

Module 03

# Scanning Networks

# Learning Objectives

- 01** Explain Network Scanning Concepts
- 02** Demonstrate Various Scanning Techniques for Host Discovery
- 03** Demonstrate Various Scanning Techniques for Port and Service Discovery
- 04** Demonstrate Various Scanning Techniques for OS Discovery
- 05** Demonstrate Various Techniques for Scanning Beyond IDS and Firewall
- 06** Explain Network Scanning Countermeasures

Objective **01**

# Explain Network Scanning Concepts

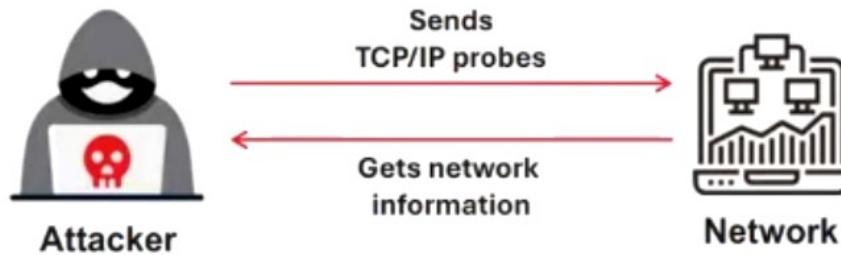
# Overview of Network Scanning

Network scanning refers to a set of procedures used for **identifying hosts, ports, and services** in a network

Network scanning is one of the **components of information gathering** which can be used by an attacker to create a profile of the target organization

Attackers use tools such as **Nmap, Hping3, Metasploit, and NetScanTools Pro** to perform network scanning

## Network Scanning Process

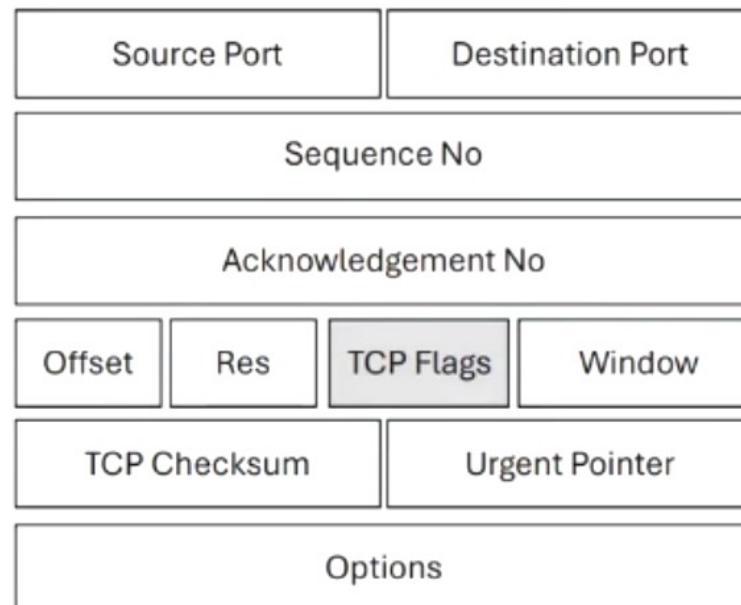


## Objectives of Network Scanning

- To discover live hosts, IP address, and open ports of live hosts
- To discover operating systems and system architecture
- To discover services running on hosts
- To discover vulnerabilities in live hosts

# TCP Communication Flags

|  |  |                                      |
|--|--|--------------------------------------|
| Data contained in the packet should be processed immediately | There will be no further transmissions | Resets a connection                  |
| <b>URG</b><br>(Urgent)                                       | <b>FIN</b><br>(Finish)                 | <b>RST</b><br>(Reset)                |
| Sends all buffered data immediately                          | Acknowledges the receipt of a packet   | Initiates a connection between hosts |

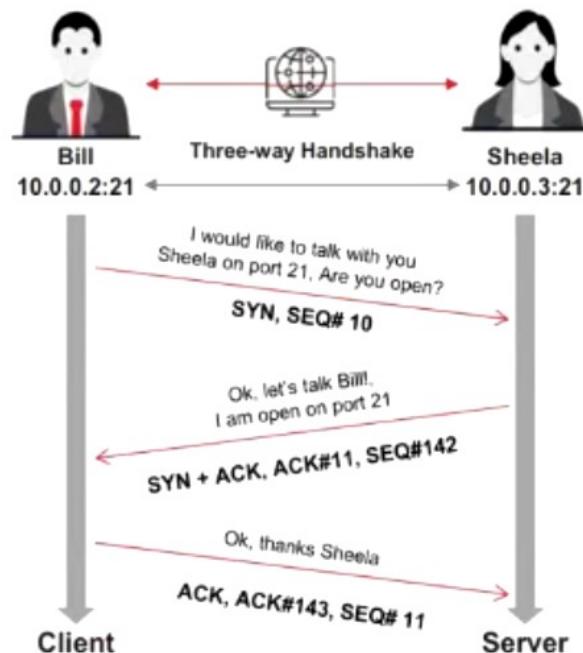


← 0-31 Bits →

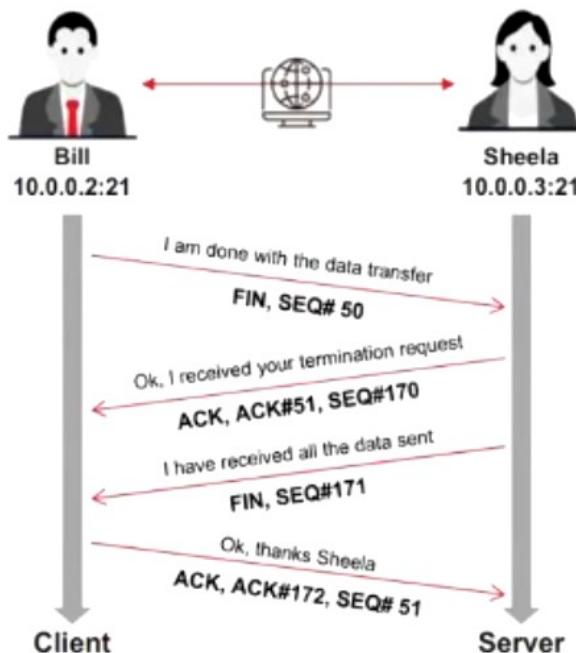
Standard TCP communications are controlled by flags in the TCP packet header

# TCP/IP Communication

TCP Session Establishment  
(Three-way Handshake)



TCP Session Termination

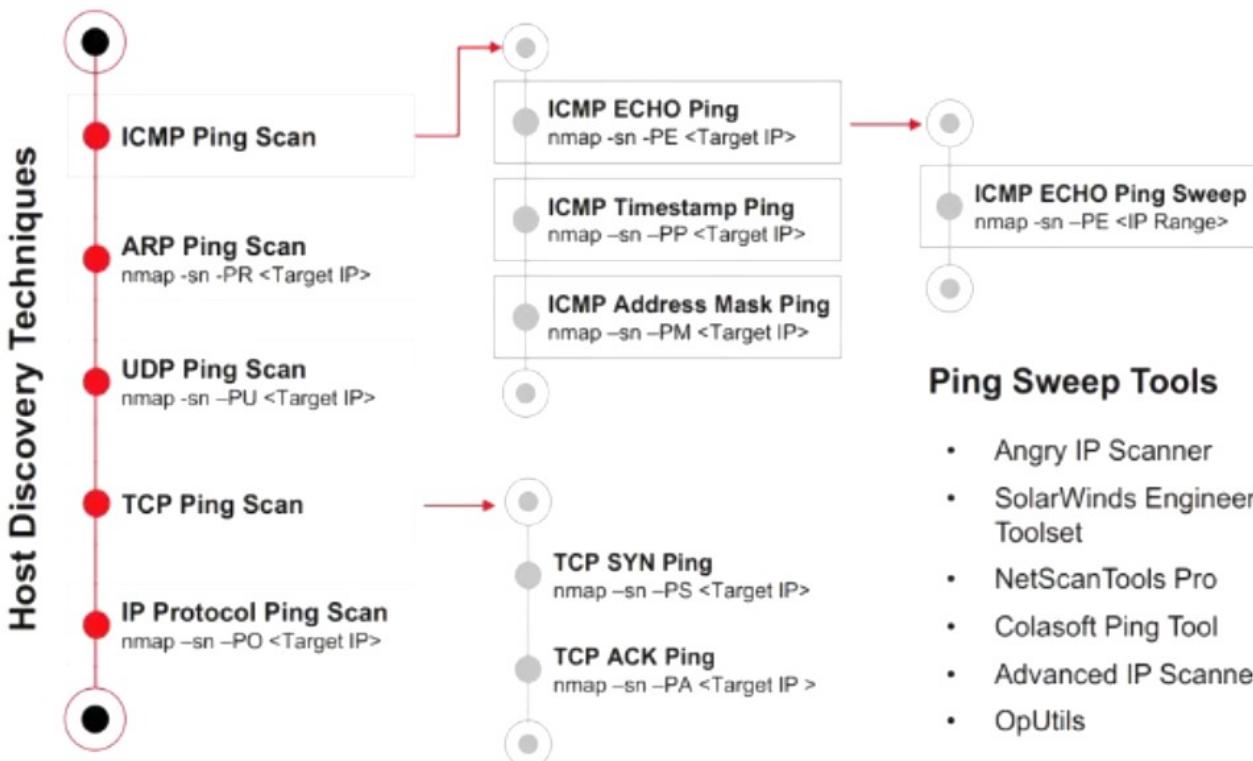


Objective **02**

# Demonstrate Various Scanning Techniques for Host Discovery

# Host Discovery Techniques

Host discovery techniques are used to **identify the active/live systems** in the network



## Ping Sweep Tools

- Angry IP Scanner
- SolarWinds Engineer's Toolset
- NetScanTools Pro
- Colasoft Ping Tool
- Advanced IP Scanner
- OpUtils

## TCP SYN Ping Scan

```

nmap -sn -PS 10.10.1.11 -Parrot Terminal
File Edit View Search Terminal Help
[nmap@parrot:~/home/attacker]
[nmap -sn -PS 10.10.1.11]
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 03:43 EDT
Nmap scan report for 10.10.1.11
HOST is up (0.00070s latency).
MAC Address: 00:15:5D:01:88:00 (Microsoft)
Nmap done: 1 IP address (1 host up) scanned in 0.15 seconds
  
```

## ICMP ECHO Ping Sweep

```

nmap -sn -PE 10.10.1.5-24 -Parrot Terminal
File Edit View Search Terminal Help
[nmap@parrot:~/home/attacker]
[nmap -sn -PE 10.10.1.5-24]
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-04 03:48 EDT
Nmap scan report for 10.10.1.9
HOST is up (0.00049s latency).
MAC Address: 02:15:5D:40:98:F1 (Unknown)
Nmap scan report for 10.10.1.11
HOST is up (0.00063s latency).
MAC Address: 00:15:5D:01:88:00 (Microsoft)
Nmap scan report for 10.10.1.14
HOST is up (0.00056s latency).
MAC Address: 02:15:5D:40:98:F2 (Unknown)
Nmap scan report for www.goodshopping.com (10.10.1.19)
HOST is up (0.00076s latency).
MAC Address: 02:15:5D:40:98:EF (Unknown)
Nmap scan report for 10.10.1.22
HOST is up (0.00041s latency).
MAC Address: 00:15:5D:01:88:02 (Microsoft)
Nmap scan report for 10.10.1.13
HOST is up
Nmap done: 20 IP addresses (16 hosts up) scanned in 1.32 seconds
  
```

## Host Discovery with AI

An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as

- "Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt"
  - "Run a fast but comprehensive Nmap scan against scan1.txt with low verbosity and write the results to scan2.txt"
  - "Use Nmap to perform ICMP ECHO ping sweep on the target network 10.10.1.0/24"

```
sgpt--chat scan --shell "Scan the target network 10.10.1.0/24 for active hosts and place only IP addresses into a file scan1.txt"
File Edit View Search Terminal Help
[+]root@parrot:[-]
└─#sgpt --chat scan --shell "Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt"
nmap -sn 10.10.1.0/24 > scan1.txt
[E]xecute, [D]escribe, [A]bort: E
[+]root@parrot:[-]
└─#sgpt --chat scan --shell "Run a fast but comprehensive Nmap scan against scan1.txt with low verbosity and write the results to scan2.txt"
nmap -T4 -O --script vuln -oN scan2.txt scan1.txt
[E]xecute, [D]escribe, [A]bort: E
[+]root@parrot:[-]
└─#sgpt --chat scan --shell "Use Nmap to perform ICMP ECHO ping sweep on the target network 10.10.1.0/24"
File Edit View Search Terminal Help
[+]root@parrot:[-]
└─#sgpt --chat scan --shell "Use Nmap to perform ICMP ECHO ping sweep on the target network 10.10.1.0/24"
nmap -sn PE 10.10.1.0/24
[E]xecute, [D]escribe, [A]bort: E
```

The image shows three terminal windows side-by-side, each displaying network scan results from Nmap.

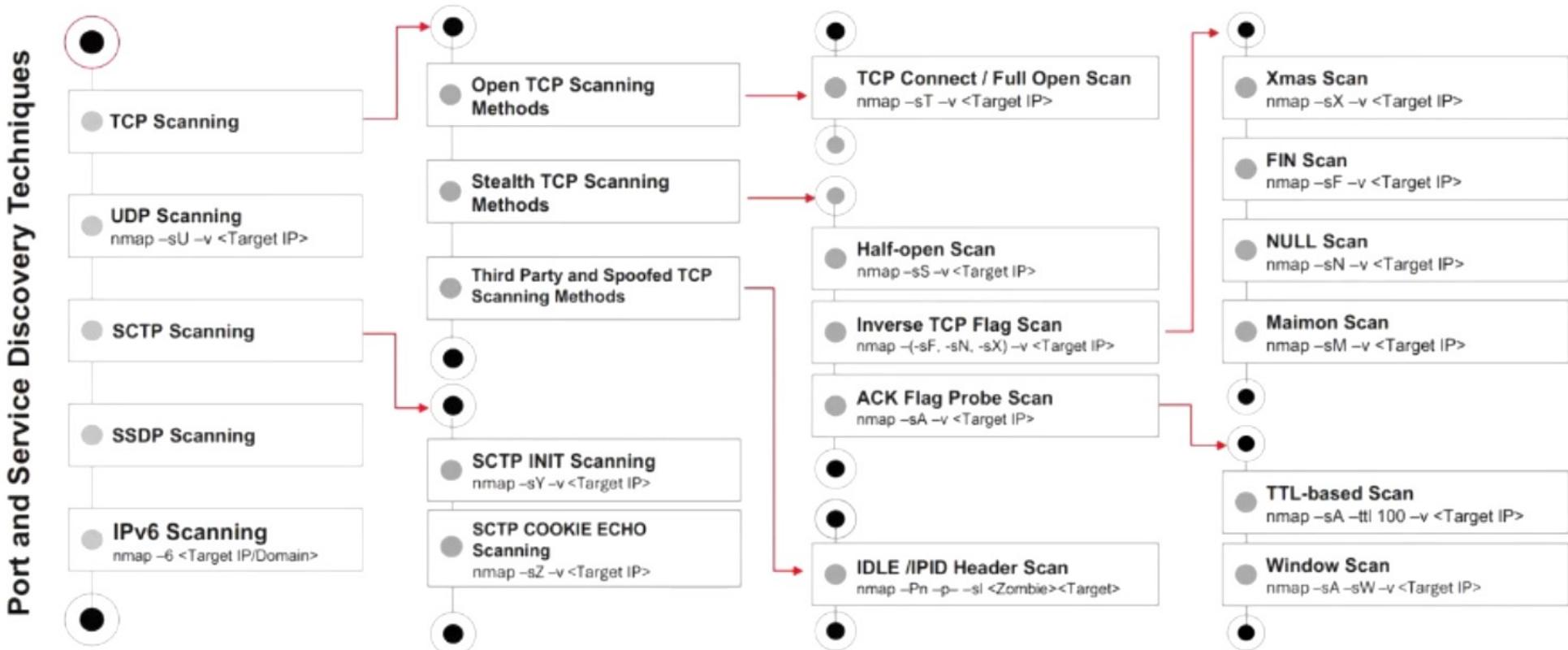
- Terminal 1:** Scan of 192.168.1.0/24. It lists hosts 192.168.1.2 through 192.168.1.13. The output includes port information and MAC addresses for each host.
- Terminal 2:** Scan of 192.168.1.1. It shows a single host with ports 21, 80, 139, 445, and 3389 open, and MAC address 00:15:50:01:00:00 (Microsoft).
- Terminal 3:** Scan of www.goodshopping.com. It shows a single host with ports 22, 80, 139, 445, and 3389 open, and MAC address 00:15:50:01:00:00 (Microsoft).

Objective **03**

# Demonstrate Various Scanning Techniques for Port and Service Discovery

# Port Scanning Techniques

The port scanning techniques are categorized according to the type of protocol used for communication



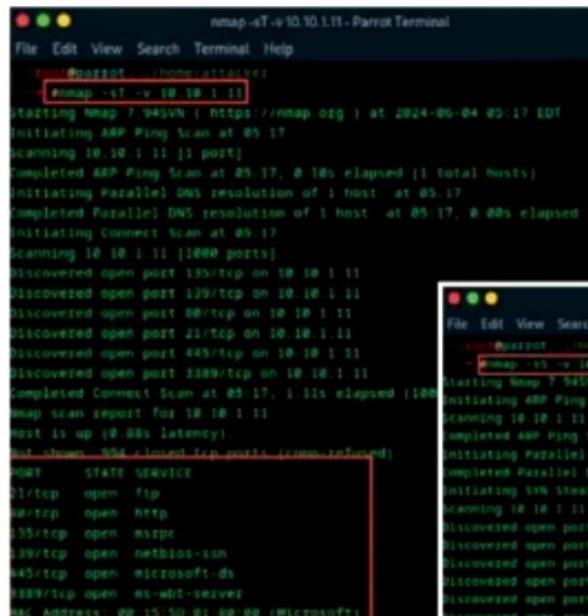
# Port Scanning Techniques (Cont'd)

## TCP Connect/ Full-Open Scan

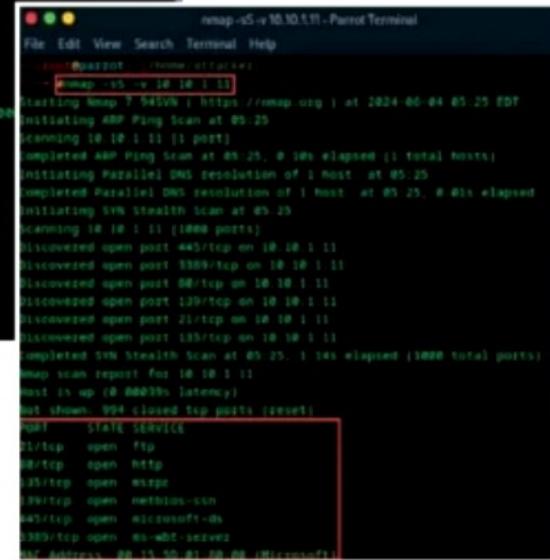
- The TCP Connect scan detects when a port is open after completing the **three-way handshake**
- TCP Connect scan **establishes a full connection** and then closes the connection by sending an **RST packet**

## Stealth Scan (Half-Open Scan)

- Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of **three-way handshake signals**, thus leaving the connection half-open
- Attackers use stealth scanning techniques to **bypass firewall rules** as well as **logging mechanisms**, and hide themselves under the appearance of regular network traffic



```
nmap -sT -v 10.10.1.11
Starting Nmap 7.94 ( https://nmap.org ) at 2024-06-04 05:17 EDT
Initiating ARP Ping Scan at 05:17
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 05:17, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host... at 05:17
Completed Parallel DNS resolution of 1 host... at 05:17, 0.00s elapsed
Initiating Connect Scan at 05:17
Scanning 10.10.1.11 [1000 ports]
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 88/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Discovered open port 443/tcp on 10.10.1.11
Discovered open port 3389/tcp on 10.10.1.11
Completed Connect Scan at 05:17, 1.11s elapsed (1000 total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
88/tcp    open  http
139/tcp   open  msrpc
3389/tcp  open  netbios-ssn
443/tcp   open  microsoft-ids
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:81:8E:8E (Microsoft)
```



```
nmap -sS -v 10.10.1.11
Starting Nmap 7.94 ( https://nmap.org ) at 2024-06-04 05:25 EDT
Initiating ARP Ping Scan at 05:25
Scanning 10.10.1.11 [1 port]
Completed ARP Ping Scan at 05:25, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host... at 05:25
Completed Parallel DNS resolution of 1 host... at 05:25, 0.00s elapsed
Initiating SYN Stealth Scan at 05:25
Scanning 10.10.1.11 [1000 ports]
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 3389/tcp on 10.10.1.11
Discovered open port 88/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Discovered open port 21/tcp on 10.10.1.11
Discovered open port 139/tcp on 10.10.1.11
Completed SYN Stealth Scan at 05:25, 1.14s elapsed (1000 total ports)
Nmap scan report for 10.10.1.11
Host is up (0.00039s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
88/tcp    open  http
139/tcp   open  msrpc
3389/tcp  open  netbios-ssn
443/tcp   open  microsoft-ids
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:81:8E:8E (Microsoft)
```

# Port Scanning with AI

An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as

- “Use Nmap to find open ports on target IP 10.10.1.11”
- “Perform stealth scan on target IP 10.10.1.11 and display the results”
- “Perform an XMAS scan on target IP 10.10.1.11”

```
[attacker@parrot] ~
→ $sgpt --chat sn --shell " Use Nmap to find open ports on target IP 10.10.1.11"
nmap 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 08:16 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00076s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3889/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 17.97 seconds
Stealth scan completed on 10.10.1.11
```

```
[root@parrot] ~
→ #sgpt --shell "Perform stealth scan on target IP 10.10.1.11 and display the results"
nmap -sS -Pn 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 06:12 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00066s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3889/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 17.97 seconds
Stealth scan completed on 10.10.1.11
```

```
[root@parrot] ~
→ #sgpt --shell "Perform an XMAS scan on target IP 10.10.1.11"
nmap -sX 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 06:22 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00044s latency).
All 1000 scanned ports on 10.10.1.11 are in ignored states.
Not shown: 1000 closed tcp ports (reset)
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.52 seconds
```

# Port Scanning with AI (Cont'd)

- “Use Nmap to scan for open ports and services against a list of IP addresses in scan1.txt and copy only the port, service and version information with the respective IP address to a new file called scan3.txt”

```
sgpt--chat scan --shell "Use Nmap to scan for open ports and services against a list of IP addresses in scan1.txt and copy only the port, service and version information with the respective IP address to a new file called scan3.txt"
File Edit View Search Terminal Help
[zoot@parrot:~]#
→ #sgpt --chat scan --shell "Use Nmap to scan for open ports and services ag
ainst a list of IP addresses in scan1.txt and copy only the port, service and v
ersion information with the respective IP address
"
nmap -sV -O -T4 scan1.txt > scan3.txt
[E]xecute, [D]escribe, [A]bort: E
[zoot@parrot:~]#
→ #
```

scan3.txt (-) - Pluma (as supervisor)

| IP Address | Port    | Service            | Version   |
|------------|---------|--------------------|---|
| 10.10.1.2  | 53/tcp  | domain             | Unbound   |
| 10.10.1.2  | 88/tcp  | http               | nginx   |
| 10.10.1.9  | 22/tcp  | ssh                | OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0) |
| 10.10.1.9  | 80/tcp  | http               | Apache httpd/2.4.52 ((Ubuntu))                                |
| 10.10.1.11 | 21/tcp  | ftp                | Microsoft ftpd  |
| 10.10.1.11 | 80/tcp  | http               | Microsoft IIS http/10.0                                       |
| 10.10.1.11 | 135/tcp | msrpc              | Microsoft Windows   |
| 10.10.1.11 | 139/tcp | netbios-ssn        | Microsoft Windows   |
| 10.10.1.11 | 445/tcp | microsoft-ds       | Microsoft Windows   |
| 10.10.1.11 | 389/tcp | ssl/ms-wbt-server? | Microsoft Windows   |

```
15 (10.10.1.19) : 445/tcp open microsoft-ds?
16 (10.10.1.19) : 1801/tcp open msmq?
17 (10.10.1.19) : 2103/tcp open msrpc Microsoft Windows RPC
18 (10.10.1.19) : 2105/tcp open msrpc Microsoft Windows RPC
19 (10.10.1.19) : 2107/tcp open msrpc Microsoft Windows RPC
20 (10.10.1.19) : 3389/tcp open ms-wbt-server Microsoft Terminal Services
21 10.10.1.22 : 53/tcp open domain Simple DNS Plus
22 10.10.1.22 : 80/tcp open http Microsoft IIS httpd/10.0
23 10.10.1.22 : 88/tcp open kerberos-sec Microsoft Windows Kerberos
(server time: 2024-03-04 09:10:12Z)
24 10.10.1.22 : 135/tcp open msrpc Microsoft Windows RPC
25 10.10.1.22 : 139/tcp open netbios-ssn Microsoft Windows netbios-ssn
26 10.10.1.22 : 389/tcp open ldap Microsoft Windows Active Directory LDAP (Domain: CEH.com\$, Site: Default-First-Site-Name)
27 10.10.1.22 : 445/tcp open microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: CEH)
28 10.10.1.22 : 464/tcp open kpasswds?
29 10.10.1.22 : 593/tcp open ncacn_http Microsoft Windows RPC over HTTP 1.0
30 10.10.1.22 : 636/tcp open tcpwrapped
31 10.10.1.22 : 1801/tcp open msmq?
```

# Service Version Discovery

- Service version detection helps attackers to obtain information about running **services and their versions** on a target system
- Obtaining an accurate service version number allows attackers to **determine the vulnerability of target system to particular exploits**
- In Zenmap, the **-sV** option is used to detect service versions



Zenmap

Scan Tools Profile Help

Target: 10.10.1.11 Profile:

Command: nmap -sV 10.10.1.11

| Hosts   | Services   |
|---------|------------|
| OS Host | 10.10.1.11 |

**Nmap Output**

nmap -sV 10.10.1.11

```

Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-11 23:51
Time: Nmap scan report for 10.10.1.11
Host is up (0.001ms latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE          VERSION
21/tcp    open  ftp              Microsoft ftplib
80/tcp    open  http             Microsoft IIS httpd 10.0
135/tcp   open  msrpc            Microsoft Windows RPC
139/tcp   open  netbios-ssn      Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    Microsoft Windows 7 - 10
microsoft-ds (workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server?
MAC Address: 00:15:5D:01:60:00 (Microsoft)
Service Info: Host: WINDOWS11; OS: Windows; CPE: cpe:/o:microsoft:windows

Service detection performed. Please report any incorrect results
at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 66.55 seconds

```

Filter Hosts

# Service Version Discovery with AI

An attacker can also leverage AI powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as

- “Use Nmap to scan open ports, MAC details, services running on open ports with their versions on target IP 10.10.1.11”



```
→ $sgpt --chat sn --shell "Use Nmap to scan open ports, MAC details, services running on open ports with their versions on target IP 10.10.1.11"
nmap -sV --reason -v -sT 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 09:20 EDT
NSE: Loaded 46 scripts for scanning.
Initiating Ping Scan at 09:20
```

```
Initiating NSE at 09:21
Completed NSE at 09:21, 0.04s elapsed
Initiating NSE at 09:21
Completed NSE at 09:21, 0.02s elapsed
Nmap scan report for 10.10.1.11
Host is up, received conn-refused (0.00047s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE          REASON  VERSION
21/tcp    open  ftp              syn-ack Microsoft ftpd
80/tcp    open  http             syn-ack Microsoft IIS httpd 10.0
135/tcp   open  msrpc            syn-ack Microsoft Windows RPC
139/tcp   open  netbios-ssn      syn-ack Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds    syn-ack Microsoft Windows 7 - 10 microsoft-ds
(workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server? syn-ack
Service Info: Host: WINDOWS11; OS: Windows; CPE: cpe:/o:microsoft:windows
```

Objective **04**

# Demonstrate Various Scanning Techniques for OS Discovery

# OS Discovery/Banner Grabbing

Banner grabbing or OS fingerprinting is the method used to **determine the operating system running on a remote target system**. There are two types of banner grabbing: active and passive

Identifying the OS used on the target host allows an attacker to **figure out the vulnerabilities possessed by the system** and the exploits that might work on a system to further **carry out additional attacks**

## Active Banner Grabbing

- **Specially crafted packets** are sent to the remote OS and the responses are noted
- The responses are then compared with a database to **determine the OS**
- Responses from different OSes vary due to differences in the **TCP/IP stack implementation**

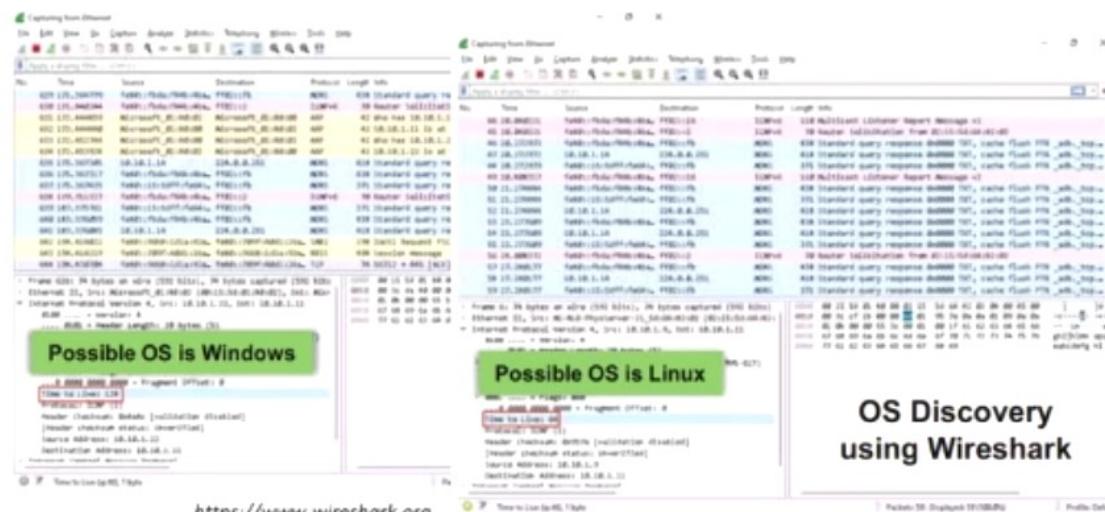
## Passive Banner Grabbing

- **Banner grabbing from error messages**  
Error messages provide information such as the type of server, type of OS, and SSL tool used by the target remote system.
- **Sniffing the network traffic**  
Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
- **Banner grabbing from page extensions**  
Looking for an extension in the URL may assist in determining the application's version.
- Example: .aspx => IIS server and Windows platform

**Note:** We will discuss passive banner grabbing in later modules.

# How to Identify Target System OS

- Attackers can identify the OS running on the target machine by looking at the **Time To Live (TTL)** and **TCP window size** in the IP header of the first packet in a TCP session
- Sniff/capture the response** generated from the target machine using packet-sniffing tools like Wireshark and observe the TTL and TCP window size fields



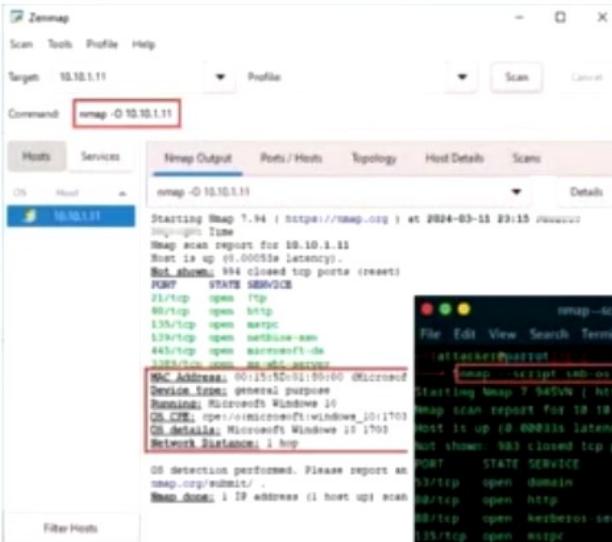
Window size values for OS

| Operating System | Time To Live | TCP Window Size            |
|------------------|--------------|----------------------------|
| Linux            | 64           | 5840                       |
| FreeBSD          | 64           | 65535                      |
| OpenBSD          | 255          | 16384                      |
| Windows          | 128          | 65,535 bytes to 1 Gigabyte |
| Cisco Routers    | 255          | 4128                       |
| Solaris          | 255          | 8760                       |
| AIX              | 255          | 16384                      |

# OS Discovery using Nmap and Nmap Script Engine

In Nmap, the **-O** option is used to perform OS discovery, providing OS details of the target machine

<https://nmap.org>



Nmap, **smb-os-discovery** is an inbuilt script that can be used for collecting OS information on the target machine through the **SMB protocol**

```
nmap -script smb-os-discovery.nse 10.10.1.22
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-12 01:48 EDT
Nmap scan report for 10.10.1.22
Host is up (0.00053s latency).
Not shown: 983 closed tcp ports (conn-refused)
PORT      STATE SERVICE
23/tcp    open  domain
80/tcp    open  http
88/tcp    open  kerberos-sec
135/tcp   open  netbios-ssn
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
464/tcp   open  kpasswd5
700/tcp   open  http-rpc-epmap
1360/tcp  open  idapsasl
1881/tcp  open  eseq
2187/tcp  open  zephyr-clt
2189/tcp  open  ekrlogin
2197/tcp  open  nmap-agent
7208/tcp  open  globalaccelDAP
7209/tcp  open  globalaccelDAPssl
3389/tcp  open  ms-wbt-server

PORT      SCRIPT RESULTS
SMB-OI-DISCOVERY
OS: Windows Server 2022 Standard 20340 (Windows Server 2022 Standard 6.3)
Computer name: Serv12822
```

# OS Discovery with AI

An attacker can also leverage AI-powered ChatGPT or other generative AI technology to perform this task by using an appropriate prompt such as

- “Use TTL to identify the operating system running on the target IP address 10.10.1.11”
- “Use TTL to identify the operating system running on the target IP address 10.10.1.9”

```
└─ $sgpt --chat sn --shell "Use TTL to identify the operating system running on the target IP address 10.10.1.11"
ping -c 1 10.10.1.11 && echo "Check the TTL value from the response to infer the OS (Linux/Unix: 64, Windows: 128)"
[E]xecute, [D]escribe, [A]bort: E
PING 10.10.1.11 (10.10.1.11) 56(84) bytes of data.
64 bytes from 10.10.1.11: icmp_seq=1 ttl=128 time=0.573 ms
--- 10.10.1.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.573/0.573/0.573/0.000 ms
Check the TTL value from the response to infer the OS (Linux/Unix: 64, Windows: 128)
```

```
[attacker@parrot]-(~)
└─ $sgpt --chat sn --shell "Use TTL to identify the operating system running on the target IP address 10.10.1.9"
ping -c 1 10.10.1.9 | grep "ttl"
[E]xecute, [D]escribe, [A]bort: E
64 bytes from 10.10.1.9: icmp_seq=1 ttl=64 time=1.83 ms
```

# OS Discovery with AI (Cont'd)

"Use Nmap script engine to perform OS discovery on the target IP addresses in scan1.txt"

```
pluma os_discovery_results.txt - Parrot Terminal
File Edit View Search Terminal Help
[zoom] [parrot] [-]
→ #nmap --script os-discovery --script-args nmap-newtargets=scan1.txt > os_discovery_results.txt
[E]xecute, [D]escribe, [A]bort: E
```

```
pluma os_discovery_results.txt - Parrot Terminal
File Edit View Search Terminal Help
Nmap scan report for 10.10.1.9
Host is up (0.00004s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
          ssh-hostkey:
          256 3b:23:12:8c:e2:d5:91:d3:e5:5a:93:82:11:b9:fb:f6 (ECDSA)
          256 ae:80:12:14:aa:cb:96:ea:ec:cb:5a:e1:3a:33:76:f4 (ED25519)
80/tcp    open  http
          http-title: Apache2 Ubuntu Default Page: It works
MAC Address: 02:15:5D:34:B1:80 (Unknown)
Device type: general purpose
Running: Linux 4.15.5.8
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.8
Network Distance: 1 hop
```

```
pluma os_discovery_results.txt - Parrot Terminal
File Edit View Search Terminal Help
host script results:
smb2-security-mode:
 3.1.1:
  Message signing enabled and required
clock-skew mean: 1h36m00s, deviation: 3h34m40s, median: 0s
smb2-time:
  date: 2024-03-04T06:03:23
  start_date: N/A
smb-os-discovery:
  OS: Windows Server 2022 Standard 20340 (Windows Server 2022 Standard 6.3)
  Computer name: Server2022
  NetBIOS computer name: SERVER2022\x00
  Domain name: CEH.com
  Forest name: CEH.com
  FQDN: Server2022.CEH.com
  System time: 2024-03-03T22:03:22-08:00
  NetBIOS_computer_name: SERVER2022
  DNS_Domain_Name: Windows11
  DNS_Computer_Name: Windows11
  Product_Version: 10.0.22000
  System_Time: 2024-03-04T06:03:20+00:00
  MAC Address: 00:15:5D:01:80:00 (Microsoft)
  Device type: general purpose
  Running_Microsoft_Windows_10
  OS CPE: cpe:/o:microsoft:windows_10:1703
  OS details: Microsoft Windows 10 1703
  Network Distance: 1 hop
```

# Create and Run Custom Script to Automate Network Scanning Tasks **with AI**

- An attacker can also leverage AI-powered ChatGPT or other generative AI technology to automate various network scanning tasks by creating custom script that will run automatically and fetch the desired result. Attacker can use appropriate prompt like as follows to create custom scripts

*"Develop a script which will automate network scanning efforts and find out live systems, open ports, running services, service versions, etc. on target IP range 10.10.1.0/24"*

```
[attacker@parrot]:~$ ssgpt --chat sn --shell " develop a script which will  
ning efforts and find out live systems, open ports, running services, etc. on target IP range 10.10.1.0/24"  
#!/bin/bash  
  
nmap -sP 10.10.1.0/24 > hosts.txt | awk '/Up$/ {print $2}' > live_hosts.txt  
nmap -IL live_hosts.txt -sV -oA scan_results &&  
cat scan_results nmap  
  
[E]xecute, [D]escribe, [A]bort: E  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15  
Nmap scan report for 10.10.1.9  
Host is up (0.00042s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE VERSION  
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; rev: 1)  
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

```
Nmap scan report for 10.10.1.11
Host is up (0.00051s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Microsoft ftpd
80/tcp    open  http         Microsoft IIS httpd
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: WORKGROUP)
3389/tcp  open  ssl/ms-wbt-server?
Service Info: Host: WINDOWS11; OS: Windows; CPE: cpe:/o:microsoft:windows_10

Nmap scan report for 10.10.1.13
Host is up (0.00036s latency).
All 1000 scanned ports on 10.10.1.13 are in ignored state
Not shown: 1000 closed tcp ports (conn-refused)

Nmap scan report for 10.10.1.14
Host is up (0.00043s latency).
Not shown: 999 closed tcp ports (conn-refused)

[REDACTED] Nmap scan report for 10.10.1.22
Host is up (0.00059s latency).
Not shown: 983 closed tcp ports (conn-refused)
PORT      STATE SERVICE      VERSION
53/tcp    open  domain       Simple DNS Plus
80/tcp    open  http         Microsoft IIS httpd 10.8
88/tcp    open  kerberos-sec Microsoft Windows Kerberos (server time: 2024-03-10 05:00:35Z)
135/tcp   open  msrpc        Microsoft Windows RPC
139/tcp   open  netbios-ssn  Microsoft Windows netbios-ssn
389/tcp   open  ldap         Microsoft Windows Active Directory LDAP (Domain: CH.com.cn, Site: Default-First-Site-Name)
445/tcp   open  microsoft-ds Microsoft Windows Server 2008 R2 - 2012 microsoft-ds (workgroup: CEN)
464/tcp   open  kpasswd5?
593/tcp   open  ncacn_http  Microsoft Windows RPC over HTTP 1.0
636/tcp   open  tcpwrapped
1801/tcp  open  msmq?
2103/tcp  open  msrpc        Microsoft Windows RPC
3195/tcp  open  msrpc        Microsoft Windows RPC
```

Objective **05**

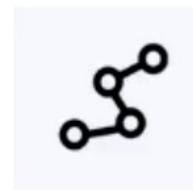
# Demonstrate Various Techniques for Scanning Beyond IDS and Firewall

# Scanning Beyond IDS and Firewall

Though firewalls and IDSs can prevent malicious traffic (packets) from entering a network, attackers can manage to **send intended packets to the target** by **evading an IDS or firewall** through the following techniques:



Packet  
Fragmentation



Source Routing



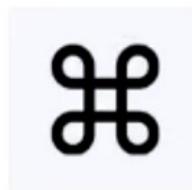
Source Port  
Manipulation



IP Address  
Decoy



IP Address  
Spoofing



MAC Address  
Spoofing



Creating  
Custom Packets



Randomizing Host  
Order and Sending  
Bad Checksums



Proxy Servers



Anonymizers

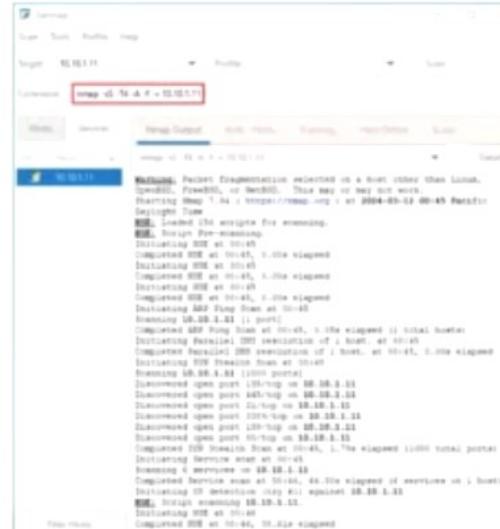
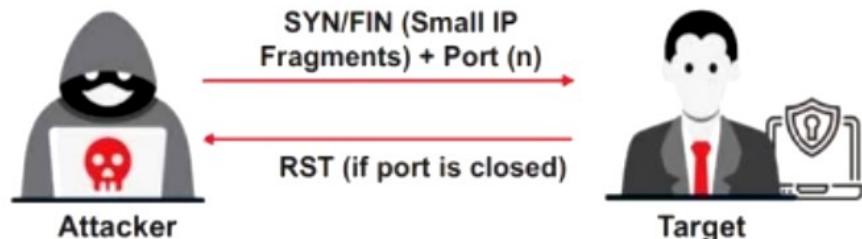
# Packet Fragmentation

Packet fragmentation refers to the **splitting of a probe packet into several smaller packets** (fragments) while sending it to a network

It is not a new scanning method but a **modification of the previous techniques**

The **TCP header** is split into several packets so that the packet filters are not able to detect what the packets are intended to do

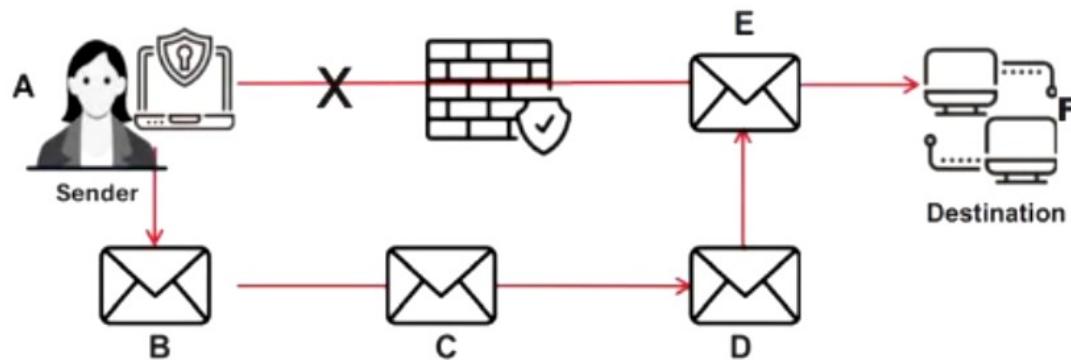
## SYN/FIN Scanning Using IP Fragments



# Source Routing

- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- Source routing refers to sending a packet to the intended destination with a partially or completely **specified route** (without firewall-/IDS-configured routers) in order to evade an IDS or firewall
- In source routing, the **attacker** makes some or all of these decisions on the router

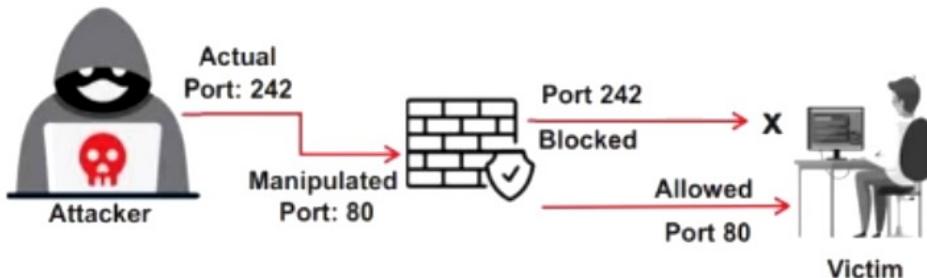
This figure shows source routing, where the originator dictates the eventual route of the traffic



# Source Port Manipulation

- Source port manipulation refers to **manipulating actual port numbers with common port numbers** in order to evade an IDS or firewall
- It occurs when a firewall is **configured to allow packets** from well-known ports like HTTP, DNS, FTP, etc.
- Nmap uses the **-g** or **--source-port** options to perform source port manipulation

**Firewall allowing manipulated Port 80 to the victim from attacker**



Zenmap

Scan Tools Profile Help

Target: 10.10.1.11 Profile:

Command: `nmap -g 80 10.10.1.11`

Hosts Services

OS Host

10.10.1.11

Nmap Output Ports / Hosts Topology Host Details Scans

`nmap -g 80 10.10.1.11`

Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-12 00:53 Pacific Daylight Time

Nmap scan report for 10.10.1.11

Host is up (0.0042s latency).

Not shown: 994 closed tcp ports (reset)

| PORT     | STATE | SERVICE       |
|----------|-------|---------------|
| 21/tcp   | open  | ftp           |
| 80/tcp   | open  | http          |
| 135/tcp  | open  | msrpc         |
| 139/tcp  | open  | netbios-ssn   |
| 445/tcp  | open  | microsoft-ds  |
| 3389/tcp | open  | ms-wbt-server |

MAC Address: 00:15:5D:01:00:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.65 seconds

<https://nmap.org>

# IP Address Decoy

- IP address decoy technique refers to **generating or manually specifying the IP addresses of decoys** in order to evade an IDS or firewall
- It appears to the target that the **decoys as well as the host(s)** are scanning the network
- This technique makes it **difficult for the IDS or firewall to determine** which IP address was actually scanning the network and which IP addresses were decoys

## Decoy Scanning using Nmap

Nmap has two options for decoy scanning:

**nmap -D RND:10 [target]**

(Generates a random number of decoys)

**nmap -D decoy1,decoy2,decoy3,... etc.**

(Manually specify the IP addresses of the decoys)

```
nmap -D RND:10 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
root@parrot:~$ nmap -D RND:10 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-08 07:13 EST
Nmap scan report for 10.10.1.11
Host is up (0.00067s latency).
Not shown: 995 filtered tcp ports (no-response)
PORT      STATE SERVICE
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 4.93 seconds
```

<https://nmap.org>

# IP Address Spoofing

- IP spoofing refers to **changing the source IP addresses** so that the attack **appears to be coming from someone else**
- When the victim replies to the address, it goes back to the **spoofed address** rather than the **attacker's real address**
- Attackers modify the **address information** in the IP packet header and the source address bits field in order to bypass the IDS or firewall



**Note:** You will not be able to complete the three-way handshake and open a successful TCP connection with spoofed IP addresses

# MAC Address Spoofing

- The MAC address spoofing technique involves spoofing a MAC address with the MAC address of a legitimate user on the network
- Attackers use the **--spoof-mac** Nmap option to set a specific MAC address for the packets to evade firewalls

```
nmap -sT -Pn --spoof-mac 0 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
└─$ nmap -sT -Pn --spoof-mac 0 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 04:50 EDT
Spoofing MAC address A1:AB:5A:04:9B:20 (No registered vendor)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.00025s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.93 seconds
```

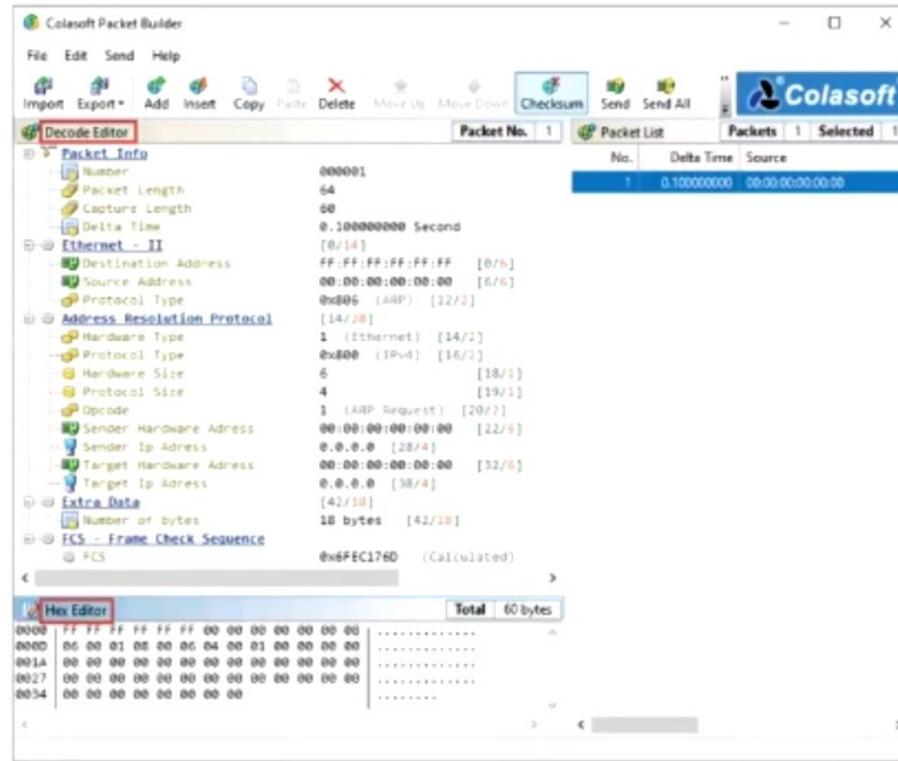
```
nmap -sT -Pn --spoof-mac Dell 10.10.1.11 - Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]~[/home/attacker]
└─$ nmap -sT -Pn --spoof-mac Dell 10.10.1.11
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-12 04:54 EDT
Spoofing MAC address 00:00:97:A2:AE:71 (Dell EMC)
You have specified some options that require raw socket access.
These options will not be honored for TCP Connect scan.
Nmap scan report for 10.10.1.11
Host is up (0.00030s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server

Nmap done: 1 IP address (1 host up) scanned in 1.74 seconds
```

# Creating Custom **Packets**

## Creating Custom Packets by using Packet Crafting Tools

- Attackers **create custom TCP packets** using various packet crafting tools like **Colasoft Packet Builder, NetScanTools Pro**, etc. to scan a target beyond a firewall



<https://www.colasoft.com>

# Randomizing Host Order and Sending Bad Checksums

## Randomizing Host Order

Attackers **scan the number of hosts** in the target network **in random order** to scan an intended target that is behind a firewall

Zenmap interface showing a scan configuration. The 'Command' field contains `nmap --randomize-hosts 10.10.1.11`. The results pane shows a single host entry for 10.10.1.11 with various open ports (21/tcp, 80/tcp, 135/tcp, 139/tcp, 445/tcp, 3389/tcp) and their corresponding service details.

```

nmap --randomize-hosts 10.10.1.11
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-12 02:45
Nmap scan report for 10.10.1.11
Host is up (0.00022s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 00:19:5D:01:00:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 1.63 seconds
  
```

## Sending Bad Checksums

Attackers send packets with bad or bogus **TCP/UDP checksums** to the intended target to avoid certain firewall rulesets

Zenmap interface showing a scan configuration. The 'Command' field contains `nmap --badsum 10.10.1.11`. The results pane shows a single host entry for 10.10.1.11 with all 1000 scanned ports listed as 'ignored states' due to bad checksums. The MAC address is also listed.

```

nmap --badsum 10.10.1.11
Starting Nmap 7.94 ( https://nmap.org ) at 2024-03-12 02:49
Nmap scan report for 10.10.1.11
Host is up (0.0010s latency).
All 1000 scanned ports on 10.10.1.11 are in ignored states.
Not shown: 1000 filtered tcp ports (no-response)
MAC Address: 00:19:5D:01:00:00 (Microsoft)

Nmap done: 1 IP address (1 host up) scanned in 23.12 seconds
  
```

<https://nmap.org>

# Proxy Servers

A proxy server is an application that can serve as an intermediary for connecting with other computers

## Why Attackers Use Proxy Servers?

- 1 To hide the actual source of a scan and **evade certain IDS/firewall restrictions**
- 2 To **mask the actual source** of an attack by impersonating the fake source address of the proxy
- 3 To **remotely access intranets** and other **website resources** that are normally restricted
- 4 To **interrupt all requests** sent by a user and transmit them to a third destination such that victims can only identify the proxy server address
- 5 To chain **multiple proxy servers** to avoid detection

**Note:** A search in Google will list thousands of **free proxy servers**

Objective **06**

# Explain Network Scanning Countermeasures

# Ping Sweep Countermeasures

- 1 Configure firewalls to **block incoming ICMP echo requests** from unknown or untrusted sources
- 2 Use **intrusion detection systems** (IDSes) and **intrusion prevention systems** (IPSes), such as **Snort** to detect and prevent ping sweep attempts
- 3 Carefully evaluate the **type of ICMP traffic** flowing through enterprise networks
- 4 **Terminate** the connection with any host sending **more than 10 ICMP ECHO requests**
- 5 Use a DMZ and allow only commands such as **ICMP ECHO\_REPLY**, **HOST UNREACHABLE**, and **TIME EXCEEDED** in the DMZ
- 6 Limit **ICMP traffic** with **access-control lists** (ACLs) to the ISP's specific IP addresses

# Port Scanning Countermeasures

- 1 Configure **firewall** and **IDS rules** to detect and block probes
- 2 Run **port scanning tools** against hosts on the network to determine whether the firewall properly **detects port scanning activity**
- 3 Ensure that the mechanisms used for **routing** and **filtering** at the routers and firewalls, respectively, **cannot be bypassed** using a particular source port or source routing methods
- 4 Ensure that the **router**, **IDS**, and **firewall firmware** are updated to their latest releases/versions
- 5 Use a **custom rule set** to lock down the network and **block unwanted ports** at the firewall
- 6 Filter all **ICMP messages** (i.e., inbound ICMP message types and outbound ICMP type 3 unreachable messages) at the **firewalls and routers**
- 7 Perform **TCP and UDP scanning** along with ICMP probes against your organization's IP address space to **check the network configuration and its available ports**
- 8 Ensure that **anti-scanning** and **anti-spoofing** rules are properly configured

# Banner Grabbing Countermeasures

## Disabling or Changing Banner

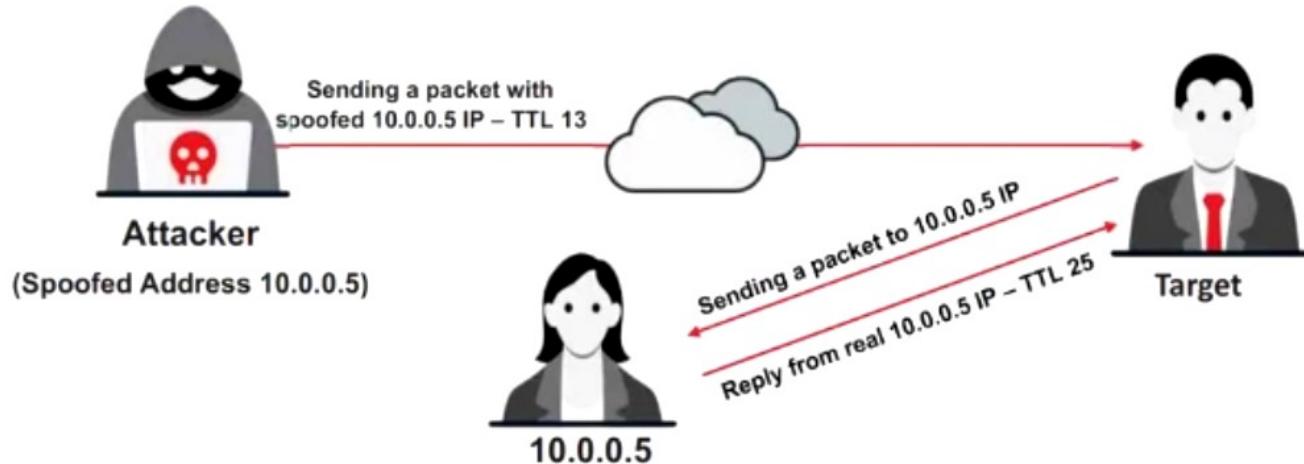
- Display **false banners** to mislead or deceive attackers
- Turn off unnecessary services on the network host to limit the disclosure of information
- Use server masking tools to disable or change banner information
- For Apache 2.x with the `mod_headers` module, use a directive in the `httpd.conf` file to change the banner information header and set the server as `New Server Name`
- Alternatively, change the `ServerSignature` line to `ServerSignature off` in the `httpd.conf` file

## Hiding File Extensions from Web Pages

- File extensions reveal information about the underlying server technology that an attacker can utilize to launch attacks
  - Hide file extensions to **mask the web technologies**
  - Replace application mappings such as `.asp` with `.htm` or `.foo`, etc. to disguise the identities of servers
  - Apache users can use `mod_negotiation` directives
- ✓ It is preferable to not use file extensions at all

# IP Spoofing Detection Techniques: Direct TTL Probes

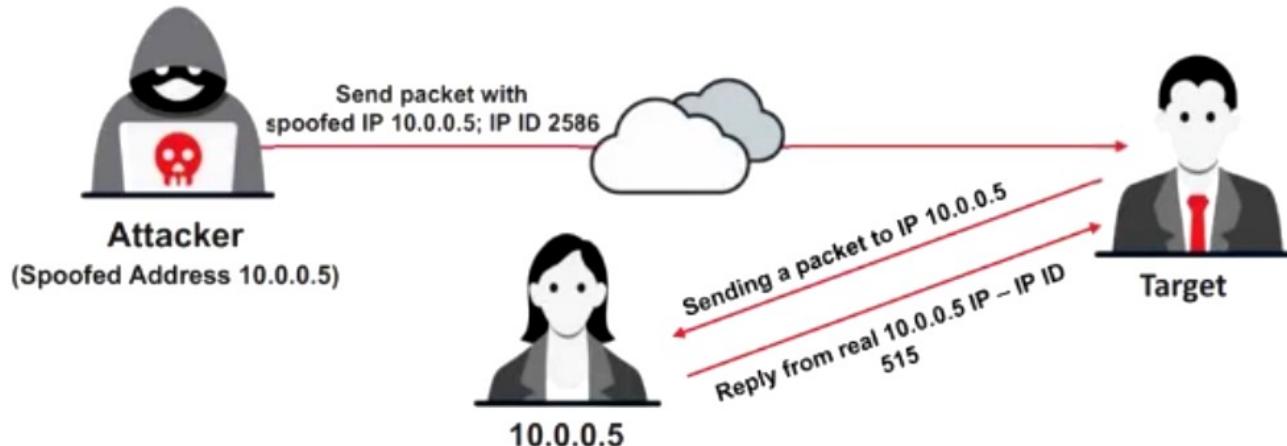
- Send a packet to the host of a suspected spoofed packet that triggers a reply and compare the TTL with that of the suspected packet; if the **TTL in the reply is not the same** as the packet being checked, this implies that it is a spoofed packet
- This technique is successful when the attacker is in a **different subnet** from that of the victim



Note: Normal traffic from one host can contrast TTLs depending on traffic patterns

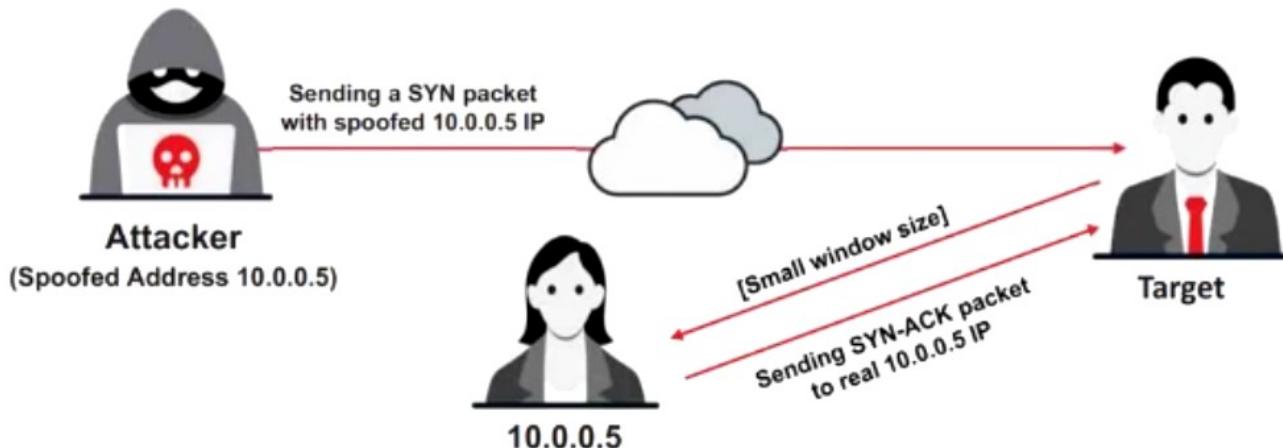
# IP Spoofing Detection Techniques: IP Identification Number

- ① Send a probe to the host of a suspected spoofed traffic that triggers a reply and **compare the IPID** with the suspected traffic
- ② If the IPIDs are **not close in value** to the packet being checked, then the suspected traffic is spoofed
- ③ This technique is considered reliable even if the attacker is in the **same subnet**



# IP Spoofing Detection Techniques: TCP Flow Control Method

- 1 Attackers sending spoofed TCP packets will not receive the target's SYN-ACK packets
- 2 Therefore, attackers cannot respond to a change in the congestion window size
- 3 When received traffic continues after a window size is exhausted, the packets are most likely spoofed



# IP Spoofing Countermeasures

- 1 Encrypt all the network traffic using cryptographic network protocols such as IPsec, TLS, SSH, and HTTPS
- 2 Use multiple firewalls to provide a multi-layered depth of protection
- 3 Do not rely on IP-based authentication
- 4 Use a random initial sequence number to prevent IP spoofing attacks based on sequence number spoofing
- 5 **Ingress Filtering:** Use routers and firewalls at your network perimeter to filter incoming packets that appear to come from an internal IP address
- 6 **Egress Filtering:** Filter all outgoing packets with an invalid local IP address as the source address

# Module Summary



- In this module, we have discussed the following:
  - How attackers discover live hosts from a range of IP addresses by sending various ping scan requests to multiple hosts
  - How attackers perform different scanning techniques to determine open ports, services, service versions, etc. on the target system
  - How attackers perform banner grabbing or OS fingerprinting to determine the operating system running on a remote target system
  - Various scanning techniques that attackers can employ to bypass IDS/firewall rules and logging mechanisms, and disguise themselves as regular network traffic
  - Network scanning countermeasures to defend against network scanning attacks
- In the next module, we will discuss in detail how attackers, as well as ethical hackers and pen-testers, perform enumeration to collect information about a target before an attack or audit