



REPORT OF SCANNING NETWORK

BY SACHCHITANAND YADAV

SCANNING NETWORK

MODULE - 3

Learning Objectives -

- Explain Network Scanning Concepts
- Demonstrate Various Scanning Techniques for Host Discovery
- Demonstrate Various Scanning Techniques for Port and Service Discovery
- Demonstrate Various Scanning Techniques for OS Discovery
- Demonstrate Various Techniques for Scanning Beyond IDS and Firewall
- Perform Network Scanning Using Various Scanning Tools
- Perform Network Scanning Using AI
- Explain Network Scanning Countermeasures

TABLE OF CONTENTS

1. Network Scanning Concepts

- 1.1 Purpose of Network Scanning
 - 1.2 Footprinting vs Scanning
 - 1.3 Key Scanning Methodologies
 - 1.4 TCP Control Flags & Session Management
 - 1.5 Three-Way Handshake & Session Termination
 - 1.6 Objectives of Network Scanning
 - 1.7 Analogy: “Doors & Windows” in Network Security
-

2. Host Discovery

- 2.1 Overview of Host Discovery
 - 2.2 Lab Scenario & Objectives
 - 2.3 Host Discovery Techniques
 - 2.4 Common Nmap Host Discovery Options
 - 2.5 Host Discovery Commands & Usage
-

3. Port and Service Discovery

- 3.1 Overview & Purpose
 - 3.2 Lab Scenario & Objectives
 - 3.3 Types of Port Scanning
 - 3.4 Nmap Port Scanning Techniques
 - 3.5 Port & Service Discovery Commands
 - 3.6 Zenmap Analysis & Topology Visuals
-

4. OS Discovery

- 4.1 Purpose of OS Fingerprinting
- 4.2 Lab Scenario & Objectives
- 4.3 Nmap OS Discovery Techniques
- 4.4 NSE Scripts for OS Fingerprinting
- 4.5 OS Discovery Commands

5. Scanning Beyond IDS and Firewalls

- 5.1 Importance of IDS/Firewall Evasion
 - 5.2 Common Evasion Techniques
 - 5.3 Packet Fragmentation
 - 5.4 Decoys, Spoofing, and Timing Tweaks
 - 5.5 Commands for Evasive Scanning
-

6. Network Scanning Using Various Tools

- 6.1 Overview of Scanning Tools
 - 6.2 Lab Scenario & Objectives
 - 6.3 Using Metasploit for Scanning
 - 6.4 Auxiliary Scanners in Metasploit
 - 6.5 Command Workflow in Metasploit
-

7. Network Scanning Using Hping3

- 7.1 Introduction to Hping3
 - 7.2 Common Hping3 Usage Commands
 - 7.3 ICMP Flood Using Hping3
 - 7.4 UDP Flood Using Random Source
 - 7.5 UDP Flood with Spoofed Source IP
 - 7.6 ACK Flood Attack
 - 7.7 SYN Flood Attack
-

8. Network Scanning Using AI

- 8.1 AI-Enhanced Recon Overview
 - 8.2 Lab Scenario & Objectives
 - 8.3 Using Gemini CLI for Scanning
 - 8.4 Integrating AI for Automation & Refinement
-

9. Network Scanning Countermeasures

- 9.1 Defensive Strategies
- 9.2 Hardening Systems Against Scanning
- 9.3 IDS/Firewall Tuning
- 9.4 Reducing Attack Surface
- 9.5 Monitoring & Logging Enhancements

SACHCHITANAND

Network Scanning Concepts: -

Network scanning is a crucial phase in a security assessment that logically follows the initial reconnaissance or **footprinting** stage. While footprinting involves passively gathering high-level, often publicly available, information about a target, the scanning phase uses this initial intelligence to actively probe the target's network.

The primary goal of network scanning is to discover more granular technical details about the target environment. This is achieved by sending packets to the target systems and analyzing the responses to identify:

- **Live hosts** on the network.
- **Open ports** and the specific **services** running on them (e.g., web servers, mail servers).
- The **operating systems** and system architecture of the target machines.

The information gathered during the scanning phase provides a detailed map of the target's attack surface, enabling a security professional to identify potential vulnerabilities that can be explored in subsequent phases of the assessment.

An Overview of Network Scanning

Network scanning is a critical phase of intelligence gathering that uses aggressive reconnaissance techniques to create a detailed profile of a target organization's network. It is a set of procedures used to identify live hosts, open ports, and active services. By discovering which machines are active and what operating systems they run, an analyst can map out the target's digital landscape. This process is fundamental for both ethical hackers performing security assessments and malicious attackers planning an intrusion.

The core of the scanning process involves sending a series of crafted **TCP/IP probes** to the target network and carefully analyzing the responses. This feedback loop allows the analyst to gather crucial information, including specific IP addresses, the target's operating system (OS) and architecture, and the services running on various ports.

The ultimate purpose is to discover exploitable communication channels and configuration lapses. The intelligence gathered during the scanning phase directly informs the development of an effective attack strategy.

Key Scanning Methodologies

Scanning can be broken down into three primary categories, each with a distinct goal.

- **Network Scanning** The primary goal here is to discover **live hosts** and their IP addresses within a network range. This procedure is essential for mapping out which systems are active and available for further probing, whether for a security assessment or a potential attack.
- **Port Scanning** This technique focuses on identifying which TCP and UDP ports on a target system are open, closed, or filtered. By probing these ports, an analyst can determine which services are running or in a "listening" state. The state of these services often provides valuable clues about the underlying OS and applications in use, which may harbor known vulnerabilities.
- **Vulnerability Scanning** This is a more advanced scan that actively checks systems for **known weaknesses**. A vulnerability scanner uses a catalog of known exploits and misconfigurations to test targets. The scanning engine sends specific requests designed to trigger a response if a vulnerability is present, such as an outdated software version or a directory traversal flaw.

The "Doors and Windows" Analogy

A useful analogy is to think of a computer system as a house. In this scenario, the network **ports** are the doors and windows—the primary points of entry. An intruder will systematically check each one to find an unlocked or weak entry point.

Generally, the more open ports a system has, the more vulnerable it is, as this increases the potential **attack surface**. However, it is crucial to understand that a single open port running a highly vulnerable service can pose a far greater risk than a system with many open ports that are properly secured and patched.

Primary Objectives of Network Scanning

The more an analyst knows about a target, the higher the probability of discovering security loopholes. The main objectives of scanning are:

1. **Identify Live Systems and Open Ports:** To discover which hosts on the network are active and which of their ports are open, providing potential pathways for an attack.
2. **Determine Operating System and Architecture:** Also known as **fingerprinting**, this involves identifying the target's OS. Since many vulnerabilities are specific to a particular operating system, this information is critical for crafting a successful exploit.

3. **Enumerate Services:** To discover precisely which services and applications are running on the target's open ports. Knowing the application and its version number allows an attacker to search for known exploits.
4. **Detect Vulnerabilities:** To use the collected information to identify specific, exploitable vulnerabilities on the target systems, which can then be used to compromise the network.

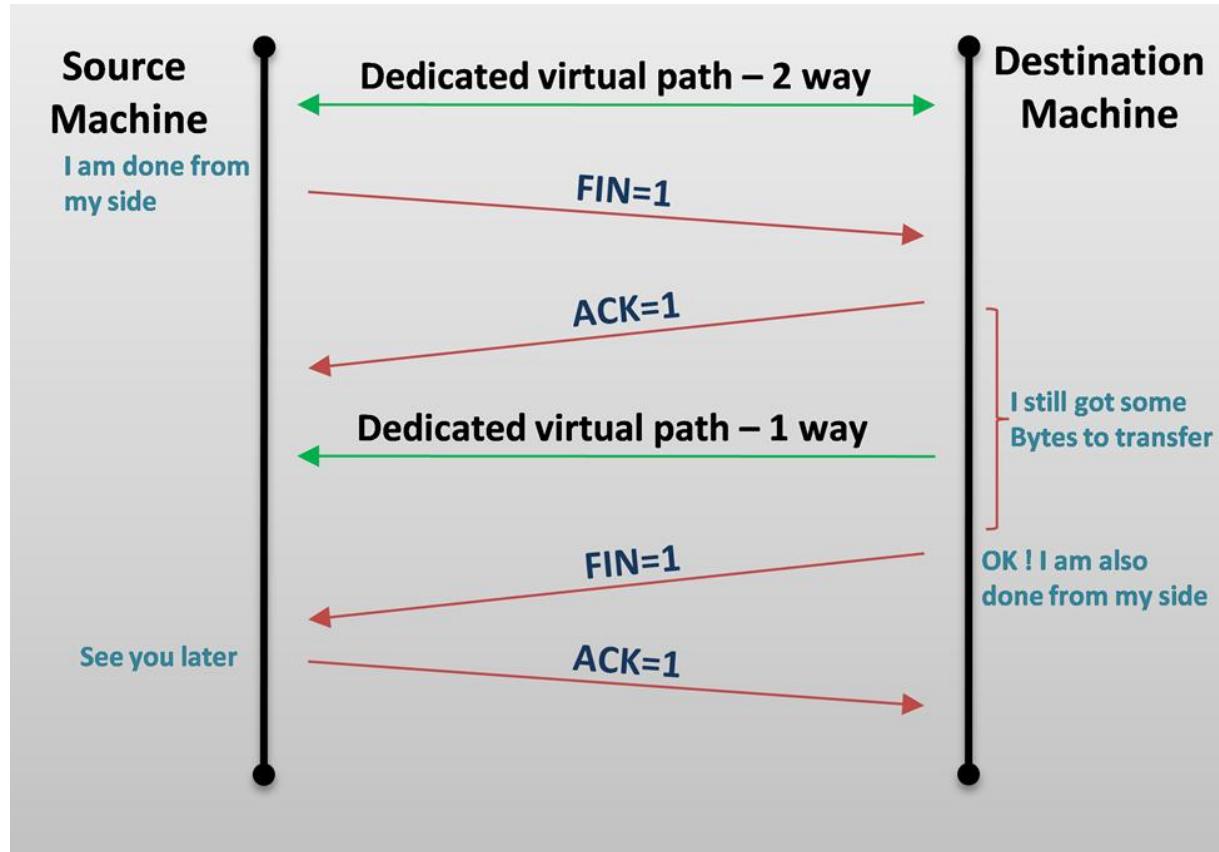
Understanding TCP Communication: Flags and Session Management

The Transmission Control Protocol (TCP) is a **connection-oriented** protocol, meaning it establishes a reliable session between two hosts before any data is transferred. This reliability is managed by a set of six control flags located within the TCP header. These one-bit flags give instructions to the system and govern the establishment, maintenance, and termination of every TCP connection.

The Role of TCP Control Flags

The six primary flags dictate the state of the connection. Four are used for session management, while two are used for data flow.

- **SYN (Synchronize):** This flag initiates a connection. It's sent in the first packet of a session to notify the destination host that a new sequence number is beginning, effectively asking, "Can we establish a connection?"
- **ACK (Acknowledgement):** This flag confirms the receipt of a packet. When a system successfully receives a packet, it sends a packet back with the ACK flag set, acknowledging the transmission and indicating the next expected sequence number.
- **FIN (Finish):** This flag gracefully terminates a connection. When a host is done sending data, it sends a packet with the FIN flag set to announce that it has no more information to transmit.
- **RST (Reset):** This flag abruptly aborts a connection in response to an error. If something unexpected happens, like a packet arriving for a closed port, a packet with the RST flag is sent to immediately tear down the session. Attackers often leverage this behavior to scan for open ports.
- **PSH (Push):** This flag tells the receiving system to process the buffered data immediately instead of waiting for more. It essentially "pushes" the data to the receiving application, which is useful for preventing buffer deadlocks.
- **URG (Urgent):** This flag indicates that certain data within the packet is urgent and should be prioritized. The system is instructed to process this urgent data as soon as possible, pausing other data processing.



TCP /IP Session Management

TCP's reliability is best illustrated by its session establishment and termination processes.

Session Establishment: The Three-Way Handshake

Before data can be exchanged, a stable connection must be created. This is accomplished through a process known as the **three-way handshake**. The mechanism is comparable to a phone call, where one person dials, the other answers, and the first person acknowledges they've heard the "hello" before starting the conversation.

1. **SYN:** The client (source) initiates the connection by sending a TCP packet with the **SYN** flag set to the server (destination). This is like saying, "Hello, I'd like to talk on this port. Are you open?"
2. **SYN/ACK:** If the server is listening on that port, it responds with a packet that has both the **SYN** and **ACK** flags set. This acknowledges the client's request and also establishes its own sequence number. It's the equivalent of, "Yes, I'm open! I acknowledge your request."

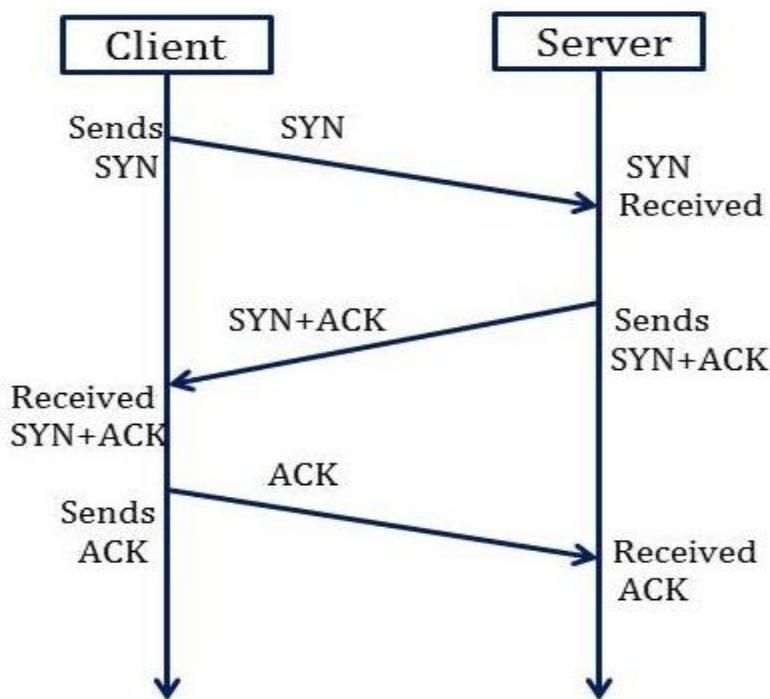
3. **ACK:** Finally, the client sends a packet with the **ACK** flag set back to the server. This confirms that the connection is mutually established. It means, "Great, I've received your confirmation. Let's talk."

Once this final ACK is received, the connection is considered "OPEN," and data transfer can begin.

Session Termination

When the data transfer is complete, the session is closed in a similar, deliberate manner:

1. The client (or server) that is finished sending data sends a packet with the **FIN** flag set.
2. The other host acknowledges this request with an **ACK** packet.
3. This host then sends its own **FIN** packet, signaling it is also ready to close the connection.
4. The original host sends a final **ACK** packet to confirm the termination request, and the connection is closed.



Perform Host Discovery

Host discovery is all about figuring out which systems are actually alive in the target network. Before diving into deeper scans, an ethical hacker needs to map the network surface and confirm which IPs respond to probes.

Lab Scenario

During a security assessment, the very first step is identifying active devices within the target IP range. This helps avoid blind scanning and ensures only valid, reachable hosts are analyzed further. Using Nmap, you can quickly verify which systems are up and responding.

Lab Objective

- Perform host discovery using **Nmap** as part of the **Initial Reconnaissance (IR)** phase.

Perform Host Discovery using Nmap

Network scanning is a process used in computer networks to identify active devices, services, and potential vulnerabilities within a network. It involves sending data packets to target systems and analyzing their responses to gather information such as IP addresses, open ports, running services, and security risks.

Objectives –

- Discovery
- Security assessment
- Vulnerability detection
- Network mapping
- Performance analysis

Common Usage Command –

- -PR = ARP Ping Scan.
 - -PE = ICMP Echo Scan.
 - -PU = UDP Ping Scan.
 - -PP = ICMP Timestamp Ping Scan.
 - -PS = TCP SYN Scan.
 - -PO = IP Protocol Scan.
 - -PM = ICMP Address Mask Scan.
- sn → No port scan, just host discovery.

-v (Verbose Mode):

Makes Nmap show more details during the scan, giving real-time progress and extra info

Command – nmap -sS -PR <target ip>

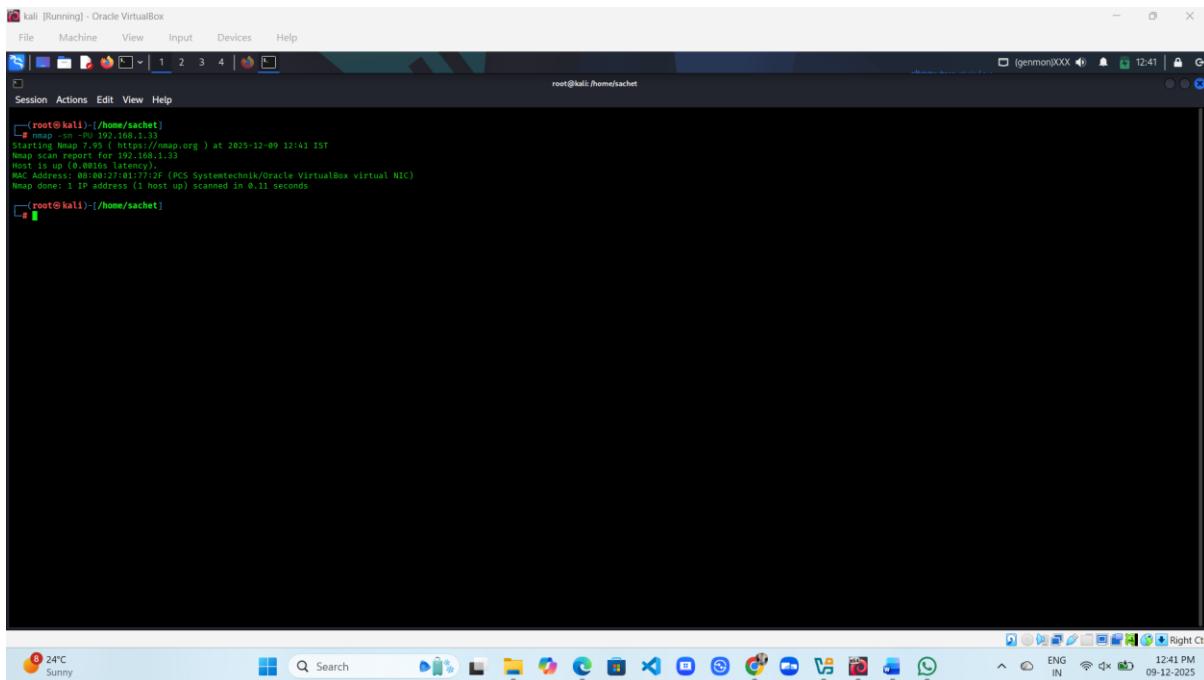
It does a **stealth SYN scan** (-sS) while using **ARP ping** (-PR) to find live hosts on a local network before scanning ports.

```
sachet@kali:~$ nmap -sS -PR 192.168.1.33
Starting Nmap 7.99 ( https://nmap.org ) at 2025-12-09 12:29 IST
Nmap scan pending on 192.168.1.33
Host is up (0.0077s latency).
All 10000 scanned ports on 192.168.1.33 are in [closed] states.
Nmap done: 1 IP address (1 host up) scanned in 21.75 seconds
sachet@kali:~$
```

MODULE – 3 SCANNING NETWORKS

Command – nmap -sn -PU <target ip>

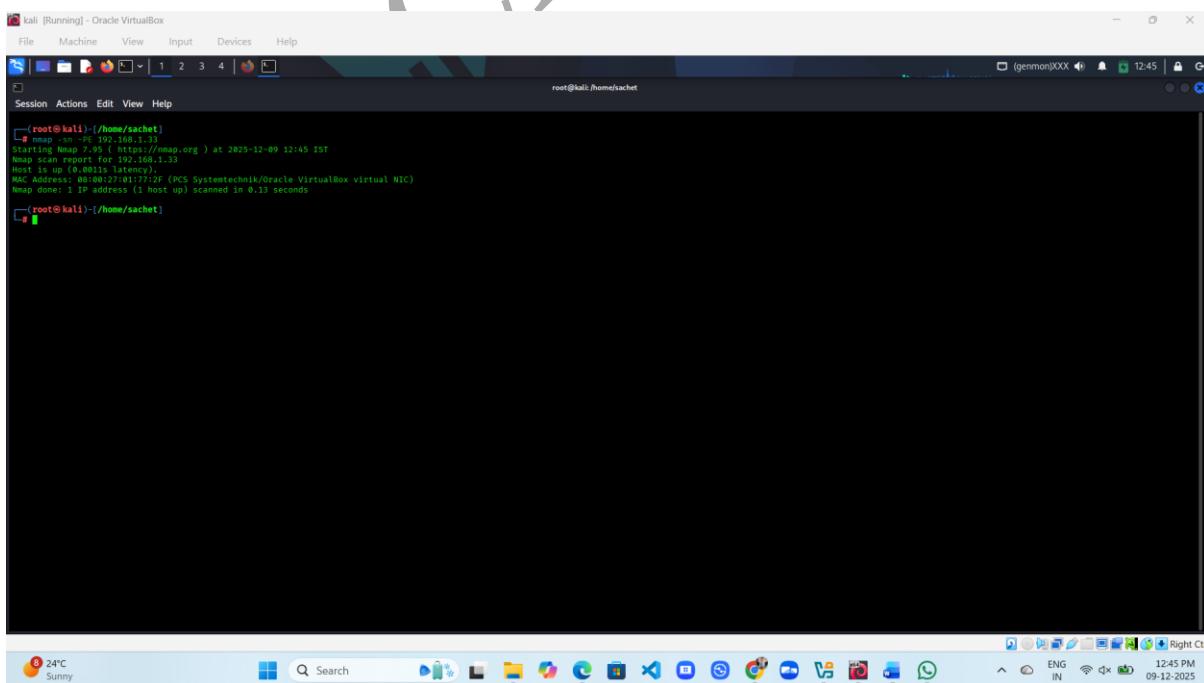
A quick **host discovery scan** that uses **UDP ping** to check if the target device is alive **without doing any port scanning**.



```
(root@kali)-[~/home/sachet]
# nmap -sn -PU 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:41 IST
Nmap scan report for 192.168.1.33
Host is up (0.0001s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.11 seconds
# [root@kali)-[~/home/sachet]
```

Command – nmap -sn -PE <target ip>

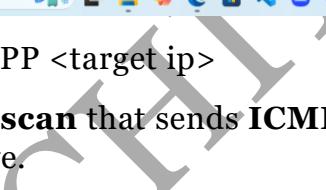
A **host discovery–only scan** that uses **ICMP Echo Request (normal ping)** to check whether the target is alive.



```
(root@kali)-[~/home/sachet]
# nmap -sn -PE 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:45 IST
Nmap scan report for 192.168.1.33
Host is up (0.0001s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
# [root@kali)-[~/home/sachet]
```

MODULE – 3 SCANNING NETWORKS

A **host-discovery-only scan** that uses **ICMP Echo Requests (normal ping)** to check which hosts are alive **from IP .33 to .45** in that range.



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~/home/sachet]
# nmap -sn -PP 192.168.1.33-45
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:55 IST
Nmap scan report for 192.168.1.33
Host is up (0.0017s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds

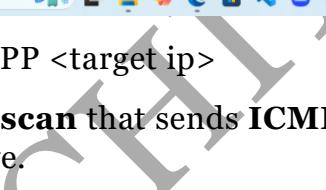
[root@kali]:~/home/sachet]
# map -sn -PP 192.168.1.33-45
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:59 IST
Failed to resolve "192.168.1.33-25".
Nmap done: 0 IP addresses (0 hosts up) so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.21 seconds

[root@kali]:~/home/sachet]
# map -sn -PP 192.168.1.33-45
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 13:00 IST
Nmap scan report for 192.168.1.33
Host is up (0.0017s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.1.35
Host is up (0.0017s latency).
MAC Address: 08:00:27:01:77:2F (Unknown)
Nmap scan report for 192.168.1.36
Host is up (0.11s latency).
MAC Address: 08:00:27:01:77:2F (Unknown)
Nmap scan report for 192.168.1.38
Host is up (0.041s latency).
MAC Address: 08:00:27:01:77:2F (Intel Corporate)
Nmap scan report for 192.168.1.40
Host is up (0.007s latency).
MAC Address: 48:17:D4:5D:0D:35 (Azurwave Technology)
Nmap scan report for 192.168.1.43
Host is up (0.048s latency).
MAC Address: 48:1A:72:07:54:30 (Intel Corporate)
Nmap scan report for 192.168.1.44
Host is up (0.008s latency).
MAC Address: A8:E2:91:01:DF:AA4A (Unknown)
Nmap done: 13 IP addresses (7 hosts up) scanned in 1.73 seconds

[root@kali]:~/home/sachet]
```

Command – `nmap -sn -PP <target ip>`

A **host-discovery-only scan** that sends **ICMP Timestamp Requests** to check if the target system is alive.



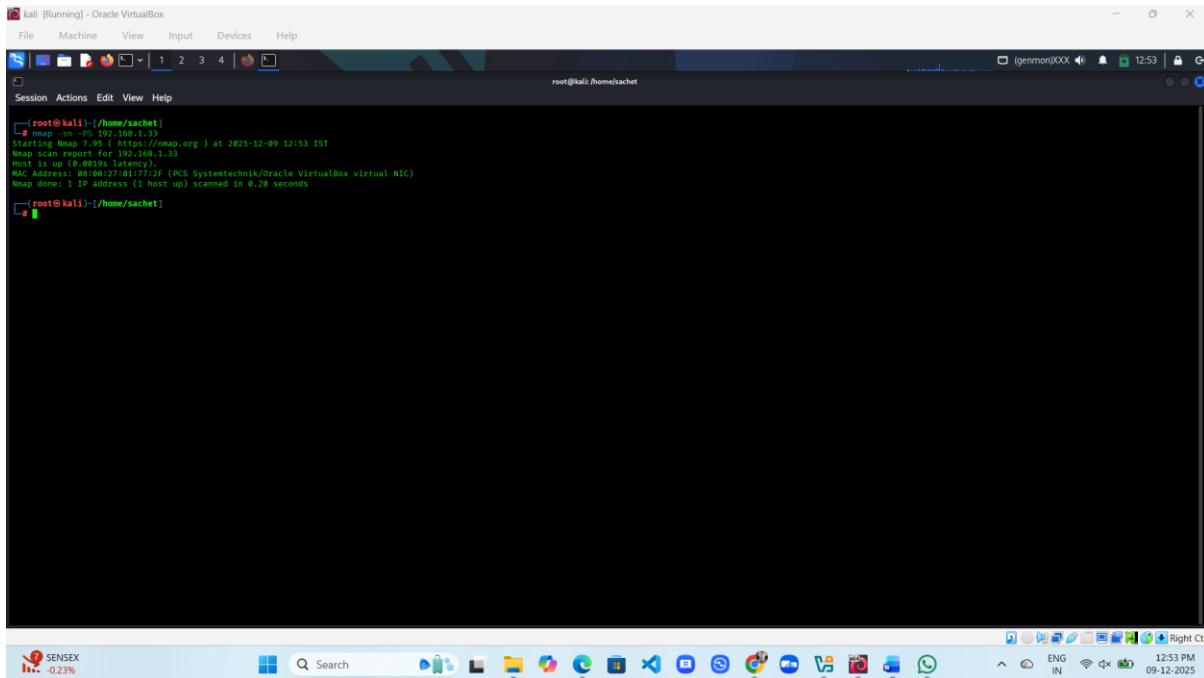
```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~/home/sachet]
# nmap -sn -TP 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:48 IST
Nmap scan report for 192.168.1.33
Host is up (0.0007s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.19 seconds

[root@kali]:~/home/sachet]
```

MODULE – 3 SCANNING NETWORKS

Command – nmap -sn -PS <target ip>

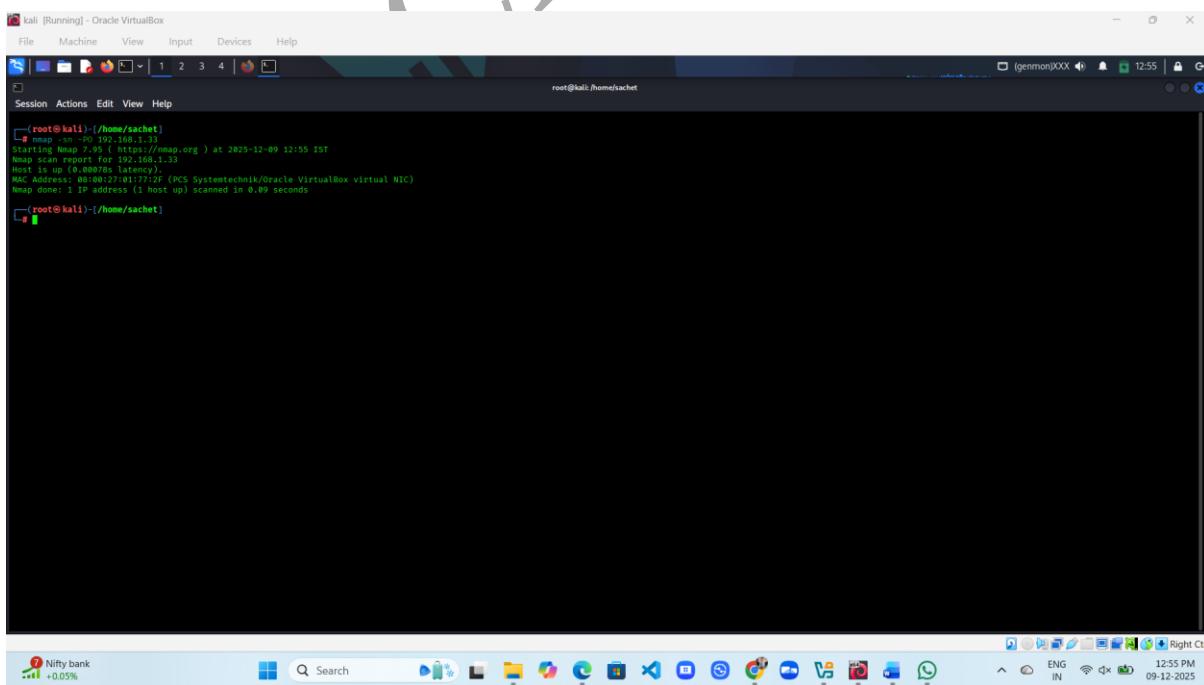
A **host-discovery-only check** that sends **TCP SYN probes** to see if the target wakes up and responds.



```
(root@kali)-[~/home/sachet]
# nmap -sn -PS 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:53 IST
Nmap scan report for 192.168.1.33
Host is up (0.0009s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
# [root@kali)-[~/home/sachet]
```

Command – nmap -sn -PO <target ip>

A **host-discovery-only scan** that sends **IP protocol packets** (like ICMP, IGMP, etc.) to check whether the target device is alive.

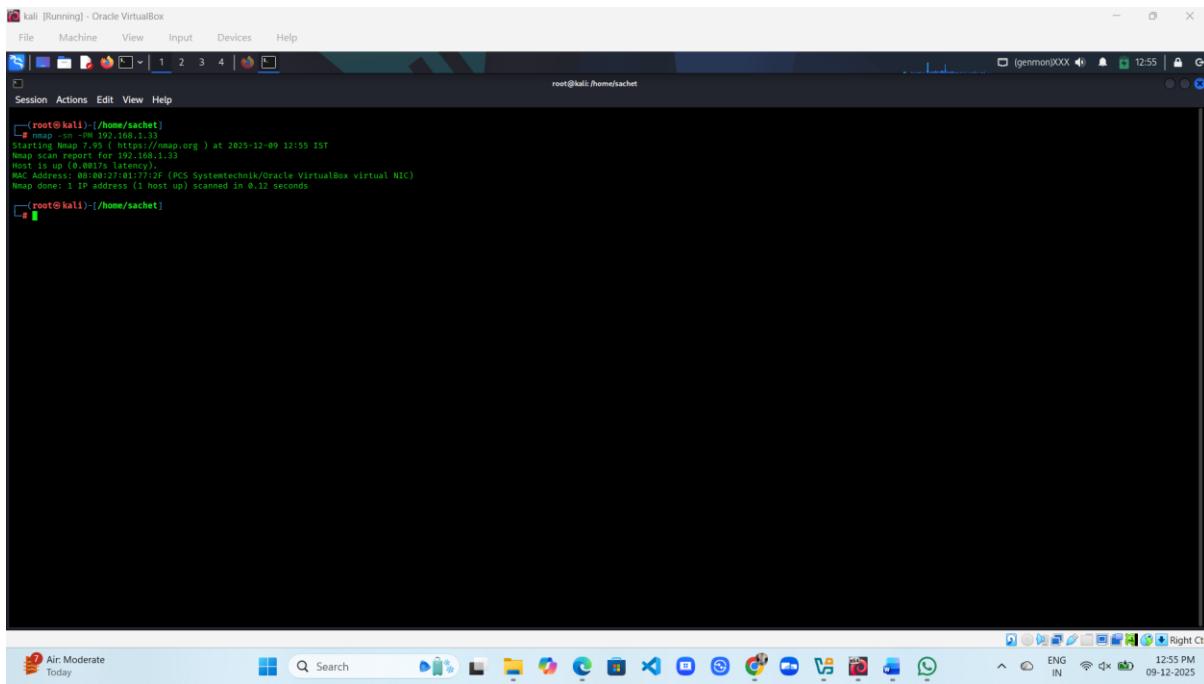


```
(root@kali)-[~/home/sachet]
# nmap -sn -PO 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:55 IST
Nmap scan report for 192.168.1.33
Host is up (0.00078s latency).
MAC Address: 08:00:27:01:77:2F (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
# [root@kali)-[~/home/sachet]
```

MODULE – 3 SCANNING NETWORKS

Command – nmap -sn -PM <target ip>

A **host-discovery-only scan** that sends **IP protocol packets** (like ICMP, IGMP, etc.) to check whether the target is alive even if normal ping is blocked.



The screenshot shows a terminal window titled "kali [Running] - Oracle VirtualBox". The command entered was "nmap -sn -PM 192.168.1.33". The output indicates that the host is up and has an interface named "eth0". The scan report was generated at 2025-12-09 12:55 IST. The total duration of the scan was 0.12 seconds. The terminal window is located on a desktop environment with a taskbar at the bottom containing various application icons.

```
(root@kali:~/home/sachet)
[+] Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 12:55 IST
Nmap scan report for 192.168.1.33
Host is up (0.0017s latency).
MAC Address: 00:0C:29:7A:7F:DF (PC Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
[+] (root@kali:~/home/sachet]
```

SACHCHI'

Perform Port and Service Discovery

Port and service discovery is the process of identifying open ports and services running on the target IP addresses/active hosts.

Lab Scenario

As a professional ethical hacker or a pen tester, the next step after discovering active hosts in the target network is to scan for open ports and services running on the target IP addresses in the target network. This discovery of open ports and services can be performed via various port scanning tools and techniques.

Lab Objectives

Explore various network scanning techniques using Nmap

Overview of Port and Service Discovery

Port scanning tells you which ports are open and what services are running, using different TCP techniques to poke at a system without being too loud.

1. TCP Scanning

- **Open scans:** Full TCP connect.
- **Stealth scans:** Half-open (SYN), Inverse TCP, ACK probes, spoofed/third-party scans.

2. UDP Scanning

- Checks for open UDP ports.

3. SCTP Scanning

- **INIT scan:** Starts SCTP association.
- **COOKIE/ECHO scan:** Verifies SCTP endpoints.

4. Other Scans

- **SSDP & List Scanning:** Checks for devices/services on network.
- **IPv6 Scanning:** Scans hosts using IPv6 addresses.

Common Nmap Commands & Usage

-sS = SYN (stealth, fast, root needed)
-sT = TCP connect (no root, noisy)
-sN = Null scan (no flags)
-sF = FIN scan (FIN flag only)
-sX = Xmas scan (FIN+PSH+URG)
-sA = ACK scan (firewall mapping)
-sO = IP protocol scan (not ports)
-sU = UDP scan (slow, ambiguous)
-sM = Maimon Scan (Uses FIN/ACK packets to quietly identify port states.)

Host discovery: -PE, -PP, -PM, -PR, -PU, -PS, -PO

Explore Various Network Scanning Techniques using Nmap

1. On the Windows 11 machine, log in using Admin / Pa\$\$wOrd. Open Windows Search, type Zenmap, and launch the app.
2. When Zenmap opens, go to the Command field and enter:

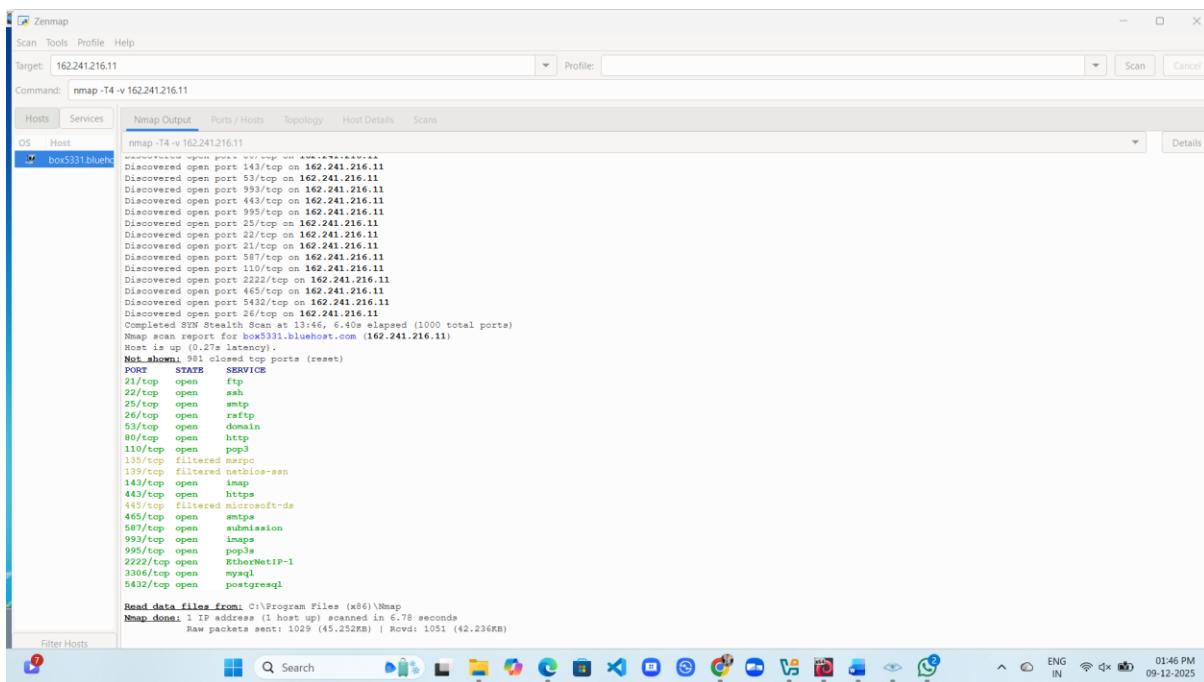
```
nmap -sT -v <target ip>
```

Then click Scan.

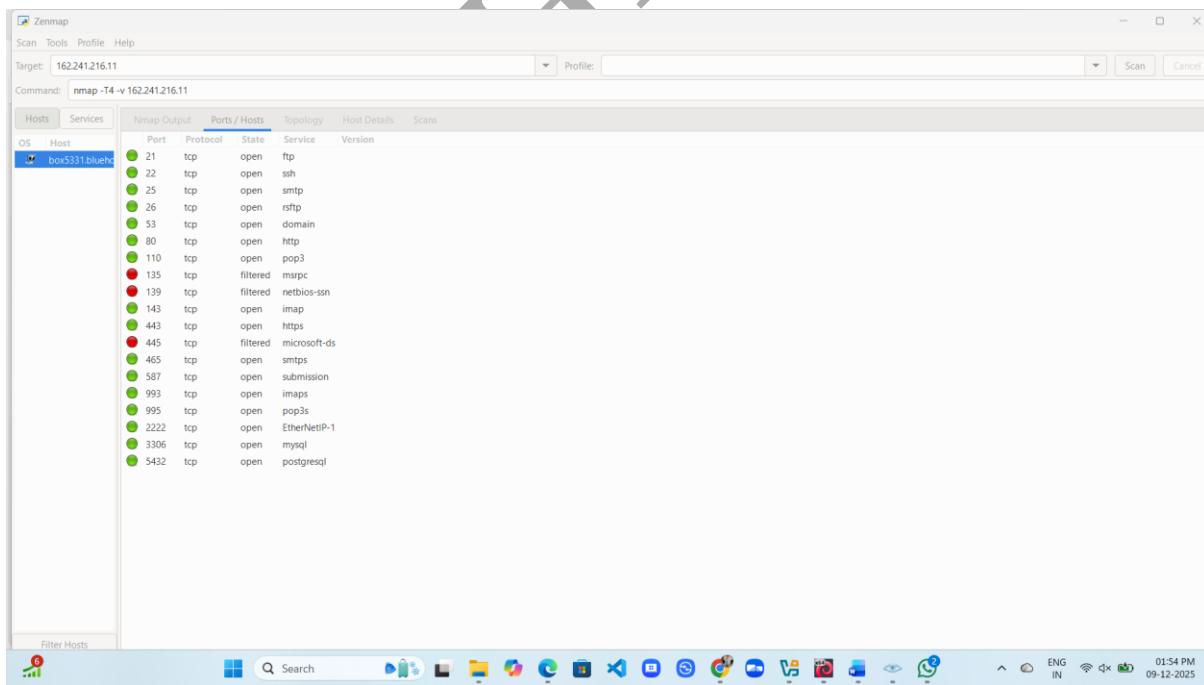
Notes:

- -sT → TCP Connect / Full Open Scan (uses full three-way handshake).
- -v → Verbose mode (shows detailed output like all hosts and ports).

MODULE – 3 SCANNING NETWORKS

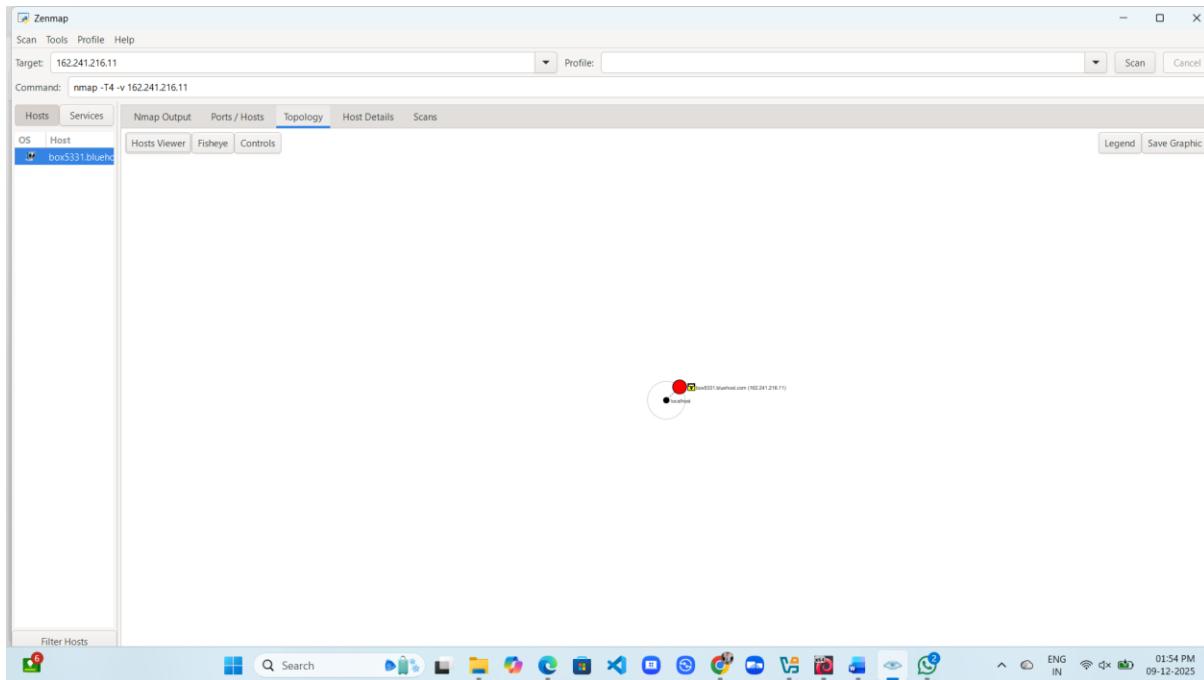


Click the **Ports/Hosts** tab to gather more information on the scan results. Nmap displays the Port, Protocol, State, Service, and Version of the scan.

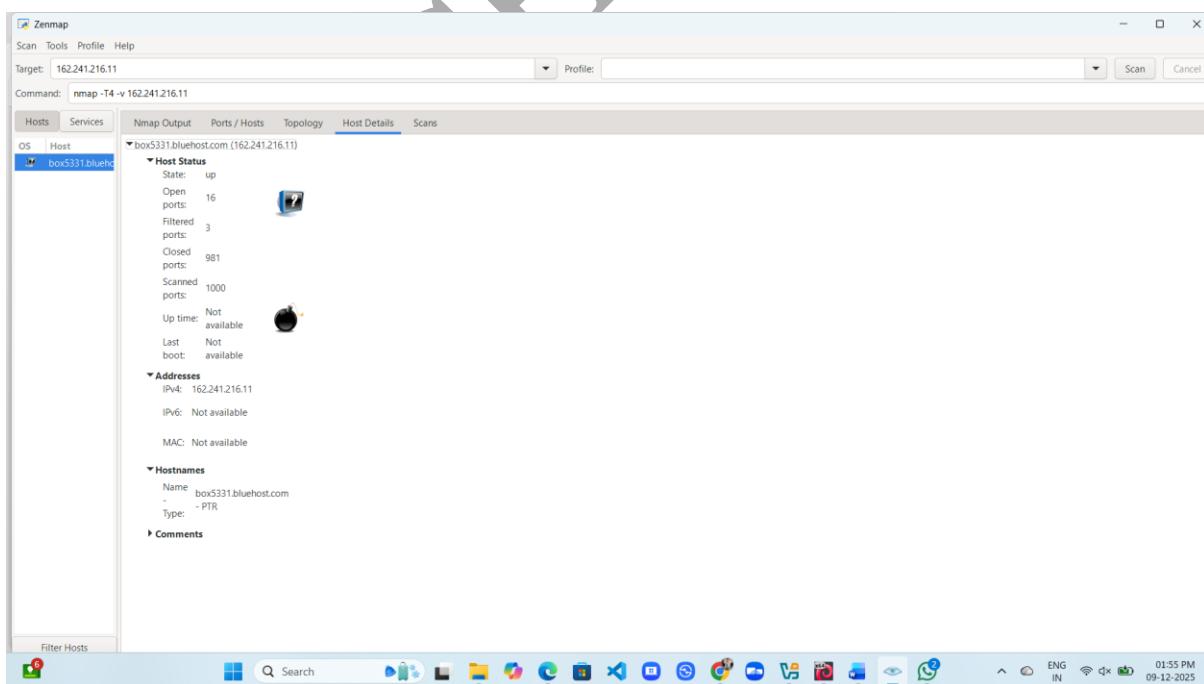


MODULE – 3 SCANNING NETWORKS

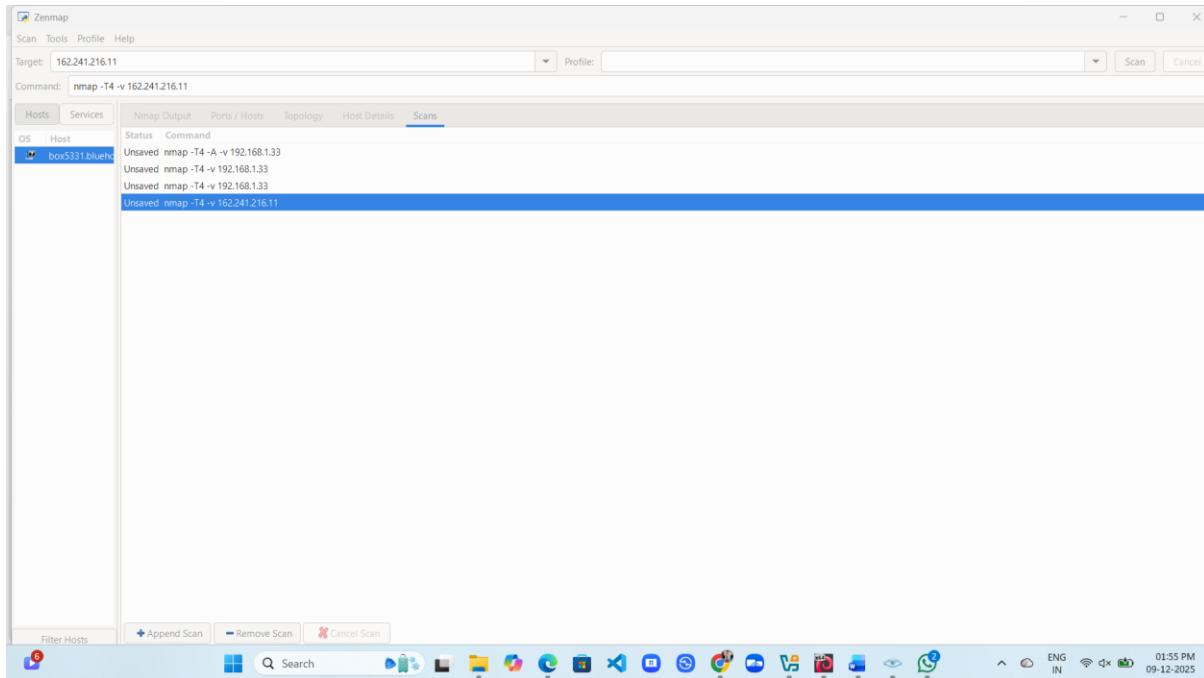
Click the Topology tab to view the topology of the target network that contains the provided IP address and click the Fisheye option to view the topology clearly.



In the same way, click the **Host Details** tab to view the details of the TCP connect scan.



MODULE – 3 SCANNING NETWORKS



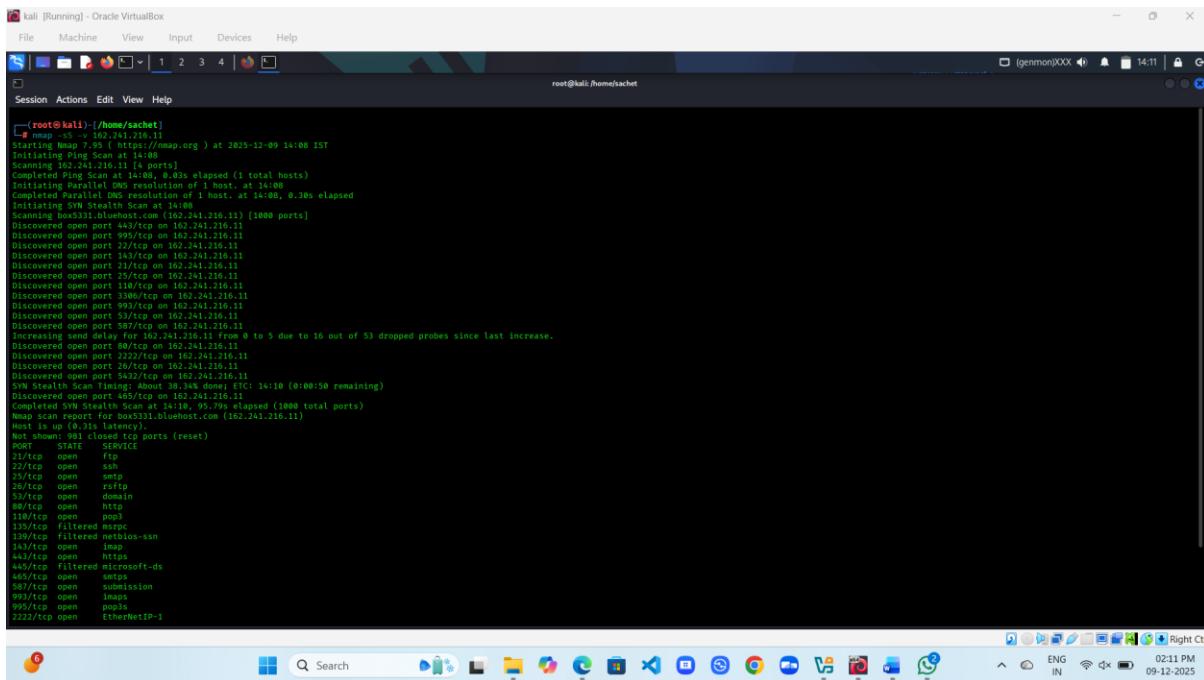
Nmap Command-Line Techniques for Port Scanning

Command- nmap -sS -v <target ip>

Use:

Runs a **SYN (stealth) scan** with **verbose output** – fast, quiet, and the classic move when you've got root access.

MODULE – 3 SCANNING NETWORKS

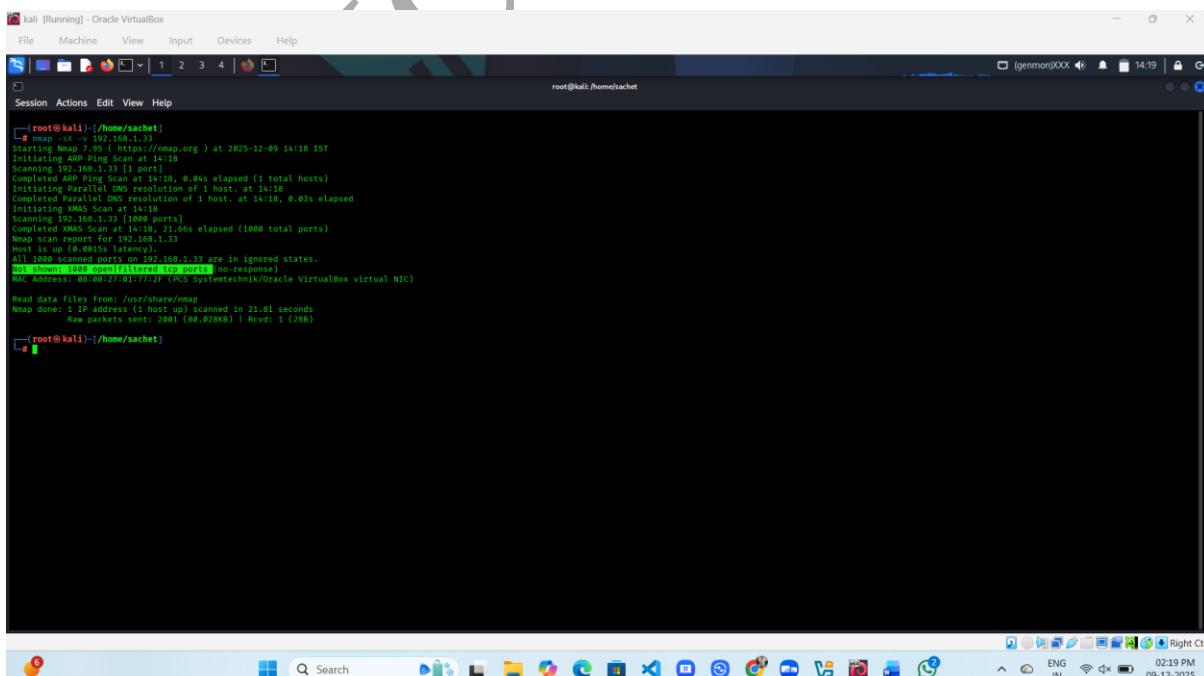


```
[root@kali:~/home/sachet]
# nmap -sX -v 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 14:08 IST
Initiating ARP Ping Scan at 14:08
Completed ARP Ping Scan at 14:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 14:08
Completed DNS resolution at 14:08, 0.00s elapsed (1 host. at 14:08, 0.00s elapsed
Initiating SYN Stealth Scan at 14:08
Scanning box331.bluehost.com (162.241.216.11) (1000 ports)
Discovered open port 80/tcp on 162.241.216.11
Discovered open port 22/tcp on 162.241.216.11
Discovered open port 23/tcp on 162.241.216.11
Discovered open port 38/tcp on 162.241.216.11
Discovered open port 42/tcp on 162.241.216.11
Discovered open port 53/tcp on 162.241.216.11
Discovered open port 587/tcp on 162.241.216.11
Tuning attack rate to 5 due to 16 out of 53 dropped probes since last increase.
Discovered open port 80/tcp on 162.241.216.11
Discovered open port 222/tcp on 162.241.216.11
Discovered open port 200/tcp on 162.241.216.11
Discovered open port 432/tcp on 162.241.216.11
SYN Stealth Scan Timing: About 38.34% done; ETC: 14:10 (0:00:50 remaining)
Discovered open port 465/tcp on 162.241.216.11
Completed SYN Stealth Scan at 14:10, 19.00s elapsed (1000 total ports)
Nmap scan report for box331.bluehost.com (162.241.216.11)
Host is up (0.31s latency).
Not shown: closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
22/tcp    open  sftp
23/tcp    open  telnet
26/tcp    open  rsync
53/tcp    open  domain
70/tcp    open  ps2ps
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
423/tcp   open  http
443/tcp   open  https
445/tcp   filtered microsoft-ds
580/tcp   open  submission
587/tcp   open  imaps
993/tcp   open  pop3s
2223/tcp  open  EtherNetIP-1
```

Command- nmap -sX -v <target ip>

Use:

Runs an **XMAS** scan with **verbose output** — sends FIN+PSH+URG flags to sniff out open/filtered ports. Old trick, still useful when you want to slip under basic filters.



```
[root@kali:~/home/sachet]
# nmap -sX -v 192.168.1.17
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 14:18 IST
Initiating ARP Ping Scan at 14:18
Completed ARP Ping Scan at 14:18, 0.00s elapsed (1 total hosts)
Initiating XMAS Scan at 14:18
Completed XMAS Scan at 14:18, 21.00s elapsed (1000 total ports)
Nmap scan report for 192.168.1.17
Host is up (0.0005s latency).
All 1000 scanned ports on 192.168.1.17 are in ignored states.
Nmap done: 1 IP address (1 host up) scanned in 21.81 seconds
Raw packets sent: 2081 (80.628KB) | Rcvd: 3 (20B)
[root@kali:~/home/sachet]
```

Command- nmap -sM -v <target ip>

Use:

Runs a **Maimon scan** with **verbose output** – a quiet, quirky technique using FIN/ACK to spot port states.

```
(root@kali:~/home/sachet)
# nmap -sM -v 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 14:20 IST
Initiating ARP Ping Scan at 14:20. 0.05s elapsed (1 total hosts)
Scanned 1 IP address (1 host up)
Completed ARP Ping Scan at 14:20. 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host at 14:20. 0.05s elapsed
Initiating ACK Scan at 14:20. 0.05s elapsed
Scanning 192.168.1.33 (1000 ports)
Completed Maimon Scan at 14:21. 21.56s elapsed (1000 total ports)
Nmap done: 1 IP address (1 host up) scanned in 21.74 seconds
Host is up (0.0018s latency).
All 1000 scanned ports on 192.168.1.33 are in ignored states.
Not shown: 1000 filtered open ports (no-response)
Nmap scan report for 192.168.1.33
Host is up (0.0018s latency).
MAC Address: 08:00:27:01:77:2F (PC Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 21.74 seconds
Raw packets sent: 2001 (80.028KB) | Rcvd: 1 (208)

(root@kali:~/home/sachet)
```

Command- nmap -sA -v <target ip>

Use:

Runs an **ACK scan** with **verbose output** – perfect for checking how a firewall handles packets. Old-school recon, but still hits hard.

```
(root@kali:~/home/sachet)
# nmap -sA -v 192.168.1.33
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 14:21 IST
Initiating ARP Ping Scan at 14:21. 0.05s elapsed (1 total hosts)
Scanned 1 IP address (1 host up)
Completed ARP Ping Scan at 14:21. 0.05s elapsed (1 total hosts)
Initiating ACK Scan at 14:21. 0.05s elapsed
Scanning 192.168.1.33 (1000 ports)
Completed ACK Scan at 14:22. 21.53s elapsed (1000 total ports)
Nmap scan report for 192.168.1.33
Host is up (0.0018s latency).
All 1000 scanned ports on 192.168.1.33 are in ignored states.
Not shown: 1000 filtered open ports (no-response)
Nmap scan report for 192.168.1.33
Host is up (0.0018s latency).
MAC Address: 08:00:27:01:77:2F (PC Systemtechnik/Oracle VirtualBox virtual NIC)

Read data files from: /usr/share/nmap
Nmap done: 1 IP address (1 host up) scanned in 21.73 seconds
Raw packets sent: 2001 (80.028KB) | Rcvd: 1 (208)

(root@kali:~/home/sachet)
```

MODULE – 3 SCANNING NETWORKS

Command- nmap -sU -v <target ip>

Use: Runs a **UDP port scan** with **verbose details** – slower but essential for finding UDP services.

```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:[/home/sachet]
└─# nmap -sU -oN /tmp/nmap.org 162.241.216.11
Starting Nmap 7.7.0 ( https://nmap.org ) at 2025-12-09 14:29 IST
Initiating Ping Scan at 14:29
Scanning 162.241.216.11 [4 ports]
Completed Ping Scan at 14:29, 0.31s elapsed (1 total hosts)
Detailed IP-level NSE scan of 1 host at 14:29
Completed Parallel DNS resolution of 1 host at 14:29, 0.00s elapsed
Initiating UDP Scan at 14:29
Completed UDP Scan at 14:29, 0.00s elapsed
Increasing send delay for 162.241.216.11 [1000 parts]
Increasing send delay for 162.241.216.11 From 0 to 50 due to max_successful_trnys increase to 4
Increasing send delay for 162.241.216.11 From 50 to 100 due to max_successful_trnys increase to 5
Increasing send delay for 162.241.216.11 From 100 to 200 due to max_successful_trnys increase to 6
Increasing send delay for 162.241.216.11 From 200 to 400 due to max_successful_trnys increase to 7
Increasing send delay for 162.241.216.11 From 400 to 800 due to max_successful_trnys increase to 8
Increasing send delay for 162.241.216.11 From 800 to 1600 due to max_successful_trnys increase to 9
Increasing send delay for 162.241.216.11 From 1600 to 3200 due to max_successful_trnys increase to 10
Increasing send delay for 162.241.216.11 From 3200 to 6400 due to max_successful_trnys increase to 11
Increasing send delay for 162.241.216.11 From 6400 to 12800 due to max_successful_trnys increase to 12
Increasing send delay for 162.241.216.11 From 12800 to 25600 due to max_successful_trnys increase to 13
Increasing send delay for 162.241.216.11 From 25600 to 51200 due to max_successful_trnys increase to 14
Increasing send delay for 162.241.216.11 From 51200 to 102400 due to max_successful_trnys increase to 15
Increasing send delay for 162.241.216.11 From 102400 to 204800 due to max_successful_trnys increase to 16
Increasing send delay for 162.241.216.11 From 204800 to 409600 due to max_successful_trnys increase to 17
Increasing send delay for 162.241.216.11 From 409600 to 819200 due to max_successful_trnys increase to 18
Increasing send delay for 162.241.216.11 From 819200 to 1638400 due to max_successful_trnys increase to 19
Increasing send delay for 162.241.216.11 From 1638400 to 3276800 due to max_successful_trnys increase to 20
Increasing send delay for 162.241.216.11 From 3276800 to 6553600 due to max_successful_trnys increase to 21
UDP Scan Timing: About 4.50s done; ETC: 16:41 (0:18:58 remaining)
Increasing send delay for 162.241.216.11 From 6553600 to 13107200 due to 11 out of 21 dropped probes since last increase,
UDP Scan Timing: About 11.44s done; ETC: 16:44 (0:13:17 remaining)
UDP Scan Timing: About 23.46s done; ETC: 16:45 (0:12:38 remaining)
UDP Scan Timing: About 26.61s done; ETC: 16:46 (0:12:39 remaining)
Stats: 0 bytes sent, 0 received (1 ms= 1 packet)
UDP Scan Timing: About 34.53s done; ETC: 16:47 (0:10:56 remaining)
Stats: 0 bytes sent, 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 46.09s done; ETC: 16:48 (0:10:01 remaining)
UDP Scan Timing: About 46.09s done; ETC: 16:48 (0:09:07 remaining)
Discovered open port 53/tcp on 162.241.216.11
UDP Scan Timing: About 46.62s done; ETC: 16:48 (0:10:01 remaining)
UDP Scan Timing: About 46.62s done; ETC: 16:48 (0:09:07 remaining)
UDP Scan Timing: About 56.60s done; ETC: 16:49 (0:07:22 remaining)
UDP Scan Timing: About 61.90s done; ETC: 16:49 (0:06:28 remaining)
UDP Scan Timing: About 61.90s done; ETC: 16:49 (0:06:28 remaining)
UDP Scan Timing: About 71.96s done; ETC: 16:49 (0:04:45 remaining)
UDP Scan Timing: About 77.44s done; ETC: 16:49 (0:03:51 remaining)
UDP Scan Timing: About 82.73s done; ETC: 16:49 (0:02:56 remaining)
UDP Scan Timing: About 87.84s done; ETC: 16:49 (0:02:04 remaining)

(root@kali:[/home/sachet]
└─#
```

Command- nmap -sT -v <target ip>

Use: Does a **TCP Connect scan** with **verbose output** – loud, obvious, but reliable when you don't have root access.

MODULE – 3 SCANNING NETWORKS

Command - nmap -sN -T4 -A -v 162.241.216.11

Runs a Null scan with **aggressive detection**, **faster timing**, and **verbose output** on the target.

Command- nmap -sV -v <target ip>

Use: Scan open ports + detect service versions with extra details.

Command- nmap -sS -sV <target ip>

Use:

Runs a **SYN (stealth) scan** to find open ports, and then **probes each open port to detect service versions**.

Fast, quiet, and super useful when you want to know *what's open* and *what exactly is running on it*.

```

kali [Running] - OracleVirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
root@kali:~# nmap -sS -sV 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 15:31 IST
Nmap scan report for box5331.bluehost.com (162.241.216.11)
Host is up (0.31s latency).
Not shown: 981 closed tcp ports (reset)
PORT      STATE    SERVICE      VERSION
21/tcp    open     ftp          Pure-FTPD
22/tcp    open     ssh          OpenSSH 8.7 (protocol 2.0)
25/tcp    open     tcpwrapped
26/tcp    open     tcpwrapped
53/tcp    open     domain      ISC BIND 9.16.23 (RedHat Linux)
80/tcp    open     http         Apache httpd
110/tcp   open     pop3        Dovecot pop3d
135/tcp   filtered msrpc
139/tcp   filtered netbios-ssn
143/tcp   open     imap        Dovecot imapsd
443/tcp   open     ssl/http    Apache httpd
445/tcp   filtered microsoft-ds
465/tcp   open     tcpwrapped
587/tcp   open     tcpwrapped
993/tcp   open     ssl/imap    Dovecot imapsd
995/tcp   open     ssl/pop3   Dovecot pop3d
2222/tcp  open     tcpwrapped
3306/tcp  open     mysql       MySQL (blocked - too many connection errors)
5432/tcp  open     postgresql PostgreSQL DB 14.7 - 14.9
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 64.38 seconds

```

Command- nmap -A <target-ip>

Use:

Runs aggressive scanning – OS detection, service versions, scripts, and traceroute all in one hit.

```

root@kali:~# nmap -A 192.168.1.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 14:55 IST
Nmap scan report for 192.168.1.10
Host is up (0.32s latency).
All 1000 scanned ports on 192.168.1.10 are in ignored states.
No service details available for any port (no-response)
MAC Address: WC:D0:24:E9:97:2E (Intel Corporate)
Too many fingerprints match this host to give specific OS details
Network Distances: 1 hop

TRACEROUTE
HOP RTT      ADDRESS
1  932.04 ms 192.168.1.10

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 357.47 seconds

```

Perform OS Discovery

The image you provided outlines a lab scenario focused on **OS Discovery** (also known as banner grabbing or OS fingerprinting).

Overview

- **Goal:** To determine the **Operating System (OS)** running on a remote target system.
- **Context:** As a professional ethical hacker or penetration tester, OS discovery is a critical step **after** identifying open ports and services.
- **Purpose:** Knowing the target's OS allows you to:
 - Assess the system's specific vulnerabilities.
 - Identify potential exploits that might be effective for further attacks.

Lab Objective

- Perform OS discovery using Nmap Script Engine (NSE).
- **-O:** Enable OS detection.
- **--script:** Run Nmap Scripting Engine (NSE) scripts.
- **-A :** Runs aggressive scanning
- **-sV :** service/version detection

Perform OS Discovery using Nmap Script Engine (NSE)

Command- nmap -A <target-ip>

Use:

Runs aggressive scanning – OS detection, service versions, scripts, and traceroute all in one hit.

kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Session Actions Edit View Help

(root@kali:~/home/sachet)

map -A 102.241.216.11

Starting Nmap 7.05 (https://nmap.org) at 2025-12-09 15:10 IST

Host is up (0.11s latency).

Host shows no open ports (reset).

OS: Linux STATE SERVICE VERSION

22/tcp open ftp Pure-FTPd

22/tcp open ssh OpenSSH 8.7 (protocol 2.0)

23/tcp open telnet Exim 4.90

|_ nmap-commands: box5331.bluehost.com [122.170.198.168], SIZE 524288000, LIMITS MAILMAX=50 RCPTMAX=50000, 8BITMIME, PIPELINING, PIPECONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP

_| Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA DOTAT NOOP QUIT RSET HELP

26/tcp open http Apache 2.4.42

|_ nmap-commands: box5331.bluehost.com [122.170.198.168], SIZE 524288000, LIMITS MAILMAX=50 RCPTMAX=50000, 8BITMIME, PIPELINING, PIPECONNECT, AUTH PLAIN LOGIN, STARTTLS, HELP

_| Commands supported: AUTH STARTTLS HELO EHLO MAIL RCPT DATA DOTAT NOOP QUIT RSET HELP

53/tcp open domain ISC BIND 9.16.23 (Redhat Linux)

|_ bind-version: 9.16.23-0H

80/tcp open http Apache Httpd

|_http-title: Did not Follow redirect to https://box5331.bluehost.com/

110/tcp open pop3 Dovecot pop3d

_pop3-capabilities: SASL(PLAIN LOGIN) CAPA USER RESP-CODES STLS TOP AUTH-RESP-CODE UIDL PIPELINING

|_subject-alternative-name: box5331.bluehost.com, DNS:bluehost.com

|_not-valid-before: 2025-01-27T00:00:00

|_not-valid-after: 2026-01-27T23:59:59

|_xslt-date: TLS randomness does not represent time

135/tcp filtered msrpc

139/tcp filtered netbios-ssn

143/tcp open imap Dovecot imapd

|_imap-capabilities: Pre-login DIA AUTH+PLAIN LOGIN+REFERRALS IMAP4rev1 ENABLE NAMESPACE AUTH:LOGIN#0001 have SASL-IR LITERAL+ STARTTLS capabilities listed more OK IDLE

|_ssl-cert: Subject: CommonName=*, bluehost.com

|_subject-alternative-name: box5331.bluehost.com, DNS:bluehost.com

|_not-valid-before: 2025-01-27T00:00:00

|_not-valid-after: 2026-01-27T23:59:59

|_xslt-date: TLS randomness does not represent time

443/tcp open https Apache Httpd

|_http-server-header: Apache

|_http-title: 404 Not Found

|_subject-alternative-name: box5331.bluehost.com, DNS:bluehost.com

|_not-valid-before: 2025-01-27T00:00:00

|_not-valid-after: 2026-01-27T23:59:59

|_xslt-date: TLS randomness does not represent time

455/tcp filtered microsoft-ds

465/tcp open ssl/smtp Exim smtpd 4.98.2

|_nmap-commands: Couldn't establish connection on port 465

587/tcp open smtp Exim smtpd 4.98.2

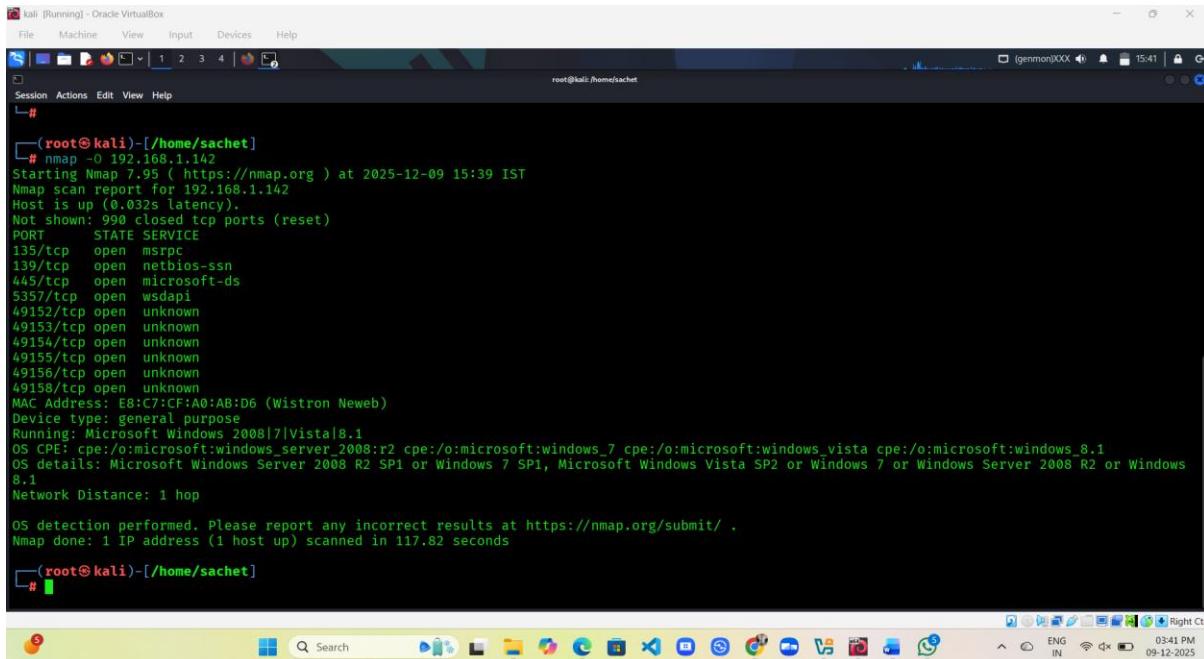
Command- nmap -O <target-ip>

Use:

Runs OS detection, trying to figure out what operating system the target is running.

Think of it like reading someone's vibe from a distance — Nmap checks packet fingerprints, quirks, and responses, then guesses the OS. Not always perfect, but old-school reliable when it hits.

MODULE – 3 SCANNING NETWORKS



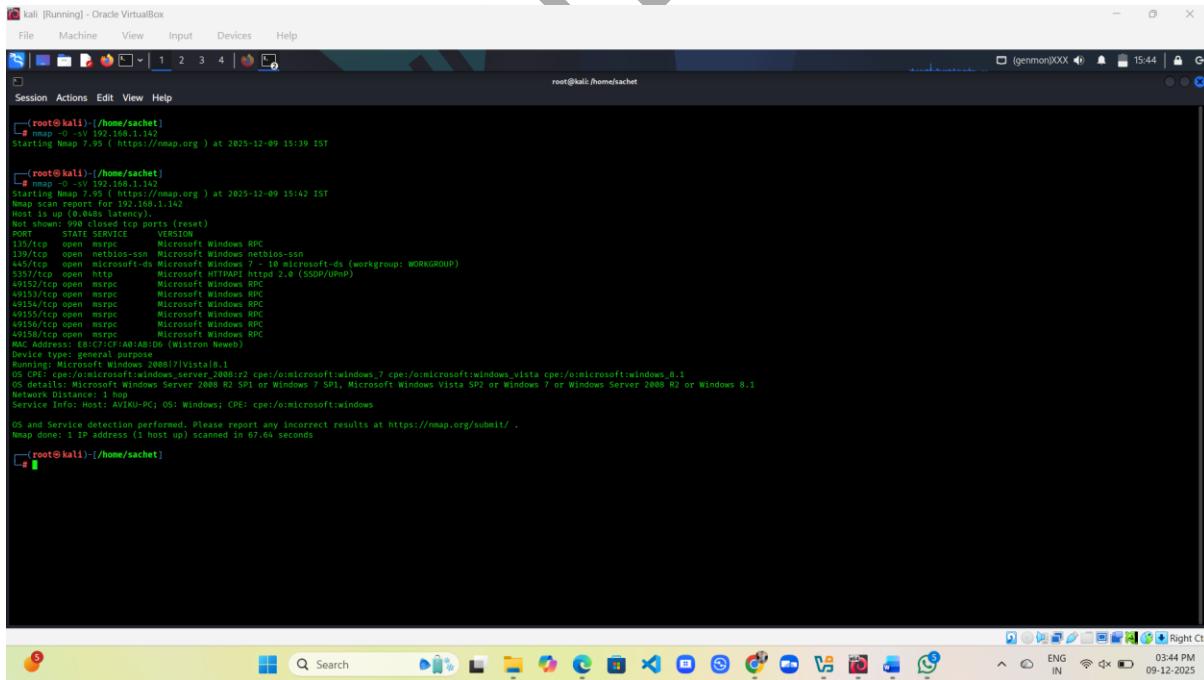
```
# nmap -O -sV 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 15:39 IST
Nmap scan report for 192.168.1.142
Host is up (0.032s latency).
Not shown: 990 closed tcp ports (reset)
PORT      STATE SERVICE
135/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp    open  microsoft-ds
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: E8:C7:CF:A0:AB:D6 (Wistron Neweb)
Device type: general purpose
Running: Microsoft Windows 2008 R2 SP1 or Windows 7 SP1
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Server 2008 R2 SP1 or Windows 7 SP1, Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 117.82 seconds
```

Command- nmap -O -sV <target-ip>

Use:

Runs **OS detection (-O)** plus **service/version detection (-sV)** on the target.



```
[root@kali]~/.home/sachet]
# nmap -O -sV 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 15:42 IST
Nmap scan report for 192.168.1.142
Host is up (0.048s 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 RPC (version: 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: E8:C7:CF:A0:AB:D6 (Wistron Neweb)
Device type: general purpose
Running: Microsoft Windows 2008 R2 SP1 or Windows 7 SP1, Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
OS CPE: cpe:/o:microsoft:windows_server_2008:r2 cpe:/o:microsoft:windows_vista cpe:/o:microsoft:windows_8.1
OS details: Microsoft Windows Server 2008 R2 SP1 or Windows 7 SP1, Microsoft Windows Vista SP2 or Windows 7 or Windows Server 2008 R2 or Windows 8.1
Network Distance: 1 hop
Service Info: Host: AVIKU-PC; OS: Windows; CPE: cpe:/o:microsoft:windows

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 67.64 seconds
```

Command - nmap --script smb-os-discovery.nse 192.168.1.142

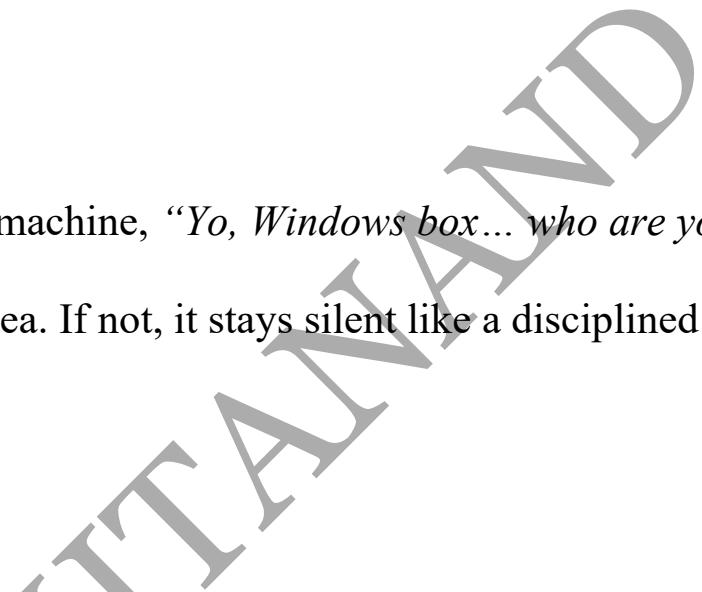
Use:

Runs the **SMB OS Discovery script** against the target.
It uses SMB (Windows file-sharing service) to pull details like:

- Operating system
- Computer name
- Domain/workgroup
- SMB signing status
- System time

Basically, you're asking the machine, “*Yo, Windows box... who are you really?*”

If SMB is open, it spills the tea. If not, it stays silent like a disciplined old server.



```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[root@kali]-[~/home/sachet]
# nmap --script smb-os-discovery.nse 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 15:36 IST
Nmap scan report for 192.168.1.142
Host is up (0.096s 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  wsddapi
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: E8:C7:CF:A0:AB:D6 (Wistron Neweb)

Host script results:
| smb-os-discovery:
|_ OS: Windows 7 Ultimate 7600 (Windows 7 Ultimate 6.1)
|_ OS CPE: cpe:/o:microsoft:windows_7::-
|_ Computer name: aviku-PC
|_ NetBIOS computer name: AVIKU-PC\x00
|_ Workgroup: WORKGROUP\x00
|_ System time: 2025-12-09T15:38:10-08:00

Nmap done: 1 IP address (1 host up) scanned in 22.57 seconds
[root@kali]-[~/home/sachet]
#

```

Scanning Beyond IDS and Firewall

When you're stepping into this phase, you're basically trying to **slip past the security guards** (IDS + firewalls) without setting off alarms. Not to be edgy — just to check how good the defenses actually are. Classic pentester duty, timeless as it gets.

Common Evasion Techniques

Here's the toolbox you usually pull from:

1. Packet Fragmentation

You break your packets into tiny pieces. IDS tries to reassemble them... and sometimes fails. Old trick, but still throws off lazy configurations.

2. Source Routing

You tell packets exactly **which path** to take. Sometimes, you can steer around firewalls like taking a back alley.

3. IP Spoofing

You pretend the packets came from somewhere else. Confuses logs, evades filters — risky but effective in testing.

4. Decoy Scanning

Throw multiple fake source IPs into the scan. IDS sees a crowd and can't tell who the real scanner is.

5. Timing Tweaks

Slow your scan down (`-T0` or `-T1`) so the IDS doesn't notice patterns. Patience becomes a weapon.

6. Using ACK, Null, FIN, or Xmas Scans

These “weird” packet styles sometimes slip past firewall rules designed only to block SYN scans.

Overview: Scanning Beyond IDS and Firewall

Even the sharpest security guards — IDS and firewalls — aren't flawless. They're built to stop unauthorized access, but crafty hackers (or ethical pentesters) can still sneak packets past using clever techniques. The goal here isn't chaos — it's testing defenses to see what can slip through.

Scan beyond IDS/Firewall using Various Evasion Techniques

- **-f:** Fragment packets (for evasion).
- **-g or --source-port:** Spoof source port (e.g., port 53).
- **--mtu:** Specify the Maximum Transmission Unit size.
- **-D or --decoy:** Decoy scanning (obfuscate source IP).

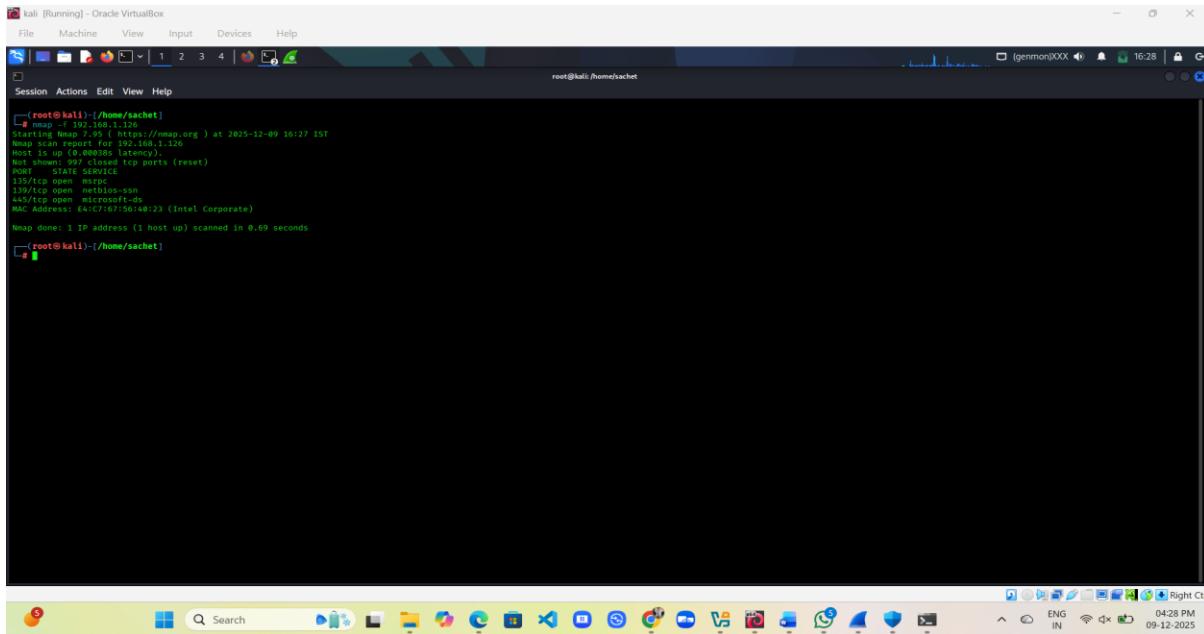
Command- nmap -f 192.168.1.126

Use:

Runs a scan using packet fragmentation.

Nmap splits the packets into tiny pieces so IDS/firewalls have a harder time reassembling them and spotting the scan.

MODULE – 3 SCANNING NETWORKS



```
(root@kali)-[~/home/sachet]
# nmap -g 80 192.168.1.126
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:27 IST
Nmap scan report for 192.168.1.126
Host is up (0.0005s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 64:C7:67:56:48:23 (Intel Corporate)

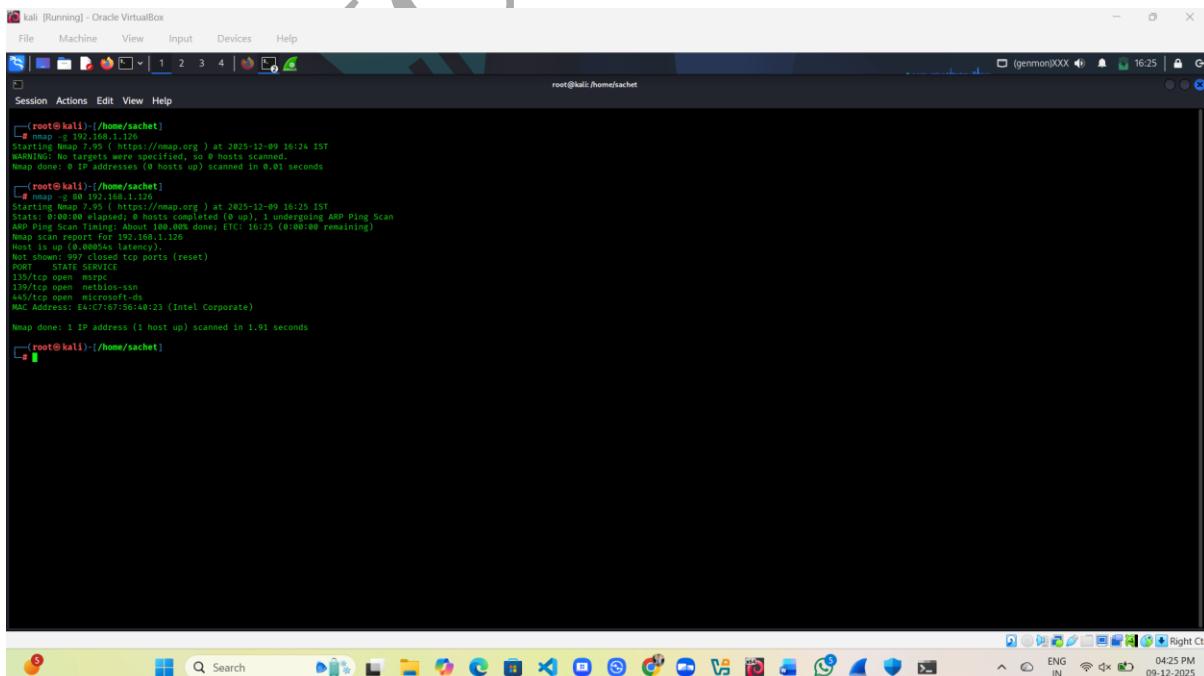
Nmap done: 1 IP address (1 host up) scanned in 0.69 seconds
#
```

Command - nmap -g 80 192.168.1.126

Use: Runs a scan while spoofing the source port as 80.

Port 80 is common web traffic, so weak firewalls often trust it.

This trick helps slip your scan past filters that only block “suspicious” or uncommon ports.



```
(root@kali)-[~/home/sachet]
# nmap -g 192.168.1.126
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:24 IST
Nmap scan report for 192.168.1.126
Host is up (0.0005s latency).
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.02 seconds
#
```

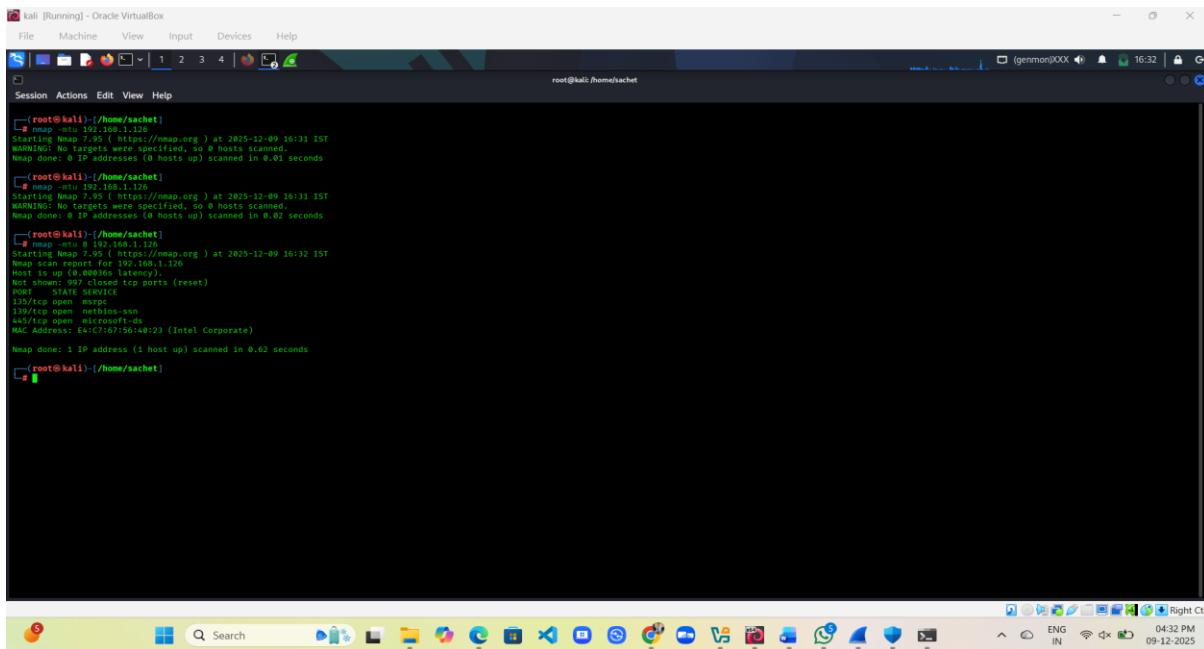
```
(root@kali)-[~/home/sachet]
# nmap -g 192.168.1.126
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:25 IST
Stats: 0:00:00 elapsed; 0 hosts completed (0 up), 1 undergoing ARP Ping Scan
ARP ping scan complete (1 host up) | 1 hosts down | ETC: 16:25 (0:00:00 remaining)
Nmap scan report for 192.168.1.126
Host is up (0.0005s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 64:C7:67:56:48:23 (Intel Corporate)

Nmap done: 1 IP address (1 host up) scanned in 1.91 seconds
#
```

MODULE – 3 SCANNING NETWORKS

Command - nmap -mtu 8 162.241.216.11

Use: Forces Nmap to send packets with an MTU size of 8 bytes, breaking them into tiny, unusual-sized fragments.



```
[root@kali]~# nmap -mtu 8 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:31 IST
Nmap scan report for 162.241.216.11
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds

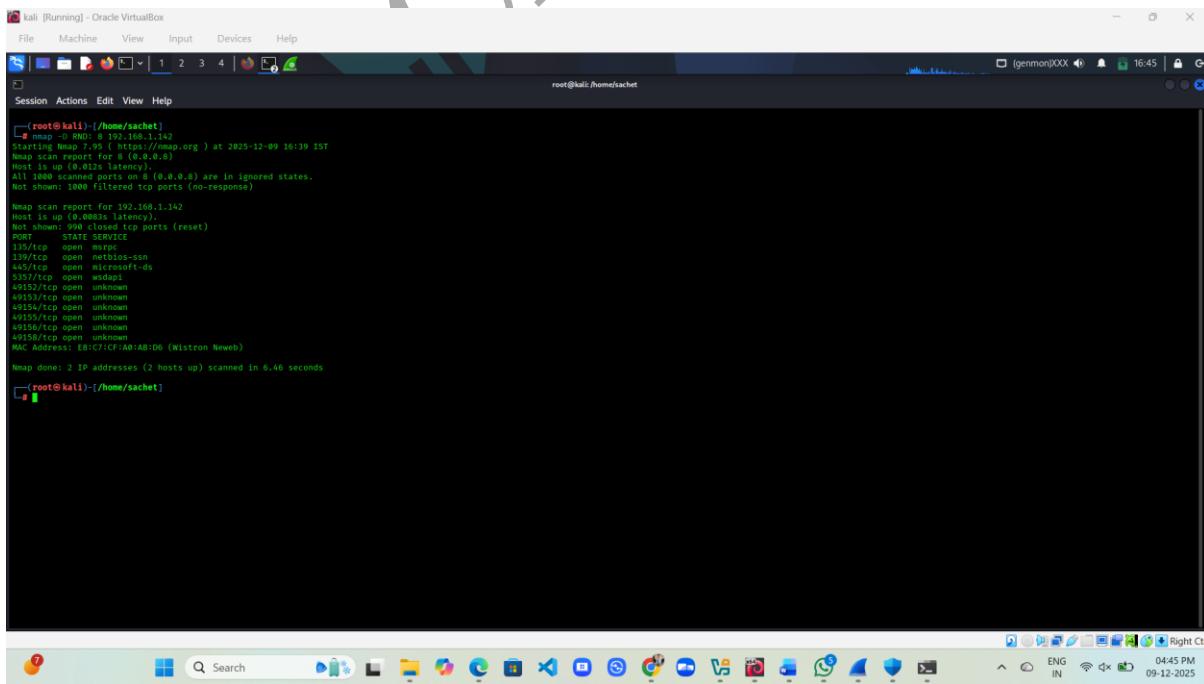
[root@kali]~# nmap -mtu 126 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:31 IST
Nmap scan report for 162.241.216.11
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds

[root@kali]~# nmap -mtu 128 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:31 IST
Nmap scan report for 162.241.216.11
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds

[root@kali]~# nmap -mtu 129 162.241.216.11
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:32 IST
Nmap scan report for 162.241.216.11
Nmap done: 1 IP address (1 host up) scanned in 0.02 seconds
[root@kali]~#
```

Command - nmap -D RND:8 192.168.1.142

→ Uses 8 random decoy IPs to hide your real IP during the scan and confuse IDS/firewall logs.



```
[root@kali]~# nmap -D RND:8 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:39 IST
Nmap scan report for 8 (0.0.0.0)
Host is up (0.000s latency).
All 10000 scanned ports on 8 (0.0.0.0) are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)

Nmap scan report for 192.168.1.142
Host is up (0.000s latency).
Not shown: 1000 closed tcp ports (reset)
PORT      STATE SERVICE
123/tcp   open  snmp
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5353/tcp  open  vidas
5357/tcp  open  vidas
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: 8C:C7:CF:AA:AB:D6 (Wistron Neweb)

Nmap done: 2 IP addresses (2 hosts up) scanned in 0.46 seconds
[root@kali]~#
```

MODULE – 3 SCANNING NETWORKS

Command - nmap -sT -Pn --spoof-mac 0 192.168.1.142

Use:

Runs a TCP Connect scan while skipping host discovery and spoofing a random MAC address from a popular vendor list.

```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help

[+] root@kali:[/home/sachet]
# nmap -sT -Pn --spooftx 192.168.1.142
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-09 16:47 IST
Nmap scan report for 192.168.1.142
Host is up (0.010s latency).
MAC Address: 0A:0D:05:80:32:58 (No registered vendor)
Nmap scan timing rules applied as --min-rate=1000 --max-rate=1000
These options will not be honored for TCP Connect scan.
Nmap scan report for 192.168.1.142
Host is up (0.010s latency).
Nmap scan timing rules applied as --min-rate=1000 --max-rate=1000
PORT      STATE SERVICE
135/tcp    open  msrpc
3389/tcp   open  msTerminalServices
445/tcp    open  microsoft-ds
445/tcp    open  Microsoft-DNS
4457/tcp   open  wsdapi
49152/tcp  open  unknown
49153/tcp  open  unknown
49154/tcp  open  unknown
49155/tcp  open  unknown
49156/tcp  open  unknown
49157/tcp  open  unknown

Nmap done: 1 IP address (1 host up) scanned in 4.82 seconds
[+] root@kali:[/home/sachet]
```

Perform Network Scanning Using Various Scanning Tools

When the usual scans don't cough up enough info, you bring in heavier, smarter tools. The mission stays the same: **dig deeper, map clearer, spot weaknesses**, and do it without getting sloppy.

Lab Scenario

Sometimes the earlier recon is just the tip of the iceberg. Systems hide things, networks play quiet, and you—like any savvy pentester—push further. Other tools help crack open more details when Nmap alone isn't giving the whole picture.

Lab Objective

- **Use Metasploit to scan a target network**
(Metasploit has auxiliary scanners and modules that uncover services, versions, vulnerabilities, and endpoints.)

Overview of Network Scanning Tools

Think of scanning tools as your digital binoculars — they let you peek across the network jungle and spot who's alive, what ports are open, which services are humming, and how machines identify themselves.

These tools spill everything from **active hosts, TCP/UDP ports, running services, NetBIOS details, to location-based info**.

All that intel lets an ethical hacker sketch a sharp, reliable profile of the target org and figure out which doors in the network are wide open, half-open, or pretending to be locked.

Scan a Target Network Using Metasploit

msfconsole

It's the main command-line interface of Metasploit.

You use it to run scanners, exploits, payloads, auxiliary modules — basically the whole hacking toolkit under one roof.

How to use it :-

MODULE – 3 SCANNING NETWORKS

Step 1 : Open terminal and type msfconsole

```
parrot [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Applications Places System Parrot Terminal
File Edit View Search Terminal Help
[root@parrot] ~/.home/user]
└─# msfconsole
Metasploit tip: View all productivity tips with the tips command

*Neutrino_Cannon*PrettyBeefy*PostalTime*binbash*deadastronauts*EvilBunnyWrote*L1T*Mail.ru*() { :}; echo vulnerable*
*Team sorceror*ADACTF*BisonSquad*socialdistancing*LeukeTeamNaam*OWASP Moncton*Alegori*exit*Vampire Bunnies*APT593*
*Querfasazombies*NetSecBG*coincoin*ShroomZ*Slow Coders*Scavenger Security*Bruh*NoTeamname*Terminal Cult*
*edsphnre*BFG*MagentaHats*0x0IDA*Kaczukszl*AlphaPwners*FILAH*Raffaela*HackSurYvette*outout*HackSouth*Corax*yeeb0iz*
*SKUUA*Cyber COBRA*flaghunters*0x0IDA Generated*CSEC*0x0mmd*IFS*CTF_Circle*InnotechLabs*baaf00d*BitSwitchers*0xnoobs*
*ITPwns - InterGalactic Team of Pwners*PCCsquared*fz334aks*runMD*0x194*Kapital Krakens*ReadyPlayer1337*Team 443*
*HACKSNOW*InfoQUE*CTF Community*DCZia*NiceWay*0xBlueSky*ME3*Tipi*Hack*Porg Pwl Platoon*HackerTy*hackstreetboys*
*idegeneine007*eggcellent*H4x*cwG7*localhorst*Original Cyan Lonkero*Sad_Pandas*FalseFlag*OurHeartBleedsOrange*SWASD*
*Cult of the Dead Turkey*doesthismatter*crayontheft*Cyber Mausoleum*scriptorz*VetSec*noRobot*Delta Squad Zero*Mukesh*
*x00*x00*BlackATM*AESX*cp*Vaporsec*purplehex*redTeamMITU*Usalama team*vitaminK*K15*forK00MD444*hownowbrownrow*
*ethernot*cheesebaguette*downgrade*FR13NDS*badInfluence*Cut3DriNgOn*dc615*nora*Polaris One*team*hal*hydra*Takoyaki*
*Sudo Society*Infecto*flash*TheScientifics*Tea Party*Reapers of Pwnage*OldBoys*MOUL3Fr1tB13r3*bearswithsaws*DC540*
*iMosuke*Infosec_zitro*CrackTheFlag*TheConquerors*Asur*4fun*Rogue-CTF*Cyber*TMHC*The_Pihacks*btwUseArch*MadDaws*
*HInc*The Mighty Mangolins*CSFE_RamSec*x4n0n*x0rc3rs*emehaci*Ph4n70m_R34p3r*humzig*PreeminenCe*UMGC*ByteBrigade*
*TeamFastMark*Towson-Cyberkatz*meow*xrzhev*PA Hackers*Kulemento*Nakateam*L0gic*80b*NOVA-InfoSec*teamstyle*Panic*
*8ONGR3*
*Les Tontons Flageurs*
* UNION SELECT *password*
*burner_hexZ0*
*here_there_be_trolls*
*rat5_*6unrg4nd*NYUSEC*
*IkastenIO*TWC*balkansec*
*TofuEelRoll*Trash Pandas*
*Astro*Got Schwartz?*tmux*
*lnls Juicy white peach*
*HackerKnights*
*Pentest Rangers*
    "les Cadets Rouges"buf*
        "404 : Flag Not Found"
        "0CD247*Sparkle Pony"
        "Killshot*ConEmu"
        "echo:hacked"
        "karamelle4"
        "cybersecurity.li"
        "OneManArmy*cyb3r_w1z4rd5"
        "AreYouStuck*Mr.Robot.0"
        "EPITA Rennes"
        "guildOfGengar*Titans"
        "The Libbyrators"
└─#
```

Command – nmap -Pn -A -oX Test <Target ip>

Skips ping checks, performs an aggressive scan, and saves the results in XML format named *Test*.

parrot [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

Parrot

```
[*] metasploit v6.4.71-dev
+ --=[ 2529 exploits - 1302 auxiliary - 431 post
+ --=[ 1669 payloads - 49 encoders - 13 nops
+ --=[ 9 evasion

Metasploit Documentation: https://docs.metasploit.com/

[*]{Jobs:0 Agents:0} >> nmap -Pn -A -oX Test 192.168.1.56
[*] exec: nmap -Pn -A -oX Test 192.168.1.56

Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-12-09 18:05 UTC
Nmap scan report for 192.168.1.56
Host is up.
All 1000 scanned ports on 192.168.1.56 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
Too many fingerprints match this host to give specific OS details

TRACEROUTE (using proto 1/icmp)
HOP RTT ADDRESS
1 2.44 ms dlink_router (192.168.0.1)
2 4.97 ms png-78.229.133.103-wyrisnetworks.com (103.133.229.78)
3 4.09 ms png-77.229.133.103-wyrisnetworks.com (103.133.229.77)
4 ...
5 8.28 ms 125.20.136.49
6 ... 30

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 252.02 seconds
[*]{Jobs:0 Agents:0} >>
```

MODULE – 3 SCANNING NETWORKS

Step 2 : type search portscan

parrot [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Applications Places System Terminal Help

Parrot Terminal

TRACEROUTE (using proto 1/icmp)
HOP RTT ADDRESS
1 2.44 ms dlink.router (192.168.0.1)
2 4.97 ms pnq-78.229.133.103-wyrisnetworks.com (103.133.229.78)
3 4.09 ms pnq-77.229.133.103-wyrisnetworks.com (103.133.229.77)
4 ...
5 8.28 ms 125.20.136.49
6 ... 30

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 252.02 seconds
[msf] {Jobs:0 Agents:0} >> search portscan

Matching Modules

#	Name	Disclosure Date	Rank	Check	Description
0	auxiliary/scanner/portscan/ftpbounce	.	normal	No	FTP Bounce Port Scanner
1	auxiliary/scanner/natmp/natmp_portscan	.	normal	No	NAT-PMP External Port Scanner
2	auxiliary/scanner/sap/sap_router_portscanner	.	normal	No	SAPRouter Port Scanner
3	auxiliary/scanner/portscan/xmas	.	normal	No	TCP "XMas" Port Scanner
4	auxiliary/scanner/portcan/ack	.	normal	No	TCP ACK Firewall Scanner
5	auxiliary/scanner/portcan/tcp	.	normal	No	TCP Port Scanner
6	auxiliary/scanner/portcan/syn	.	normal	No	TCP SYN Port Scanner
7	auxiliary/scanner/http/wordpress_pingback_access	.	normal	No	Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access

[msf] {Jobs:0 Agents:0} >>

File Machine View Input Devices Help

Applications Places System Terminal Help

Parrot Terminal

14°C Clear

Search

11:55 PM 09-12-2025

ENG IN

Right Ctrl

Step 3 : type use <**number of port scan**> to set which port scan are you want and press enter and then type **show options** to check , selected port scan number is set or not

- Example - : use 5

parrot [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Applications Places System Parrot Terminal

File Edit View Search Terminal Help

JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS	192.168.1.56	yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
SNAPLEN	65535	yes	The number of bytes to capture
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	500	yes	The reply read timeout in milliseconds

View the full module info with the info, or info -d command.

```
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/syn) >> run
[*] Caught interrupt from the console...
[*] Auxiliary module execution completed
[msf] (Jobs:0 Agents:0) auxiliary/scanner/portscan/syn) >> use 5
[msf] (Jobs:0 Agents:0) auxiliary/scanner/portscan/tcp) >> show options
```

Module options (auxiliary/scanner/portscan/tcp):

Name	Current Setting	Required	Description
CONCURRENCY	10	yes	The number of concurrent ports to check per host
DELAY	0	yes	The delay between connections, per thread, in milliseconds
JITTER	0	yes	The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS	1-10000	yes	Ports to scan (e.g. 22-25,80,110-900)
RHOSTS		yes	The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS	1	yes	The number of concurrent threads (max one per host)
TIMEOUT	1000	yes	The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.

```
[msf] (Jobs:0 Agents:0) auxiliary/scanner/tcp) >>
```

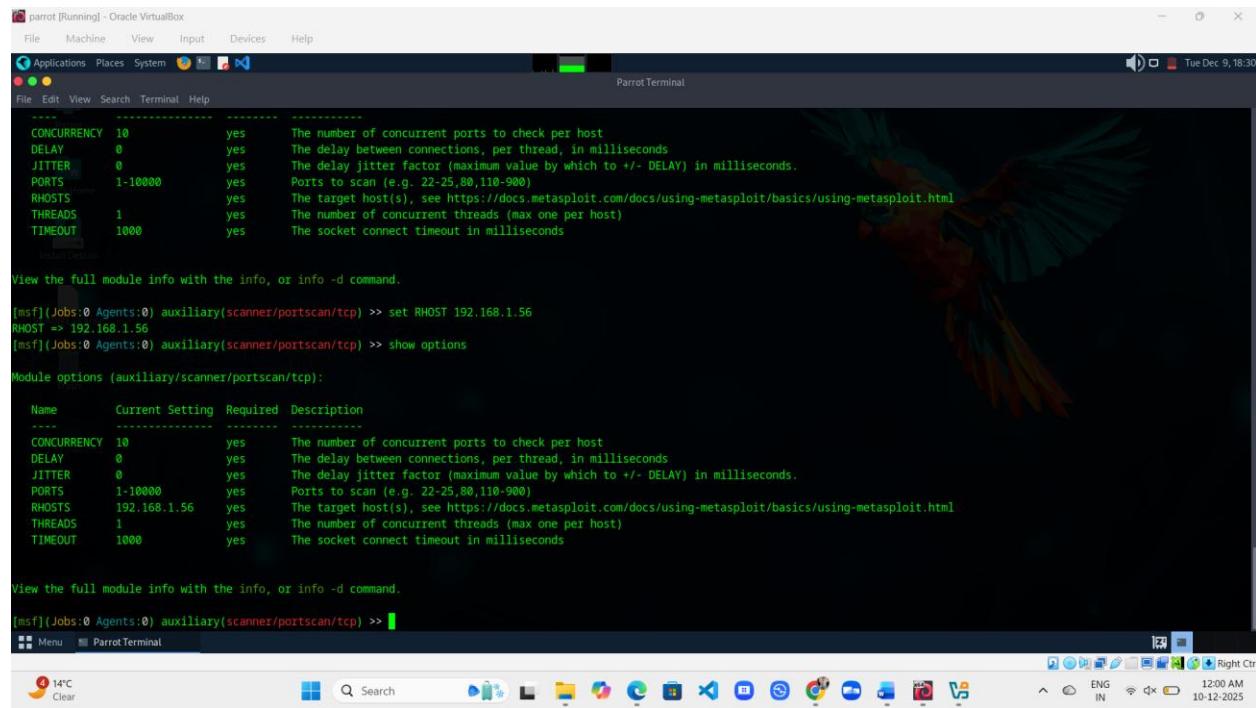
☰ Menu Parrot Terminal

14°C Clear Search

ENG IN 10-12-2025 12:00 AM

MODULE – 3 SCANNING NETWORKS

Step 4 : Then type **set RHOST** to set target ip .



The screenshot shows a terminal window titled "Parrot [Running] - Oracle VirtualBox". The window displays the following Metasploit module configuration:

```
CONCURRENCY 10      yes   The number of concurrent ports to check per host
DELAY          0      yes   The delay between connections, per thread, in milliseconds
JITTER         0      yes   The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS          1-10000  yes   Ports to scan (e.g. 22-25,80,110-900)
RHOSTS         yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS        1      yes   The number of concurrent threads (max one per host)
TIMEOUT        1000   yes   The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.

(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >> set RHOST 192.168.1.56
RHOST => 192.168.1.56
(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >> show options

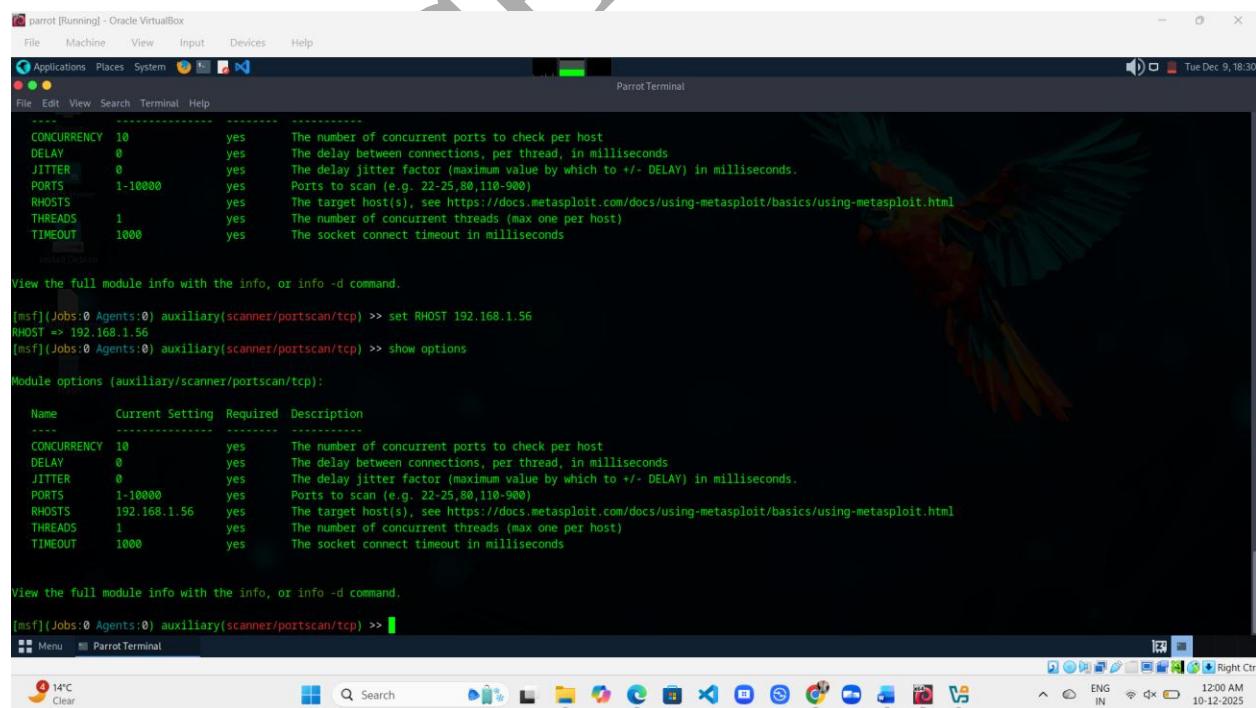
Module options (auxiliary/scanner/portscan/tcp):

Name      Current Setting  Required  Description
-----  -----  -----
CONCURRENCY 10      yes   The number of concurrent ports to check per host
DELAY          0      yes   The delay between connections, per thread, in milliseconds
JITTER         0      yes   The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS          1-10000  yes   Ports to scan (e.g. 22-25,80,110-900)
RHOSTS         yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS        1      yes   The number of concurrent threads (max one per host)
TIMEOUT        1000   yes   The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.

(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >>
```

Step 5 : type **show options** to check target ip are set or not



The screenshot shows a terminal window titled "Parrot [Running] - Oracle VirtualBox". The window displays the following Metasploit module configuration, identical to the previous screenshot but with the RHOST option already set:

```
CONCURRENCY 10      yes   The number of concurrent ports to check per host
DELAY          0      yes   The delay between connections, per thread, in milliseconds
JITTER         0      yes   The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS          1-10000  yes   Ports to scan (e.g. 22-25,80,110-900)
RHOSTS         yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS        1      yes   The number of concurrent threads (max one per host)
TIMEOUT        1000   yes   The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.

(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >> set RHOST 192.168.1.56
RHOST => 192.168.1.56
(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >> show options

Module options (auxiliary/scanner/portscan/tcp):

Name      Current Setting  Required  Description
-----  -----  -----
CONCURRENCY 10      yes   The number of concurrent ports to check per host
DELAY          0      yes   The delay between connections, per thread, in milliseconds
JITTER         0      yes   The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS          1-10000  yes   Ports to scan (e.g. 22-25,80,110-900)
RHOSTS         yes    The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS        1      yes   The number of concurrent threads (max one per host)
TIMEOUT        1000   yes   The socket connect timeout in milliseconds

View the full module info with the info, or info -d command.

(msf){Jobs:0 Agents:0} auxiliary(scanner/portscan/tcp) >>
```

MODULE – 3 SCANNING NETWORKS

Step 6 : type run .

parrot [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Applications Places System Terminal Help

ParrotTerminal

```
CONCURRENCY 10      yes   The number of concurrent ports to check per host
DELAY     0          yes   The delay between connections, per thread, in milliseconds
JITTER    0          yes   The delay jitter factor (maximum value by which to +/- DELAY) in milliseconds.
PORTS     1-10000    yes   Ports to scan (e.g. 22-25,80,110-900)
RHOSTS   192.168.1.56 yes   The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
THREADS   1          yes   The number of concurrent threads (max one per host)
TIMEOUT   1000       yes   The socket connect timeout in milliseconds
```

View the full module info with the info, or info -d command.

```
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >> run

[*] 192.168.1.56      - Caught interrupt from the console...
[*] Auxiliary module execution completed
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >> set RHOST 192.168.0.206
RHOST => 192.168.0.206
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >>
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >> run
[+] 192.168.0.206
[-] 192.168.0.206:700 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:1688 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:1745 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:2314 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:5644 - TCP OPEN
[*] 192.168.0.206
[-] 192.168.0.206:5894 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:8838 - TCP OPEN
[+] 192.168.0.206
[-] 192.168.0.206:8955 - TCP OPEN
[*] 192.168.0.206      - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >>
[msf] (Jobs:0 Agents:0) auxiliary(scanner/portscan/tcp) >> |
```

Menu ParrotTerminal

12°C Clear Search

ENG IN 10-12-2025 Right Ctrl

SACHCHIDĀ

Network Scanning Using Hping3

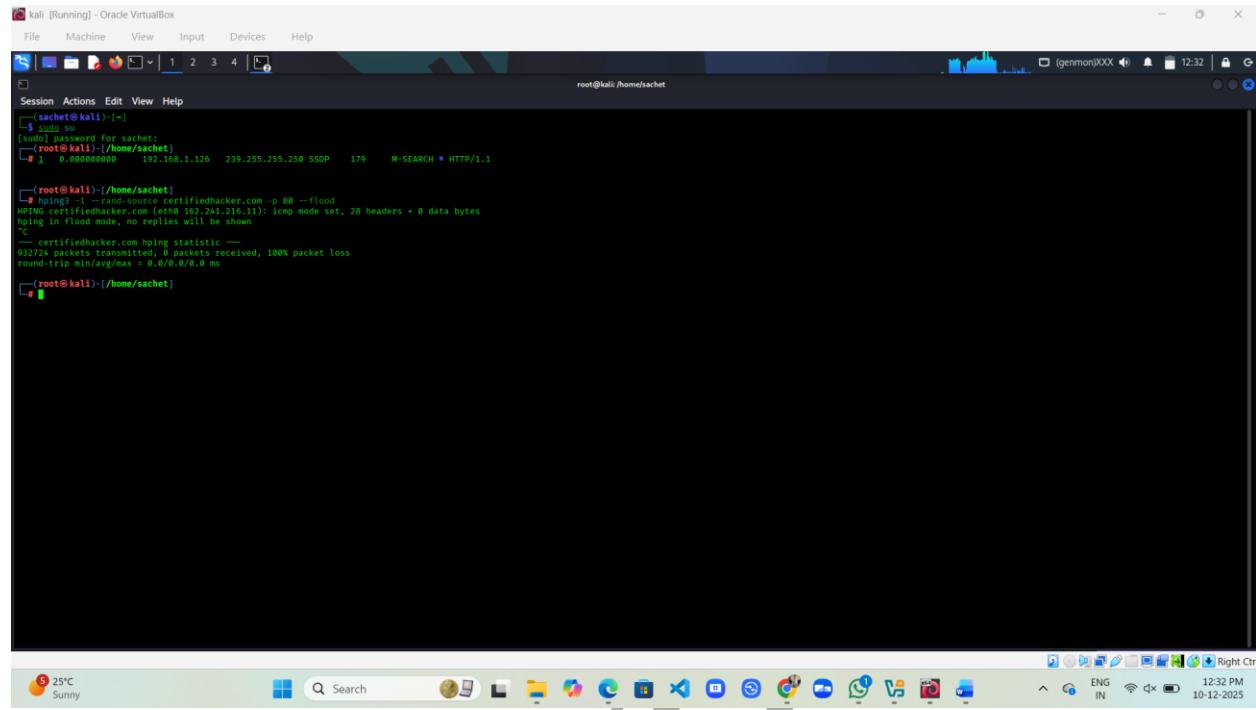
Hping3 is a powerful command-line network tool used for security auditing, network testing, and troubleshooting. It is particularly effective for crafting and sending custom TCP/IP packets, making it a versatile choice for both attackers.

Common Usage Commands –

- -h – help .
- -v – version .
- --fast .
- --faster.
- --flood.
- -V –verbose.
- -1 –ICMP.
- -2 –UDP .
- -8 –Scan .
- -f –fragments.
- -S - -SYN Flag
- -A - -ACK Flag
- -U - -URG Flag
- -R - -RST Flag
- -P - -PUSH Flag
- -X – Xmass

MODULE – 3 SCANNING NETWORKS

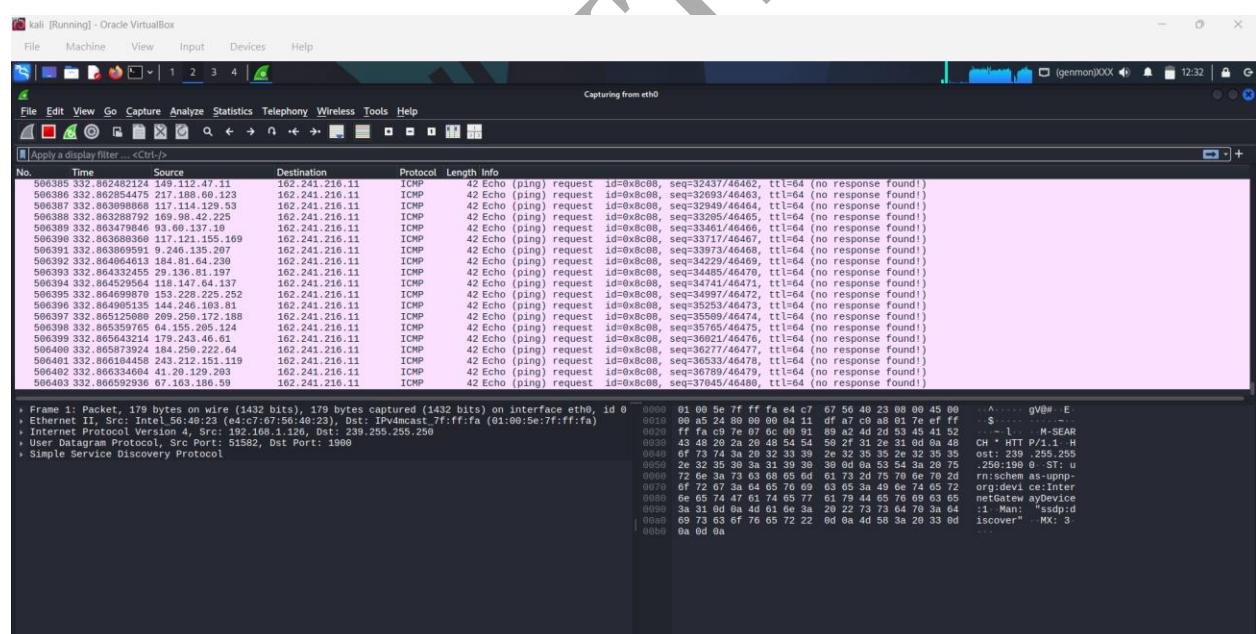
Command- hping3 -1 --rand-source certifiedhacker.com -p 80 –flood



```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[sachet@kali:~]
$ sudo su
[sudo] password for sachet:
(root@kali:~/home/sachet)
[root@kali:~/home/sachet]
# ./hping3 -1 --rand-source certifiedhacker.com -p 80 --flood
HPING certifiedhacker.com (eth0 162.241.216.11): icmp mode set, 28 headers + 0 data bytes
hping in Flood mode, no replies will be shown
932726 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@kali:~/home/sachet]
# 

```



No.	Time	Source	Destination	Protocol	Length	Info
560385	332.862482124	149.112.47.11	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=32437/446462, ttl=64 (no response found!)
560386	332.862482125	217.188.68.123	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=32438/446463, ttl=64 (no response found!)
560387	332.863998860	114.14.14.83	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33289/446464, ttl=64 (no response found!)
560388	332.863288792	169.98.42.225	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33285/446465, ttl=64 (no response found!)
560389	332.863479846	93.66.137.10	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33403/446466, ttl=64 (no response found!)
560390	332.863688366	117.121.155.169	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33717/446467, ttl=64 (no response found!)
560391	332.863708130	148.121.124.120	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33718/446468, ttl=64 (no response found!)
560392	332.864684613	184.81.84.230	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=33720/446469, ttl=64 (no response found!)
560393	332.864332456	29.136.81.197	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=34485/446470, ttl=64 (no response found!)
560394	332.864529564	118.147.64.137	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=34743/446471, ttl=64 (no response found!)
560395	332.864699874	153.228.225.252	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=34997/446472, ttl=64 (no response found!)
560396	332.865125988	103.103.103.103	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=35580/446473, ttl=64 (no response found!)
560397	332.865125989	289.259.172.188	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=35580/446474, ttl=64 (no response found!)
560398	332.865359765	64.155.205.124	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=35765/446475, ttl=64 (no response found!)
560399	332.865643214	179.243.46.63	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=36923/446476, ttl=64 (no response found!)
560400	332.865873924	184.259.222.64	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=36277/446477, ttl=64 (no response found!)
560401	332.866334694	141.20.129.203	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=36780/446479, ttl=64 (no response found!)
560402	332.866334694	41.20.129.203	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=36780/446479, ttl=64 (no response found!)
560403	332.866592936	67.163.186.59	162.241.216.11	ICMP	42	Echo (ping) request id=0x8c08, seq=37045/446480, ttl=64 (no response found!)

> Frame 1: Packet, 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface eth0, id 0 0000 01 00 5e 7f ff fa e4 c7 67 56 40 23 00 00 45 00 ^....gVB# E-
 0010 00 a5 00 00 00 00 00 00 00 00 00 00 00 00 00 00 \$.....-
 0020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 S.....-
 0030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 S.....-
 0040 0f 73 4a 3a 31 32 33 39 30 32 35 36 2e 32 35 35 CH * HTT P/1.1 H
 0050 2e 32 35 39 3a 31 39 39 30 3d 0a 53 54 3a 29 75 .250:190 0 ST: u
 0060 72 6e 3a 73 68 6d 63 2d 75 70 6e 2d rnschem asimp-
 0070 6e 65 74 47 61 74 65 77 63 65 74 65 65 65 65 65 orpervis Inter-
 0080 3a 31 0d 8a 4d 61 6e 3a 29 22 73 73 64 70 3a 64 :1:Man: "ssdp:d
 0090 69 73 63 6f 76 65 72 22 0d 0a 4d 58 3a 20 33 0d iscover" - MX: 3-
 00a0 0a 0d 0a

eth0: <live capture in progress>

MODULE – 3 SCANNING NETWORKS

Command- hping3 -2 --rand-source certifiedhacker.com -p 80 --flood

```
[root@kali :~]# hping3 -2 --rand-source certifiedhacker.com -p 80 --flood
Hping3 certifiedhacker.com (eth0 162.241.216.13): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
```
certifiedhacker.com hping statistic
69035 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0/0.0/0.0 ms
[root@kali :~]
```

| No.    | Time          | Source                    | Destination                               | Protocol | Length | Info             |
|--------|---------------|---------------------------|-------------------------------------------|----------|--------|------------------|
| 569538 | 393.375589194 | 145.87.196.134            | 162.241.216.11                            | UDP      | 42     | 30647 - 80 Len=0 |
| 569539 | 393.375784322 | 135.93.174.123            | 162.241.216.11                            | UDP      | 42     | 30647 - 80 Len=0 |
| 569540 | 393.376011222 | 171.208.222.120           | 162.241.216.11                            | UDP      | 42     | 30648 - 80 Len=0 |
| 569541 | 393.376111916 | 247.188.222.127           | 162.241.216.11                            | UDP      | 42     | 30650 - 80 Len=0 |
| 569542 | 393.376278325 | 174.72.66.181             | 162.241.216.11                            | UDP      | 42     | 30651 - 80 Len=0 |
| 569543 | 393.376440184 | 178.184.9.125             | 162.241.216.11                            | UDP      | 42     | 30652 - 80 Len=0 |
| 569544 | 393.376630323 | 103.159.95.143            | 162.241.216.11                            | UDP      | 42     | 30653 - 80 Len=0 |
| 569545 | 393.376795144 | 162.241.216.11            | 162.241.216.11                            | UDP      | 42     | 30654 - 80 Len=0 |
| 569546 | 393.376917648 | 15.63.252.189             | 162.241.216.11                            | UDP      | 42     | 30655 - 80 Len=0 |
| 569547 | 393.377077640 | 243.125.253.163           | 162.241.216.11                            | UDP      | 42     | 30656 - 80 Len=0 |
| 569548 | 393.377248965 | 124.117.80.214            | 162.241.216.11                            | UDP      | 42     | 30657 - 80 Len=0 |
| 569549 | 393.377414715 | 171.208.117.103           | 162.241.216.11                            | UDP      | 42     | 30658 - 80 Len=0 |
| 569550 | 393.377584444 | 145.15.121.11             | 162.241.216.11                            | UDP      | 42     | 30659 - 80 Len=0 |
| 569551 | 393.377736993 | 95.21.189.248             | 162.241.216.11                            | UDP      | 42     | 30660 - 80 Len=0 |
| 569552 | 393.377982477 | 127.189.42.123            | 162.241.216.11                            | UDP      | 42     | 30661 - 80 Len=0 |
| 569553 | 393.378254345 | 22.205.289.136            | 162.241.216.11                            | UDP      | 42     | 30662 - 80 Len=0 |
| 569554 | 393.378425000 | 162.241.216.11            | 162.241.216.11                            | UDP      | 42     | 30663 - 80 Len=0 |
| 569555 | 393.378481980 | 163.247.46.118            | 162.241.216.11                            | UDP      | 42     | 30664 - 80 Len=0 |
| 569556 | 393.378865418 | PCSystemTec_37:24:...:ARP | 42 Who has 192.168.1.1? Tell 192.168.1.31 |          |        |                  |

> Frame 1: Packet, 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface eth0, id 0 0000 01 00 5e 7f ff e4 c7 67 56 40 23 08 00 45 00 ... ^..... gV@# E
> Ethernet II, Src: Intel\_56:48:23 (e4:c7:67:56:40:23), Dst: IPv4mcast\_7f:ffff:fa (01:00:5e:7f:ff:fa)
> Internet Protocol Version 4, Src: 192.168.1.126, Dst: 239.255.255.250
> User Datagram Protocol, Src Port: 51582, Dst Port: 1900
> Simple Service Discovery Protocol

Profile: Default

## MODULE – 3 SCANNING NETWORKS

**Command- hping3 -2 certifiedhacker.com -a certifiedhacker.com -p 80 --flood**

- **-2** → UDP mode (fires UDP packets)
- **certifiedhacker.com** → Target host
- **-a certifiedhacker.com** → Spoofs the source IP as the same domain (fake sender)
- **-p 80** → Target port 80
- **--flood** → Sends packets nonstop, as fast as the machine can push

```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[root@kali] ~[Home/sachet]
hping3 -2 certifiedhacker.com -a certifiedhacker.com -p 80 --flood
Unable to resolve 'certifiedhacker.com'
[root@kali] ~[Home/sachet]

[root@kali] ~[Home/sachet]
hping3 -2 certifiedhacker.com -a certifiedhacker.com -p 80 --flood
HPING certifiedhacker.com (eth0 162.241.216.11): udp mode set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
C:
*** certifiedhacker.com hping statistic ***
135580 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@kali] ~[Home/sachet]


```

| No.   | Time        | Source         | Destination    | Protocol | Length | Info             |
|-------|-------------|----------------|----------------|----------|--------|------------------|
| 76307 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36765 - 88 Len=8 |
| 76308 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36756 - 88 Len=8 |
| 76309 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36757 - 88 Len=8 |
| 76310 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36758 - 88 Len=8 |
| 76311 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36759 - 88 Len=8 |
| 76312 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36760 - 88 Len=8 |
| 76313 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36761 - 88 Len=8 |
| 76314 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36762 - 88 Len=8 |
| 76315 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36763 - 88 Len=8 |
| 76316 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36764 - 88 Len=8 |
| 76317 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36765 - 88 Len=8 |
| 76318 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36766 - 88 Len=8 |
| 76319 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36767 - 88 Len=8 |
| 76320 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36768 - 88 Len=8 |
| 76321 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36769 - 88 Len=8 |
| 76322 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36770 - 88 Len=8 |
| 76323 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36771 - 88 Len=8 |
| 76324 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36772 - 88 Len=8 |
| 76325 | 0.000000000 | 162.241.216.11 | 162.241.216.11 | UDP      | 42     | 36773 - 88 Len=8 |

> Frame is: Packet, 178 bytes on wire (1432 bits), 178 bytes captured (1432 bits) on interface eth0, id 0
 Ethernet II, Src: Kali Linux (00:0c:29:7e:ff:fa), Dst: IPv4mcast (ff:ff:fa) (01:00:5e:f0:ff:fa)
 Internet Protocol Version 4, Src: 192.168.1.126, Dst: 239.250.255.250
 User Datagram Protocol, Src Port: 51582, Dst Port: 1900
 Simple Service Discovery Protocol

0000 01 00 5e 7f ff fa e4 c7 67 56 40 23 08 00 45 00 .^... gV# E-
 0001 59 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ..\$... .
 0002 ff f4 c9 7e 07 60 00 01 00 02 4d 20 00 00 45 1f 52 .L.. M-SEAR
 0003 43 48 29 1a 20 48 54 54 56 2f 31 2e 31 0d 0a 48 CH \* HTT P/1.1. H
 0040 6f 73 74 3a 20 32 33 39 26 32 35 35 26 32 35 35 ost: 239.255.255
 0050 2e 32 35 39 38 39 38 09 0d 0a 53 54 30 29 75 229.250.250 ST- u
 0060 70 6f 65 74 47 61 74 65 77 61 79 44 65 76 69 63 65 rnisches as:comp
 0070 6f 72 67 3a 64 65 70 69 63 05 3a 49 0e 74 65 72 org:dev1 ce:Inter
 0080 6e 65 74 47 61 74 65 77 61 79 44 65 76 69 63 65 netGatew ayDevice
 0090 3a 31 0d 0a 4d 61 0e 3a 29 22 73 64 70 3a 64 :1-Man: "ssdp:d
 0080 69 73 63 0f 76 65 72 22 0d 0a 4d 58 3a 20 33 0d iscover" -MX: 3-
 0000 0a 00 00

## MODULE – 3 SCANNING NETWORKS

**Command- hping3 -A certifiedhacker.com -a certifiedhacker.com -p 80 –flood**

```
(root㉿kali)-[~/home/sachet]
hping3 -A certifiedhacker.com -a certifiedhacker.com -p 80 --flood
HPING certifiedhacker.com (eth0 162.241.216.13) A set, 48 headers + 0 data bytes
hpPing in flood mode, no replies will be shown
```
certifiedhacker.com hpPing statistic
143270 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
[root@kali]-[~/home/sachet]
```

No.	Time	Source	Destination	Protocol	Length	Info
18260...	748.482394989	162.241.216.11	162.241.216.11	TCP	54	[TCP Previous segment not captured] 43847 -> 80 [ACK] Seq=1218827122 Ack=3878332389 Win=512 Len=0
18267...	748.483543887	162.241.216.11	162.241.216.11	TCP	54	43848 -> 80 [ACK] Seq=1218827123 Ack=3878332390 Win=512 Len=0
18267...	748.484124995	162.241.216.11	162.241.216.11	TCP	54	43850 -> 80 [ACK] Seq=1218827125 Ack=3878332392 Win=512 Len=0
18267...	748.484242499	162.241.216.11	162.241.216.11	TCP	54	43851 -> 80 [ACK] Seq=1218827126 Ack=3878332393 Win=512 Len=0
18267...	748.484538613	162.241.216.11	162.241.216.11	TCP	54	43852 -> 80 [ACK] Seq=1218827127 Ack=3878332394 Win=512 Len=0
18267...	748.484995895	162.241.216.11	162.241.216.11	TCP	54	43852 -> 80 [ACK] Seq=1218827128 Ack=3878332395 Win=512 Len=0
18267...	748.485160686	162.241.216.11	162.241.216.11	TCP	54	43854 -> 80 [ACK] Seq=1218827129 Ack=3878332396 Win=512 Len=0
18267...	748.485253369	162.241.216.11	162.241.216.11	TCP	54	43855 -> 80 [ACK] Seq=1218827130 Ack=3878332397 Win=512 Len=0
18267...	748.485616066	162.241.216.11	162.241.216.11	TCP	54	43855 -> 80 [ACK] Seq=1218827131 Ack=3878332398 Win=512 Len=0
18267...	748.485825360	162.241.216.11	162.241.216.11	TCP	54	43855 -> 80 [ACK] Seq=1218827132 Ack=3878332399 Win=512 Len=0
18267...	748.486070289	162.241.216.11	162.241.216.11	TCP	54	43856 -> 80 [ACK] Seq=1218827133 Ack=3878332400 Win=512 Len=0
18267...	748.486316182	162.241.216.11	162.241.216.11	TCP	54	43857 -> 80 [ACK] Seq=1218827134 Ack=3878332401 Win=512 Len=0
18267...	748.486595777	162.241.216.11	162.241.216.11	TCP	54	43858 -> 80 [ACK] Seq=1218827135 Ack=3878332402 Win=512 Len=0
18267...	748.486877177	162.241.216.11	162.241.216.11	TCP	54	43859 -> 80 [ACK] Seq=1218827136 Ack=3878332403 Win=512 Len=0
18267...	748.487113086	162.241.216.11	162.241.216.11	TCP	54	43860 -> 80 [ACK] Seq=1218827137 Ack=3878332404 Win=512 Len=0
18267...	748.487286484	162.241.216.11	162.241.216.11	TCP	54	43861 -> 80 [ACK] Seq=1218827138 Ack=3878332405 Win=512 Len=0
18267...	748.487452538	162.241.216.11	162.241.216.11	TCP	54	43862 -> 80 [ACK] Seq=1218827139 Ack=3878332406 Win=512 Len=0
18267...	748.487789221	162.241.216.11	162.241.216.11	TCP	54	43863 -> 80 [ACK] Seq=1218827140 Ack=3878332407 Win=512 Len=0
18267...	748.487952897	162.241.216.11	162.241.216.11	TCP	54	43865 -> 80 [ACK] Seq=1218827141 Ack=3878332408 Win=512 Len=0

Frame 990018: Packet, 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0 0000 08 00 27 37 24 09 a0 91 ca 62 bb 61 08 00 45 00 ...'7\$... b a E
Ethernet II, Src: NokiaSolutio_62:bb:01 (a0:91:ca:02:bb:01), Dst: PCSSystemte_37:24:09 (08:00:27:37:24:09)
Internet Protocol Version 4, Src: 162.241.216.11, Dst: 162.241.216.11
Transmission Control Protocol, Src Port: 80, Dst Port: 16726, Seq: 88680815, Len: 8

MODULE – 3 SCANNING NETWORKS

Command- hping3 -S certifiedhacker.com -a certifiedhacker.com -p 80 -flood

kali [Running] - Oracle VirtualBox

File Machine View Input Devices Help

Session Actions Edit View Help

```
[root@kali:~/home/sachet]
# hping3 -s certifiedhacker.com -a certifiedhacker.com -p 80 --flood
hping3: missing host argument
Try 'hping3 --help' for more information.

[root@kali:~/home/sachet]
# hping3 -s certifiedhacker.com -a certifiedhacker.com -p 80 --flood
hping3: missing host argument
Try 'hping3 --help' for more information.

[root@kali:~/home/sachet]
# hping3 -s certifiedhacker.com -a certifiedhacker.com -p 80 --flood
hping3: missing host argument
Try 'hping3 --help' for more information.

[root@kali:~/home/sachet]
# hping3 -s certifiedhacker.com -a certifiedhacker.com -p 80 --flood
HPING certifiedhacker.com (eth0 162.241.216.11): S set, 40 headers + 0 data bytes
Hping in flood mode, no replies will be shown
[...]
-- certifiedhacker.com hping statistic --
308429 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

[root@kali:~/home/sachet]
```

Capturing from eth0

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
882317	683.085548325	162.241.216.11	162.241.216.11	TCP	54	543235 - 88 [SYN] Seq=0 Win=512 Len=0
882318	683.086762189	162.241.216.11	162.241.216.11	TCP	54	543236 - 88 [SYN] Seq=0 Win=512 Len=0
882319	683.086762189	162.241.216.11	162.241.216.11	TCP	54	543237 - 88 [SYN] Seq=0 Win=512 Len=0
882320	683.086924478	162.241.216.11	162.241.216.11	TCP	54	543238 - 88 [SYN] Seq=0 Win=512 Len=0
882321	683.089485577	162.241.216.11	162.241.216.11	TCP	54	543239 - 88 [SYN] Seq=0 Win=512 Len=0
882322	683.089756614	162.241.216.11	162.241.216.11	TCP	54	543240 - 88 [SYN] Seq=0 Win=512 Len=0
882323	683.099984963	162.241.216.11	162.241.216.11	TCP	54	543241 - 88 [SYN] Seq=0 Win=512 Len=0
882324	683.099984963	162.241.216.11	162.241.216.11	TCP	54	543242 - 88 [SYN] Seq=0 Win=512 Len=0
882325	683.010453754	162.241.216.11	162.241.216.11	TCP	54	543243 - 88 [SYN] Seq=0 Win=512 Len=0
882326	683.010820296	162.241.216.11	162.241.216.11	TCP	54	543244 - 88 [SYN] Seq=0 Win=512 Len=0
882327	683.011089834	162.241.216.11	162.241.216.11	TCP	54	543245 - 88 [SYN] Seq=0 Win=512 Len=0
882328	683.011089834	162.241.216.11	162.241.216.11	TCP	54	543246 - 88 [SYN] Seq=0 Win=512 Len=0
882329	683.011989718	162.241.216.11	162.241.216.11	TCP	54	543247 - 88 [SYN] Seq=0 Win=512 Len=0
882330	683.012260431	162.241.216.11	162.241.216.11	TCP	54	543248 - 88 [SYN] Seq=0 Win=512 Len=0
882331	683.013559961	162.241.216.11	162.241.216.11	TCP	54	543249 - 88 [SYN] Seq=0 Win=512 Len=0
882332	683.013559961	162.241.216.11	162.241.216.11	TCP	54	543250 - 88 [SYN] Seq=0 Win=512 Len=0
882333	683.013559961	162.241.216.11	162.241.216.11	TCP	54	543251 - 88 [SYN] Seq=0 Win=512 Len=0
882334	683.014297380	162.241.216.11	162.241.216.11	TCP	54	543252 - 88 [SYN] Seq=0 Win=512 Len=0
882335	683.014555629	162.241.216.11	162.241.216.11	TCP	54	543253 - 88 [SYN] Seq=0 Win=512 Len=0

Frame 1: Packet, 179 bytes on wire (1432 bits), 179 bytes captured (1432 bits) on interface eth0, id 0

Ethernet II, Src: Intel_56:40:23 (e4:67:56:40:23:ff), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)

Internet Protocol Version 4, Src: 192.168.1.126, Dst: 239.255.255.256

User Datagram Protocol, Src Port: 1990, Dst Port: 1990

Simple Service Discovery Protocol

0000 01 00 5e 7f ff fa e4 c7 67 56 40 23 08 00 45 00 gVB# - E

0010 00 a5 24 89 00 99 04 11 df a7 c9 a8 01 7e ef ff S

0020 77 6e 73 63 3a 05 60 61 73 2d 70 6e 5f 52 52 52 l - H -SEAR

0030 03 48 28 2a 08 54 54 50 2f 31 2e 31 0d 0a 48 CH * HTT P/1.1 - H

0040 6f 73 74 3a 20 32 33 39 2e 32 35 36 2e 32 35 35 ost: 239 - 255.255

0050 2e 32 35 38 3a 31 39 38 38 0d 0a 53 54 3a 29 75 :259:199 0 - S: t u

0060 77 6e 73 63 3a 05 60 61 73 2d 70 6e 5f 52 52 52 rnschem as-ump-

0070 0f 67 47 61 65 77 69 44 65 76 69 63 65 67 72 0riginator

0080 6e 65 47 61 74 65 77 69 44 65 76 69 63 65 netGatew sdevice

0090 3a 31 0d 6f 4d 61 6e 3a 20 22 73 73 64 70 3a 64 :1 Man: sdspd

00a0 69 73 03 6f 76 65 72 22 0d 0a 4d 58 3a 28 33 0d discover" MX: 3

00b0 0a 0d 0a . . .

Packets: 802335 | Packets: 802335 | Profile: Default

Perform Network Scanning Using AI

Leveraging AI for network scanning boosts the efficiency and accuracy of security assessments. AI-driven tools can quickly pinpoint weaknesses, unusual behavior, and misconfigurations that might slip past traditional scanners.

Lab Scenario

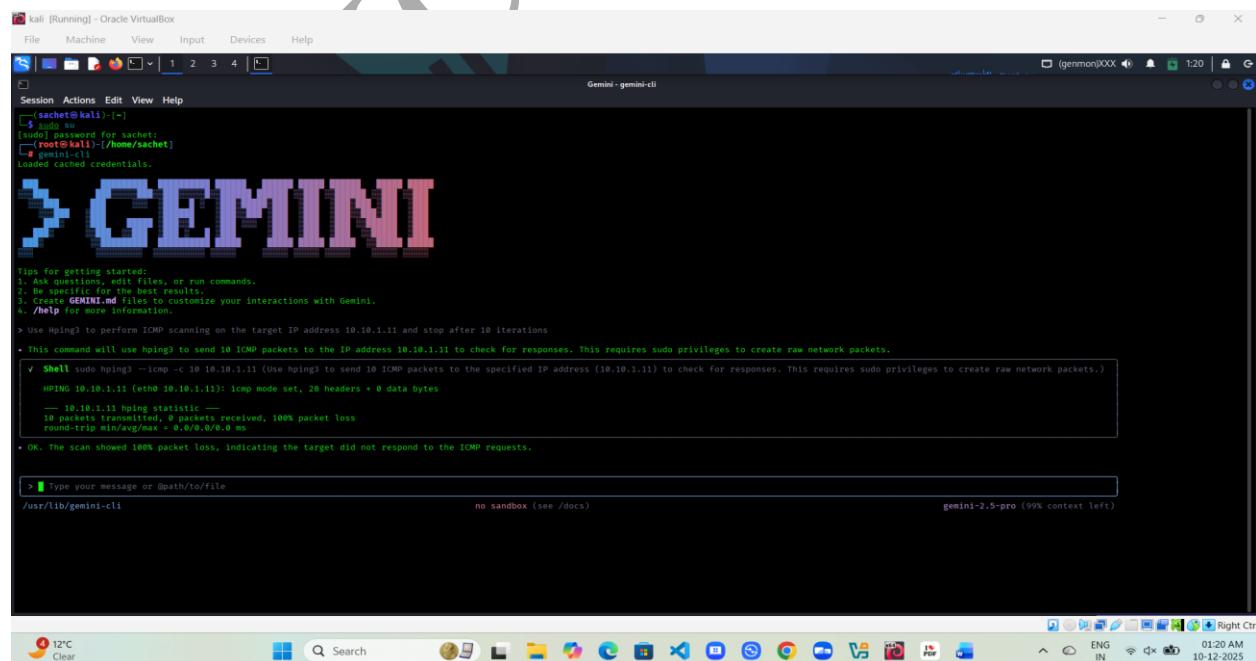
As ethical hackers and penetration testers, tapping into AI-powered automation becomes a major advantage. This lab explores how to use **Gemini CLI**, an AI-enhanced scanning assistant, to extract detailed information about hosts and services within a target network. By combining AI with established reconnaissance techniques, we can reveal vulnerabilities that would otherwise stay hidden.

Lab Objectives

- Scan a target using **Gemini-cli**
- Generate, refine, and automate scan commands through AI assistance

Scan a Target using Gemini cli

Prompt - Use Hping3 to perform ICMP scanning on the target IP address 10.10.1.11 and stop after 10 iterations



```

kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
[sachet@kali:~]
[sudo] password for sachet:
(root@sachet:~/) gemini-cli
# gemini-cli
Loaded cached credentials.

> Use Hping3 to perform ICMP scanning on the target IP address 10.10.1.11 and stop after 10 iterations
This command will use hping3 to send 10 ICMP packets to the IP address 10.10.1.11 to check for responses. This requires sudo privileges to create raw network packets.
V Shell sudo hping3 -icmp -c 10 10.10.1.11 (Use hping3 to send 10 ICMP packets to the specified IP address (10.10.1.11) to check for responses. This requires sudo privileges to create raw network packets.)
HPING 10.10.1.11 (eth0 10.10.1.11): icmp mode set, 28 headers + 0 data bytes
10.10.1.11 hping statistic
10 ICMP packets transmitted, 100% packet loss
round-trip min/avg/max = 0.0/0/0.0 ms
OK. The scan showed 100% packet loss, indicating the target did not respond to the ICMP requests.

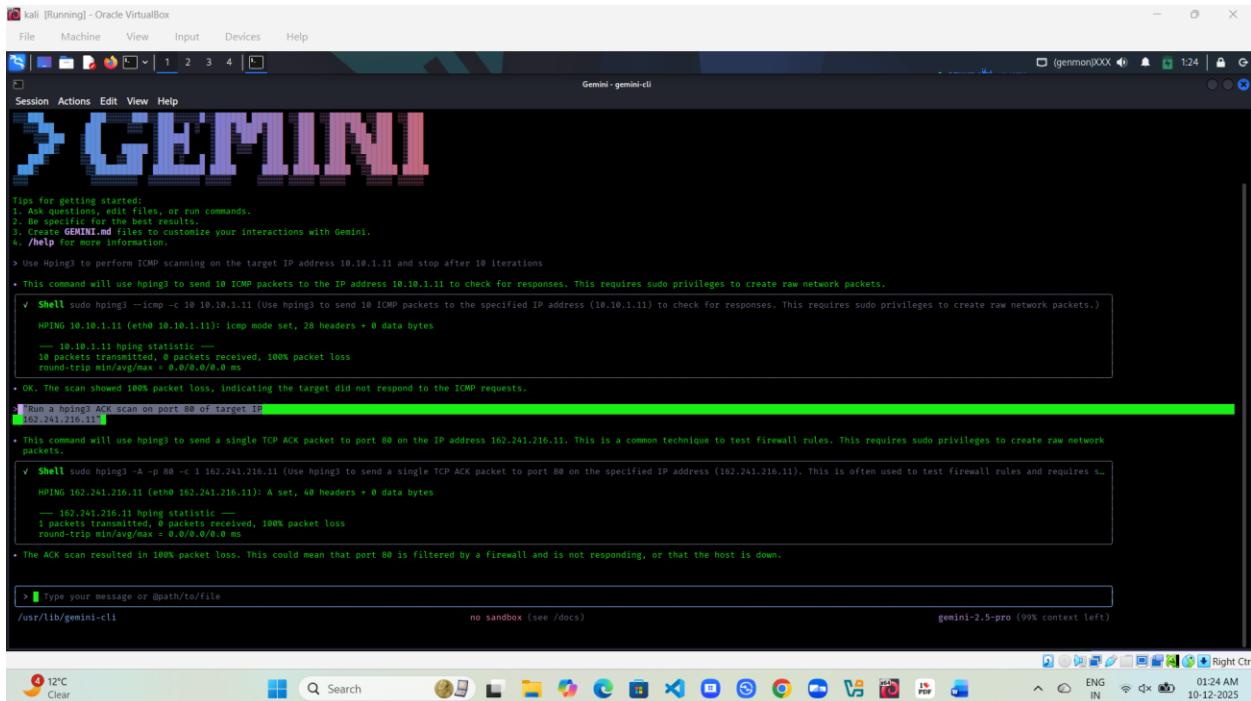
> Type your message or @path/to/File
/usr/lib/gemini-cli
no sandbox (see /docs)
gemini-2.5-pro (99% context left)

12°C Clear
Search
10:20 AM 10-12-2025
ENG IN Right Ctrl

```

MODULE – 3 SCANNING NETWORKS

Prompt - "Run a hping3 ACK scan on port 80 of target IP 162.241.216.11"



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Gemini - gemini-cli
Session Actions Edit View Help
> GEMINI
Tip for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Explore GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> Use Hping3 to perform ICMP scanning on the target IP address 10.10.1.11 and stop after 10 iterations
• This command will use hping3 to send 10 ICMP packets to the IP address 10.10.1.11 to check for responses. This requires sudo privileges to create raw network packets.
v Shell sudo hping3 --icmp -c 10 10.10.1.11 (Use hping3 to send 10 ICMP packets to the specified IP address (10.10.1.11) to check for responses. This requires sudo privileges to create raw network packets.)
-- 10.10.1.11 hping statistic --
10 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

• OK. The scan showed 100% packet loss, indicating the target did not respond to the ICMP requests.

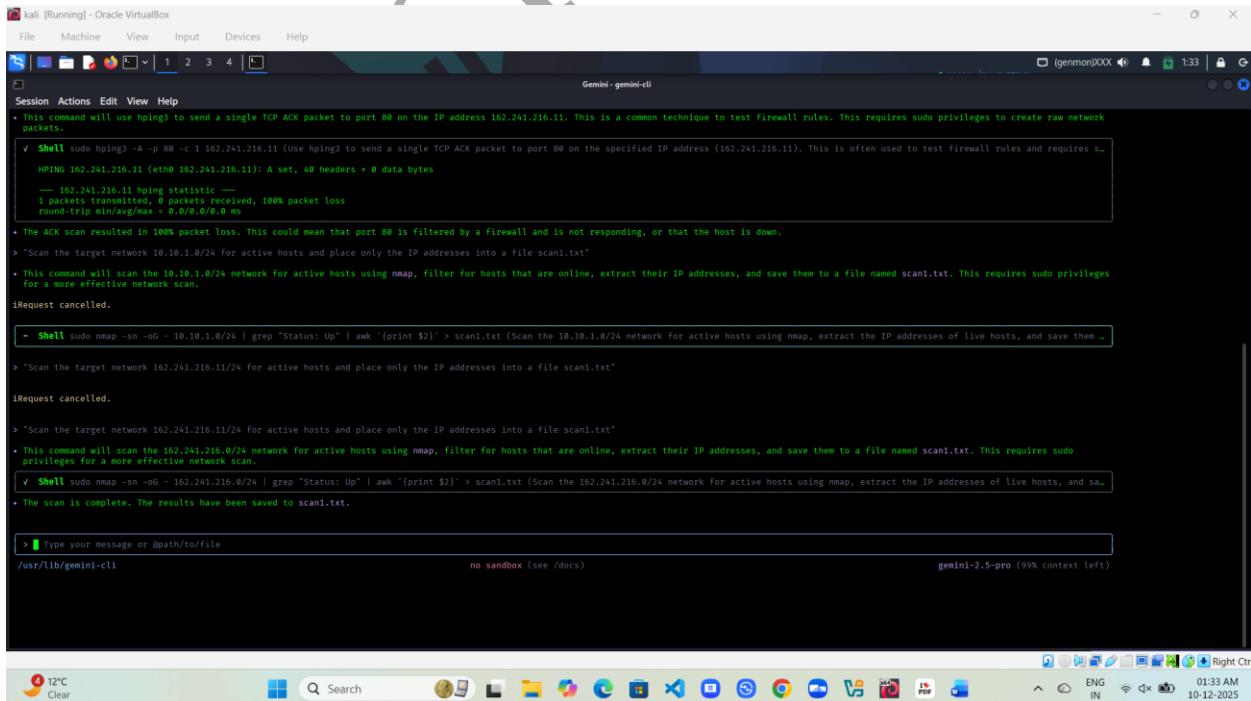
> Run a hping3 ACK scan on port 80 of target IP 162.241.216.11
v Shell sudo hping3 -A -p 80 -c 1 162.241.216.11 (Use hping3 to send a single TCP ACK packet to port 80 on the specified IP address (162.241.216.11). This is often used to test firewall rules and requires s...
-- 162.241.216.11 hping statistic --
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

• The ACK scan resulted in 100% packet loss. This could mean that port 80 is filtered by a firewall and is not responding, or that the host is down.

> Type your message or @path/to/file
/usr/lib/gemini-cli
no sandbox (see /docs)
gemini-2.5-pro (99% context left)

12°C Clear
Search ENG IN 01:24 AM 10-12-2025 Right Ctrl
```

Prompt - "Scan the target network 162.241.216.11/24 for active hosts and place only the IP addresses into a file scan1.txt"



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Gemini - gemini-cli
Session Actions Edit View Help
> GEMINI
Tip for getting started:
1. Ask questions, edit files, or run commands.
2. Be specific for the best results.
3. Explore GEMINI.md files to customize your interactions with Gemini.
4. /help for more information.

> This command will use hping3 to send a single TCP ACK packet to port 80 on the IP address 162.241.216.11. This is often used to test firewall rules and requires s...
v Shell sudo hping3 -A -p 80 -c 1 162.241.216.11 (Use hping3 to send a single TCP ACK packet to port 80 on the specified IP address (162.241.216.11). This is often used to test firewall rules and requires s...
-- 162.241.216.11 hping statistic --
1 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms

• The ACK scan resulted in 100% packet loss. This could mean that port 80 is filtered by a firewall and is not responding, or that the host is down.

> Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt*
• This command will scan the 10.10.1.0/24 network for active hosts using nmap, filter for hosts that are online, extract their IP addresses, and save them to a file named scan1.txt. This requires sudo privileges for a more effective network scan.

iRequest cancelled.

> Shell sudo nmap -sn -oG - 10.10.1.0/24 | grep "Status: Up" | awk '{print $2}' > scan1.txt (Scan the 10.10.1.0/24 network for active hosts using nmap, extract the IP addresses of live hosts, and save them ...
> Scan the target network 162.241.216.11/24 for active hosts and place only the IP addresses into a file scan1.txt*
iRequest cancelled.

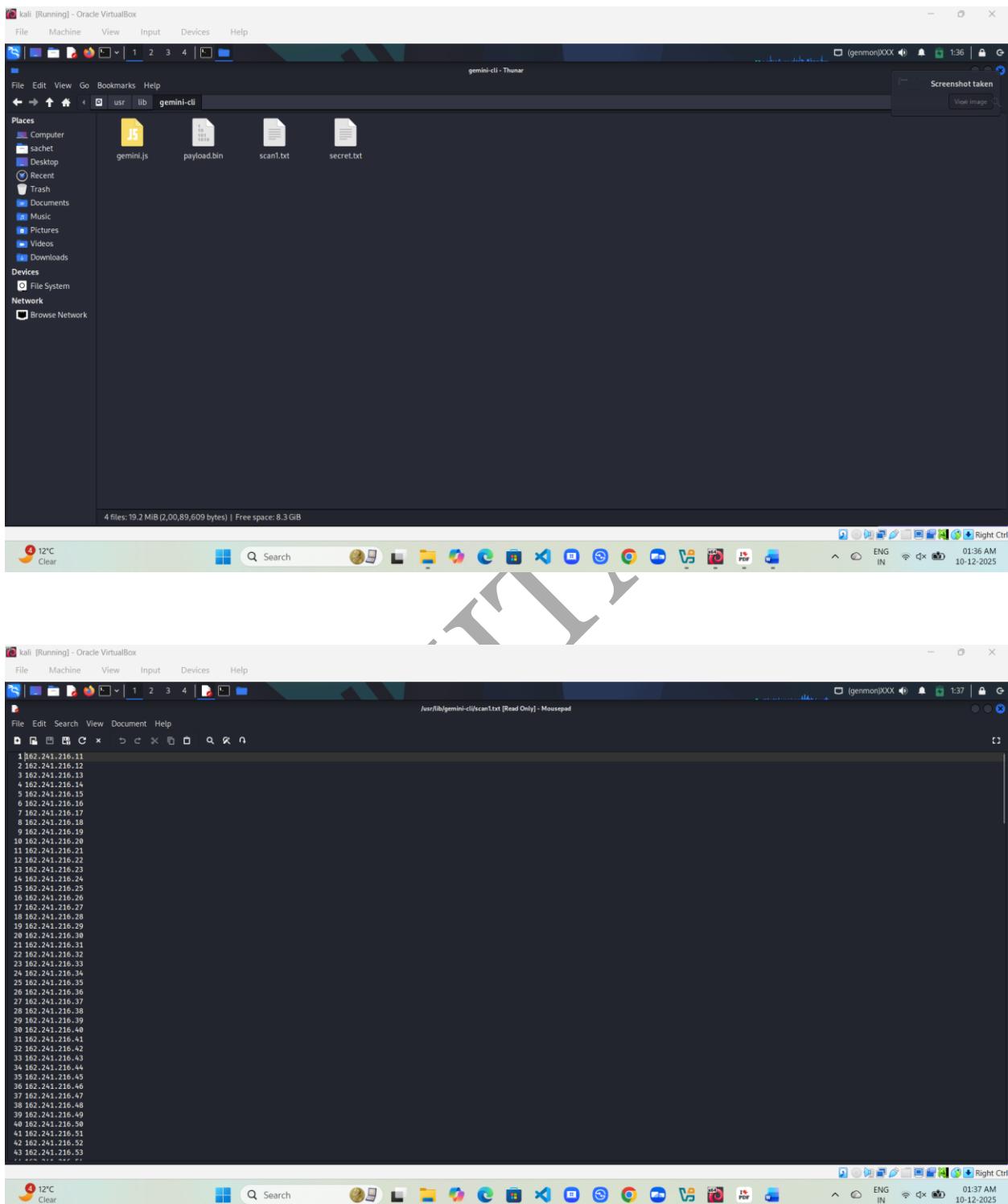
> Scan the target network 162.241.216.11/24 for active hosts and place only the IP addresses into a file scan1.txt*
• This command will scan the 162.241.216.0/24 network for active hosts using nmap, filter for hosts that are online, extract their IP addresses, and save them to a file named scan1.txt. This requires sudo privileges for a more effective network scan.
v Shell sudo nmap -sn -oG - 162.241.216.0/24 | grep "Status: Up" | awk '{print $2}' > scan1.txt (Scan the 162.241.216.0/24 network for active hosts using nmap, extract the IP addresses of live hosts, and save them ...
• The scan is complete. The results have been saved to scan1.txt.

> Type your message or @path/to/file
/usr/lib/gemini-cli
no sandbox (see /docs)
gemini-2.5-pro (99% context left)

12°C Clear
Search ENG IN 01:33 AM 10-12-2025 Right Ctrl
```

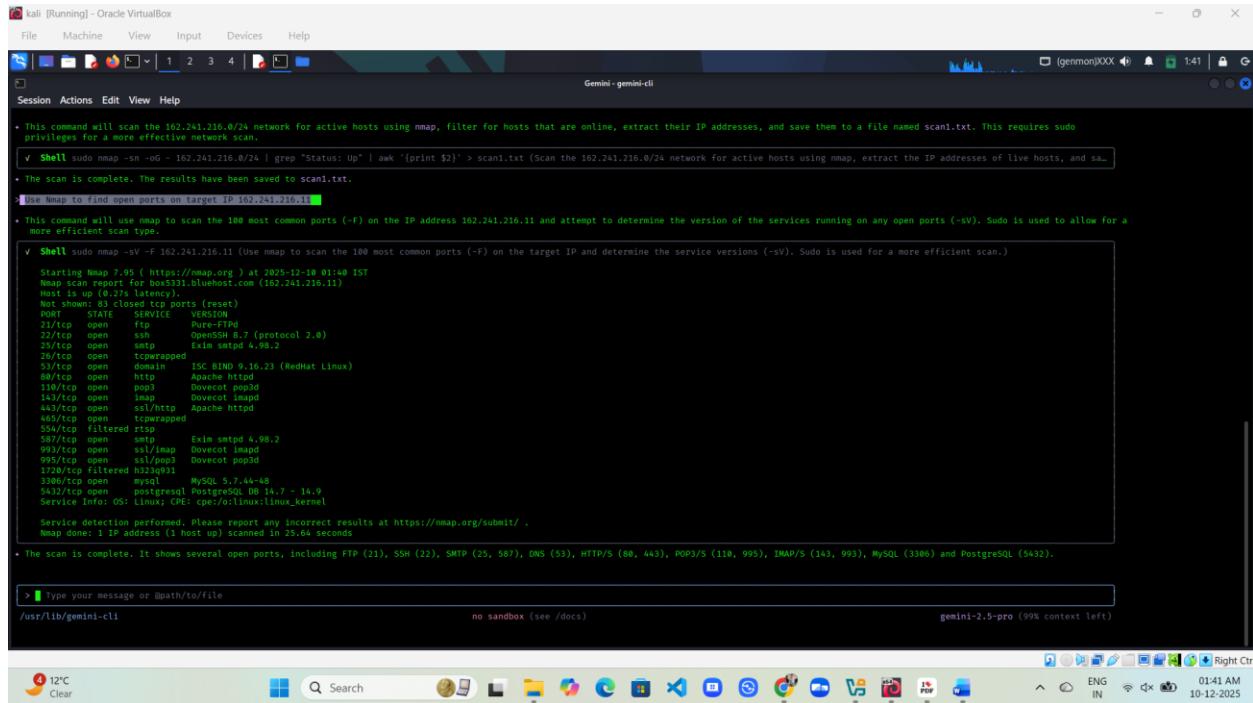
MODULE – 3 SCANNING NETWORKS

The scan is complete. The results have been saved to scan1.txt.



MODULE – 3 SCANNING NETWORKS

Prompt - Use Nmap to find open ports on target IP 162.241.216.11



```
kali [Running] - Oracle VirtualBox
File Machine View Input Devices Help
Session Actions Edit View Help
Gemini - gemini-cli
* This command will scan the 162.241.216.0/24 network for active hosts using nmap, filter for hosts that are online, extract their IP addresses, and save them to a file named scan1.txt. This requires sudo privileges for a more effective network scan.
v Shell sudo nmap -sn -oG - 162.241.216.0/24 | grep "Status: Up" | awk '{print $2}' > scan1.txt (Scan the 162.241.216.0/24 network for active hosts using nmap, extract the IP addresses of live hosts, and save them to a file named scan1.txt)
* The scan is complete. The results have been saved to scan1.txt.
* Use Nmap to find open ports on target IP 162.241.216.11
* This command will use nmap to scan the 100 most common ports (-F) on the IP address 162.241.216.11 and attempt to determine the version of the services running on any open ports (-sV). Sudo is used to allow for a more efficient scan type.
v Shell sudo nmap -sV -F 162.241.216.11 (Use nmap to scan the 100 most common ports (-F) on the target IP and determine the service versions (-sV). Sudo is used for a more efficient scan.)
Starting Nmap 7.95 ( https://nmap.org ) at 2025-12-18 01:40 IST
Nmap scan Report for box321.bluehost.com (162.241.216.11)
Host is up (0.27s latency).
Not shown: 83 closed tcp ports (reset)
      STATE      SERVICE      VERSION
  21/tcp      open  ssh        OpenSSH 8.7 (protocol 2.0)
  22/tcp      open  ssh        OpenSSH 8.7 (protocol 2.0)
  25/tcp      open  smtp       Exim smtpd 4.90.2
  26/tcp      open  tcptrapped
  42/tcp      open  dircat     ISC RIND 9.16.0.23 (RedHat Linux)
  80/tcp      open  http       Apache httpd
  110/tcp     open  pop3      Dovecot pop3d
  119/tcp     open  ipp       Dovecot impd
  433/tcp     open  ssl/http  apache httpd
  465/tcp     open  smtp      Exim smtpd 4.90.2
  534/tcp     Filtered rtsp
  587/tcp     open  smtp      Exim smtpd 4.90.2
  993/tcp     open  ssl/pop3  Dovecot pop3d
  995/tcp     open  ssl/imap  Dovecot impd
  1720/tcp    Filtered h222q931
  3306/tcp    open  mysql     MySQL 5.7.44-44
  5432/tcp    open  postgresql PostgreSQL DB 14.7 - 16.9
Service Info: OS: Linux | CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 25.66 seconds
* The scan is complete. It shows several open ports, including FTP (21), SSH (22), SMTP (25, 587), DNS (53), HTTPS (80, 443), POP3/S (110, 995), IMAP/S (143, 993), MySQL (3306) and PostgreSQL (5432).
```

Network Scanning Countermeasures

Defensive Strategies

Think of this as your "old-school wisdom meets modern hustle" playbook. The goal? Stop attackers *before* they even get the chance to vibe with your network.

Key moves include:

- Tightening policies so no one wanders in accidentally.
- Using layered security — firewall + IDS/IPS + monitoring.
- Keeping systems patched so vulnerabilities don't hang around like unwanted guests.

It's basically giving your network a disciplined routine: wake up early, eat clean, stay ready.

Hardening Systems Against Scanning

System hardening is like telling your servers:
“Stand tall, don’t overshare, and don’t trust strangers.”

This involves:

- Disabling unused services and ports — less noise, less attack surface.
- Enforcing strong authentication.
- Regular configuration audits.
- Removing default credentials (because seriously... attackers LOVE those).

The whole vibe is: keep things simple, clean, and tight.

IDS/Firewall Tuning

Firewalls and IDS are your bouncers — but even bouncers need training.

Tuning includes:

- Setting rules to detect weird traffic patterns (like an attacker ping-sweeping you at 3 AM).
- Enabling rate-limiting to slow down aggressive scanners.
- Blocking malformed packets or suspicious TCP flag combos.
- Whitelisting trusted IPs and tightening down on unknowns.

This is you telling your security tools:
"Don't snooze. Stay suspicious. Catch the shady stuff early."

Reducing Attack Surface

Honestly, this one's the classic:
Less exposed = less to break.

Ways to shrink your network's "visible footprint":

- Segment networks so everything isn't exposed in one giant buffet.
- Hide internal details via NAT and VLANs.
- Disable ICMP responses if not needed.
- Use VPN for admin access instead of exposing management ports publicly.

This is like closing extra doors and windows so attackers can't peek inside.

Monitoring & Logging Enhancements

Logs are receipts — and attackers *hate* receipts.

Good monitoring means:

- Centralizing logs (SIEM tools are your besties here).
- Alerting on unusual patterns like repeated port scans.
- Tracking failed login attempts, odd connection attempts, or high-frequency probes.
- Retaining logs long enough so nothing slips through memory holes.

SACHCHITANAND
THANK YOU