Scanning Networks

Introduction to Network Scanning

Network scanning is the process of systematically examining a network to identify active hosts (devices), open ports, and potential vulnerabilities. It is an essential component of network reconnaissance and assessment, and it can serve various purposes, both legitimate and malicious. In the context of ethical hacking, network scanning is used to assess the security of a network and identify weaknesses that could be exploited by potential attackers.

1 Host Discovery:

Identify devices on the network through ping sweeps or probing packets to check for responsive IP addresses.

2 Port Scanning:

Scan active hosts for open ports, revealing services actively listening for incoming connections and potential vulnerabilities.

3 Service Detection:

Examine banners or responses from open ports to identify services and applications, understanding their nature and version.

4 Vulnerability Assessment:

Perform scans for known vulnerabilities associated with identified services or software versions to identify and address potential weaknesses.

5 Operating System Detection:

Determine the operating system of target hosts, aiding in tailoring specific exploits or attacks for increased success.

6 Firewall and IDS Evasion:

Skilled attackers may use techniques to evade or bypass firewalls and intrusion detection systems for stealthy scans, emphasizing a critical consideration for both attackers and defenders.

7 Network Mapping:

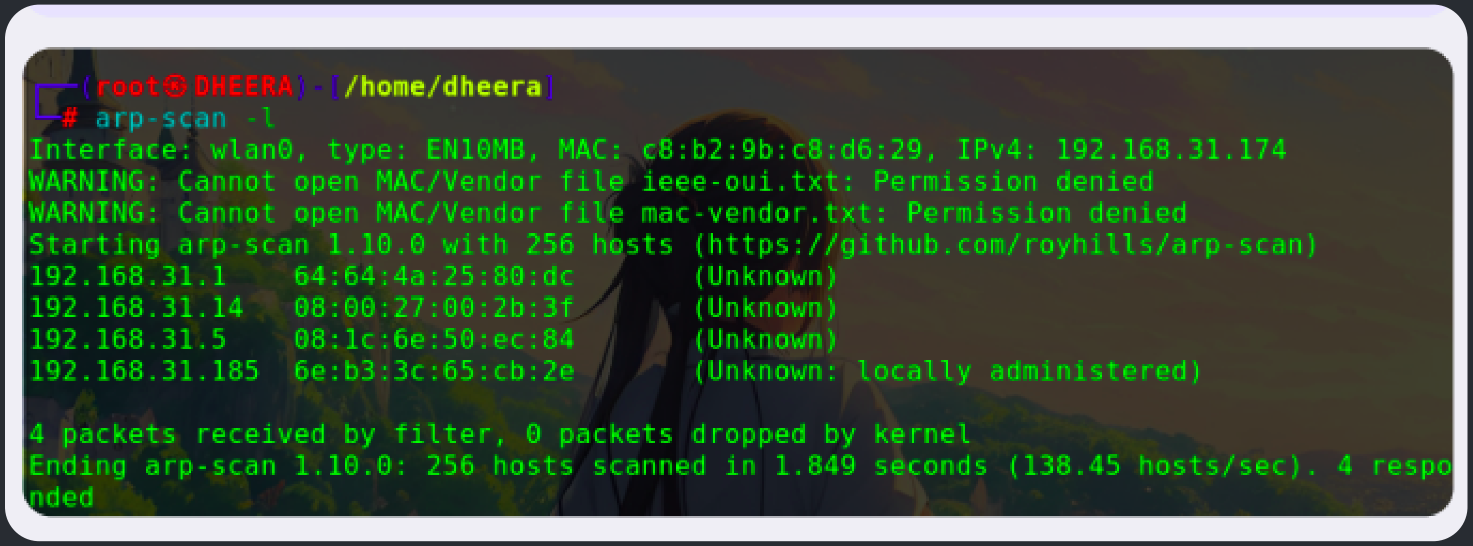
Create network maps to understand topology, identify vulnerabilities, and plan security measures based on the gathered information.

Identify Live Systems in the Network

Identifying live hosts on a network is a fundamental step in network scanning and reconnaissance. Various methods and tools can be used to determine which hosts are active or alive on a network.

ARP Scanning: Address Resolution Protocol (ARP) scanning is used in local networks. It involves sending ARP requests to the entire network subnet to discover live hosts.

`arp-scan -l`

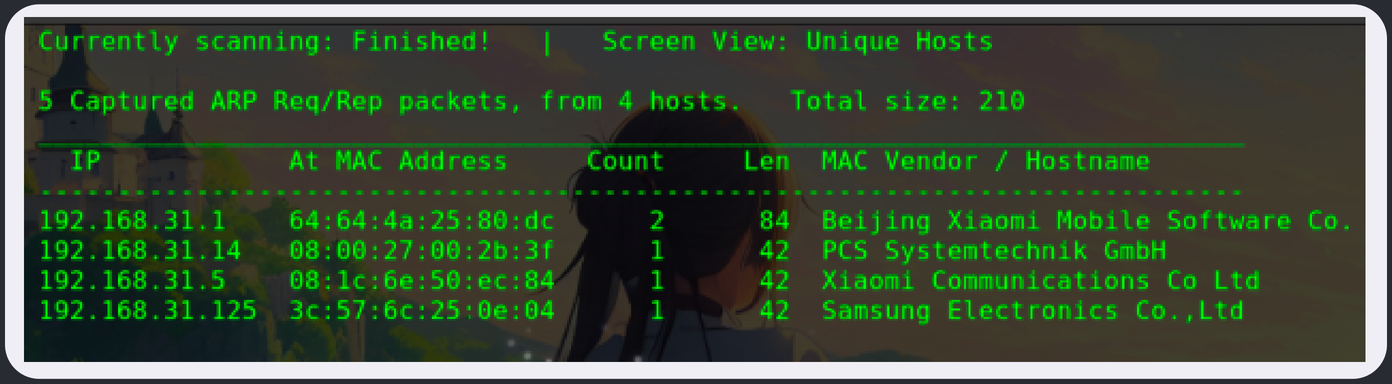


```
┌──(root㉿DHEERA)-[/home/dheera]
└─# arp-scan -l
Interface: wlan0, type: EN10MB, MAC: c8:b2:9b:c8:d6:29, IPv4: 192.168.31.174
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.31.1    64:64:4a:25:80:dc       (Unknown)
192.168.31.14   08:00:27:00:2b:3f       (Unknown)
192.168.31.5    08:1c:6e:50:ec:84       (Unknown)
192.168.31.185  6e:b3:3c:65:cb:2e       (Unknown: locally administered)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.849 seconds (138.45 hosts/sec). 4 respo
nded
```

Using Netdiscover Tool

The netdiscover tool is a network scanning and discovery tool used to identify active hosts on a local network. It can help you discover devices that are connected to your local network by sending ARP requests and analyzing the responses. The tool is often used for reconnaissance or network mapping purposes.

`netdiscover -i eth -r 192.168.31.1/24`



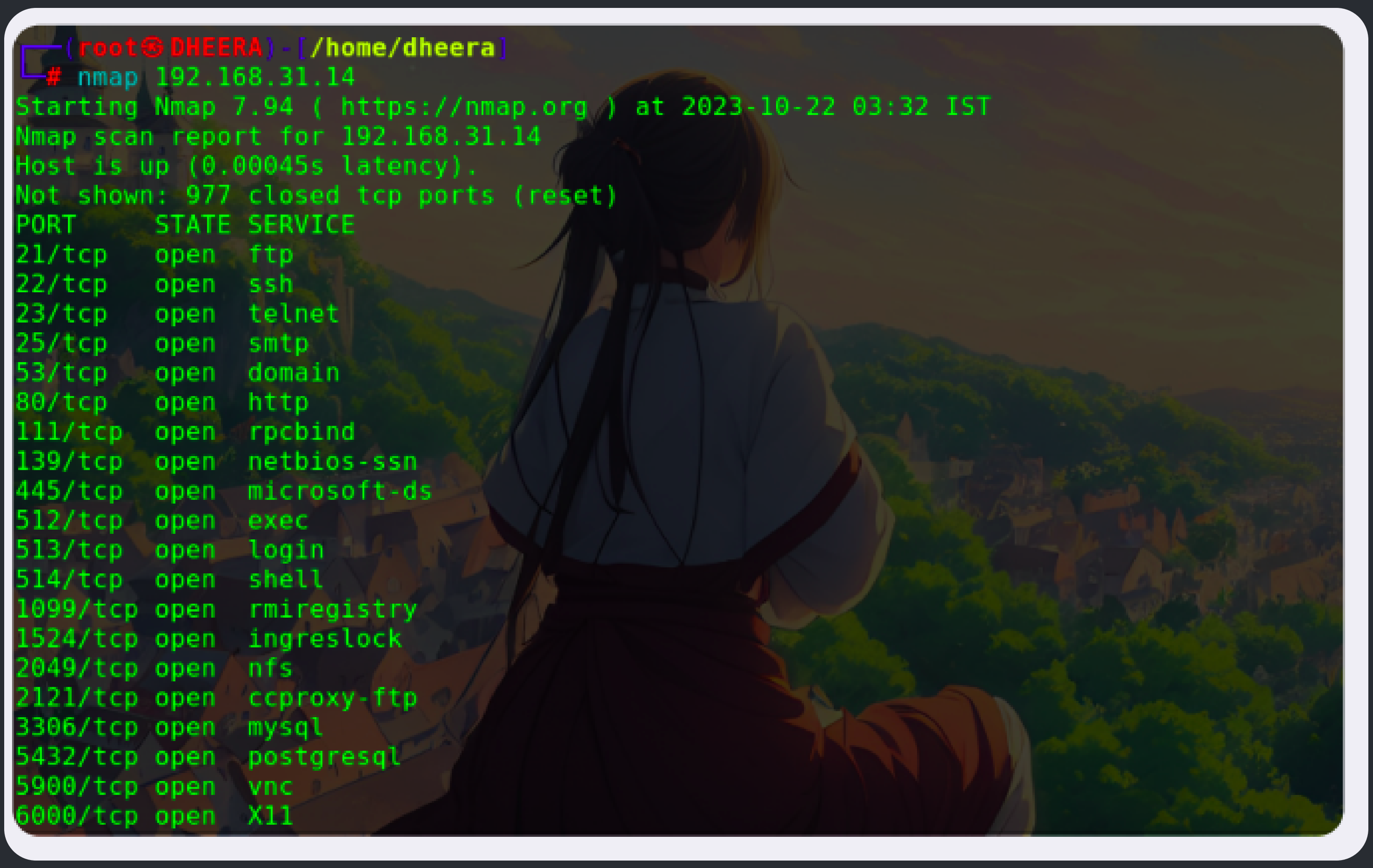
```
 Currently scanning: Finished!   |   Screen View: Unique Hosts

 5 Captured ARP Req/Rep packets, from 4 hosts.   Total size: 210
 _____________________________________________________________________________
   IP            At MAC Address     Count     Len  MAC Vendor / Hostname
 -----------------------------------------------------------------------------
 192.168.31.1    64:64:4a:25:80:dc      2      84  Beijing Xiaomi Mobile Software Co.
 192.168.31.14   08:00:27:00:2b:3f      1      42  PCS Systemtechnik GmbH
 192.168.31.5    08:1c:6e:50:ec:84      1      42  Xiaomi Communications Co Ltd
 192.168.31.125  3c:57:6c:25:0e:04      1      42  Samsung Electronics Co.,Ltd
```

Scanning using NMAP

Nmap is a free, open-source tool used for host and service discovery on computer networks.

Features:

- Host discovery: Identifies responsive hosts using TCP, ICMP requests, or open ports.
- Port scanning: Enumerates open ports on target hosts.
- OS detection: Determines operating system and hardware characteristics of network devices.
- Version detection: Interrogates network services to identify application names and versions.

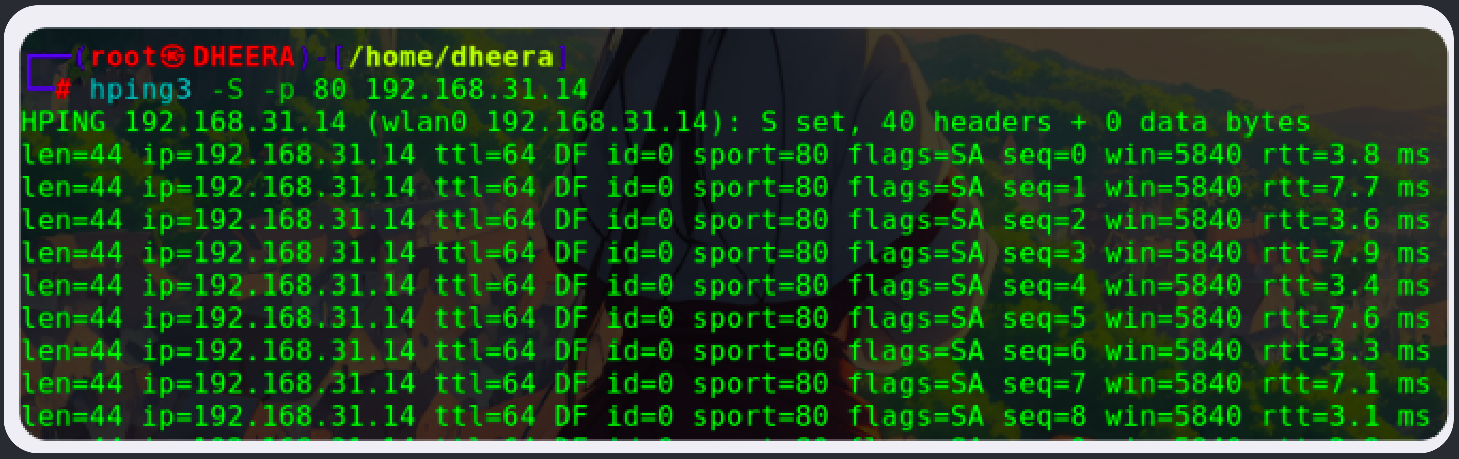


```
┌──(root㉿DHEERA)-[/home/dheera]
└─# nmap 192.168.31.14
Starting Nmap 7.94 ( https://nmap.org ) at 2023-10-22 03:32 IST
Nmap scan report for 192.168.31.14
Host is up (0.00045s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE
21/tcp   open  ftp
22/tcp   open  ssh
23/tcp   open  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
```

1. **Ping Scan:**
   - `nmap -sn [target]`
   - Performs a simple ping scan to identify live hosts without scanning for open ports.
2. **TCP SYN Scan:**
   - `nmap -sS [target]`
   - Initiates a TCP SYN scan to identify open ports. It's stealthy and faster than other scans.
3. **TCP Connect Scan:**
   - `nmap -sT [target]`
   - Completes a full TCP connection to the specified ports, useful for firewall testing.
4. **UDP Scan:**
   - `nmap -sU [target]`
   - Conducts a UDP scan to identify open UDP ports on the target.
5. **Intense Scan Plus UDP:**
   - `nmap -sS -sU -T4 -A -v [target]`
   - Combines SYN scan, UDP scan, service/version detection, and OS detection for detailed information.
6. **Service Version Detection:**
   - `nmap -sV [target]`
   - Detects service versions running on open ports.
7. **Operating System Detection:**
   - `nmap -O [target]`
   - Attempts to identify the operating system of the target.
8. **Aggressive Scan:**
   - `nmap -A [target]`
   - Performs an aggressive scan with host discovery, port scanning, service version detection, and OS detection.
9. **Fast Scan:**
   - `nmap -F [target]`
   - Conducts a fast scan by only scanning the most common 100 ports.
10. **Nmap Scripting Engine (NSE):**
    - `nmap -sC [target]`
    - Runs default scripts from the NSE for additional information.

Scanning Using Hping3

HPING3 is a command-line tool that allows users to generate TCP/IP packets and send them to a target to perform various types of scans and tests.



```
┌──(root㉿DHEERA)-[/home/dheera]
└─# hping3 -S -p 80 192.168.31.14
HPING 192.168.31.14 (wlan0 192.168.31.14): S set, 40 headers + 0 data bytes
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=0 win=5840 rtt=3.8 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=1 win=5840 rtt=7.7 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=2 win=5840 rtt=3.6 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=3 win=5840 rtt=7.9 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=4 win=5840 rtt=3.4 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=5 win=5840 rtt=7.6 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=6 win=5840 rtt=3.3 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=7 win=5840 rtt=7.1 ms
len=44 ip=192.168.31.14 ttl=64 DF id=0 sport=80 flags=SA seq=8 win=5840 rtt=3.1 ms
```

1. **TCP SYN Scan:**

```
hping3 -S -p [port] [target]
```

- Perform a TCP SYN scan to identify open ports.

1. **UDP Scan:**

```
hping3 -2 -p [port] [target]
```

- Conduct a UDP scan to identify open UDP ports.

1. **ICMP Echo Request (Ping) Scan:**

```
hping3 --icmp -c 3 [target]
```

- Use ICMP echo requests to identify live hosts.

1. **Fragmented Packets Scan:**

```
hping3 --frag -p [port] [target]
```

- Send fragmented packets to bypass some packet-filtering mechanisms.

1. **Idle Scan (Spoofed IP Scan):**

```
hping3 -I eth0 -a [spoofed_ip] -p [port] [target]
```

- Conduct an idle scan using a spoofed IP address to hide the true source.

1. **SYN Flood Attack:**

```
hping3 --flood --rand-source -S -p [port] [target]
```

- Generate a large number of SYN packets to overwhelm the target.

1. **ICMP Timestamp Scan:**

```
hping3 --icmp-ts -p [port] [target]
```

- Use ICMP timestamp requests to identify live hosts.

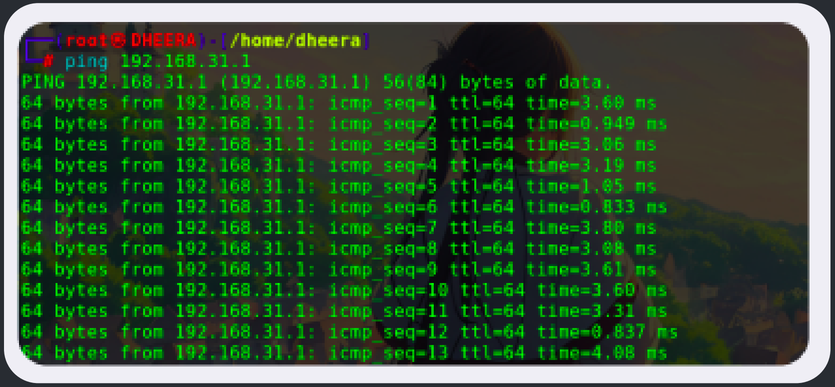
1. **Firewall Evasion (Fragmentation and Decoy IP):**

```
hping3 --frag --rand-source --ttl [ttl_value] -p [port] [target]
```

- Fragment packets and use decoy IP addresses to evade firewalls.

Operating System Identification via TTL:

Passive OS fingerprinting involves analyzing network traffic, focusing on the Time to Live (TTL) value and Window Size.



```
┌──(root㉿DHEERA)-[/home/dheera]
└─# ping 192.168.31.1
PING 192.168.31.1 (192.168.31.1) 56(84) bytes of data.
64 bytes from 192.168.31.1: icmp_seq=1 ttl=64 time=3.60 ms
64 bytes from 192.168.31.1: icmp_seq=2 ttl=64 time=0.949 ms
64 bytes from 192.168.31.1: icmp_seq=3 ttl=64 time=3.06 ms
64 bytes from 192.168.31.1: icmp_seq=4 ttl=64 time=3.19 ms
64 bytes from 192.168.31.1: icmp_seq=5 ttl=64 time=1.05 ms
64 bytes from 192.168.31.1: icmp_seq=6 ttl=64 time=0.833 ms
64 bytes from 192.168.31.1: icmp_seq=7 ttl=64 time=3.80 ms
64 bytes from 192.168.31.1: icmp_seq=8 ttl=64 time=3.08 ms
64 bytes from 192.168.31.1: icmp_seq=9 ttl=64 time=3.61 ms
64 bytes from 192.168.31.1: icmp_seq=10 ttl=64 time=3.60 ms
64 bytes from 192.168.31.1: icmp_seq=11 ttl=64 time=3.31 ms
64 bytes from 192.168.31.1: icmp_seq=12 ttl=64 time=0.837 ms
64 bytes from 192.168.31.1: icmp_seq=13 ttl=64 time=4.08 ms
```

1. **Windows:**
   - Common TTL values: 128 (older versions), 255 (recent versions).
2. **Linux/Unix:**
   - Consistent TTL value: 64 (across various distributions).
3. **Cisco Devices:**
   - Typical TTL value: 254 (routers and switches).
4. **FreeBSD:**
   - TTL value: 64 (similar to Linux and Unix).
5. **AIX (IBM):**
   - Possible TTL value: 255.
6. **Solaris (Oracle):**
   - Possible TTL value: 255.

Network Scanning Tools for Mobile There are several basic & advanced network tools:-

1. Network Scanner
2. Fing- Network Tool
3. Network Discovery Tool
4. Port Droid Too