Ex:No:12

Date:

12.Nmap to discover live hosts using ARP scan, ICMP scan, and TCP/UDP ping scan in TryHackMe Platform.

Aim:

To use the **Nmap** network scanning tool to perform live host discovery on a target network using **ARP Scan**, **ICMP Scan**, and **TCP/UDP Ping Scan** techniques, documenting the methodology and results for each to understand their differences and effectiveness.

Nmap Host Discovery Techniques and Commands:

The primary Nmap command to disable the default port scanning and focus only on host discovery is **-sn** (formerly -sP).

A. ARP Scan (Address Resolution Protocol)

This technique is effective only on the **local subnet** (Layer 2). Nmap sends an ARP request for each target IP and considers a host "up" if it receives an ARP reply (which contains the host's MAC address). This bypasses most ICMP-based firewalls.

Parameter Description

-PR ARP Ping (Address Resolution Protocol).

-sn Disable port scan (host discovery only).

Command Syntax:

Bash

sudo nmap -sn -PR <TARGET_IP_RANGE># Example: sudo nmap -sn -PR 10.10.10.0/24

Expected Outcome: Hosts that are alive on the local subnet will respond with their MAC address and will be listed as **Host is up**.

B. ICMP Scan (Internet Control Message Protocol)

This is the classic "ping" method. Nmap sends an ICMP Echo Request and considers the host "up" if it receives an ICMP Echo Reply. Two common variations are used to bypass simple filtering:

Parameter	Description
-PE	ICMP Echo Request (standard ping).
-PP	ICMP Timestamp Request.
-sn	Disable port scan (host discovery only).

Command Syntax (ICMP Echo):

Bash

```
nmap -sn -PE <TARGET_IP_OR_RANGE># Example: nmap -sn -PE 10.10.10.5
```

Command Syntax (ICMP Timestamp - often less filtered):

Bash

```
nmap -sn -PP <TARGET_IP_OR_RANGE>
```

Expected Outcome: Hosts that respond to the specific ICMP probe will be listed as **Host is up**. If ICMP traffic is blocked by a firewall, the host may incorrectly appear to be down.

C. TCP/UDP Ping Scan (Transport Layer)

These techniques send packets to specific ports and look for responses, making them effective for bypassing ICMP-blocking firewalls.

1. TCP SYN Ping

Nmap sends a **SYN** (Synchronize) packet to a common port (e.g., 80 or 443). A host is considered "up" if it responds with a **SYN/ACK** (port open) or **RST** (port closed).

Parameter	Description
-PS<port(s)>	TCP SYN Ping to the specified port(s).
-sn	Disable port scan.

Command Syntax:

Bash

```
nmap -sn -PS22,80,443 <TARGET_IP_OR_RANGE># Scans ports 22 (SSH), 80 (HTTP), and 443 (HTTPS)
```

2. TCP ACK Ping

Nmap sends an **ACK** (Acknowledge) packet. A host is considered "up" if it responds with an **RST** (Reset). This is excellent for mapping firewall rules.

Parameter	Description
-PA<port(s)>	TCP ACK Ping to the specified port(s).
-sn	Disable port scan.

Command Syntax:

Bash

```
nmap -sn -PA80 <TARGET_IP_OR_RANGE>
```

3. UDP Ping Scan

Nmap sends a UDP packet to a port (e.g., 53 or 40125). A host is considered "up" if it receives a reply or, more commonly, an **ICMP Port Unreachable** error (which signifies that the host is up but the port is closed).

Parameter	Description

Parameter	Description
-PU<port(s)>	UDP Ping to the specified port(s).
-sn	Disable port scan.

Command Syntax:

Bash

```
nmap -sn -PU53,161 <TARGET_IP_OR_RANGE># Scans ports 53 (DNS) and 161 (SNMP)
```

---

3. Observations and Results

Scan Type	Command Executed	TryHackMe Live Host Count	Reason for Success/Failure
ARP Scan	sudo nmap -sn -PR <Target/24>	[Record the number]	*(e.g., Successful because we were on the same subnet, bypassing firewall rules.)*
ICMP Echo	nmap -sn -PE <Target/24>	[Record the number]	*(e.g., Partially successful; some hosts may have blocked ICMP traffic.)*
TCP SYN Ping	nmap -sn -PS80,443 <Target/24>	[Record the number]	*(e.g., Highly successful as most networks keep ports 80/443 open or filtered, resulting in an RST/SYN-ACK reply.)*
UDP Ping	nmap -sn -PU53 <Target/24>	[Record the number]	*(e.g., Successful in finding hosts that returned an ICMP Port Unreachable error.)*