Port Scanning Detection Using Wireshark

Abstract:
This project aims to detect and analyze port scanning techniques using Wireshark, a network packet analyzer. It explores TCP and UDP scan types performed using Nmap and illustrates how to identify scanning based on packet patterns.

Objectives:

- **Understand port scanning methods and their intent.**
- **Use Nmap to simulate scanning scenarios.**
- **Capture packets using Wireshark.**
- **Analyze packet flows to detect scan behavior.**

Tools Used:

- **Wireshark**
- **Nmap**
- **Linux OS**
- **TCP/IP Stack**

Port Scanning Detection Using Wire shark

Port Scanning

What is Port Scanning?

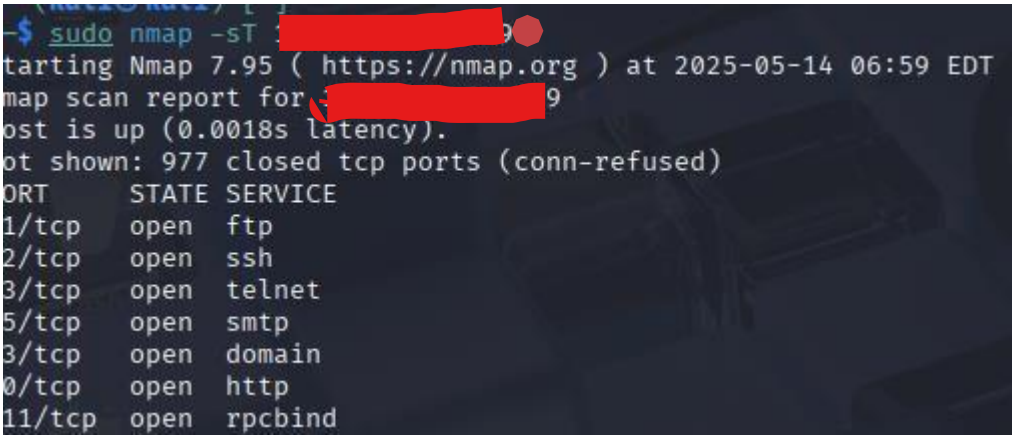
Port scanning is a technique used to determine which ports on a network host are open and listening for connections. In TCP/IP networking, ports are virtual endpoints that allow different applications and services to run on the same device.

Causes of Port Scanning

Port scanning can be caused by various factors, including:

- **Misconfiguration: A system might be misconfigured, causing it to send out port scans unintentionally.**
- **Malware: Malware might use port scanning to find vulnerable systems to infect.**
- **Vulnerability Research: Security researchers might conduct port scans to identify new vulnerabilities.**
- **Cyber Warfare: Nation-state actors might use port scanning as part of a larger cyber warfare campaign.**

Types of Nmap Scans:



-sT (TCP Connect Scan): Nmap completes the full TCP three-way handshake to open a connection, revealing open ports.

```
└─$ sudo nmap -sS [redacted]
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 06:57 EDT
Nmap scan report for 192.168.240.129
Host is up (0.0037s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  telnet
25/tcp   open  smtp
53/tcp   open  domain
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
512/tcp  open  exec
513/tcp  open  login
514/tcp  open  shell
1099/tcp open  rmiregistry
1524/tcp open  ingreslock
2049/tcp open  nfs
2121/tcp open  ccproxy-ftp
3306/tcp open  mysql
5432/tcp open  postgresql
5900/tcp open  vnc
6000/tcp open  X11
6667/tcp open  irc
8009/tcp open  ajp13
8180/tcp open  unknown
MAC Address: 00:0C:29:7E:3E:85 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 0.40 seconds
```

-sS(TCP SYN Scan): Nmap sends a SYN packet and checks for a SYN/ACK response, indicating an open port, without completing the handshake.

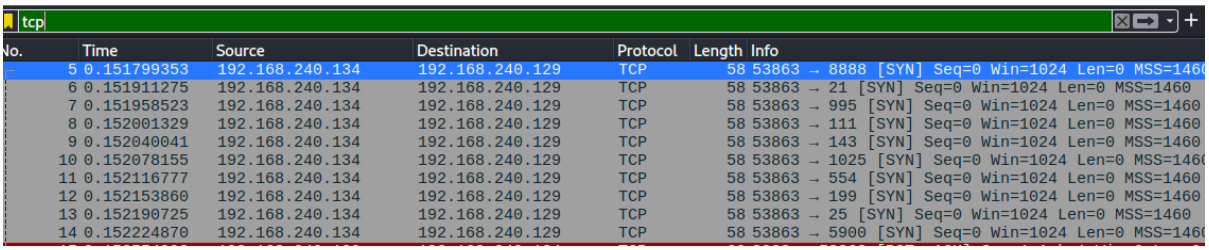
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sU [redacted]
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-14 07:02 EDT
Stats: 0:02:13 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.81% done; ETC: 07:18 (0:13:50 remaining)
Stats: 0:02:15 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 13.92% done; ETC: 07:18 (0:13:48 remaining)

┌──(kali㉿kali)-[~]
└─$
```

-sU (UDP Scan): Nmap sends UDP packets and waits for responses, like an ICMP error if the port is closed, to determine UDP port status.

Port Scanning Detection using wire shark

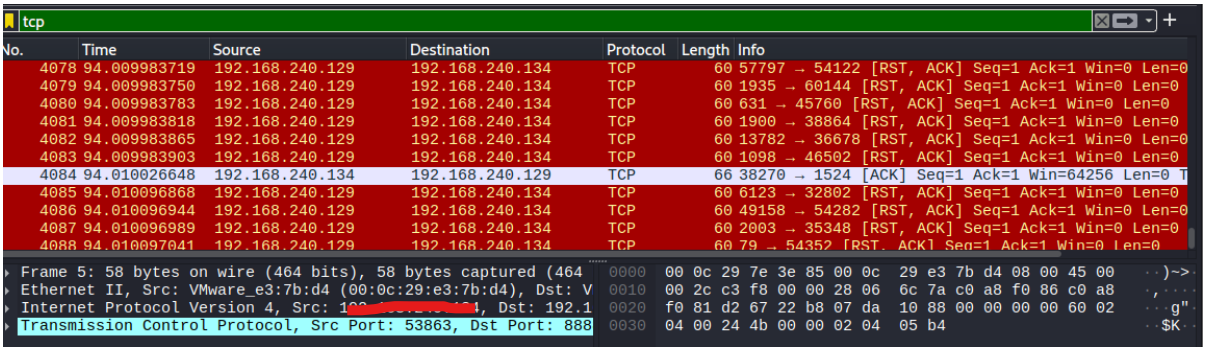
Detection of Syn scan using wire shark:



tcp

No.	Time	Source	Destination	Protocol	Length	Info
5	0.151799353	192.168.240.134	192.168.240.129	TCP	58	53863 → 8888 [SYN] Seq=0 Win=1024 Len=0 MSS=146
6	0.151911275	192.168.240.134	192.168.240.129	TCP	58	53863 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
7	0.151958523	192.168.240.134	192.168.240.129	TCP	58	53863 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
8	0.152001329	192.168.240.134	192.168.240.129	TCP	58	53863 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
9	0.152040041	192.168.240.134	192.168.240.129	TCP	58	53863 → 143 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
10	0.152078155	192.168.240.134	192.168.240.129	TCP	58	53863 → 1025 [SYN] Seq=0 Win=1024 Len=0 MSS=146
11	0.152116777	192.168.240.134	192.168.240.129	TCP	58	53863 → 554 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
12	0.152153860	192.168.240.134	192.168.240.129	TCP	58	53863 → 199 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
13	0.152190725	192.168.240.134	192.168.240.129	TCP	58	53863 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
14	0.152224870	192.168.240.134	192.168.240.129	TCP	58	53863 → 5900 [SYN] Seq=0 Win=1024 Len=0 MSS=146

Detection of TCP three-way handshake:



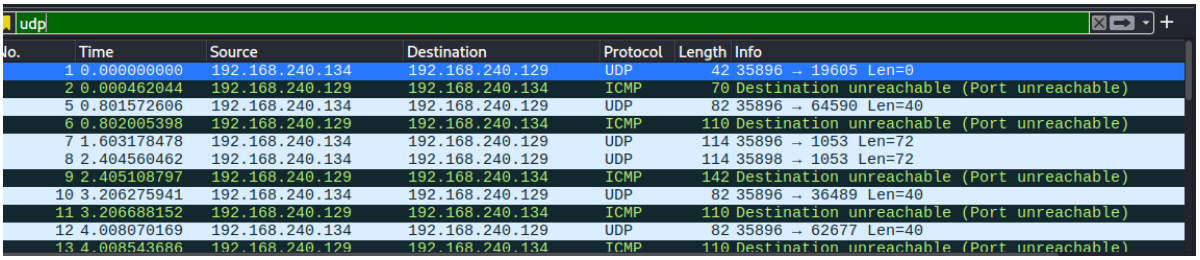
tcp

No.	Time	Source	Destination	Protocol	Length	Info
4078	94.009983719	192.168.240.129	192.168.240.134	TCP	60	57797 → 54122 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4079	94.009983750	192.168.240.129	192.168.240.134	TCP	60	1935 → 60144 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4080	94.009983783	192.168.240.129	192.168.240.134	TCP	60	631 → 45760 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4081	94.009983818	192.168.240.129	192.168.240.134	TCP	60	1900 → 38864 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4082	94.009983865	192.168.240.129	192.168.240.134	TCP	60	13782 → 36678 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4083	94.009983903	192.168.240.129	192.168.240.134	TCP	60	1098 → 46502 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4084	94.010026648	192.168.240.134	192.168.240.129	TCP	66	38270 → 1524 [ACK] Seq=1 Ack=1 Win=64256 Len=0 T
4085	94.010096868	192.168.240.129	192.168.240.134	TCP	60	6123 → 32802 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4086	94.010096944	192.168.240.129	192.168.240.134	TCP	60	49158 → 54282 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4087	94.010096989	192.168.240.129	192.168.240.134	TCP	60	2003 → 35348 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
4088	94.010097041	192.168.240.129	192.168.240.134	TCP	60	79 → 54352 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0

```
▸ Frame 5: 58 bytes on wire (464 bits), 58 bytes captured (464
▸ Ethernet II, Src: VMware_e3:7b:d4 (00:0c:29:e3:7b:d4), Dst: V
▸ Internet Protocol Version 4, Src: 1[redacted]4, Dst: 192.1
▸ Transmission Control Protocol, Src Port: 53863, Dst Port: 888

0000  00 0c 29 7e 3e 85 00 0c  29 e3 7b d4 08 00 45 00   ··)~>···  ··)·{···  ··E·
0010  00 2c c3 f8 00 00 28 06  6c 7a c0 a8 f0 86 c0 a8   ·,····(·  lz····
0020  f0 81 d2 67 22 b8 07 da  10 88 00 00 00 00 60 02   ···g"···  ····`·
0030  04 00 24 4b 00 00 02 04  05 b4                     ··$K····  ····
```

Detection of Udp Scan:



udp

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	192.168.240.134	192.168.240.129	UDP	42	35896 → 19605 Len=0
2	0.000462044	192.168.240.129	192.168.240.134	ICMP	70	Destination unreachable (Port unreachable)
5	0.801572606	192.168.240.134	192.168.240.129	UDP	82	35896 → 64590 Len=40
6	0.802005398	192.168.240.129	192.168.240.134	ICMP	110	Destination unreachable (Port unreachable)
7	1.603178478	192.168.240.134	192.168.240.129	UDP	114	35896 → 1053 Len=72
8	2.404560462	192.168.240.134	192.168.240.129	UDP	114	35898 → 1053 Len=72
9	2.405108797	192.168.240.129	192.168.240.134	ICMP	142	Destination unreachable (Port unreachable)
10	3.206275941	192.168.240.134	192.168.240.129	UDP	82	35896 → 36489 Len=40
11	3.206688152	192.168.240.129	192.168.240.134	ICMP	110	Destination unreachable (Port unreachable)
12	4.008070169	192.168.240.134	192.168.240.129	UDP	82	35896 → 62677 Len=40
13	4.008543686	192.168.240.129	192.168.240.134	ICMP	110	Destination unreachable (Port unreachable)