Network Scanning

by Durgesh Thakare

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

Network scanning often begins with the identification of all devices (hosts) connected to a network. This can be done by sending probing packets or requests to IP addresses to see if they respond, a technique known as "ping sweeps."

2

Port Scanning

Once active hosts are identified, the next step is to scan for open ports on these hosts. Ports are used for communication between devices, and open ports indicate services that are actively listening for incoming connections. Port scanning helps determine which services are running and can potentially be exploited.

3

Service Detection

After finding open ports, network scanning tools can attempt to identify the services and applications running on those ports. This is typically done by examining the banner or response sent by the service, and it helps in understanding the nature of the service and its version.

4

Vulnerability Assessment

Once the network is mapped and the services are identified, ethical hackers or security professionals may perform vulnerability scanning to check if there are known vulnerabilities associated with the identified services or software versions. This helps in understanding potential weaknesses that should be addressed.

5

Operating System Detection

Some network scanning tools can attempt to determine the operating system of the target hosts. Knowing the OS can be useful in tailoring specific exploits or attacks that are more likely to succeed on that platform.

6

Firewall and IDS Evasion

Skilled attackers might employ techniques to evade or bypass firewalls and intrusion detection systems (IDS) to carry out stealthy scans. This is an important consideration for both attackers and defenders.

7

Network Mapping

Network scanning often leads to the creation of network maps, which help in understanding the network's topology, identifying vulnerabilities, and planning security measures.

TCP Flags

TCP (Transmission Control Protocol) uses various flags or control bits to manage and control the communication between two devices on a network. These flags are set in the TCP header and play a crucial role in establishing, maintaining, and terminating TCP connections. Here are some of the most commonly used TCP flags:

1. **SYN (Synchronize): The SYN flag is used to initiate a new connection. When a device wants to establish a connection with another device, it sends a TCP packet with the SYN flag set. The other device, if it's willing to establish the connection, responds with a packet that has both the SYN and ACK (acknowledge) flags set.**

2. **ACK (Acknowledge): The ACK flag is used to acknowledge the receipt of data or to confirm that a connection has been established. In a TCP three-way handshake, after the SYN flag is set by the sender and the receiving device agrees to establish the connection, it responds with a packet that has both the ACK and SYN flags set.**

3. **PSH (Push): The PSH flag is used to indicate that the data should be pushed to the receiving application immediately, without waiting for the buffer to fill up. It's often used for real-time or interactive applications.**

4. **URG (Urgent): The URG flag is used to indicate that the data in the segment is urgent. It is typically used for emergency data that should be handled with high priority.**

5. **RST (Reset): The RST flag is used to reset a connection. It's sent when a device wants to abruptly terminate a connection or if it receives a packet that doesn't make sense in the current context.**

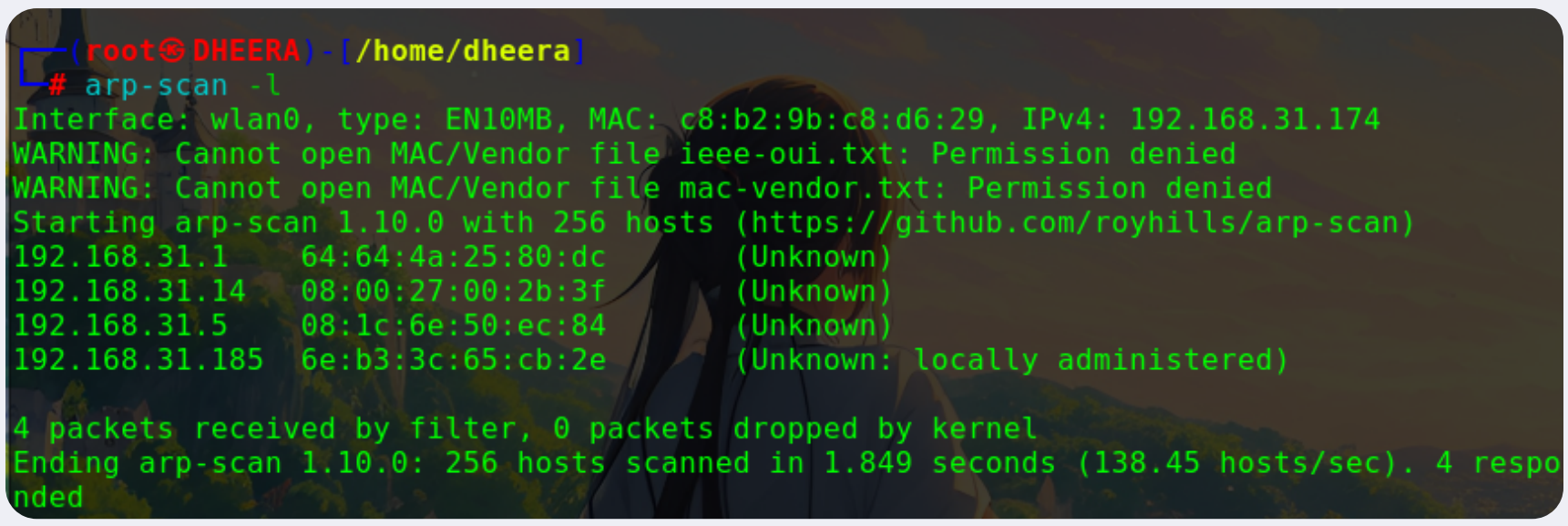
6. **FIN (Finish): The FIN flag is used to signal the end of data transmission. When a device has finished sending data, it sets the FIN flag to indicate that it's done, and the other side should prepare to close the connection.**

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.

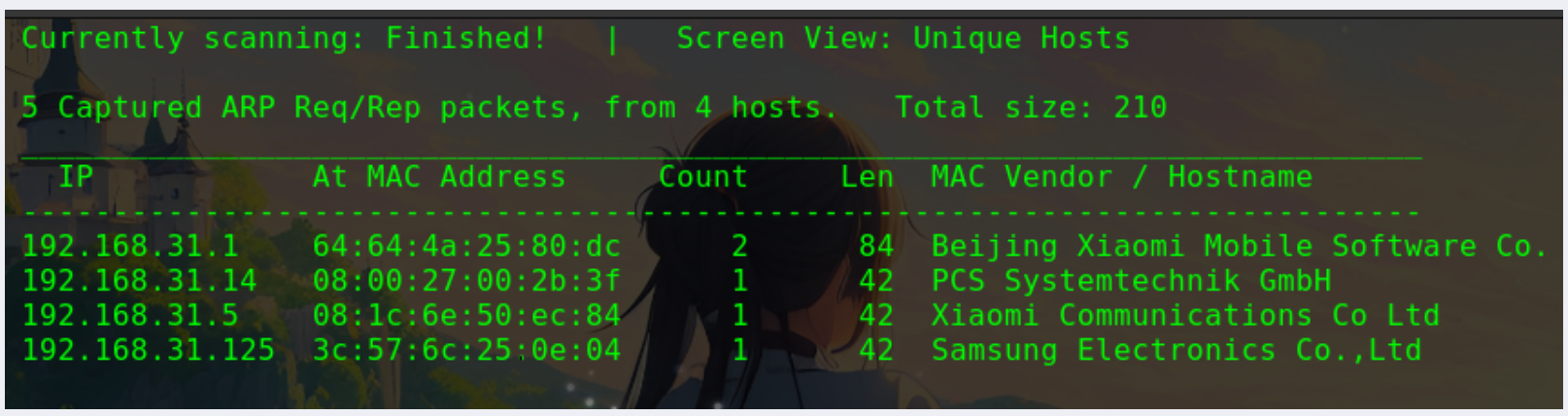
ARP Scanning: Address Resolution Protocol (ARP) scanning is used in local networks. It involves sending ARP requests to the entire network subnet to discover live hosts.

```
arp-scan -l
```



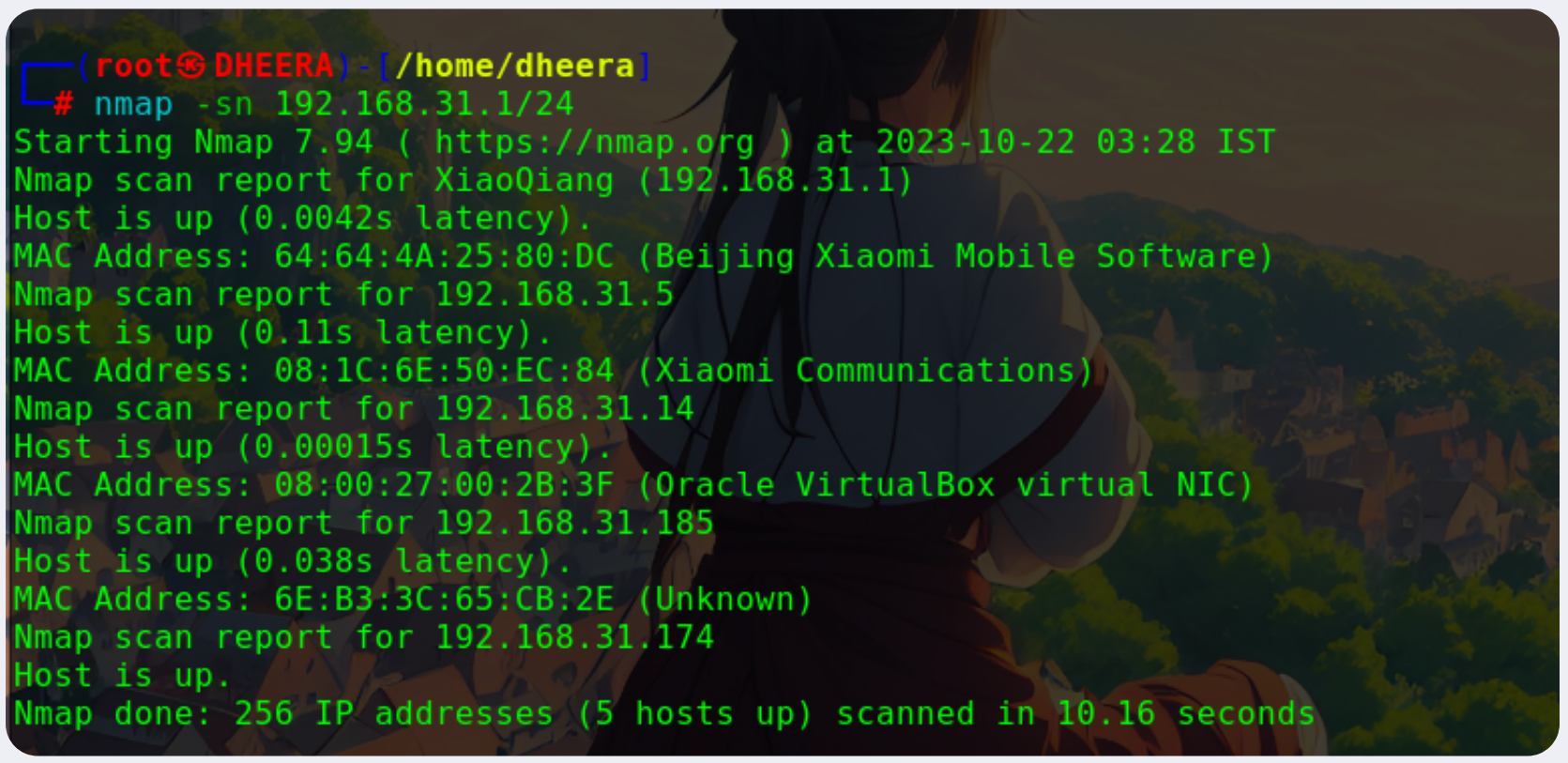
Using Netdiscover Tool : The `netdiscover` tool is a network scanning and discovery tool used to identify active hosts on a local network. It can help you discover devices that are connected to your local network by sending ARP requests and analyzing the responses. The tool is often used for reconnaissance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

```
netdiscover -i eth0 -r 192.168.1.1/16
```



Using Nmap Tool: The `nmap` tool is a versatile and powerful network scanning and discovery tool that can be used for various network reconnaissance tasks. The command you provided is using `nmap` to perform a "ping scan" on a specific IP range. Here's a breakdown of the command:

```
nmap -sn 192.168.1.1/24
```



Scanning using NMAP

Nmap is a free open source tool, employed to discover hosts and services on a computer network by sending packets and analyzing the retrieved responses. Nmap offers some features for probing computer networks, including host discovery and service and operating system detection.

- Nmap can provide further information on targets, including reverse DNS names, device types, and MAC addresses.
- Host discovery – Identifying hosts on a network. For example, listing the hosts that respond to TCP and/or ICMP requests or have a particular port open.
- Port scanning – Enumerating the open ports on target hosts.
- OS detection – Determining the operating system and hardware characteristics of network devices.
- Version detection – Interrogating network services on remote devices to determine the application name and version number.
- Scriptable interaction with the target support using the Nmap Scripting Engine (NSE).

NMAP Commands

Basic Scanning Commands

Goal	Command	Example
Scan a Single Target	nmap [target]	nmap 192.168.0.1
Scan Multiple Targets	nmap [target1, target2, etc	nmap 192.168.0.1 192.168.0.2
Scan a Range of Hosts	nmap [range of ip addresses]	nmap 192.168.0.1-10
Scan an Entire Subnet	nmap [ip address/cdir]	nmap 192.168.0.1/24
Scan Random Hosts	nmap -iR [number]	nmap -iR 0
Excluding Targets from a Scan	nmap [targets] – exclude [targets]	nmap 192.168.0.1/24 –exclude 192.168.0.100, 192.168.0.200
Excluding Targets Using a List	nmap [targets] – excludefile [list.txt]	nmap 192.168.0.1/24 – excludefile notargets.txt
Perform an Aggressive Scan	nmap -A [target]	nmap -A 192.168.0.1
Scan an IPv6 Target	nmap -6 [target]	nmap -6 1aff:3c21:47b1:0000:0000:0000:0000:2afe

Discovery Options

Goal	Command	Example
Perform a Ping Only Scan	nmap -sP [target]	nmap -sP 192.168.0.1
Don't Ping	nmap -PN [target]	nmap -PN 192.168.0.1
TCP SYN Ping	nmap -PS [target]	nmap -PS 192.168.0.1
TCP ACK Ping	nmap -PA [target]	nmap -PA 192.168.0.1
UDP Ping	nmap -PU [target]	nmap -PU 192.168.0.1
SCTP INIT Ping	nmap -PY [target]	nmap -PY 192.168.0.1
ICMP Echo Ping	nmap -PE [target]	nmap -PE 192.168.0.1
ICMP Timestamp Ping	nmap -PP [target]	nmap -PP 192.168.0.1
CMP Address Mask Ping	nmap -PM [target]	nmap -PM 192.168.0.1
IP Protocol Ping	nmap -PO [target]	nmap -PO 192.168.0.1

ARP Ping	nmap -PR [target]	nmap -PR 192.168.0.1
Traceroute	nmap –traceroute [target]	nmap –traceroute 192.168.0.1
Force Reverse DNS Resolution	nmap -R [target]	nmap -R 192.168.0.1
Disable Reverse DNS Resolution	nmap -n [target]	nmap -n 192.168.0.1
Alternative DNS Lookup	nmap –system-dns [target]	nmap –system-dns 192.168.0.1
Manually Specify DNS Server(s)	nmap –dns-servers [servers] [target]	nmap –dns-servers 201.56.212.54 192.168.0.1
Create a Host List	nmap -sL [targets]	nmap -sL 192.168.0.1/24

Advanced Scanning Options

Goal	Command	Example
TCP SYN Scan	nmap -sS [target]	nmap -sS 192.168.0.1
TCP Connect Scan	nmap -sT [target]	nmap -sT 192.168.0.1
UDP Scan	nmap -sU [target]	nmap -sU 192.168.0.1
TCP NULL Scan	nmap -sN [target]	nmap -sN 192.168.0.1
TCP FIN Scan	nmap -sF [target]	nmap -sF 192.168.0.1
Xmas Scan	nmap -sX [target]	nmap -sX 192.168.0.1
TCP ACK Scan	nmap -sA [target]	nmap -sA 192.168.0.1
Custom TCP Scan	nmap –scanflags [flags] [target]	nmap –scanflags SYNFIN 192.168.0.1
IP Protocol Scan	nmap -sO [target]	nmap -sO 192.168.0.1
Send Raw Ethernet Packets	nmap –send-eth [target]	nmap –send-eth 192.168.0.1
Send IP Packets	nmap –send-ip [target]	nmap –send-ip 192.168.0.1

Port Scanning Options

Goal	Command	Example
Perform a Fast Scan	nmap -F [target]	nmap -F 192.168.0.1
Scan Specific Ports	nmap -p [port(s)] [target]	nmap -p 21-25,80,139,8080 192.168.1.1
Scan Ports by Name	nmap -p [port name(s)] [target]	nmap -p ftp,http* 192.168.0.1
Scan Ports by Protocol	nmap -sU -sT -p U: [ports],T: [ports] [target]	nmap -sU -sT -p U:53,111,137,T:21-25,80,139,8080 192.168.0.1
Scan All Ports	nmap -p '*' [target]	nmap -p '*' 192.168.0.1
Scan Top Ports	nmap –top-ports [number] [target]	nmap –top-ports 10 192.168.0.1
Perform a Sequential Port Scan	nmap -r [target]	nmap -r 192.168.0.1

Version Detection

Goal	Command	Example
Operating System Detection	nmap -O [target]	nmap -O 192.168.0.1
Attempt to Guess an Unknown OS	nmap -O –osscan guess [target]	nmap -O –osscan-guess 192.168.0.1
Service Version Detection	nmap -sV [target]	nmap -sV 192.168.0.1
Troubleshooting Version Scans	nmap -sV –version trace [target]	nmap -sV –version-trace 192.168.0.1
Perform a RPC Scan	nmap -sR [target]	nmap -sR 192.168.0.1

Firewall Evasion Techniques

Goal	Command	Example
augment Packets	nmap -f [target]	nmap -f 192.168.0.1
pacify a Specific MTU	nmap –mtu [MTU] [target]	nmap –mtu 32 192.168.0.
Use a Decoy	nmap -D RND:[number] [target]	nmap -D RND:10 192.168.0.1
le Zombie Scan	nmap -sI [zombie] [target]	nmap -sI 192.168.0.38
Manually Specify a Source Port	nmap –source-port [port] [target]	nmap –source-port 10 192.168.0.1
Append Random Data	nmap –data-length [size] [target]	nmap –data-length 2 192.168.0.1
Randomize Target Scan Order	nmap –randomize-hosts [target]	nmap –randomize-ho 192.168.0.1-20
Spoof MAC Address	nmap –spoof-mac [MAC|0|vendor] [target]	nmap –spoof-mac Cis 192.168.0.1
Send Bad Checksums	nmap –badsum [target]	nmap –badsum 192.168.0.1

Troubleshooting And Debugging

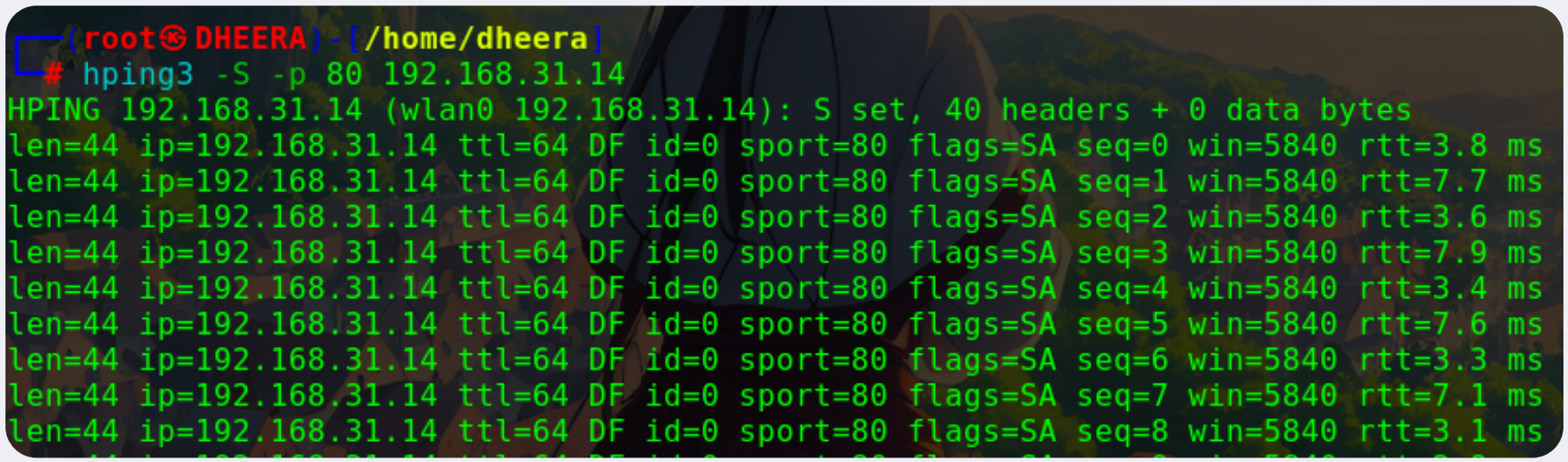
Goal	Command	Example
Getting Help	nmap -h	nmap -h
Display Nmap Version	nmap -V	nmap -V
Verbose Output	nmap -v [target]	nmap -v 192.168.0.1
Debugging	nmap -d [target]	nmap -d 192.168.0.1
Display Port State Reason	nmap –reason [target]	nmap –reason 192.168.0.1
Only Display Open Ports	nmap –open [target]	nmap –open 192.168.0.1
Trace Packets	nmap –packet-trace [target]	nmap –packet-trace 192.168.0.1
Display Host Networking	nmap –iflist	nmap –iflist
Specify a Network Interface	nmap -e [interface] [target]	nmap -e eth0 192.168.0.1

NMAP Scripting Engine

Goal	Command	Example
Execute Individual Scripts	nmap –script [script.nse] [target]	nmap –script banner.nse 192.168.0.1
Execute Multiple Scripts	nmap -script [expression] [target]	nmap –script 'http-*' 192.168.0.1
Script Categories	all, auth, default, discovery, external, intrusive, malware, safe, vuln	
Execute Scripts by Category	nmap –script [category] [target]	nmap –script 'not intrusive' 192.168.0.1
Execute Multiple Script Categories	nmap –script [category1,category2,etc]	nmap –script 'default or safe' 192.168.0.1
Troubleshoot Scripts	nmap –script [script] –script trace [target]	nmap –script banner.nse –script-trace 192.168.0.1
Update the Script Database	nmap –script-updatedb	nmap –script-updatedb

Scanning Using Hping3

hping is a powerful and flexible command-line tool used for network scanning, testing, and packet manipulation. It allows you to craft and send custom packets to network hosts, making it a valuable tool for network administrators, security professionals, and ethical hackers. hping is available on various Unix-like operating systems and is especially popular for its versatility in network testing and troubleshooting. Here are some of the common uses and features of hping:



Ping: hping can be used to send ICMP (Internet Control Message Protocol) ping packets to a target host to check if it is reachable. This can be useful for network diagnostics and monitoring.Example:

```
hping3 -c 4 -I eth0 -1 -S target_ip
```

TCP Scanning: You can use hping to perform TCP port scans by sending TCP packets to a range of ports on a target host. It can be configured to send various types of TCP packets, such as SYN, ACK, RST, and FIN, making it a versatile tool for probing network services.Example (SYN scan):

```
hping3 -S -p 80 target_ip
```

UDP Scanning: hping supports UDP scans, where you can send UDP packets to specific ports to check for open UDP services.Example:

```
hping3 -2 -p 53 target_ip
```

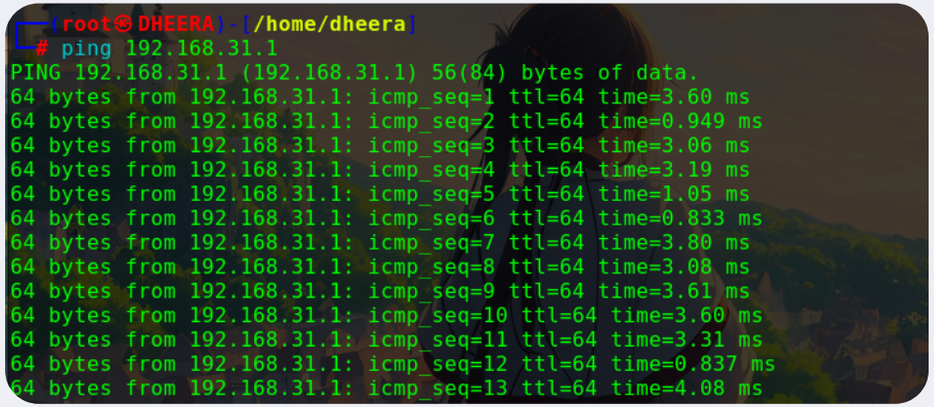
Traceroute: hping can be used to perform traceroute-like functionality by sending packets with increasing TTL (Time to Live) values to discover the path to a destination host.Example:

```
hping3 --traceroute target_ip
```

Identifying an operating system by the Time to Live

Passive OS Fingerprinting perform by the Analysing Network Traffic along with the special inspection of Time to Live (TTL) value & Window Size.

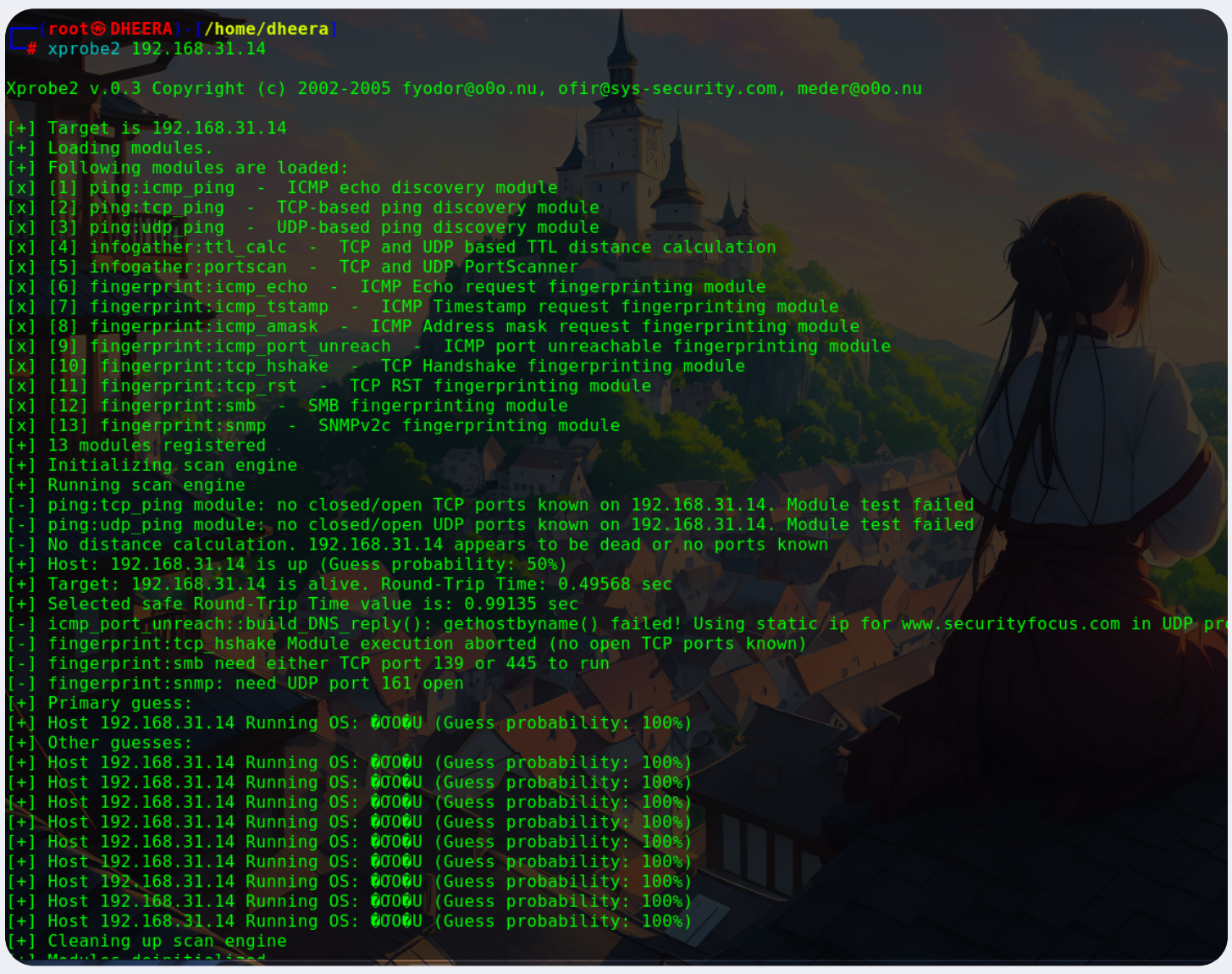
1. **Windows:** Windows-based systems often have a TTL value of 128 or 255. The TTL of 128 is commonly seen in older Windows versions, while 255 is common in more recent versions.
2. **Linux/Unix:** Linux and Unix systems tend to use a TTL of 64. This value is often consistent across various Linux distributions.
3. **Cisco Devices:** Cisco network devices, such as routers and switches, typically use a TTL of 254.
4. **FreeBSD:** FreeBSD systems often have a TTL of 64, similar to Linux and Unix.
5. **AIX (IBM):** AIX systems from IBM may use a TTL of 255.
6. **Solaris (Oracle):** Solaris systems may use a TTL of 255.



Xprobe2

Xprobe2 is an open-source, advanced network fingerprinting tool that helps identify the underlying operating system of a remote host by analyzing its responses to specially crafted packets. It is designed for network reconnaissance and is commonly used by security professionals, ethical hackers, and network administrators to gain insights into the types of systems and devices on a network.

1. **Operating System Fingerprinting:** Xprobe2 specializes in identifying the operating system of remote hosts by sending a series of carefully crafted packets and analyzing the responses. This is often referred to as "active OS fingerprinting."
2. **Lightweight and Efficient:** Xprobe2 is known for its efficiency and low impact on network traffic. It sends a minimal number of packets to perform its analysis.
3. **Active and Passive Modes:** Xprobe2 offers both active and passive fingerprinting modes. In active mode, it actively sends packets to the target host to collect responses, while in passive mode, it can analyze traffic passively without directly communicating with the target.
4. **Stealth and Evasion Techniques:** Xprobe2 can employ stealth and evasion techniques to avoid detection or interference from intrusion detection systems (IDS) or firewalls.
5. **Extensive Database:** Xprobe2 maintains an extensive database of known operating system fingerprints and characteristics. It compares the responses it receives with this database to make educated guesses about the target's OS.
6. **Customization:** Users can customize Xprobe2 to focus on specific aspects of the fingerprinting process and fine-tune its behavior according to their requirements.
7. **Scriptable:** Xprobe2 can be used as part of scripted network reconnaissance and scanning processes



Install Xprobe2 using apt manager. Use the following command.

```
sudo apt-get install xprobe2
```

Example 1: List xprobe2 Modules

```
xprobe2 -L
```

Example 2: Fingerprint with xprobe2

```
sudo xprobe2 192.169.144.130
```

Example 3: Fingerprint an Unknown System

```
sudo xprobe2 google.com
```

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 Tool