UJAR TECH SOLUTION

NAME: Rithik varma

INTERN ID: UTS1103

TASK 2

Perform a Basic Network Scan Using Nmap:

Understand and perform network scanning to identify open ports and services using (NMAP) whit the help of Kali Linux Terminal.

PRACTICAL DESCRIPTION

Problem:- *Explore a Linux machine to builds your foundational skills in ethical hacking and reconnaissance using Nmap (Network Mapper) tool for port scan for https address (scanme.nmap.org).*

Key Concepts of Nmap:

THE TERM NMAP - (***NETWORK MAPPER***) IT IS **FREE AND OPEN-SOURCE** TOOL USED FOR **NETWORK DISCOVERY** AND **SECURITY AUDITING.**IT IS MOSTLY WIDELY USED TOOLS BY CYBERSECURITY PROFESSIONALS, SYSTEM ADMINISTRATORS, AND ETHICAL HACKERS.

Uses of Nmap:

- IT HELPS TO FIND WHICH DEVICES ARE ONLINE THAT **IS DISCOVER HOST** ON A NETWORK.
- IT SCAN PORTS TO CHECK WHICH ARE OPEN , CLOSED, OR FILTERED, UNFILTERED.
- IDENTIFY SERVICES RUNNING ON OPEN PORTS LIKE WEB SERVER.
- DETECT OPERATING SYSTEMS AND DEVICE TYPES AND DETECT VULNERABILITIES.

Working of Nmap:

LET US UNERSTAND WITH EXAMPLE

THINK OF A **BUILDING WITH MANY DOORS** (PORTS).

- SOME DOORS ARE **OPEN** → ANYONE CAN WALK IN.
- SOME ARE **CLOSED** → NO ENTRY.
- SOME HAVE A **SECURITY GUARD** (FIREWALL) → ONLY CERTAIN PEOPLE ALLOWED.

NMAP SENDS SMALL PACKETS TO THESE “DOORS” TO SEE:

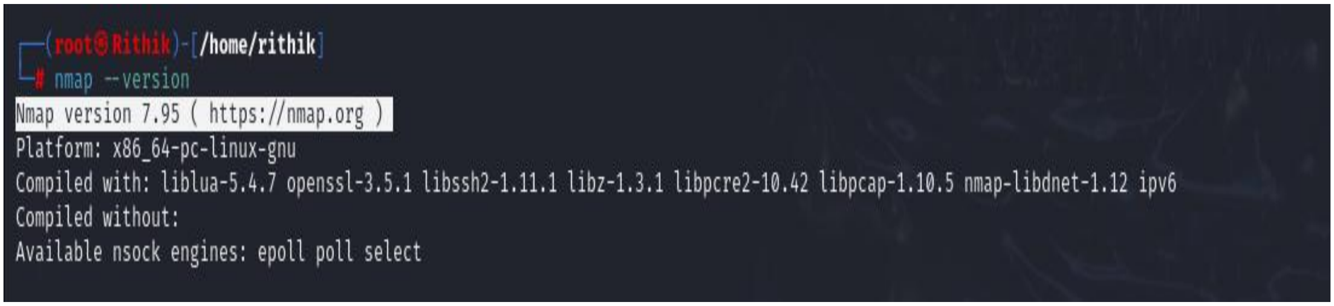
- IS THE DOOR OPEN?
- WHAT’S INSIDE? (SERVICE)
- WHAT KIND OF BUILDING IS IT? (OPERATING SYSTEM)

Common Nmap Commands used in this practical:

Command	*Purpose*
nmap scanme.nmap.org	*Basic scan for open ports.*
nmap -sV scanme.nmap.org	*Detect service versions.*
nmap -O scanme.nmap.org	*Detect the operating system.*
nmap -p- scanme.nmap.org	*Scan all 65535 ports.*
nmap -A scanme.nmap.org	*Aggressive scan (services + OS + scripts).*

Practical of Nmap Using Kali Linux:

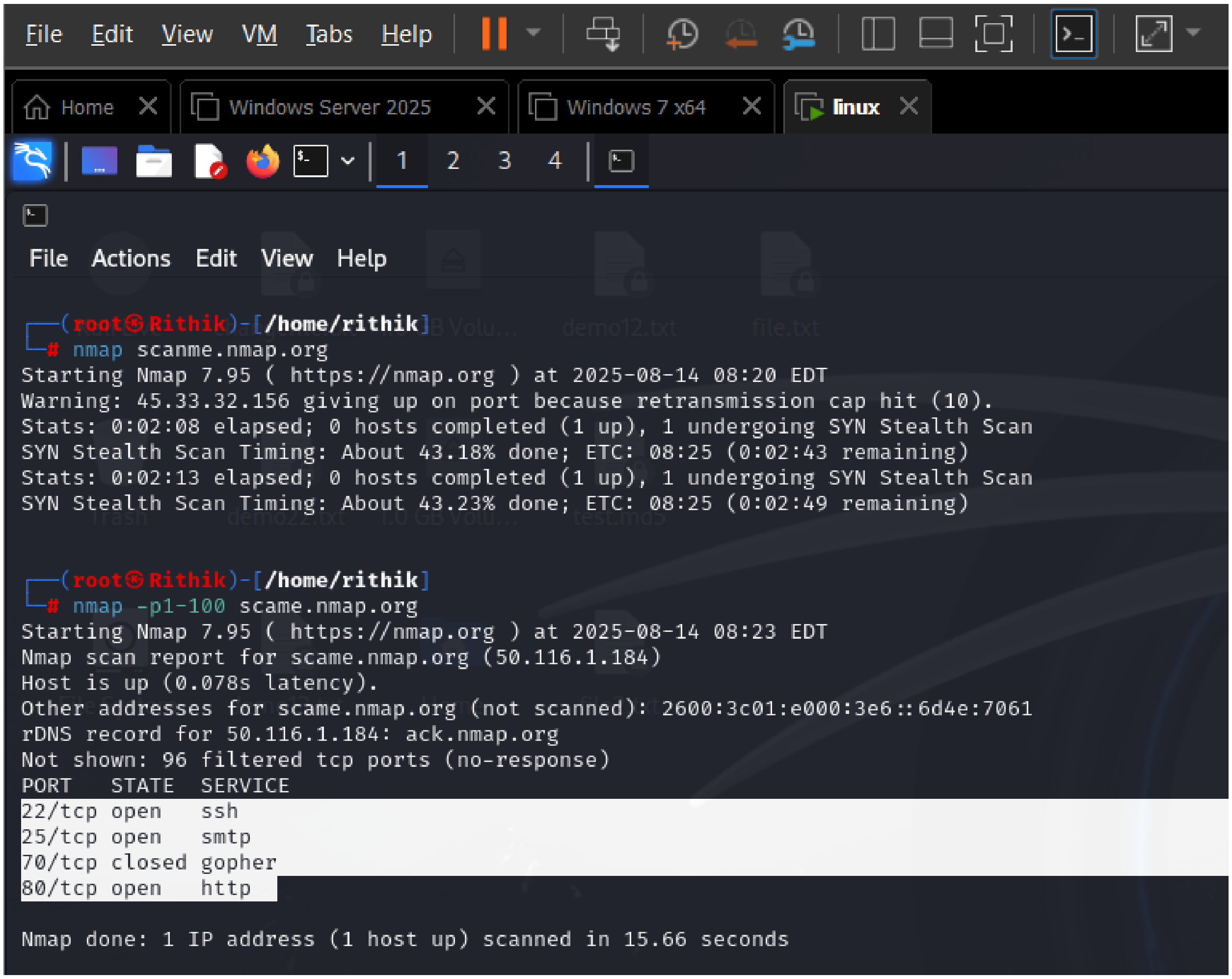
1. Installing Nmap: -



```
┌──(root㉿Rithik)-[/home/rithik]
└─# nmap --version
Nmap version 7.95 ( https://nmap.org )
Platform: x86_64-pc-linux-gnu
Compiled with: liblua-5.4.7 openssl-3.5.1 libssh2-1.11.1 libz-1.3.1 libpcre2-10.42 libpcap-1.10.5 nmap-libdnet-1.12 ipv6
Compiled without:
Available nsock engines: epoll poll select
```

Here as there is pre install nmap tool with updated Version 7.95 . To install we use (sudo apt install nmap) commond.

2. Basic Scan: -

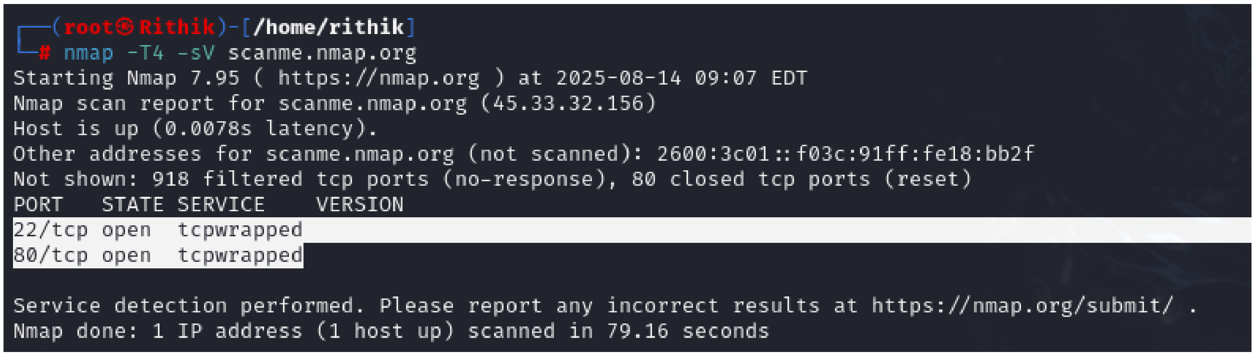


```
┌──(root㉿Rithik)-[/home/rithik]
└─# nmap scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 08:20 EDT
Warning: 45.33.32.156 giving up on port because retransmission cap hit (10).
Stats: 0:02:08 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 43.18% done; ETC: 08:25 (0:02:43 remaining)
Stats: 0:02:13 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 43.23% done; ETC: 08:25 (0:02:49 remaining)


┌──(root㉿Rithik)-[/home/rithik]
└─# nmap -p1-100 scame.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 08:23 EDT
Nmap scan report for scame.nmap.org (50.116.1.184)
Host is up (0.078s latency).
Other addresses for scame.nmap.org (not scanned): 2600:3c01:e000:3e6::6d4e:7061
rDNS record for 50.116.1.184: ack.nmap.org
Not shown: 96 filtered tcp ports (no-response)
PORT   STATE  SERVICE
22/tcp open   ssh
25/tcp open   smtp
70/tcp closed gopher
80/tcp open   http

Nmap done: 1 IP address (1 host up) scanned in 15.66 seconds
```

Here the process of Scan the public server for open ports using (nmap scanme.nmap.org) and received a port status that some are open/close

- ***22/tcp open ssh → Secure remote login service.***
- ***80/tcp open http → Web server running.***
- ***25/tcp open smtp → Email sending service.***
- ***70/tcp closed gopher.***

3. Service Detection: -

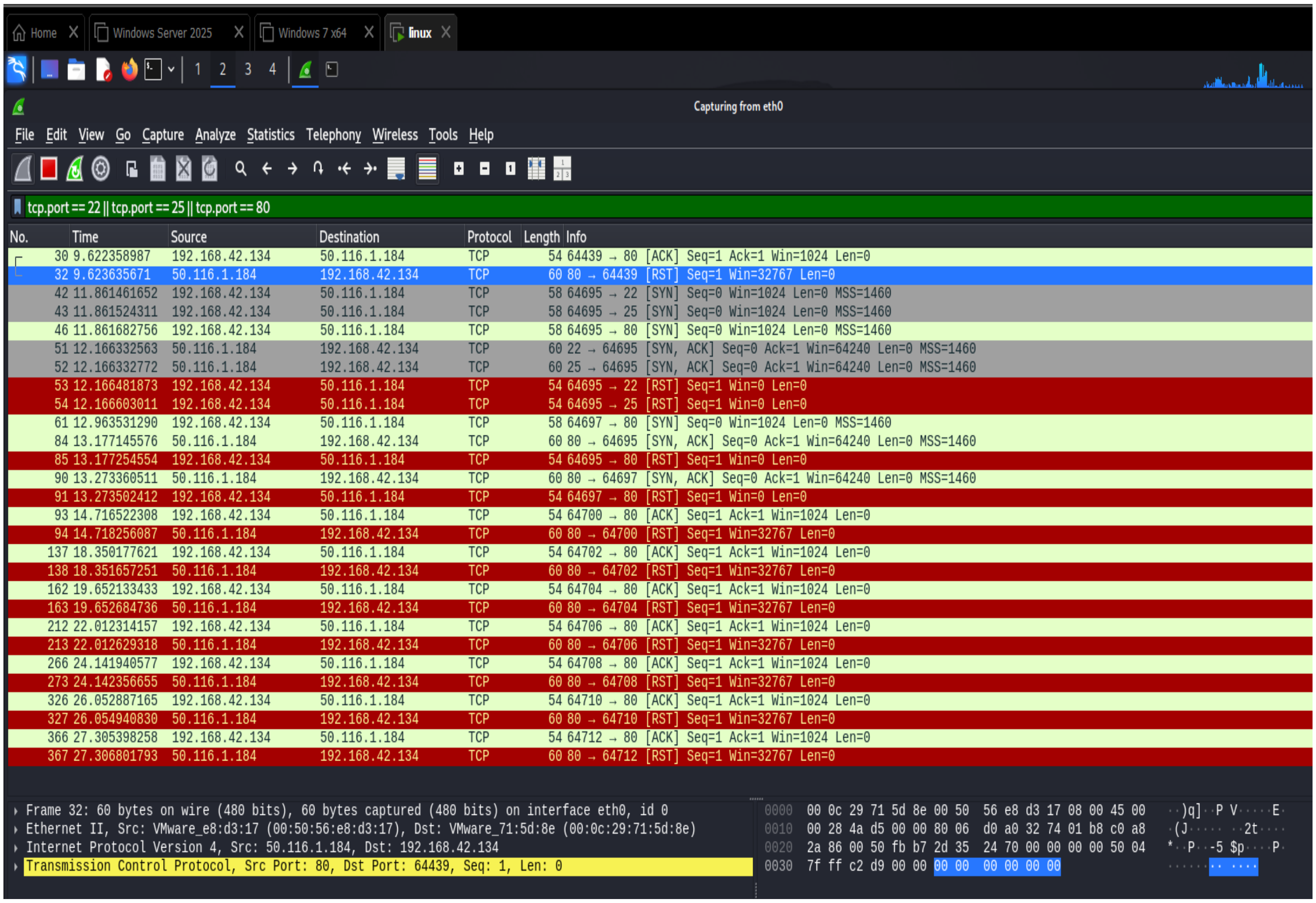


```
┌──(root㉿Rithik)-[/home/rithik]
└─# nmap -T4 -sV scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 09:07 EDT
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.0078s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 918 filtered tcp ports (no-response), 80 closed tcp ports (reset)
PORT   STATE SERVICE    VERSION
22/tcp open  tcpwrapped
80/tcp open  tcpwrapped

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 79.16 seconds
```

Here we run (nmap -sV scanme.nmap.org) command to Detects which services and versions are running.

As the you can see command executed in image is quite different as I have stated above is because (nmap -sV scanme.nmap.org) → takes lots of time to scan all 65,536ports. So, we use -T4 (nmap -T4 -sV scanme.nmap.org)→ to reduce time.

In the fallowing image we capture port number 22/tcp ,80/tcp open state.



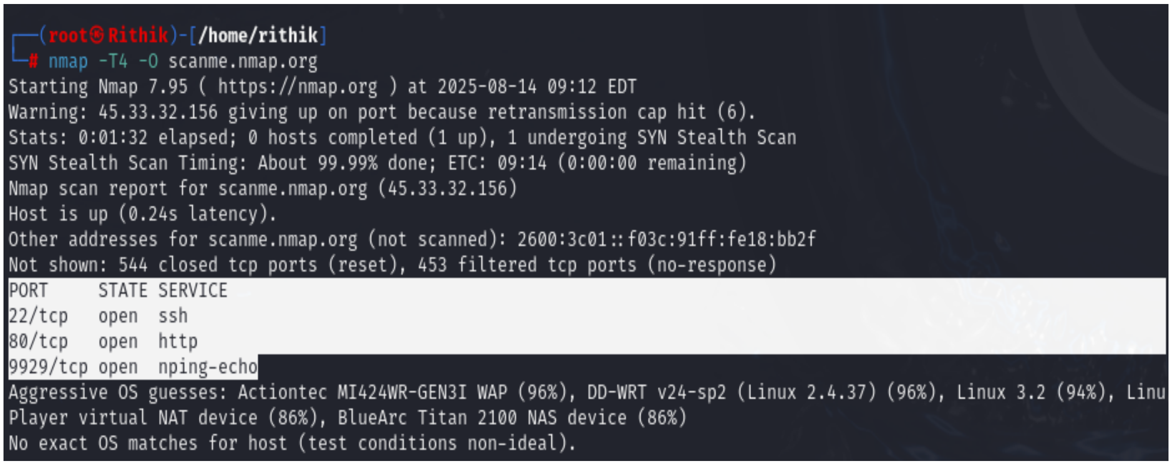
tcp.port == 22 || tcp.port == 25 || tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
30	9.622358987	192.168.42.134	50.116.1.184	TCP	54	64439 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
32	9.623635671	50.116.1.184	192.168.42.134	TCP	60	80 → 64439 [RST] Seq=1 Win=32767 Len=0
42	11.861461652	192.168.42.134	50.116.1.184	TCP	58	64695 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
43	11.861524311	192.168.42.134	50.116.1.184	TCP	58	64695 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
46	11.861682756	192.168.42.134	50.116.1.184	TCP	58	64695 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
51	12.166332563	50.116.1.184	192.168.42.134	TCP	60	22 → 64695 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
52	12.166332772	50.116.1.184	192.168.42.134	TCP	60	25 → 64695 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
53	12.166481873	192.168.42.134	50.116.1.184	TCP	54	64695 → 22 [RST] Seq=1 Win=0 Len=0
54	12.166603011	192.168.42.134	50.116.1.184	TCP	54	64695 → 25 [RST] Seq=1 Win=0 Len=0
61	12.963531290	192.168.42.134	50.116.1.184	TCP	58	64697 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
84	13.177145576	50.116.1.184	192.168.42.134	TCP	60	80 → 64695 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
85	13.177254554	192.168.42.134	50.116.1.184	TCP	54	64695 → 80 [RST] Seq=1 Win=0 Len=0
90	13.273360511	50.116.1.184	192.168.42.134	TCP	60	80 → 64697 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
91	13.273502412	192.168.42.134	50.116.1.184	TCP	54	64697 → 80 [RST] Seq=1 Win=0 Len=0
93	14.716522308	192.168.42.134	50.116.1.184	TCP	54	64700 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
94	14.718256087	50.116.1.184	192.168.42.134	TCP	60	80 → 64700 [RST] Seq=1 Win=32767 Len=0
137	18.350177621	192.168.42.134	50.116.1.184	TCP	54	64702 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
138	18.351657251	50.116.1.184	192.168.42.134	TCP	60	80 → 64702 [RST] Seq=1 Win=32767 Len=0
162	19.652133433	192.168.42.134	50.116.1.184	TCP	54	64704 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
163	19.652684736	50.116.1.184	192.168.42.134	TCP	60	80 → 64704 [RST] Seq=1 Win=32767 Len=0
212	22.012314157	192.168.42.134	50.116.1.184	TCP	54	64706 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
213	22.012629318	50.116.1.184	192.168.42.134	TCP	60	80 → 64706 [RST] Seq=1 Win=32767 Len=0
266	24.141940577	192.168.42.134	50.116.1.184	TCP	54	64708 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
273	24.142356655	50.116.1.184	192.168.42.134	TCP	60	80 → 64708 [RST] Seq=1 Win=32767 Len=0
326	26.052887165	192.168.42.134	50.116.1.184	TCP	54	64710 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
327	26.054940830	50.116.1.184	192.168.42.134	TCP	60	80 → 64710 [RST] Seq=1 Win=32767 Len=0
366	27.305398258	192.168.42.134	50.116.1.184	TCP	54	64712 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0
367	27.306801793	50.116.1.184	192.168.42.134	TCP	60	80 → 64712 [RST] Seq=1 Win=32767 Len=0

```
› Frame 32: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface eth0, id 0
› Ethernet II, Src: VMware_e8:d3:17 (00:50:56:e8:d3:17), Dst: VMware_71:5d:8e (00:0c:29:71:5d:8e)
› Internet Protocol Version 4, Src: 50.116.1.184, Dst: 192.168.42.134
› Transmission Control Protocol, Src Port: 80, Dst Port: 64439, Seq: 1, Len: 0
```

```
0000  00 0c 29 71 5d 8e 00 50  56 e8 d3 17 08 00 45 00   ··)q]··P V·····E·
0010  00 28 4a d5 00 00 80 06  d0 a0 32 74 01 b8 c0 a8   ·(J·····  ··2t····
0020  2a 86 00 50 fb b7 2d 35  24 70 00 00 00 00 50 04   *··P··-5 $p····P·
0030  7f ff c2 d9 00 00 00 00  00 00 00 00               ········ ····
```

4. OS Detection: -



```
┌──(root㉿Rithik)-[/home/rithik]
└─# nmap -T4 -O scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-14 09:12 EDT
Warning: 45.33.32.156 giving up on port because retransmission cap hit (6).
Stats: 0:01:32 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 99.99% done; ETC: 09:14 (0:00:00 remaining)
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.24s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 544 closed tcp ports (reset), 453 filtered tcp ports (no-response)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
9929/tcp open  nping-echo
Aggressive OS guesses: Actiontec MI424WR-GEN3I WAP (96%), DD-WRT v24-sp2 (Linux 2.4.37) (96%), Linux 3.2 (94%), Linu
Player virtual NAT device (86%), BlueArc Titan 2100 NAS device (86%)
No exact OS matches for host (test conditions non-ideal).
```

Here we Attempts to identify the operating system (45.33.32.156) used the command (nmap -T4 -O scanme.nmap.org).
The result is showing that port

- ***22/tcp→ open***
- ***80/tcp→ open***
- ***9929/tcp→ open***

Of an operating system.

5. Scan All Ports

6. Summary: -

I successfully pre-installed and updated and used Nmap to scan a public server. I learned how a port scan looks and how attackers use this technique to find potential entry points into a system.
The scans revealed open ports (SSH, HTTP, SMTP), identified services and versions, and suggested the server runs Linux. By exploring flags like -sV, -O, and -p-, I became comfortable with Nmap commands and result interpretation.