Cyber Security Tools and Technologies

Dr. Arshad Ali
Associate Professor
NUCES Lahore

Acknowledgement

- **TryHackMe (https://tryhackme.com/)**

Nmap Basic Port Scans

- Task 1: Introduction
- Task 2: TCP and UDP Ports
- Task 3: TCP Flags
- Task 4: TCP Connect Scan
- Task 5: TCP SYN Scan
- Task 6: UDP Scan
- Task 7: Fine-Tuning Scope and Performance
- Task 8: Summary

1. Introduction

First room of Nmap: focused on discovering online systems.

- Covered three steps of a Nmap scan:

I. Enumerate targets

II. Discover live hosts

III. Reverse-DNS lookup

Next step: checking which ports are open and listening and which ports are closed.

1. Introduction

This room and the next one: focus on port scanning and the different types of port scans used by nmap

It explains

I. TCP connect port scan
II. TCP SYN port scan
III. UDP port scan

Moreover: it discussed the different options to specify the ports, the scan rate, and the number of parallel probes.

2. TCP and UDP Port Scans

- Just like an IP address specifies a host on a network among many others
- a TCP port or UDP port is used to identify a network service running on that host.
  - A **server** provides network service, and it adheres to a specific network protocol.
  - **Examples**: providing time, responding to DNS queries, and serving web pages.
  - A port is usually linked to a service using that specific port number.
  - **Ex:** an HTTP server would bind to TCP port 80 by default;

2.TCP and UDP Port Scans

- If the HTTP server supports SSL/TLS, it would listen on TCP port 443.
- TCP ports 80 and 443 are the default ports for HTTP and HTTPS
- no more than one service can listen on any TCP or UDP port (on the same IP address).

- We can classify ports in two states:
- **Open port:** some service listening on that port.
- **Closed port:** no service listening on that port.
- However, in practical situations, we need to consider the **impact of firewalls**.
- Like, a port might be open, but a firewall might be blocking the packets.

2. TCP and UDP Port Scans

Nmap considers the following **six states**:

1. Open: a service is listening on the specified port.

2. Closed: no service is listening on the specified port, although the port is accessible.

- Accessible means that it is reachable and is not blocked by a firewall or other security appliances / programs.

3. Filtered: Nmap cannot determine if the port is open or closed as it is not accessible.

- usually due to a firewall preventing Nmap from reaching that port.
- Nmap's packets may be blocked from reaching the port;
- alternatively, the responses are blocked from reaching Nmap's host.

2. TCP and UDP Port Scans

4. Unfiltered: means that Nmap cannot determine if the port is open or closed, although the port is accessible.

- This state is encountered when using an ACK scan

-sA.

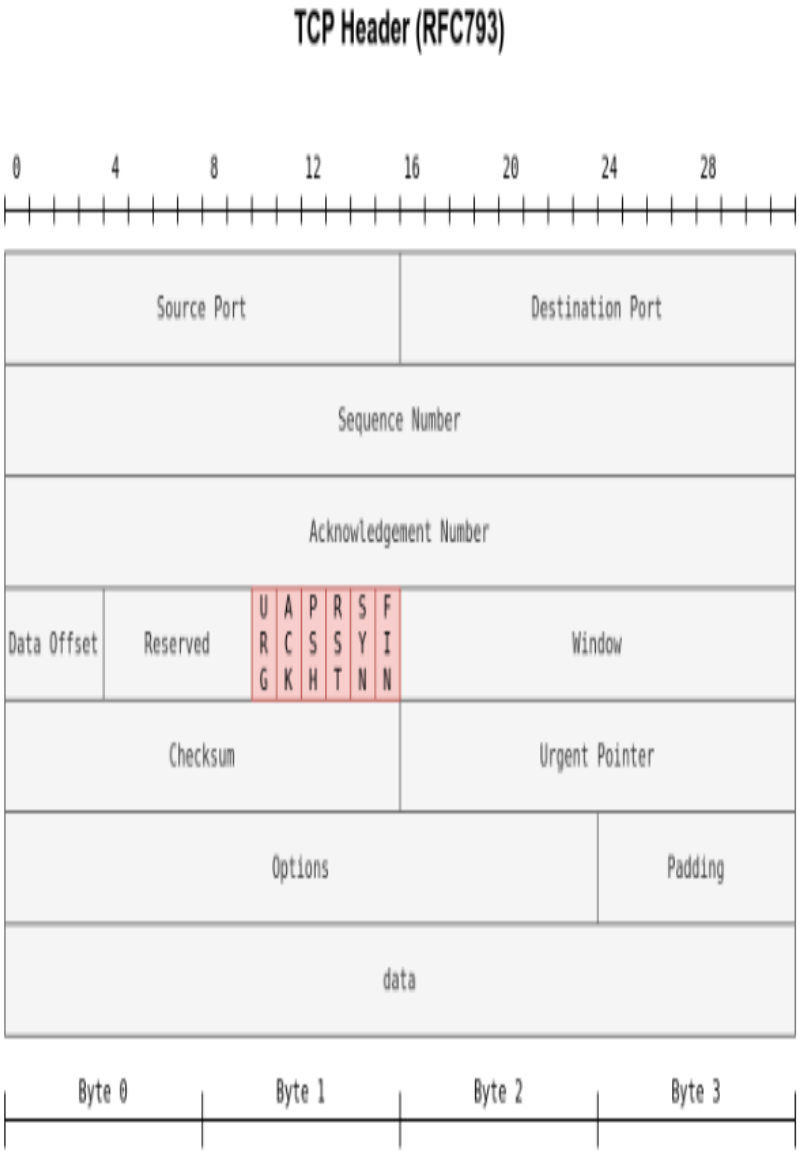
5. Open|Filtered: This means that Nmap cannot determine whether the port is open or filtered.

6. Closed|Filtered: This means that Nmap cannot decide whether a port is closed or filtered.

3. TCP Flags

- Nmap supports different types of TCP port scans.
- First brief review about the TCP header.

In the first row, we have the source TCP port number and the destination port number.



3. TCP Flags

- **TCP header flags** that Nmap can set or unset

1. URG: it indicates that

- the urgent pointer filed is significant and the incoming data is urgent,
- a TCP segment with the URG flag set is processed immediately without consideration of having to wait on previously sent TCP segments.

2. ACK: ACK number is significant. It is used ACK the receipt of a TCP segment.

3. PSH: it asks TCP to pass the data to the application promptly.

4. RST: It is used to reset the connection.

- Another device, such as a firewall, might send it to tear a TCP connection.
- This flag is also used when data is sent to a host and there is no service on the receiving end to answer.

3. TCP Flags

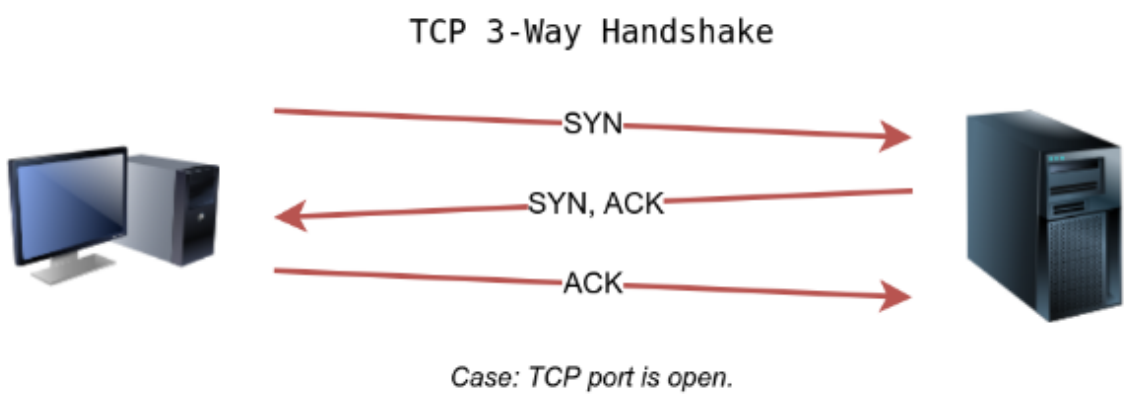
5. SYN: Itis used to initiate a TCP 3-way handshake and synchronize sequence numbers with the other host.

- The sequence number should be set randomly during TCP connection establishment.

6. FIN: The sender has no more data to send.

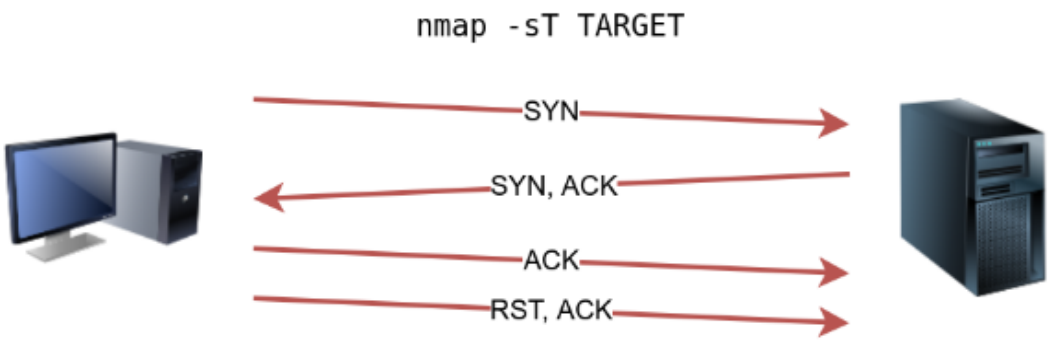
4. TCP Connect Scan

- TCP connect scan works by completing the TCP 3-way handshake.
- In standard TCP connection establishment, the client sends a TCP packet with SYN flag set, and the server responds with SYN/ACK if the port is open;
- finally, the client completes the 3-way handshake by sending an ACK.



4. TCP Connect Scan

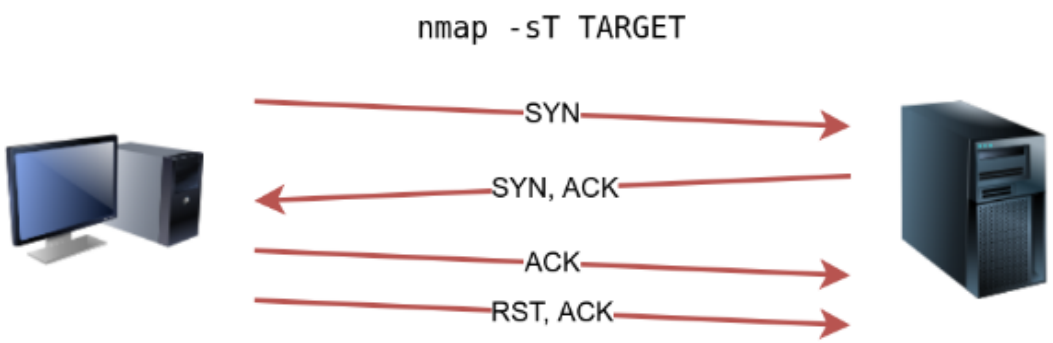
- We are interested in learning whether the TCP port is open, not establishing a TCP connection.
- So, the connection is torn as soon as its state is confirmed by sending a RST/ACK.
- You can choose to run TCP connect scan using -sT



Case: TCP port is open.

4. TCP Connect Scan

- We are interested in learning whether the TCP port is open, not establishing a TCP connection.
- Hence the connection is torn as soon as its state is confirmed by sending a RST/ACK.
- You can choose to run TCP connect scan using -sT

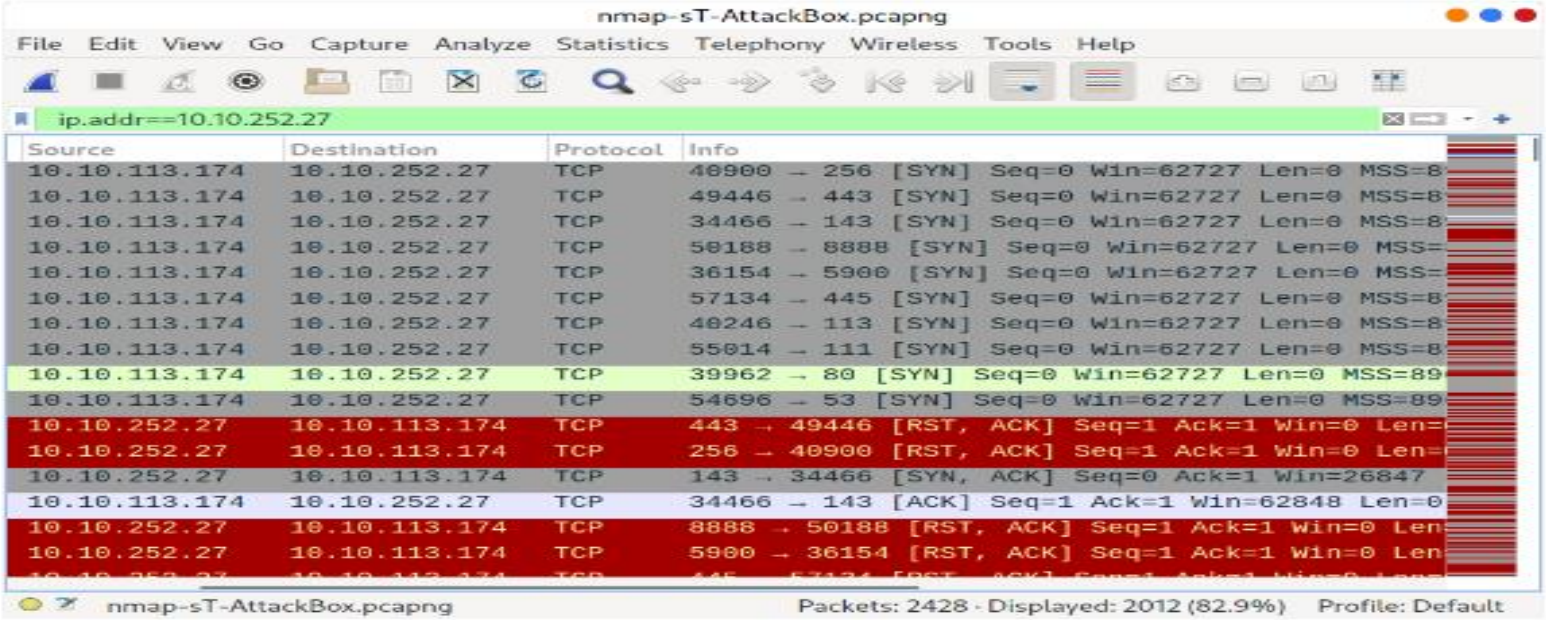


Case: TCP port is open.

Note: if you are not a privileged user (root or sudoer), a TCP connect scan is the only possible option to discover open TCP ports.

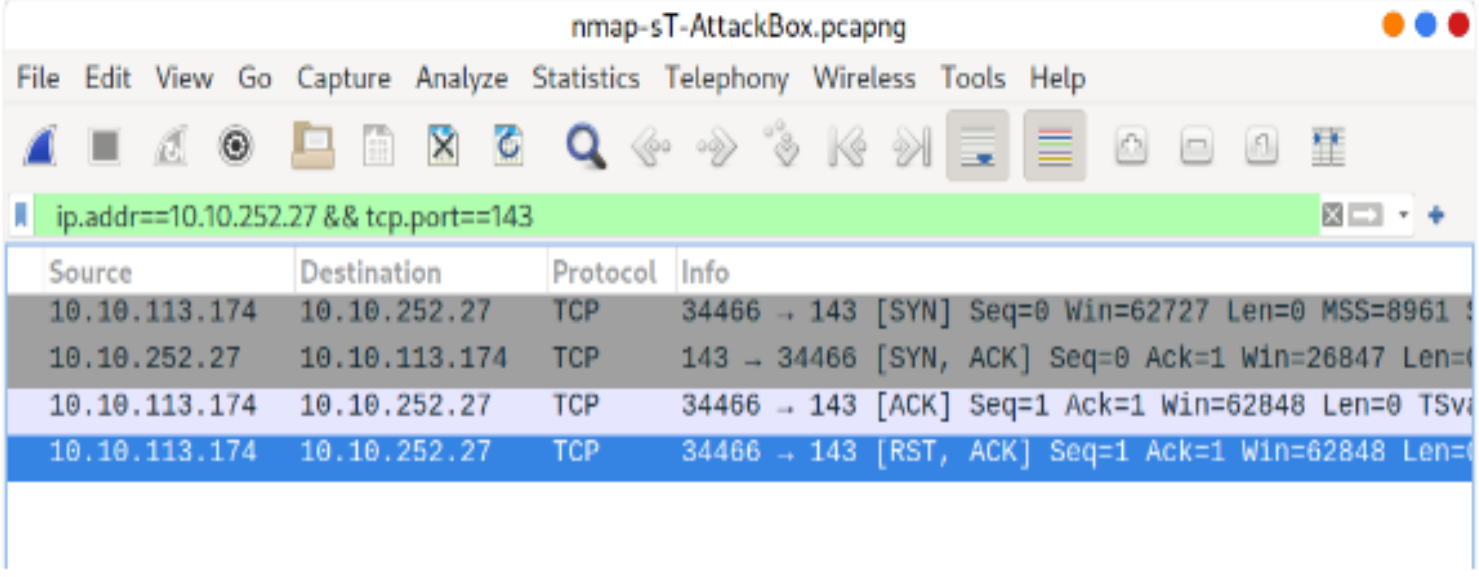
4. TCP Connect Scan

- In Wireshark packet capture window
  - Nmap sending TCP packets with SYN flag set to various ports, 256, 443, 143, and so on
- By default, Nmap attempts to connect to the **1000** most common ports.
- A closed TCP port responds to a SYN packet with RST/ACK to indicate that it is not open.



4. TCP Connect Scan

- Note that port 143 is open, so it replied with a SYN/ACK, and Nmap sent an ACK to complete the 3-way handshake
- Then, the fourth packet tears it down with an RST/ACK packet.
-



4. TCP Connect Scan

- The following command returned a detailed list of the open ports
  - -sT (TCP Connect Scan)
- We can use **–F** enable fast mode and decrease the number of scanned ports from 1000 to 100 most common ports
- **-r** option can also be added to scan the ports in consecutive order instead of random order.
- This option is useful when testing whether ports open in a consistent manner, for instance, when a target boots up.

Pentester Terminal

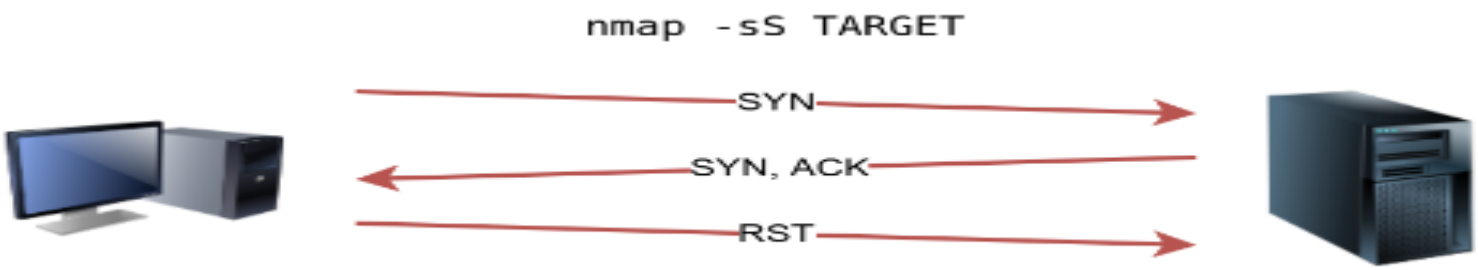
```
pentester@TryHackMe$ nmap -sT MACHINE_IP

Starting Nmap 7.60 ( https://nmap.org ) at 2021-08-30 09:53 BST
Nmap scan report for MACHINE_IP
Host is up (0.0024s latency).
Not shown: 995 closed ports
PORT     STATE SERVICE
22/tcp   open  ssh
25/tcp   open  smtp
80/tcp   open  http
111/tcp  open  rpcbind
143/tcp  open  imap
MAC Address: 02:45:BF:8A:2D:6B (Unknown)

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

5. TCP SYN Scan

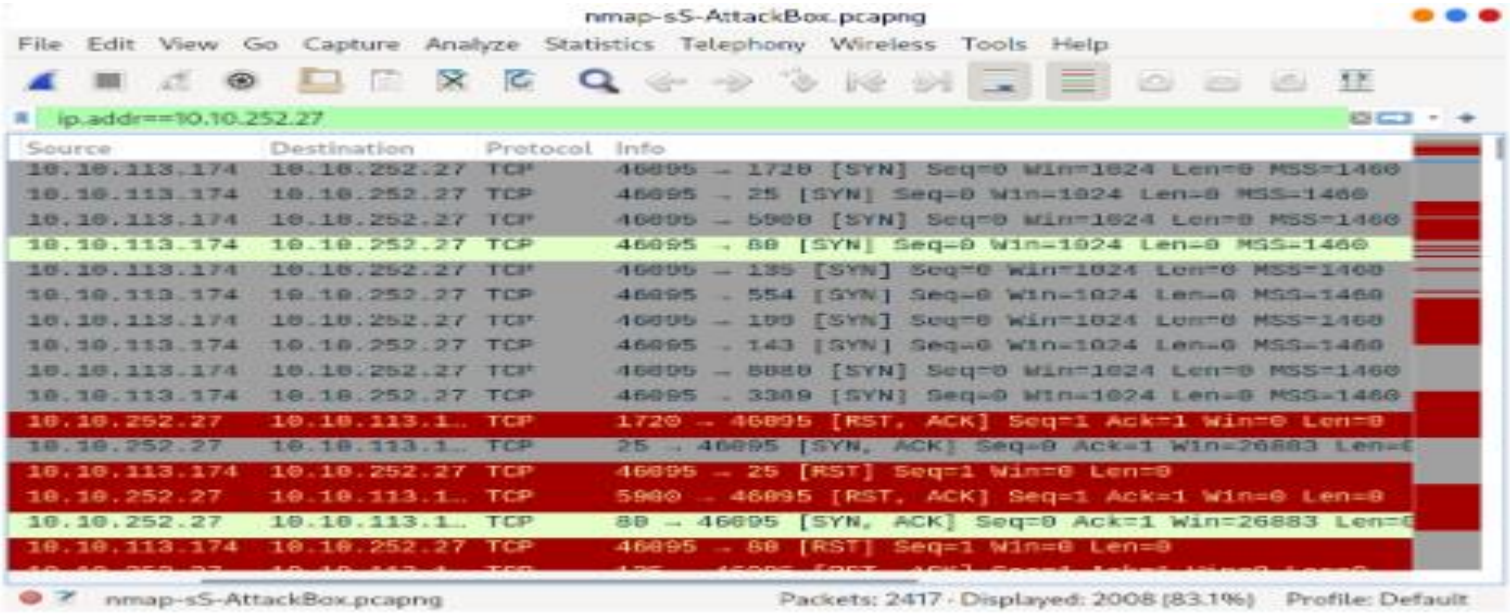
- Unprivileged users are limited to connect scan.
- However, the default scan mode is SYN scan, and it requires a privileged (root or sudoer) user to run it.
- SYN scan does not need to complete the TCP 3-way handshake; instead, it tears down the connection once it receives a response from the server.
- Because we didn't establish a TCP connection, this decreases the chances of the scan being logged.
- We can select this scan type using -sS option



Case: TCP port is open.

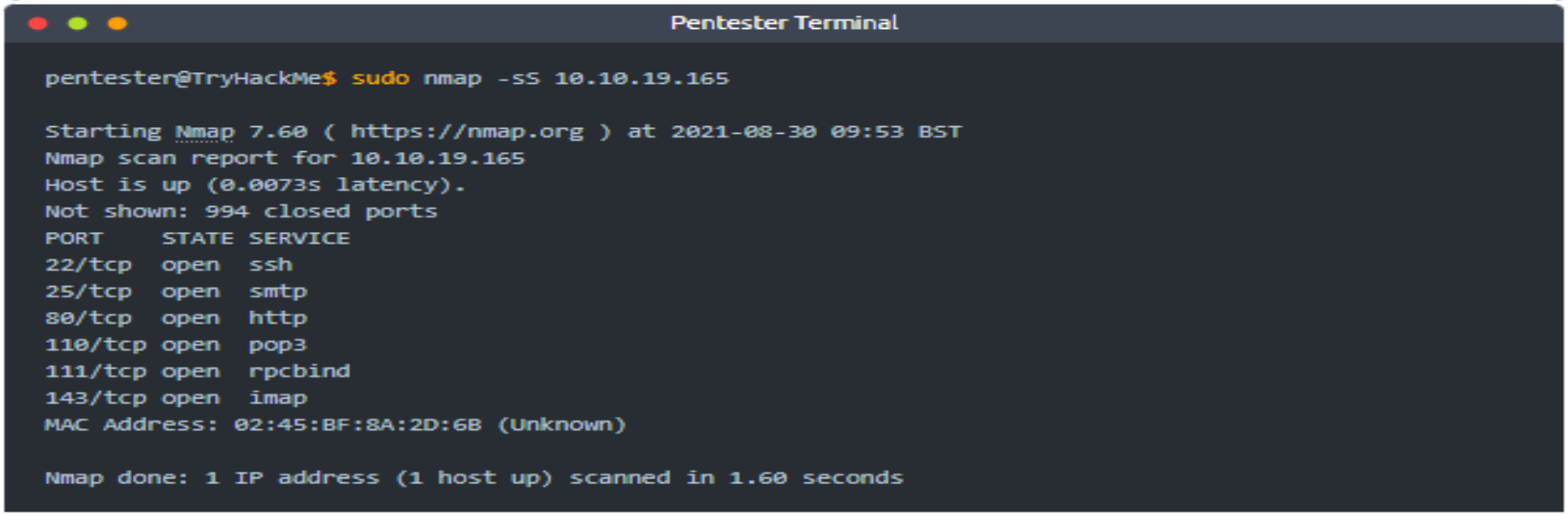
5. TCP SYN Scan

- The following screenshot from Wireshark shows a TCP SYN scan. The behaviour in the case of closed TCP ports is similar to that of the TCP connect scan.



5. TCP SYN Scan

- Difference between the two scans (TCP CONNECT vs SYN)

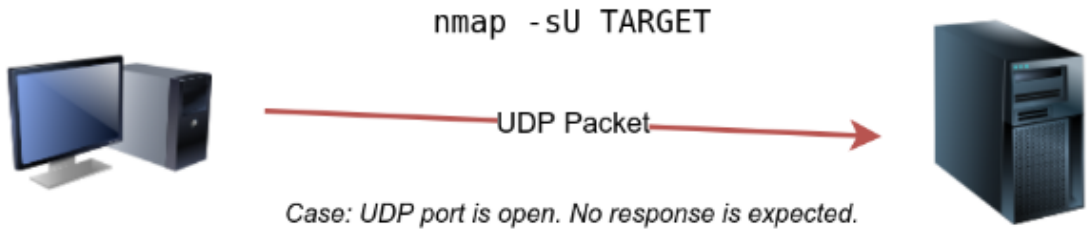


6. UDP Scan

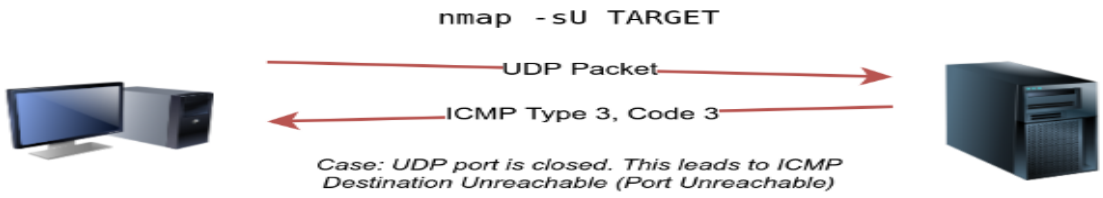
- UDP: a connectionless protocol, does not require any handshake for connection establishment.
- We cannot guarantee that a service listening on a UDP port would respond to our packets.
- However, if a UDP packet is sent to a closed port, an ICMP port unreachable error (type 3, code 3) is returned.
- You can select UDP scan using the -sU option;
- If we send a UDP packet to an open UDP port, we cannot expect any reply in return.

6. UDP Scan

- If we send a UDP packet to an open UDP port, we cannot expect any reply in return.



- we expect to get an ICMP packet of type 3, destination unreachable, and code 3, port unreachable.
- every closed port will generate an ICMP packet



7. Fine-Tuning Scope and Performance

- Control the scan timing using -T<0-5>
- -T0 is the slowest while –T5 is the fastest
- According to Nmap manual page, there are six templates:
- paranoid (0)
- sneaky (1)
- polite (2)
- normal (3)
- aggressive (4)
- insane (5)

To avoid IDS alerts, you might consider -T0 or –T1

-T0 scans one port at a time and waits 5 minutes between sending each probe

7. Fine-Tuning Scope and Performance

- -T4 is often used during CTFs and when learning to scan on practice targets, whereas

- -T1 is often used during real engagements where stealth is more important.
- Alternatively, you can choose to control the packet rate using
- **--min-rate <number> and --max-rate <number>**
  - --min-rate 10 or --max-rate =10
  - ensures that your scanner is not sending more than ten packets per second

7. Fine-Tuning Scope and Performance

- Moreover, you can control probing parallelization using

min parallelism <numprobes?

- Nmap probes the targets to discover which hosts are live and which ports are open;
- probing parallelization specifies the number of such probes that can be run in parallel.
- **Example: min parallelism = 512**

pushes Nmap to maintain at least 512 probes in parallel

7. Fine-Tuning Scope and Performance

Option	Purpose
-p-	all ports
-p1-1023	scan ports 1 to 1023
-F	100 most common ports
-r	scan ports in consecutive order
-T<0-5>	-T0 being the slowest and T5 the fastest
--max-rate 50	rate <= 50 packets/sec
--min-rate 15	rate >= 15 packets/sec
--min-parallelism 100	at least 100 probes in parallel

7. Summary

- This room covered three types of scans.

Port Scan Type	Example Command
TCP Connect Scan	nmap -sT 10.10.19.165
TCP SYN Scan	sudo nmap -sS 10.10.19.165
UDP Scan	sudo nmap -sU 10.10.19.165