PRACTICAL-1

AIM:

Perform port scanning using nmap on a single port and capture the packets using wireshark and analyze the output

THEORY:

1. Nmap:

- Nmap is a free and open-source network scanner created by Gordon Lyon. Nmap is used
 to discover hosts and services on a computer network by sending packets and analyzing
 the responses.
- Nmap provides a number of features for probing computer networks, including host discovery and service and operating system detection.
- These features are extensible by scripts that provide more advanced service detection, vulnerability detection, and other features.
- Nmap can adapt to network conditions including latency and congestion during a scan.
- Nmap started as a Linux utility and was ported to other systems including Windows, macOS, and BSD. It is most popular on Linux, followed by Windows.

2. Wireshark:

- Wireshark is a free and open-source packet analyser.
- It is used for network troubleshooting, analysis, software and communications protocol development, and education.
- Originally named Ethereal, the project was renamed Wireshark in May 2006 due to trademark issues.
- Wireshark is cross-platform, using the Qt widget toolkit in current releases to implement its user interface, and using pcap to capture packets; it runs on Linux, macOS, BSD, Solaris, some other Unix-like operating systems, and Microsoft Windows.

• There is also a terminal-based (non-GUI) version called TShark. Wireshark, and the other programs distributed with it such as TShark, are free software, released under the terms of version 2 of the GNU General Public License.

3. open

- An application is actively accepting TCP connections, UDP datagrams or SCTP associations on this port.
- Finding these is often the primary goal of port scanning. Security-minded people know that each open port is an avenue for attack.
- Attackers and pen-testers want to exploit the open ports, while administrators try to close or protect them with firewalls without thwarting legitimate users.
- Open ports are also interesting for non-security scans because they show services available for use on the network.

4. closed

- A closed port is accessible (it receives and responds to Nmap probe packets), but there
 is no application listening on it.
- They can be helpful in showing that a host is up on an IP address (host discovery, or ping scanning), and as part of OS detection. Because closed ports are reachable, it may be worth scanning later in case some open up.
- Administrators may want to consider blocking such ports with a firewall. Then they would appear in the filtered state, discussed next.

5. filtered

- Nmap cannot determine whether the port is open because packet filtering prevents its probes from reaching the port.
- The filtering could be from a dedicated firewall device, router rules, or host-based firewall software.
- These ports frustrate attackers because they provide so little information. Sometimes
 they respond with ICMP error messages such as type 3 code 13 (destination
 unreachable: communication administratively prohibited), but filters that simply drop
 probes without responding are far more common.
- This forces Nmap to retry several times just in case the probe was dropped due to network congestion rather than filtering. This slows down the scan dramatically.

6. Unfiltered

- The unfiltered state means that a port is accessible, but Nmap is unable to determine whether it is open or closed.
- Only the ACK scan, which is used to map firewall rulesets, classifies ports into this state.
- Scanning unfiltered ports with other scan types such as Window scan, SYN scan, or FIN scan, may help resolve whether the port is open.

7. open|filtered

- Nmap places ports in this state when it is unable to determine whether a port is open or filtered. This occurs for scan types in which open ports give no response.
- The lack of response could also mean that a packet filter dropped the probe or any response it elicited.
- So Nmap does not know for sure whether the port is open or being filtered. The UDP, IP protocol, FIN, NULL, and Xmas scans classify ports this way.

8. closed|filtered

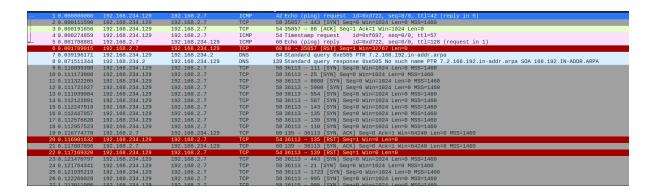
- This state is used when Nmap is unable to determine whether a port is closed or filtered.
- It is only used for the IP ID idle scan.

IMPLEMENTATION:

- Start the wireshark and start capturing the packets
- Firstly, we will write the following command:
- Write sudo nmap ip address of device
- This is the basic format for **Nmap**, and it will return information about the ports on that system.

```
sudo nmap 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 09:34 EST
Nmap scan report for 192.168.2.7
Host is up (0.0029s latency).
Not shown: 994 filtered tcp ports (no-response)
        STATE SERVICE
PORT
135/tcp open msrpc
139/tcp open netbios-ssn
445/tcp open microsoft-ds
902/tcp open
              iss-realsecure
2869/tcp open
              icslap
7070/tcp open realserver
Nmap done: 1 IP address (1 host up) scanned in 4.75 seconds
```

 Below are the glimpses of the packets captured by wireshark when the above command was executed



• Below are the packets captured for PORT 135



- If we want to scan for a range of ip address then, enter the following command
- Write sudo nmap ip address range

```
[root  kali]-[~]

# sudo nmap 192.168.2.7-21

Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 09:49 EST
```

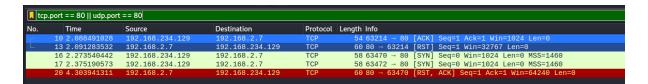
- You will get the result of scan for the whole range
- To know the status of a particular port, enter the following command

```
(root © kali)-[~]
    sudo nmap -p 80 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:32 EST
Nmap scan report for 192.168.2.7
Host is up (0.0033s latency).

PORT STATE SERVICE
80/tcp filtered http

Nmap done: 1 IP address (1 host up) scanned in 0.61 seconds
```

• Packets captured in wireshark



• For multiple ports, type the following command

```
(root kali)-[~]

# sudo nmap -p 80,443 192.168.2.7

Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:37 EST

Nmap scan report for 192.168.2.7

Host is up (0.0011s latency).

PORT STATE SERVICE

80/tcp filtered http

443/tcp filtered https

Nmap done: 1 IP address (1 host up) scanned in 5.43 seconds
```

Packets captured by Wireshark

■ tcp	T tcp.port == 80 tcp.port==443							
No.	Time	Source	Destination	Protocol	Length Info			
	2 0.000058156	192.168.234.129	192.168.2.7	TCP	58 45726 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460			
	3 0.000097921	192.168.234.129	192.168.2.7	TCP	54 45726 → 80 [ACK] Seq=1 Ack=1 Win=1024 Len=0			
	6 0.001075758	192.168.2.7	192.168.234.129	TCP	60 80 → 45726 [RST] Seq=1 Win=32767 Len=0			
	11 2.037711847	192.168.2.7	192.168.234.129	TCP	60 443 → 45726 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0			
	14 4.104896081	192.168.234.129	192.168.2.7	TCP	58 45982 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460			
	15 4.104996848	192.168.234.129	192.168.2.7	TCP	58 45982 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460			
	16 5.210332239	192.168.234.129	192.168.2.7	TCP	58 45984 → 443 [SYN] Seq=0 Win=1024 Len=0 MSS=1460			
	17 5.210406532	192.168.234.129	192.168.2.7	TCP	58 45984 → 80 [SYN] Seq=0 Win=1024 Len=0 MSS=1460			

• To scan all the possible ports, write the following command

• To scan for all available TCP ports, enter the following command

```
(root © kali)-[~]
# sudo nmap -p0 192.168.2.7
Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:42 EST
Nmap scan report for 192.168.2.7
Host is up (0.0011s latency).

PORT STATE SERVICE
0/tcp filtered unknown
```

• To go for tcp syn scan, enter the following command

• Some of the packets captured in wireshark

2914 87.844507018	192.168.234.129	192.168.2.7	TCP	58 43763 → 6699 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2915 87.952750504	192.168.234.129	192.168.2.7	TCP	58 43759 → 1114 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2916 87.991386158	192.168.2.7	192.168.234.129	TCP	60 1163 → 43759 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
2917 88.057185122	192.168.234.129	192.168.2.7	TCP	58 43761 → 1114 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2918 88.160342768	192.168.234.129	192.168.2.7	TCP	58 43763 → 1114 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2919 88.264821098	192.168.234.129	192.168.2.7	TCP	58 43759 → 20222 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2920 88.310870228	192.168.2.7	192.168.234.129	TCP	60 9999 → 43759 [RST, ACK] Seq=1 Ack=1 Win=64240 Len=0
2921 88.367499724	192.168.234.129	192.168.2.7	TCP	58 43884 → 5902 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2922 88.470674636	192.168.234.129	192.168.2.7	TCP	58 43761 → 20222 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2923 88.575796086	192.168.234.129	192.168.2.7	TCP	58 43763 → 20222 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2924 88.680052776	192.168.234.129	192.168.2.7	TCP	58 43759 → 5009 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
2925 88.784195629	192 168 234 129	192 168 2 7	TCP	58 43761 → 5009 [SYN] Seg=0 Win=1024 Len=0 MSS=1460

• To scan for ping scan, enter the following command

```
(root⊕ kali)-[~]

# sudo nmap -sP 192.168.2.7

Starting Nmap 7.92 ( https://nmap.org ) at 2022-02-06 11:50 EST

Nmap scan report for 192.168.2.7

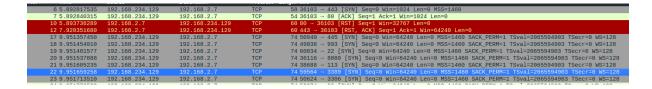
Host is up (0.0011s latency).

Nmap done: 1 IP address (1 host up) scanned in 13.16 seconds
```

N i	[A] icmp							
No.	Time	Source	Destination	Protocol	Length Info			
→	3 10.941095208	192.168.234.129	192.168.2.7	ICMP	42 Echo (ping) request	id=0x1a05, seq=0/0, ttl=47 (reply in 7)		
	6 10.941269259	192.168.234.129	192.168.2.7	ICMP	54 Timestamp request	id=0xda12, seq=0/0, ttl=40		
+	7 10.942182743	192.168.2.7	192.168.234.129	ICMP	60 Echo (ping) reply	id=0x1a05, seq=0/0, ttl=128 (request in 3)		

• For TCP Connect scan, enter the following command

Wireshark packets captured



• For UDP Scan, use the following command

• From the wireshark

				,
9 0.085415924	192.168.234.129	192.168.2.7	RADIUS	62 Access-Request id=0
10 0.085498360	192.168.234.129	192.168.2.7	UDP	82 45784 → 64080 Len=40
11 0.085530128	192.168.234.129	192.168.2.7	UDP	42 45784 → 902 Len=0
12 0.085561512	192.168.234.129	192.168.2.7	UDP	82 45784 → 49198 Len=40
13 0.085585926	192.168.234.129	192.168.2.7	UDP	82 45784 → 51554 Len=40
14 0.085637670	192.168.234.129	192.168.2.7	UDP	42 45784 → 21212 Len=0
15 0.085677079	192.168.234.129	192.168.2.7	UDP	42 45784 → 17823 Len=0
16 0.085720814	192.168.234.129	192.168.2.7	UDP	82 45784 → 57410 Len=40
17 0.085764765	192.168.234.129	192.168.2.7	UDP	42 45784 → 539 Len=0

CONCLUSION:

• By performing the above practical, I learnt about the basics of nmap and it's functionalities