

**MIT WORLD PEACE UNIVERSITY**

**Attack Research and Documentation  
Fourth Year B. Tech, Semester 5**

---

---

**STUDY OF VARIOUS TOOLS FOR SIMULATED  
ATTACK SCENARIOS**

---

---

**LAB ASSIGNMENT 1 REPORT**

**Prepared By**

**Krishnaraj Thadesar  
Cyber Security and Forensics  
Batch A1, PA 15**

**February 24, 2025**

## Contents

<b>1</b>	<b>Aim</b>	<b>1</b>
<b>2</b>	<b>Wireshark</b>	<b>1</b>
2.1	Installation . . . . .	1
2.2	Working . . . . .	1
2.3	Pros . . . . .	2
2.4	Cons . . . . .	2
2.5	Using Wireshark to Capture Packet and get password details. . . . .	2
<b>3</b>	<b>Nmap</b>	<b>6</b>
3.1	Need/Purpose of Nmap . . . . .	6
3.2	Advantages of Nmap . . . . .	6
3.3	Disadvantages of Nmap . . . . .	7
3.4	Implementation . . . . .	7
3.5	Get ip Address . . . . .	7
3.6	Scan 1 port, current IP . . . . .	8
3.6.1	Syntax . . . . .	8
3.7	Scan any IP . . . . .	9
3.7.1	Syntax . . . . .	9
3.8	Scan a range of IPs . . . . .	9
3.8.1	Syntax . . . . .	9
3.9	Scan 1 Port . . . . .	11
3.9.1	Syntax . . . . .	11
3.10	Scan a range of ports . . . . .	11
3.10.1	Syntax . . . . .	11
3.11	Fragmented Scan . . . . .	12
3.11.1	Syntax . . . . .	12
3.12	TCP SYN Scan . . . . .	13
3.12.1	Syntax . . . . .	13
3.13	OS Detection . . . . .	13
3.13.1	Syntax . . . . .	13
3.14	Syn Scan for specific ports with ping . . . . .	15
3.14.1	Syntax . . . . .	15
3.15	Syn Scan for specific ports without ping . . . . .	15
3.15.1	Syntax . . . . .	15
3.16	Nmap Timing Templates . . . . .	17
3.16.1	Syntax . . . . .	17
3.17	Scannig Vulnerabilities . . . . .	19
3.17.1	Syntax . . . . .	19
3.18	Sweeping IP Ranges for Live host using ARP Scan . . . . .	21
3.18.1	Syntax . . . . .	21
3.19	Sweeping IP Ranges for Live host using ICMP Scan . . . . .	22
3.19.1	Syntax . . . . .	22
3.20	Sweeping IP Ranges for Live host using TCP Scan . . . . .	23
3.20.1	Syntax . . . . .	23
3.21	Sweeping IP Ranges for Live host using UDP Scan . . . . .	24

3.21.1 Syntax . . . . .	24
<b>References</b>	<b>26</b>

## 1 Aim

To study and document the working of various tools used for simulated attack scenarios.

## 2 Wireshark

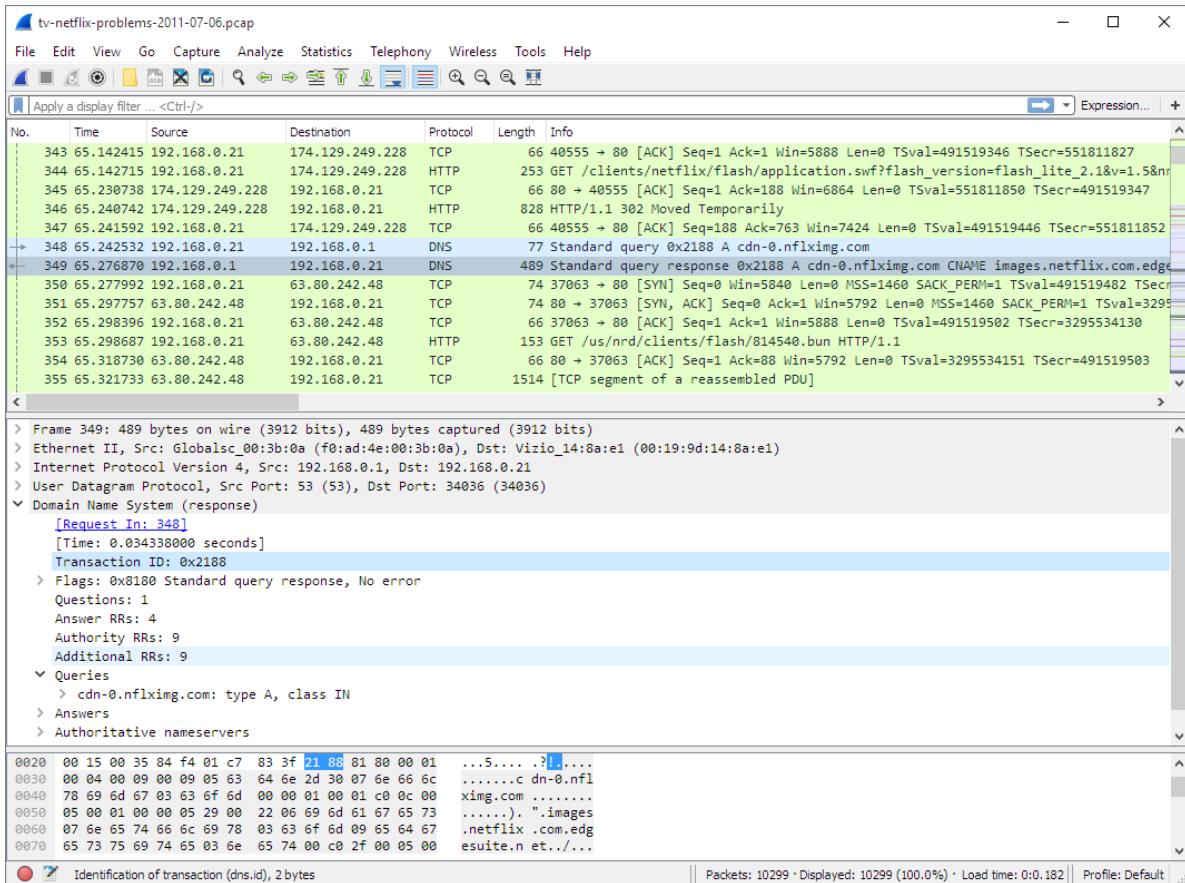


Figure 1: Wireshark GUI

### 2.1 Installation

- Procedure:** Wireshark can be installed on various operating systems, including Windows, macOS, and Linux. Visit the official Wireshark website (<https://www.wireshark.org/>) and follow the installation instructions for your specific platform.
- Dependencies:** Wireshark may require the installation of WinPcap (Windows), libpcap (Linux), or npcap (Windows) for packet capture.

### 2.2 Working

- Wireshark captures and analyzes packets on a network in real-time.
- Users can apply various filters to focus on specific types of traffic.
- The captured data can be displayed in different formats, facilitating detailed protocol analysis.

## 2.3 Pros

- User-friendly interface with powerful features.
- Extensive protocol support for in-depth analysis.
- Active community and regular updates.

## 2.4 Cons

- May consume significant system resources during packet capture.
- Beginners might find the wealth of features overwhelming.
- Limited to the capabilities of the network interface card (NIC).

## 2.5 Using Wireshark to Capture Packet and get password details.

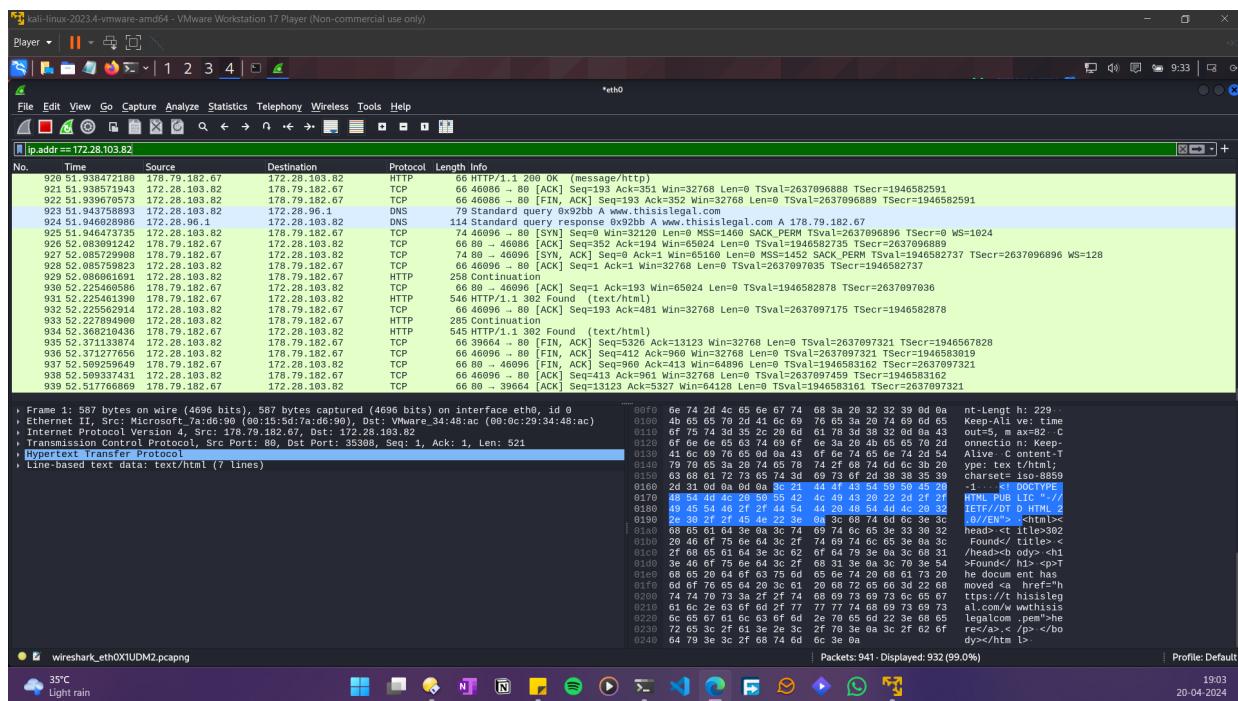


Figure 2: Filtering by target ip

TEST and Demonstration site for **Acunetix Web Vulnerability Scanner**

home | categories | artists | disclaimer | your cart | guestbook | AJAX Demo

search art  go

Browse categories

Browse artists

Your cart

Signup

Your profile

Our guestbook

AJAX Demo

If you are already registered please enter your login information below:

Username : krish

Password :

login

You can also [signup here](#).  
Signup disabled. Please use the username **test** and the password **test**.

Figure 3: Login Request from ip

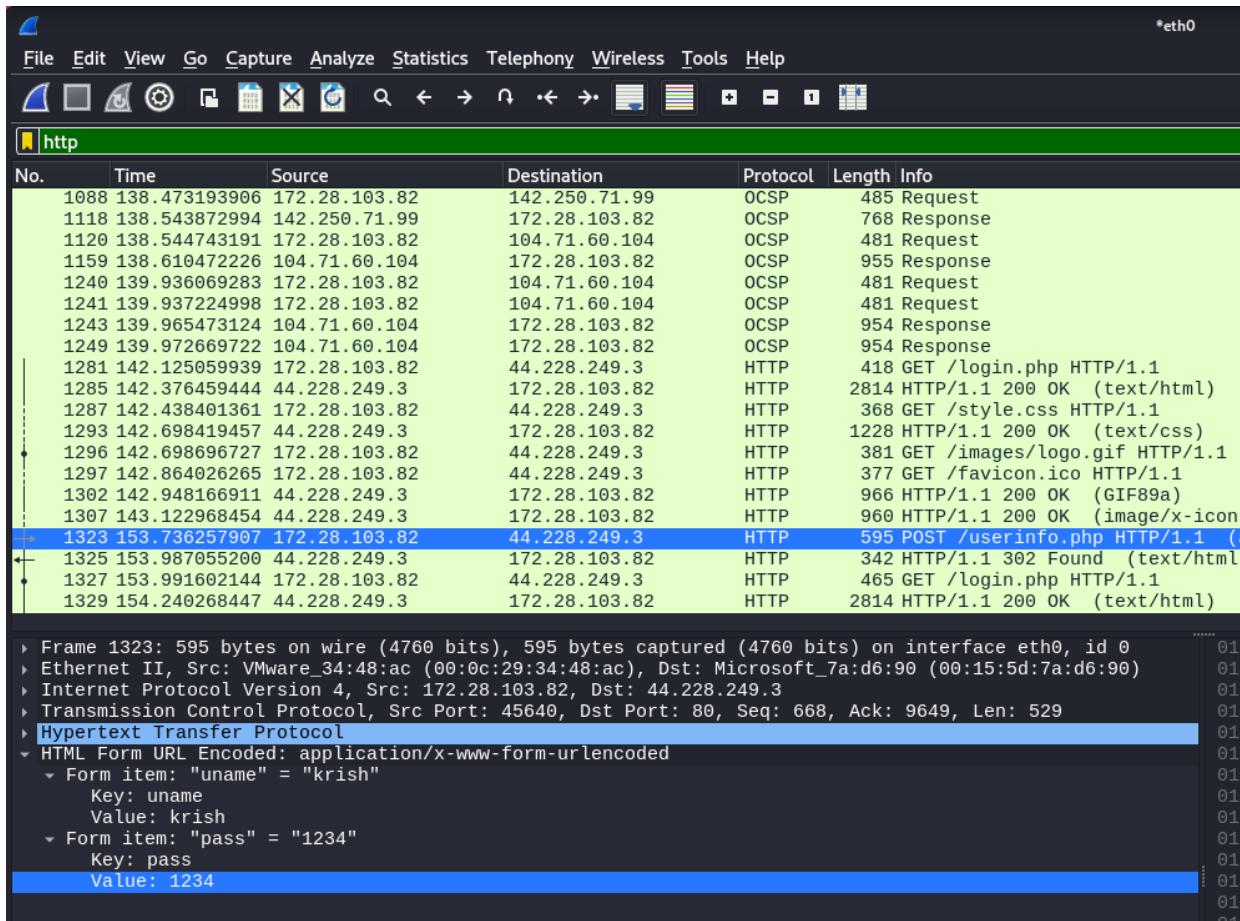


Figure 4: Viewing Http packets to get username and password

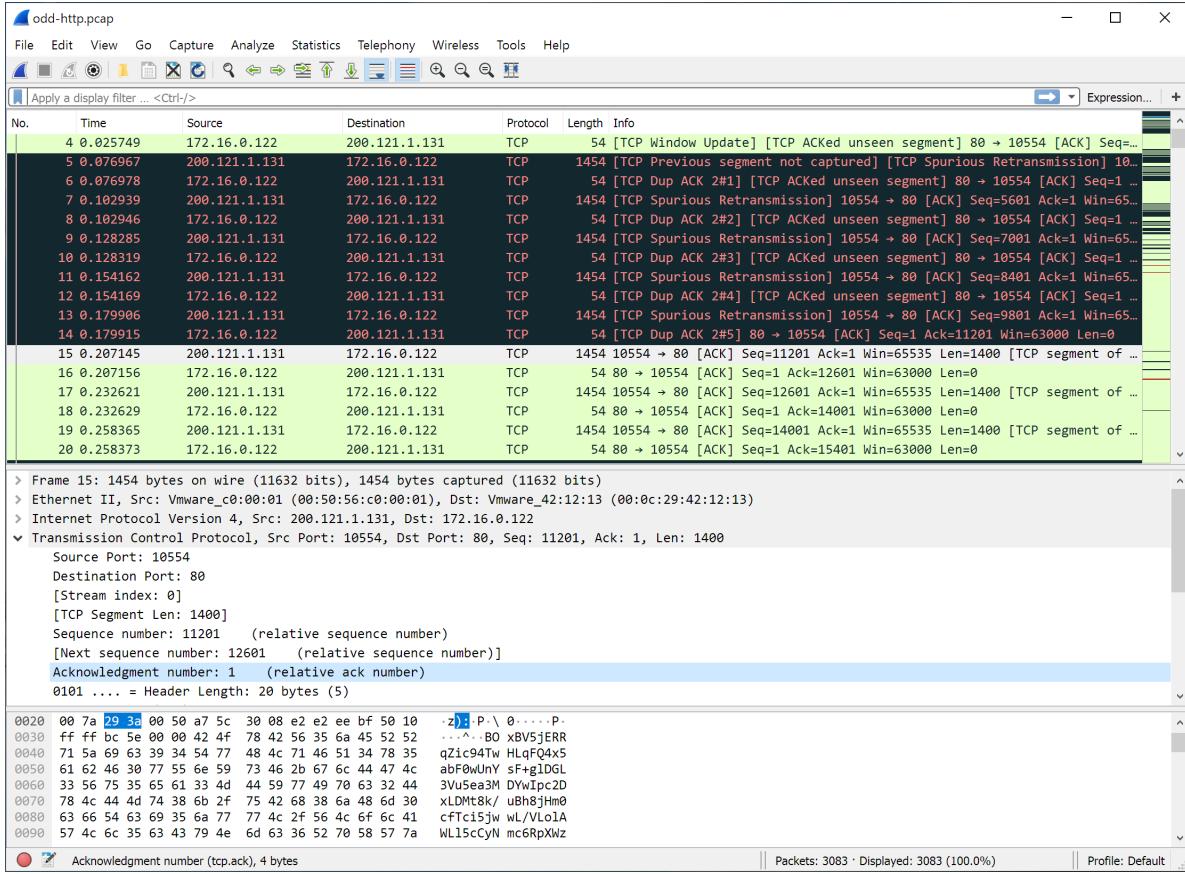


Figure 5: A command-line window executing the aireplay-ng-death tool to deauthenticate clients from a Wi-Fi network.

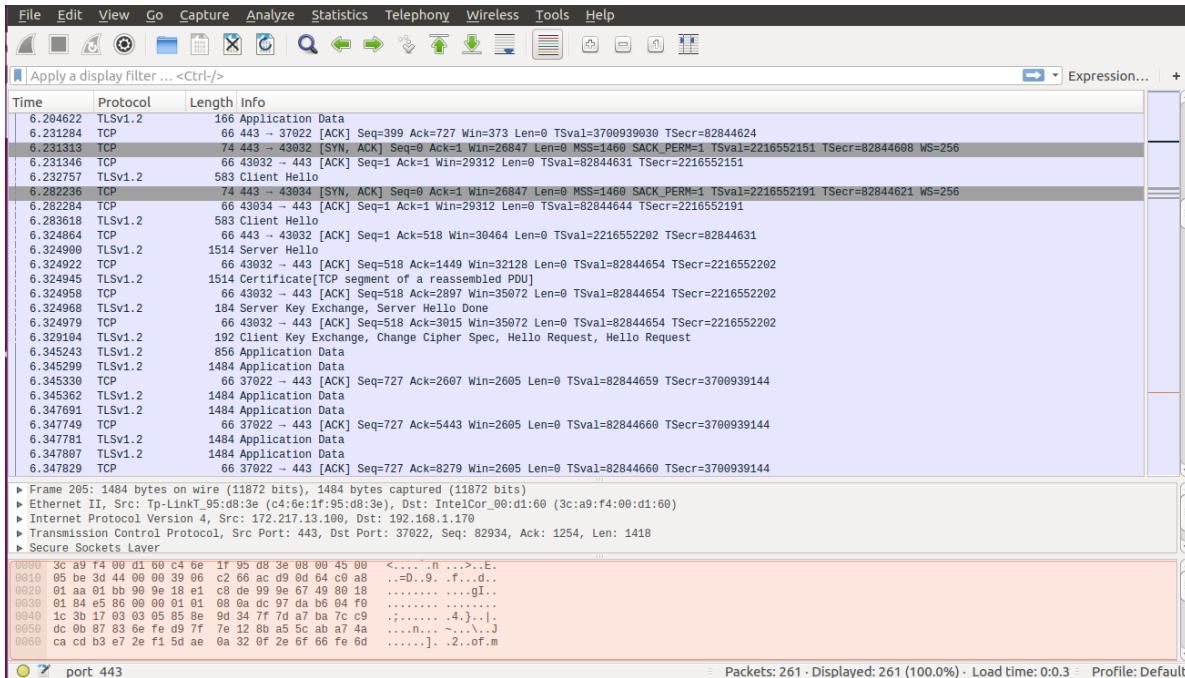


Figure 6: Wi-Fi network scan results on a Kali Linux system.

### 3 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.

#### 3.1 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.

#### 3.2 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.

### **3.3 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.

### **3.4 Implementation**

#### **3.5 Get ip Address**

##### **Syntax**

```
$ifconfig
```

##### **Command**

```
$ifconfig
```

##### **Purpose**

To get the IP Address of the machine.

## Output

```
krishnaraj@Krishnaraj-Arch ~ ↵ master ± ifconfig
enp2s0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 54:e1:ad:c9:5a:ba txqueuelen 1000 (Ethernet)
      RX packets 0 bytes 0 (0.0 B)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 0 bytes 0 (0.0 B)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
      inet6 ::1 prefixlen 128 scopeid 0x10<host>
        loop txqueuelen 1000 (Local Loopback)
          RX packets 9131884 bytes 15562442503 (14.4 GiB)
          RX errors 0 dropped 0 overruns 0 frame 0
          TX packets 9131884 bytes 15562442503 (14.4 GiB)
          TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.38 netmask 255.255.255.0 broadcast 192.168.1.255
      inet6 2402:e280:3e28:3a0:374:e8a4:8415:b314 prefixlen 64 scopeid 0x0<global>
      inet6 fe80::14ca:3110:dc3c45 prefixlen 64 scopeid 0x20<link>
      inet6 2402:e280:3e28:3a0:1a34:3945:8c34:2563 prefixlen 64 scopeid 0x0<global>
    ether 60:f6:77:52:55:a3 txqueuelen 1000 (Ethernet)
      RX packets 41245372 bytes 34695633022 (32.3 GiB)
      RX errors 0 dropped 24 overruns 0 frame 0
      TX packets 44209237 bytes 32285170928 (30.0 GiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

Figure 7: Get IP Address

### **3.6 Scan 1 port, current IP**

### 3.6.1 Syntax

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

## Command

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

## Purpose

To get the IP Address of the machine.

## Output

```
krishnaraj@Krishnaraj-Arch ~ % nmap 192.168.1.38
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:25 IST
Nmap scan report for 192.168.1.38
Host is up (0.00012s latency).
All 1000 scanned ports on 192.168.1.38 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)
187  \section{Conclusion}
Nmap done: 1 IP address (1 host up) scanned in 0.07 seconds
```

Figure 8: Get IP Address

## 3.7 Scan any IP

### 3.7.1 Syntax

```
$ nmap <ip>
```

### Command

```
$ nmap 192.168.1.38
```

### Purpose

Scan a single ip

## Output

```
krishnaraj@Krishnaraj-Arch ~ % nmap www.google.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:30 IST
Nmap scan report for www.google.com (142.250.70.36)
Host is up (0.013s latency).
Other addresses for www.google.com (not scanned): 2404:6800:4009:82a::2004
rDNS record for 142.250.70.36: pbomb-aa-in-f4.1e100.net
Not shown: 998 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
```

Figure 9: Scan google.com

## 3.8 Scan a range of IPs

### 3.8.1 Syntax

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

### Command

```
$ nmap 192.168.1.38-40
```

**Purpose**

To Scan a range of IPs.

**Output**

```
krishnaraj@Krishnaraj-Arch ~ master ± nmap 192.168.1.38/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 12:20 IST
Nmap scan report for 192.168.1.1
Host is up (0.0035s latency).
Not shown: 995 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
21/tcp    open     ftp
22/tcp    filtered ssh
53/tcp    open     domain
80/tcp    open     http
443/tcp   open     https

Nmap scan report for 192.168.1.33
Host is up (0.0068s latency).
Not shown: 990 closed tcp ports (conn-refused)
PORT      STATE    SERVICE
21/tcp    filtered ftp
23/tcp    filtered telnet
53/tcp    filtered domain
110/tcp   filtered pop3
135/tcp   filtered msrpc
256/tcp   filtered fw1-secureremote
995/tcp   filtered pop3s
1720/tcp  filtered h323q931
5900/tcp  filtered vnc
8888/tcp  filtered sun-answerbook
```

Figure 10: scan range of ips.



```

Nmap scan report for 192.168.1.38
Host is up (0.000038s latency).
All 1000 scanned ports on 192.168.1.38 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.1.45
Host is up (0.013s latency).
All 1000 scanned ports on 192.168.1.45 are in ignored states.
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 192.168.1.50
Host is up (0.0077s latency).
Not shown: 911 filtered tcp ports (no-response), 88 closed tcp ports (conn-refused)
PORT      STATE SERVICE
2179/tcp  open  vmsrdp

Nmap done: 256 IP addresses (5 hosts up) scanned in 12.40 seconds
krishnaraj@Krishnaraj-Arch ~ ⚡ master ± |

```

Figure 11: scan range of ips.

### 3.9 Scan 1 Port

#### 3.9.1 Syntax

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

#### Command

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

#### Purpose

To perform a scan on a single port.

#### Output

```

krishnaraj@Krishnaraj-Arch ~ ⚡ master ± nmap -p 80 192.168.1.38
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:44 IST
Nmap scan report for 192.168.1.38
Host is up (0.000079s latency).

PORT      STATE SERVICE
80/tcp    closed http

Nmap done: 1 IP address (1 host up) scanned in 0.04 seconds

```

Figure 12: Scan a single port

### 3.10 Scan a range of ports

#### 3.10.1 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**

```
krishnaraj@Krishnaraj-Arch: ~ % nmap -p 80-90 192.168.1.38
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:44 IST
Nmap scan report for 192.168.1.38
Host is up (0.000078s latency).

PORT      STATE SERVICE
80/tcp    closed http
81/tcp    closed hosts2-ns
82/tcp    closed xfer
83/tcp    closed mit-ml-dev
84/tcp    closed ctf
85/tcp    closed mit-ml-dev
86/tcp    closed mfcobol
87/tcp    closed priv-term-l
88/tcp    closed kerberos-sec
89/tcp    closed su-mit-tg
90/tcp    closed dnsix

Nmap done: 1 IP address (1 host up) scanned in 0.05 seconds
```

Figure 13: Scan a range of ports

### 3.11 Fragmented Scan

#### 3.11.1 Syntax

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

**Command**

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

**Purpose**

Fragmented Scan is used to evade firewalls.

## Output

```
krishnaraj@Krishnaraj-Arch ~ % sudo nmap -F 192.168.1.38
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:49 IST
Nmap scan report for 192.168.1.38
Host is up (0.000015s latency).
All 100 scanned ports on 192.168.1.38 are in ignored states.
Not shown: 100 closed tcp ports (reset)

Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Figure 14: Perform a fragmented scan.

## 3.12 TCP SYN Scan

### 3.12.1 Syntax

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

### Command

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

### Purpose

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

## Output

```
krishnaraj@Krishnaraj-Arch ~ % sudo nmap -sS www.example.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 11:59 IST
Nmap scan report for www.example.com (93.184.216.34)
Host is up (0.25s latency).
Other addresses for www.example.com (not scanned): 2606:2800:220:1:248:1893:25c8:1946
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
1119/tcp  closed bnetgame
1935/tcp  closed rtmp

Nmap done: 1 IP address (1 host up) scanned in 16.14 seconds
```

Figure 15: Check if tcp syn scan is possible on a host.

## 3.13 OS Detection

### 3.13.1 Syntax

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

### Command

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

## Purpose

To scan operating system of a host.

## Output

```
krishnaraj@Krishnaraj-Arch ~ % sudo nmap -O www.example.com
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 12:05 IST
Nmap scan report for www.example.com (93.184.216.34)
Host is up (0.24s latency).
Other addresses for www.example.com (not scanned): 2606:2800:220:1:248:1893:25c8:1946
Not shown: 996 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
443/tcp   open  https
1119/tcp  closed bnetgame
1935/tcp  closed rtmp
Device type: general purpose|phone
Running (JUST GUESSING): OpenBSD 4.X (87%), Google Android 5.X (85%)
OS CPE: cpe:/o:openbsd:openbsd:4.0 cpe:/o:google:android:5.0.1
Aggressive OS guesses: OpenBSD 4.0 (87%), Android 5.0.1 (85%)
No exact OS matches for host (test conditions non-ideal).

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 20.79 seconds
```

Figure 16: Scan Operating System of example.com

```
krishnaraj@Krishnaraj-Arch ~ % sudo nmap -O 192.168.1.38/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-19 12:05 IST
Nmap scan report for 192.168.1.1
Host is up (0.0039s latency).
Not shown: 995 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    filtered ssh
53/tcp    open  domain
80/tcp    open  http
443/tcp   open  https
MAC Address: B4:3D:08:08:D7:90 (GX International BV)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

Nmap scan report for 192.168.1.33
Host is up (0.023s latency).
All 1000 scanned ports on 192.168.1.33 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: EC:30:B3:33:46:5C (Xiaomi Communications)
Too many fingerprints match this host to give specific OS details
Network Distance: 1 hop
```

Figure 17: Scan Operating System of host

### 3.14 Syn Scan for specific ports with ping

#### 3.14.1 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

```
krishnaraj@Krishnaraj-Arch ~ ~/Downloads/Parth/3rd-Year-main/Semester2/Vulnerability Identification and Penetration Testing/LAB/Assignment_2/karad $ master ✘ sudo nmap -sS -p80-90 172.16.182.162
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-02 11:11 IST
Nmap scan report for 172.16.182.162
Host is up (0.0000070s latency).

PORT      STATE SERVICE
80/tcp    closed http
81/tcp    closed hosts2-ns
82/tcp    closed xfer
83/tcp    closed mit-ml-dev
84/tcp    closed ctf
85/tcp    closed mit-ml-dev
86/tcp    closed mfcobol
87/tcp    closed priv-term-l
88/tcp    closed kerberos-sec
89/tcp    closed su-mit-tg
90/tcp    closed dnsix

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

Figure 18: scan with ping

### 3.15 Syn Scan for specific ports without ping

#### 3.15.1 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

```
krishnaraj@Krishnaraj-Arch ~ ~/Downloads/Parth/3rd-Year-main/Semester2/Vulnerability Identification and Penetration Testing/LAB/Assignment_2/karad # master ✘ sudo nmap -sS -Pn -p80-90 172.16.182.162
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-02 11:10 IST
Nmap scan report for 172.16.182.162
Host is up (0.0000080s latency).

PORT      STATE SERVICE
80/tcp    closed http
81/tcp    closed hosts2-ns
82/tcp    closed xfer
83/tcp    closed mit-ml-dev
84/tcp    closed ctf
85/tcp    closed mit-ml-dev
86/tcp    closed mfcobol
87/tcp    closed priv-term-l
88/tcp    closed kerberos-sec
89/tcp    closed su-mit-tg
90/tcp    closed dnsix

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

Figure 19: scan without ping

### 3.16 Nmap Timing Templates

Table 6.3. Timing templates and their effects

Name	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 20:

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.

#### 3.16.1 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 21: 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 22: 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 23: 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

└─[root@Krishnaraj-Home-PC-/home/krishnaraj-kali/Documents]
#
```

Figure 24: With T2

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

### 3.17 Scannig Vulnerabilities

#### 3.17.1 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] nssock_iod_new2(): nssock_iod_new (IOD #1)
NSOCK INFO [1.2110s] nssock_connect_udp(): UDP connection requested to 172.25.144.1:53 (IOD #1) EID 8
NSOCK INFO [1.2120s] nssock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 18
NSOCK INFO [1.2120s] nssock_write(): Write request for 45 bytes to IOD #1 EID 27 [172.25.144.1:53]
NSOCK INFO [1.2120s] nssock_trace_handler_callback(): Callback: CONNECT SUCCESS for EID 8 [172.25.144.1:53]
NSOCK INFO [1.2120s] nssock_trace_handler_callback(): Callback: WRITE SUCCESS for EID 27 [172.25.144.1:53]
NSOCK INFO [2.2460s] nssock_trace_handler_callback(): Callback: READ SUCCESS for EID 18 [172.25.144.1:53] (112 bytes)
NSOCK INFO [2.2460s] nssock_read(): Read request from IOD #1 [172.25.144.1:53] (timeout: -1ms) EID 34
NSOCK INFO [2.2460s] nssock_iod_delete(): nssock_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 25: 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 26: 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).

## 3.18 Sweeping IP Ranges for Live host using ARP Scan

### 3.18.1 Syntax

```
$ nmap -PR -sn <ip range>
```

### Command

```
$ nmap -PR -sn 172.16.182.224/24
```

## Purpose

To scan live hosts using ARP scan.

## Output

```
in ➔ sudo nmap -PR -sn 172.16.179.175/24
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-09 11:28 IST
Nmap scan report for 172.16.179.0
Host is up (0.011s latency).
MAC Address: FF:FF:FF:FF:FF (Unknown)
Nmap scan report for 172.16.179.1
Host is up (0.00037s latency).
MAC Address: 4C:E1:75:B3:70:5F (Cisco Systems)
Nmap scan report for 172.16.179.21
Host is up (0.00029s latency).
MAC Address: 24:BE:05:13:C8:B9 (Hewlett Packard)
Nmap scan report for 172.16.179.22
Host is up (0.00029s latency).
MAC Address: B8:AC:6F:45:F9:71 (Dell)
Nmap scan report for 172.16.179.23
Host is up (0.0036s latency).
MAC Address: EC:B1:D7:60:0D:36 (Hewlett Packard)
Nmap scan report for 172.16.179.26
Host is up (0.00030s latency).
MAC Address: 24:BE:05:0F:AC:9F (Hewlett Packard)
Nmap scan report for 172.16.179.30
Host is up (0.00034s latency).
MAC Address: F8:B1:56:DB:41:CC (Dell)
Nmap scan report for 172.16.179.33
Host is up (0.00081s latency).
MAC Address: 04:0E:3C:96:8F:C5 (HP)
Nmap scan report for 172.16.179.37
Host is up (0.0011s latency).
MAC Address: CC:96:E5:0C:AD:65 (Dell)
Nmap scan report for 172.16.179.38
Host is up (0.00045s latency).
MAC Address: 24:BE:05:13:C7:F3 (Hewlett Packard)
```

Figure 27: To scan live hosts using arp scan.

### 3.19 Sweeping IP Ranges for Live host using ICMP Scan

#### 3.19.1 Syntax

```
$ nmap -PP -sn <ip range>
```

#### Command

```
$ nmap -PP -sn 172.16.182.224
```

## Purpose

To scan live hosts using ICMP scan.

## Output

```
krishnaraj@Krishnaraj-Arch ~ % /run/media/krishnaraj/Classes/University/Third Year/Project/ICMP_Scan/ICMP_Scan.py
in ➤ sudo nmap -PP -sn 172.16.179.175/24

Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-09 11:28 IST
Nmap scan report for 172.16.179.0
Host is up (0.0054s latency).
MAC Address: FF:FF:FF:FF:FF (Unknown)
Nmap scan report for 172.16.179.1
Host is up (0.012s latency).
MAC Address: 4C:E1:75:B3:70:5F (Cisco Systems)
Nmap scan report for 172.16.179.21
Host is up (0.00021s latency).
MAC Address: 24:BE:05:13:C8:B9 (Hewlett Packard)
Nmap scan report for 172.16.179.22
Host is up (0.00021s latency).
MAC Address: B8:AC:6F:45:F9:71 (Dell)
Nmap scan report for 172.16.179.23
Host is up (0.00040s latency).
MAC Address: EC:B1:D7:60:0D:36 (Hewlett Packard)
Nmap scan report for 172.16.179.26
Host is up (0.00026s latency).
MAC Address: 24:BE:05:0F:AC:9F (Hewlett Packard)
Nmap scan report for 172.16.179.30
Host is up (0.00033s latency).
MAC Address: F8:B1:56:DB:41:CC (Dell)
Nmap scan report for 172.16.179.33
Host is up (0.00080s latency).
MAC Address: 04:0E:3C:96:8F:C5 (HP)
Nmap scan report for 172.16.179.37
Host is up (0.0011s latency).
MAC Address: CC:96:E5:0C:AD:65 (Dell)
Nmap scan report for 172.16.179.38
Host is up (0.00059s latency).
```

Figure 28: To scan live hosts using ICMP scan.

## 3.20 Sweeping IP Ranges for Live host using TCP Scan

### 3.20.1 Syntax

```
$ nmap -PA -sn <ip range>
```

### Command

```
$ nmap -PA -sn 172.16.182.224
```

## Purpose

To scan live hosts using TCP scan. This performs 3 way handshaking as opposed to the -sS syn scan option which does not perform 3 way handshaking.

## Output

```
krishnaraj@Krishnaraj-Arch: /run/media/krishnaraj/Classes/University/Thirin ➤ sudo nmap -PA -sn 172.16.179.175/24

Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-09 11:29 IST
Nmap scan report for 172.16.179.0
Host is up (0.012s latency).
MAC Address: FF:FF:FF:FF:FF (Unknown)
Nmap scan report for 172.16.179.1
Host is up (0.00048s latency).
MAC Address: 4C:E1:75:B3:70:5F (Cisco Systems)
Nmap scan report for 172.16.179.21
Host is up (0.00024s latency).
MAC Address: 24:BE:05:13:C8:B9 (Hewlett Packard)
Nmap scan report for 172.16.179.22
Host is up (0.00023s latency).
MAC Address: B8:AC:6F:45:F9:71 (Dell)
Nmap scan report for 172.16.179.23
Host is up (0.011s latency).
MAC Address: EC:B1:D7:60:0D:36 (Hewlett Packard)
Nmap scan report for 172.16.179.26
Host is up (0.00027s latency).
MAC Address: 24:BE:05:0F:AC:9F (Hewlett Packard)
Nmap scan report for 172.16.179.30
Host is up (0.00032s latency).
MAC Address: F8:B1:56:DB:41:CC (Dell)
Nmap scan report for 172.16.179.33
Host is up (0.00054s latency).
MAC Address: 04:0E:3C:96:8F:C5 (HP)
Nmap scan report for 172.16.179.37
Host is up (0.0015s latency).
MAC Address: CC:96:E5:0C:AD:65 (Dell)
Nmap scan report for 172.16.179.38
Host is up (0.00056s latency).
```

Figure 29: To scan live hosts using TCP scan.

### 3.21 Sweeping IP Ranges for Live host using UDP Scan

#### 3.21.1 Syntax

```
$ nmap -PU -sn <ip range>
```

#### Command

```
$ nmap -PU -sn 172.16.182.224
```

**Purpose**

To scan live hosts using UDP scan.

**Output**

```
in ➤ krishnaraj@Krishnaraj-Arch: /run/media/krishnaraj/Classes/University/Thin  
in ➤ sudo nmap -PU -sn 172.16.179.175/24  
  
Starting Nmap 7.94 ( https://nmap.org ) at 2024-02-09 11:29 IST  
Nmap scan report for 172.16.179.0  
Host is up (0.011s latency).  
MAC Address: FF:FF:FF:FF:FF (Unknown)  
Nmap scan report for 172.16.179.1  
Host is up (0.00044s latency).  
MAC Address: 4C:E1:75:B3:70:5F (Cisco Systems)  
Nmap scan report for 172.16.179.21  
Host is up (0.00024s latency).  
MAC Address: 24:BE:05:13:C8:B9 (Hewlett Packard)  
Nmap scan report for 172.16.179.22  
Host is up (0.00023s latency).  
MAC Address: B8:AC:6F:45:F9:71 (Dell)  
Nmap scan report for 172.16.179.23  
Host is up (0.00042s latency).  
MAC Address: EC:B1:D7:60:0D:36 (Hewlett Packard)  
Nmap scan report for 172.16.179.26  
Host is up (0.00030s latency).  
MAC Address: 24:BE:05:0F:AC:9F (Hewlett Packard)  
Nmap scan report for 172.16.179.30  
Host is up (0.00042s latency).  
MAC Address: F8:B1:56:DB:41:CC (Dell)  
Nmap scan report for 172.16.179.33  
Host is up (0.00066s latency).  
MAC Address: 04:0E:3C:96:8F:C5 (HP)  
Nmap scan report for 172.16.179.37  
Host is up (0.0013s latency).  
MAC Address: CC:96:E5:0C:AD:65 (Dell)  
Nmap scan report for 172.16.179.38
```

Figure 30: To scan live hosts using UDP scan.

## **References**

- [1] Wireshark.  
Website: <https://www.wireshark.org/>
- [2] Tshark.  
Website: <https://www.wireshark.org/docs/man-pages/tshark.html>
- [3] Tcpdump.  
Website: <https://www.tcpdump.org/>
- [4] AirCrack-ng.  
Website: <https://www.aircrack-ng.org/>
- [5] AirSnort.  
Website: <https://sourceforge.net/projects/airsnort/>