

MIT WORLD PEACE UNIVERSITY

Attack Research and Documentation
Fourth Year B. Tech, Semester 8

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

1 Aim

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

2 Wireshark

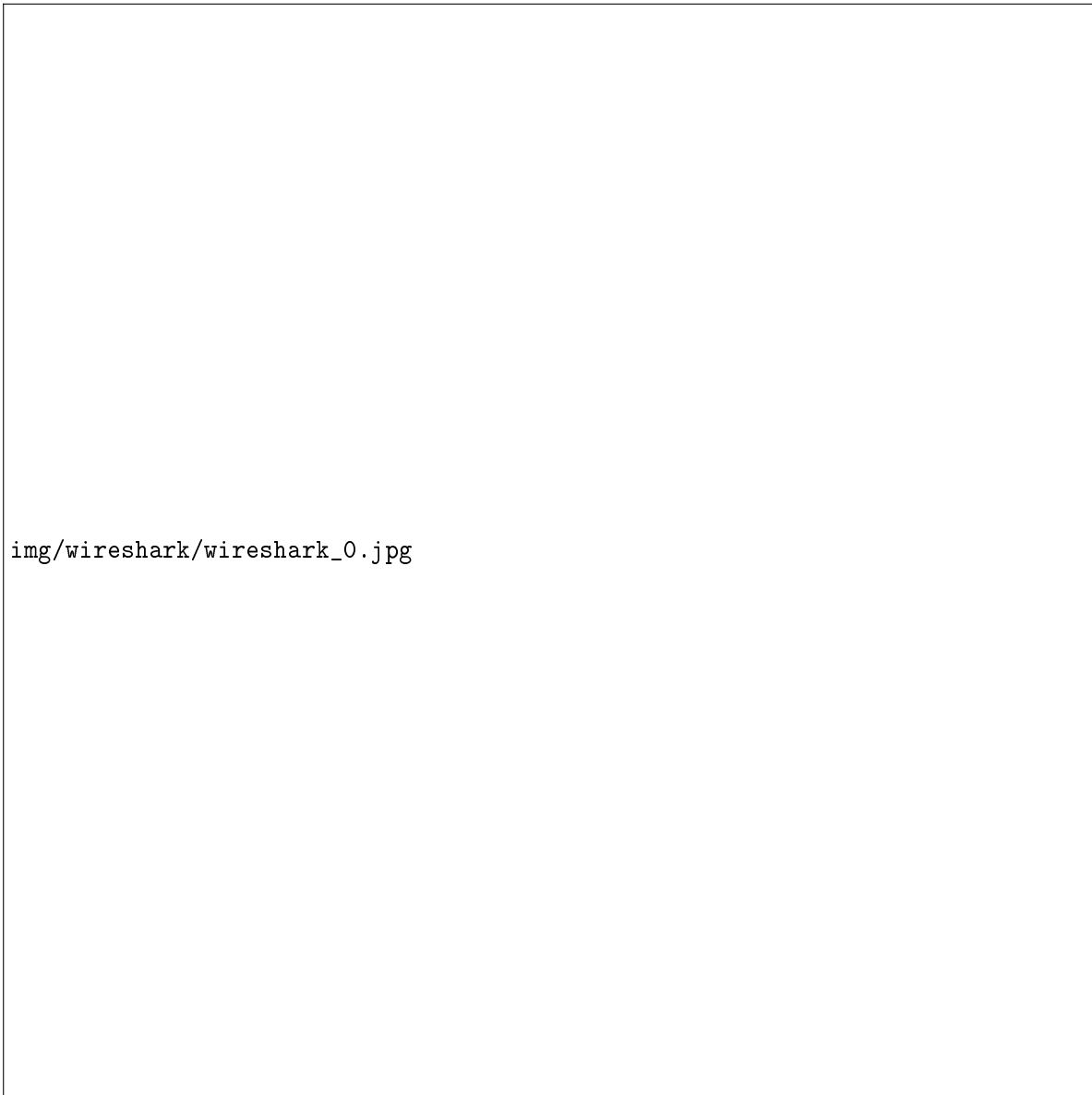


Figure 1: Wireshark GUI

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.

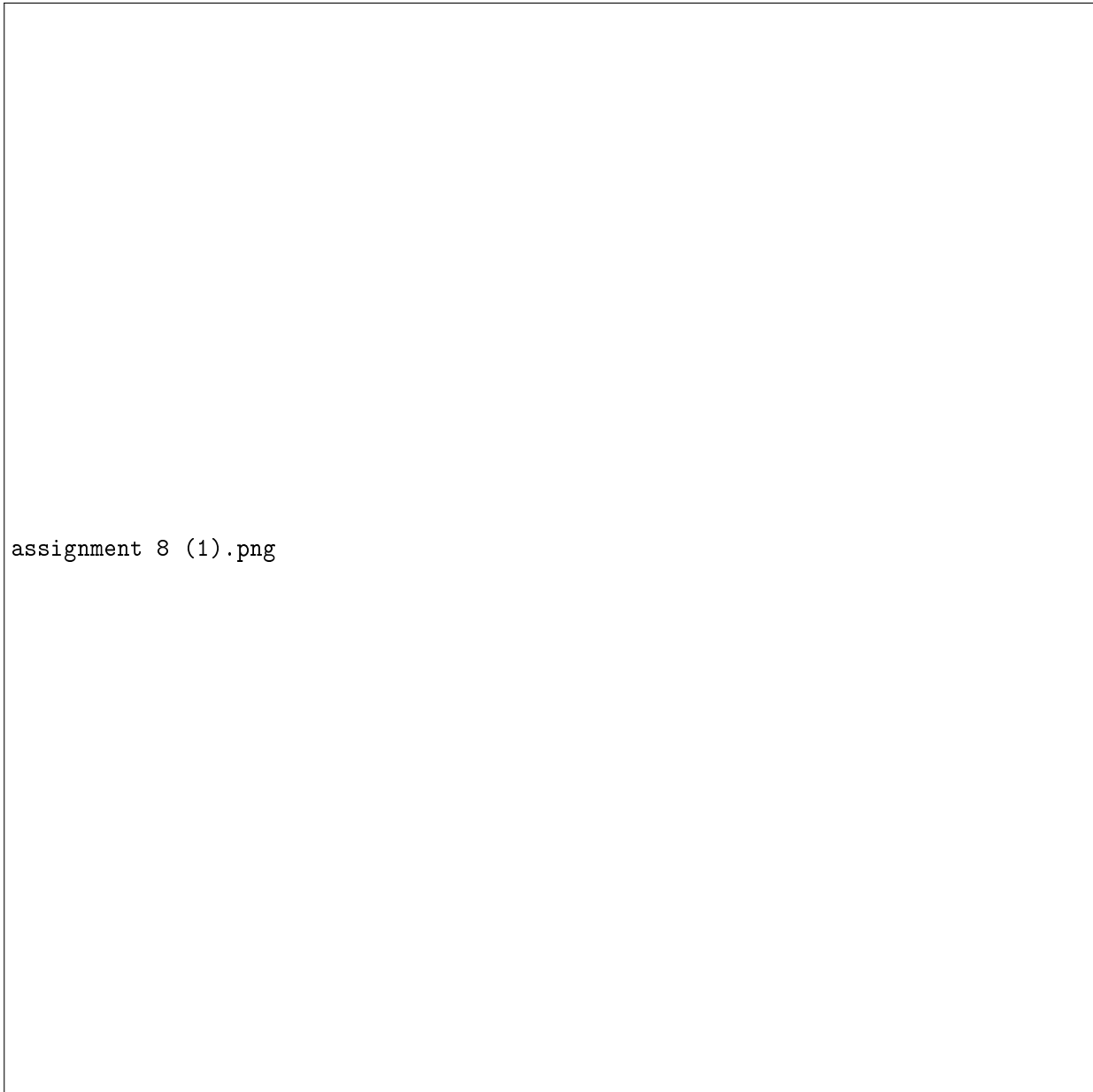
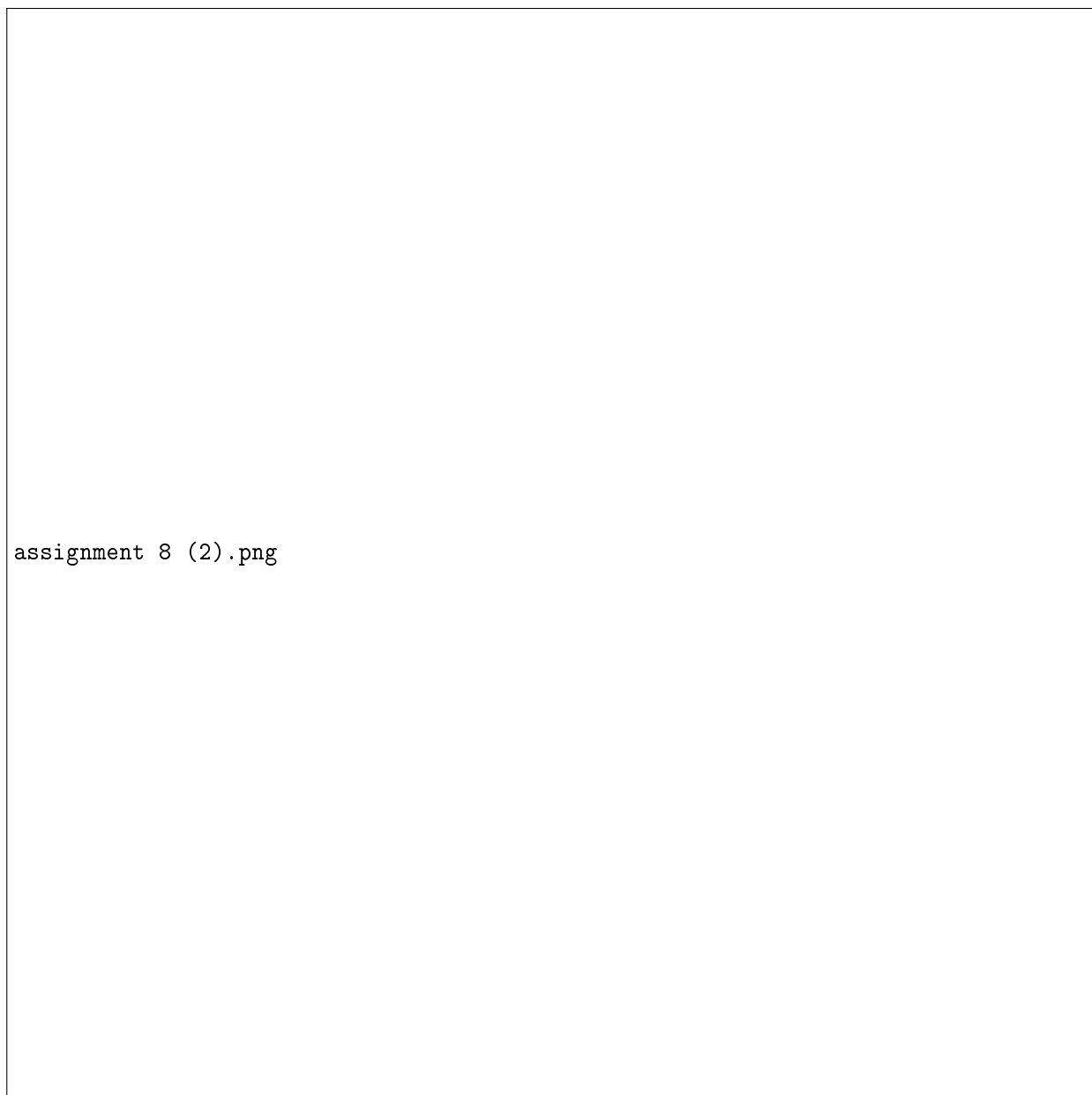


Figure 2: Filtering by target ip

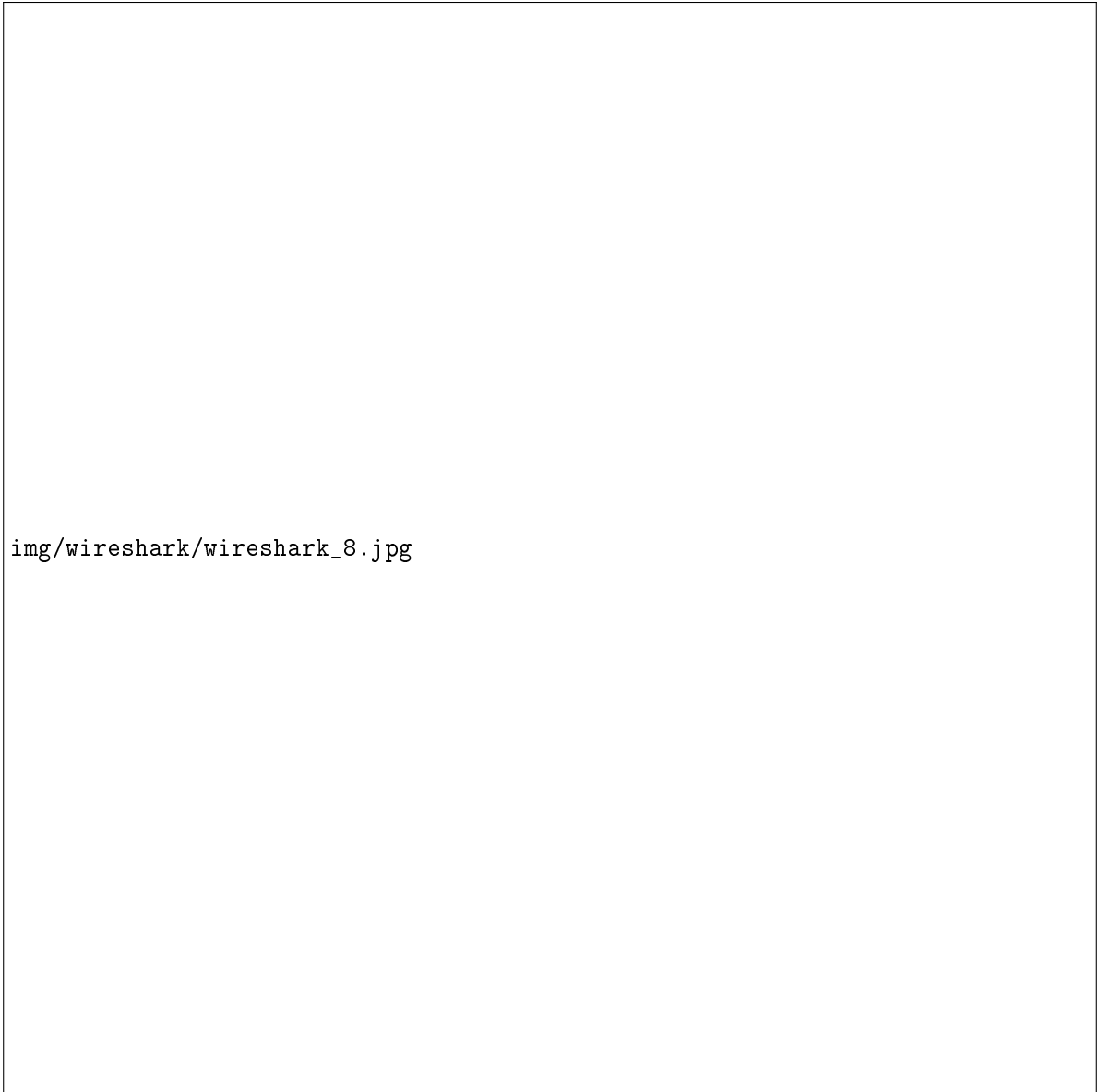
assignment 8 (3).png

Figure 3: Login Request from ip



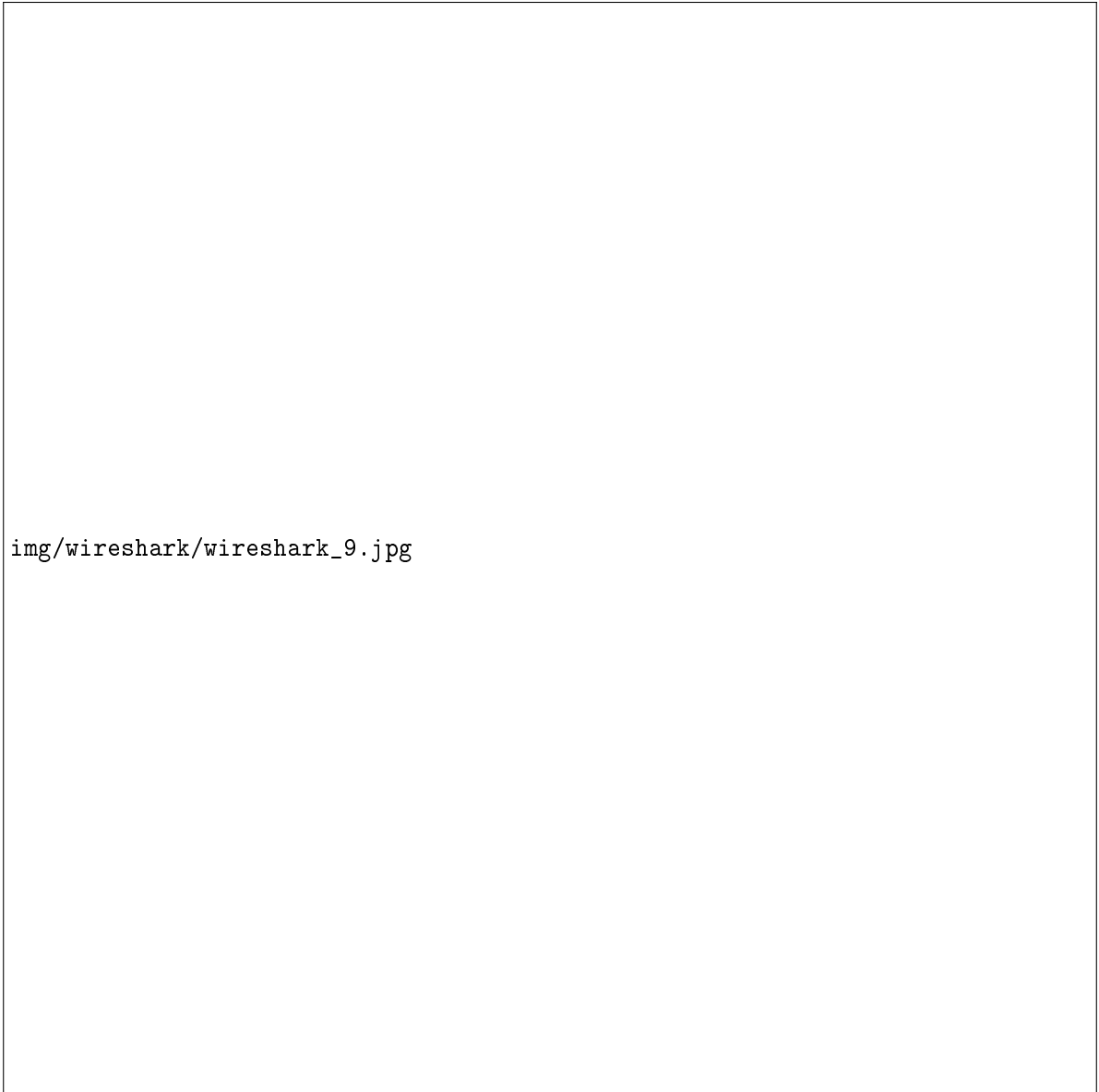
assignment 8 (2).png

Figure 4: Viewing Http packets to get username and password



img/wireshark/wireshark_8.jpg

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



img/wireshark/wireshark_9.jpg

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
```

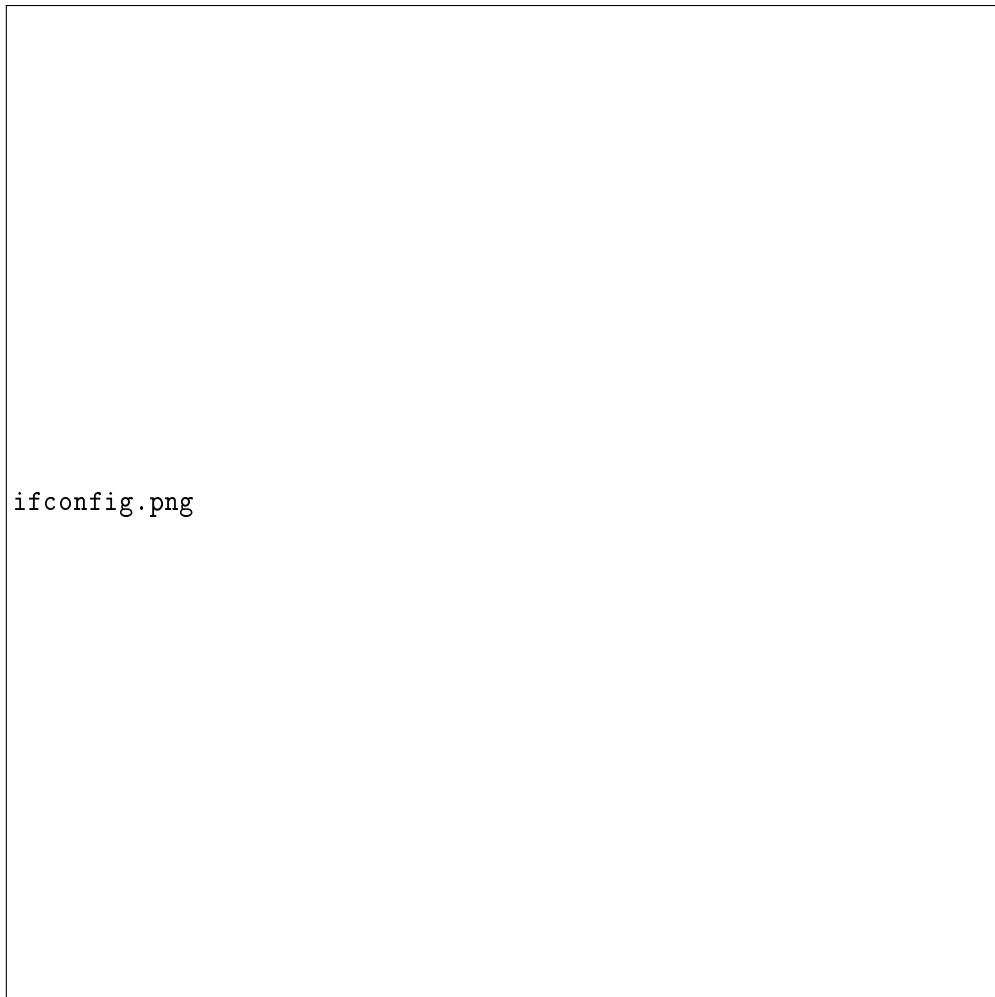
Command

```
$ifconfig
```

Purpose

To get the IP Address of the machine.

Output



ifconfig.png

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

Figure 8: Get IP Address

3.7 Scan any IP**3.7.1 Syntax**

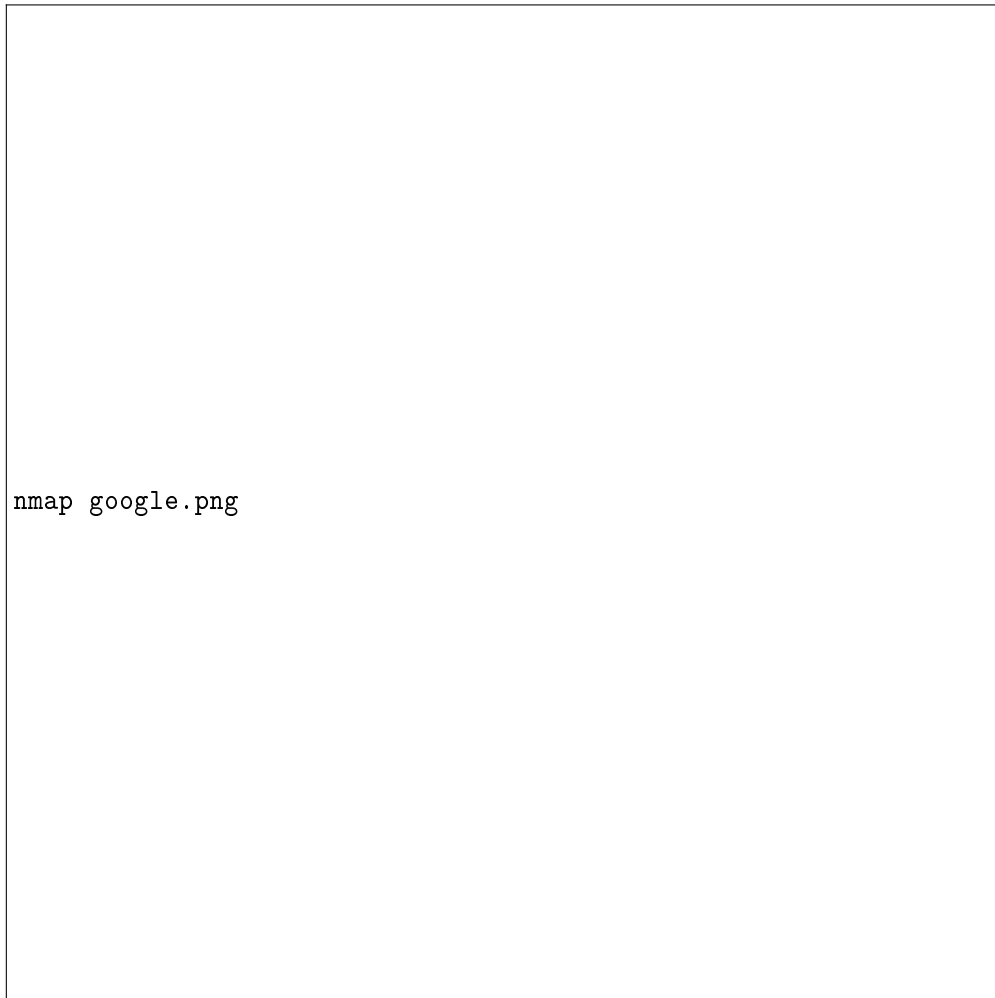
```
$ nmap <ip>
```

Command

```
$ nmap 192.168.1.38
```

Purpose

Scan a single ip

Output

nmap google.png

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



scan ip range 1.png

Figure 10: scan range of ips.

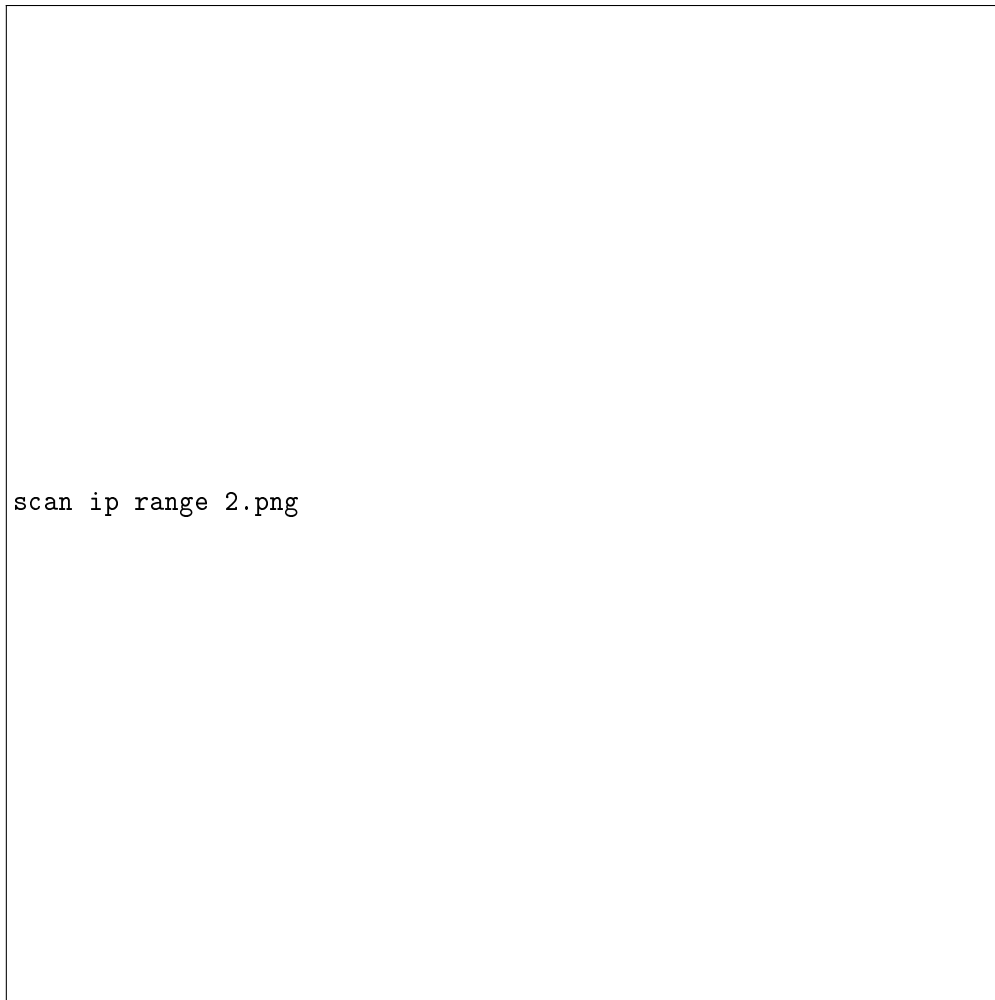


Figure 11: scan range of ips.

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

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

nmap range of ports.png

Figure 13: Scan a range of ports

3.11 Fragmented Scan**3.11.1 Syntax**

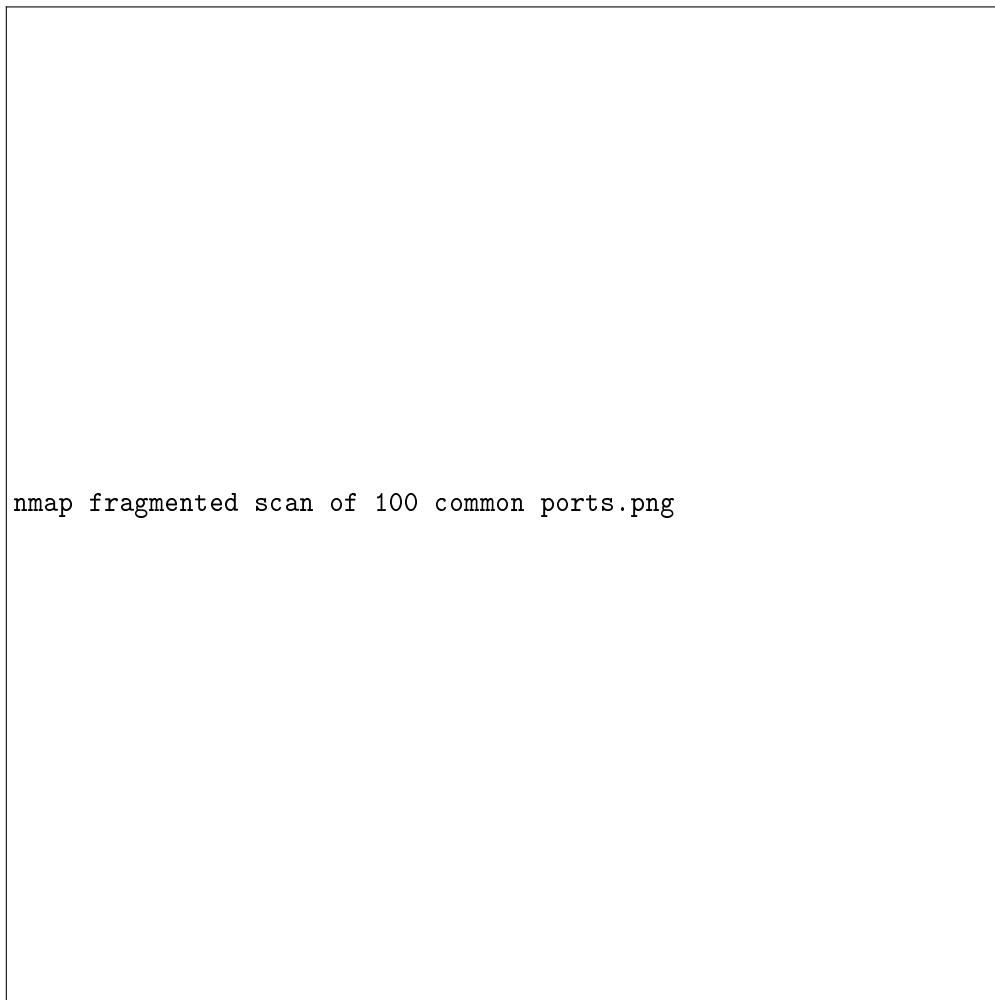
```
$ nmap -F <ip>
```

Command

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

Purpose

Fragmented Scan is used to evade firewalls.

Output

nmap fragmented scan of 100 common ports.png

Figure 14: Perform a fragmented scan.

3.12 TCP SYN Scan**3.12.1 Syntax**

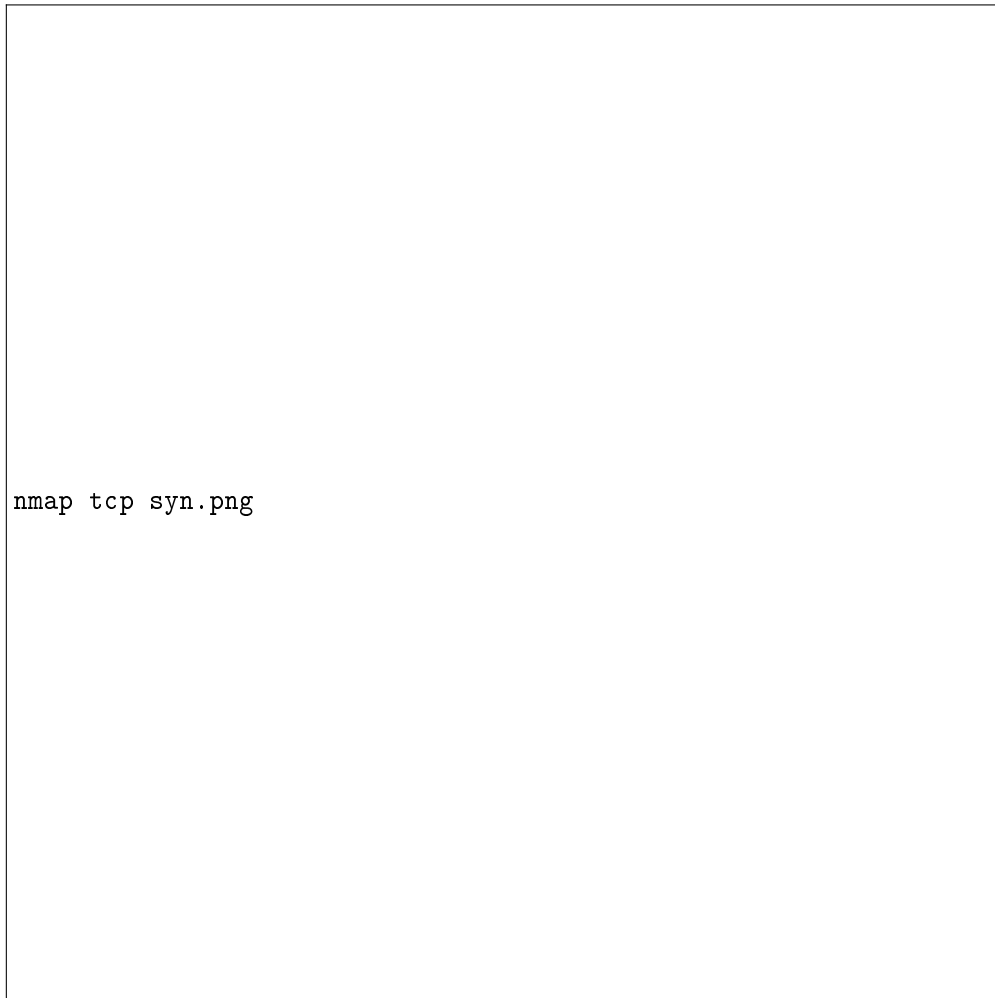
```
$ nmap -sS <ip>
```

Command

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

Purpose

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

Output

nmap tcp syn.png

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

3.13 OS Detection**3.13.1 Syntax**

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

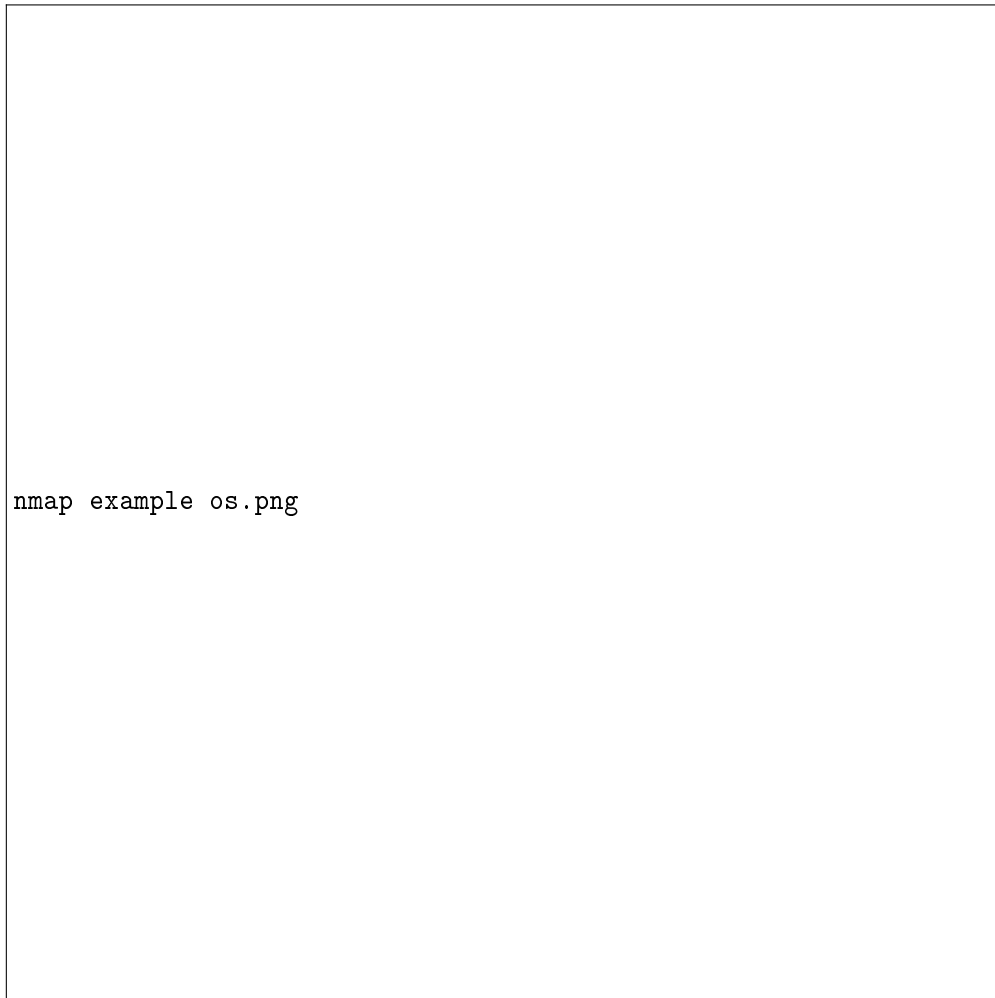
Command

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

Purpose


To scan operating system of a host.

Output



nmap example os.png

Figure 16: Scan Operating System of example.com



```
nmap os host.png
```

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

with ping scan.png

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

Figure 19: scan without ping

3.16 Nmap Timing Templates



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



t5.png

Figure 21: With T5



Figure 22: With T4



Figure 23: With T3

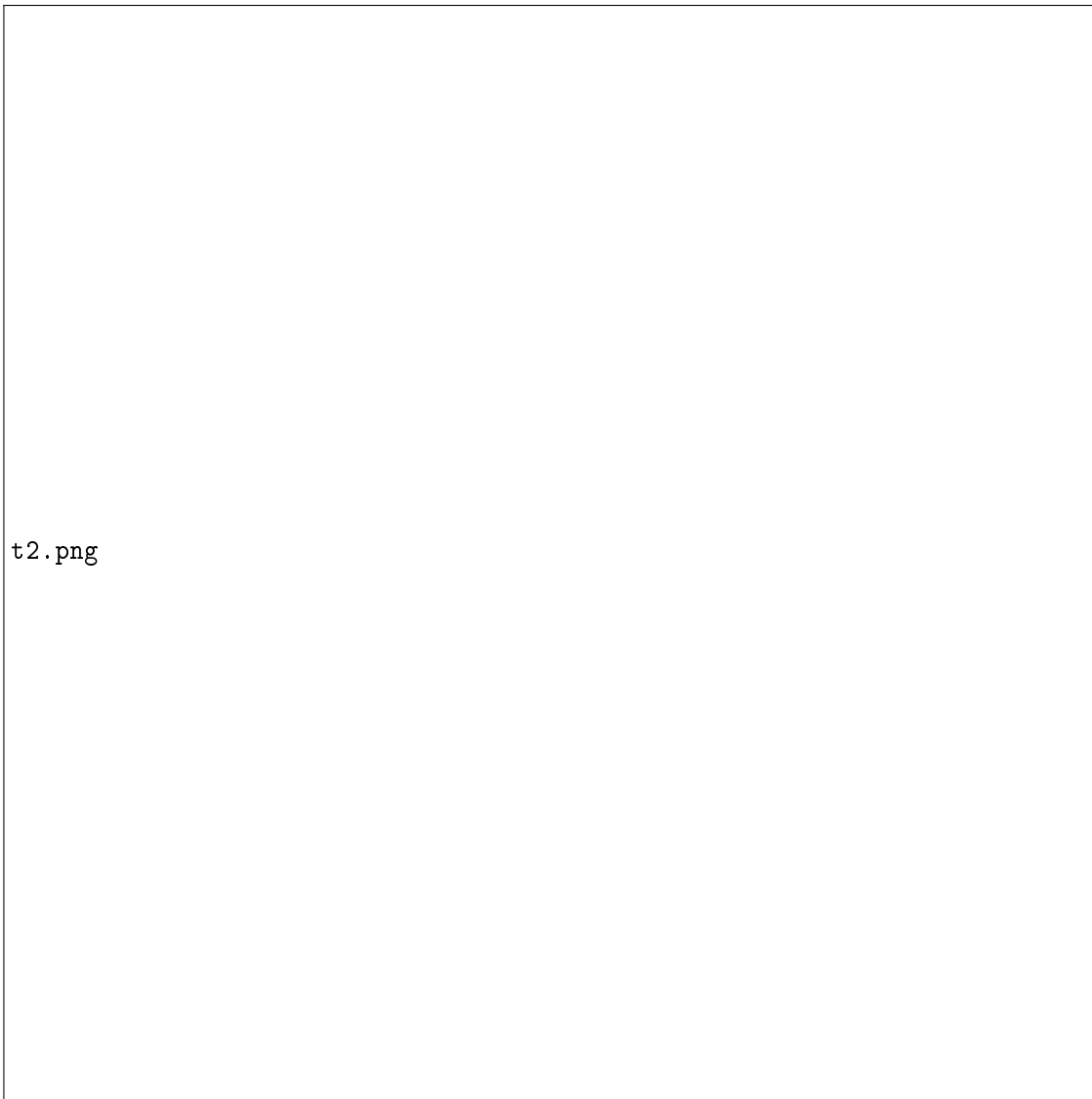


Figure 24: With T2

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



Figure 25: Scan for vulnerabilities

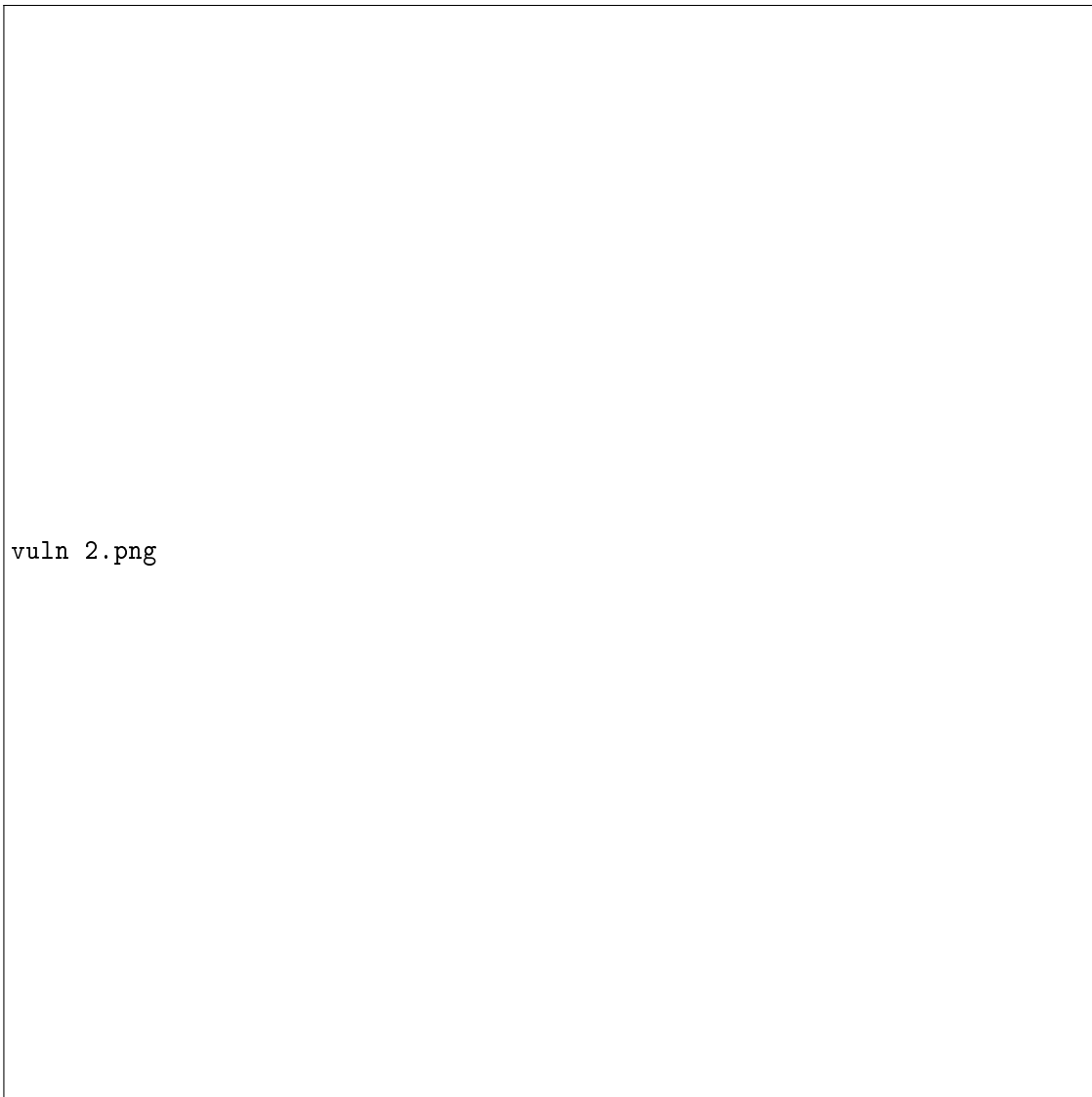


Figure 26: Scan for vulnerabilities

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.

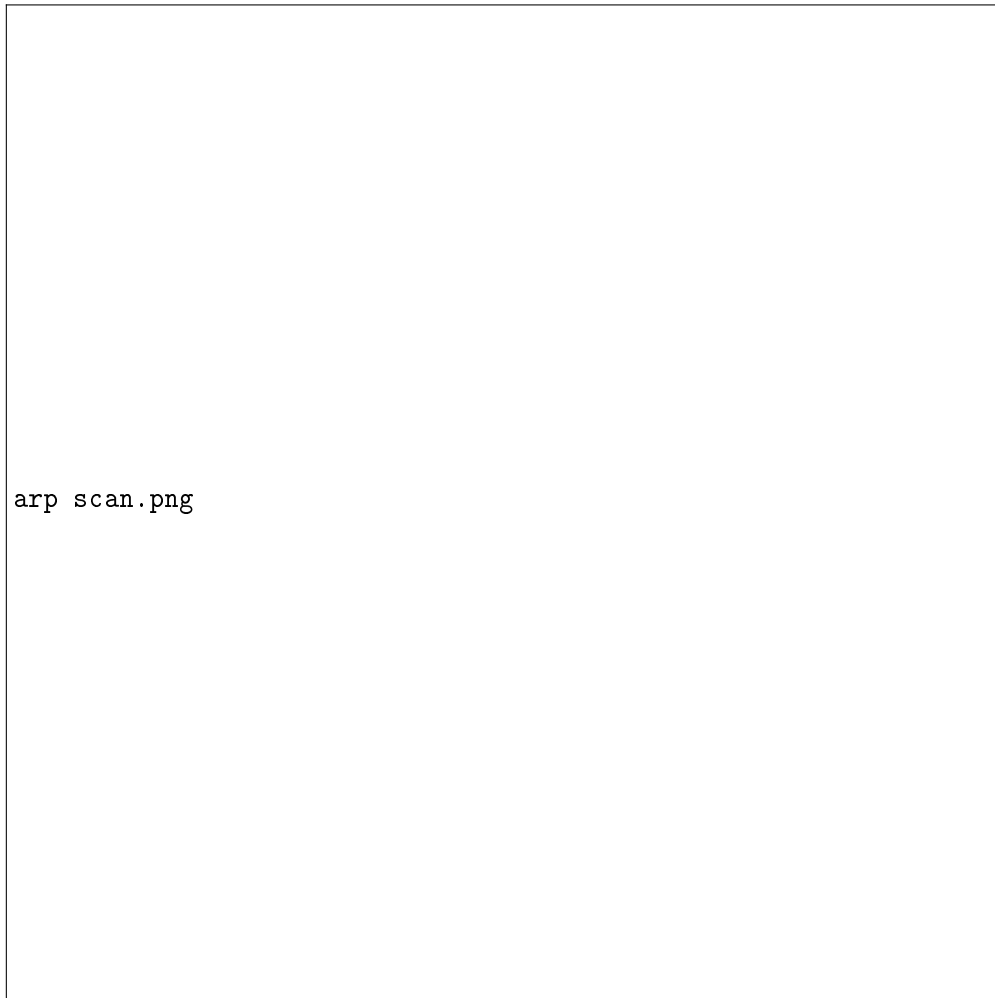
Output

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

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

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

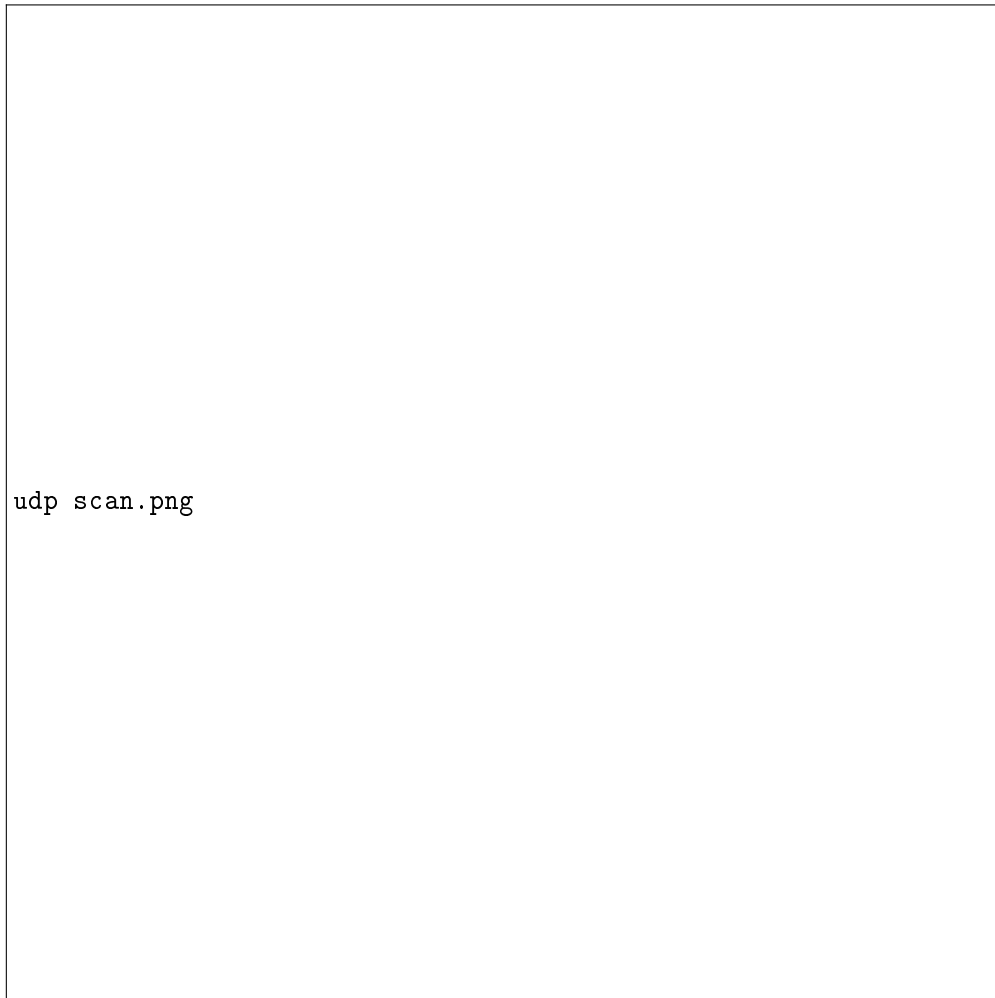


Figure 30: To scan live hosts using UDP scan.

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/>