MIT World Peace University

Vulnerability Identification and Penetration Testing
Third Year B. Tech, Semester 6

Scanning with nmap

Assignment 2

Prepared By

Krishnaraj Thadesar
Cyber Security and Forensics
Batch A1, PA 10

February 9, 2024

Contents

1 Aim

To perform scanning with nmap.

2 Objectives

1. To learn about nmap.
2. To perform live host scanning.

3 Theory

4 Introduction to Nmap

Nmap, short for Network Mapper, is a widely-used open-source tool designed for network exploration and security auditing. It provides a comprehensive view of a network by discovering hosts and services running on them.

4.1 Need/Purpose of Nmap

Nmap serves various purposes in the field of cybersecurity and network management. Its primary objectives include:

- **Host Discovery:** Identifying active hosts on a network, aiding in network mapping.
- **Port Scanning:** Determining open ports on a system, crucial for understanding potential vulnerabilities.
- **Service Version Detection:** Identifying the version and type of services running on open ports.
- **OS Fingerprinting:** Attempting to determine the operating system of target hosts.
- **Vulnerability Assessment:** Assessing potential security risks and vulnerabilities within a network.

4.2 Advantages of Nmap

Nmap offers several advantages that make it a preferred choice in the cybersecurity community:

- **Versatility:** Nmap can be used for a wide range of network exploration and security auditing tasks.
- **Accuracy:** It provides accurate information about hosts, open ports, and services.
- **Scripting Engine:** Nmap's scripting engine allows users to create custom scripts for specific tasks.
- **Community Support:** Being open-source, Nmap benefits from a large and active user community, ensuring continuous improvement.
- **Platform Independence:** Nmap is available on multiple platforms, making it accessible to a diverse range of users.

4.3 Disadvantages of Nmap

Despite its many strengths, Nmap has some limitations and potential drawbacks:

- **Firewall Interference:** Firewalls may block Nmap scans, limiting the tool's effectiveness.
- **Legal and Ethical Concerns:** Improper use of Nmap for unauthorized scanning may lead to legal and ethical issues.
- **False Positives:** In certain scenarios, Nmap might produce false positives, leading to inaccurate assessments.
- **Resource Intensive:** Intensive scanning can consume significant network resources and slow down target systems.
- **Limited Stealth:** While Nmap offers stealthy scanning options, complete stealth is challenging to achieve in some situations.

5 Implementation

5.1 Get ip Address

Syntax

```
$ifconfig
```

Command

```
$ifconfig
```

Purpose

To get the IP Address of the machine.

Output

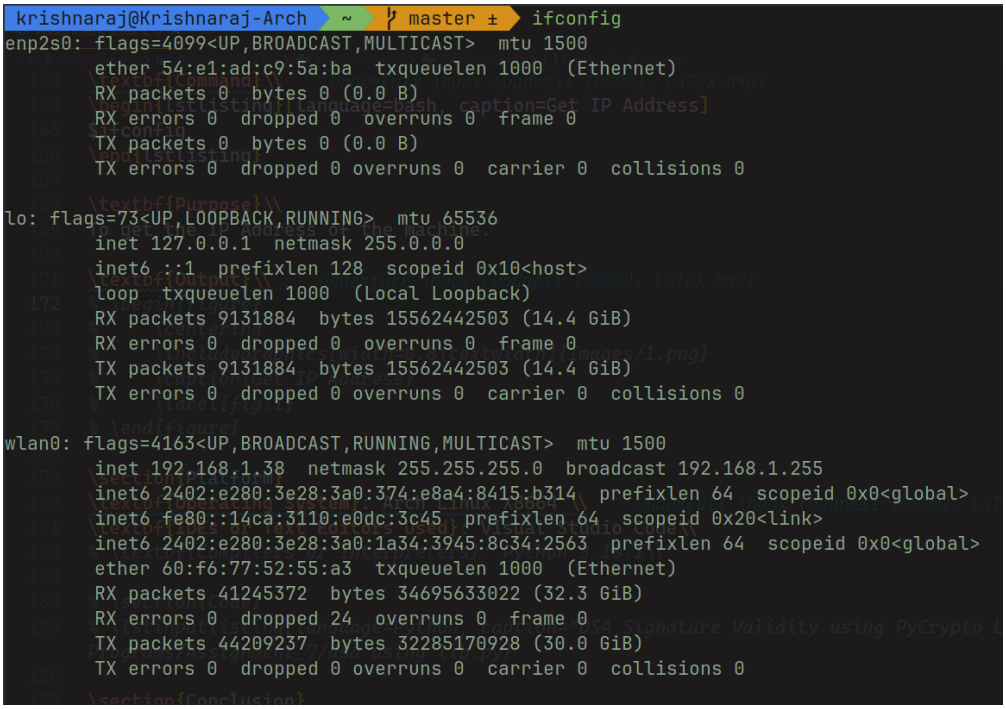

Figure 1: Get IP Address

5.2 Scan 1 port, current IP

5.2.1 Syntax

```
$ nmap -p <port> <ip>
```

Command

```
$ nmap -p 80 192.168.1.38
```

Purpose

To get the IP Address of the machine.

Output

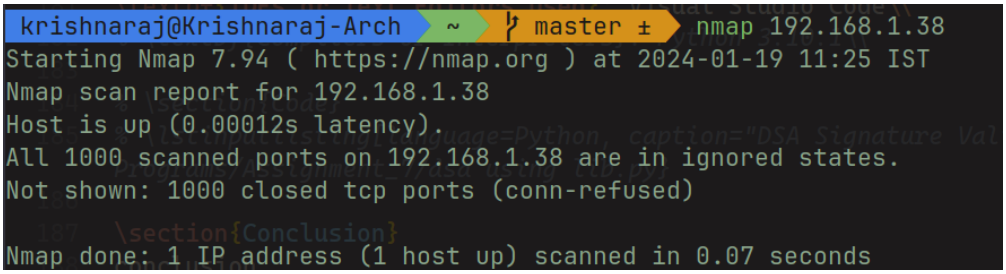

Figure 2: Get IP Address

5.3 Scan any IP

5.3.1 Syntax

```
$ nmap <ip>
```

Command

```
$ nmap 192.168.1.38
```

Purpose

Scan a single ip

Output

Figure 3: Scan google.com

5.4 Scan a range of IPs

5.4.1 Syntax

```
$ nmap <ip range>
```

Command

```
$ nmap 192.168.1.38-40
```

Purpose

To Scan a range of IPs.

Output

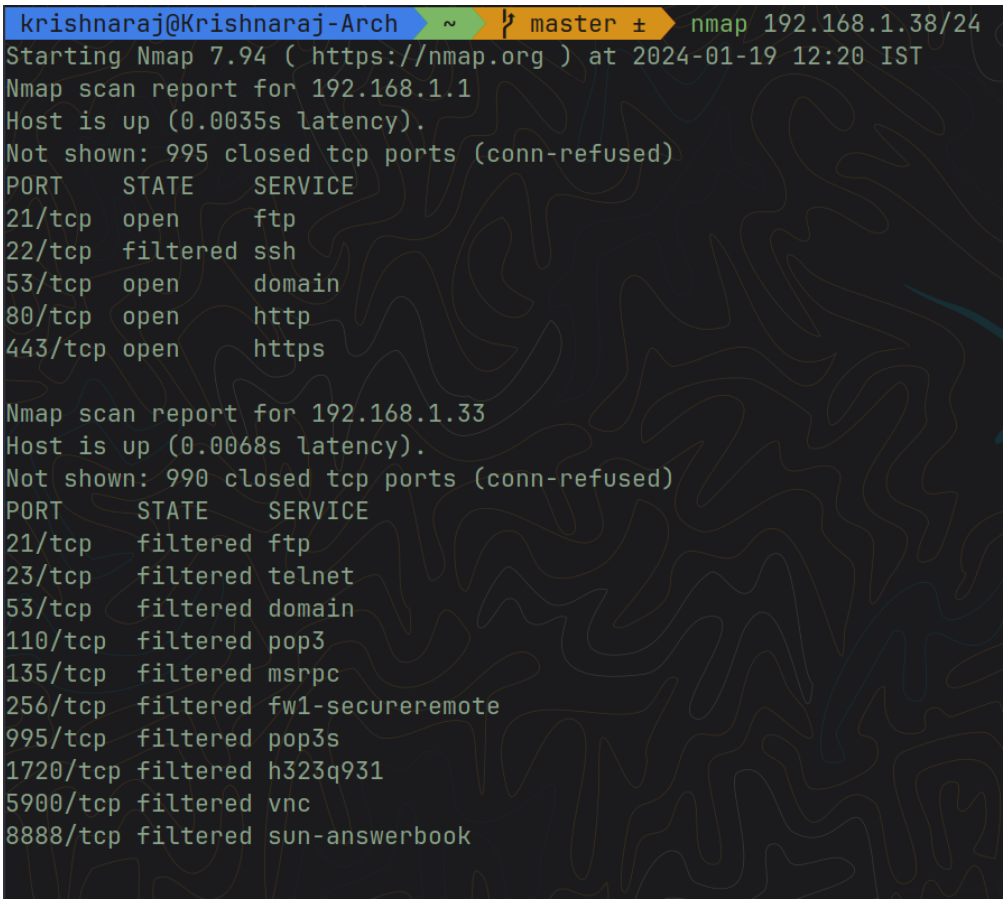

Figure 4: scan range of ips.

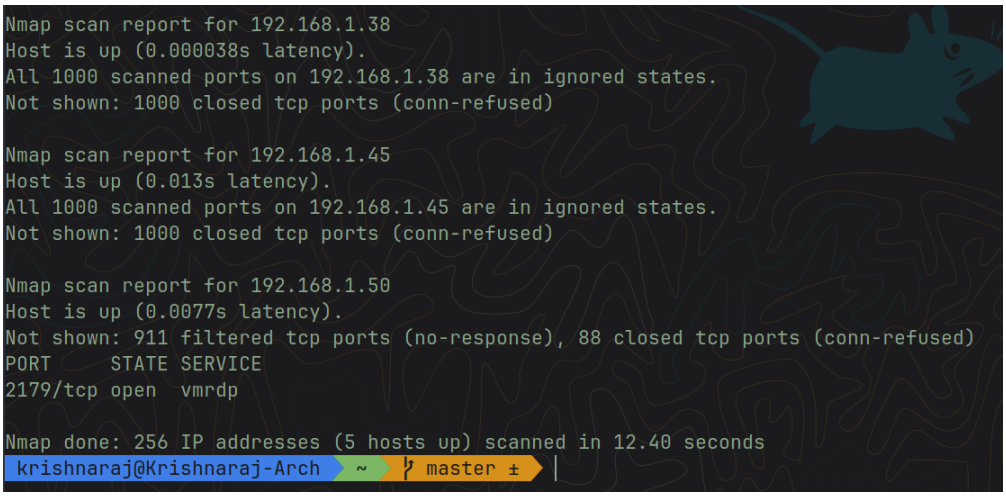

Figure 5: scan range of ips.

5.5 Scan 1 Port

5.5.1 Syntax

```
$ nmap -p <port> <ip>
```

Command

```
$ nmap -p 80 www.example.com
```

Purpose

To perform a scan on a single port.

Output

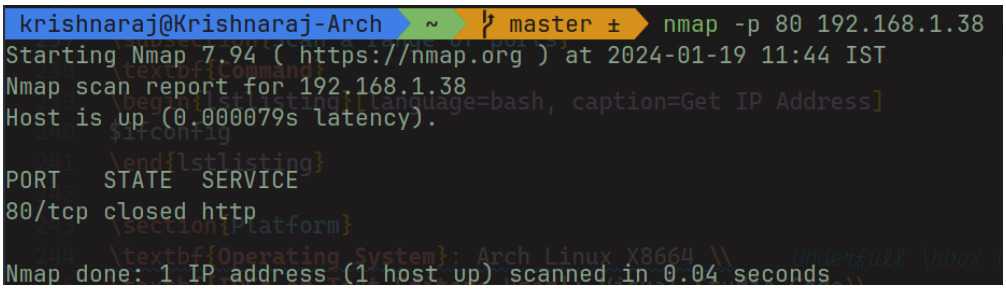

Figure 6: Scan a single port

5.6 Scan a range of ports

5.6.1 Syntax

```
$ nmap -p <port range> <ip>
```

Command

```
$ nmap -p 1-100 www.example.com
```

Purpose

To perform a scan on a range of ports.

Output

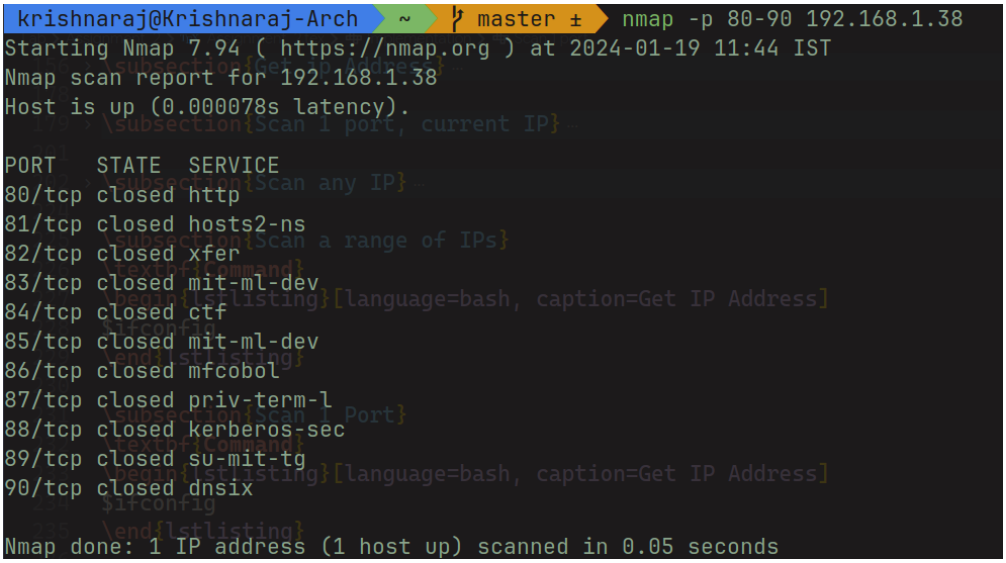

Figure 7: Scan a range of ports

5.7 Fragmented Scan

5.7.1 Syntax

```
$ nmap -F <ip>
```

Command

```
$ nmap -F www.example.com
```

Purpose

Fragmented Scan is used to evade firewalls.

Output

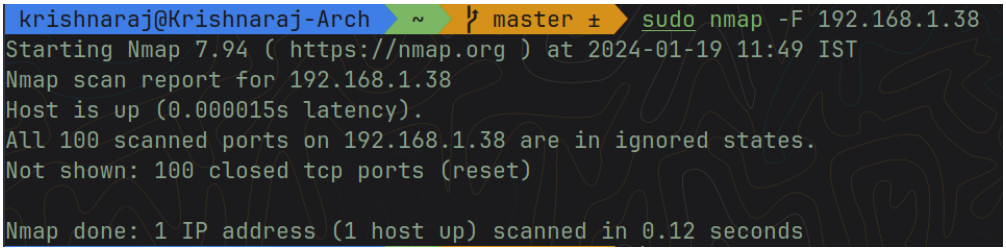

Figure 8: Perform a fragmented scan.

5.8 TCP SYN Scan

5.8.1 Syntax

```
$ nmap -sS <ip>
```

Command

```
$ nmap -sS www.example.com
```

Purpose

To scan a host for open ports using TCP SYN scan.

Output

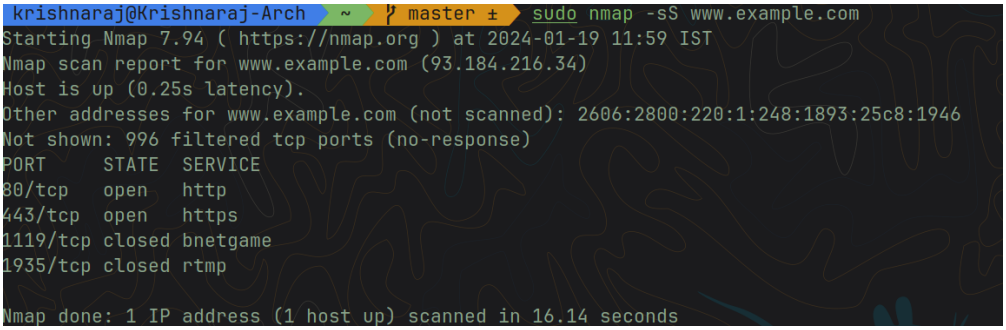

Figure 9: Check if tcp syn scan is possible on a host.

5.9 OS Detection

5.9.1 Syntax

```
$ nmap -O <ip>
```

Command

```
$ nmap -O www.example.com
```

Purpose

To scan operating system of a host.

Output

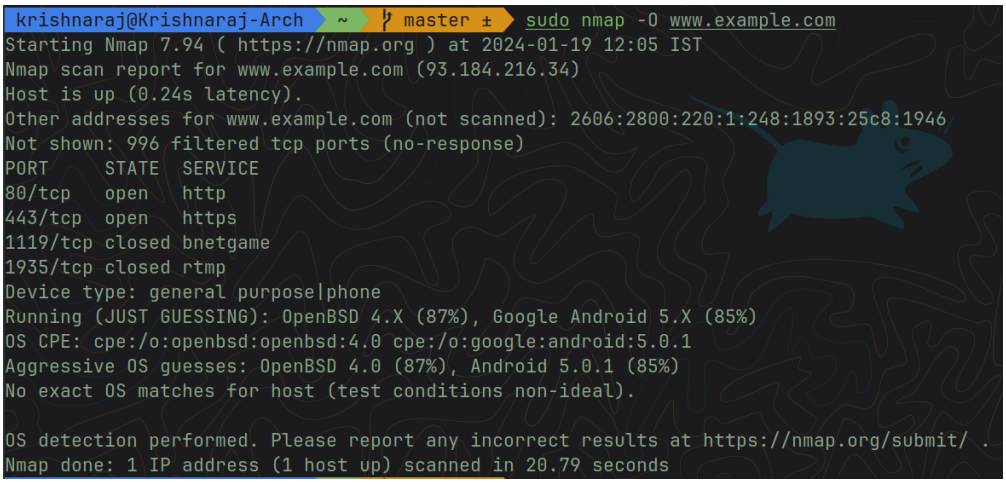

Figure 10: Scan Operating System of example.com

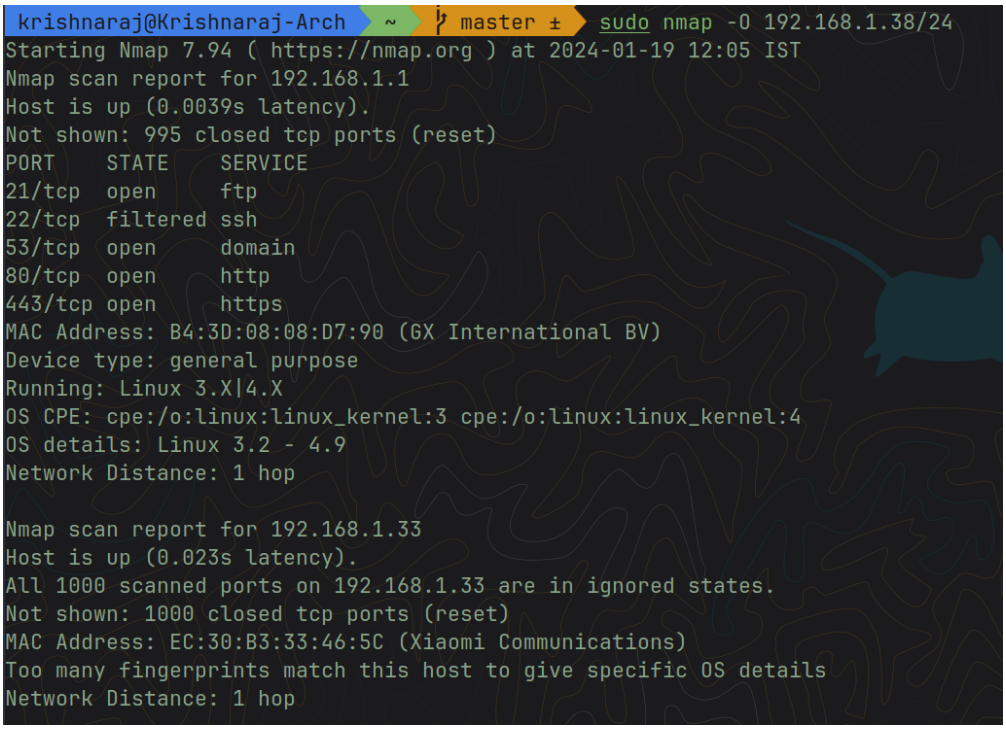

Figure 11: Scan Operating System of host

5.10 Syn Scan for specific ports with ping

5.10.1 Syntax

```
$ sudo nmap -sS -p< <ip>
```

Command

```
$ sudo nmap -sS -p80-90 172.16.182.162
```

Purpose

To perform a syn scan on specific ports with ping.

Output

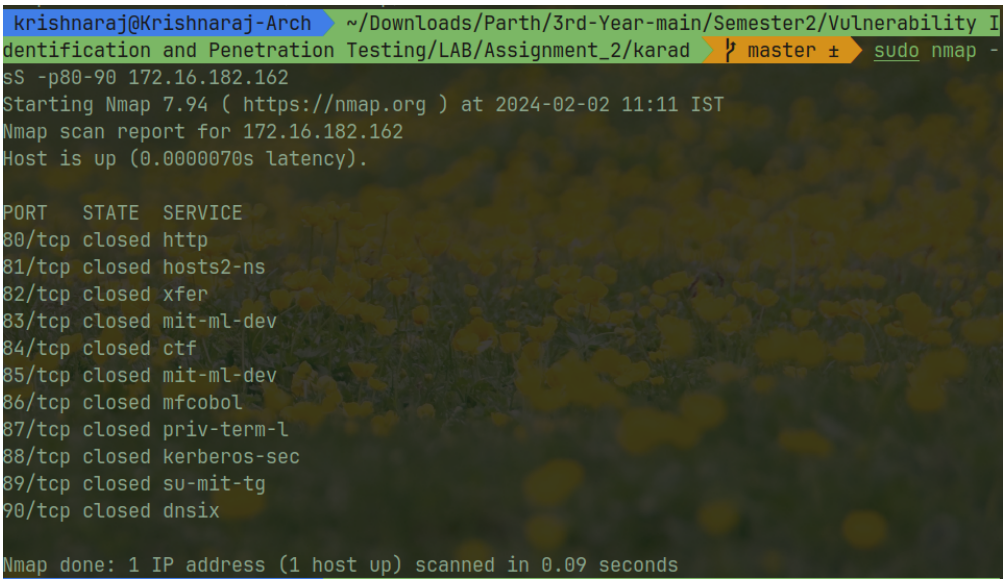

Figure 12: scan with ping

5.11 Syn Scan for specific ports without ping

5.11.1 Syntax

```
$ sudo nmap -sS -Pn -p<port or range> <ip>
```

Command

```
$ sudo nmap -sS -Pn -p40-6000 172.16.182.162
```

Purpose

To scan the open ports of a host without ping to reduce time.

What is the use of ports from 80 to 90?

1. **Port 80:** HTTP (Hypertext Transfer Protocol): Standard port used for serving web pages over the internet.
2. **Port 81:** Alternative HTTP: Sometimes used as an alternative to port 80 for serving HTTP traffic.
3. **Port 82:** Reserved: Not assigned for any specific use by the IANA.
4. **Port 83:** Reserved: Not officially assigned for any specific use.
5. **Port 84:** Commonly Unassigned: Doesn't have a well-known or standardized use.
6. **Port 85:** Commonly Unassigned: No specific use assigned.
7. **Port 86:** Commonly Unassigned: Typically not assigned.
8. **Port 87:** Commonly Unassigned: Not typically used for any specific purpose.
9. **Port 88:** Kerberos: Used by the Kerberos authentication system.
10. **Port 89:** Commonly Unassigned: No well-known or standardized use.

Output

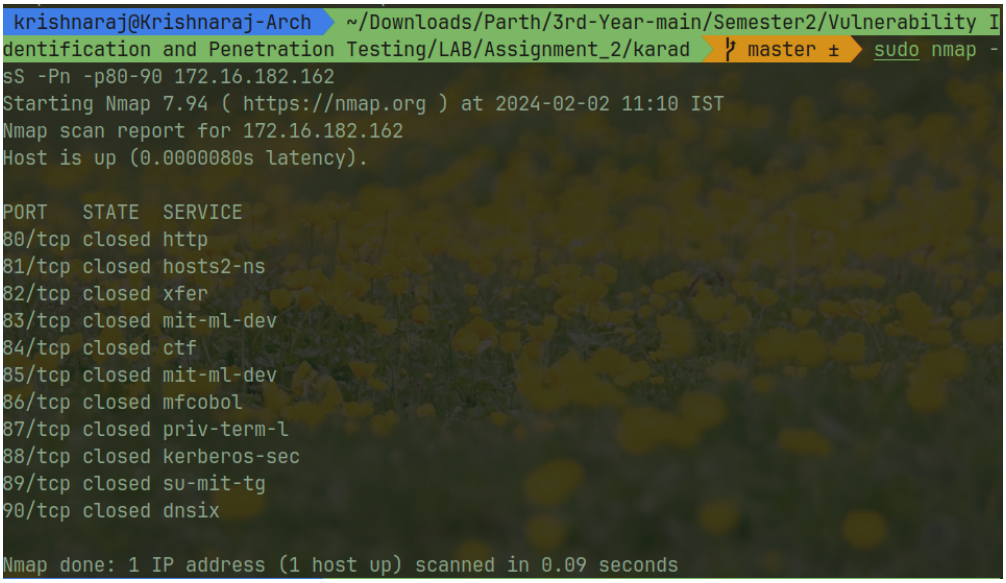

Figure 13: scan without ping

5.12 Nmap Timing Templates

Table 6.3. Timing templates and their effects

	T0	T1	T2	T3	T4	T5
Name	Paranoid	Sneaky	Polite	Normal	Aggressive	Insane
`min-rtt-timeout`	100 ms	100 ms	100 ms	100 ms	100 ms	50 ms
`max-rtt-timeout`	5 minutes	15 seconds	10 seconds	10 seconds	1250 ms	300 ms
`initial-rtt-timeout`	5 minutes	15 seconds	1 second	1 second	500 ms	250 ms
`max-retries`	10	10	10	10	6	2
Initial (and minimum) scan delay (`--scan-delay`)	5 minutes	15 seconds	400 ms	0	0	0
Maximum TCP scan delay	5 minutes	15,000	1 second	1 second	10 ms	5 ms
Maximum UDP scan delay	5 minutes	15 seconds	1 second	1 second	1 second	1 second
`host-timeout`	0	0	0	0	0	15 minutes
`script-timeout`	0	0	0	0	0	10 minutes
`min-parallelism`	Dynamic, not affected by timing templates					
`max-parallelism`	1	1	1	Dynamic	Dynamic	Dynamic
`min-hostgroup`	Dynamic, not affected by timing templates					
`max-hostgroup`	Dynamic, not affected by timing templates					
`min-rate`	No minimum rate limit					
`max-rate`	No maximum rate limit					
`defeat-rst-ratelimit`	Not enabled by default					

Figure 14:

The use of these timing templates is to control the speed of the scan.
From the nmap documentation:

> While the fine-grained timing controls discussed in the previous section are powerful and effective, some people find them confusing. Moreover, choosing the appropriate values can sometimes take more time than the scan you are trying to optimize. So Nmap offers a simpler approach, with six timing templates. You can specify them with the -T option and their number (0–5) or their name. The template names are paranoid (0), sneaky (1), polite (2), normal (3), aggressive (4), and insane (5). The first two are for IDS evasion. Polite mode slows down the scan to use less bandwidth and target machine resources. Normal mode is the default and so -T3 does nothing. Aggressive mode speeds scans up by making the assumption that you are on a reasonably fast and reliable network. Finally insane mode assumes that you are on an extraordinarily fast network or are willing to sacrifice some accuracy for speed.

5.12.1 Syntax

```
$ sudo nmap --packet-trace <ip> -T<0-6>
```

Command

```
$ sudo nmap --packet-trace antibrutus.surge.sh -T5
```

Purpose

To perform packet tracing with timing templates.

Output

```
┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# sudo nmap --packet-trace antibrutus.surge.sh -T5 > t5.txt
NSOCK INFO [0.8220s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.8230s] nsock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [0.8230s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.8230s] nsock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [0.8230s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [0.8230s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [1.9740s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [1.9740s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [1.9740s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [1.9740s] nevent_delete(): nevent_delete on event #34 (type READ)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# cat t5.txt | tail
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
443/tcp  open     https
9001/tcp open     tor-orport
9002/tcp open     dynamid
9003/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 11.02 seconds

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─#
```

Figure 15: With T5

```
┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# sudo nmap --packet-trace antibrutus.surge.sh -T4 > t4.txt
NSOCK INFO [0.8280s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.8280s] nsock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [0.8280s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.8280s] nsock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [0.8280s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [0.8280s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [1.8630s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [1.8630s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [1.8630s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [1.8630s] nevent_delete(): nevent_delete on event #34 (type READ)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# cat t4.txt | tail
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
443/tcp  open     https
9001/tcp open     tor-orport
9002/tcp open     dynamid
9003/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 11.46 seconds

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
```

Figure 16: With T4

```
┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# sudo nmap --packet-trace antibrutus.surge.sh -T4 > t4.txt
NSOCK INFO [0.8280s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [0.8280s] nsock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [0.8280s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [0.8280s] nsock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [0.8280s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [0.8280s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [1.8630s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [1.8630s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [1.8630s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [1.8630s] nevent_delete(): nevent_delete on event #34 (type READ)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# cat t4.txt | tail
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
443/tcp  open     https
9001/tcp open     tor-orport
9002/tcp open     dynamid
9003/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 11.46 seconds

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
```

Figure 17: With T3

```
┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# sudo nmap --packet-trace antibrutus.surge.sh -T2 > t2.txt
NSOCK INFO [1.2110s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [1.2110s] nsock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [1.2120s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [1.2120s] nsock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [1.2120s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [1.2120s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [2.2460s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [2.2460s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [2.2460s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [2.2460s] nevent_delete(): nevent_delete on event #34 (type READ)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# cat t2.txt | tail
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
443/tcp  open     https
9001/tcp open     tor-orport
9002/tcp open     dynamid
9003/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 405.43 seconds
```

Figure 18: With T2

As we can see, the time taken per scan increases as we go from T5 to T2.

5.13 Scannig Vulnerabilities

5.13.1 Syntax

```
$ sudo nmap -Pn --script vuln <ip> -v
```

Command

```
$ sudo nmap -Pn --script vuln www.antibrutus.surge.sh -v
```

Purpose

To scan for vulnerabilities in a host.

Output

```
┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# sudo nmap --packet-trace antibrutus.surge.sh -T2 > t2.txt
NSOCK INFO [1.2110s] nsock_iod_new2(): nsock_iod_new (IOD #1)
NSOCK INFO [1.2110s] nsock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [1.2120s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [1.2120s] nsock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [1.2120s] nsock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [1.2120s] nsock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [2.2460s] nsock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [2.2460s] nsock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [2.2460s] nsock_iod_delete(): nsock_iod_delete (IOD #1)
NSOCK INFO [2.2460s] nevent_delete(): nevent_delete on event #34 (type READ)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali/Documents]
└─# cat t2.txt | tail
25/tcp   filtered smtp
53/tcp   open     domain
80/tcp   open     http
111/tcp  open     rpcbind
443/tcp  open     https
9001/tcp open     tor-orport
9002/tcp open     dynamid
9003/tcp open     unknown

Nmap done: 1 IP address (1 host up) scanned in 405.43 seconds
```

Figure 19: Scan for vulnerabilities

```
Completed NSE at 02:12, 600.75s elapsed
Initiating NSE at 02:12
Completed NSE at 02:12, 0.05s elapsed
Nmap scan report for www.simpli.com (151.101.2.114)
Host is up (0.011s latency).
Other addresses for www.simpli.com (not scanned): 151.101.66.114 151.101.130.114 151.101.194.114
Not shown: 998 filtered tcp ports (no-response)
PORT    STATE SERVICE
80/tcp  open  http
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-csrf: Couldn't find any CSRF vulnerabilities.
443/tcp open  https
|_http-stored-xss: Couldn't find any stored XSS vulnerabilities.
|_http-dombased-xss: Couldn't find any DOM based XSS.
|_http-csrf: Couldn't find any CSRF vulnerabilities.

NSE: Script Post-scanning.
Initiating NSE at 02:12
Completed NSE at 02:12, 0.00s elapsed
Initiating NSE at 02:12
Completed NSE at 02:12, 0.00s elapsed
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 615.48 seconds
           Raw packets sent: 1999 (87.956KB) | Rcvd: 3 (132B)

┌──(root㉿Krishnaraj-Home-PC)-[/home/krishnaraj-kali]
└─#
```

Figure 20: Scan for vulnerabilities

Meaning of Scanned Vulnerabilities and Output

Host Status

- **Host is up (0.011s latency):** Indicates that the host (www.simpli.com in this case) is up and responsive with a latency of 0.011 seconds.

Scanned Ports

- **80/tcp open http:** Port 80 is open and running an HTTP service, typically used for serving web pages.
- **443/tcp open https:** Port 443 is open and running an HTTPS service, which is a secure version of HTTP.

Vulnerability Detection

DOM-based XSS: DOM-based Cross-Site Scripting

Description: DOM-based Cross-Site Scripting (XSS) is a type of XSS attack that occurs when an attacker injects malicious code into a web application, which is then executed by the victim's browser. The attack exploits vulnerabilities in the Document Object Model (DOM) of the web page to manipulate its content.

Stored XSS: Stored Cross-Site Scripting

Description: Stored Cross-Site Scripting (XSS), also known as persistent XSS, occurs when an attacker injects malicious code into a web application, which is then stored and displayed to other users. The injected code is executed when other users visit the affected page, making it a serious security vulnerability.

CSRF: Cross-Site Request Forgery

Description: Cross-Site Request Forgery (CSRF) is an attack that tricks a user into unknowingly executing unwanted actions on a web application in which they are authenticated. The attack occurs when an attacker exploits the user's active session to execute malicious requests without their consent. CSRF attacks can lead to unauthorized actions such as changing account settings or making financial transactions.

NSE Scripts

- NSE scripts were initiated and completed successfully, but no vulnerabilities were detected.

Scan Summary

- Nmap completed scanning 1 IP address with 1 host up in 615.48 seconds.
- 998 TCP ports were filtered (no response), and 2 ports were open (HTTP and HTTPS).

6 Platform

Operating System: Arch Linux X8664
IDEs or Text Editors Used: Visual Studio Code

7 Conclusion

Thus, we have successfully performed scanning with nmap, and learnt about the various options available with nmap.