Port Scanning, Packet Analysis, and Network Monitoring using Nmap, Wireshark, and Wazuh

In the field of cybersecurity, understanding how network reconnaissance and analysis tools operate is essential for both offensive and defensive security. Two of the most widely used tools in this domain are Nmap and Wireshark.

Port scanning is a technique used to discover open ports and services running on a target system. It helps security professionals identify potential vulnerabilities, as each open port can represent a possible entry point into the system. Nmap (Network Mapper) is a powerful and flexible command-line tool used for performing various types of port scans, service discovery, and network mapping. It provides detailed information about the status of ports (open, closed, filtered), the presence of firewalls, and the types of services running on a host.

On the other hand, Wireshark is a widely-used packet analyser that captures and displays the traffic passing through a network interface. It allows for in-depth inspection of network protocols and packet structures. By analysing packets during Nmap scans, one can observe how different scan types behave on the network, how targets respond, and how firewalls or intrusion detection systems react.

This project combines the use of Nmap for active scanning and Wireshark for passive traffic analysis. By conducting various types of Nmap scans against a target machine and capturing the network traffic in real-time, we gain a clearer understanding of the nature of each scan, how they differ in stealth and detection, and how they are represented at the packet level.

The goal of this exercise is to build practical skills in reconnaissance, scanning, and packet-level inspection — which are critical for roles in penetration testing, network defense, and digital forensics.

Tools used:

- VMware Workstation
- Kali Linux (Attacker Machine)
- Cyber Security Lab VM (Target Machine)
- Wazuh (For Monitoring)
- Wireshark (For Packet Analysis)
- Nmap (Network Mapper)

Lab Setup:

- Attacker IP (Kali Linux): 10.55.255.215
- Target IP (Cybersecurity Lab): 10.55.255.145
- Wazuh Agent Installed on Target Machine

Objective:

To perform various Nmap scans on the target machine, capture the traffic using Wireshark, and analyse the network behaviour of each scan type.

3-Way Handshaking process:

The 3-way handshake is the process used by TCP (Transmission Control Protocol) to establish a reliable connection between a client and a server before data is exchanged.

Step 1: SYN (Synchronize)

- The client wants to start communication.
- It sends a SYN packet to the server.
- This packet includes an initial sequence number (ISN).

Step 2: SYN-ACK (Synchronize-Acknowledge)

- The server receives the SYN.
- It responds with a SYN-ACK packet.
- This means:
    - SYN to initiate a connection back to the client.
    - ACK to acknowledge the client’s SYN.
- It includes its own ISN and acknowledges the client's sequence number.

Step 3: ACK (Acknowledge)

- The client sends an ACK back to the server.
- This packet acknowledges the server’s SYN.
- After this, the connection is established.

Nmap Command and Wireshark Packet Behaviour:

- **Command: nmap -sS 10.55.255.145**

Scan Type: TCP SYN Scan

Definition: Stealth scan. Sends SYN packets to check for open ports without completing handshake. Stealthy and fast.

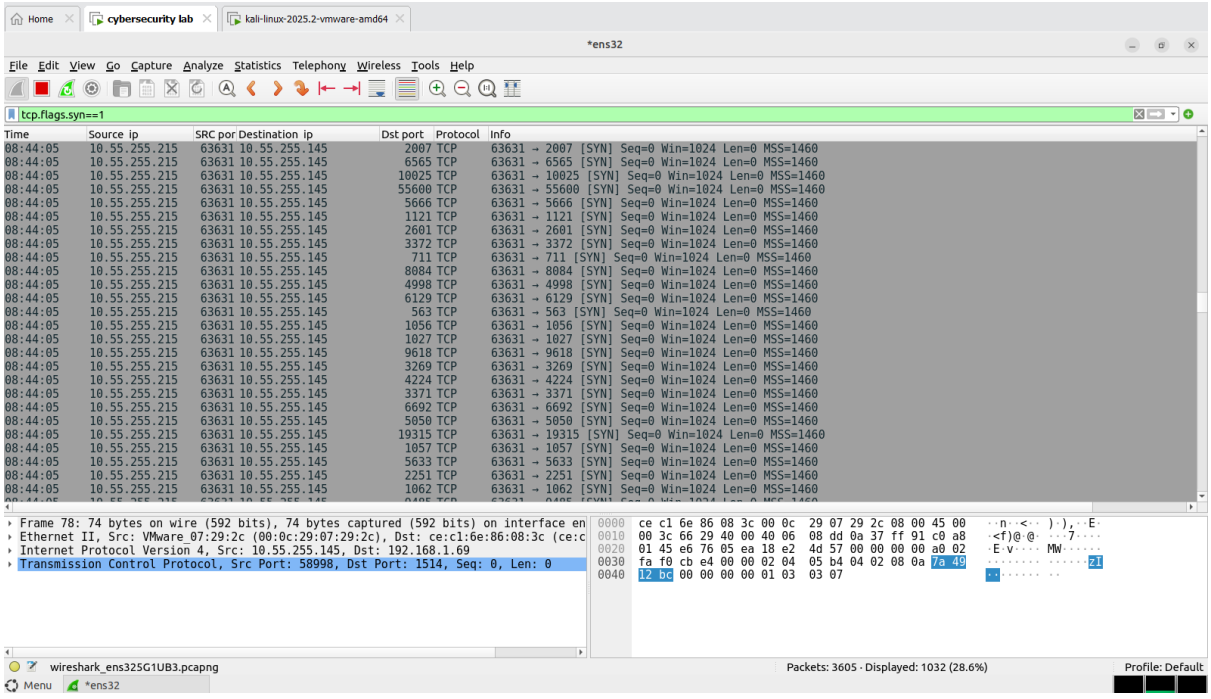
```
┌──(root㉿kali)-[~]
└─# nmap -sS 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 09:30 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00024s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

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

What Happens in Wireshark:

When you run this command, Nmap sends TCP SYN packets to multiple ports on the target IP. In Wireshark, you will see SYN packets going out from the attacker to the target.

- If the port is open, the target replies with SYN-ACK.
- If the port is closed, the target replies with RST.
  Nmap never completes the handshake—it sends no ACK—so this scan appears as a partial connection. This makes it stealthy and harder to detect.

Wireshark command: tcp.flags.syn ==1



❖ Command: nmap -sT 10.55.255.145

Scan Type: TCP Connect Scan

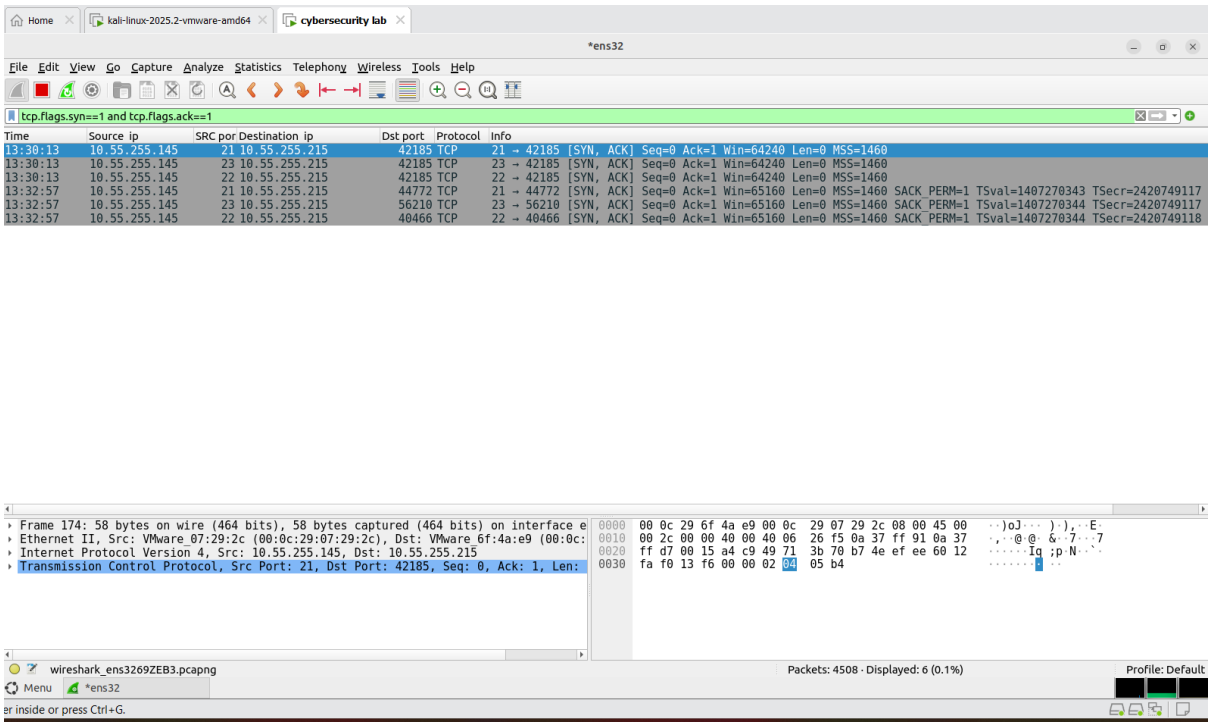
Definition: Completes full TCP handshake. More detectable. Used when SYN scan is not allowed.

```
┌──(root㉿kali)-[~]
└─# nmap -sT 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 04:47 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00024s latency).
Not shown: 997 closed tcp ports (conn-refused)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

What Happens in Wireshark:
This time, Nmap performs a full TCP handshake. In Wireshark, you will observe:

- SYN → SYN-ACK → ACK
  For every open port, this full sequence will appear. After the handshake, Nmap closes the connection.
  This scan is noisy and detectable, as it completes legitimate connections.

Wireshark commands: tcp.flags.syn==1 and tcp.flags.ack==1



❖ **Command: nmap -sA 10.55.255.145**

Scan Type: ACK Scan

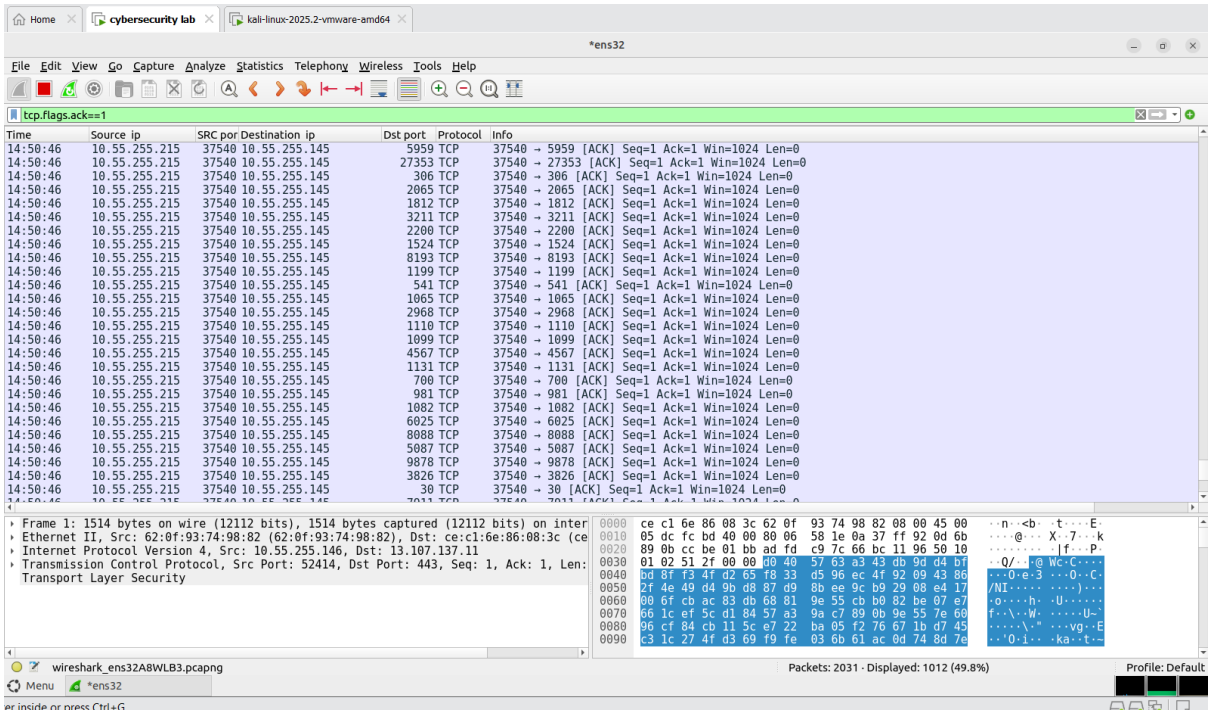
Definition: Checks firewall rules, not port status. If RST received → unfiltered.

```
┌──(root㉿kali)-[~]
└─# nmap -sA 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:50 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00014s latency).
All 1000 scanned ports on 10.55.255.145 are in ignored states.
Not shown: 1000 unfiltered tcp ports (reset)
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.23 seconds
```

What Happens in Wireshark:
Nmap sends TCP packets with just the ACK flag set. These aren't used to identify open ports but rather to check if a firewall is filtering traffic.
In Wireshark:

- You'll see ACK packets sent to the target.
- If the port is unfiltered, the target replies with RST.
- If filtered, there may be no response at all.
- This scan helps map firewall rules.

Wireshark commands: tcp.flags.ack==1



❖ **Command: nmap 10.153.68.145 -sU**

Scan Type: UDP Scan

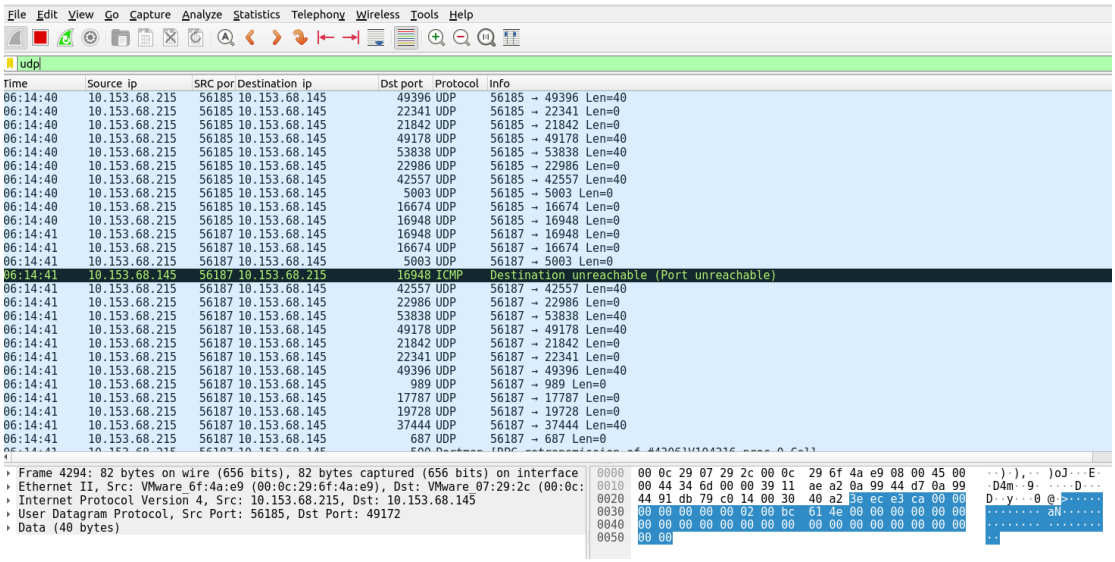
Definition: Scans UDP ports. Slower, used for DNS, SNMP,DHCP etc.

```
┌──(root㉿kali)-[~]
└─# nmap -sU -T5 10.153.68.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-24 02:14 EDT
Warning: 10.153.68.145 giving up on port because retransmission cap hit (2).
Nmap scan report for 10.153.68.145
Host is up (0.0012s latency).
Not shown: 983 open|filtered udp ports (no-response)
PORT      STATE  SERVICE
111/udp   closed rpcbind
123/udp   open   ntp
500/udp   closed isakmp
1993/udp  closed snmp-tcp-port
9000/udp  closed cslistener
16838/udp closed unknown
16948/udp closed unknown
17331/udp closed unknown
19956/udp closed unknown
21556/udp closed unknown
31195/udp closed unknown
40711/udp closed unknown
44253/udp closed unknown
49172/udp closed unknown
49208/udp closed unknown
49226/udp closed unknown
64727/udp closed unknown
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 23.94 seconds
```

What Happens in Wireshark:
UDP is connectionless, so things look different in Wireshark.

- You will see UDP packets sent from the attacker to different ports.
- If a port is closed, the target responds with ICMP Port Unreachable.
- If the port is open, there is often no reply.
  This makes interpreting UDP scans trickier, but Wireshark helps catch the ICMP errors.

Wireshark commands: udp



❖ **Command: nmap -p 23 10.55.255.145**

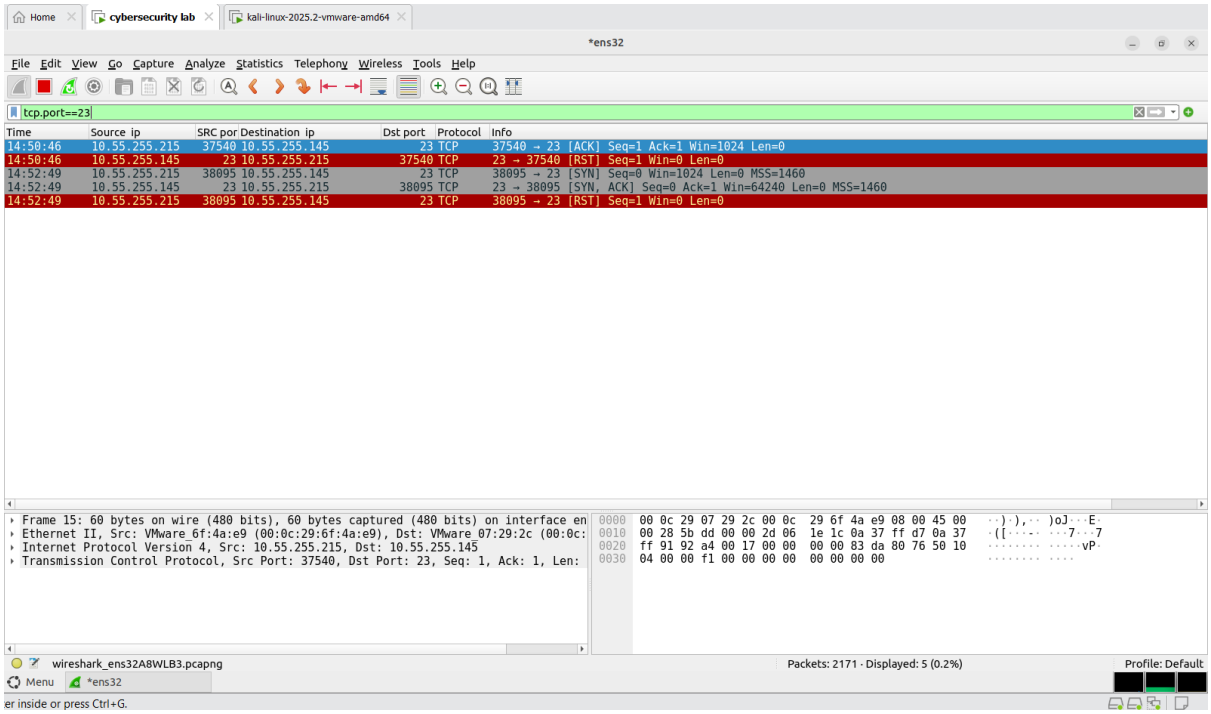
Scan Type: Specific Port Scan
Definition: Scans only port 23 (commonly Telnet) on the target system.

```
┌──(root㉿kali)-[~]
└─# nmap -p 23 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:52 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00062s latency).

PORT   STATE SERVICE
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.21 seconds
```

What Happens in Wireshark?

Shows TCP packets sent to port 23 with SYN flag. You may see SYN-ACK or RST responses depending on port status.

Wireshark commands: tcp.port==23



❖ Command: nmap -p 23-100 10.55.255.145

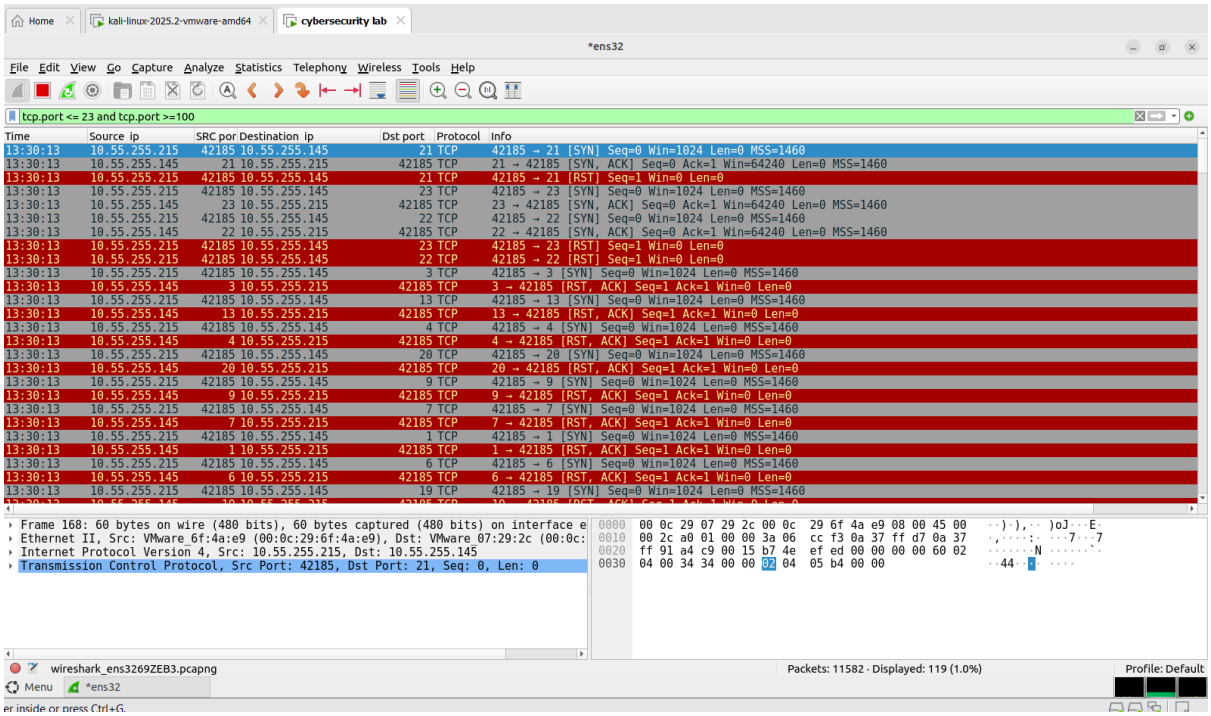
Scan Type: Port Range Scan
Definition: Scans all ports between 23 to 100.

```
┌──(root㉿kali)-[~]
└─# nmap -p 23-100 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 09:49 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00024s latency).
Not shown: 77 closed tcp ports (reset)
PORT   STATE SERVICE
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.60 seconds
```

What Happens in Wireshark?

You'll see multiple SYN packets for each port in the range. Responses vary per port (SYN-ACK for open, RST for closed).

Wireshark command: tcp.port<=23 and tcp.port>=100



❖ **Command: nmap -pU:110,T:23-25,443 10.55.255.145**

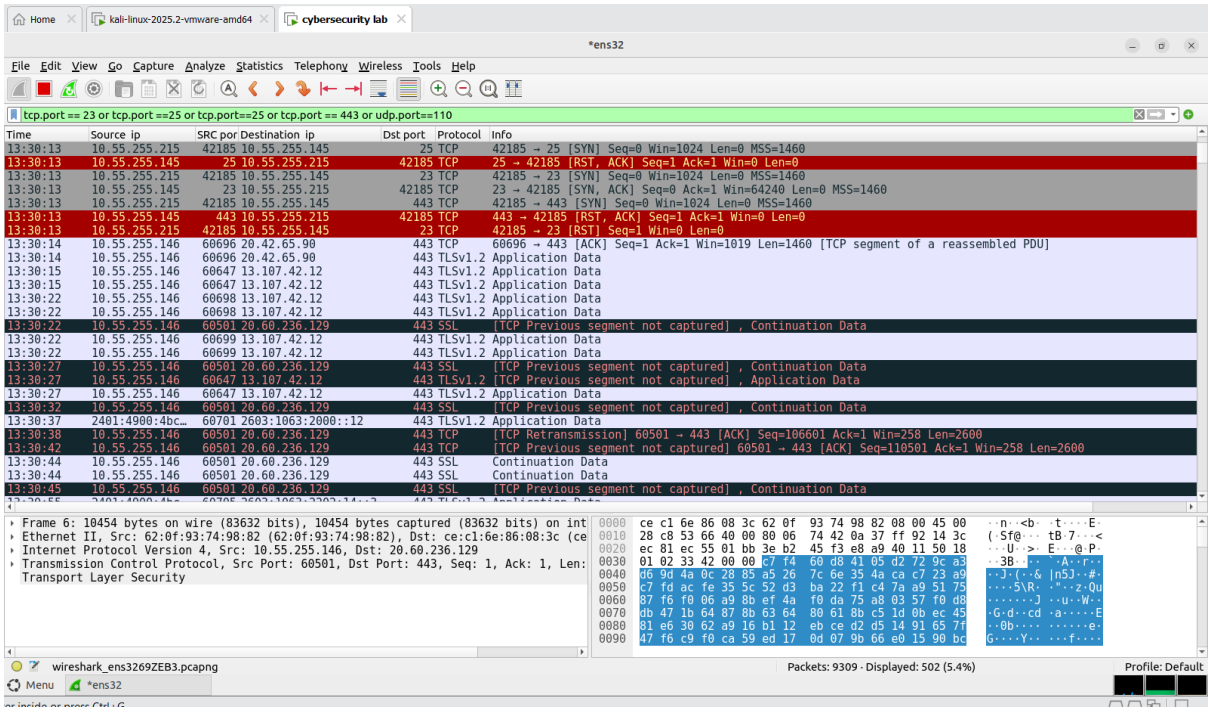
Scan Type: Mixed UDP and TCP Scan
Definition: Scans UDP port 110 and TCP port 23, 443

```
┌──(root㉿kali)-[~]
└─# nmap -pU:110,T:23-25,443 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 09:55 EDT
WARNING: Your ports include "U:" but you haven't specified UDP scan with -sU.
Nmap scan report for 10.55.255.145
Host is up (0.00052s latency).

PORT    STATE  SERVICE
23/tcp  open   telnet
24/tcp  closed priv-mail
25/tcp  closed smtp
443/tcp closed https
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

What Happens in Wireshark?

You'll see both UDP and TCP traffic. UDP requests and possible ICMP Port Unreachable responses for closed UDP ports. SYN packets for TCP ports.

Wireshark command: tcp.port==23 or tcp.port==24 or tcp.port==25 or udp.port==110s



❖ **Command: nmap -sF 10.55.255.145**

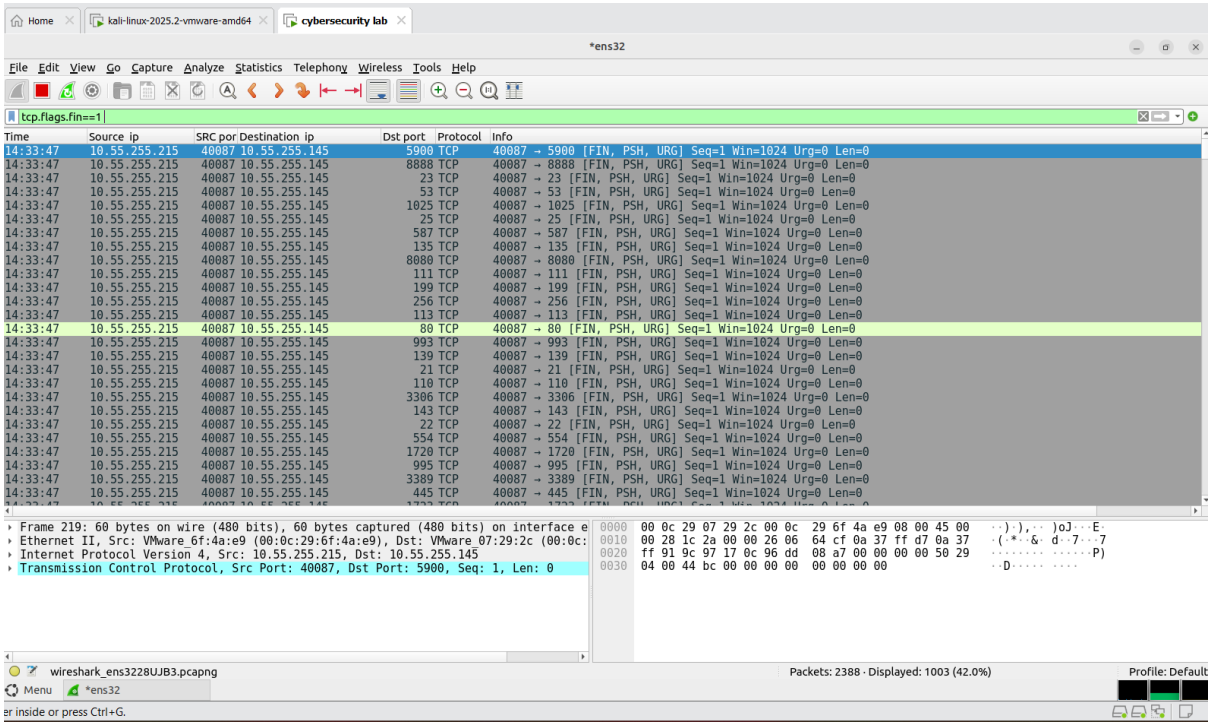
Scan Type: Fast Scan
Definition: Quickly scans only the most common 100 ports.

```
┌──(root㉿kali)-[~]
└─# nmap -sF 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:36 EDT
setup_target: failed to determine route to 10.55.255.145
WARNING: No targets were specified, so 0 hosts scanned.
Nmap done: 0 IP addresses (0 hosts up) scanned in 0.87 seconds
```

What Happens in Wireshark?

You'll see SYN packets to ports like 22, 80, 443, etc. A lighter scan—less noise compared to full scan.

Wireshark command: tcp.flags.fin==1



❖ **Command: nmap -r 10.55.255.145**

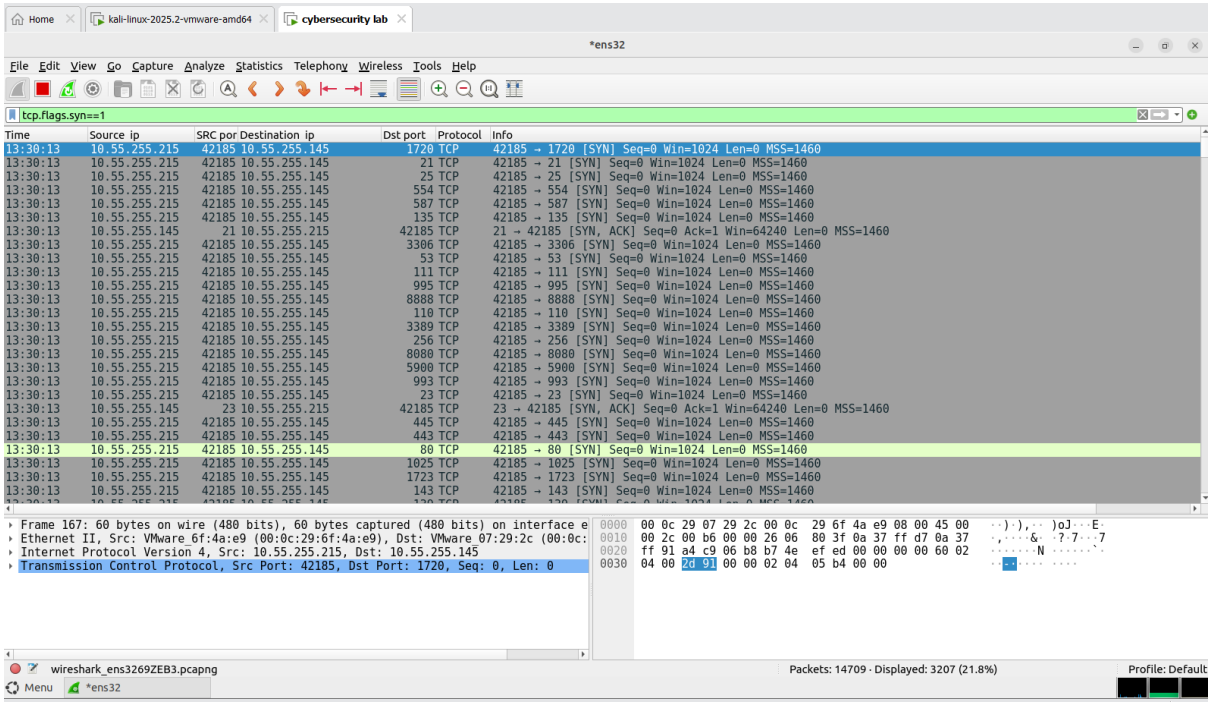
Scan Type: Sequential Scan
Definition: Scans ports in numerical order instead of randomizing.

```
┌──(root㉿kali)-[~]
└─# nmap -r 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:11 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00033s latency).
Not shown: 997 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 1.26 seconds
```

What Happens in Wireshark?

Same packet types (SYN) as usual, but you'll notice the port numbers increase sequentially in the capture.

Wireshark command: Tcp.flags.syn==1



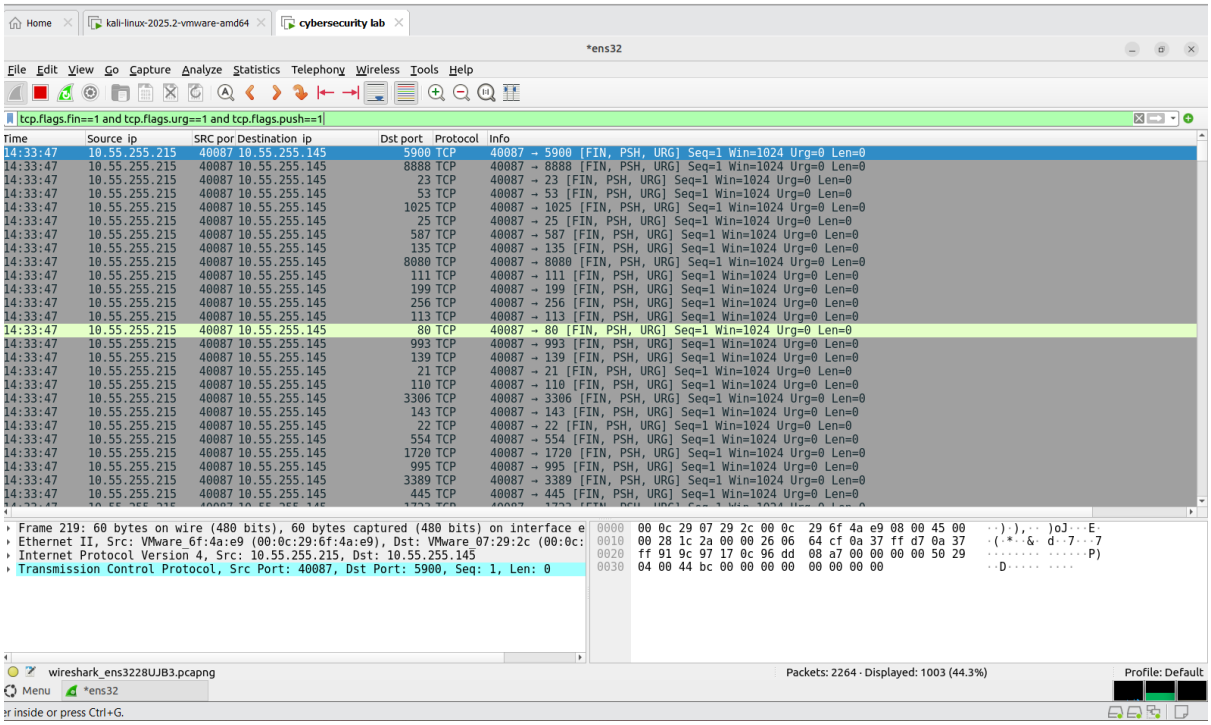
❖ Command: nmap -sX 10.55.255.145

Scan Type: Xmas Scan
Definition: Sends TCP packets with FIN, URG, and PSH flags set (like a "Christmas tree" of flags).

```
┌──(root㉿kali)-[~]
└─# nmap -sX 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:33 EDT
Nmap scan report for 10.55.255.145
Host is up (0.0050s latency).
Not shown: 997 closed tcp ports (reset)
PORT    STATE         SERVICE
21/tcp open|filtered ftp
22/tcp open|filtered ssh
23/tcp open|filtered telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.06 seconds
```

What Happens in Wireshark?
You'll see packets with unusual TCP flag combinations (FIN, URG, PSH) sent to various ports. If the port is closed, you get a RST response. If the port is open or filtered, there is no response.

Wireshark command: tcp.flags.fin==1 and tcp.flags.urg==1 and tcp.flags.push==1



❖ Command: nmap -sL 10.55.255.145

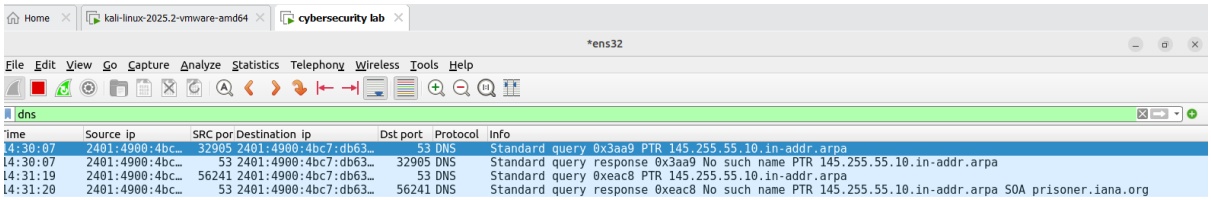
Type: List Scan
Definition: Lists all the IPs in the target range without sending any packets to the targets.

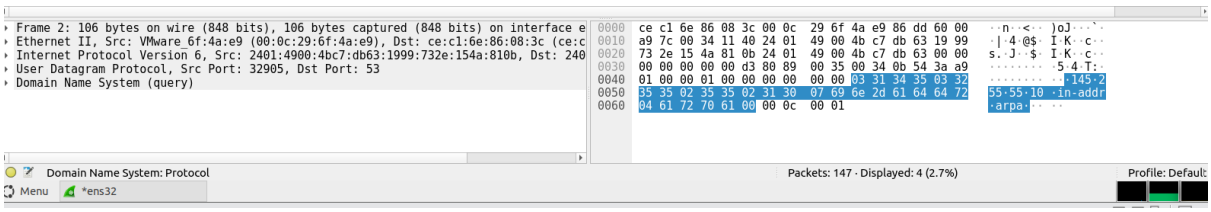
```
┌──(root㉿kali)-[~]
└─# nmap -sL 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:31 EDT
Nmap scan report for 10.55.255.145
Nmap done: 1 IP address (0 hosts up) scanned in 0.14 seconds
```

What Happens in Wireshark?

Almost no traffic to the target is seen in Wireshark. You may see DNS queries if hostnames are being resolved.

Wireshark command: dns





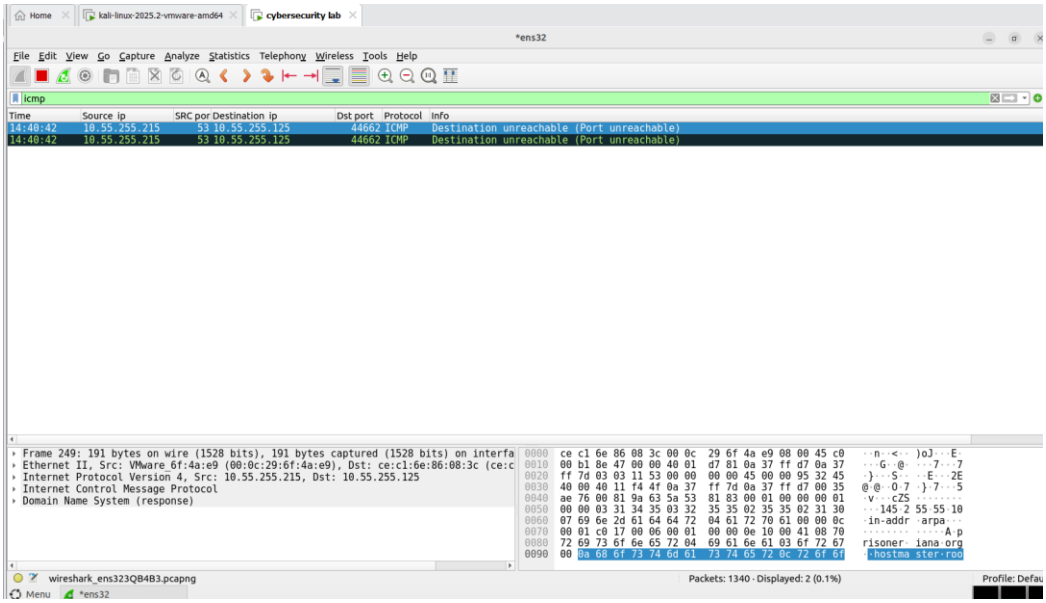
❖ **Command: nmap -sP 10.55.255.145**

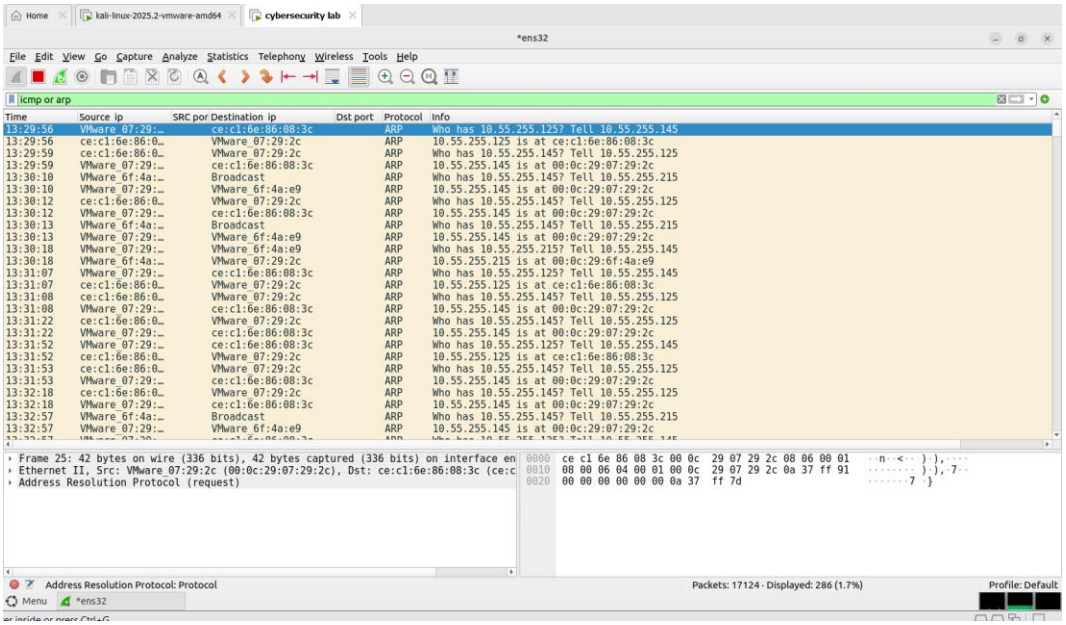
Scan Type: Ping Scan
Definition: Sends ICMP Echo Requests or ARP Requests to check if the host is up.

```
┌──(root㉿kali)-[~]
└─# nmap -sP 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:15 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00047s latency).
MAC Address: 00:0C:29:07:29:2C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.36 seconds
```

What Happens in Wireshark?
You'll see ICMP Echo Request (type 8) packets sent to the target, and possibly ARP requests if on the same subnet. If the host is up, it replies with ICMP Echo Reply (type 0).

Wireshark command: icmp or arp





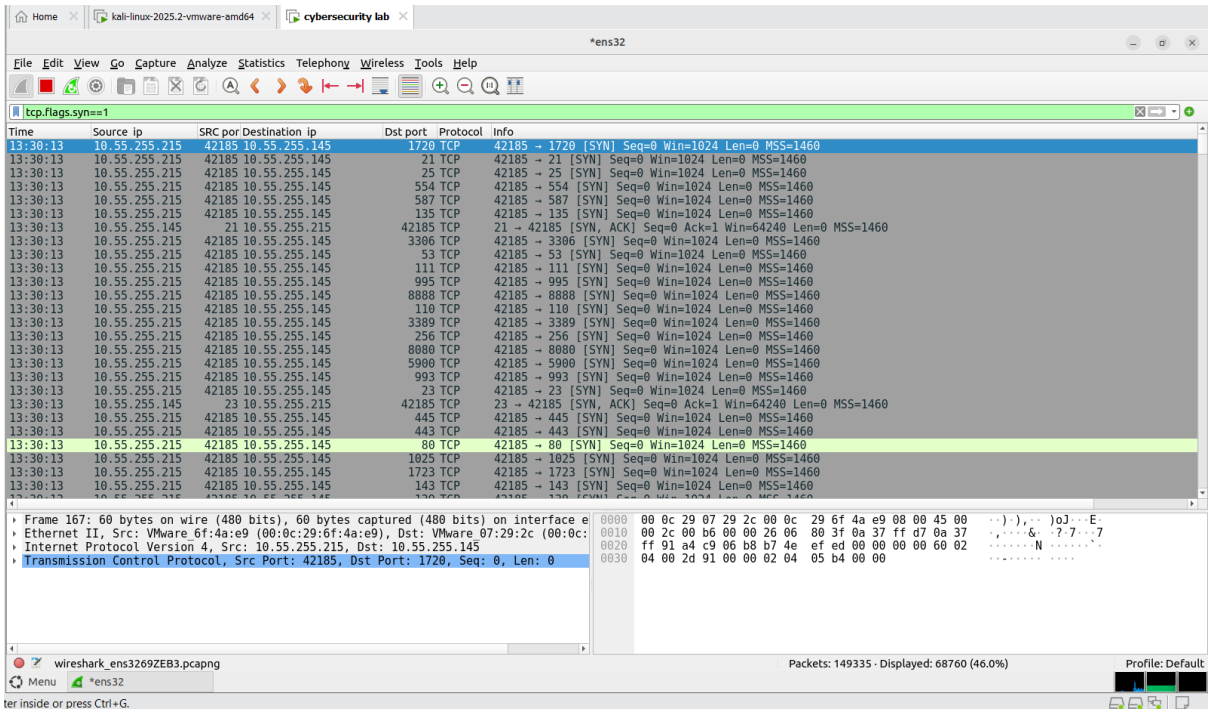
❖ Command: nmap -p- 10.55.255.145

Scan Type: Full Port Range Scan
Definition: Scans all 65,535 TCP ports (instead of the default 1–1000).

```
┌──(root㉿kali)-[~]
└─# nmap -p- 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:17 EDT
Nmap scan report for 10.55.255.145
Host is up (0.0040s latency).
Not shown: 65532 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 2.40 seconds
```

What Happens in Wireshark?
You'll observe SYN packets sent to every port, from 1 to 65535. The target will respond with either SYN-ACK (open) or RST (closed). High volume of traffic.

Wireshark command: tcp.flags.syn==1



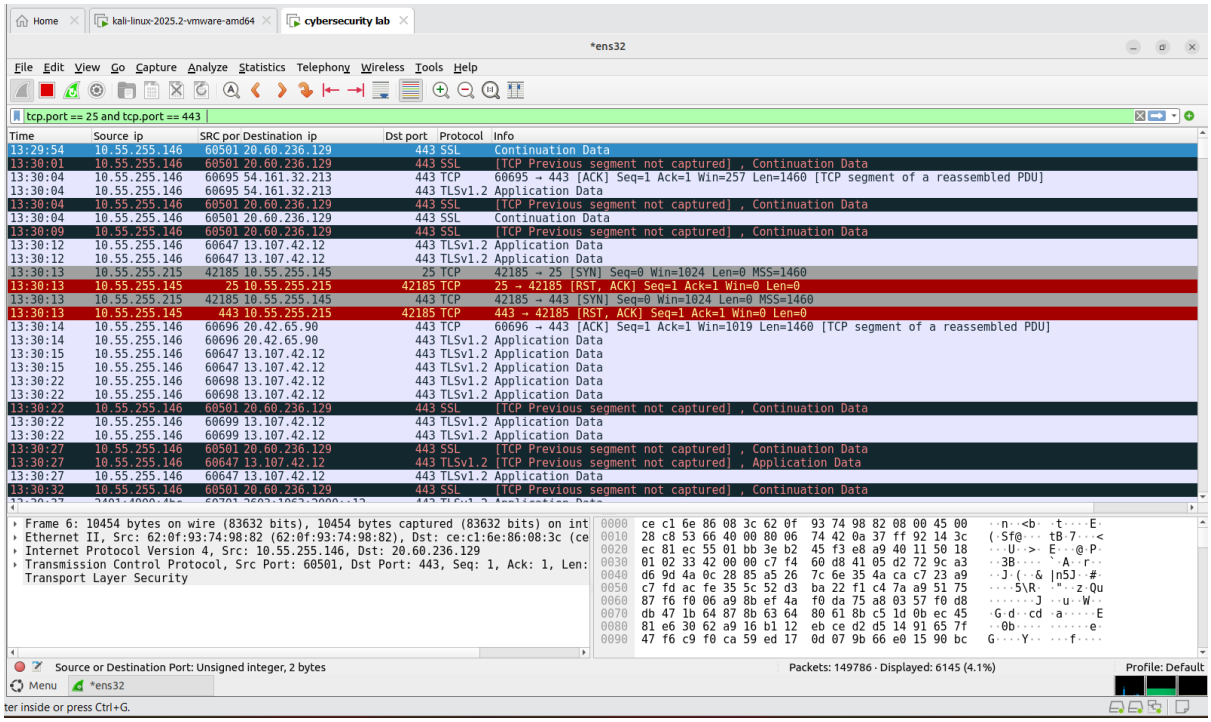
❖ **Command: nmap -p smtp,https 10.55.255.145**

Scan Type: Targeted Port Scan
Definition: Scans only specified ports – SMTP (25) and HTTPS (443).

```
┌──(root㉿kali)-[~]
└─# nmap -p smtp,https 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:19 EDT
Stats: 0:00:00 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 100.00% done; ETC: 10:19 (0:00:00 remaining)
Nmap scan report for 10.55.255.145
Host is up (0.00044s latency).

PORT    STATE  SERVICE
25/tcp  closed smtp
443/tcp closed https
MAC Address: 00:0C:29:07:29:2C (VMware)
```

What Happens in Wireshark?
You'll see SYN packets sent only to ports 25 and 443. The responses will be SYN-ACK (open) or RST (closed).

Wireshark command: Tcp.port == 25 and tcp.port == 443



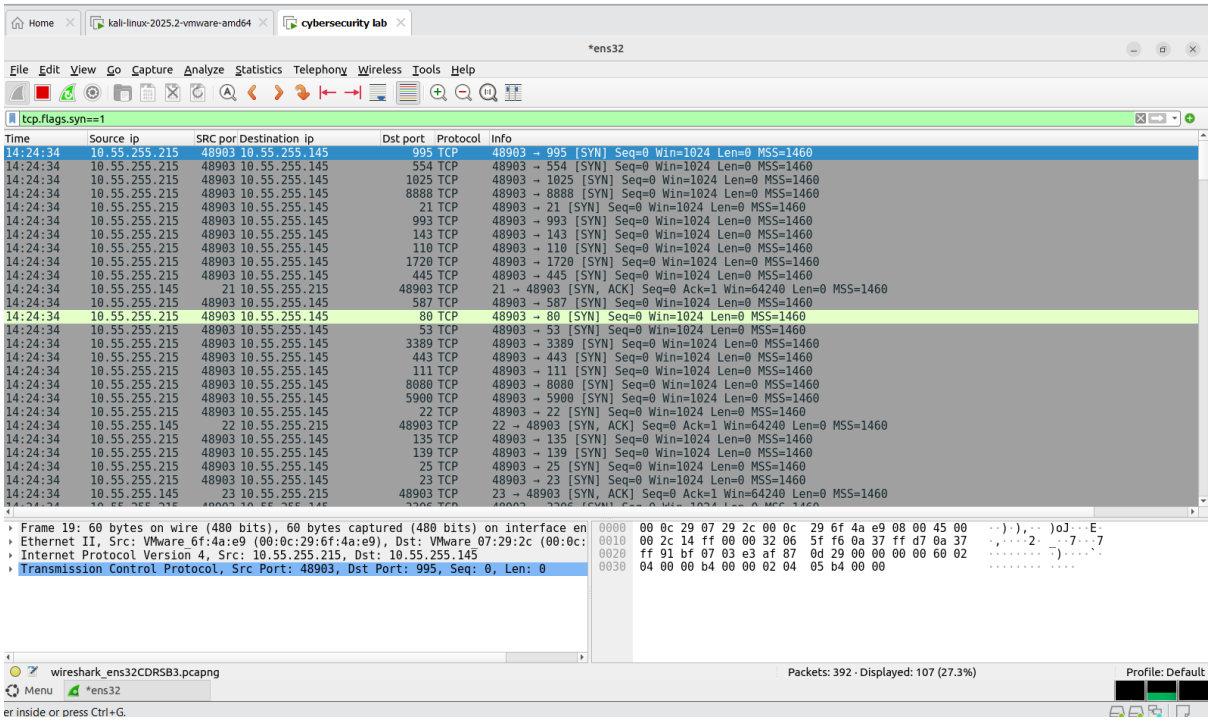
❖ **Command: nmap -F 10.55.255.145**

Scan Type: Fast Scan
Definition: Scans fewer ports (top 100 most common) instead of the default 1000.

```
┌──(root㉿kali)-[~]
└─# nmap -F 10.55.255.145
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:24 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00073s latency).
Not shown: 97 closed tcp ports (reset)
PORT   STATE SERVICE
21/tcp open  ftp
22/tcp open  ssh
23/tcp open  telnet
MAC Address: 00:0C:29:07:29:2C (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.55 seconds
```

What Happens in Wireshark?
You'll see SYN packets only to top 100 common ports (like 80, 443, 22, etc.). Less traffic compared to a full scan. Faster.

Wireshark command: tcp.flags.syn==1



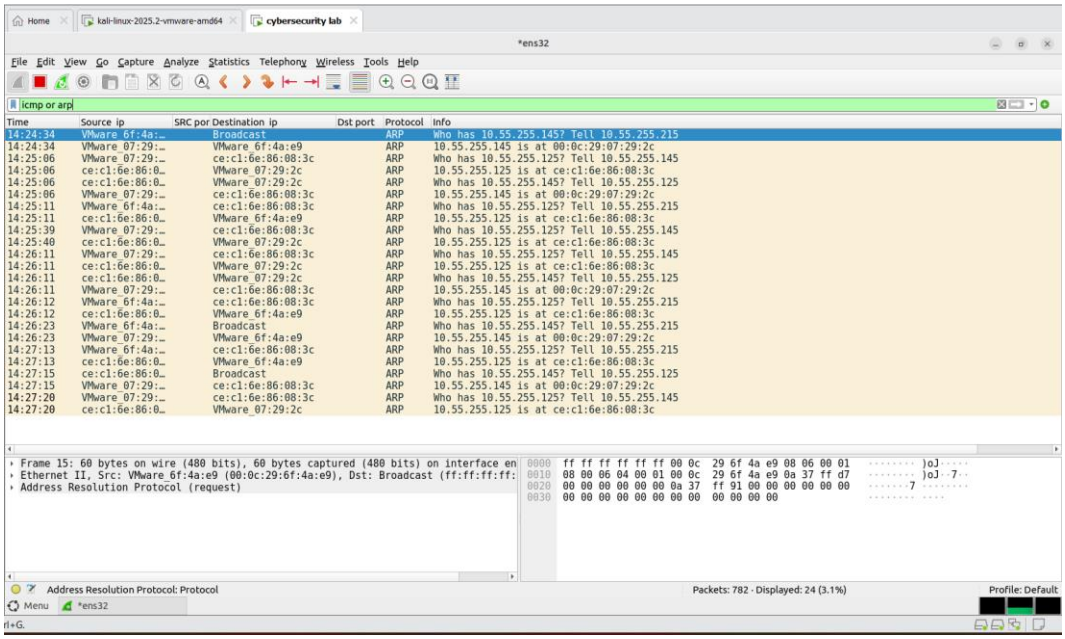
Command: nmap 10.55.255.145 -sn

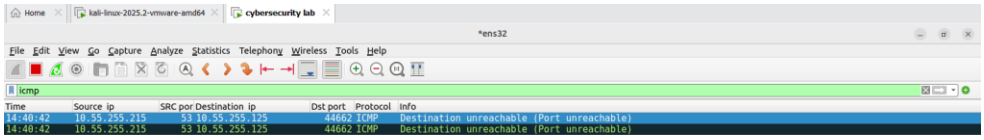
Scan Type: No Port Scan (Ping Only)
Definition: Disables port scanning; only checks if the host is online using ICMP or ARP.

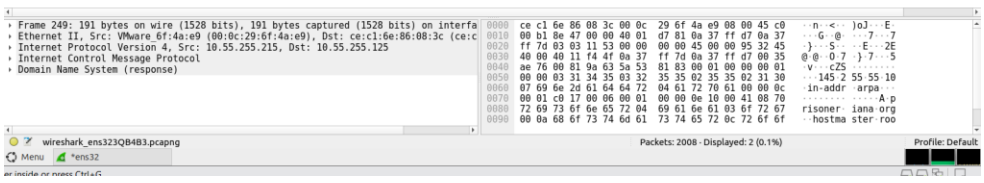
```
┌──(root㉿kali)-[~]
└─# nmap 10.55.255.145 -sn
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-27 10:26 EDT
Nmap scan report for 10.55.255.145
Host is up (0.00063s latency).
MAC Address: 00:0C:29:07:29:2C (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.20 seconds
```

What Happens in Wireshark?
Similar to -sP, you'll see ICMP Echo Requests, and potentially ARP requests. No SYN packets or port probing.

Wireshark command: icmp or arp

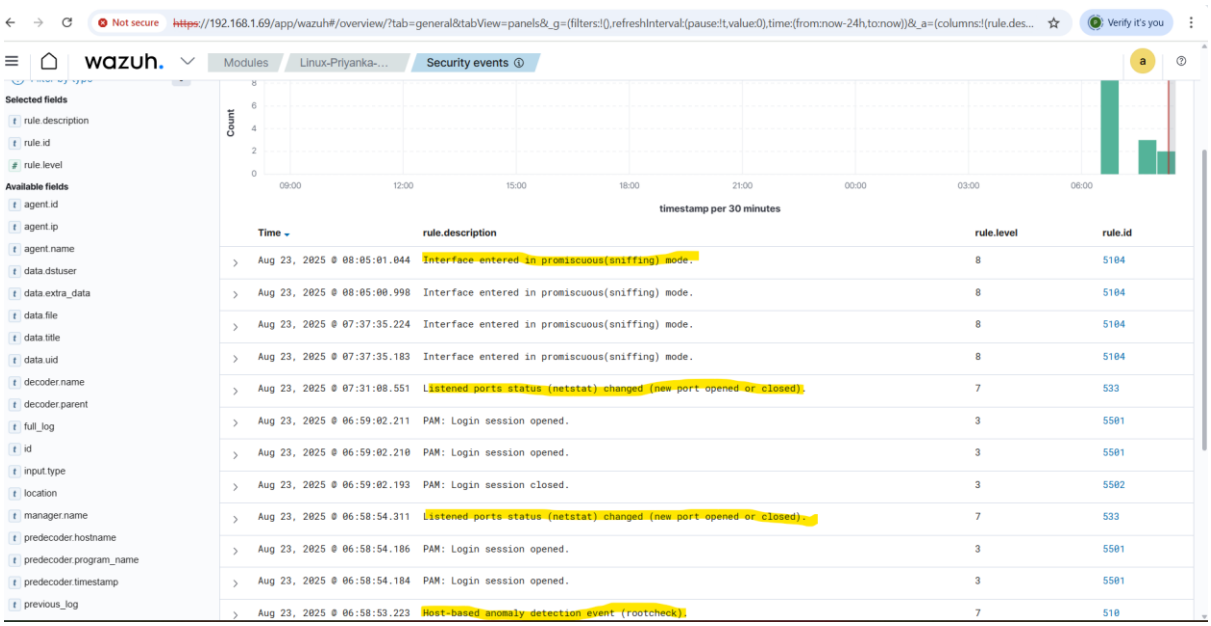






Wazuh alerts observed in the screenshot after performing a port scan from Kali Linux

sssssssss



The highlighted entries in your Wazuh Security Events screenshot represent alerts triggered by suspicious or abnormal activities — likely due to the port scanning performed from Kali Linux. Below is a detailed explanation of each marked (highlighted) log entry:

1. Interface entered in promiscuous (sniffing) mode.

- Rule ID: 5104
- Rule Level: 8

What It Means:
This alert indicates that a network interface on the monitored system was switched to promiscuous mode. In this mode, the system can capture all traffic on the network segment, not just traffic destined for it.

Why It Matters:
This behavior is often associated with network sniffing or surveillance tools like Wireshark. It could be a sign of reconnaissance activity or malware trying to capture sensitive data.

2. Listened ports status (netstat) changed (new port opened or closed).

- Rule ID: 533
- Rule Level: 7
  What It Means:
  Wazuh detected a change in the list of open (listening) ports on the system, based on netstat output comparison. This might mean:
    - A new service started listening on a port, or

- An existing service closed its port.

Why It Matters:
These changes may indicate:

- A scan has probed the system, causing responses.
- A malicious or unauthorized service has been launched.
- Normal but noteworthy activity (e.g., system update or config change).

In our Case:
This likely resulted from our Nmap scan, which attempts to connect to ports, triggering a change in the host's network behavior.

3. Host-based anomaly detection event (rootcheck)

- Rule ID: 510
- Rule Level: 5

What It Means:
This alert is from Wazuh's rootcheck module, which scans the system for:

- Suspicious binaries
- Modified files
- Malware patterns
- Rootkit signs

Why It Matters:
It's a host-based intrusion detection alert, pointing to anomalies that might indicate a compromise attempt or misconfiguration.

Project Summary:

I installed VMware Workstation and set up a Cybersecurity Lab, Kali Linux, and Wazuh.

I added the Cybersecurity Lab VM to Wazuh as an agent.

Then, I used Kali Linux to scan the IP address of the Cybersecurity Lab to check which ports are open or closed.

After scanning, Wazuh showed alerts like:

- Sniffing (promiscuous mode) detected
- Ports opened/closed (due to scanning)
- Anomaly in the system

This shows that Wazuh can detect port scanning activities.

Author Details

Name: Priyanka H S
Project Title: Port Scanning, Packet Analysis, and Network Monitoring using Nmap, Wireshark, and Wazuh