Introduction to Nmap

Cybersecurity Internship – Phase 1, Task 2

Name: Parmar Rakesh
Date:12,AUG,2025
Internship Program: Cybersecurity Internship – UJAR TECH

Introduction

If someone wants to passively or actively scan a system or identify the services running on it, the first tool that often comes to mind for penetration testers and cybersecurity professionals is **Nmap**. For many newcomers in the cybersecurity field, their journey begins with Nmap, an open-source and highly powerful network mapping tool. It can scan any domain or IP to reveal running services, application names and versions, operating systems and their versions, packet filters or firewalls in use, and many other details by using raw IP packets in innovative ways. Designed to rapidly scan large networks yet equally effective for single hosts, Nmap runs on all major operating systems and is available in both command-line and graphical versions.[5]

Basic Guide how to use Nmap

How to install nmap?

Installing nmap in Linux

Open the terminal and run the following commands to get Nmap installed:

- CentOS/Fedora: sudo dnf install nmap
- Ubuntu/Debian: sudo apt-get install nmap

That's it. Nmap is now installed on Linux.

Installing Nmap on Windows

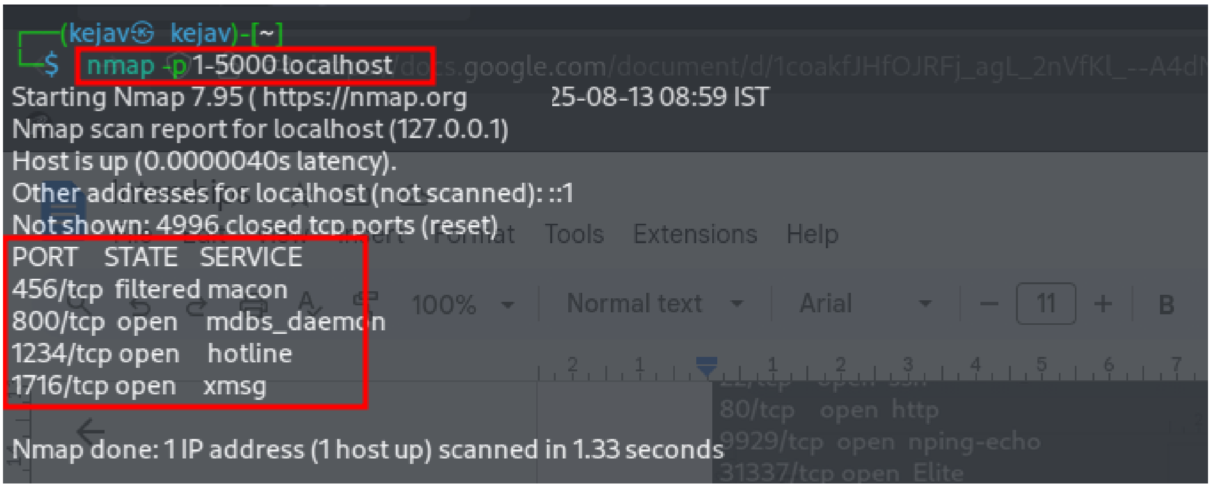
Go to nmap official site download nmap installer for windows.Once you download the installer, execute it and install it. The automated installer should take care of configuring Nmap for you in mere seconds.

Installing Nmap on MacOS

Mac users also have a full automated installer. Just run the Nmap-mpkg file to begin the installation. After a few seconds,

Nmap will be ready on your MacOS.

How nmap works?



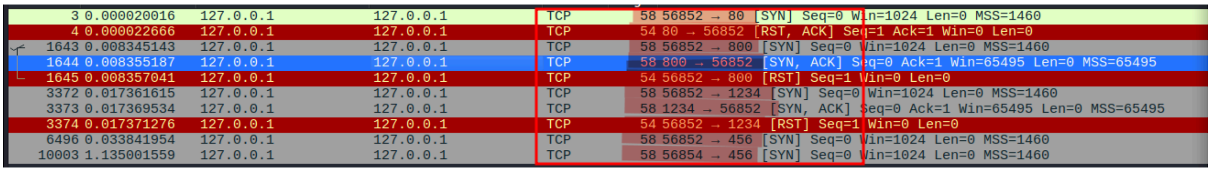
```
┌──(kejav㉿ kejav)-[~]
└─$ nmap -p 1-5000 localhost
Starting Nmap 7.95 ( https://nmap.org        25-08-13 08:59 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000040s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 4996 closed tcp ports (reset)
PORT     STATE    SERVICE
456/tcp  filtered macon
800/tcp  open     mdbs_daemon
1234/tcp open     hotline
1716/tcp open     xmsg

Nmap done: 1 IP address (1 host up) scanned in 1.33 seconds
```

```
80/tcp    open  http
9929/tcp  open  nping-echo
31337/tcp open  Elite
```

As we can see here, in Wireshark, when we observed how Nmap scans a system or host, this image shows the sequence of events. First, Nmap sent a **SYN** flag from its localhost — which is the first step of the TCP handshake[6] — to the ports I had kept open on my system, such as **port 800** and **port 1234**.

Those open ports replied with **SYN + ACK** packets, which is the second step of the handshake. When Nmap received the response from these open ports, it sent **RST** packets instead of completing the handshake. This way, Nmap was able to identify that these ports were open.

Next, as we can see, Nmap also scanned **port 80**, but since I had kept port 80 closed on my system, it replied with an **RST** packet saying it was closed. Nmap then marked it as a closed port and moved on to the next scan.



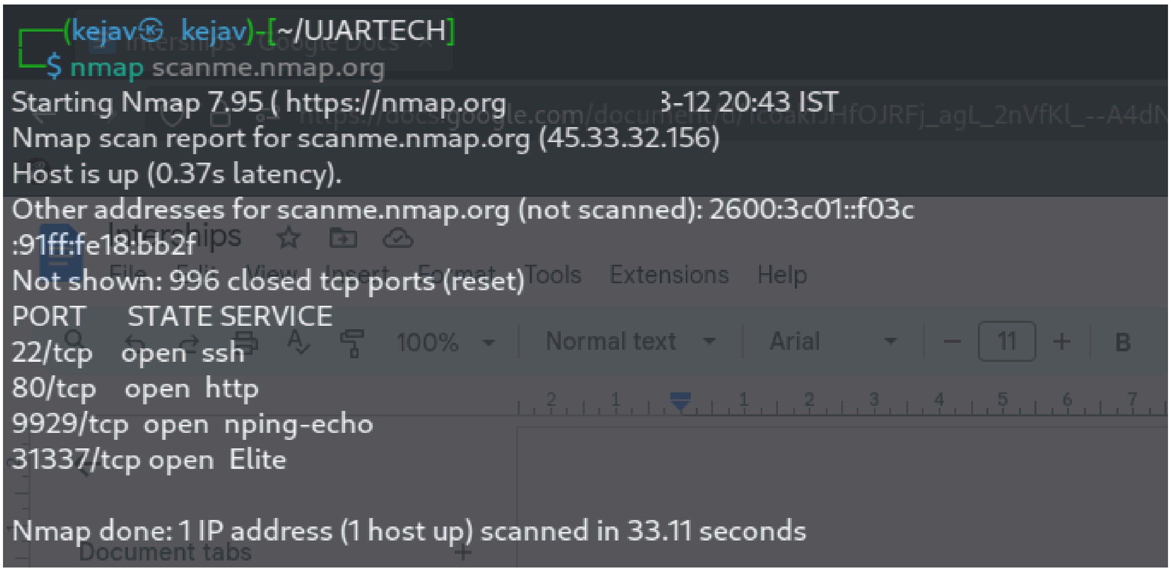
```
    3 0.000020016  127.0.0.1    127.0.0.1    TCP    58 56852 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
    4 0.000022666  127.0.0.1    127.0.0.1    TCP    54 80 → 56852 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
 1643 0.008345143  127.0.0.1    127.0.0.1    TCP    58 56852 → 800 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 1644 0.008355187  127.0.0.1    127.0.0.1    TCP    58 800 → 56852 [SYN, ACK] Seq=0 Ack=1 Win=65495 Len=0 MSS=65495
 1645 0.008357041  127.0.0.1    127.0.0.1    TCP    54 56852 → 800 [RST] Seq=1 Win=0 Len=0
 3372 0.017361615  127.0.0.1    127.0.0.1    TCP    58 56852 → 1234 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
 3373 0.017369534  127.0.0.1    127.0.0.1    TCP    58 1234 → 56852 [SYN, ACK] Seq=0 Ack=1 Win=65495 Len=0 MSS=65495
 3374 0.017371276  127.0.0.1    127.0.0.1    TCP    54 56852 → 1234 [RST] Seq=1 Win=0 Len=0
 6496 0.033841954  127.0.0.1    127.0.0.1    TCP    58 56852 → 456 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10003 1.135001559  127.0.0.1    127.0.0.1    TCP    58 56854 → 456 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

For **456 filtered ports**, the behavior is different. Here, Nmap sends a SYN packet, but it receives neither a reply nor an RST packet. Because of this, Nmap cannot determine the state of the port and marks it as **filtered**. This usually happens because there might be firewall rules applied to that port.

Open Port	**Closed Port**	**Filtered Ports**
client → Server: SYN	client → Server: SYN	client → Server: SYN
Server → Client: SYN-ACK	Server → Client: RST	No reply
Client → Server: RST	Connection closed	–

Nmap Command Examples

1.Basic Scan

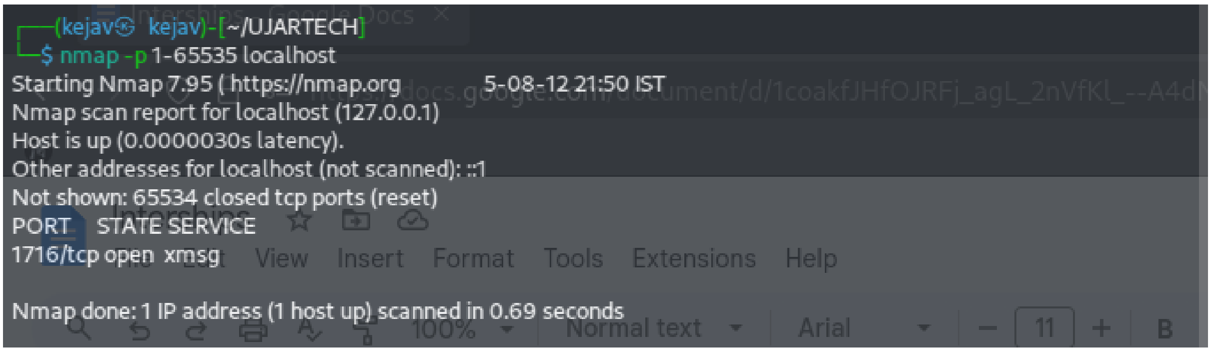


```
┌──(kejav㉿ kejav)-[~/UJARTECH]
└─$ nmap scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org            3-12 20:43 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.37s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c
:91ff:fe18:bb2f
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
9929/tcp open  nping-echo
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 33.11 seconds
```

As you can see in the screenshot above, we have run a simple basic command to scan scanme.nmap.org , and its output is shown just below. From this, we can see that we have received a lot of information that can be used during further testing. This is the most basic scan in Nmap. If you want to scan multiple IPs or domains at the same time, you can run a command like:

- nmap scanme.nmap.org 1.1.1.1 8.8.8.8

2.Scan specific ports or scan entire port ranges

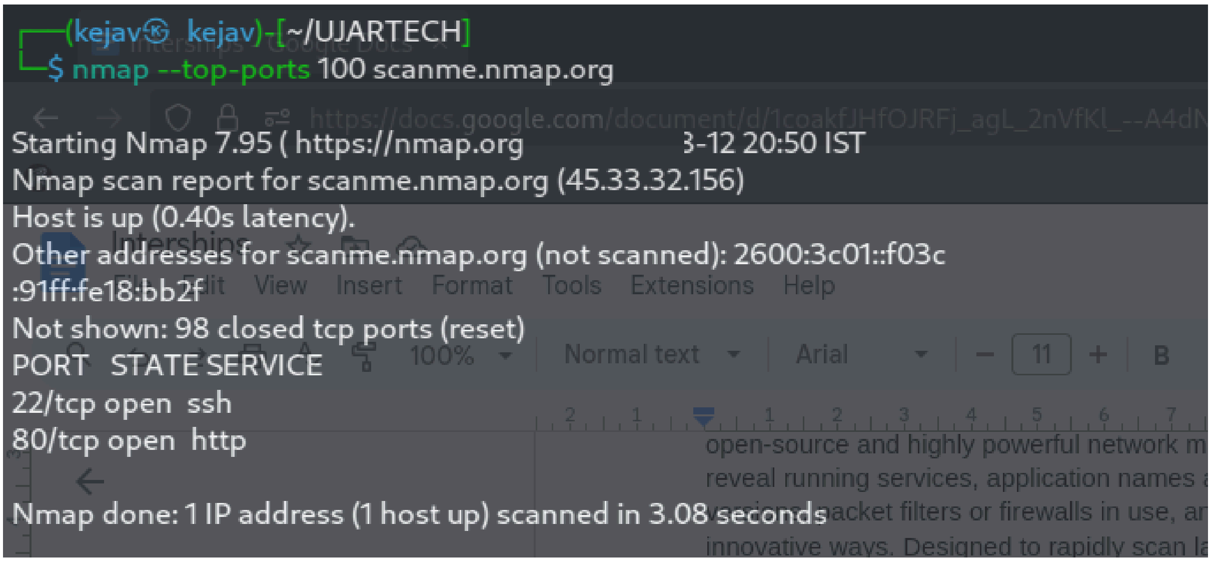


```
┌──(kejav㉿ kejav)-[~/UJARTECH]
└─$ nmap -p 1-65535 localhost
Starting Nmap 7.95 ( https://nmap.org          5-08-12 21:50 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 65534 closed tcp ports (reset)
PORT     STATE SERVICE
1716/tcp open  xmsg

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

In this image, we scanned all 65535 ports for our localhost computer.Nmap is able to scan all possible ports, but you can also scan specific ports, which will report faster results. See below:

- nmap -p 80,443 localhost scanme.nmap.org

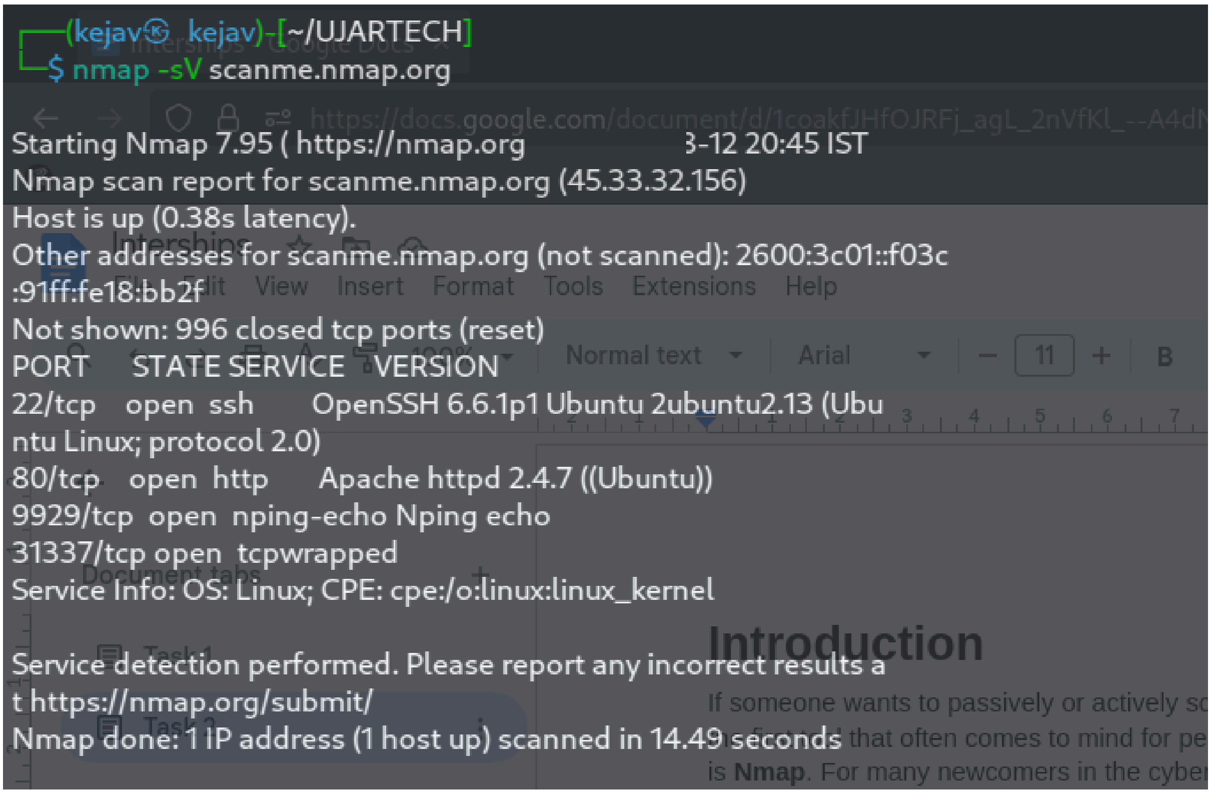
3.Scan the most popular ports



```
┌──(kejav㉿kejav)-[~/UJARTECH]
└─$ nmap --top-ports 100 scanme.nmap.org

Starting Nmap 7.95 ( https://nmap.org           3-12 20:50 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.40s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c
:91ff:fe18:bb2f
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
80/tcp open  http

Nmap done: 1 IP address (1 host up) scanned in 3.08 seconds
```

Using “--top-ports” parameter along with a specific number lets you scan the top X most common ports for that host, as we can in image and replace “100” with the desired number.

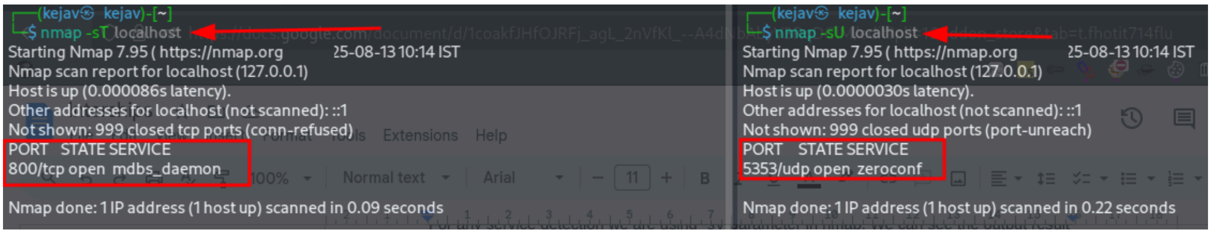
4.Service Detection



```
┌──(kejav㉿kejav)-[~/UJARTECH]
└─$ nmap -sV scanme.nmap.org

Starting Nmap 7.95 ( https://nmap.org           3-12 20:45 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.38s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c
:91ff:fe18:bb2f
Not shown: 996 closed tcp ports (reset)
PORT      STATE SERVICE    VERSION
22/tcp    open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubu
ntu Linux; protocol 2.0)
80/tcp    open  http       Apache httpd 2.4.7 ((Ubuntu))
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results a
t https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.49 seconds
```

For any service detection we are using -sV parameter in nmap. We can see the output result in our image. When we are scanning scanme.nmap.org using -sV parameter we getting service info like port 22/TCP running OpenSSH 6.6.1p1,port 80/TCP has Apache httpd 2.4.7,etc. After Getting this info we search specific exploit for that version.

5.Scan using TCP or UDP protocols



```
┌──(kejav㉿ kejav)-[~]
└─$ nmap -sT localhost
Starting Nmap 7.95 ( https://nmap.org        25-08-13 10:14 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.000086s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE
800/tcp open  mdbs_daemon

Nmap done: 1 IP address (1 host up) scanned in 0.09 seconds
```

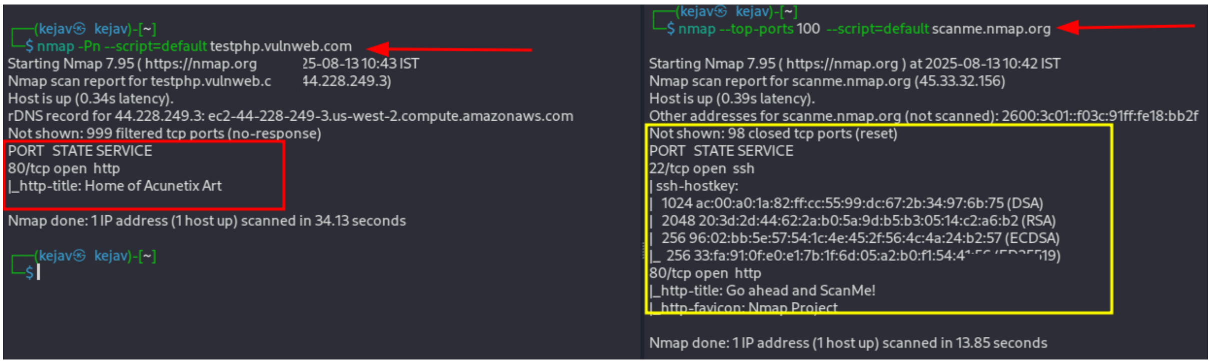
```
┌──(kejav㉿ kejav)-[~]
└─$ nmap -sU localhost
Starting Nmap 7.95 ( https://nmap.org        25-08-13 10:14 IST
Nmap scan report for localhost (127.0.0.1)
Host is up (0.0000030s latency).
Other addresses for localhost (not scanned): ::1
Not shown: 999 closed udp ports (port-unreach)
PORT     STATE SERVICE
5353/udp open  zeroconf

Nmap done: 1 IP address (1 host up) scanned in 0.22 seconds
```

As we can see in this image, we performed separate TCP and UDP scans. The reason for this is that most services usually run on TCP, so Nmap sometimes does not perform a UDP scan by default, and as a result, we may not get any UDP-related output. That's why we can run a separate UDP scan to gather that information.It's also not just about scanning TCP and UDP. If we want to analyze the target more deeply, we can use various specialized scan types, for example:

- **SYN Scan (-sS)** → Fastest scan, half-open handshake.
- **Connect Scan (-sT)** → Full TCP connection.
- **ACK Scan (-sA)** → Used for firewall detection.
- **FIN Scan (-sF)**, **NULL Scan (-sN)**, **Xmas Scan (-sX)** → Stealth scanning methods.

By running these scans, we can collect more detailed information about our target.

6.CVE detection using Nmap

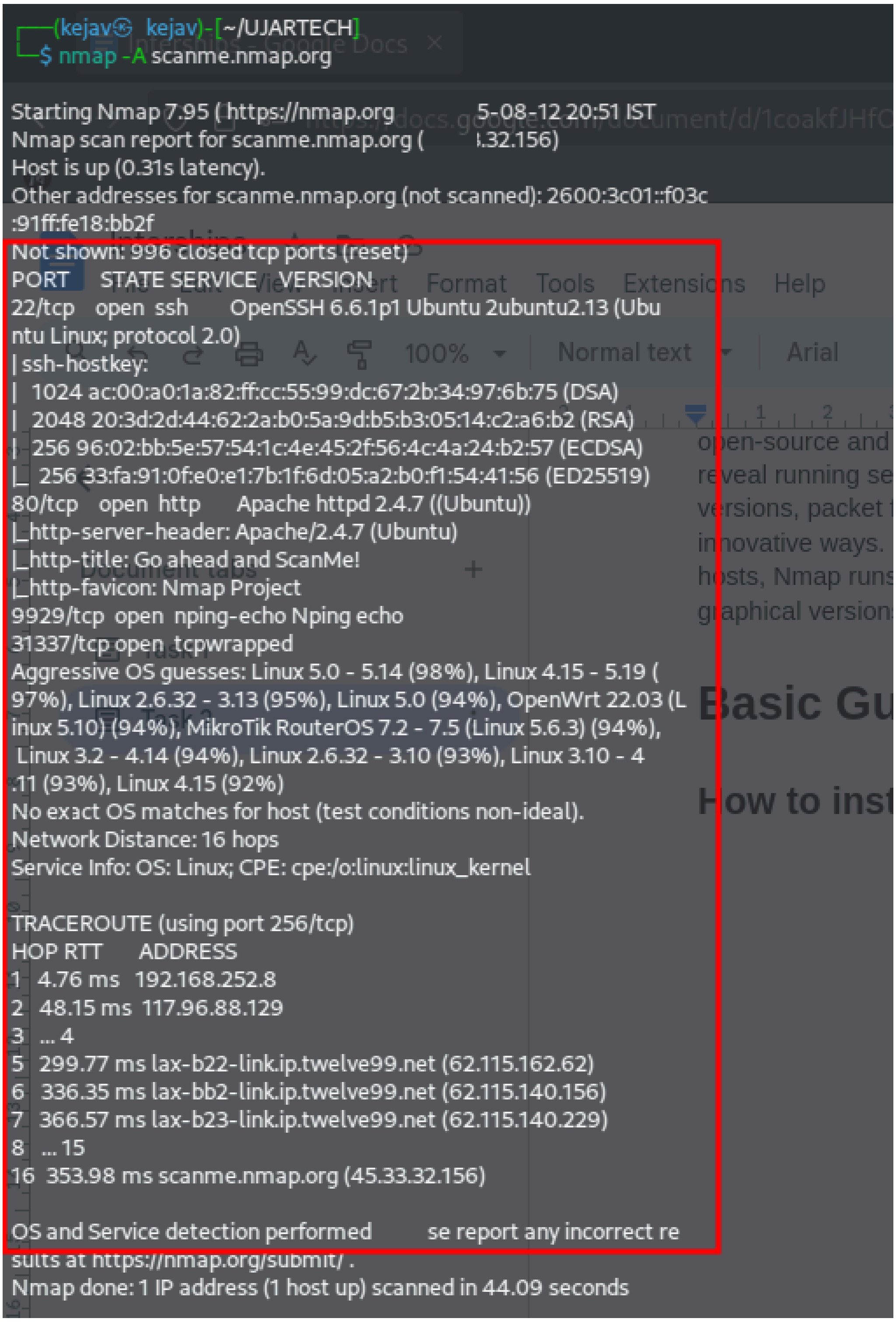


```
┌──(kejav㉿ kejav)-[~]
└─$ nmap -Pn --script=default testphp.vulnweb.com
Starting Nmap 7.95 ( https://nmap.org        25-08-13 10:43 IST
Nmap scan report for testphp.vulnweb.c       44.228.249.3)
Host is up (0.34s latency).
rDNS record for 44.228.249.3: ec2-44-228-249-3.us-west-2.compute.amazonaws.com
Not shown: 999 filtered tcp ports (no-response)
PORT   STATE SERVICE
80/tcp open  http
|_http-title: Home of Acunetix Art

Nmap done: 1 IP address (1 host up) scanned in 34.13 seconds

┌──(kejav㉿ kejav)-[~]
└─$ |
```

```
┌──(kejav㉿ kejav)-[~]
└─$ nmap --top-ports 100  --script=default scanme.nmap.org

Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-13 10:42 IST
Nmap scan report for scanme.nmap.org (45.33.32.156)
Host is up (0.39s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 98 closed tcp ports (reset)
PORT   STATE SERVICE
22/tcp open  ssh
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp open  http
|_http-title: Go ahead and ScanMe!
|_http-favicon: Nmap Project

Nmap done: 1 IP address (1 host up) scanned in 13.85 seconds
```

One of the most powerful features in Nmap is its **NSE (Nmap Scripting Engine) scripts**. These scripts are well-written and beginner-friendly, allowing us to run CVE or vulnerability scans on any domain or IP address.As shown in the image above, I ran two scans where we used Nmap's default NSE scripts to search for CVEs on the target. If we want to go further, we can even create and run our **own custom NSE scripts** using the **Lua programming language**.Additionally, if we want a specific script to run on a specific port, Nmap allows that as well. For more details on creating and using NSE scripts[7] effectively, we can visit Nmap's official documentation to make our scans even more powerful.

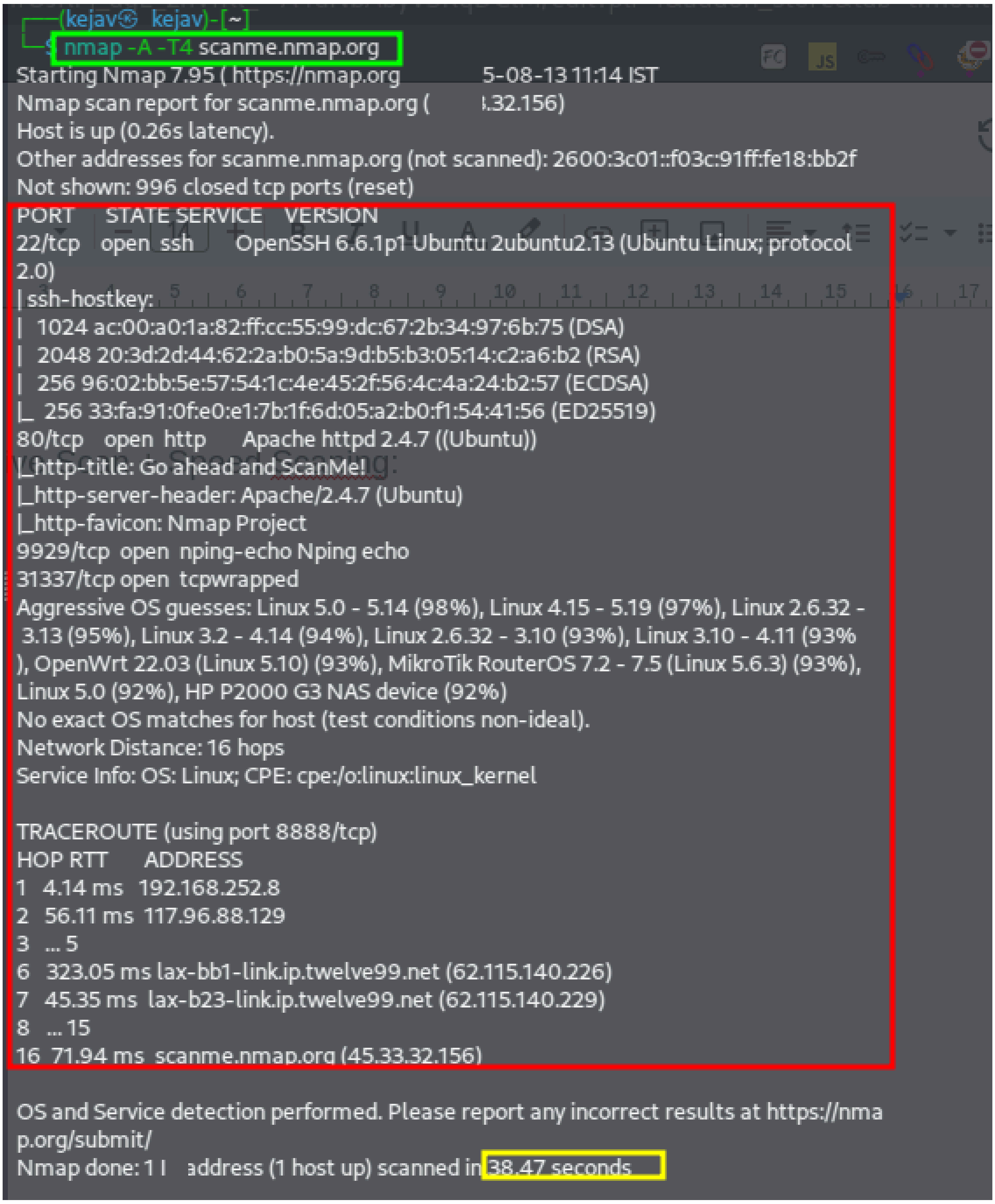
7. Aggressive Scan + Speed Scanning :



```
┌──(kejav㉿ kejav)-[~/UJARTECH]
└─$ nmap -A scanme.nmap.org

Starting Nmap 7.95 ( https://nmap.org        5-08-12 20:51 IST
Nmap scan report for scanme.nmap.org (        .32.156)
Host is up (0.31s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c
:91ff:fe18:bb2f
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubu
ntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp   open  http     Apache httpd 2.4.7 ((Ubuntu))
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-title: Go ahead and ScanMe!
|_http-favicon: Nmap Project
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Aggressive OS guesses: Linux 5.0 - 5.14 (98%), Linux 4.15 - 5.19 (
97%), Linux 2.6.32 - 3.13 (95%), Linux 5.0 (94%), OpenWrt 22.03 (L
inux 5.10) (94%), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3) (94%),
 Linux 3.2 - 4.14 (94%), Linux 2.6.32 - 3.10 (93%), Linux 3.10 - 4
.11 (93%), Linux 4.15 (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 16 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 256/tcp)
HOP RTT       ADDRESS
1   4.76 ms   192.168.252.8
2   48.15 ms  117.96.88.129
3   ... 4
5   299.77 ms lax-b22-link.ip.twelve99.net (62.115.162.62)
6   336.35 ms lax-bb2-link.ip.twelve99.net (62.115.140.156)
7   366.57 ms lax-b23-link.ip.twelve99.net (62.115.140.229)
8   ... 15
16  353.98 ms scanme.nmap.org (45.33.32.156)

OS and Service detection performed        se report any incorrect re
sults at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 44.09 seconds
```

In Nmap, the **Aggressive Scan** is a very useful parameter because it performs multiple tasks in a single run — including running default scripts, OS detection, and service version detection — all at once. Although it is a bit slower compared to other scans, it is highly effective for quickly gathering detailed information about the target in one go.

Speed Scanning:



```
┌──(kejav㉿ kejav)-[~]
└─$ nmap -A -T4 scanme.nmap.org
Starting Nmap 7.95 ( https://nmap.org          5-08-13 11:14 IST
Nmap scan report for scanme.nmap.org (         .32.156)
Host is up (0.26s latency).
Other addresses for scanme.nmap.org (not scanned): 2600:3c01::f03c:91ff:fe18:bb2f
Not shown: 996 closed tcp ports (reset)
PORT     STATE SERVICE    VERSION
22/tcp   open  ssh        OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol
2.0)
| ssh-hostkey:
|   1024 ac:00:a0:1a:82:ff:cc:55:99:dc:67:2b:34:97:6b:75 (DSA)
|   2048 20:3d:2d:44:62:2a:b0:5a:9d:b5:b3:05:14:c2:a6:b2 (RSA)
|   256 96:02:bb:5e:57:54:1c:4e:45:2f:56:4c:4a:24:b2:57 (ECDSA)
|_  256 33:fa:91:0f:e0:e1:7b:1f:6d:05:a2:b0:f1:54:41:56 (ED25519)
80/tcp   open  http       Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Go ahead and ScanMe!
|_http-server-header: Apache/2.4.7 (Ubuntu)
|_http-favicon: Nmap Project
9929/tcp  open  nping-echo Nping echo
31337/tcp open  tcpwrapped
Aggressive OS guesses: Linux 5.0 - 5.14 (98%), Linux 4.15 - 5.19 (97%), Linux 2.6.32 -
 3.13 (95%), Linux 3.2 - 4.14 (94%), Linux 2.6.32 - 3.10 (93%), Linux 3.10 - 4.11 (93%
), OpenWrt 22.03 (Linux 5.10) (93%), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3) (93%),
Linux 5.0 (92%), HP P2000 G3 NAS device (92%)
No exact OS matches for host (test conditions non-ideal).
Network Distance: 16 hops
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE (using port 8888/tcp)
HOP RTT       ADDRESS
1   4.14 ms   192.168.252.8
2   56.11 ms  117.96.88.129
3   ... 5
6   323.05 ms lax-bb1-link.ip.twelve99.net (62.115.140.226)
7   45.35 ms  lax-b23-link.ip.twelve99.net (62.115.140.229)
8   ... 15
16  71.94 ms  scanme.nmap.org (45.33.32.156)

OS and Service detection performed. Please report any incorrect results at https://nma
p.org/submit/
Nmap done: 1 I   address (1 host up) scanned in 38.47 seconds
```

The **Aggressive Scan** in Nmap is a bit slow, but we can speed it up using a built-in parameter that controls scanning speed. However, the faster the scan, the higher the chances of getting **false positives** in the results.

In Nmap, we can adjust the timing from **T0** to **T5**, where the default is **T3** (normal speed). For example, as shown in our two aggressive scan tests:

- Running `-A` alone (default T3) took **44.09 seconds**.

- Running `-A` with `-T4` took **38.47 seconds**.

The difference in output can clearly be seen. In this way, we can increase scanning speed when needed, or slow it down to scan the target more deeply and accurately.

8.Saving Output of Nmap

In cybersecurity, keeping proper notes of what we observe is extremely important because we never know when a piece of information might become useful. For this reason, Nmap provides several output and logging parameters to save and review scan results:

- **`-oN/-oX/-oS/-oG <file>`** ⇒ Save scan output in Normal, XML, Script-Kiddie style, or Grepable format to the specified filename.

- **`-oA <basename>`** ⇒ Save output in all three major formats at once.

In the example shown in the image, I used the **`--vv`** command, which gives a **very verbose** output. This is useful when we want to see exactly which packets are being sent or received at what time, and how they are functioning during the scan.

Conclusion

In conclusion, I would say that Nmap can be used far more effectively if we think beyond our limits. Nmap's name in cybersecurity is not famous without reason — it's an incredibly powerful tool. Even if someone doesn't know how to fully use it, they can start with **OSINT** to explore its capabilities.

For example, when I began, I only knew the basics of Nmap. But by doing OSINT and searching for different ways it can be used, I discovered many features I hadn't known before. One of the most interesting things I learned was the difference between **TCP and UDP scanning**, which I found particularly engaging. I even read an article on Nmap from Recorded Future[8], which helped me analyze it more deeply and understand its real potential.