gateway . . . . . : 10.0.2.1
Tunnel adapter isatap..:
Media State . . . . . : Media disconnected
Connection-specific DNS Suffix . :

Kali


kali@kali: ~
File Actions Edit View Help
── 10.0.2.15 ping statistics ──
21 packets transmitted, 0 received, 100% packet loss, time 20567ms
┌──(kali㉿kali)-[~]
└─$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
inet 10.0.2.4 netmask 255.255.255.0 broadcast 10.0.2.255
inet6 fe80::d49:1536:c875:6d45 prefixlen 64 scopeid 0×20<link>
ether 08:00:27:21:b1:d0 txqueuelen 1000 (Ethernet)
RX packets 245 bytes 112893 (110.2 KiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 735 bytes 88883 (86.7 KiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
inet 127.0.0.1 netmask 255.0.0.0
inet6 ::1 prefixlen 128 scopeid 0×10<host>
loop txqueuelen 1000 (Local Loopback)
RX packets 4 bytes 240 (240.0 B)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 4 bytes 240 (240.0 B)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
┌──(kali㉿kali)-[~]
└─$

2. Ping


PS C:\Users\410> ping 10.0.2.4
Pinging 10.0.2.4 with 32 bytes of data:
Reply from 10.0.2.4: bytes=32 time=10ms TTL=64
Reply from 10.0.2.4: bytes=32 time<1ms TTL=64
Reply from 10.0.2.4: bytes=32 time<1ms TTL=64
Reply from 10.0.2.4: bytes=32 time<1ms TTL=64
Ping statistics for 10.0.2.4:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 2ms
PS C:\Users\410>
Right Ctrl

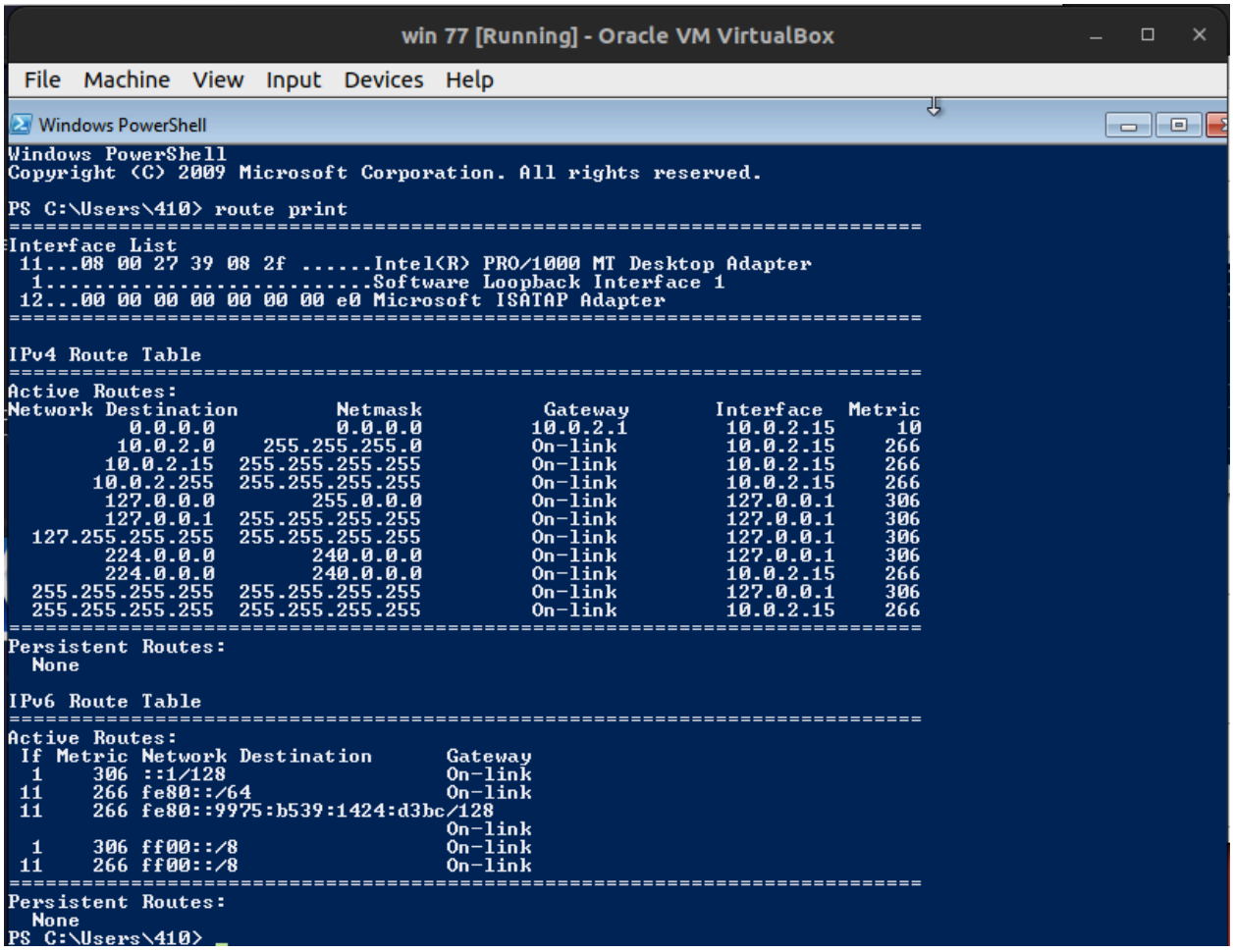
```
┌──(kali㉿kali)-[~]
└─$ ping 10.0.2.15
PING 10.0.2.15 (10.0.2.15) 56(84) bytes of data.
^C
── 10.0.2.15 ping statistics ──
460 packets transmitted, 0 received, 100% packet loss, time 471989ms
```

3. Route (Traceroute/ Tracert)

Kali

```
┌──(kali㉿kali)-[~]
└─$ ip route show
default via 10.0.2.1 dev eth0 proto dhcp src 10.0.2.4 metric 100
10.0.2.0/24 dev eth0 proto kernel scope link src 10.0.2.4 metric 100
```

Windows



4. Arp

Kali

```
┌──(kali㉿kali)-[~]
└─$ arp -n
Address                  HWtype  HWaddress           Flags Mask            Iface
10.0.2.1                 ether   52:54:00:12:35:00   C                     eth0
10.0.2.15                ether   08:00:27:39:08:2f   C                     eth0
10.0.2.3                 ether   08:00:27:e0:a7:8d   C                     eth0

┌──(kali㉿kali)-[~]
└─$
```

Windows

```
PS C:\Users\410> arp -a

Interface: 10.0.2.15 --- 0xb
  Internet Address      Physical Address      Type
  10.0.2.1              52-54-00-12-35-00     dynamic
  10.0.2.3              08-00-27-e0-a7-8d     dynamic
  10.0.2.4              08-00-27-21-b1-d0     dynamic
  10.0.2.255            ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static
PS C:\Users\410>
```

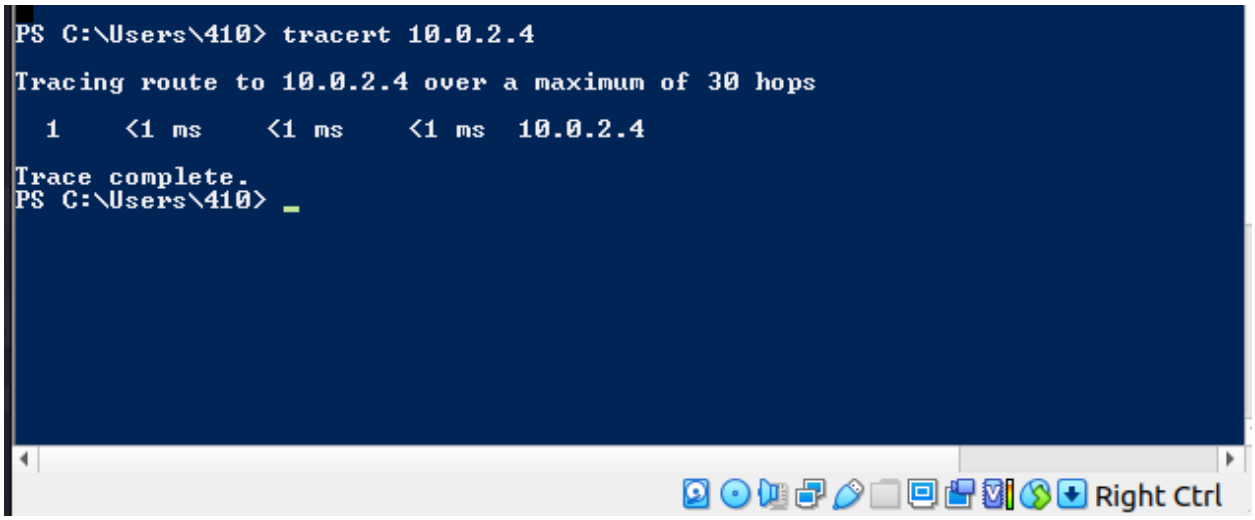
Right Ctrl

5. Traceroute

Kali

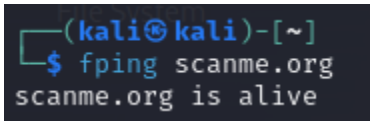
```
kali@kali: ~
File  Actions  Edit  View  Help
┌──(kali㉿kali)-[~]
└─$ traceroute 10.0.2.15
traceroute to 10.0.2.15 (10.0.2.15), 30 hops max, 60 byte packets
 1  * * *
 2  * * *
 3  * * *
 4  * * *
 5  * * *
 6  * * *
 7  * * *
 8  * * *
 9  * * *
10  * * *
11  * * *
12  * * *
13  * * *
14  * * *
15  * * *
16  * * *
17  * * *
18  * * *
19  * * *
20  * * *
21  * * *
22  * * *
23  * * *
24  * * *
25  * * *
26  * * *
27  * * *
28  * * *
29  * * *
30  * * *
```

Windows



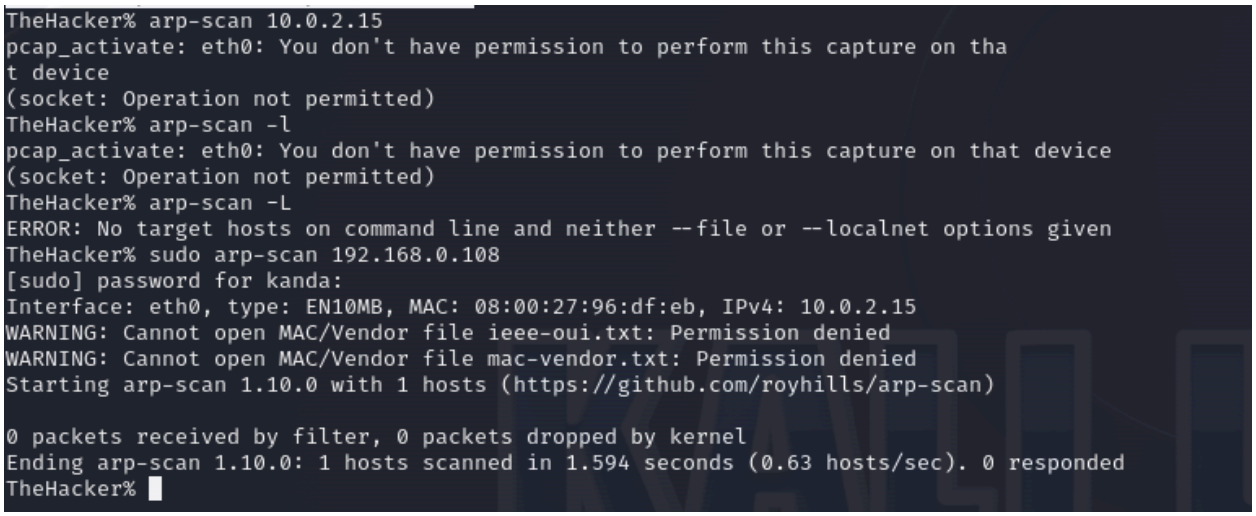
6. Fping

Kali



Windows

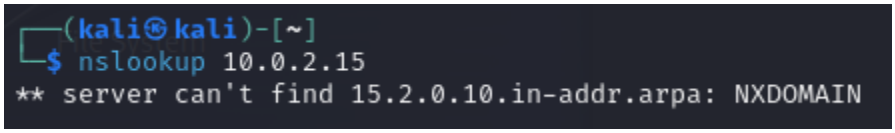
7. Arp-scan



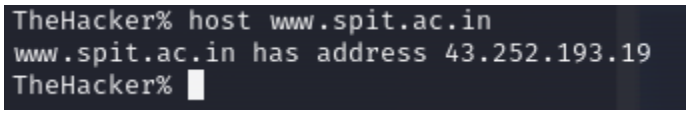
1B.2 DNS Interrogation

1. Nslookup

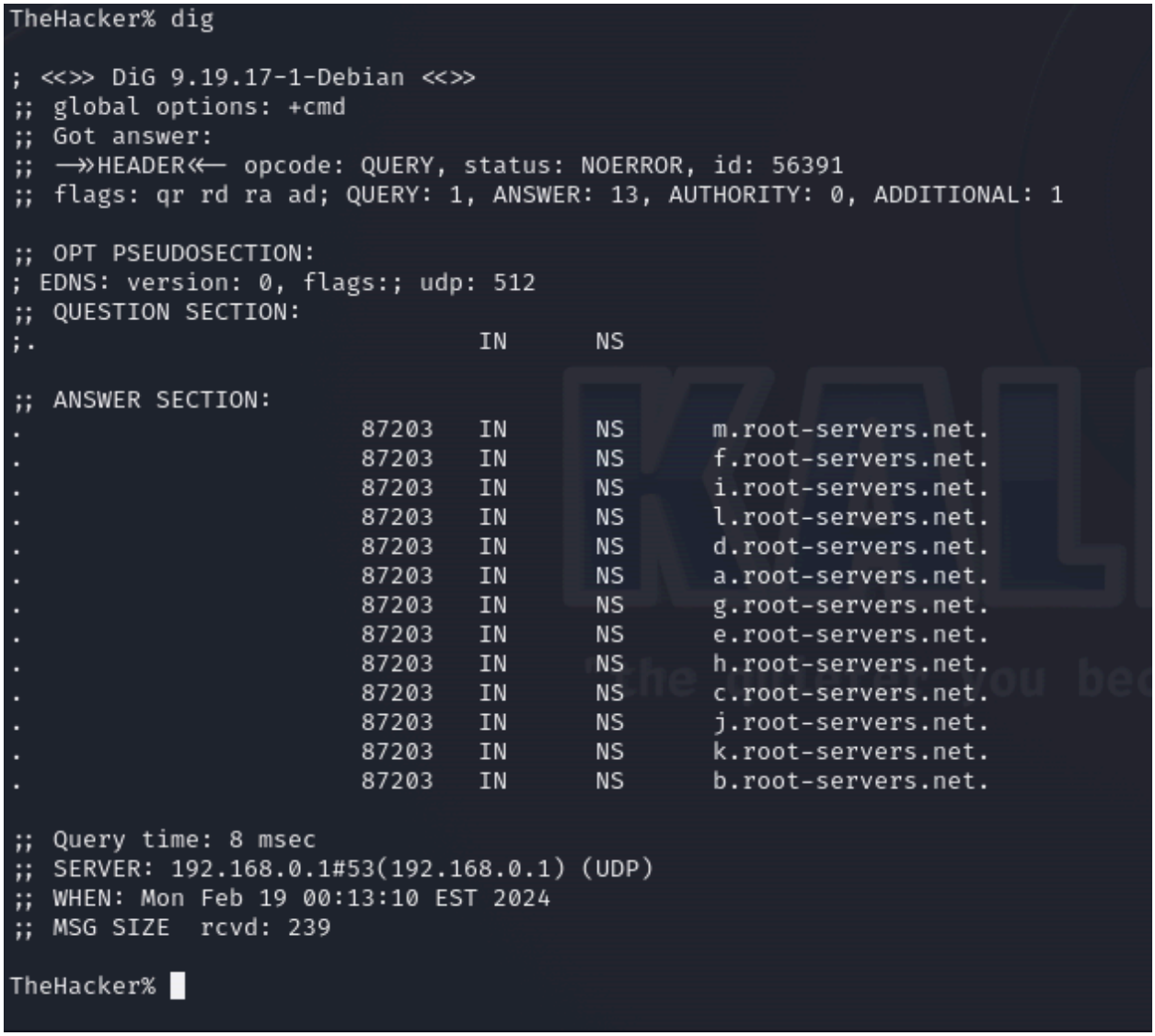
Kali



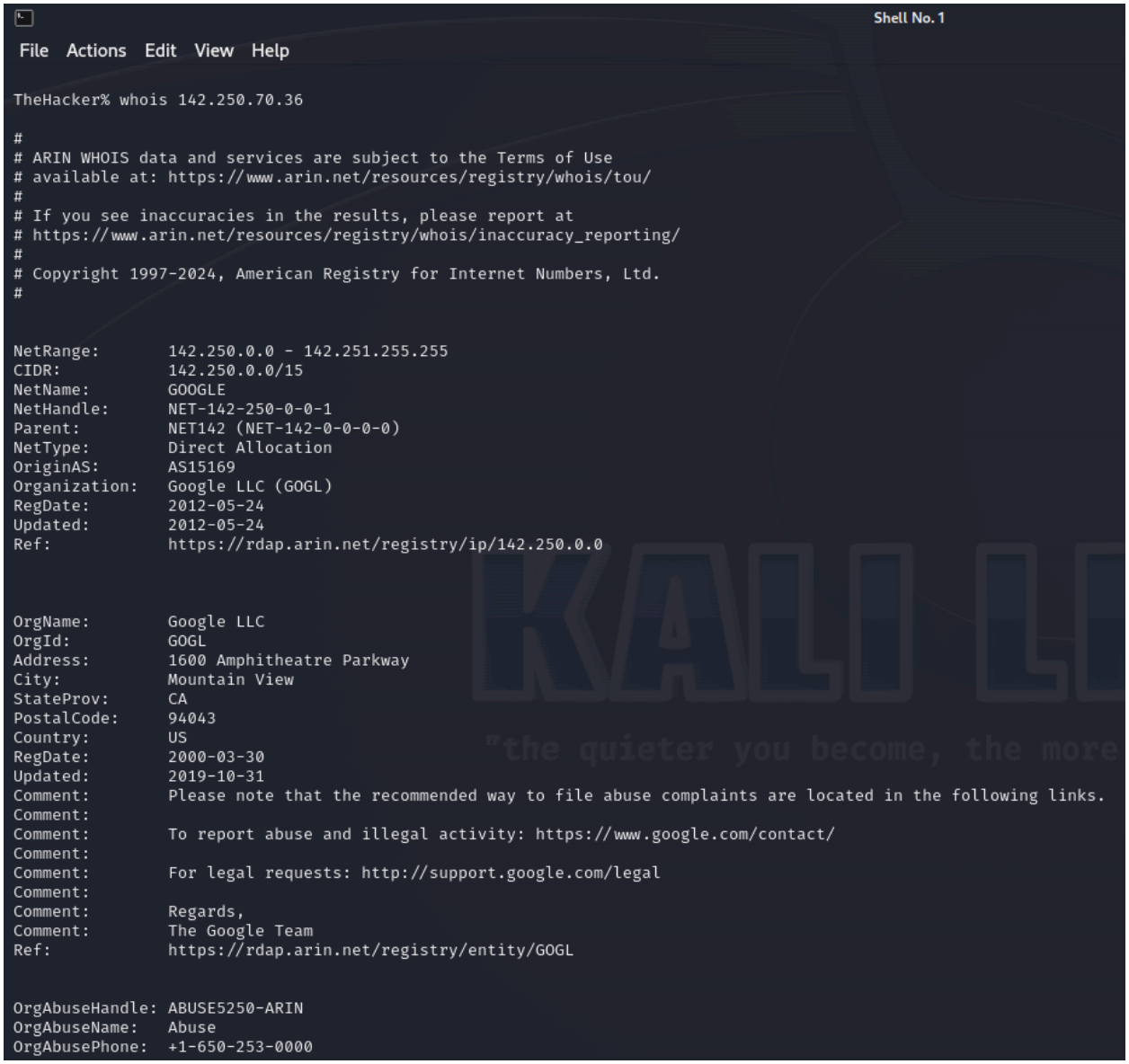
2. Host



3. Dig


TheHacker% dig
; <<>> DiG 9.19.17-1-Debian <<>>
;; global options: +cmd
;; Got answer:
;; →»HEADER«← opcode: QUERY, status: NOERROR, id: 56391
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 1
;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
;. IN NS
;; ANSWER SECTION:
. 87203 IN NS m.root-servers.net.
. 87203 IN NS f.root-servers.net.
. 87203 IN NS i.root-servers.net.
. 87203 IN NS l.root-servers.net.
. 87203 IN NS d.root-servers.net.
. 87203 IN NS a.root-servers.net.
. 87203 IN NS g.root-servers.net.
. 87203 IN NS e.root-servers.net.
. 87203 IN NS h.root-servers.net.
. 87203 IN NS c.root-servers.net.
. 87203 IN NS j.root-servers.net.
. 87203 IN NS k.root-servers.net.
. 87203 IN NS b.root-servers.net.
;; Query time: 8 msec
;; SERVER: 192.168.0.1#53(192.168.0.1) (UDP)
;; WHEN: Mon Feb 19 00:13:10 EST 2024
;; MSG SIZE rcvd: 239
TheHacker%

4. Whois


Shell No. 1
File Actions Edit View Help
TheHacker% whois 142.250.70.36
#
ARIN WHOIS data and services are subject to the Terms of Use
available at: https://www.arin.net/resources/registry/whois/tou/
#
If you see inaccuracies in the results, please report at
https://www.arin.net/resources/registry/whois/inaccuracy_reporting/
#
Copyright 1997-2024, American Registry for Internet Numbers, Ltd.
#
NetRange: 142.250.0.0 - 142.251.255.255
CIDR: 142.250.0.0/15
NetName: GOOGLE
NetHandle: NET-142-250-0-0-1
Parent: NET142 (NET-142-0-0-0-0)
NetType: Direct Allocation
OriginAS: AS15169
Organization: Google LLC (GOGL)
RegDate: 2012-05-24
Updated: 2012-05-24
Ref: https://rdap.arin.net/registry/ip/142.250.0.0
OrgName: Google LLC
OrgId: GOGL
Address: 1600 Amphitheatre Parkway
City: Mountain View
StateProv: CA
PostalCode: 94043
Country: US
RegDate: 2000-03-30
Updated: 2019-10-31
Comment: Please note that the recommended way to file abuse complaints are located in the following links.
Comment:
Comment: To report abuse and illegal activity: https://www.google.com/contact/
Comment:
Comment: For legal requests: http://support.google.com/legal
Comment:
Comment: Regards,
Comment: The Google Team
Ref: https://rdap.arin.net/registry/entity/GOGL
OrgAbuseHandle: ABUSE5250-ARIN
OrgAbuseName: Abuse
OrgAbusePhone: +1-650-253-0000

1B.3 Web Reconnaissance

1. **DNSstuff**
2. **Whois**

Lab1 B- Network Scanning

Objective: Scan the network to identify open ports, OS detection, service scanning and vulnerability scanning.

Outcomes:

1. To install and use network scanner (nmap) and web server scanner (nikto)
2. To explore various scanning mechanisms.
3. To enumerate the open ports and identify vulnerable services.
4. To detect the operating system and associated vulnerability
5. To identify the exploit with respect to vulnerable services.

System Requirements:

3 workstations installed with Kali Linux/Fedora Linux Core/Ubuntu and Windows XP

Nmap, nmapfe, zenmap etc

Background: ISO-OSI Layered Architecture of Computer Communication Network

OSI Model Layer No	Layer	Layer Description	Protocols
Layer 7	Application	This layer involves the application software that is sending and receiving data	HTTP, FTP, and Telnet
Layer 6	Presentation	This layer defines how data is formatted or organized	ASCII, JPEG, PDF, PNG, and DOCX
Layer 5	Session	This layer involves application session control, management, synchronization, and termination	NetBIOS, PPTP, RPC, and SOCKS
Layer 4	Transport	This layer involves end-to-end communication services	TCP and UDP

Layer 3	Network	This layer involves logical system addressing	IPv4, IPv6, ICMP, and IPSec
Layer 2	Data link	This layer involves physical system addressing	ARP
Layer 1	Physical	This layer involves the data stream that is passed over the wire	

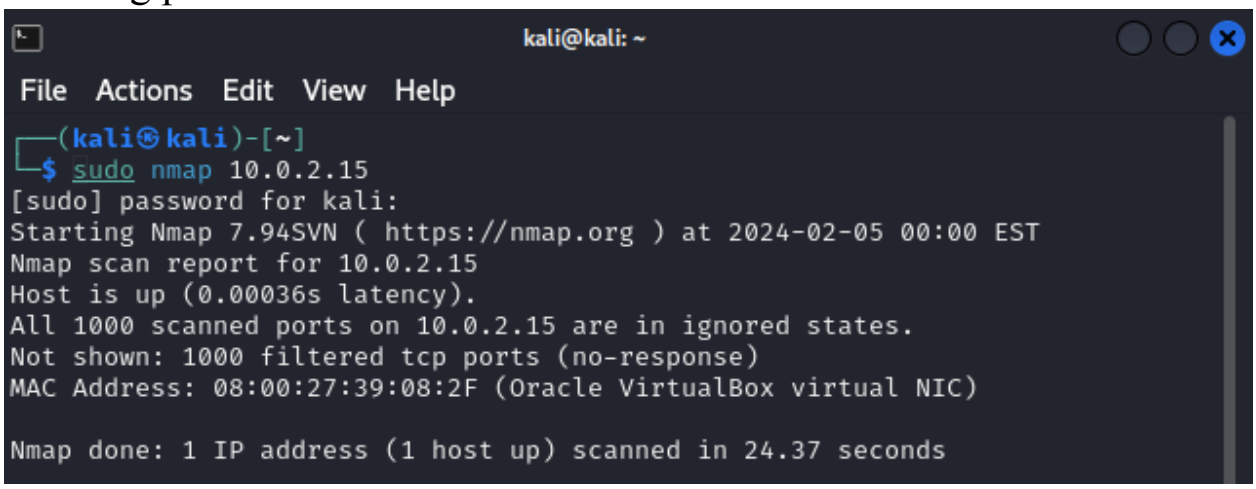
Discovery scanning:
Discovery scanning is the process of identifying live hosts on a network. In the context of penetration testing, this is usually performed to identify potential targets for attack. The objective here is not to exhaust resources in gathering information about targets but instead to merely find out where the targets are logically located. The final product of our discovery should be a list of IP addresses that we can use for further analysis. In this laboratory, we will use how to discover hosts on a network by using protocols operating at **layer 2, layer 3, and layer 4 of the OSI model.** This will include each of the following steps using:

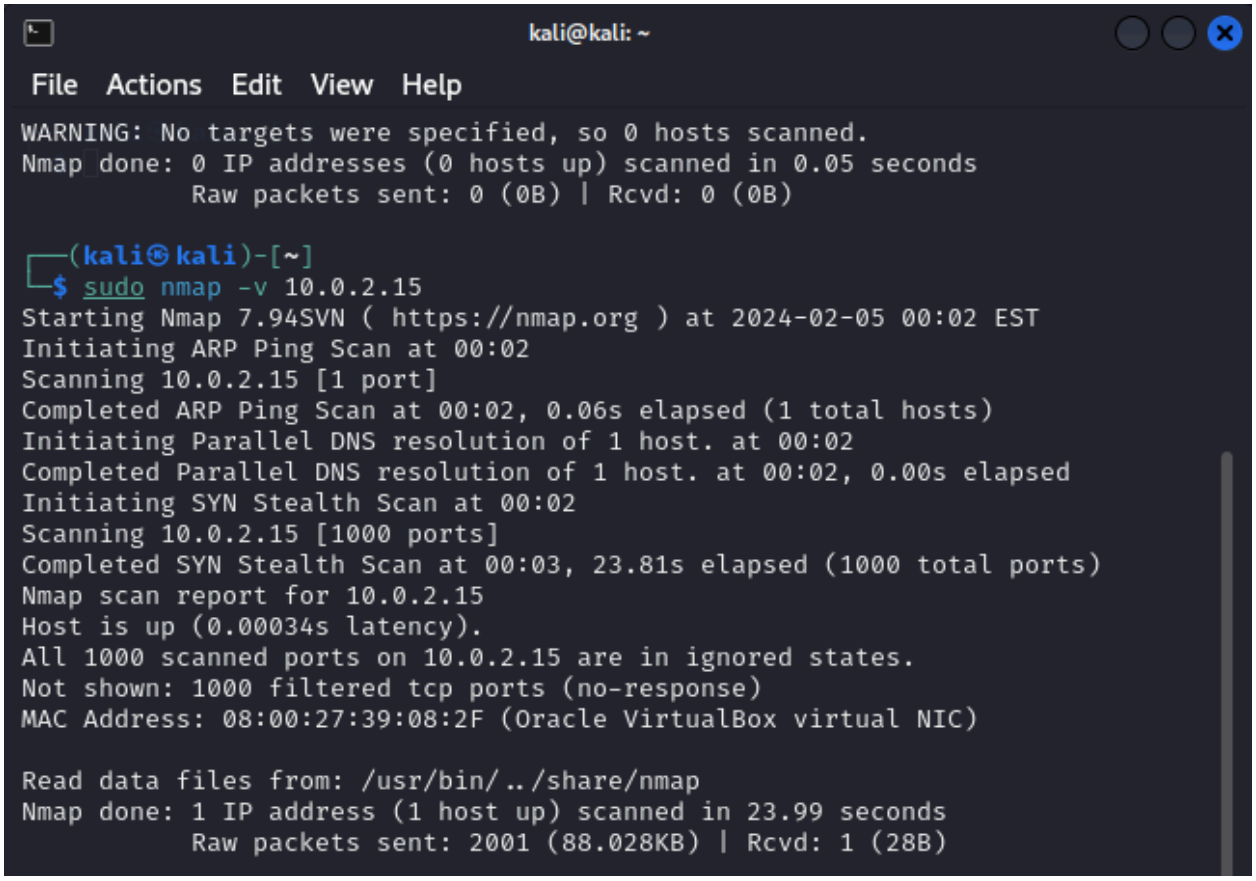
- **Scapy** to perform layer 2 discovery
- **ARPing** to perform layer 2 discovery
- **Nmap** to perform layer 2 discovery
- **NetDiscover** to perform layer 2 discovery
- **Metasploit** to perform layer 2 discovery
- **ICMP** ping to perform layer 3 discovery
- **Scapy** to perform layer 3 discovery
- **Nmap** to perform layer 3 discovery
- **fping** to perform layer 3 discovery

- **hping3** to perform layer 3 discovery
- **Scapy** to perform layer 4 discovery
- **Nmap** to perform layer 4 discovery
- **hping3** to perform layer 4 discovery

Procedure: Let explore nmap: Network host scanner, port scanner, OS fingerprinting, Service scanner and vulnerability scanner.
NMAP: Network exploration tool and security/port scanner
Description:
Nmap is short for Network Mapper. It is an open-source security tool for network exploration, security scanning, and auditing.

1: Scan a single host or an IP address (IPv4)
$sudo nmap 192.168.1.1
Scan a host name with more inf
$sudo nmap -v servername
Short analysis:
The command uses Nmap to check if the specified server (given by "servername") or IP address is active and running, while also scanning all available ports on the server. The -v flag in the Nmap command enables verbose mode, providing more detailed output during the scanning process.





2. Scan multiple IP address or subnet (IPv4)

$sudo nmap 192.168.1.1 192.168.1.2 192.168.1.3

works with same subnet i.e. 192.168.1.0/24

$sudo nmap 192.168.1.1,2,3

You can scan a range of IP address too:

$sudo nmap 192.168.1.1-20

You can scan a range of IP address using a wildcard:

$sudo nmap 192.168.1.*

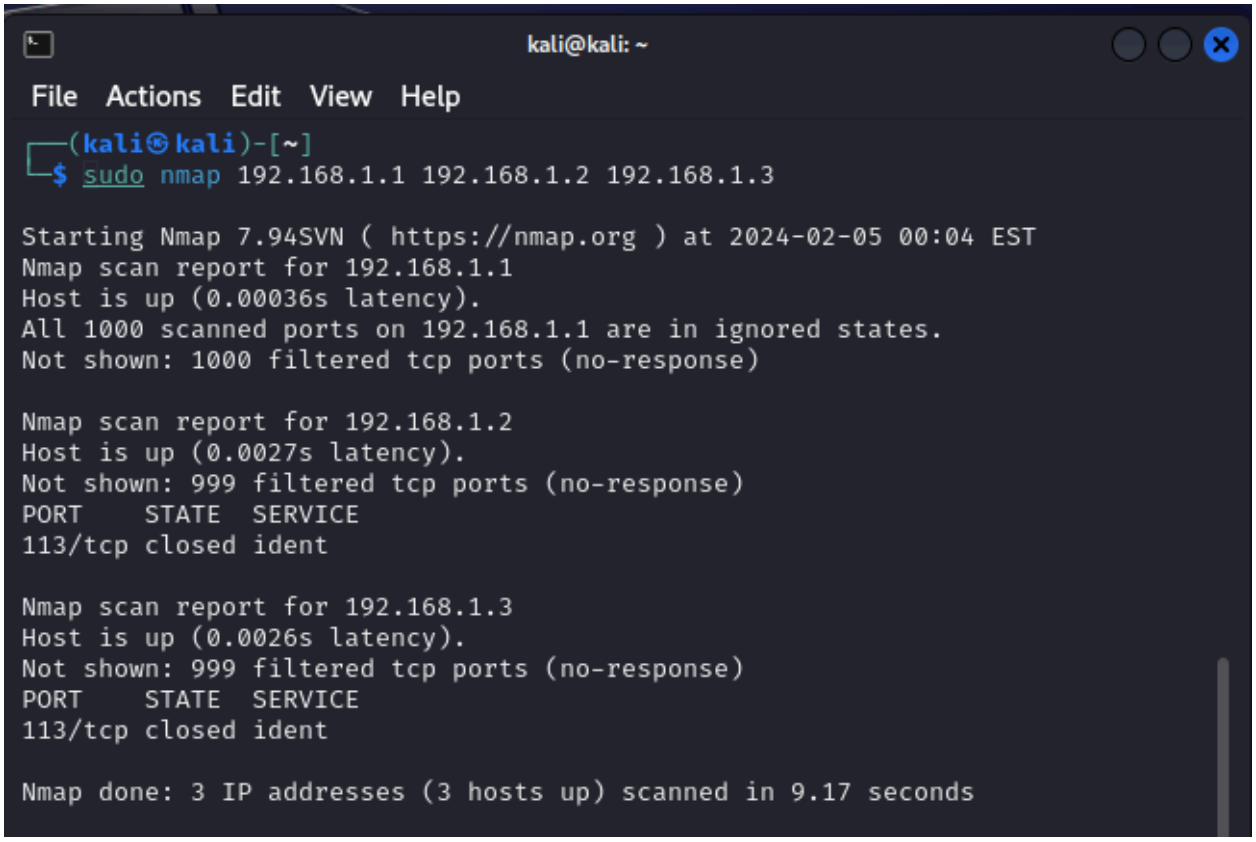
Finally, you scan an entire subnet:

$sudo nmap 192.168.1.0/24

Short analysis:

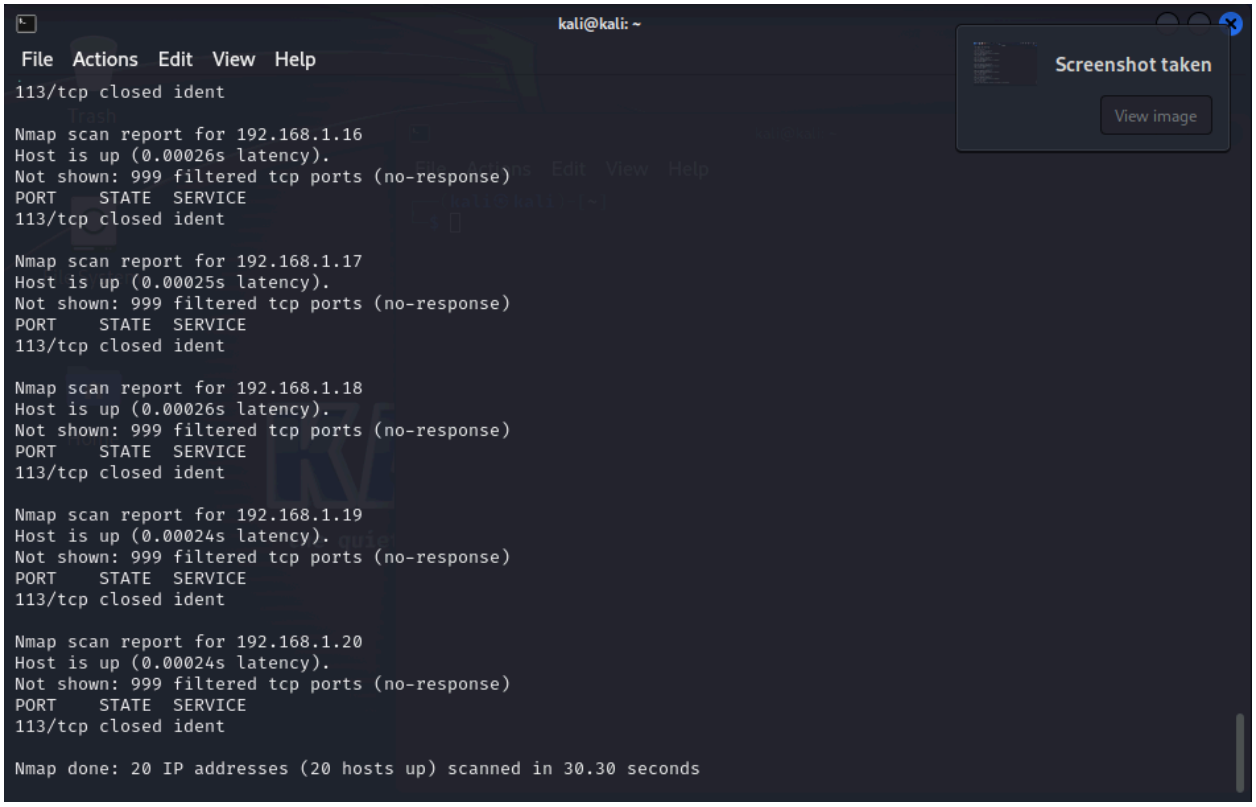
The above command show us that there are different ways to scan multiple IP addresses such as 192.168.1.1-20 which scans a range of IP addresses from 192.168.1.1 to 192.168.1.20, inclusive and we can also use wildcard (192.168.1.*) to scan all hosts in the 192.168.1.0 subnet,

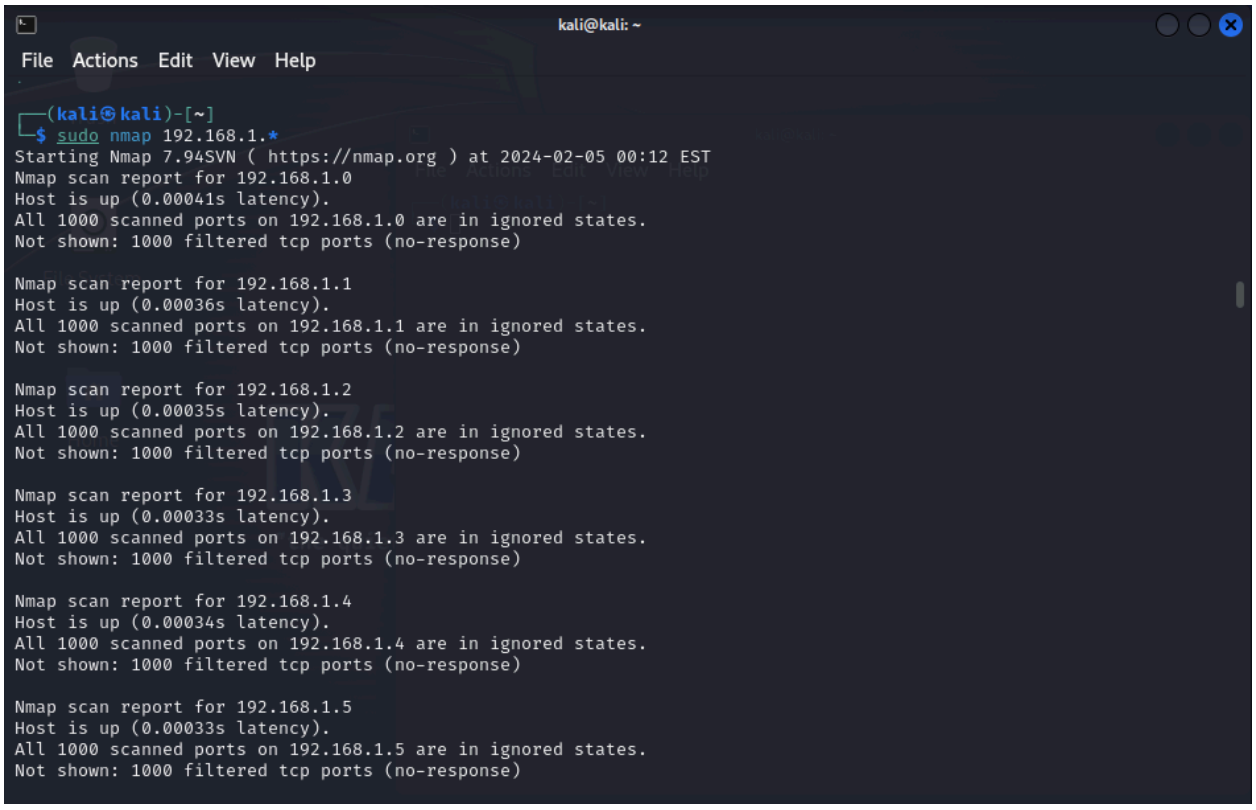
where '*' represents any valid host number (e.g., 192.168.1.1, 192.168.1.2, etc.).

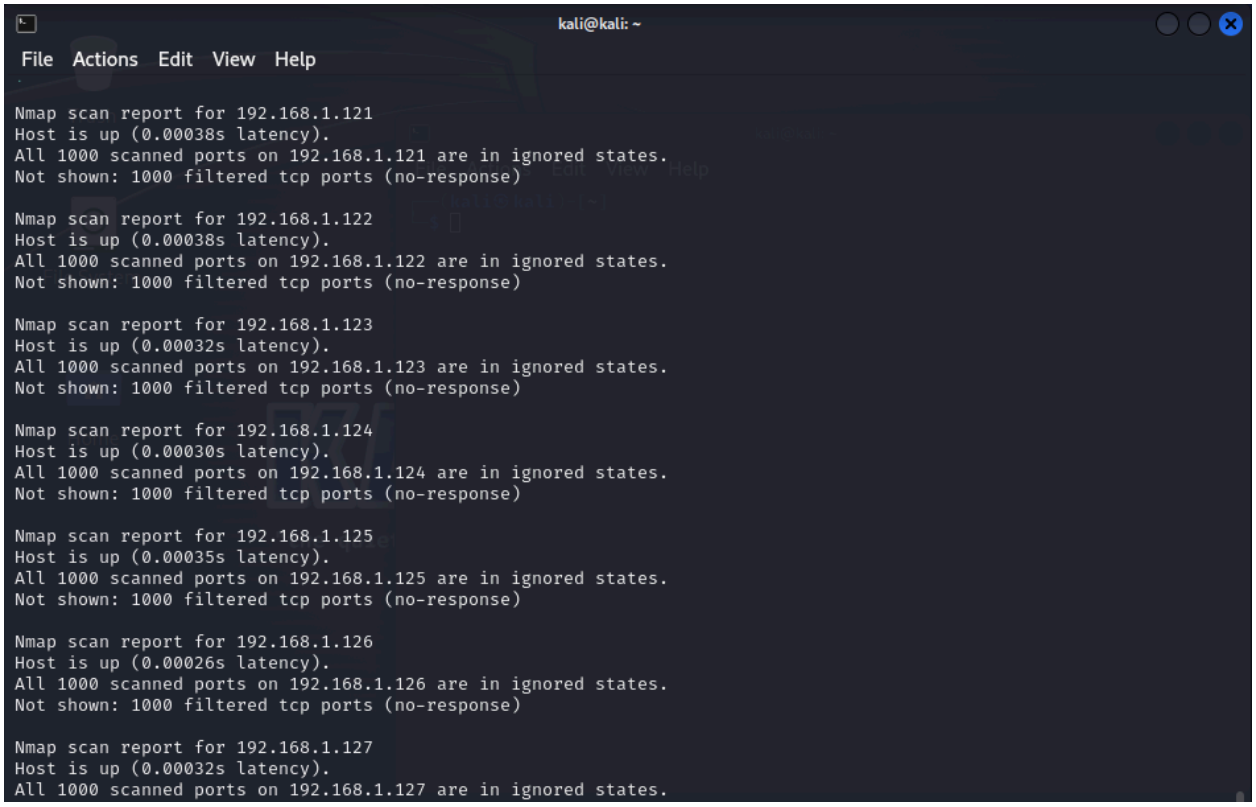


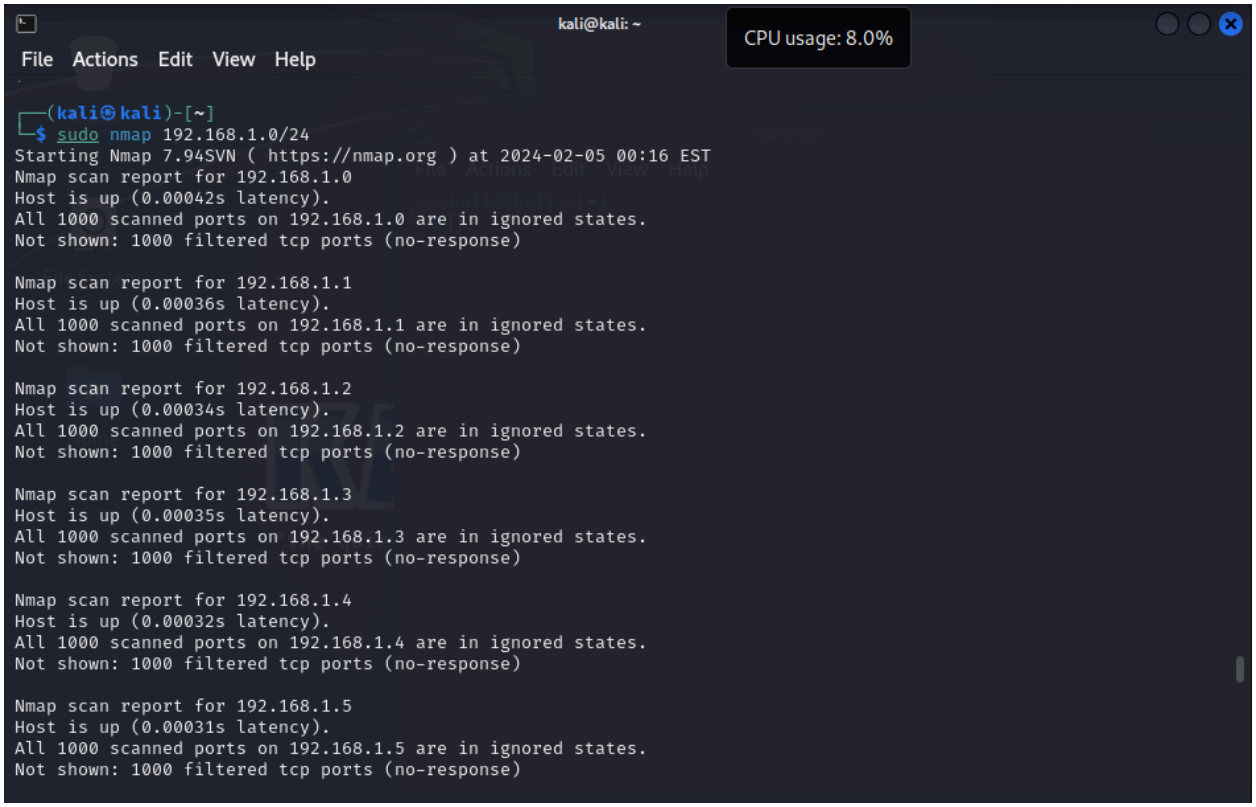
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.1.1,2,3
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 00:08 EST
Nmap scan report for 192.168.1.1
Host is up (0.00047s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.2
Host is up (0.00036s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.3
Host is up (0.00043s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap done: 3 IP addresses (3 hosts up) scanned in 10.18 seconds
```

```
kali@kali: ~
File  Actions  Edit  View  Help
└─$ sudo nmap 192.168.1.1-20
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 00:09 EST
Nmap scan report for 192.168.1.1
Host is up (0.00033s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.2
Host is up (0.00029s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.3
Host is up (0.00028s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.4
Host is up (0.00027s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.5
Host is up (0.00027s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT    STATE  SERVICE
113/tcp closed ident

Nmap scan report for 192.168.1.6
Host is up (0.00027s latency).
```


kali@kali: ~
File Actions Edit View Help
113/tcp closed ident
Nmap scan report for 192.168.1.16
Host is up (0.00026s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
Nmap scan report for 192.168.1.17
Host is up (0.00025s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
Nmap scan report for 192.168.1.18
Host is up (0.00026s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
Nmap scan report for 192.168.1.19
Host is up (0.00024s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
Nmap scan report for 192.168.1.20
Host is up (0.00024s latency).
Not shown: 999 filtered tcp ports (no-response)
PORT STATE SERVICE
113/tcp closed ident
Nmap done: 20 IP addresses (20 hosts up) scanned in 30.30 seconds
Screenshot taken
View image


kali@kali: ~
File Actions Edit View Help
(kali㉿kali)-[~]
$ sudo nmap 192.168.1.*
Starting Nmap 7.94SVN (https://nmap.org) at 2024-02-05 00:12 EST
Nmap scan report for 192.168.1.0
Host is up (0.00041s latency).
All 1000 scanned ports on 192.168.1.0 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for 192.168.1.1
Host is up (0.00036s latency).
All 1000 scanned ports on 192.168.1.1 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for 192.168.1.2
Host is up (0.00035s latency).
All 1000 scanned ports on 192.168.1.2 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for 192.168.1.3
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.1.3 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for 192.168.1.4
Host is up (0.00034s latency).
All 1000 scanned ports on 192.168.1.4 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Nmap scan report for 192.168.1.5
Host is up (0.00033s latency).
All 1000 scanned ports on 192.168.1.5 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)





3: Read list of hosts/networks from a file (IPv4)

The -iL option allows you to read the list of target systems using a text file. This is useful to
scan a large number of hosts/networks. Create a text file as follows:
cat > /tmp/test.txt
Sample outputs:
192.168.1.0/24
192.168.1.1/24
10.1.2.3
localhost
The syntax is:
$sudo nmap -iL /tmp/test.txt

Short analysis:

This command reads a file consisting of IP addresses and then scans them and get their network report.

4: Excluding hosts/networks (IPv4)

When scanning a large number of hosts/networks you can exclude hosts from a scan:
$sudo nmap 192.168.1.0/24 --exclude 192.168.1.5

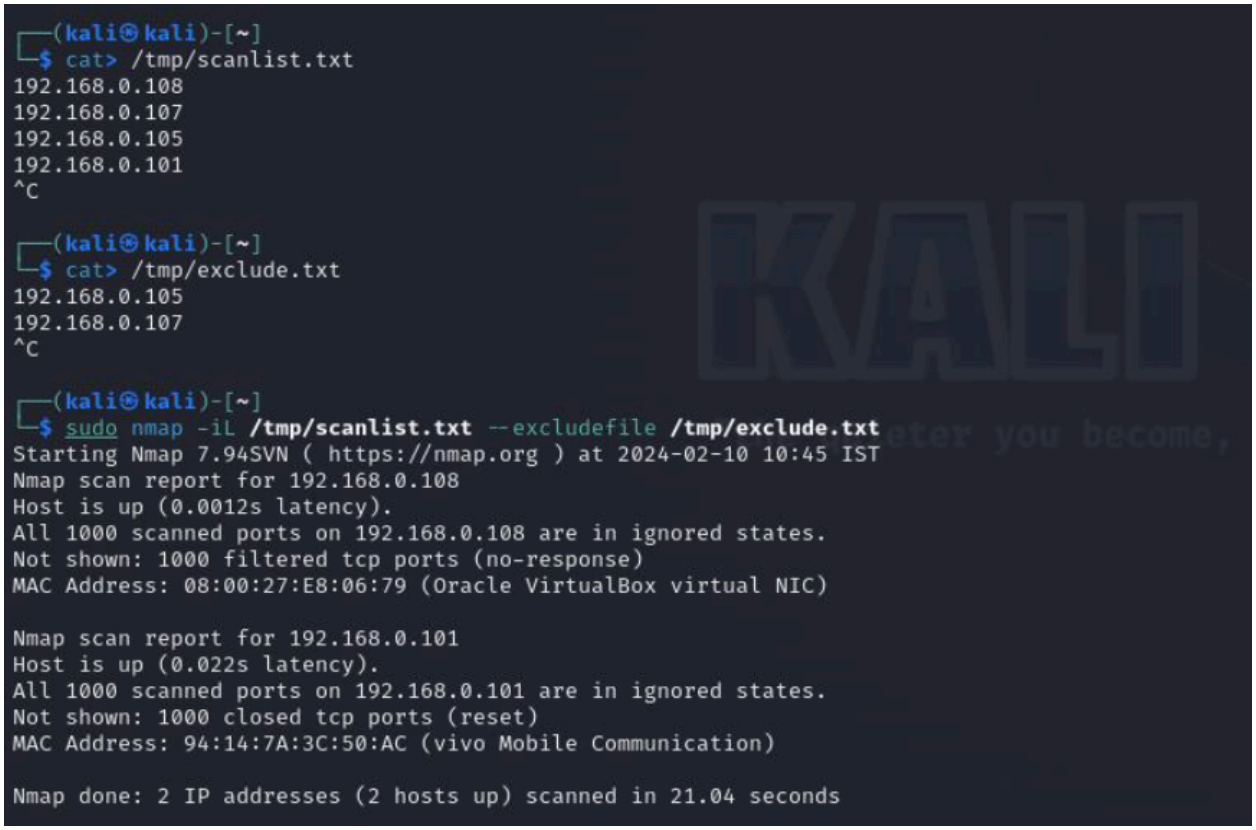
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.0.0/24 --exclude 192.168.0.105
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 10:28 IST
Nmap scan report for 192.168.0.1
Host is up (0.0066s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
1900/tcp open  upnp
MAC Address: 9C:53:22:F7:F3:34 (Unknown)

Nmap scan report for 192.168.0.108
Host is up (0.0011s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.0.107
Host is up (0.0000030s latency).
All 1000 scanned ports on 192.168.0.107 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 255 IP addresses (3 hosts up) scanned in 33.69 seconds
```

$sudo nmap 192.168.1.0/24 --exclude 192.168.1.5,192.168.1.254

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap 192.168.0.0/24 --exclude 192.168.0.105,192.168.0.107
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 10:32 IST
Nmap scan report for 192.168.0.1
Host is up (0.085s latency).
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE
23/tcp   open  telnet
53/tcp   open  domain
80/tcp   open  http
1900/tcp open  upnp
MAC Address: 9C:53:22:F7:F3:34 (Unknown)

Nmap scan report for 192.168.0.108
Host is up (0.00063s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 254 IP addresses (2 hosts up) scanned in 19.43 seconds
```

OR exclude list from a file called /tmp/exclude.txt

$sudo nmap -iL /tmp/scanlist.txt --excludefile /tmp/exclude.txt

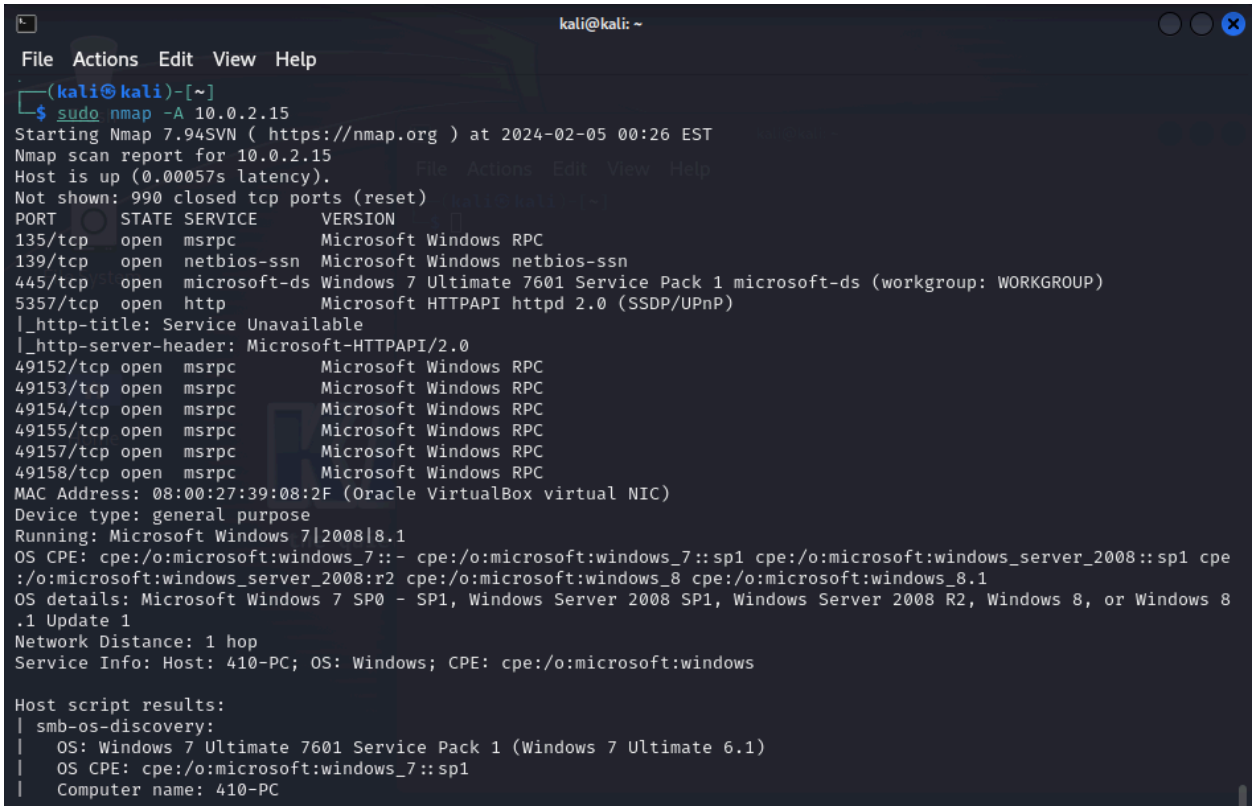


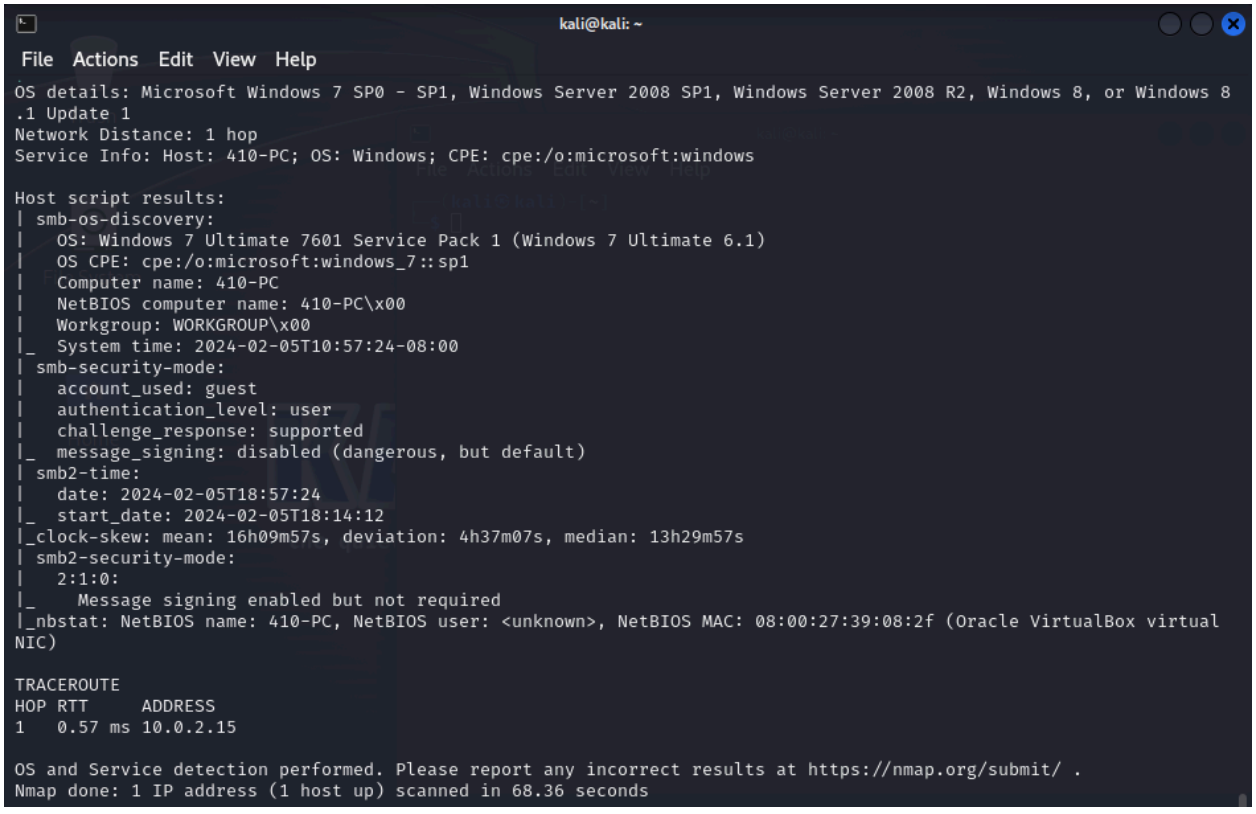
Short analysis:

To exclude specific IP addresses from scanning we can use –exclude option or we can make a file which contains IP addresses that we don't want to scan and use –excludefile option.

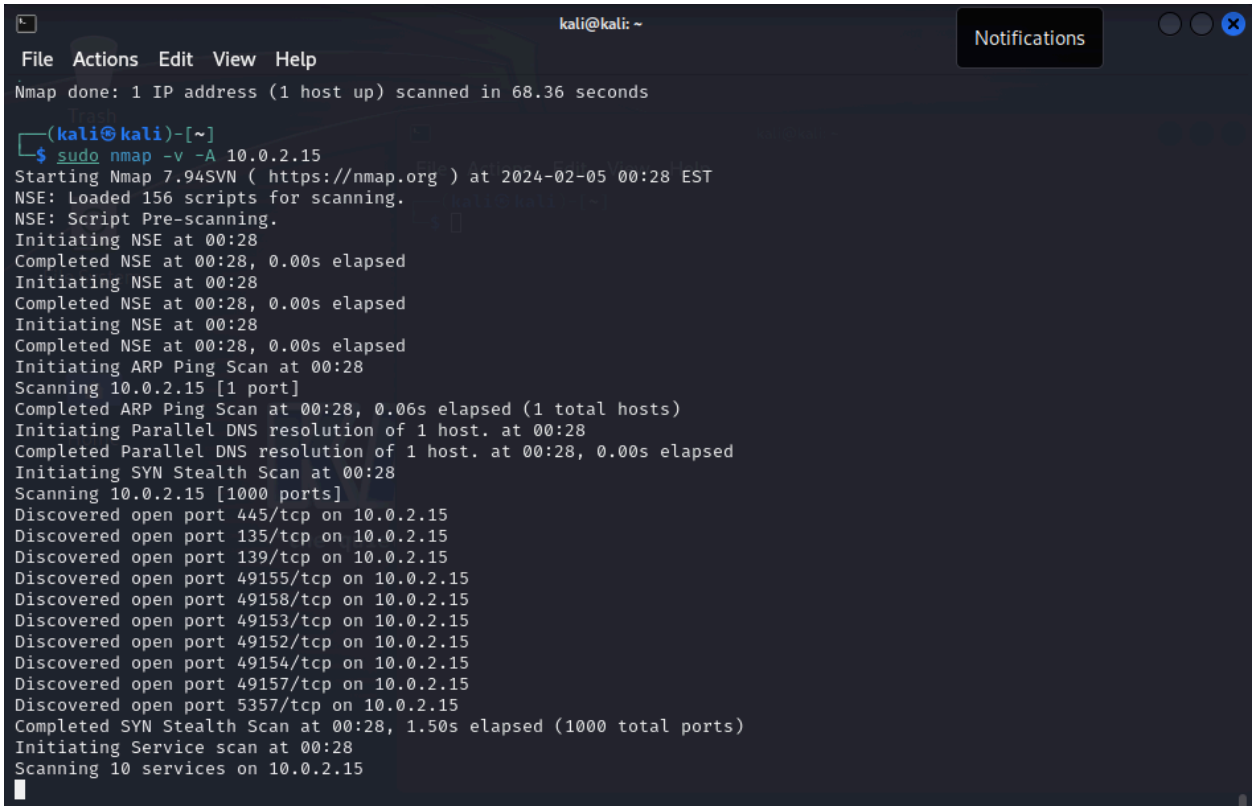
5: Turn on OS and version detection scanning script (IPv4)

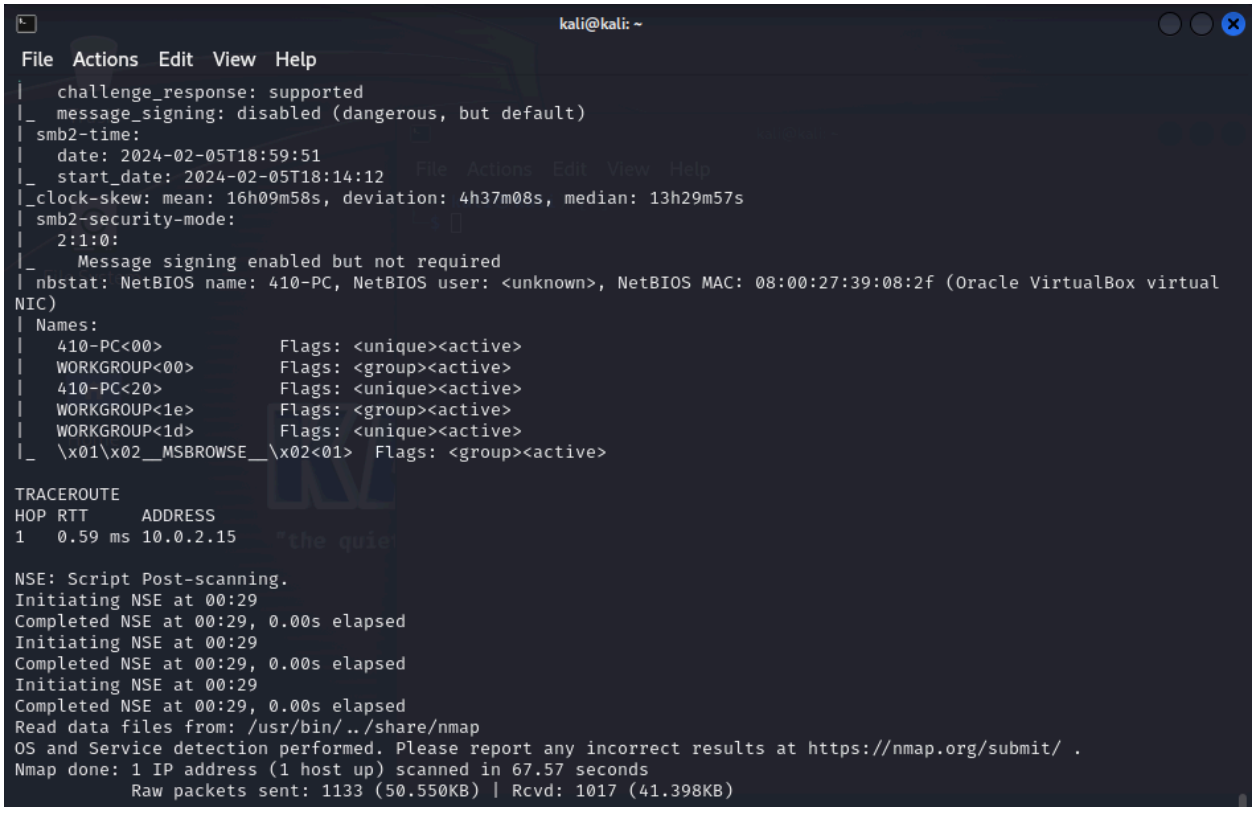
$sudo nmap -A 192.168.1.254





$sudo nmap -v -A 192.168.1.1


kali@kali: ~
File Actions Edit View Help
Notifications
Nmap done: 1 IP address (1 host up) scanned in 68.36 seconds
(kali㉿kali)-[~]
$ sudo nmap -v -A 10.0.2.15
Starting Nmap 7.94SVN (https://nmap.org) at 2024-02-05 00:28 EST
NSE: Loaded 156 scripts for scanning.
NSE: Script Pre-scanning.
Initiating NSE at 00:28
Completed NSE at 00:28, 0.00s elapsed
Initiating NSE at 00:28
Completed NSE at 00:28, 0.00s elapsed
Initiating NSE at 00:28
Completed NSE at 00:28, 0.00s elapsed
Initiating ARP Ping Scan at 00:28
Scanning 10.0.2.15 [1 port]
Completed ARP Ping Scan at 00:28, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 00:28
Completed Parallel DNS resolution of 1 host. at 00:28, 0.00s elapsed
Initiating SYN Stealth Scan at 00:28
Scanning 10.0.2.15 [1000 ports]
Discovered open port 445/tcp on 10.0.2.15
Discovered open port 135/tcp on 10.0.2.15
Discovered open port 139/tcp on 10.0.2.15
Discovered open port 49155/tcp on 10.0.2.15
Discovered open port 49158/tcp on 10.0.2.15
Discovered open port 49153/tcp on 10.0.2.15
Discovered open port 49152/tcp on 10.0.2.15
Discovered open port 49154/tcp on 10.0.2.15
Discovered open port 49157/tcp on 10.0.2.15
Discovered open port 5357/tcp on 10.0.2.15
Completed SYN Stealth Scan at 00:28, 1.50s elapsed (1000 total ports)
Initiating Service scan at 00:28
Scanning 10 services on 10.0.2.15


kali@kali: ~
File Actions Edit View Help
| challenge_response: supported
|_ message_signing: disabled (dangerous, but default)
| smb2-time:
| date: 2024-02-05T18:59:51
|_ start_date: 2024-02-05T18:14:12
|_clock-skew: mean: 16h09m58s, deviation: 4h37m08s, median: 13h29m57s
| smb2-security-mode:
| 2:1:0:
|_ Message signing enabled but not required
| nbstat: NetBIOS name: 410-PC, NetBIOS user: <unknown>, NetBIOS MAC: 08:00:27:39:08:2f (Oracle VirtualBox virtual NIC)
| Names:
| 410-PC<00> Flags: <unique><active>
| WORKGROUP<00> Flags: <group><active>
| 410-PC<20> Flags: <unique><active>
| WORKGROUP<1e> Flags: <group><active>
| WORKGROUP<1d> Flags: <unique><active>
|_ \x01\x02__MSBROWSE__\x02<01> Flags: <group><active>
TRACEROUTE
HOP RTT ADDRESS
1 0.59 ms 10.0.2.15
NSE: Script Post-scanning.
Initiating NSE at 00:29
Completed NSE at 00:29, 0.00s elapsed
Initiating NSE at 00:29
Completed NSE at 00:29, 0.00s elapsed
Initiating NSE at 00:29
Completed NSE at 00:29, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 67.57 seconds
Raw packets sent: 1133 (50.550KB) | Rcvd: 1017 (41.398KB)

$sudo nmap -A -iL /tmp/scanlist.txt

Short analysis:

In this we are performing aggressive scan (-A) on the host. The aggressive scan includes OS detection, version detection, script scanning, and traceroute.

6: Find out if a host/network is protected by a firewall

$sudo nmap -sA 192.168.1.254

$sudo nmap -sA server1

Short analysis:

Here we are performing TCP ACK scan, Nmap sends TCP ACK packets to determine the state of the ports if ports are filtered then it indicate it has a firewall and if it has unfiltered port then it don’t has firewall.

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sA 10.0.2.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 00:30 EST
Nmap scan report for 10.0.2.15
Host is up (0.00059s latency).
All 1000 scanned ports on 10.0.2.15 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)
MAC Address: 08:00:27:39:08:2F (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.95 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sA 192.168.0.108
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:15 IST
Nmap scan report for 192.168.0.108
Host is up (0.00069s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.73 seconds

┌──(kali㉿kali)-[~]
└─$ sudo nmap -sA 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:23 IST
Nmap scan report for 192.168.0.108
Host is up (0.0012s latency).
All 1000 scanned ports on 192.168.0.108 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 21.49 seconds
```

7: Scan a host when protected by the firewall

$sudo nmap -PN 192.168.1.1

$sudo nmap -PN server1

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -PN 10.0.2.15
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-05 00:32 EST
Nmap scan report for 10.0.2.15
Host is up (0.00040s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49157/tcp open  unknown
49158/tcp open  unknown
MAC Address: 08:00:27:39:08:2F (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.57 seconds
```

Short analysis:

Can be used for scanning host when it is protected with firewall and obtain information. It performs scans on a host without using
host discovery. It assumes that the target is online, regardless of whether it responds to ping requests.

8: Scan a network and find out which servers and devices are up and running

This is known as host discovery or ping scan:

$sudo nmap -sP 192.168.1.0/24

$sudo nmap done: 256 IP addresses (4 hosts up) scanned in 2.80 second

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sP 192.168.0.0/24
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:57 IST
Nmap scan report for 192.168.0.1
Host is up (0.0066s latency).
MAC Address: 9C:53:22:F7:F3:34 (Unknown)
Nmap scan report for 192.168.0.105
Host is up (0.00043s latency).
MAC Address: DC:F5:05:ED:FE:C1 (AzureWave Technology)
Nmap scan report for 192.168.0.108
Host is up (0.00086s latency).
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.0.107
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.08 seconds
```

Short analysis: Uses the -sP option to perform a ping scan to determine which hosts are online.

9: How do I perform a fast scan?

$sudo nmap -F 192.168.1.1

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -F 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 15:59 IST
Nmap scan report for 192.168.0.108
Host is up (0.0011s latency).
Not shown: 91 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
49152/tcp open  unknown
49153/tcp open  unknown
49154/tcp open  unknown
49155/tcp open  unknown
49156/tcp open  unknown
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

Short analysis:

Uses the -F option to perform a fast scan. The fast scan is a quicker version of the default scan and focusing on identifying open ports on the target host.

10: Display the reason a port is in a particular state

$sudo nmap --reason 192.168.1.1

$sudo nmap --reason server1

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --reason 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 16:00 IST
Nmap scan report for 192.168.0.108
Host is up, received arp-response (0.00067s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE      REASON
135/tcp   open  msrpc        syn-ack ttl 128
139/tcp   open  netbios-ssn  syn-ack ttl 128
445/tcp   open  microsoft-ds syn-ack ttl 128
5357/tcp  open  wsdapi       syn-ack ttl 128
49152/tcp open  unknown      syn-ack ttl 128
49153/tcp open  unknown      syn-ack ttl 128
49154/tcp open  unknown      syn-ack ttl 128
49155/tcp open  unknown      syn-ack ttl 128
49156/tcp open  unknown      syn-ack ttl 128
49158/tcp open  unknown      syn-ack ttl 128
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.72 seconds
```

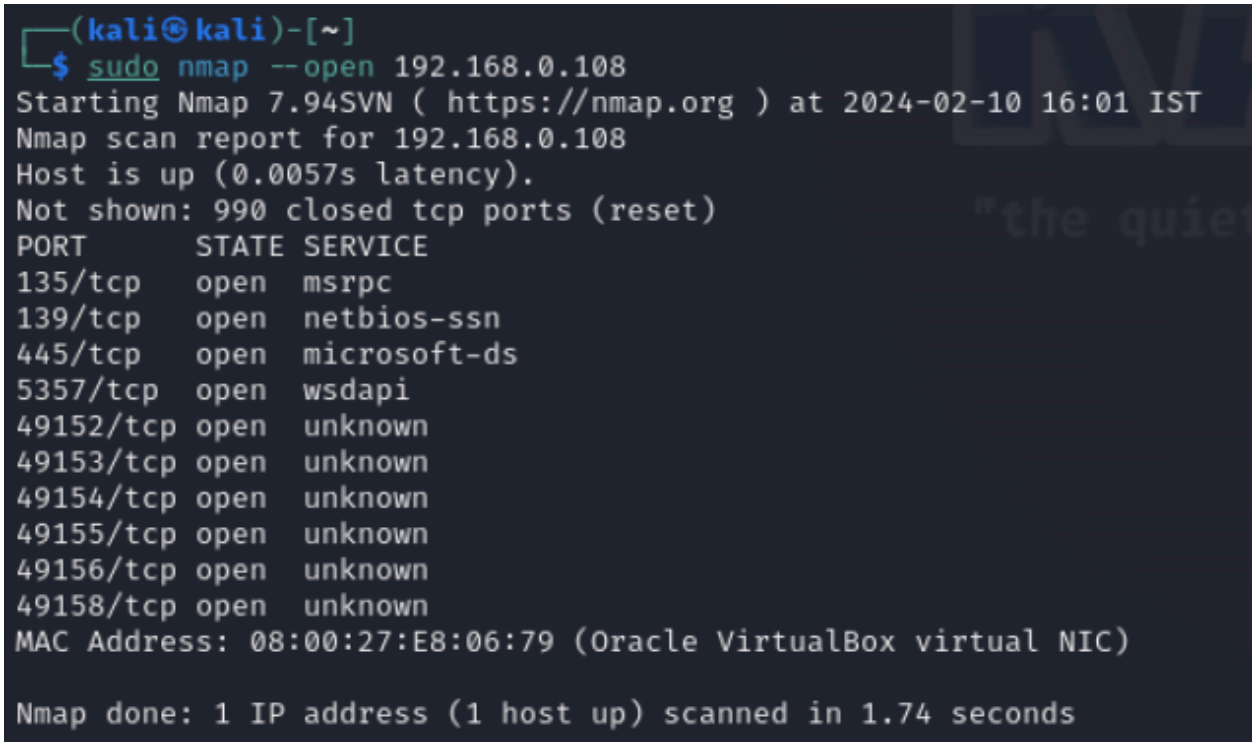
Short analysis:

Uses the --reason option to include additional information about why a port is in a particular state.

11: Only show open (or possibly open) ports

$sudo nmap --open 192.168.1.1

$sudo nmap --open server1



Short analysis:
Uses the --open option to filter and display only the open ports during the scan.

12: Show all packets sent and received
$sudo nmap --packet-trace 192.168.1.1
$sudo nmap --packet-trace server1

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap --packet-trace 192.168.0.108
Starting Nmap 7.945VN ( https://nmap.org ) at 2024-02-10 16:02 IST
SENT (0.0511s) ARP who-has 192.168.0.108 tell 192.168.0.107
RCVD (0.0517s) ARP reply 192.168.0.108 is-at 08:00:27:E8:06:79
NSOCK INFO [0.1180s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.1180s] nsock_connect_udp(): UDP connection requested to 192.168.0.1:53 (IOD #1) EID 8
NSOCK INFO [0.1180s] nsock_read(): Read request from IOD #1 [192.168.0.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.1180s] nsock_write(): Write request for 44 bytes to IOD #1 EID 27 [192.168.0.1:53]
NSOCK INFO [0.1190s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [192.168.0.1:53]
NSOCK INFO [0.1190s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [192.168.0.1:53]
NSOCK INFO [0.1240s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [192.168.0.1:53] (44 bytes): ............108.0.168.192.in-addr.arpa.....
NSOCK INFO [0.1240s] nsock_read(): Read request from IOD #1 [192.168.0.1:53] (timeout: -1ms) EID 34
NSOCK INFO [0.1240s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [0.1240s] nevent_delete(): nevent_delete on event #34 (type READ)
SENT (0.1528s) TCP 192.168.0.107:59090 > 192.168.0.108:554 S ttl=45 id=11064 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1557s) TCP 192.168.0.107:59090 > 192.168.0.108:53 S ttl=43 id=38289 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1578s) TCP 192.168.0.107:59090 > 192.168.0.108:23 S ttl=55 id=39057 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1581s) TCP 192.168.0.107:59090 > 192.168.0.108:80 S ttl=53 id=28250 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1585s) TCP 192.168.0.107:59090 > 192.168.0.108:1025 S ttl=38 id=54997 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1586s) TCP 192.168.0.107:59090 > 192.168.0.108:139 S ttl=57 id=8781 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1587s) TCP 192.168.0.107:59090 > 192.168.0.108:25 S ttl=43 id=33375 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1587s) TCP 192.168.0.107:59090 > 192.168.0.108:143 S ttl=45 id=50534 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1588s) TCP 192.168.0.107:59090 > 192.168.0.108:587 S ttl=55 id=22148 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1589s) TCP 192.168.0.107:59090 > 192.168.0.108:22 S ttl=39 id=29144 iplen=44  seq=1964250931 win=1024 <mss 1460>
RCVD (0.1522s) TCP 192.168.0.108:554 > 192.168.0.107:59090 RA ttl=128 id=9455 iplen=40  seq=0 win=0
RCVD (0.1544s) TCP 192.168.0.108:53 > 192.168.0.107:59090 RA ttl=128 id=9456 iplen=40  seq=0 win=0
RCVD (0.1570s) TCP 192.168.0.108:23 > 192.168.0.107:59090 RA ttl=128 id=9457 iplen=40  seq=0 win=0
RCVD (0.1587s) TCP 192.168.0.108:80 > 192.168.0.107:59090 RA ttl=128 id=9458 iplen=40  seq=0 win=0
RCVD (0.1591s) TCP 192.168.0.108:1025 > 192.168.0.107:59090 RA ttl=128 id=9459 iplen=40  seq=0 win=0
RCVD (0.1591s) TCP 192.168.0.108:139 > 192.168.0.107:59090 SA ttl=128 id=9460 iplen=44  seq=4204074983 win=8192 <mss 1460>
RCVD (0.1591s) TCP 192.168.0.108:25 > 192.168.0.107:59090 RA ttl=128 id=9461 iplen=40  seq=0 win=0
RCVD (0.1594s) TCP 192.168.0.108:143 > 192.168.0.107:59090 RA ttl=128 id=9462 iplen=40  seq=0 win=0
RCVD (0.1595s) TCP 192.168.0.108:587 > 192.168.0.107:59090 RA ttl=128 id=9463 iplen=40  seq=0 win=0
RCVD (0.1600s) TCP 192.168.0.108:22 > 192.168.0.107:59090 RA ttl=128 id=9464 iplen=40  seq=0 win=0
SENT (0.1602s) TCP 192.168.0.107:59090 > 192.168.0.108:135 S ttl=59 id=58776 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1603s) TCP 192.168.0.107:59090 > 192.168.0.108:443 S ttl=56 id=10442 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1604s) TCP 192.168.0.107:59090 > 192.168.0.108:21 S ttl=41 id=22462 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1605s) TCP 192.168.0.107:59090 > 192.168.0.108:8888 S ttl=39 id=64607 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1606s) TCP 192.168.0.107:59090 > 192.168.0.108:3389 S ttl=50 id=47220 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1607s) TCP 192.168.0.107:59090 > 192.168.0.108:1723 S ttl=46 id=3620 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1608s) TCP 192.168.0.107:59090 > 192.168.0.108:113 S ttl=54 id=34540 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1609s) TCP 192.168.0.107:59090 > 192.168.0.108:1720 S ttl=48 id=33374 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (0.1610s) TCP 192.168.0.107:59090 > 192.168.0.108:995 S ttl=42 id=36498 iplen=44  seq=1964250931 win=1024 <mss 1460>
```

```
SENT (1.6173s) TCP 192.168.0.107:59090 > 192.168.0.108:8222 S ttl=57 id=36936 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6177s) TCP 192.168.0.107:59090 > 192.168.0.108:2135 S ttl=42 id=42681 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6187s) TCP 192.168.0.107:59090 > 192.168.0.108:9 S ttl=59 id=47749 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6191s) TCP 192.168.0.107:59090 > 192.168.0.108:9943 S ttl=42 id=60614 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6198s) TCP 192.168.0.107:59090 > 192.168.0.108:5221 S ttl=37 id=28873 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6202s) TCP 192.168.0.107:59090 > 192.168.0.108:19801 S ttl=54 id=41183 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6206s) TCP 192.168.0.107:59090 > 192.168.0.108:6543 S ttl=44 id=13527 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6213s) TCP 192.168.0.107:59090 > 192.168.0.108:3998 S ttl=51 id=27861 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6221s) TCP 192.168.0.107:59090 > 192.168.0.108:43 S ttl=59 id=17202 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6230s) TCP 192.168.0.107:59090 > 192.168.0.108:3211 S ttl=40 id=47856 iplen=44  seq=1964250931 win=1024 <mss 1460>
SENT (1.6233s) TCP 192.168.0.107:59090 > 192.168.0.108:1166 S ttl=43 id=186 iplen=44  seq=1964250931 win=1024 <mss 1460>
RCVD (1.6129s) TCP 192.168.0.108:2005 > 192.168.0.107:59090 RA ttl=128 id=10437 iplen=40  seq=0 win=0
RCVD (1.6133s) TCP 192.168.0.108:3268 > 192.168.0.107:59090 RA ttl=128 id=10438 iplen=40  seq=0 win=0
RCVD (1.6140s) TCP 192.168.0.108:711 > 192.168.0.107:59090 RA ttl=128 id=10439 iplen=40  seq=0 win=0
RCVD (1.6151s) TCP 192.168.0.108:8022 > 192.168.0.107:59090 RA ttl=128 id=10440 iplen=40  seq=0 win=0
RCVD (1.6158s) TCP 192.168.0.108:27352 > 192.168.0.107:59090 RA ttl=128 id=10441 iplen=40  seq=0 win=0
RCVD (1.6161s) TCP 192.168.0.108:873 > 192.168.0.107:59090 RA ttl=128 id=10442 iplen=40  seq=0 win=0
RCVD (1.6169s) TCP 192.168.0.108:464 > 192.168.0.107:59090 RA ttl=128 id=10443 iplen=40  seq=0 win=0
RCVD (1.6172s) TCP 192.168.0.108:8222 > 192.168.0.107:59090 RA ttl=128 id=10444 iplen=40  seq=0 win=0
RCVD (1.6180s) TCP 192.168.0.108:2135 > 192.168.0.107:59090 RA ttl=128 id=10445 iplen=40  seq=0 win=0
RCVD (1.6186s) TCP 192.168.0.108:9 > 192.168.0.107:59090 RA ttl=128 id=10446 iplen=40  seq=0 win=0
RCVD (1.6194s) TCP 192.168.0.108:9943 > 192.168.0.107:59090 RA ttl=128 id=10447 iplen=40  seq=0 win=0
RCVD (1.6198s) TCP 192.168.0.108:5221 > 192.168.0.107:59090 RA ttl=128 id=10448 iplen=40  seq=0 win=0
RCVD (1.6205s) TCP 192.168.0.108:19801 > 192.168.0.107:59090 RA ttl=128 id=10449 iplen=40  seq=0 win=0
RCVD (1.6208s) TCP 192.168.0.108:6543 > 192.168.0.107:59090 RA ttl=128 id=10450 iplen=40  seq=0 win=0
RCVD (1.6216s) TCP 192.168.0.108:3998 > 192.168.0.107:59090 RA ttl=128 id=10451 iplen=40  seq=0 win=0
RCVD (1.6226s) TCP 192.168.0.108:43 > 192.168.0.107:59090 RA ttl=128 id=10452 iplen=40  seq=0 win=0
RCVD (1.6226s) TCP 192.168.0.108:3211 > 192.168.0.107:59090 RA ttl=128 id=10453 iplen=40  seq=0 win=0
RCVD (1.6312s) TCP 192.168.0.108:1166 > 192.168.0.107:59090 RA ttl=128 id=10454 iplen=40  seq=0 win=0
Nmap scan report for 192.168.0.108
Host is up (0.0016s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5357/tcp  open  wsdapi
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: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.67 seconds
```

Short analysis:

Uses the --packet-trace option to show the detailed packet tracing information during the scan. It shows the individual packets sent and received during the scanning process.

13. To find the identify particular vulnerability for further exploit

$sudo nmap -sT -A --script=smb-check-vulns -Pn --script-args=unsafe=1 192.168.56.103

Or

$sudo nmap -n -sV 192.168.56.103

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -n -sV 192.168.0.108
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-02-10 16:14 IST
Nmap scan report for 192.168.0.108
Host is up (0.00079s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows 7 - 10 microsoft-ds (workgroup: WORKGROUP)
5357/tcp  open  http         Microsoft HTTPAPI httpd 2.0 (SSDP/UPnP)
49152/tcp open  msrpc        Microsoft Windows RPC
49153/tcp open  msrpc        Microsoft Windows RPC
49154/tcp open  msrpc        Microsoft Windows RPC
49155/tcp open  msrpc        Microsoft Windows RPC
49156/tcp open  msrpc        Microsoft Windows RPC
49158/tcp open  msrpc        Microsoft Windows RPC
MAC Address: 08:00:27:E8:06:79 (Oracle VirtualBox virtual NIC)
Service Info: Host: SANKET-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 61.57 seconds
```

Short analysis:

Performs a version detection scan on the host. The -n option disables DNS resolution for faster scanning, and -sV enables the identification of service and version information on open ports.

Analysis of nmap scan results

Refer:Nmap Scan to CSV (R3)

https://laconicwolf.com/2018/02/04/nmap-scan-csv/

Nmap results save as xml

$nmap -xO nmap_scan.xml -sT ....

Converting nmap_scan.xml to nmap_scan.csv is simple:
$python3 nmap_xml_parser.py -f nmap_scan.xml -csv nmap_scan.csv

Conclusion:

Therefore, we have utilized diverse network exploration commands with nmap to extract information about both the network and the host, providing additional details that could be exploited for potential malicious attacks on a target.