The background features a series of concentric circles in white and light gray. Arrows point from one circle to the next in a clockwise direction. Numerical values are printed along the outer edge of the circles, starting at 40 and increasing by increments of 10 up to 260. The background has a gradient from red on the left to blue on the right.

# SCANNING NETWORKS



## Module Objectives



Understanding Network Scanning Concepts

Understanding various Scanning Tools

Understanding various Host Discovery and Port Scanning Techniques

Understanding OS Discovery

Understanding various Techniques to Scan Beyond IDS and Firewall

Drawing Network Diagrams



## Module Flow

1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Host Discovery**

7

**Draw Network Diagrams**

4

**Port and Service Discovery**

5

**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

6

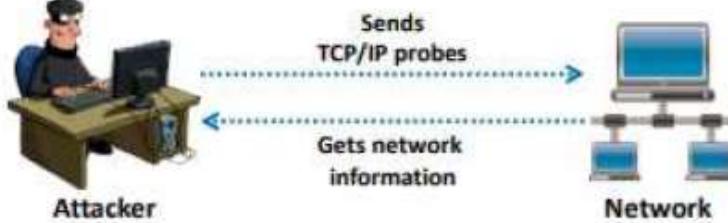
**Scanning Beyond IDS and Firewall**

# 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 intelligence gathering** which can be used by an attacker to create a profile of the target organization

**Network Scanning Process**

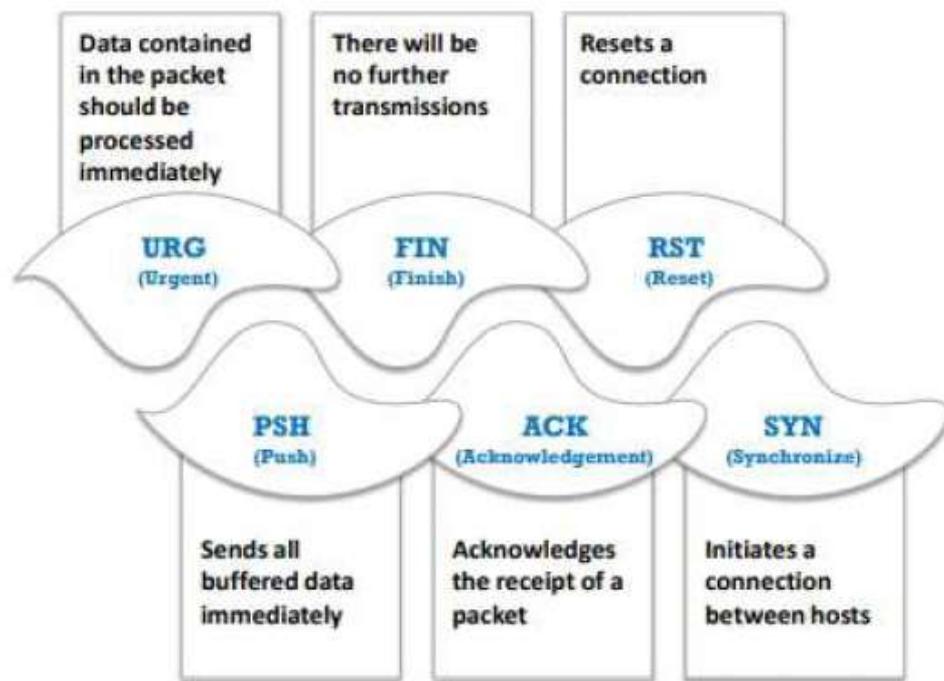


## Objectives of Network Scanning

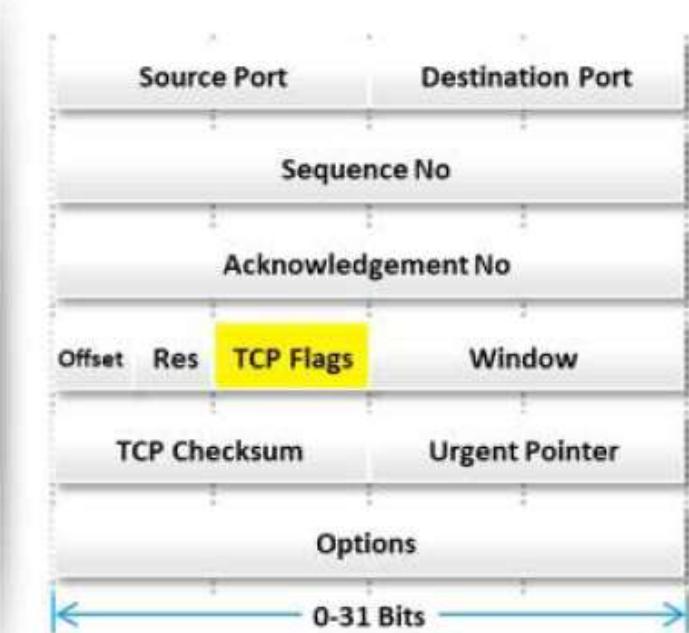
- To discover live hosts, IP address, and open ports of live hosts
- To discover operating systems and system architecture
- To discover services running on hosts
- To discover vulnerabilities in live hosts



# TCP Communication Flags



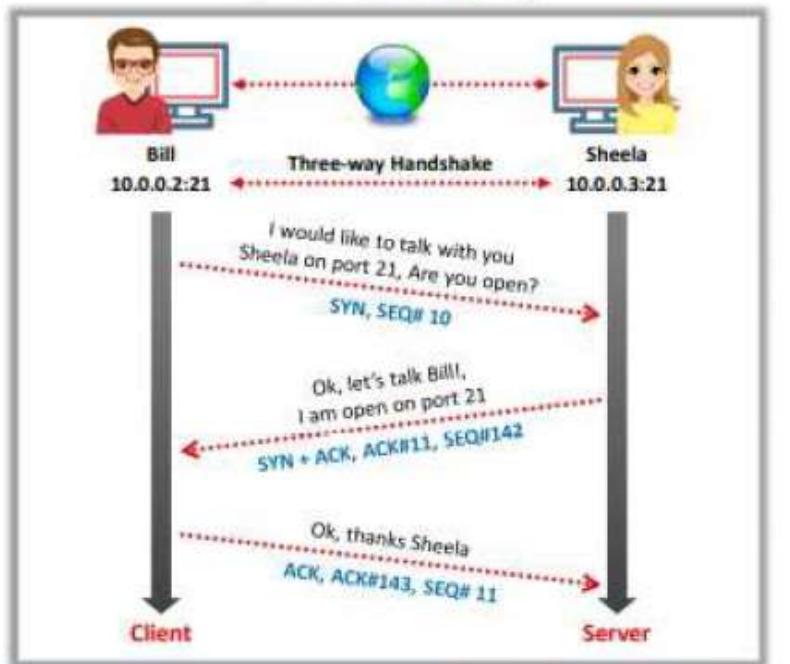
Standard TCP communications are controlled by flags in the TCP packet header



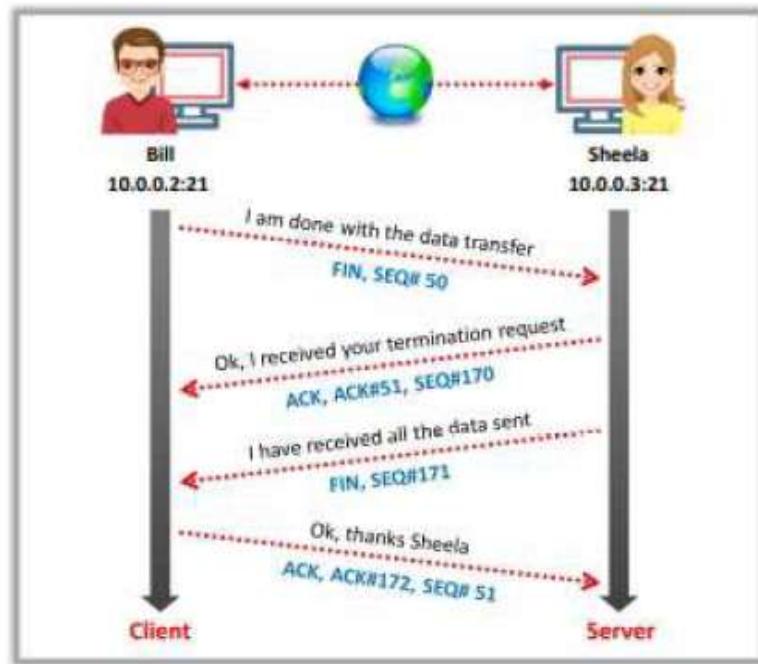
# TCP/IP Communication



## TCP Session Establishment (Three-way Handshake)



## TCP Session Termination



# Module Flow



1

Network Scanning Concepts

2

Scanning Tools

3

Host Discovery

7

Draw Network Diagrams

4

Port and Service Discovery

5

OS Discovery (Banner Grabbing/  
OS Fingerprinting)

6

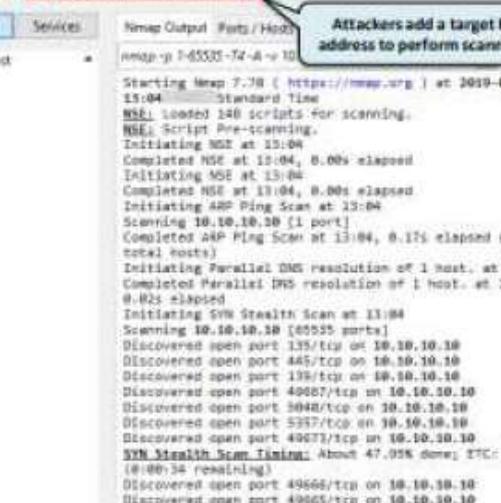
Scanning Beyond IDS and Firewall

## Scanning Tools: Nmap



- Network administrators can use Nmap for **inventorying a network**, managing service upgrade schedules, and monitoring host or service uptime
  - Attackers use Nmap to extract information such as **live hosts on the network**, **open ports**, **services** (application name and version), **types of packet filters/firewalls**, as well as **operating systems and versions used**





The screenshot shows the Zmap interface with the following details:

- Target:** 10.10.10.10
- Profile:** Interact scan, all TCP ports
- Command:** `zmap -p 1-65535 -T4 -A -v 10.10.10.10`
- Buttons:** Roots, Services, Network Output, Ports / Hosts, Scan, Cancel
- OS:** Host
- Log Output:** Shows the execution of the Zmap command, including:
  - Starting Zmap 7.08 ( https://zmap.org ) at 2019-06-07 13:04 - Standard Time
  - NSE: loaded 140 scripts for scanning.
  - NSE: Script Pre-scanning.
  - Initiating NSE at 13:04
  - Completed NSE at 13:04, 0.00s elapsed
  - Initiating NSE at 13:04
  - Completed NSE at 13:04, 0.00s elapsed
  - Initiating ARP Ping Scan at 13:04
  - Scanning 10.10.10.10 [1 port]
  - Completed ARP Ping Scan at 13:04, 0.17s elapsed (1 total hosts)
  - Initiating Parallel DNS resolution of 1 host, at 13:04
  - Completed Parallel DNS resolution of 1 host. at 13:04, 0.02s elapsed
  - Initiating SYN Stealth Scan at 13:04
  - Scanning 10.10.10.10 [65535 ports]
  - Discovered open port 139/tcp on 10.10.10.10
  - Discovered open port 445/tcp on 10.10.10.10
  - Discovered open port 1396/tcp on 10.10.10.10
  - Discovered open port 49887/tcp on 10.10.10.10
  - Discovered open port 3048/tcp on 10.10.10.10
  - Discovered open port 5357/tcp on 10.10.10.10
  - Discovered open port 49673/tcp on 10.10.10.10
  - SYN Stealth Scan Summary:** About 47.95% done; ZTC: 13:05 (4:00:34 remaining)
  - Discovered open port 49666/tcp on 10.10.10.10
  - Discovered open port 49665/tcp on 10.10.10.10
  - Discovered open port 49604/tcp on 10.10.10.10
  - Discovered open port 49668/tcp on 10.10.10.10
  - Discovered open port 49667/tcp on 10.10.10.10
  - Completed SYN Stealth Scan at 13:05, 45.60s elapsed (65535 total ports)

Attackers add a target IP address to perform scanning.

Obtains list of open ports, OS details, MAC details, and services along with their versions

**Obtains list of open ports, OS details, MAC details, and services along with their versions**

PORT	STATE	SERVICE	VERSION
139/tcp	open	msrpc	Microsoft Windows RPC
1394/tcp	open	msnetv2-ssn	Microsoft Windows ntlanman
445/tcp	open	microsoft-ds	Microsoft Windows 10 Enterprise
17263/tcp	closed	microsoft-ds	(background: 00000000)
59000/tcp	open	unknown	
59001/tls	open	https	Microsoft HTTPAPI/2.0
23/telnet	closed	telnet	
443/https	open	https	Microsoft HTTPAPI/2.0
446/https	open	https	Microsoft HTTPAPI/2.0
4462/https	open	https	Microsoft HTTPAPI/2.0
4463/https	open	https	Microsoft HTTPAPI/2.0
4464/https	open	https	Microsoft HTTPAPI/2.0
4465/https	open	https	Microsoft HTTPAPI/2.0
4466/https	open	https	Microsoft HTTPAPI/2.0
4467/https	open	https	Microsoft HTTPAPI/2.0
4468/https	open	https	Microsoft HTTPAPI/2.0
4469/https	open	https	Microsoft HTTPAPI/2.0
4470/https	open	https	Microsoft HTTPAPI/2.0
4471/https	open	https	Microsoft HTTPAPI/2.0
446-447/https	open	https	Microsoft HTTPAPI/2.0 (VMware)
4466-4471/https	open	https	Microsoft Windows Longhorn (94%
), Microsoft Windows 10 1709 (92%), Microsoft Windows 10 1903 (91%), Microsoft Windows Server 2008 SP2 (91%			
), Microsoft Windows 8.1 (90%), Microsoft Windows 10 1909 (89%), Microsoft Windows 10 1909 (88%)			

<https://nmap.org>

# Scanning Tools: Hping2/Hping3



1 Command line **network scanning** and **packet crafting** tool for the TCP/IP protocol

2 It can be used for **network security auditing**, **firewall testing**, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, etc.

## ICMP Scanning

```
File Edit View Search Terminal Help
root@parrot:~# ./hping3 -A 10.10.10.10
HPING 10.10.10.10 (eth0 10.10.10.10) ICMP mode set, 28 headers + 0 data bytes
Len=48 ip=10.10.10.10 ttl=128 id=46777 icmp seq=0 rtt=4.9 ms
Len=48 ip=10.10.10.10 ttl=128 id=46778 icmp seq=1 rtt=4.2 ms
Len=48 ip=10.10.10.10 ttl=128 id=46779 icmp seq=2 rtt=3.3 ms
Len=48 ip=10.10.10.10 ttl=128 id=46780 icmp seq=3 rtt=3.1 ms
Len=48 ip=10.10.10.10 ttl=128 id=46781 icmp seq=4 rtt=2.2 ms
Len=48 ip=10.10.10.10 ttl=128 id=46782 icmp seq=5 rtt=9.1 ms
Len=48 ip=10.10.10.10 ttl=128 id=46783 icmp seq=6 rtt=4.1 ms
Len=48 ip=10.10.10.10 ttl=128 id=46784 icmp seq=7 rtt=6.8 ms
Len=48 ip=10.10.10.10 ttl=128 id=46785 icmp seq=8 rtt=4.1 ms
...
-- 10.10.10.10 hping statistic --
9 packets transmitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 2.2/3.2/9.1 ms
root@parrot:~#
```

## ACK Scanning on port 80

```
File Edit View Search Terminal Help
root@parrot:~# ./hping3 -A 10.10.10.10 -p 80
HPING 10.10.10.10 (eth0 10.10.10.10) A set, 48 headers + 0 data bytes
Len=48 ip=10.10.10.10 ttl=128 DF id=46786 sport=80 flags=RA seq=0 win=0 rtt=7.9 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46787 sport=80 flags=RA seq=1 win=0 rtt=5.9 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46788 sport=80 flags=RA seq=2 win=0 rtt=7.7 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46789 sport=80 flags=RA seq=3 win=0 rtt=2.6 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46790 sport=80 flags=RA seq=4 win=0 rtt=3.9 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46791 sport=80 flags=RA seq=5 win=0 rtt=2.0 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46792 sport=80 flags=RA seq=6 win=0 rtt=2.2 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46793 sport=80 flags=RA seq=7 win=0 rtt=2.0 ms
Len=48 ip=10.10.10.10 ttl=128 DF id=46794 sport=80 flags=RA seq=8 win=0 rtt=2.4 ms
...
-- 10.10.10.10 hping statistic --
9 packets transmitted, 9 packets received, 0% packet loss
round-trip min/avg/max = 2.0/2.1/8.4 ms
root@parrot:~#
```

<http://www.hping.org>

# Hping Commands



ICMP Ping

```
hping3 -1 10.0.0.25
```



ACK scan on port 80

```
hping3 -A 10.0.0.25 -p 80
```



UDP scan on port 80

```
hping3 -2 10.0.0.25 -p 80
```



Collecting Initial Sequence Number

```
hping3 192.168.1.103 -Q -p 139 -s
```



Firewalls and Timestamps

```
hping3 -S 72.14.207.99 -p 80 --tcp-timestamp
```



SYN scan on port 50-60

```
hping3 -S 50-60 -S 10.0.0.25 -V
```



FIN, PUSH and URG scan on port 80

```
hping3 -F -P -U 10.0.0.25 -p 80
```



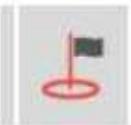
Scan entire subnet for live host

```
hping3 -1 10.0.1.x --rand-dst -I eth0
```



Intercept all traffic containing HTTP signature

```
hping3 -9 HTTP -I eth0
```



SYN flooding a victim

```
hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22  
--flood
```

Scan	Commands
ICMP ping	<code>hping3 -1 10.0.0.25</code>
ACK scan on port 80	<code>hping3 -A 10.0.0.25 -p 80</code>
UDP scan on port 80	<code>hping3 -2 10.0.0.25 -p 80</code>
Collecting initial sequence number	<code>hping3 192.168.1.103 -Q -p 139 -s</code>
Firewalls and timestamps	<code>hping3 -S 72.14.207.99 -p 80 --tcp-timestamp</code>
SYN scan on port 50-60	<code>hping3 -8 50-56 -S 10.0.0.25 -v</code>
FIN, PUSH, and URG scan on port 80	<code>hping3 -F -P -U 10.0.0.25 -p 80</code>
Scan entire subnet for live host	<code>hping3 -1 10.0.1.x --rand-dest -I eth0</code>
Intercept all traffic containing HTTP signature	<code>hping3 -9 HTTP -I eth0</code>
SYN flooding a victim	<code>hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22 --flood</code>



# Scanning Tools

## Metasploit

Metasploit is an open-source project that provides the infrastructure, content, and tools to **perform penetration tests and extensive security auditing**

A screenshot of a Mac OS X desktop showing a Terminal window. The terminal shows command-line output related to Metasploit, including:

```
# msfconsole -v 5.0.18-dev
[...]
[*] Loaded: 45, 0.18 dev
[*]  +-- SBM payloads - 3862 auxiliary - 328 post
[*]  +-- SAW payloads - 44 encoders - 39 maps
[*]  +-- 2 exploit
[*]  +-- 0 status
[*]  +-- 1 Connected to host, connection type: portforward
[*]  +-- 0.25 search partitions

Available Modules:
Name          Disclosure Date Rank Check Description
[...]
metasploit/scanner/multi/http/webscan_github_scanner      normal  Yes  Metasploit Engineless scanner
metasploit/scanner/multi/http/webscan_jenkins           normal  Yes  Metasploit External Port Scanner
metasploit/scanner/multi/tcp/ack_flood_all               normal  Yes  TCP ACK Floodall Scanner
metasploit/scanner/multi/tcp/banana_port_scanner        normal  Yes  TCP Banana Port Scanner
metasploit/scanner/multi/tcp/port_scanner                normal  Yes  TCP SYN Port Scanner
metasploit/scanner/multi/tcp/puix_scanner                 normal  Yes  TCP "Puix" Port Scanner
metasploit/scanner/multi/tcp/salt_router_port_scanner    normal  No   SaltRouter Port Scanner
[...]
```

The URL <https://www.metasploit.com> is visible at the bottom of the terminal window.

## NetScanTools Pro

NetScanTools Pro assists attackers in automatically or manually listing **IPv4/IPv6 addresses, hostnames, domain names, and URLs**



**Other Scanning Tools:**

**UnicornsScan**  
<https://sourceforge.net>

**SolarWinds Port Scanner**  
<https://www.solarwinds.com>

**PRTG Network Monitor**  
<https://www.paessler.com>

**OmniPeek Network Protocol Analyzer**  
<https://www.hotspotshield.com>

# Scanning Tools for Mobile



## IP Scanner



<https://10base-t.com>

## Fing



<https://www.fing.io>

## Network Scanner



<https://play.google.com>

# Module Flow



1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Host Discovery**

7

**Draw Network Diagrams**

4

**Port and Service Discovery**

5

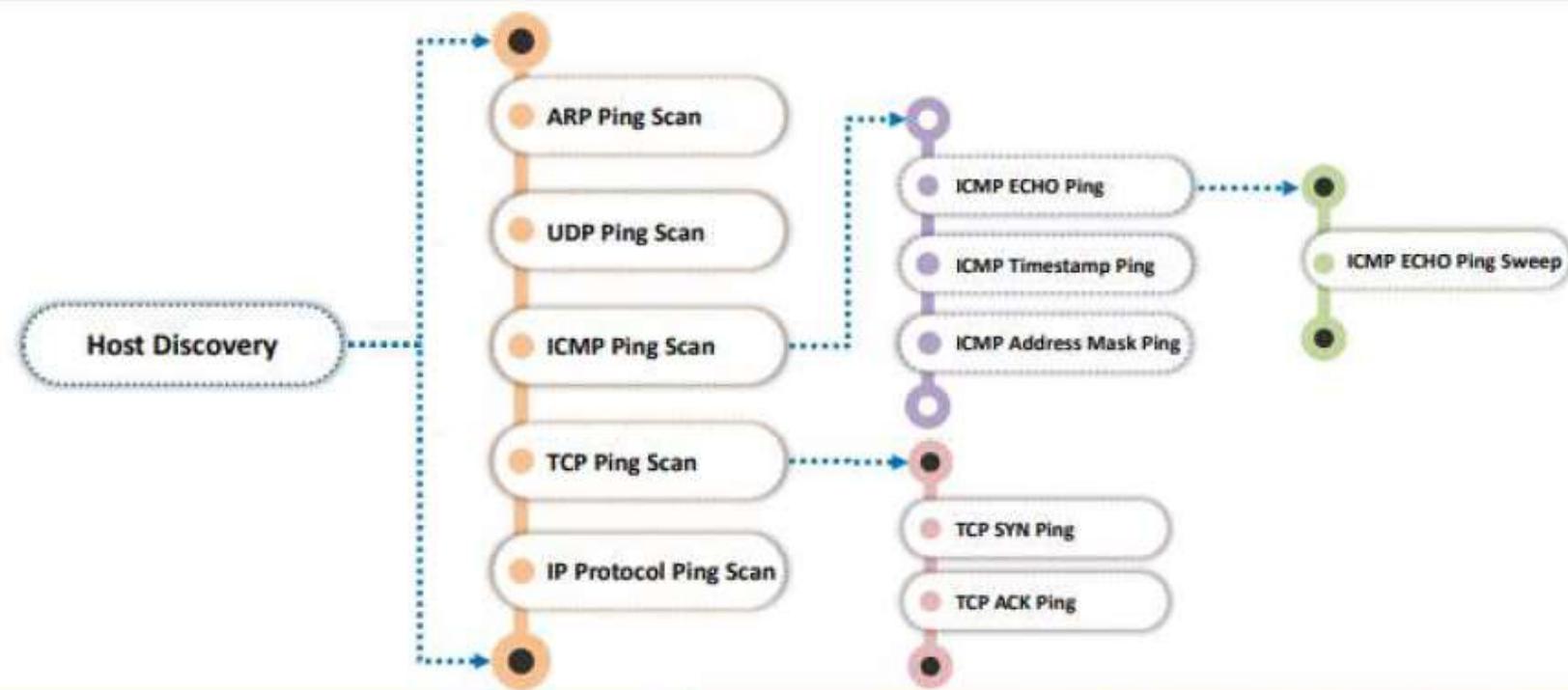
**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

6

**Scanning Beyond IDS and Firewall**

# Host Discovery Techniques

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

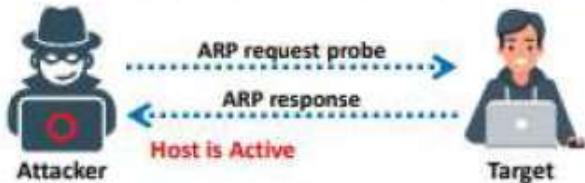


# ARP Ping Scan and UDP Ping Scan



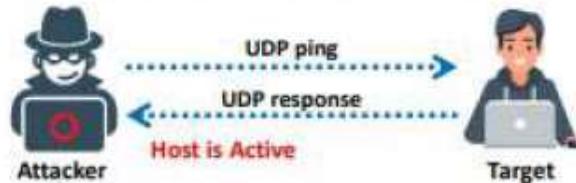
## ARP Ping Scan

- Attackers send **ARP request probes** to target hosts, and an **ARP response** indicates that the **host is active**



## UDP Ping Scan

- Attackers send **UDP packets** to target hosts, and a **UDP response** indicates that the **host is active**



Zenmap interface showing an ARP Ping Scan result for target 10.10.10.10. The command used was "nmap -sn -PR 10.10.10.10". The output shows the host is up with 0.00s latency and MAC address 00:0C:29:79:02:B9 (VMware).

OS	Host	IP	MAC Address	Latency
OS	Host	10.10.10.10	00:0C:29:79:02:B9 (VMware)	0.00s

Zenmap interface showing a UDP Ping Scan result for target 10.10.10.10. The command used was "nmap -sn -PU 10.10.10.10". The output shows the host is up with 0.00s latency and MAC address 00:0C:29:79:02:B9 (VMware).

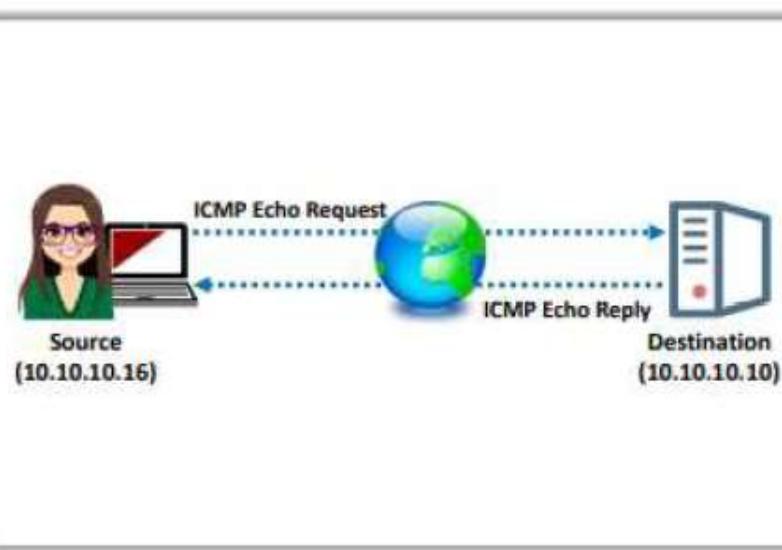
OS	Host	IP	MAC Address	Latency
OS	Host	10.10.10.10	00:0C:29:79:02:B9 (VMware)	0.00s

<https://nmap.org>

# ICMP ECHO Ping Scan



- ICMP ECHO ping scans involve sending **ICMP ECHO requests** to a host. If the host is live, it will return an ICMP ECHO reply
- This scan is useful for **locating active devices** or determining if the **ICMP is passing through a firewall**



ICMP Echo ping scan output using Zenmap

The screenshot shows the Zenmap interface with the target set to '10.10.10.10'. The command entered is 'nmap -sn -PE 10.10.10.10'. The 'Hosts' tab is selected, showing one host entry for '10.10.10.10' which is marked as 'Up'. The 'Details' pane displays the following text:

```
nmap -sn -PE 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-07
12:33   Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.015s latency).
MAC Address: 00:0C:29:79:02:B0 (VMware)
Nmap done: 1 IP address (1 host up) scanned in 0.25 seconds
```

<https://nmap.org>

# ICMP ECHO Ping Sweep

- Ping sweep is used to determine the **live hosts from a range of IP addresses** by sending ICMP ECHO requests to multiple hosts. If a host is alive, it will return an ICMP ECHO reply
- Attackers calculate subnet masks by using a **Subnet Mask Calculator** to identify the number of hosts that are present in the subnet
- Attackers subsequently use a ping sweep to create an **inventory of live systems** in the subnet



The screenshot shows the Zenmap interface with the command `nmap -sn -PE 10.10.10.5-15` entered in the Command field. The output window displays the results of the ping sweep:

```
Starting Nmap 7.00 ( https://nmap.org ) at 2019-06-14  
15:53 India Standard Time  
Nmap scan report for 10.10.10.5  
Host is up (0.000s latency).  
MAC Address: 00:0C:29:80:99:5A (VMware)  
Nmap scan report for 10.10.10.9  
Host is up (0.0000s latency).  
MAC Address: 00:0C:29:80:99:5A (VMware)  
Nmap scan report for 10.10.10.10  
Host is up (0.0000s latency).  
MAC Address: 00:0C:29:79:92:89 (VMware)  
Nmap scan report for 10.10.10.11  
Host is up (0.0000s latency).  
MAC Address: 00:0C:29:79:92:89 (VMware)  
Nmap done: 11 IP addresses (4 hosts up) scanned in  
0.73 seconds
```

The diagram illustrates the ICMP Echo Request/Reply process. A source host at IP 10.10.10.16 sends ICMP Echo Requests to five target hosts (10.10.10.6, 10.10.10.9, 10.10.10.12, 10.10.10.11, 10.10.10.10). The hosts 10.10.10.6, 10.10.10.9, 10.10.10.12, and 10.10.10.11 respond with ICMP Echo Replies, while the host at 10.10.10.10 does not.



## Ping Sweep Tools

### Angry IP Scanner

Angry IP Scanner pings each IP address to check if any of these addresses are live. Then, it optionally resolves hostnames, determines the MAC address, scans ports, etc.

### Ping Sweep Tools

- SolarWinds Engineer's Toolset (<https://www.solarwinds.com>)
- NetScanTools Pro (<https://www.netscantools.com>)
- Colasoft Ping Tool (<https://www.colasoft.com>)
- Visual Ping Tester (<http://www.pingtester.net>)
- OpUtils (<https://www.manageengine.com>)

IP Range - Angry IP Scanner				
Scan Go to Commands Favorites Tools Help				
IP Range: 10.10.10.0 to 10.10.10.255		IP Range		...
Hostname:	Server2016	IP:	Netmask	Start
IP	Ping	Hostname	Ports (1000+)	
10.10.10.10	0 ms	DESKTOP-SV6DCV1	1,7,9,13,17,19,21-23,25,42,53,80-83,91,98,...	
10.10.10.12	0 ms	WIN-OJAQ7QJBPAI	53,80,88,135,139,389,445,464,593,636	
10.10.10.16	0 ms	Server2016	80,135,139,445	
10.10.10.8	0 ms	VICTIM-B	135,139,445	
10.10.10.9	0 ms	jason-Virtual-Machine	80	
10.10.10.11	0 ms	[n/a]	80	

<https://www.angryip.org>

## Ping Sweep Countermeasures



- 1 Configure firewalls to detect and prevent ping sweep attempts instantaneously
- 2 Use intrusion detection systems and intrusion prevention systems like Snort to detect and prevent ping sweep attempts
- 3 Carefully evaluate the type of ICMP traffic flowing through enterprise networks
- 4 Cut off connections with any host that performs more than 10 ICMP ECHO requests
- 5 Use DMZs and allow only commands like ICMP ECHO\_REPLY, HOST UNREACHABLE, and TIME EXCEEDED within a DMZ
- 6 Limit ICMP traffic using Access Control Lists (ACLs) and grant permissions only to specific IP addresses such as ISPs

# Other Host Discovery Techniques

## ICMP Timestamp and Address Mask Ping Scan

- These techniques are alternatives for the traditional ICMP ECHO ping scan and are used to determine whether the target host is live, specifically when the administrators **block ICMP ECHO pings**

## TCP SYN Ping Scan

- Attackers send **empty TCP SYN packets** to a target host, and an **ACK** response means that the **host is active**
- ```
# nmap -sn -PS <target IP address>
```

## TCP ACK Ping Scan

- Attackers send **empty TCP ACK packets** to a target host, and an **RST** response means that the **host is active**
- ```
# nmap -sn -PA <target IP address>
```

## IP Protocol Ping Scan

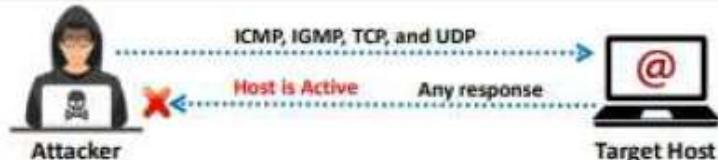
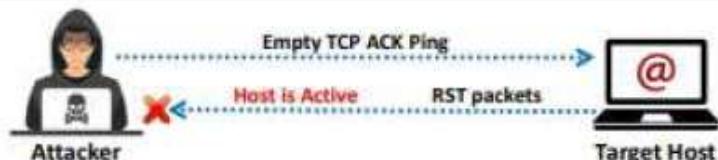
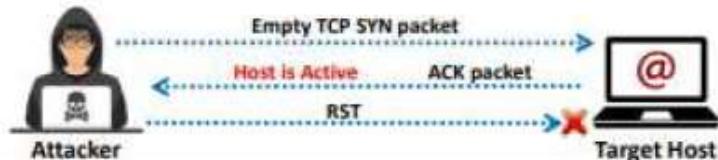
- Attackers send various **probe packets** to the target host using **different IP protocols**, and any response from any probe indicates that a host is active
- ```
# nmap -sn -PO <target IP address>
```

## ICMP Timestamp Ping Scan

```
# nmap -sn -PP <target IP address>
```

## ICMP Address Mask Ping Scan

```
# nmap -sn -PM <target IP address>
```



# Module Flow



1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Host Discovery**

7

**Draw Network Diagrams**

4

**Port and Service Discovery**

5

**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

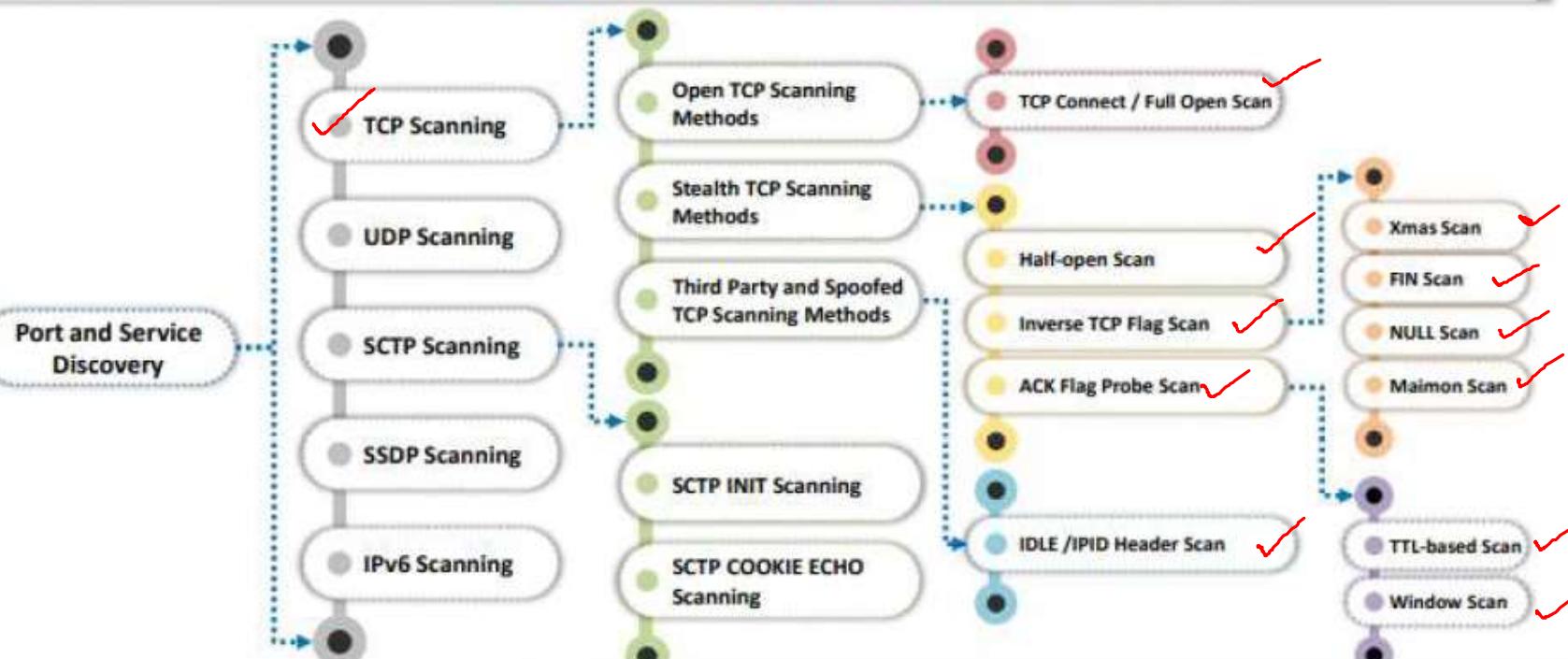
6

**Scanning Beyond IDS and Firewall**

# Port Scanning Techniques



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



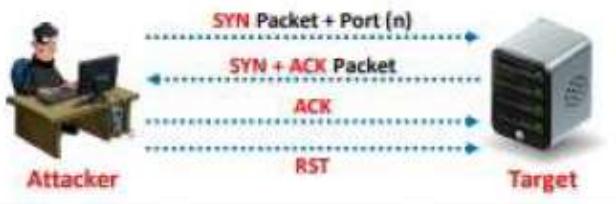
## TCP Connect/Full Open Scan

nmap -sT -v ip address

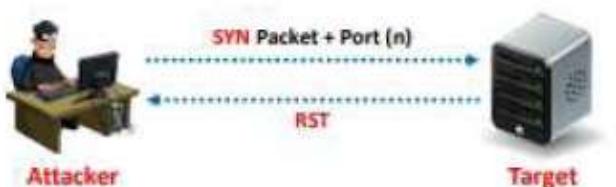


- The TCP Connect scan detects when a port is open after completing the **three-way handshake**
- TCP Connect scan **establishes a full connection** and then closes the connection by sending an **RST packet**
- It does not require **superuser privileges**

Scan result when a port is open



Scan result when a port is closed



```
Starting Nmap 7.00 ( https://nmap.org ) at 2019-10-23 15:04
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
Not shown: 944 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
5353/tcp  open  asusbt-server
5937/tcp  open  asusbt-server
MAC Address: 00:0C:29:8B:F4:91 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 45.43 seconds
Raw packets sent: 1 (28B) | Rcvd: 1 (28B)
```

<https://nmap.org>

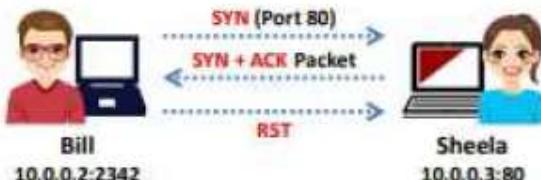
## Stealth Scan (Half-open Scan)

`nmap -sS -v`  
ip address

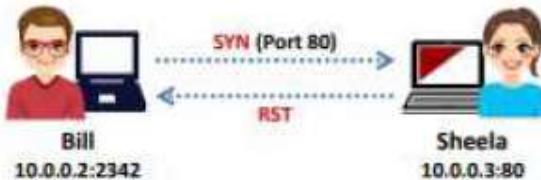


- Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of **three-way handshake signals**, thus leaving the connection half-open
- Attackers use stealth scanning techniques to **bypass firewall rules** as well as **logging mechanisms**, and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

Scan result when a port is open



Scan result when a port is closed



The screenshot shows the Nmap interface with the command `nmap -sS -v 10.0.0.10`. The output log shows the scan starting at 13:00 and completing at 13:08. It details various ports being scanned, including 80/tcp, 443/tcp, 8080/tcp, 3389/tcp, and 3307/tcp. A red bracket highlights the section of the log where Sheela's port 80 is shown as closed.

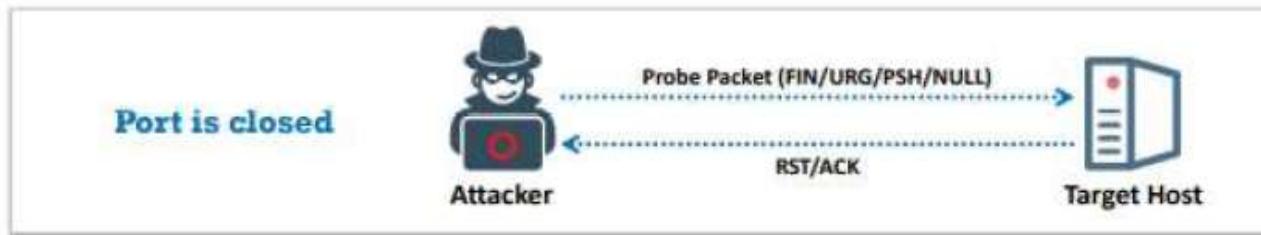
```
Starting Nmap 7.00 ( https://nmap.org ) at 2019-08-28 13:00
Initiating ARP Ping Scan at 13:00
Scanning 10.0.0.10 [1 port]
Completed ARP Ping Scan at 13:00, 0.02s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host... at 13:00
Completed Parallel DNS resolution of 1 host... at 13:00
0.02s elapsed
Initiating Stealth Scan at 13:00
Scanning 10.0.0.10 [1888 ports]
Discovered open port 80/tcp on 10.0.0.10
Discovered open port 443/tcp on 10.0.0.10
Discovered open port 8080/tcp on 10.0.0.10
Discovered open port 3389/tcp on 10.0.0.10
Discovered open port 3307/tcp on 10.0.0.10
Completed Stealth Scan at 13:08, 4.99s elapsed (1888 total ports)
Nmap scan report for 10.0.0.10
Host is up (0.00s latency).
Not shown: 1888 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-ds
3307/tcp  open  ms-ds
8080/tcp  open  http-proxy
MAC Address: 00:0C:29:8B:74:10 (Microsoft)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 5.14 seconds
Raw packets sent: 1996 (77.894B) | Rcvd: 18 (424B)
```

<https://nmap.org>

## Inverse TCP Flag Scan

- Attackers send **TCP probe packets** with a TCP flag (FIN, URG, PSH) set or with no flags, where no response implies that the port is open, whereas an RST response means that the port is closed



**Note:** Inverse TCP flag scanning is known as FIN, URG, PSH scanning based on the flag set in the probe packet. It is known as null scanning if there is no flag set

## Xmas Scan

- Using the Xmas scan, attackers send a TCP frame to a remote device with **FIN**, **URG**, and **PUSH** flags set
- FIN scanning works only with OSes that use an **RFC 793-based** TCP/IP implementation
- The Xmas scan will not work against any current version of **Microsoft Windows**



nmap -sX -v ip address



Xmas scan output using Zenmap

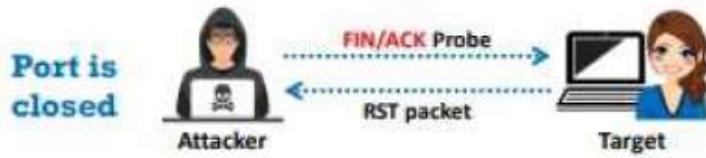
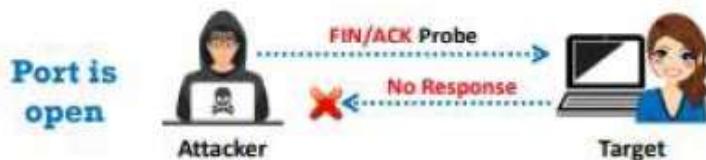
```
Zenmap
Scan Tools Profile Help
Target: 10.10.10.10 Profile: Scan Cancel
Command: nmap -sX -v 10.10.10.10
Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS * Host 10.10.10.10
# 10.10.10.10
nmap -sX -v 10.10.10.10
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-23
12:29 Standard Time
Initiating ARP Ping Scan at 12:29
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 12:29, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:29
Completed Parallel DNS resolution of 1 host. at 12:29,
0.03s elapsed
Initiating XMAS Scan at 12:29
Scanning 10.10.10.10 [1000 ports]
Completed XMAS Scan at 12:29, 23.66s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.10.10 are open[filtered]
MAC Address: 00:0C:29:00:F4:93 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 23.92 seconds
Raw packets sent: 2801 (88.828KB) | Rcvd: 5 (236B)
```

<https://nmap.org>

## TCP Maimon Scan

- Attackers send **FIN/ACK probes**, and if there is no response, then the port is **Open | Filtered**, but if an **RST packet** is sent in response, then the port is **closed**



nmap -sM -v ~~open|closed~~



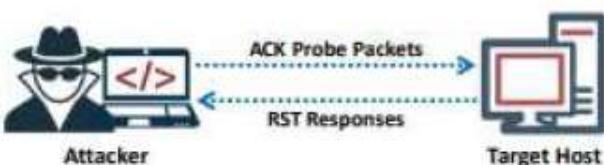
```
zenmap
Scan Tools Profile Help
Target: 10.10.10.10 Profile: Scan Cancel
Command: nmap -sM -v 10.10.10.10
Hosts Services
OS Host 10.10.10.10
nmap Output Ports / Hosts Topology Host Details Scan
Details
nmap -sM -v 10.10.10.10
Starting Nmap 7.60 ( https://nmap.org ) at 2019-10-23
12:32 Standard Time
Initiating ARP Ping Scan at 12:32
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 12:32, 0.05s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:32
Completed Parallel DNS resolution of 1 host. at 12:32, 0.03s elapsed
Initiating Maimon Scan at 12:32
Scanning 10.10.10.10 [1000 ports]
Completed Maimon Scan at 12:32, 23.47s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.10.10 are open|filtered
MAC Address: 00:0C:29:80:F4:93 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 23.77 seconds
Raw packets sent: 2001 (88.028KB) | Rcvd: 5 (236B)
https://nmap.org
```

# ACK Flag Probe Scan

- Attackers send **TCP probe packets set with an ACK flag** to a remote device, and then **analyze the header information** (TTL and WINDOW field) of received RST packets to determine if the **port is open or closed**

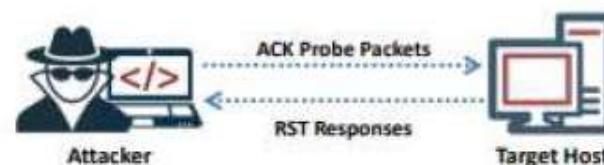
## TTL-based ACK Flag Probe scanning



```
1: host 10.2.2.11 port 20: F:RST -> ttl: 80 win: 0  
2: host 10.2.2.11 port 21: F:RST -> ttl: 80 win: 0  
3: host 10.2.2.11 port 22: F:RST -> ttl: 50 win: 0  
4: host 10.2.2.11 port 23: F:RST -> ttl: 80 win: 0
```

If the **TTL value of the RST packet** on a particular port is less than the boundary value of **64**, then that **port is open**

## Window-based ACK Flag Probe scanning



```
1: host 10.2.2.12 port 20: F:RST -> ttl: 64 win: 0  
2: host 10.2.2.12 port 21: F:RST -> ttl: 64 win: 0  
3: host 10.2.2.12 port 22: F:RST -> ttl: 64 win: 512  
4: host 10.2.2.12 port 23: F:RST -> ttl: 64 win: 0
```

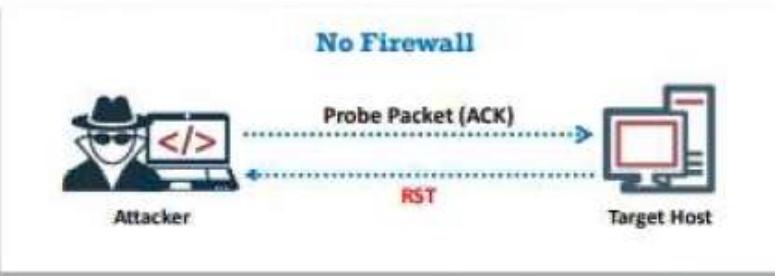
If the **window value of the RST packet** on a particular port has a **non-zero value**, then that **port is open**

## ACK Flag Probe Scan (Cont'd)

*nmap -sA -v*  
*if admin*



- ACK flag probe scanning can also be used to **check the filtering system of a target**
- Attackers send an **ACK probe packet** with a random sequence number, and no response implies that the **port is filtered** (stateful firewall is present), whereas an RST response means that the **port is not filtered**



```
ZeNmap
Scan Tools Profile Help
Target: 10.10.10.10
Profile: Standard Time
Command: nmap -sA -v 10.10.10.10
Hosts Services
OS * Host
# 10.10.10.10
nmap -sA -v 10.10.10.10
Starting nmap 7.00 ( https://nmap.org ) at 2019-10-23 13:08
Standard Time
Initiating ARP Ping Scan at 13:08
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 13:08, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host... at 13:08
Completed Parallel DNS resolution of 1 host... at 13:08, 0.02s
elapsed
Initiating ACK Scan at 13:08
Scanning 10.10.10.10 [1000 ports]
Completed ACK Scan at 13:08, 23.50s elapsed (1000 total ports)
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
All 1000 scanned ports on 10.10.10.10 are filtered
MAC Address: 00:0C:29:00:74:93 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 23.75 seconds
Raw packets sent: 2001 (88.028KB) | Rcvd: 5 (192B)
```

<https://nmap.org>

## IDLE/IPID Header Scan



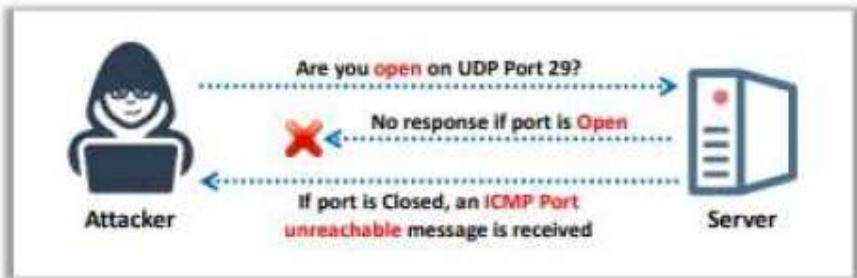
- Every IP packet on the Internet has a fragment identification number (IPID); an OS increases the IPID for each packet sent, thus, probing an IPID gives an attacker the **number of packets sent** after the last probe
- A machine that receives an **unsolicited SYN|ACK packet** will respond with an RST. An unsolicited RST will be ignored

- Send SYN + ACK packet to the zombie machine to **probe its IPID number**
- A zombie machine not expecting an SYN + ACK packet will send an **RST packet**, disclosing the IPID. Analyse the RST packet from the zombie machine to **extract the IPID**
- Send a SYN packet to the **target machine (port 80)** to spoof the IP address of the "zombie"
- If the port is open, the target will send a **SYN+ACK packet** to the zombie, and the zombie will send an RST to the target in response
- If the port is closed, the target will send an **RST to the zombie**, but the zombie will not send anything back
- Probe the zombie IPID again. An IPID increased by **2** will indicate an **open port**, whereas an IPID increased by **1** will indicate a **closed port**



# UDP Scanning

nmap -sU -v ip address



- UDP Port Open**
- There is no **three-way TCP handshake** for UDP scanning
  - The system does not respond with a message when the **port is open**
- UDP Port Closed**
- If a UDP packet is sent to a closed port, the system will respond with an **ICMP port unreachable message**
  - Spywares, Trojan horses**, and other malicious applications use UDP ports

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-07  
Nmap done: 1 IP address (1 host up) scanned in 8.23 seconds  
Raw packets sent: 2002 (57.96KB) | Rcvd: 5 (605B)
```

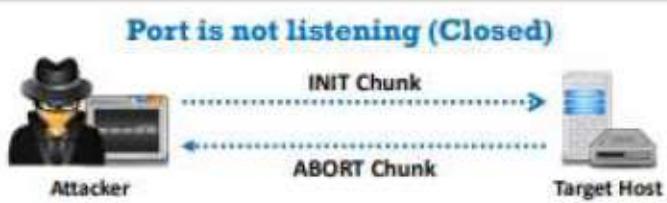
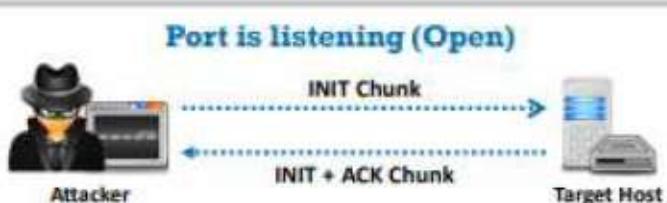
https://nmap.org

# SCTP INIT Scanning

nmap -sY -v ipaddress



- Attackers send an **INIT chunk** to the target host, and an **INIT+ACK chunk** response implies that the **port is open**, whereas an **ABORT Chunk** response means that the **port is closed**
- No response from the target, or a response of an **ICMP unreachable exception** indicates that the port is a **Filtered port**



```
Zenmap
Scan Tools Profile Help
Target: 10.10.10.10 Profile: Scan Cancel
Command: nmap -sY -v 10.10.10.10
Hosts Services Nmap Output Ports / Hosts Topology Host Details Scan
OS * Host IP: 10.10.10.10
nmap -sY -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-07
13:11 Standard Time
Initiating ARP Ping Scan at 13:11
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 13:11, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 13:11
Completed Parallel DNS resolution of 1 host. at 13:11, 0.02s elapsed
Initiating SCTP INIT Scan at 13:11
Scanning 10.10.10.10 [52 ports]
Completed SCTP INIT Scan at 13:11, 1.97s elapsed (52 total ports)
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
All 52 scanned ports on 10.10.10.10 are filtered
MAC Address: 00:0C:29:79:03:B9 (VMware)

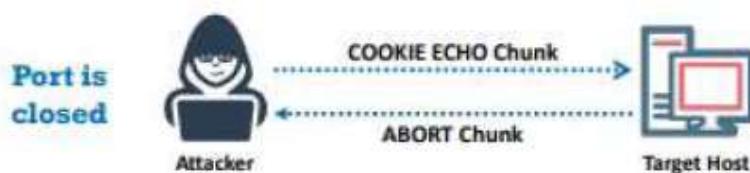
Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 2.30 seconds
Raw packets sent: 103 (5.332KB) | Rcvd: 4 (2688)
https://nmap.org
```

## SCTP COOKIE ECHO Scanning

nmap -sZ -v ipaddress



- Attackers send a **COOKIE ECHO chunk** to the target host, and **no response** implies that the **port is open**, whereas an **ABORT Chunk** response means that the **port is closed**
- It is **not blocked** by non-stateful firewall rulesets
- Only a **good IDS** will be able to **detect SCTP COOKIE ECHO chunk**



```
Zenmap
Scan Tools Profile Help
Target: 10.10.10.10
Profile: Scan Cancel
Command: nmap -sZ -v 10.10.10.10
Hosts Services Nmap Output Ports / Hosts Topology Host Details Scans
OS Host 10.10.10.10
nmap -sZ -v 10.10.10.10
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-07
11:12     Standard Time
Initiating ARP Ping Scan at 11:12
Scanning 10.10.10.10 [1 port]
Completed ARP Ping Scan at 11:12, 0.06s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 11:12
Completed Parallel DNS resolution of 1 host. at 11:12, 0.02s elapsed
Initiating SCTP COOKIE-ECHO Scan at 11:12
Scanning 10.10.10.10 [52 ports]
Completed SCTP COOKIE-ECHO Scan at 11:12, 2.25s elapsed (52 total ports)
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
Not shown: 58 open|filtered ports
PORT      STATE      SERVICE
3225/sctp filtered  rtp-fm
4739/sctp filtered  ipfix
MAC Address: 00:0C:29:79:02:B8 (VMware)

Read data files from: C:\Program Files (x86)\Nmap
Nmap done: 1 IP address (1 host up) scanned in 2.53 seconds
Raw packets sent: 104 (4.148KB) | Rcvd: 3 (164B)
```

<https://nmap.org>

# SSDP and List Scanning



## SSDP Scanning

- The Simple Service Discovery Protocol (SSDP) is a network protocol that **works in conjunction with the UPnP to detect plug and play devices**
- Vulnerabilities in UPnP may allow attackers to launch **Buffer overflow or DoS attacks**
- Attacker may use the **UPnP SSDP M-SEARCH** information discovery tool to check if the machine is vulnerable to UPnP exploits or not

```
File Edit View Search Terminal Help
nmap -u auxiliary/scanner/udns/msearch
nmap auxiliary/scanner/udns/msearch > set RHOSTS 192.168.1.10
RHOSTS: 192.168.1.10
nmap auxiliary/scanner/udns/msearch > show options
Module options (auxiliary/scanner/udns/msearch):
Name      Current Setting    Required  Description
BATCHSIZE      256        yes        The number of hosts to probe in each job
REPORT_LOCATION  False      yes        This determines whether to report the IP
UPNP_ENDPOINT_SERVICE_IDENTIFIED_BY_UPNP  True      yes        If an UPNP endpoint service is identified by UPNP
RHOSTS      192.168.1.10  yes        The target host(s), usage: CIDR, identifier
UPNP_THREADS      1000      yes        The target port, UPNP
THREADS      10         yes        The number of concurrent threads
nmap auxiliary/scanner/udns/msearch > exploit
[*] Scanning UPnP SSDP probes to 192.168.1.10-192.168.1.10 (1 hosts)
[!] No UPnP endpoints found
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
nmap auxiliary/scanner/udns/msearch >
```

## List Scanning

- This type of scan simply generates and prints a **list of IPs/Names** without actually pinging them
- A **reverse DNS resolution** is performed to identify the host names

nmap -sL -v ipaddress

ZMap  
Scan Tools Profile Help  
Target: 10.10.10.10 | Profile: | Scan | Launch  
Command: nmap -sL -v 10.10.10.10  
Hosts Services Nmap Output Ports / Hosts Topology Host Details Scan  
0/1 Host 10.10.10.10 | Details  
Starting Nmap 7.70 [ https://nmap.org ] at 2019-06-07  
11:09 Standard Time  
Initiating Parallel DNS resolution of 1 host. at 11:09  
Completed Parallel DNS resolution of 1 host. at 11:09,  
0.02s elapsed  
Nmap scan report for: 10.10.10.10  
Host down! 1 IP address (0 hosts up) scanned in 0.19  
seconds  
Filter Hosts  
<https://nmap.org>

## IPv6 Scanning

$2^{32}$  IPv4

$2^{128}$  IPv6



- IPv6 increases the IP address size from **32 bits** to **128 bits** to support more levels of address hierarchy
- Attackers need to harvest IPv6 addresses from **network traffic**, **recorded logs**, or **Received from:** header lines in archived emails
- Attackers can use the **-6** option in Zenmap to **perform IPv6 scanning**

TCP/UDP  
Well known ports



0 to  $10^{23}$

```
root@...:~# nmap -6 scanme.nmap.org
Starting Nmap 7.60 ( http://nmap.org ) at 2023-04-25 04:25 UTC
Nmap scan report for scanme.nmap.org (2600:3c01::f03c:91ff:fe18:bb2f)
Host is up (0.062s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
31337/tcp open  Elite

Nmap done: 1 IP address (1 host up) scanned in 3.94 seconds
https://nmap.org
```

nmap -6 *Scanning*

TCP/UDP  
Registered ports

$10^{24}$  to 49,151

## Service Version Discovery

nmmap -sV → ipaddresses



- 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**
  - For example, when an attacker detects **SMBv1 protocol** as a running service on a target Windows-based machine, then the attacker can easily perform the **WannaCry ransomware attack**
  - In Zenmap, the **-sV** option is used to detect service versions



The screenshot shows the Zenmap interface with the following details:

- Target:** 10.10.10.10
- Profile:** (empty)
- Command:** nmap -sV 10.10.10.10
- Hosts:** OS Host 10.10.10.10
- Nmap Output:** The output window displays the following:

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-10
17:44      Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.0014s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE      VERSION
139/tcp    open  msrpc        Microsoft Windows: RPC
389/tcp    open  netbios-ssn  Microsoft Windows netbios-ssn
445/tcp    open  microsoft-ds Microsoft Windows 7 - 10
            microsoft-ds (workgroup: WORKGROUP)
5357/tcp   open  http         Microsoft HTTPAPI httpd
            2.0 (SSDP/UPnP)
MAC Address: 00:0C:29:79:02:B9 (VMware)
Service Info: Host: DESKTOP-E3UJ5VL; OS: Windows; CPE: cpe:/o:microsoft:windows
```
- Services:** The services table shows:

| PORT     | STATE | SERVICE      | VERSION                                                         |
|----------|-------|--------------|-----------------------------------------------------------------|
| 139/tcp  | open  | msrpc        | Microsoft Windows: RPC                                          |
| 389/tcp  | open  | netbios-ssn  | Microsoft Windows netbios-ssn                                   |
| 445/tcp  | open  | microsoft-ds | Microsoft Windows 7 - 10<br>microsoft-ds (workgroup: WORKGROUP) |
| 5357/tcp | open  | http         | Microsoft HTTPAPI httpd<br>2.0 (SSDP/UPnP)                      |
- Ports / Hosts:** Shows the host 10.10.10.10.
- Topology:** Not applicable for this screenshot.
- Host Details:** Not applicable for this screenshot.
- Scans:** Not applicable for this screenshot.
- Details:** A button to view detailed information about the host.
- Bottom Status:** Service detection performed. Please report any incorrect results at <https://nmap.org/submit/>. Nmap done: 1 IP address (1 host up) scanned in 13.88 seconds

# Nmap Scan Time Reduction Techniques



💡 In Nmap, **performance** and **accuracy** can be achieved by reducing the scan timing

## Scan Time Reduction Techniques

- 1 Omit Non-critical Tests
- 2 Optimize Timing Parameters
- 3 Separate and Optimize UDP Scans
- 4 Upgrade Nmap
- 5 Execute Concurrent Nmap Instances 
- 6 Scan from a Favorable Network Location
- 7 Increase Available Bandwidth and CPU Time

# 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** by routers and for **filtering** by firewalls **cannot be bypassed** using particular source ports 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

# Module Flow



1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Host Discovery**

7

**Draw Network Diagrams**

4

**Port and Service Discovery**

5

**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

6

**Scanning Beyond IDS and Firewall**

# OS Discovery/Banner Grabbing

→ Find appropriate tool to determine OS in target machine?



- 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

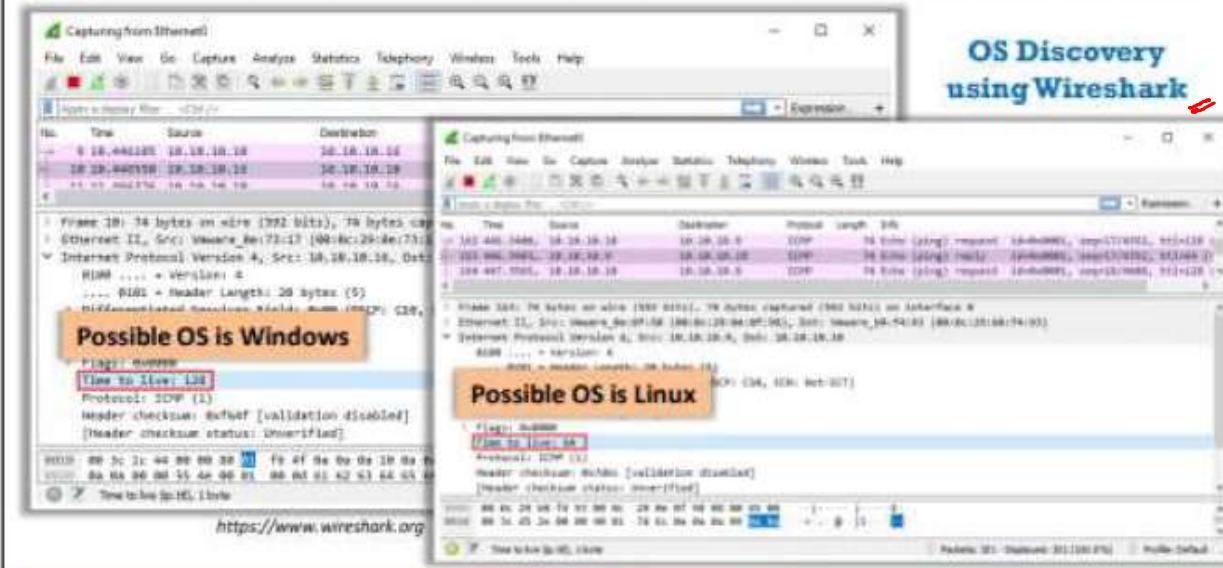
- **Banner grabbing from error messages**  
Error messages provide information such as the type of server, type of OS, and SSL tool used by the target remote system.
- **Sniffing the network traffic**  
Capturing and analyzing packets from the target enables an attacker to determine the OS used by the 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 (Kernel 2.4 and 2.6)            | 64           | 5840            |
| Google Linux                          | 64           | 5720            |
| FreeBSD                               | 64           | 65535           |
| OpenBSD                               | 64           | 16384           |
| Windows 95                            | 32           | 8192            |
| Windows 2000                          | 128          | 16384           |
| Windows XP                            | 128          | 65535           |
| Windows 98, Vista and 7 (Server 2008) | 128          | 8192            |
| iOS 12.4 (Cisco Routers)              | 255          | 4128            |
| Solaris 7                             | 255          | 8760            |
| AIX 4.3                               | 64           | 16384           |

# OS Discovery using Nmap and Unicornscan



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

The screenshot shows the Nmap interface with the command `nmap -O 10.10.10.16` highlighted. The output window displays the following information:

```
Starting Nmap 7.76 ( https://nmap.org ) at 2019-06-18 07:23 EDT
Nmap scan report for 10.10.10.16
Host is up (0.00092s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
139/tcp    open  msrpc
139/tcp    open  netbios-ssn
445/tcp   open  microsoft-ds
MAC Address: 00:0C:29:ED:C2:95 (VMware)
Device Type: general purpose
Running: Microsoft Windows 2016
OS CPE: cpe:/microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build 10586 - 14393
Network Distance: 1 hop

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

- In **Unicornscan**, the OS of the target machine can be identified by **observing the TTL values** in the acquired scan result

The screenshot shows a terminal window titled "Parrot Terminal" with the command `unicornscan 10.10.10.16 -Iv` running. The output shows a large list of TTL values. A red box highlights the first few lines of the output.

```
#UNICORNSCAN 10.10.10.16 -Iv
using 10.10.10.16/12 mode 'TCPScan' ports: 7,9,11,13,18,19,21-23,25,37,39,43,49,56,53,
55,67-79,81,89,91,109,193-197,199-211,213,218,219,222,229,239,243,250,181-184,
186-197,199,291,199-202,204-209,209-210,213-220,345,346,347,389-397,389,408-407,422-442,
45-445,487,506,512-514,517,518,520,525,533,538,540,554,563,587,618-612,631-634,636-642,
653,655,657,660,708,756-759,760,779,800,873,901,923,941,946,992-995,1001,1023-1039,1089,
1220,1214,1234-1241,1394,1399,1397,1423-1425,1431,1459,1524,1525,1645,1646,1649,1701-17
718,1719,1720,1723,1755,1812,1813,2048,2050,2161-2164,2166,2194,2223,2245,2401,243
9,2431,2432,2435,2583,2626,2770,2777,2988,2989,3029,3130,3150,3232,3296,3389,3496,3493,
3542-3545,3632,3698,3801,4088,4408,4322,4597,4884,5007,5136-5139,5150,5151-5222,5269,53
6,5354,5355,5422,5425,5439,5563,5555,5678,6469,6887,6947,6543,6544,6789,6836
,6966-6970,7046-7059,7028,7100,7983-8082,8688,8757,8879,9089,9161-9189,9225,9359,1
1000,10026,10037,10067,10088,10089,10167,10498,11201,15345,12001-17003,18753,20011,2001
,21558,22223,26274,27374,27441,22373,31335-31338,31787,31789,31790,31791,32068,32767-3
2798,33380,47262,69201,54321,57541,58968,58989,58866,58211,60980,60865,61068,6134
8,61466,61663,63485,63088,63089,64429,65000,65306,65338-65335 - pps=300
using interface(s) eth0
scanning 1.98e+09 total hosts with 3.38e+02 total packets, should take a little longer t
than 8 seconds
TCP open 10.10.10.16:2002 ttl 128
TCP open 10.10.10.16:80 ttl 128
TCP open 10.10.10.16:445 ttl 128
TCP open 10.10.10.16:139 ttl 128
TCP open 10.10.10.16:135 ttl 128
TCP open 10.10.10.16:3389 ttl 128
TCP open 10.10.10.16:88 ttl 128
```

Possible OS is Windows

<https://sourceforge.net>

# OS Discovery using Nmap Script Engine



- Nmap script engine (NSE) can be used to **automate a wide variety of networking tasks** by allowing the users to **write and share scripts**
- Attackers use various scripts in the Nmap Script Engine to **perform OS discovery** on the target machine
- For example, in Nmap, **smb-os-discovery** is an inbuilt script that can be used for **collecting OS information** on the target machine **through the SMB protocol**
- In Zenmap, the **-sC** option or **--script** option is used to activate the NSE scripts

The screenshot shows the Zenmap interface with the following details:

- Target:** 10.10.10.10
- Command:** nmap --script smb-os-discovery.nse 10.10.10.10
- Host:** 10.10.10.10
- Ports:** 135/tcp open msrpc  
139/tcp open netbios-ssn  
445/tcp open microsoft-ds  
5257/tcp open usdapi  
MAC Address: 00:0C:29:79:02:09 (VMware)
- Script Results:** smb-os-discovery:
  - OS: Windows 10 Enterprise 17763 (Windows 10 Enterprise 6.3)
  - OS CPE: cpe:/o:microsoft:windows\_10:-
  - Computer name: DESKTOP-E3UJ5VL
  - NetBIOS computer name: DESKTOP-E3UJ5VL\x00
  - Workgroup: WORKGROUP\x00
  - System time: 2019-06-10T18:14:19+05:30
- Summary:** Nmap done: 1 IP address (1 host up) scanned in 26.22 seconds

# OS Discovery using IPv6 Fingerprinting



- IPv6 Fingerprinting can be used to **Identify the OS running** on the target machine



- IPv6 fingerprinting has the **same functionality** as that of IPv4



- The difference between IPv6 and IPv4 fingerprinting is that the IPv6 uses several **additional advanced probes specific to IPv6** along with a **separate OS detection engine that is specialized for IPv6**



- In Zenmap, the **-6 option** and **-O option** are used to perform OS discovery using the IPv6 fingerprinting method

- Syntax: # nmap **-6 -O <target>**



# 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 **ServerMask** (<http://www.port80software.com>) tools to disable or change banner information
- Apache 2.x with **mod\_headers** module - use a directive in **httpd.conf** file to change banner information **Header set Server "New Server Name"**
- Alternatively, change the **ServerSignature** line to **ServerSignature Off** in **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 web technologies**
  - Change **application mappings** such as .asp with .htm or .foo, etc. to disguise the identity of servers
  - Apache users can use **mod\_negotiation** directives
  - IIS users use tools such as **PageXchanger** to manage the file extensions
- ✓ It is better if the file extensions are not used at all

# Module Flow



1

**Network Scanning Concepts**

2

**Scanning Tools**

3

**Host Discovery**

4

**Port and Service Discovery**

5

**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

6

**Scanning Beyond IDS and Firewall**

7

**Draw Network Diagrams**

# IDS/Firewall Evasion Techniques



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

**1** Packet Fragmentation

**2** Source Routing

**3** Source Port Manipulation

**4** IP Address Decoy

**5** IP Address Spoofing

**6** Creating Custom Packets

**7** Randomizing Host Order

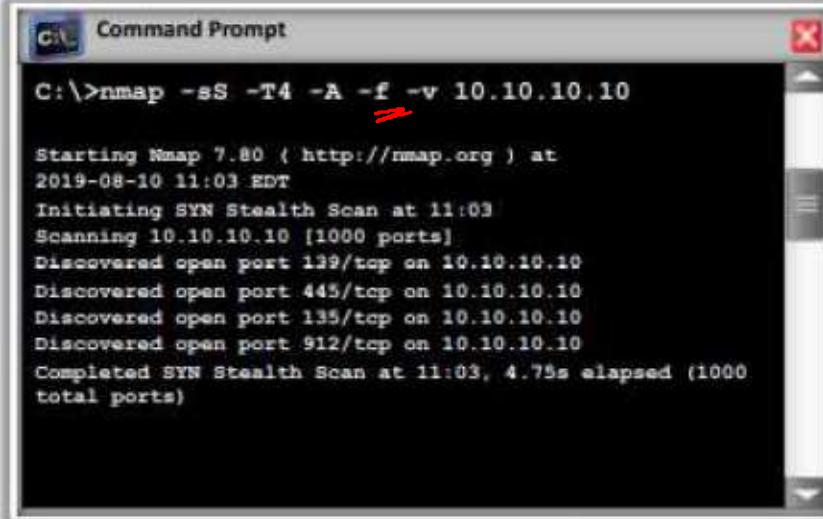
**8** Sending Bad Checksums

**9** Proxy Servers

**10** 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



```
C:\>nmap -sS -T4 -A -f -v 10.10.10.10
Starting Nmap 7.80 ( http://nmap.org ) at 2019-08-10 11:03 EDT
Initiating SYN Stealth Scan at 11:03
Scanning 10.10.10.10 [1000 ports]
Discovered open port 139/tcp on 10.10.10.10
Discovered open port 445/tcp on 10.10.10.10
Discovered open port 135/tcp on 10.10.10.10
Discovered open port 912/tcp on 10.10.10.10
Completed SYN Stealth Scan at 11:03, 4.79s elapsed (1000 total ports)
```

- 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

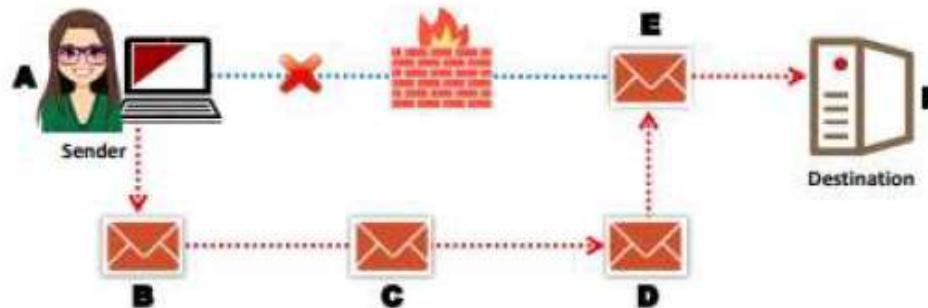


## Source Routing



- 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**

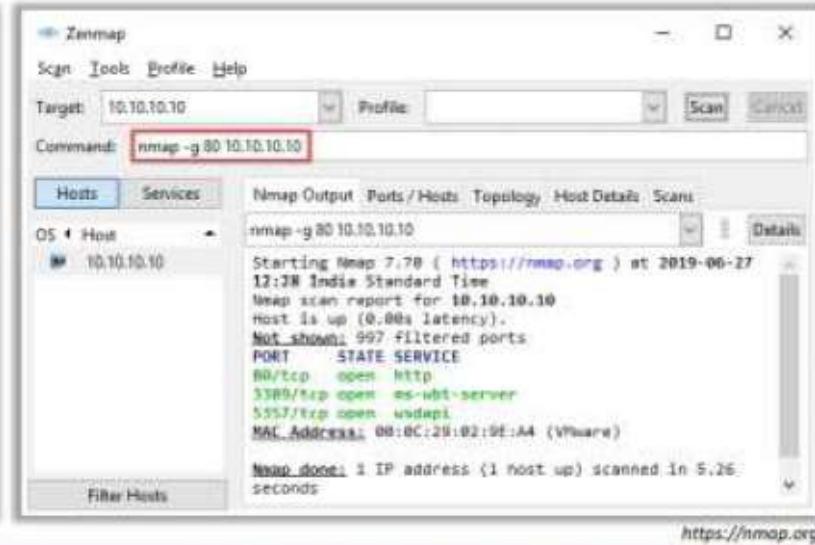
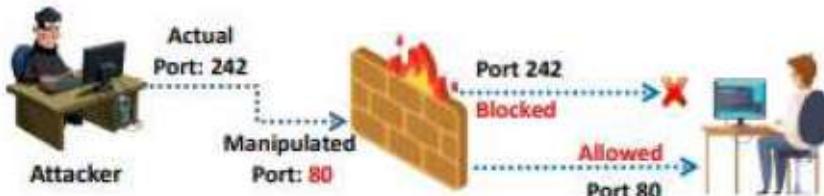
This figure shows source routing, where the originator dictates the eventual route of the traffic



## 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: 10.10.10.10 Profile:

Command: nmap -g 80 10.10.10.10

Hosts Services Nmap Output Ports / Hosts Topology Host Details Scan

OS: 1 Hour 10.10.10.10

nmap -g 80 10.10.10.10

Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-27 12:38 INDIA Standard Time

Nmap scan report for 10.10.10.10

Host is up (0.00s latency).

Not shown: 997 filtered ports

| PORT     | STATE | SERVICE       |
|----------|-------|---------------|
| 80/tcp   | open  | http          |
| 3389/tcp | open  | ms-wbt-server |
| 5957/tcp | open  | wsddapi       |

MAC Address: 08:0C:2B:02:9E:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.26 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)

The screenshot shows the ZENMAP graphical interface for Nmap. The 'Targets' field contains '10.10.10.10'. The 'Command' field shows 'nmap -D RND:10 10.10.10.10'. The 'Hosts' tab is selected, displaying the target '10.10.10.10'. The 'Services' tab shows the results of the scan:  
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-27 12:44  
India Standard Time  
Nmap scan report for 10.10.10.10  
Host is up (0.027s latency).  
Not shown: 997 filtered ports  
PORT STATE SERVICE  
80/tcp open http  
3389/tcp open ms-wbt-server  
5357/tcp open unknown  
MAC Address: 00:0C:29:83:9E:A4 (VMware)  
Nmap done: 1 IP address (1 Host up) scanned in 8.41 seconds

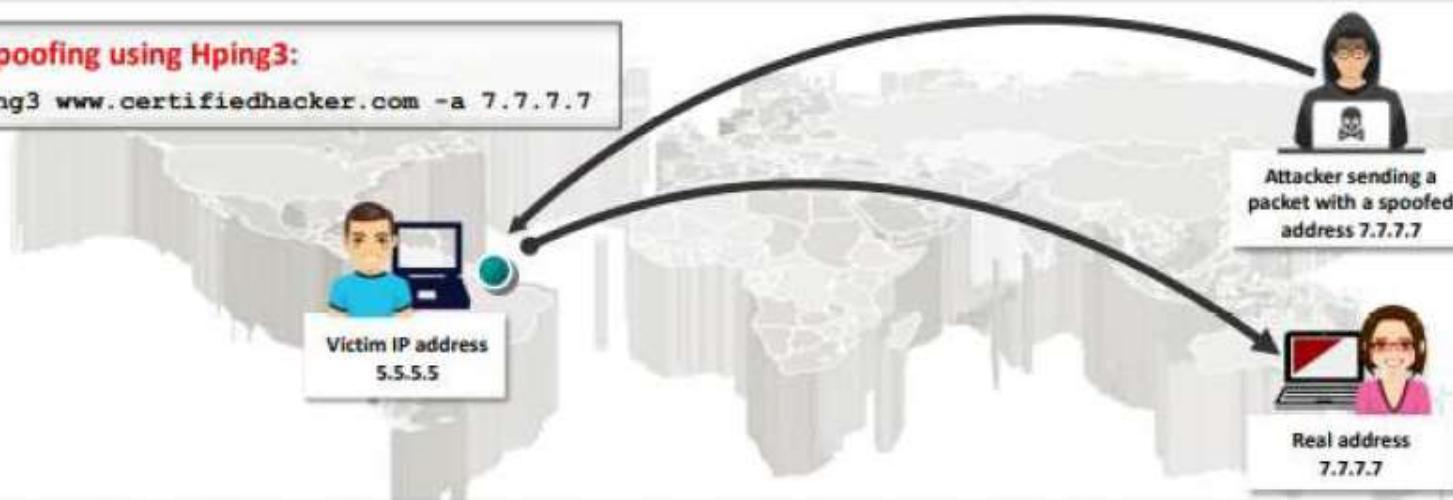
# IP Address Spoofing



- 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

## IP spoofing using Hping3:

```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```



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

## IP Spoofing Detection Techniques: Direct TTL Probes

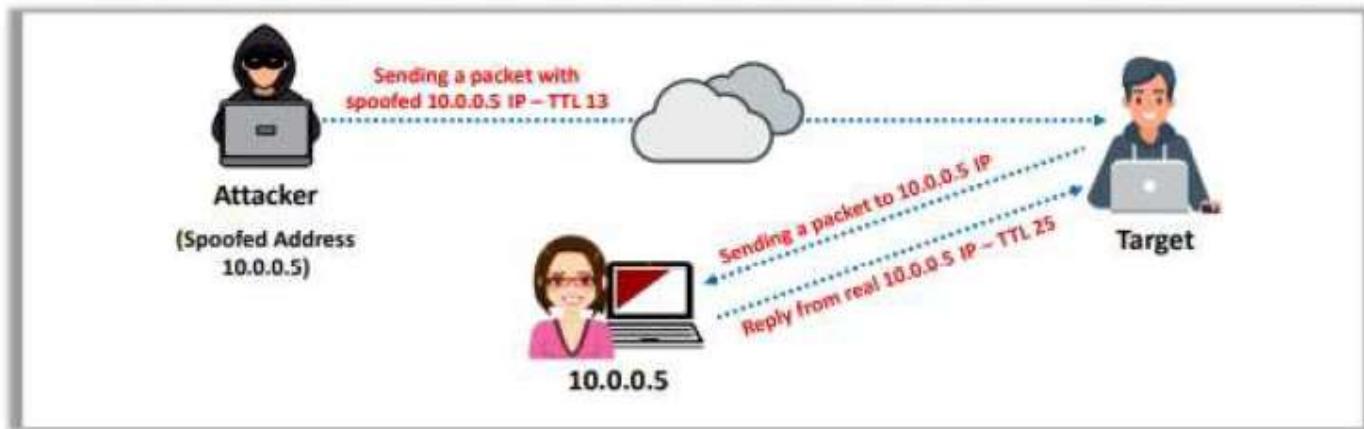


1

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

2

This technique is successful when the attacker is in a **different subnet** from that of the victim

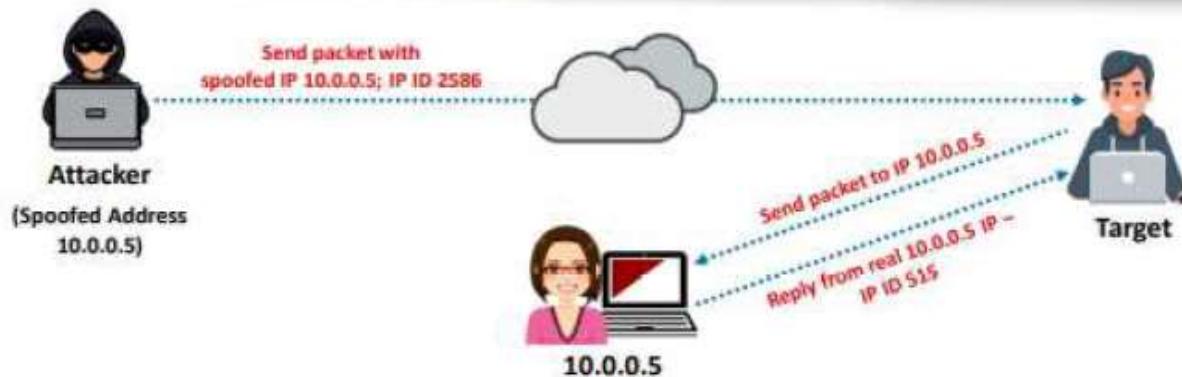


**Note:** Normal traffic from one host can contrast TTLs depending on traffic patterns

## IP Spoofing Detection Techniques: IP Identification Number



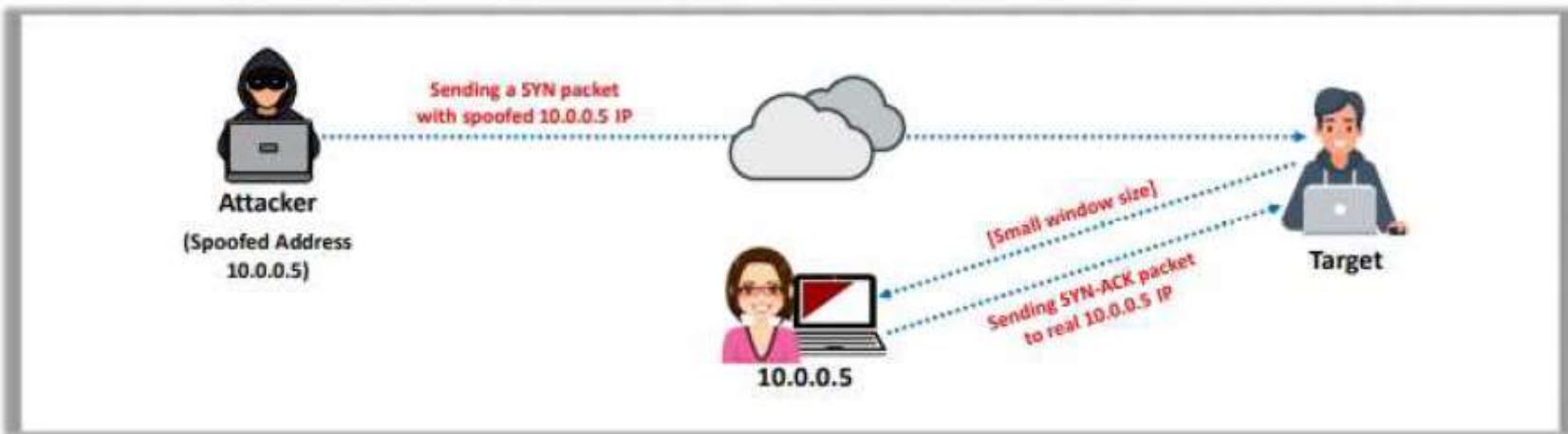
- 01** Send a probe to the host of a suspected spoofed traffic that triggers a reply and **compare the IPID** with the suspected traffic
- 02** If the IPIDs are **not close in value** to the packet being checked, then the suspected traffic is spoofed
- 03** This technique is considered reliable even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



## IP Spoofing Detection Techniques: TCP Flow Control Method



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



# IP Spoofing Countermeasures



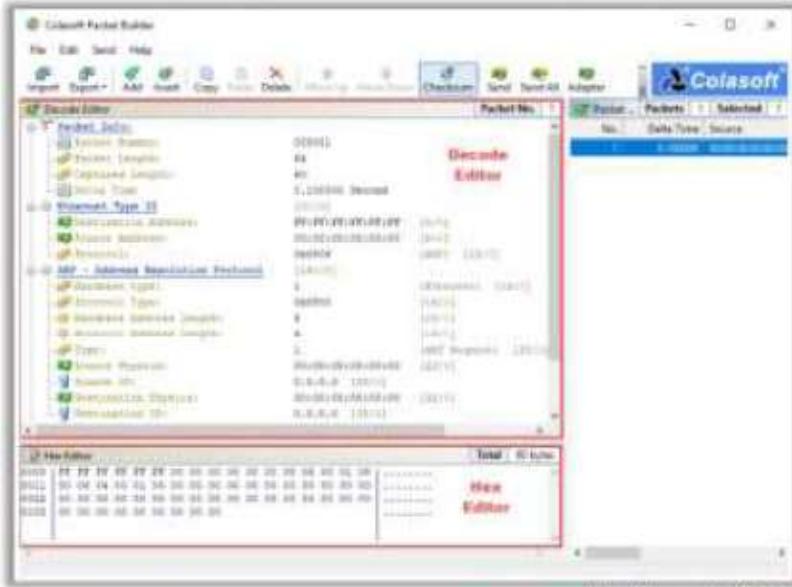
- 1 Encrypt all the network traffic using cryptographic network protocols such as IPsec, TLS, SSH, and HTTPS
- 2 Use multiple firewalls to provide a multi-layered depth of protection
- 3 Do not rely on IP-based authentication
- 4 Use a random initial sequence number to prevent IP spoofing attacks based 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

# Creating Custom Packets



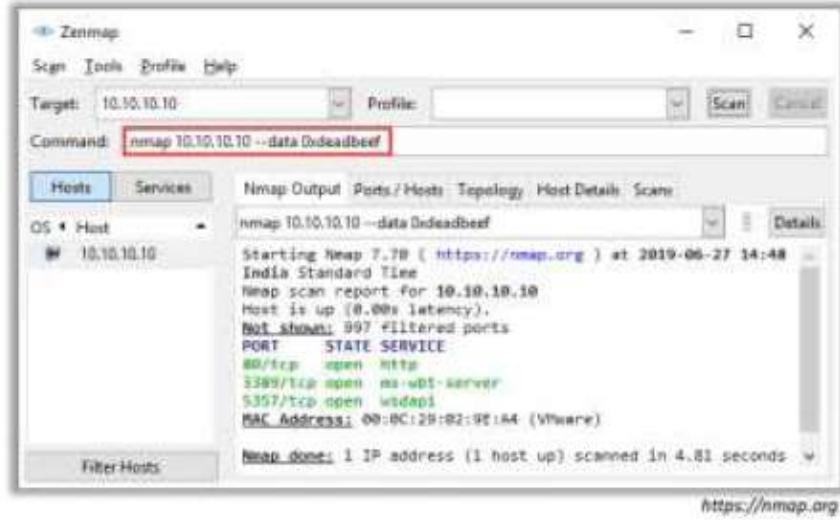
## Creating Custom Packets by using Packet Crafting Tools

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



## Creating Custom Packets by Appending Custom Binary Data

- Attackers send binary data (**0's and 1's**) as payloads in transmitted packets to scan beyond firewalls
- Example: `--data Oxdeadbeef`



# Creating Custom Packets (Cont'd)



## Creating Custom Packets by Appending Custom String

- Attackers send a **regular string as payloads** in the packets sent to the target machine for scanning beyond the firewall
- Example: `--data-string "Ph34r my l33t skills"`

Zenmap window showing the command `nmap 10.10.10.10 --data-string "Ph34r my l33t skills"` highlighted in red.

Output:

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-27 14:51
India Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-sql-server
5935/tcp  open  wsdapi
MAC Address: 00:0C:29:02:9E:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 4.97 seconds
```

## Creating Custom Packets by Appending Random Data

- Attackers **append a number of random data bytes** to most of the packets sent without any protocol-specific payloads
- Example: `--data-string 5`

Zenmap window showing the command `nmap 10.10.10.10 --data-string 5` highlighted in red.

Output:

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-06-27 14:56
India Standard Time
Nmap scan report for 10.10.10.10
Host is up (0.00s latency).
Not shown: 997 filtered ports
PORT      STATE SERVICE
80/tcp    open  http
3389/tcp  open  ms-sql-server
5935/tcp  open  wsdapi
MAC Address: 00:0C:29:02:9E:A4 (VMware)

Nmap done: 1 IP address (1 host up) scanned in 5.57 seconds
```

<https://nmap.org>

# 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 window showing the command: `nmap --randomize-hosts 10.10.10.10`. The output shows a host at 10.10.10.10 with open ports 80/tcp (HTTP) and 3389/tcp (MS-DATAGRAM). MAC Address: 00:0C:29:02:9E:A4 (VMware).

## Sending Bad Checksums

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

Zenmap window showing the command: `nmap --badsum 10.10.10.10`. The output shows a host at 10.10.10.10 with all scanned ports filtered. MAC Address: 00:0C:29:02:9E:A4 (VMware).

<https://nmap.org>

# Proxy Servers



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

- 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

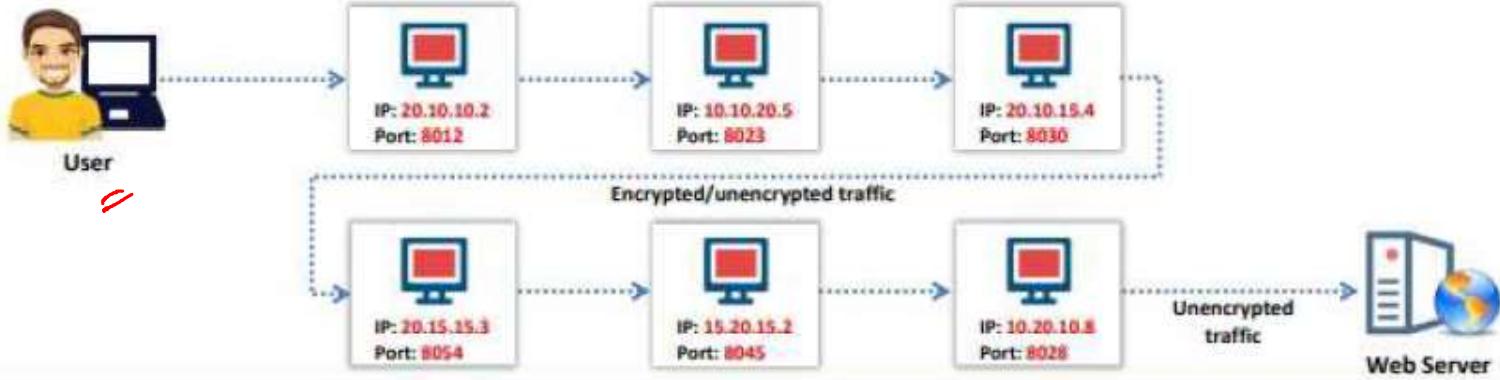
## Why Attackers Use Proxy Servers?

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

# Proxy Chaining



- ① User **requests a resource** from the destination
- ② Proxy client at the user's system connects to a **proxy server** and passes the request to proxy server
- ③ The proxy server **strips the user's identification information** and passes the request to next proxy server
- ④ This process is repeated by all the proxy servers in the **chain**
- ⑤ At the end, the **unencrypted request** is passed to the web server



# Proxy Tools



## Proxy Switcher

Proxy Switcher allows you to **surf anonymously on the Internet** without disclosing your IP address

The screenshot shows the 'Proxy Switcher Unregistered (Direct Connection)' window. On the left is a sidebar with various proxy scanning options like 'New (100%)', 'Core (5%)', 'High Anonymous (2%)', 'SSL (0%)', 'Elite (0%)', 'Dedicated (0%)', and 'Least Anonymous (100%)'. The main area displays a table of proxy servers:

| Server               | Status | Response | Country            |
|----------------------|--------|----------|--------------------|
| 103.245.180.130:3000 | Alive  | 703ms    | INDIA              |
| 103.95.254.117:8080  | Alive  | 121ms    | INDIA              |
| 107.71.128.138:3120  | Alive  | 539ms    | UNITED STATES      |
| 109.127.186.229:80   | Alive  | 520ms    | BRAZIL             |
| 109.95.251.247:3120  | Alive  | 539ms    | INDIA              |
| 123.249.193.195:3000 | Alive  | 139ms    | INDIA              |
| 209.250.123.174:3000 | Alive  | 223ms    | BRAZIL             |
| 82.64.6.11:8080      | Alive  | 164ms    | RUSSIAN FEDERATION |
| 82.162.174.31:88     | Alive  | 109ms    | BRAZIL             |
| 209.25.174.211:3120  | Alive  | 148ms    | ARGENTINA          |
| 107.71.84.127:3120   | Alive  | 121ms    | UNITED STATES      |
| 103.91.33.42:3067    | Alive  | 172ms    | HONG KONG          |
| 103.91.33.42:88      | Alive  | 105ms    | HONG KONG          |

At the bottom, there are tabs for 'Dedicated', 'Keep Alive', and 'Auto Switch'. A status bar at the bottom shows connection details for targets.

## CyberGhost VPN

CyberGhost VPN **hides your IP** and replaces it with one of your choice, thus allowing you to surf anonymously

The screenshot shows the 'All servers' list in the CyberGhost VPN application. It includes columns for Name, Distance, User, and Traffic. One server, 'Albania', is highlighted in yellow. A separate window titled 'VPN not connected!' shows a large yellow button labeled 'Connect to' and a progress bar indicating a connection attempt to 'Albania'.

| Name         | Distance  | User | Traffic |
|--------------|-----------|------|---------|
| Albania      | 6.133 km  | 48 % | ★       |
| Algeria      | 7.574 km  | 18 % | ★       |
| Andorra      | 7.645 km  | 28 % | ★       |
| Argentina    | 16.320 km | 41 % | ★       |
| Azerbaijan   | 4.120 km  | 14 % | ★       |
| Bahrain      | 8.501 km  | 73 % | ★       |
| Bangladesh   | 6.511 km  | 27 % | ★       |
| Bolivia      | 17.560 km | 27 % | ★       |
| Burkina Faso | 14.987 km | 27 % | ★       |
| England      | 1.426 km  | 11 % | ★       |

## Other Proxy Tools:

### Burp Suite

<https://www.portswigger.net>

### Tor

<https://www.torproject.org>

### CCProxy

<https://www.youngzsoft.net>

### Hotspot Shield

<https://www.hotspotshield.com>

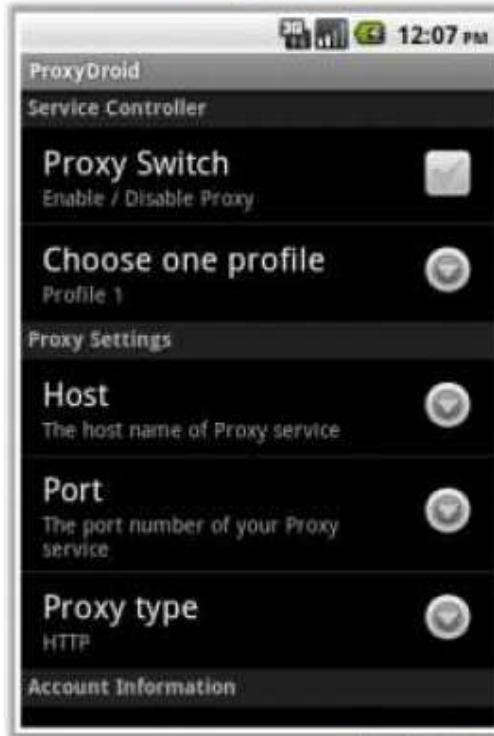
# Proxy Tools for Mobile



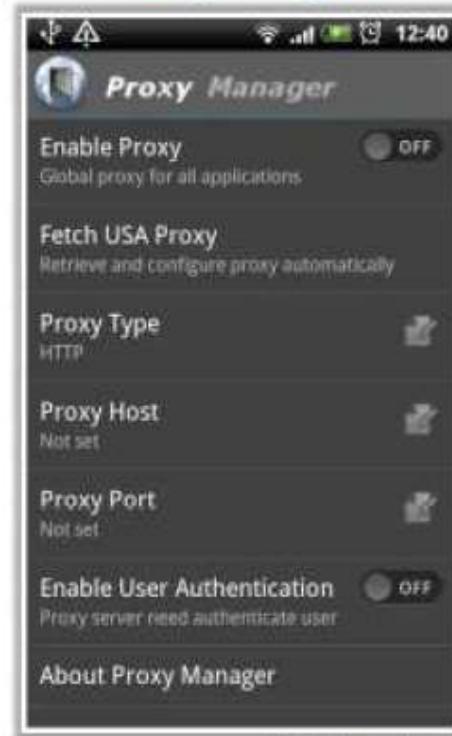
**Shadowsocks**



**ProxyDroid**



**Proxy Manager**



<https://shadowsocks.org>

<https://github.com>

<https://play.google.com>

# Anonymizers ✓



- An anonymizer **removes** all identity information from the user's computer while the user surfs the Internet
- Anonymizers make activity on the Internet **untraceable**
- Anonymizers allow you to **bypass Internet** censors



## Why use an Anonymizer?

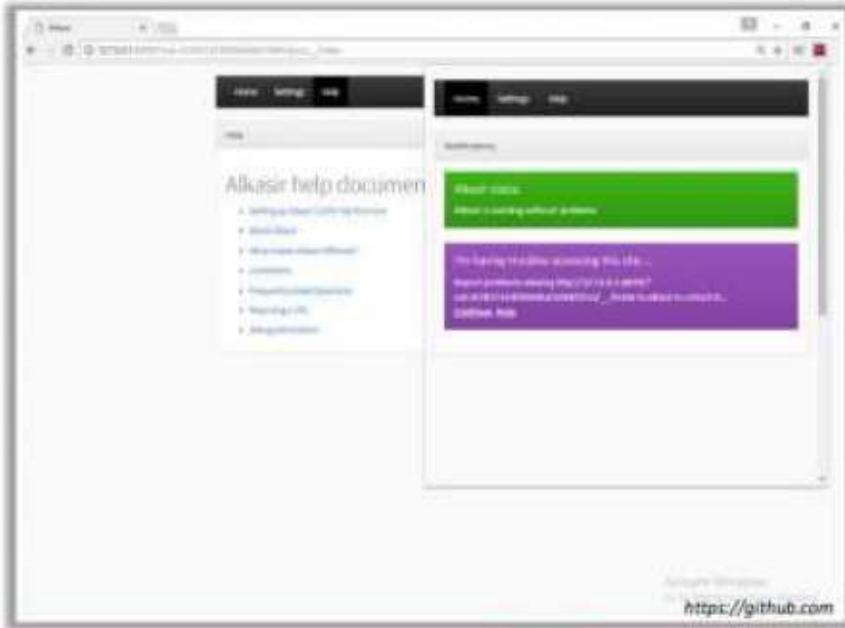
- ① Privacy and anonymity
- ② Protection against online attacks
- ③ Access restricted content
- ④ Bypass IDS and Firewall rules



# Censorship Circumvention Tools: Alkasir and Tails

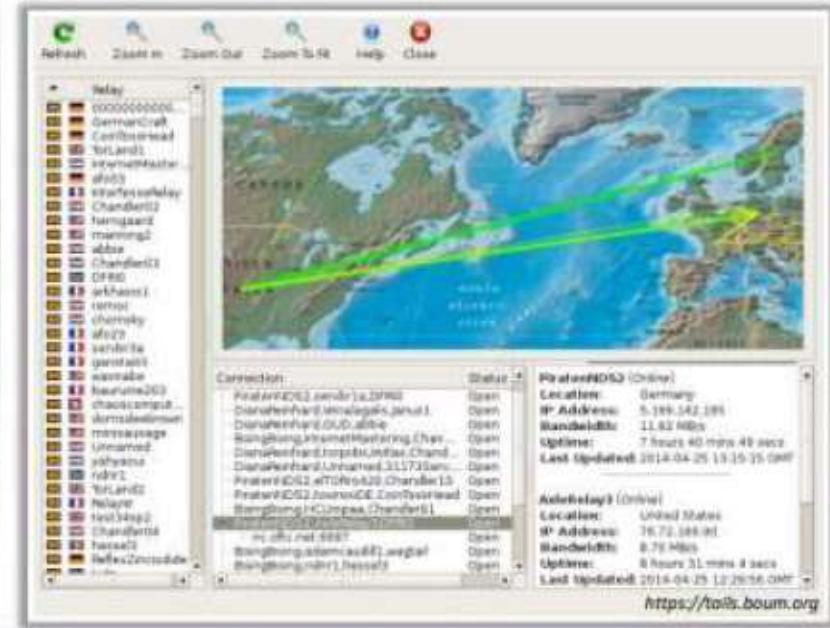
## Alkasir

Alkasir is a **cross-platform**, open-source, and robust website censorship circumvention tool that also **maps censorship patterns** around the world



## Tails

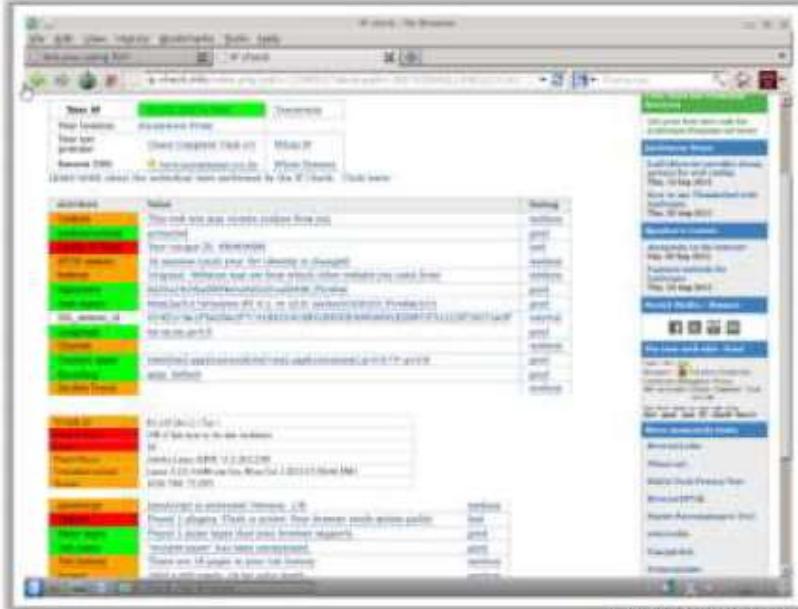
Tails is a **live operating system** that a user can start on any computer from a DVD, USB stick, or SD card



# Anonymizers

## Whonix

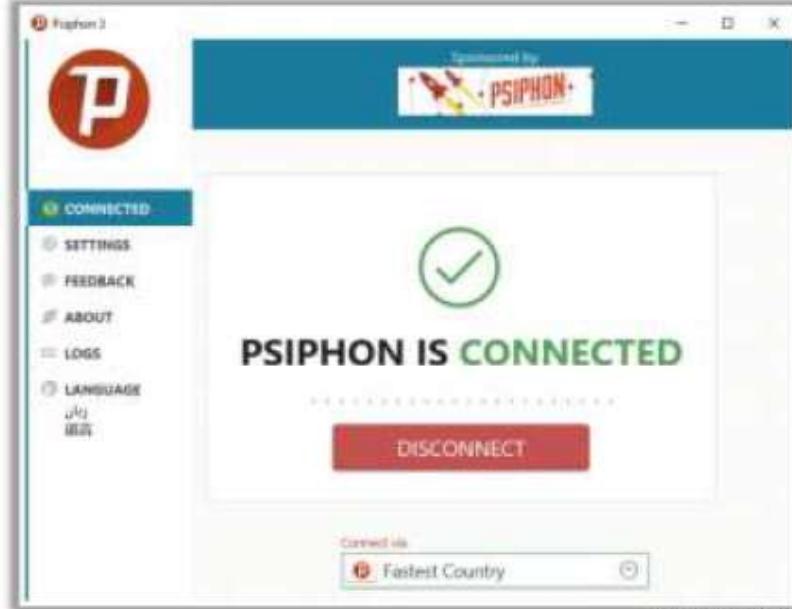
Whonix is a **desktop operating system** designed for advanced security and privacy



<https://www.whonix.org>

## Psiphon

Psiphon is an open-source anonymizer software that allows attackers to surf the internet through a **secure proxy**

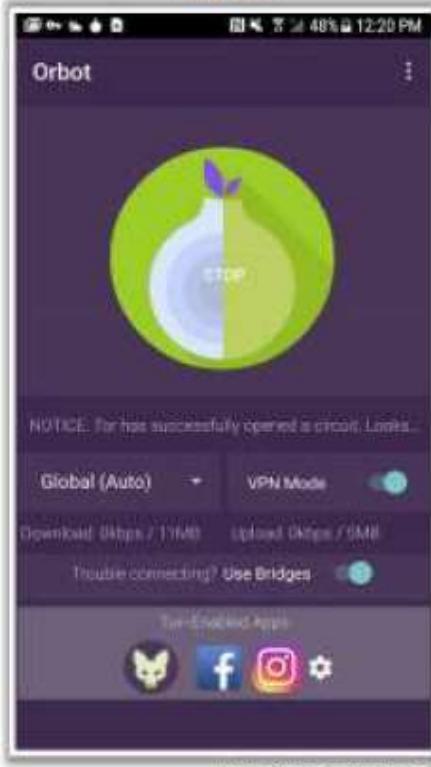


<https://psiphon.ca>

# Anonymizers for Mobile



Orbot



Psiphon



OpenDoor



# Module Flow



**1**

**Network Scanning Concepts**

**2**

**Scanning Tools**

**3**

**Host Discovery**

**4**

**Port and Service Discovery**

**5**

**OS Discovery (Banner Grabbing/  
OS Fingerprinting)**

**6**

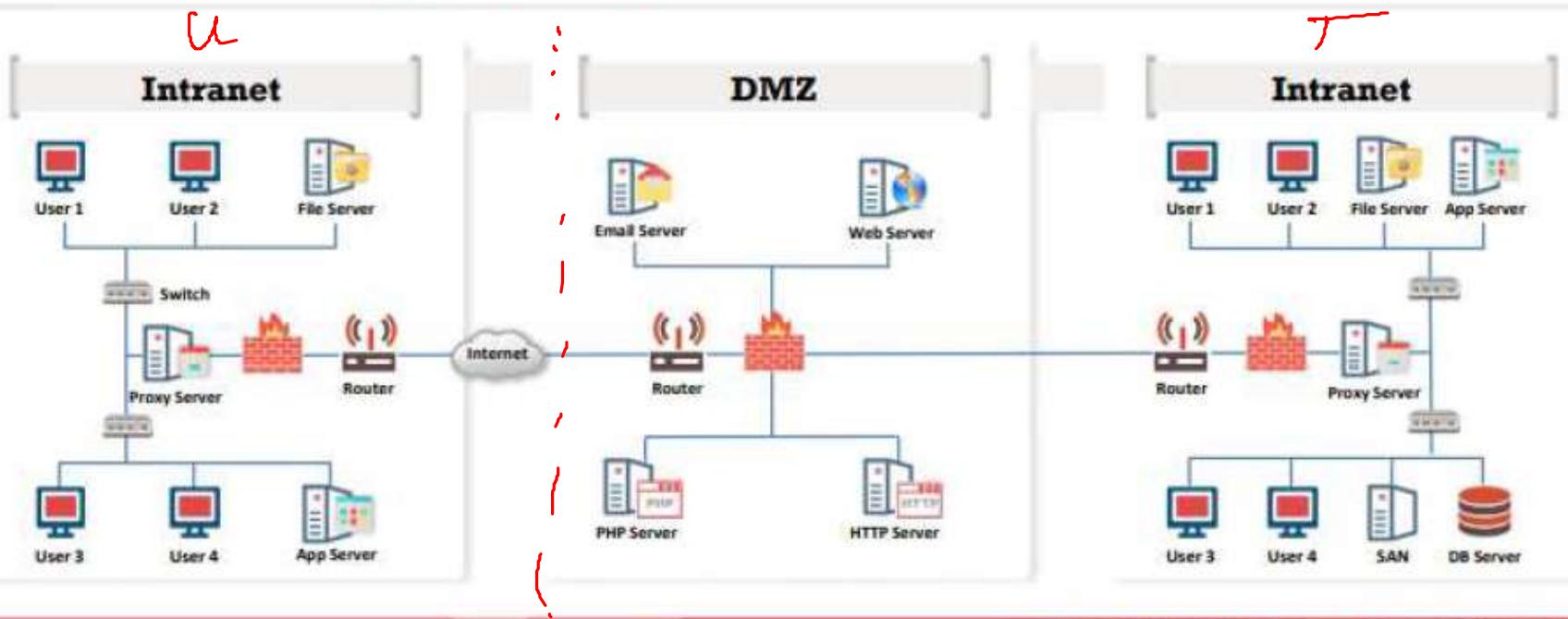
**Scanning Beyond IDS and Firewall**

**7**

**Draw Network Diagrams**

# Drawing Network Diagrams

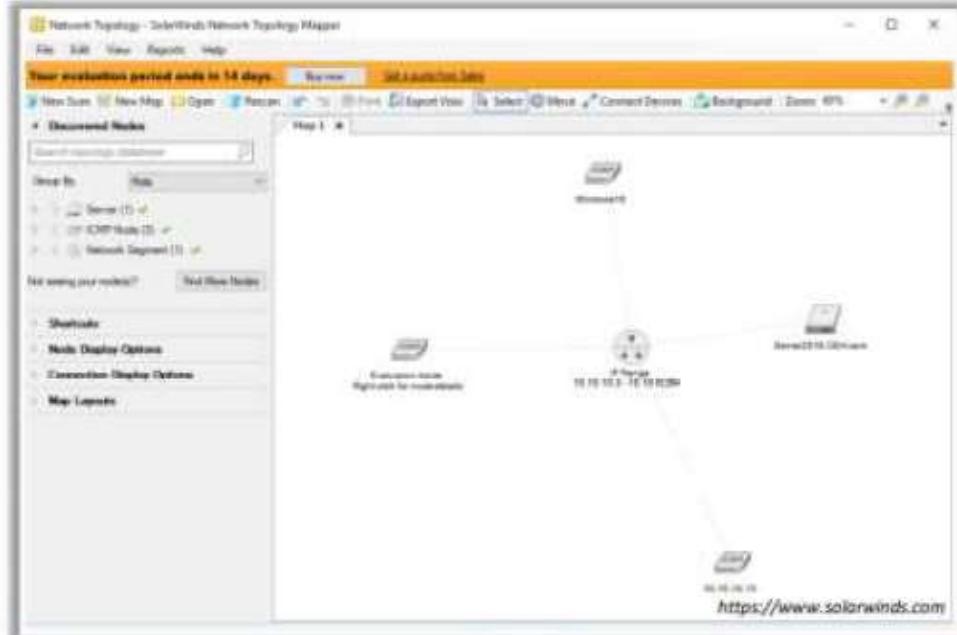
- A diagram of a target network provides an attacker with valuable information about the **network and its architecture**
- Network diagrams show **logical or physical paths** to a potential target



# Network Discovery and Mapping Tools

## Network Topology Mapper

- Network Topology Mapper discovers a network and produces a comprehensive network diagram
- It displays in-depth connections such as OSI Layer 2 and Layer 3 topology data



## OpManager

<https://www.manageengine.com>



## The Dude

<https://mikrotik.com>



## NetSurveyor

<http://nutsaboutnets.com>



## NetBrain

<https://www.netbraintech.com>



## Spiceworks Network Mapping Tool

<https://www.spiceworks.com>

# Network Discovery Tools for Mobile



Scany



<http://happymagenta.com>

Network Analyzer



<https://play.google.com>

PortDroid Network Analysis



<https://play.google.com>

## 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
  - Drawing diagrams of target networks and their significance in providing valuable information about a network and its architecture to an attacker
- ❑ 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