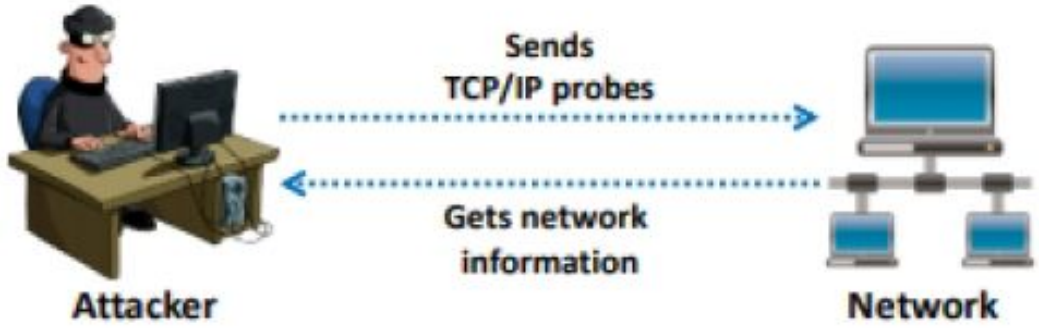
Overview of Network Scanning

- 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

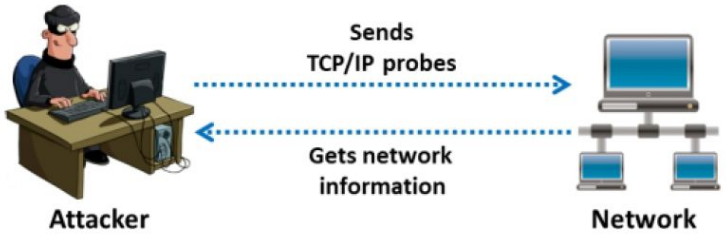
- To discover live hosts, IP address, and open ports of live hosts
- To discover operating systems and system architecture
- To discover services running on hosts
- To discover vulnerabilities in live hosts

Types of scanning

1. Port scanning
2. Network scanning
3. Vulnerability scanning

Objectives

To find Live devices, OS, IPs in use
ports(open/closed)
Vulnerabilities



LOOK FOR LIVE SYSTEMS

CHECK FOR OPEN PORTS

EVADE IDS

BANNER GRABBING

VULNERABILITY SCANS

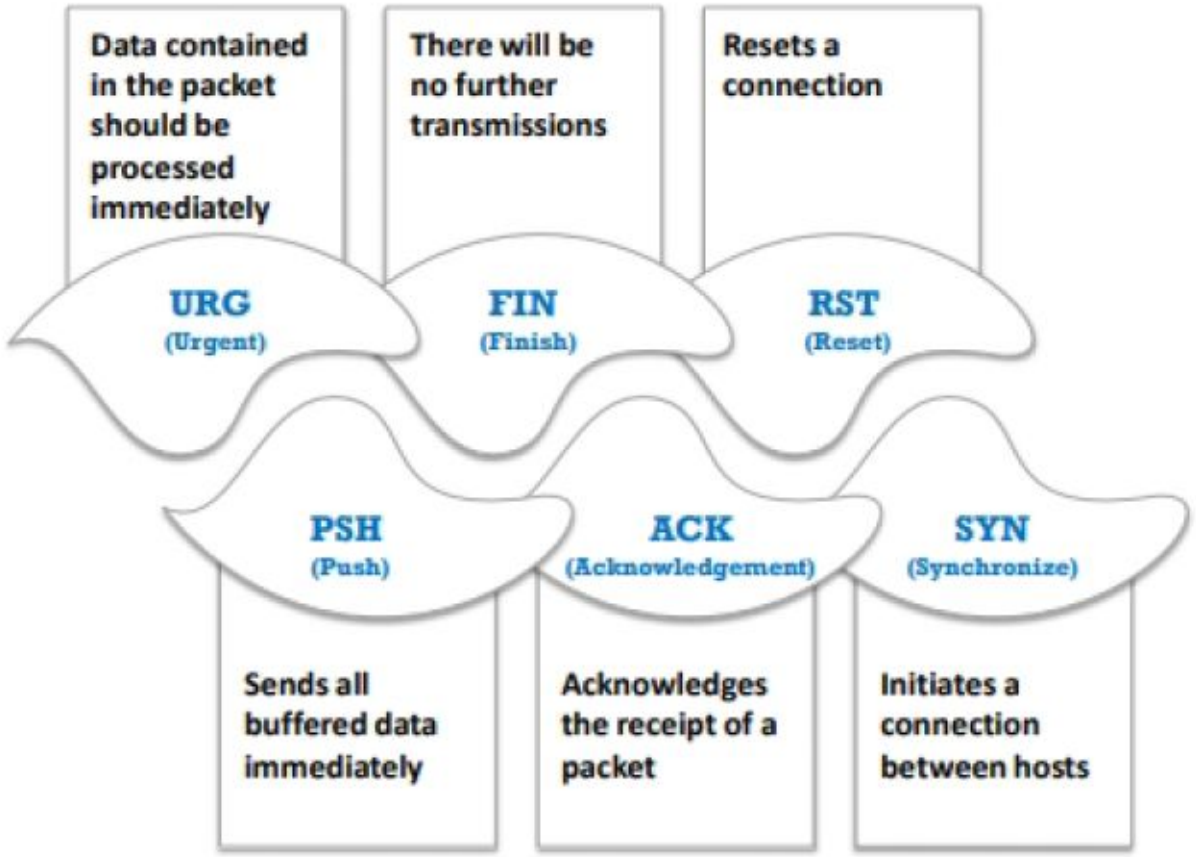
BUILD NETWORK DIAGRAMS

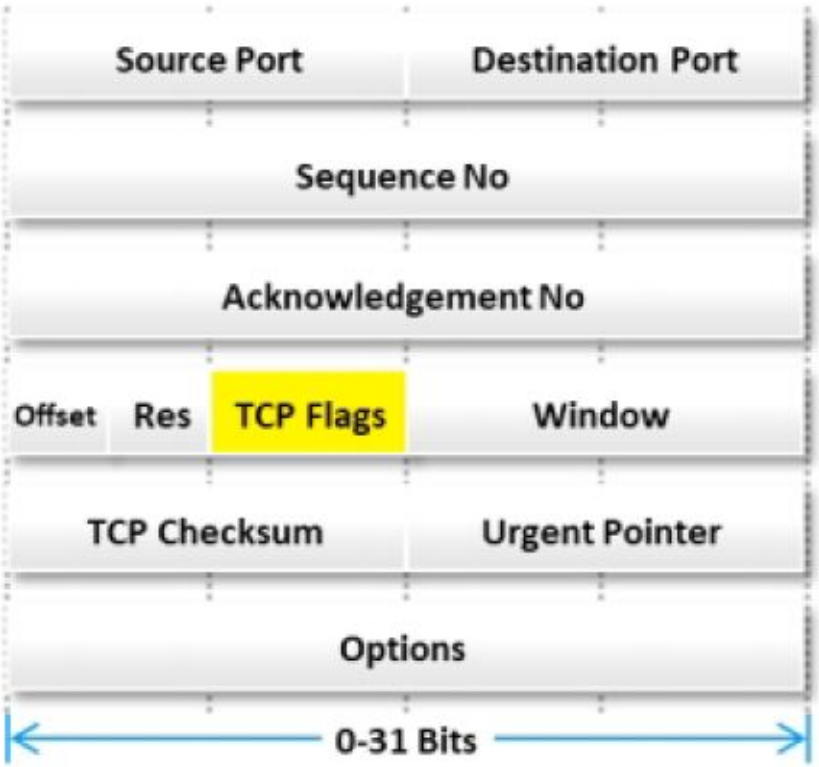
USE PROXIES

- **Port Scanning** – Lists the open ports and services. Port scanning is the process of checking the services running on the target computer by sending a sequence of messages in an attempt to break in. Port scanning involves connecting to or probing TCP and UDP ports of the target system to determine whether the services are running or are in a listening state. The listening state provides information about the OS and the application currently in use. Sometimes, active services that are listening may allow unauthorized users to misconfigure systems or to run software with vulnerabilities.
- **Network Scanning** – Lists the active hosts and IP addresses. Network scanning is a procedure for identifying active hosts on a network, either to attack them or assess the security of the network.
- **Vulnerability Scanning** – Shows the presence of known weaknesses. Vulnerability scanning is a method for checking whether a system is exploitable by identifying its vulnerabilities. A vulnerability scanner consists of a scanning engine and a catalog. The catalog includes a list of common files with known vulnerabilities and common exploits for a range of servers. A vulnerability scanner may, for example, look for backup files or directory traversal exploits. The scanning engine maintains logic for reading the exploit list, transferring the request to the web server, and analyzing the requests to ensure the safety of the server. These tools generally target vulnerabilities that secure host configurations can fix easily through updated security patches and a clean web document.

A thief who wants to break into a house looks for access points such as doors and windows. These are usually the house's points of vulnerability, as they are easily accessible. When it comes to computer systems and networks, ports are the doors and windows of a system that an intruder uses to gain access. A general rule for computer systems is that the greater the number of open ports on a system, the more vulnerable is the system. However, there are cases in which a system with fewer open ports than another machine presents a much higher level of vulnerability.

TCP Communication Flags





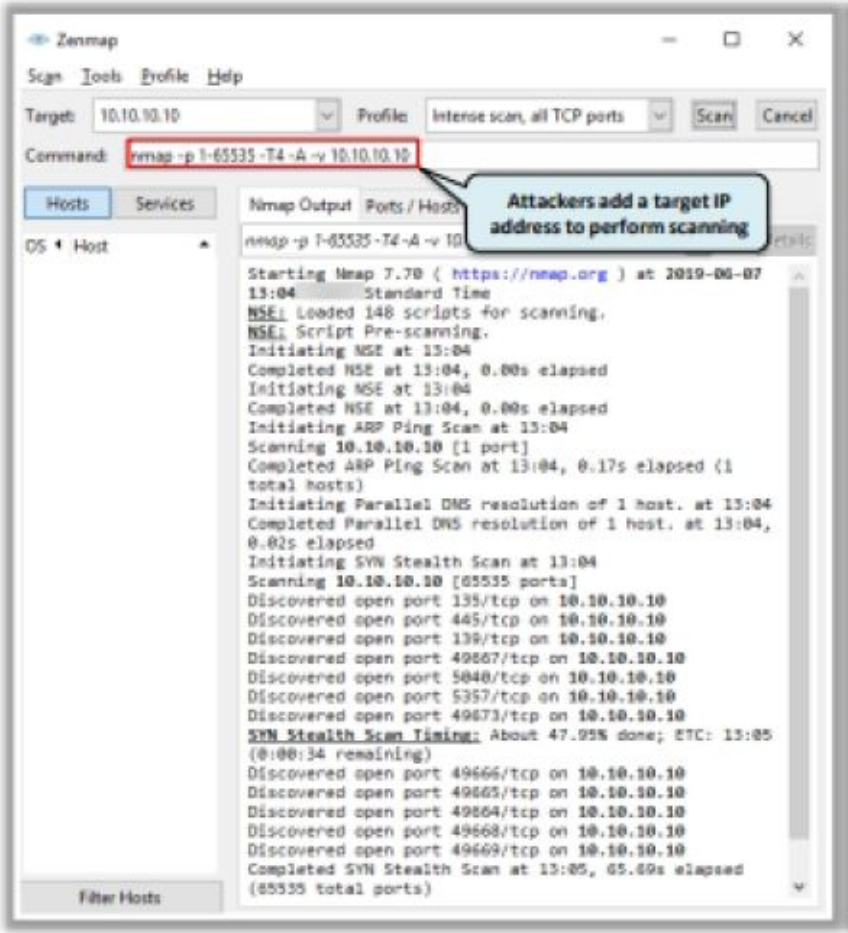
Standard TCP communications are controlled by flags in the TCP packet header

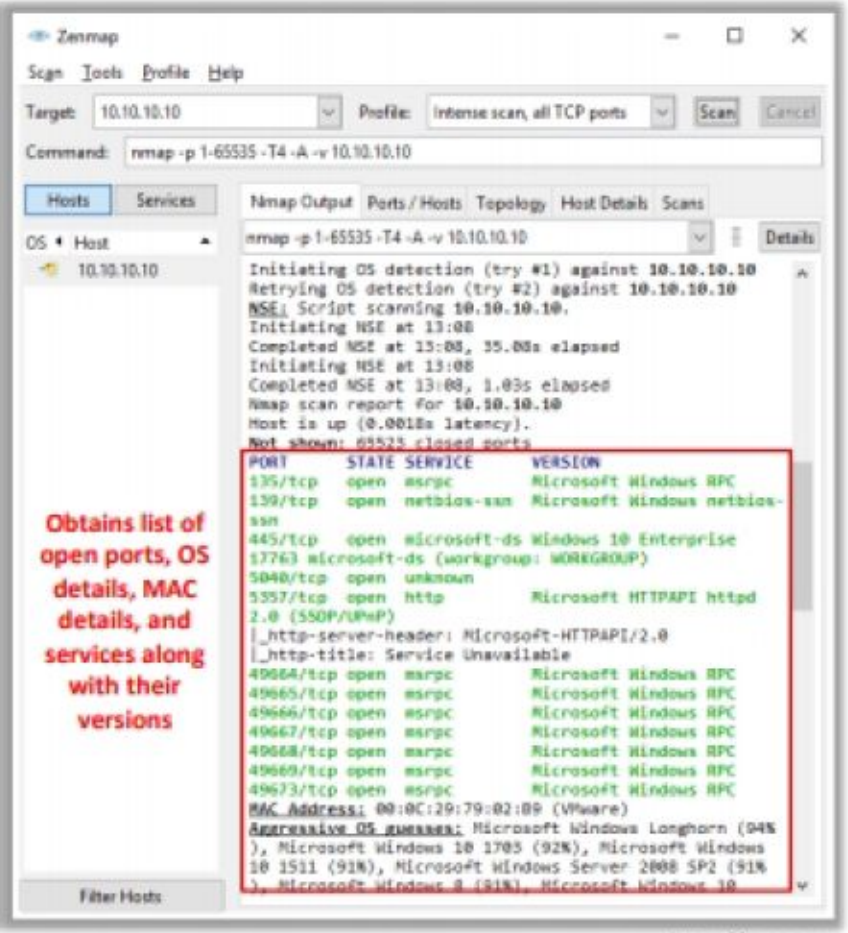
Tools used for scanning networks

- Nmap / zenmap
- Hping3/hping2
- Metasploit
- Netscantools pro

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**





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

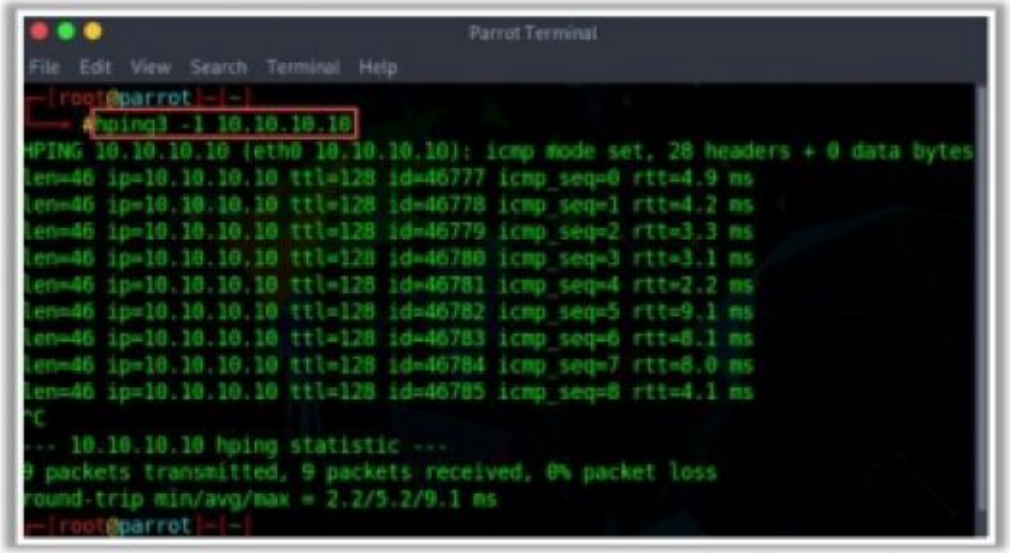
https://nmap.org

Scanning Tools: Hping2/Hping3

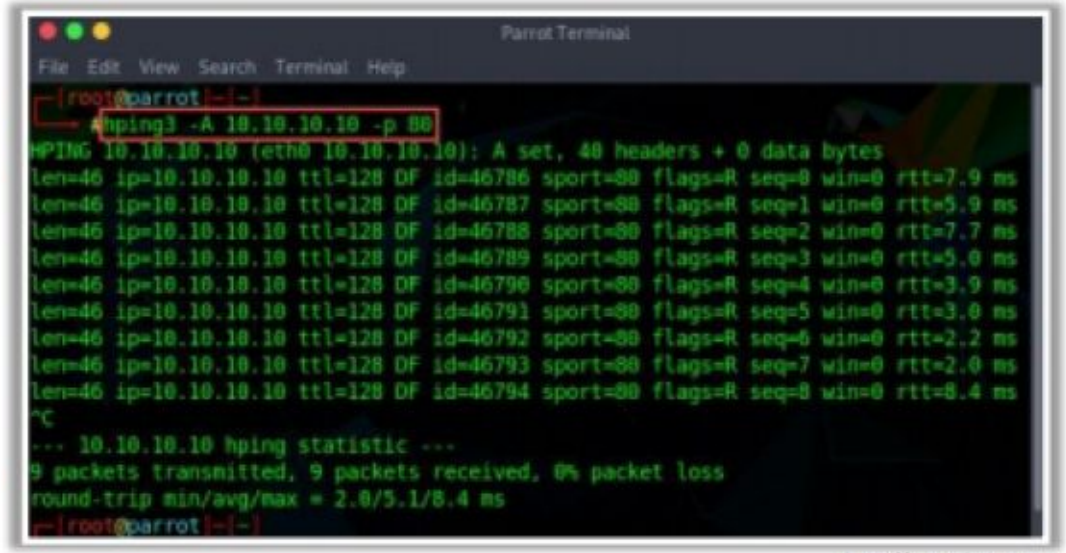
1. Command line **network scanning** and **packet crafting** tool for the TCP/IP protocol

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



ACK Scanning on port 80



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 -8 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-dest -I eth0
```

Intercept all traffic containing HTTP signature

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

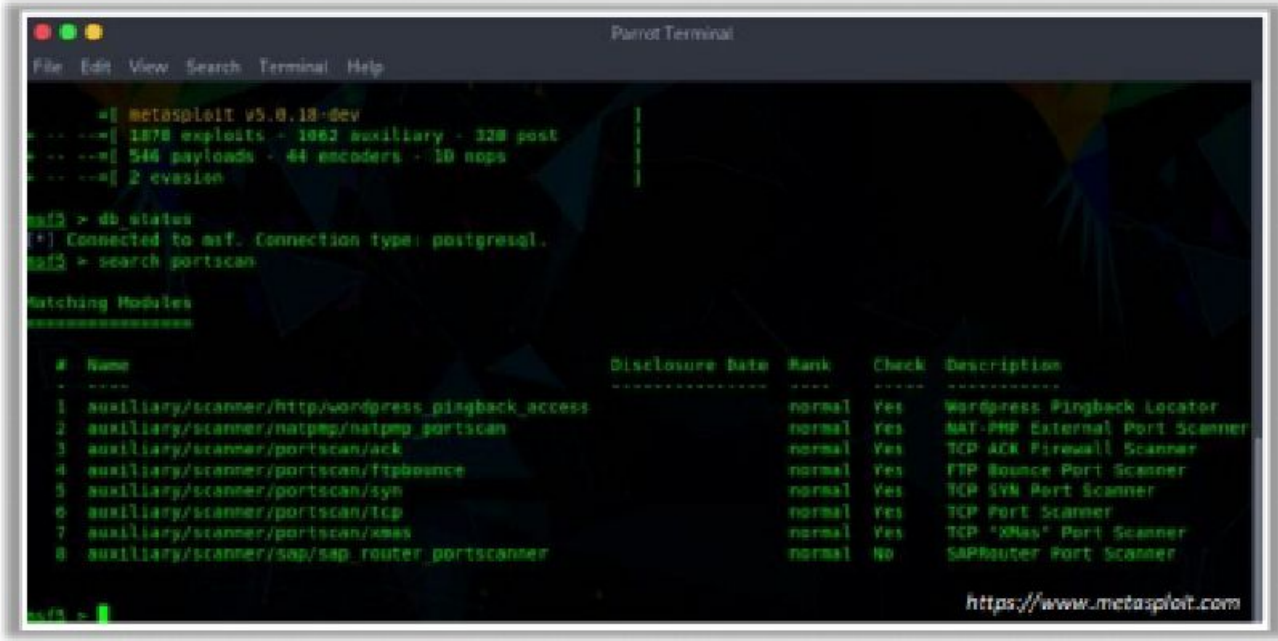
SYN flooding a victim

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

Scanning Tools

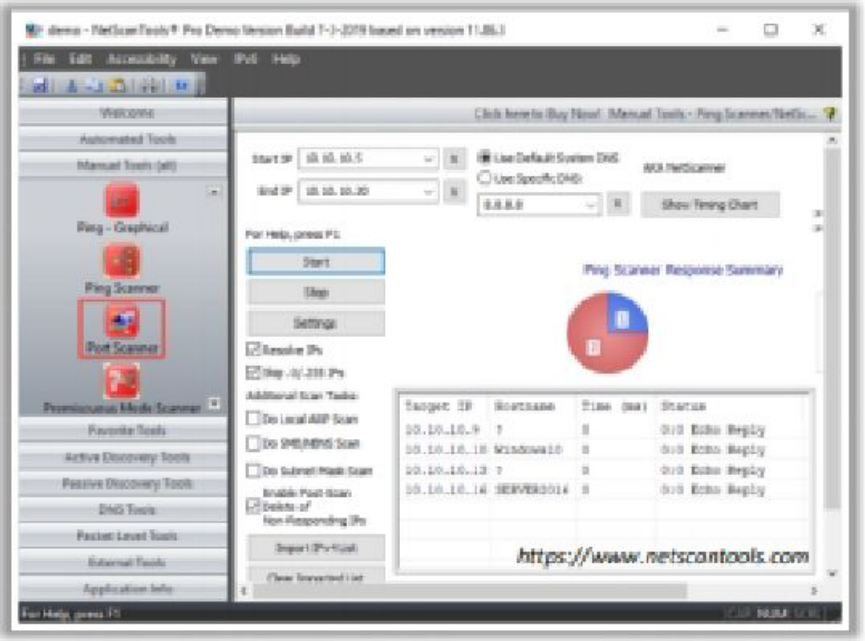
Metasploit

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



NetScanTools Pro

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

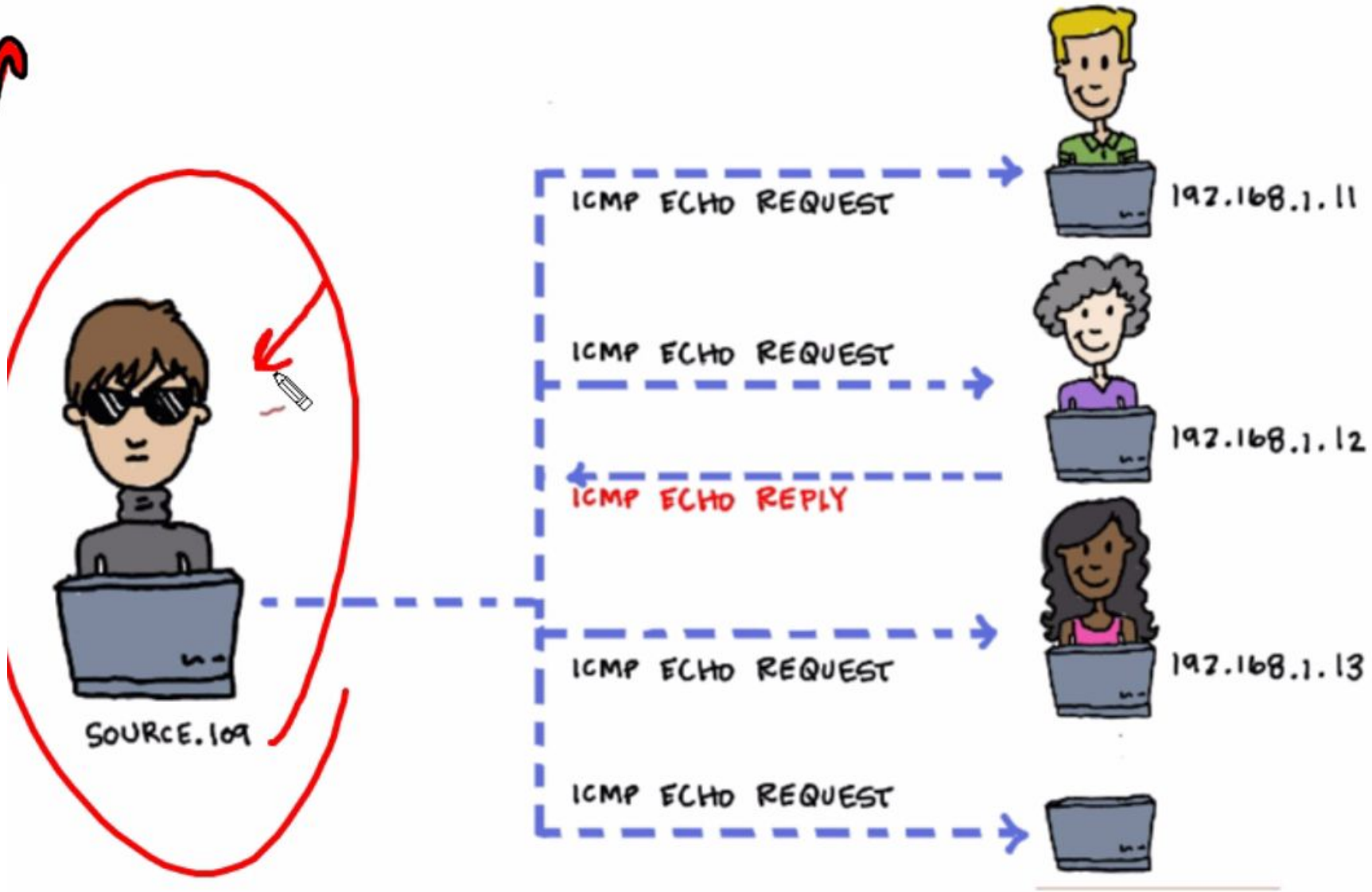


Other Scanning Tools:

- **Unicornscan** — *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*

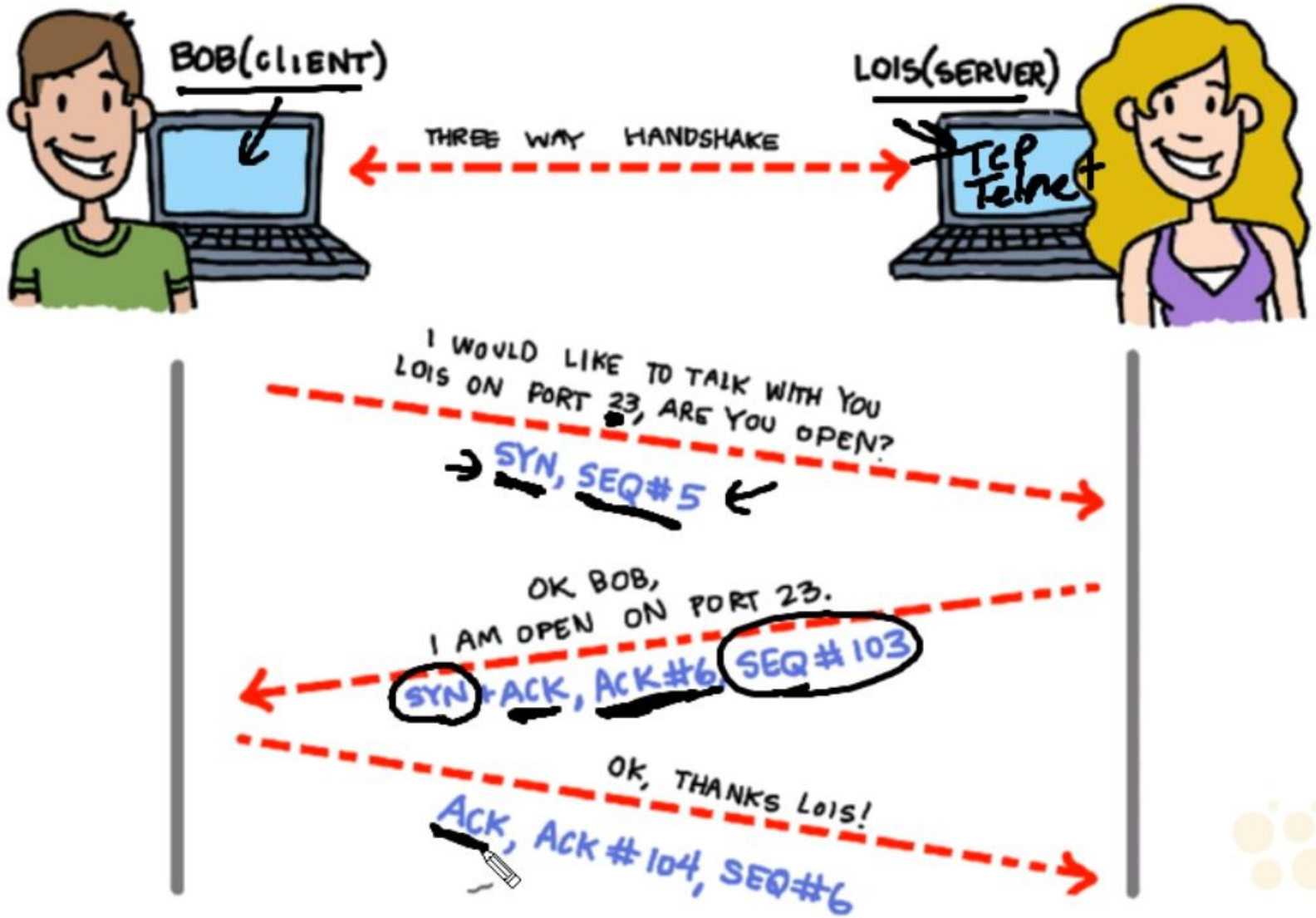
LOOK FOR LIVE SYSTEMS

nmap –sn subnet



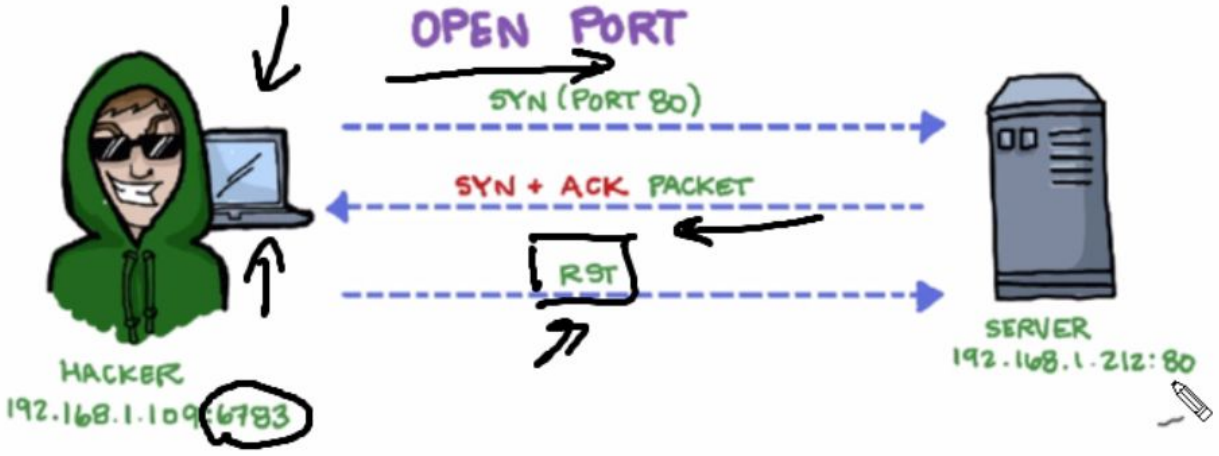
CHECK FOR OPEN PORTS

By Knocking at the Door

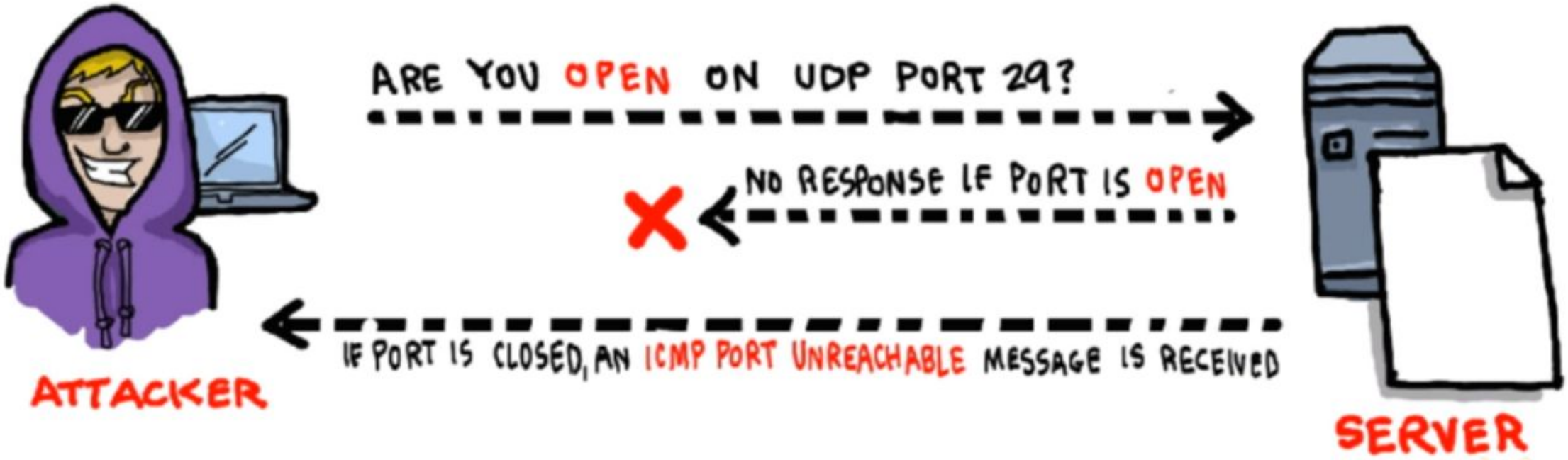


TCP SCAN

By Knocking at the Door

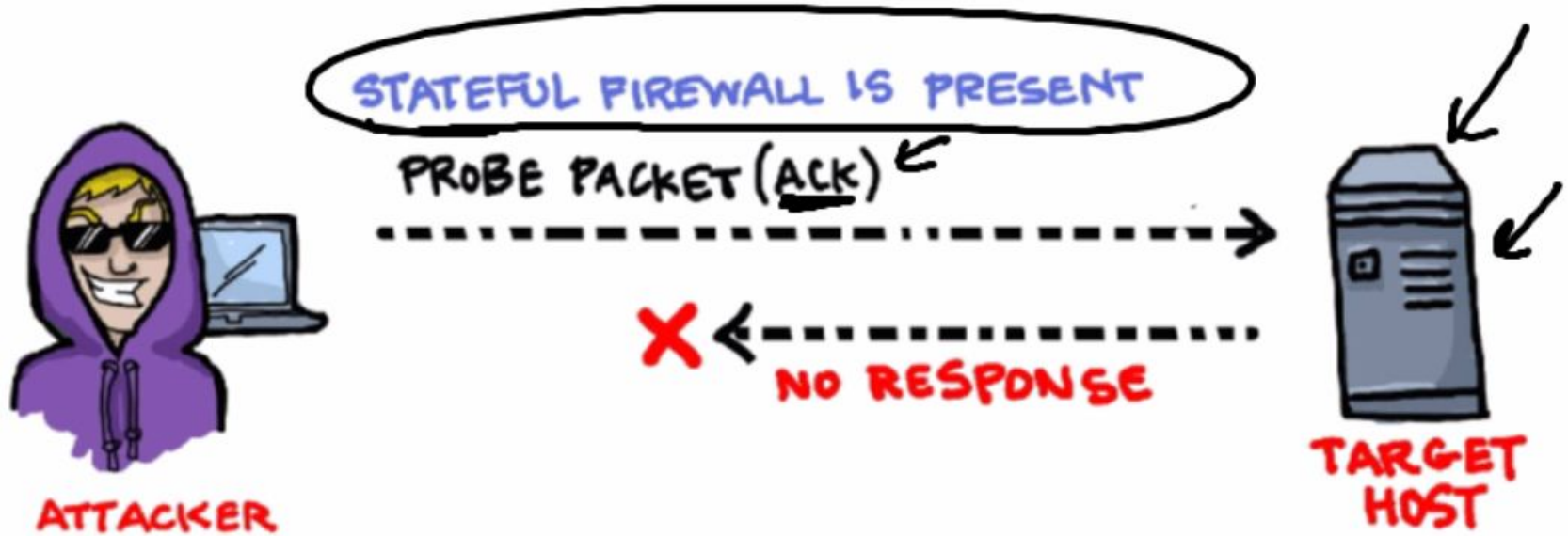


UDP SCAN


ARE YOU OPEN ON UDP PORT 29?
NO RESPONSE IF PORT IS OPEN
IF PORT IS CLOSED, AN ICMP PORT UNREACHABLE MESSAGE IS RECEIVED
ATTACKER
SERVER

CHECK FOR OPEN PORTS

By Knocking at the Door



nmap usage

Nmap –sn 192.168.1.0/24 - it will scan entire subnet
Nmap –sC -sV ip_addr - it will scan for open ports
-sV - scan for service version
-sC - default script scan
-p to specify port number
-A to aggresive scan

Hping3

Scan	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 -8 50-56 -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-dest -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`

hping3 -c 15000 -d 120 -S -w 64 -p 80 --flood --rand-source 192.168.1.159

- **ICMP ping**

  Ex. `hping3 -1 10.0.0.25`

  Hping performs an ICMP ping scan by specifying the argument -1 in the command line. You may use --ICMP or -1 as the argument in the command line. By issuing the above command, hping sends an ICMP echo request to 10.0.0.25 and receives an ICMP reply similarly to a ping utility.

- **ACK scan on port 80**

  Ex. `hping3 -A 10.0.0.25 -p 80`

  Hping can be configured to perform an ACK scan by specifying the argument -A in the command line. Here, you set the ACK flag in the probe packets and perform the scan. You perform this scan when a host does not respond to a ping request. By issuing this command, Hping checks if a host is alive on a network. If it finds a live host and an open port, it returns an RST response.

- **UDP scan on port 80**

  Ex. `hping3 -2 10.0.0.25 -p 80`

  Hping uses TCP as its default protocol. Using the argument -2 in the command line specifies that Hping operates in the UDP mode. You may use either --udp or -2 as the argument in the command line.

- **Collecting Initial Sequence Number**

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

  Using the argument -Q in the command line, Hping collects all the TCP sequence numbers generated by the target host (192.168.1.103).

- **Firewalls and Timestamps**

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

  Many firewalls drop those TCP packets that do not have the TCP Timestamp option set. By adding the --tcp-timestamp argument in the command line, you can enable the TCP timestamp option in Hping and try to guess the timestamp update frequency and uptime of the target host (72.14.207.99).

- **SYN scan on port 50-60**

  Ex. `hping3 -8 50-60 -S 10.0.0.25 -V`

  Using the argument -8 or --scan in the command line, you are operating Hping in the scan mode to scan a range of ports on the target host. Adding the argument -S allows you to perform a SYN scan.

  Therefore, the above command performs a SYN scan on ports 50–60 on the target host.

- **FIN, PUSH and URG scan on port 80**

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

  By adding the arguments –F, –P, and –U in the command line, you are setting FIN, PUSH, and URG packets in the probe packets. By issuing this command, you are performing FIN, PUSH, and URG scans on port 80 on the target host (10.0.0.25). If port 80 is open, you will not receive a response. If the port is closed, Hping will return an RST response.

- **Scan entire subnet for live host**

  Ex. `hping3 -1 10.0.1.x --rand-dest -I eth0`

  By issuing this command, Hping performs an ICMP ping scan on the entire subnet 10.0.1.x; in other words, it sends an ICMP echo request randomly (--rand-dest) to all the hosts from 10.0.1.0 to 10.0.1.255 that are connected to the interface eth0. The hosts whose ports are open will respond with an ICMP reply. In this case, you have not set a port; hence, Hping sends packets to port 0 on all IP addresses by default.

- **Intercept all traffic containing HTTP signature**

  Ex. **`hping3 -9 HTTP -I eth0`**

  The argument -9 will set the Hping to the listen mode. Hence, by issuing the command -9 HTTP, Hping starts listening on port 0 (of all the devices connected in the network to interface eth0), intercepts all the packets containing the HTTP signature, and dumps from the signature end to the packet's end.

  For example, on issuing the command **`hping2 -9 HTTP`**, if Hping reads a packet that contains data **`234-09sdflkjs45-HTTPhello_world`**, it will display the result as **`hello_world`**.

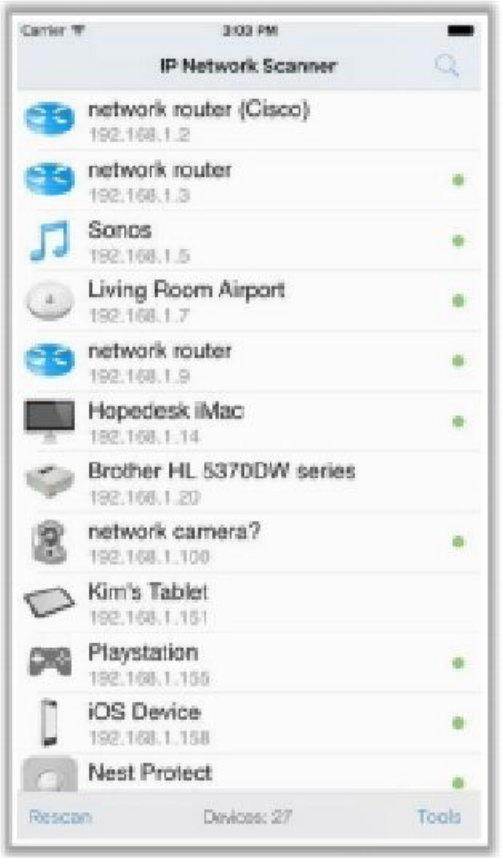
- **SYN flooding a victim**

  Ex. **`hping3 -S 192.168.1.1 -a 192.168.1.254 -p 22 --flood`**

  The attacker employs TCP SYN flooding techniques using spoofed IP addresses to perform a DoS attack.

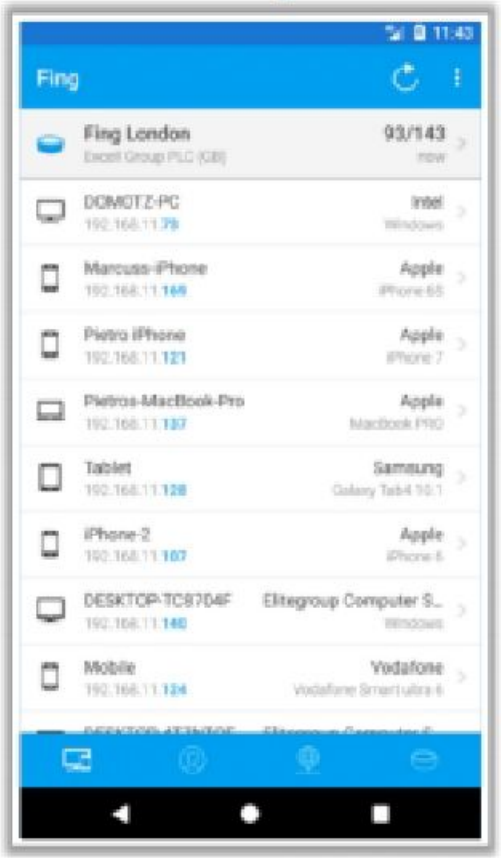
Scanning Tools for Mobile

IP Scanner



https://10base-t.com

Fing



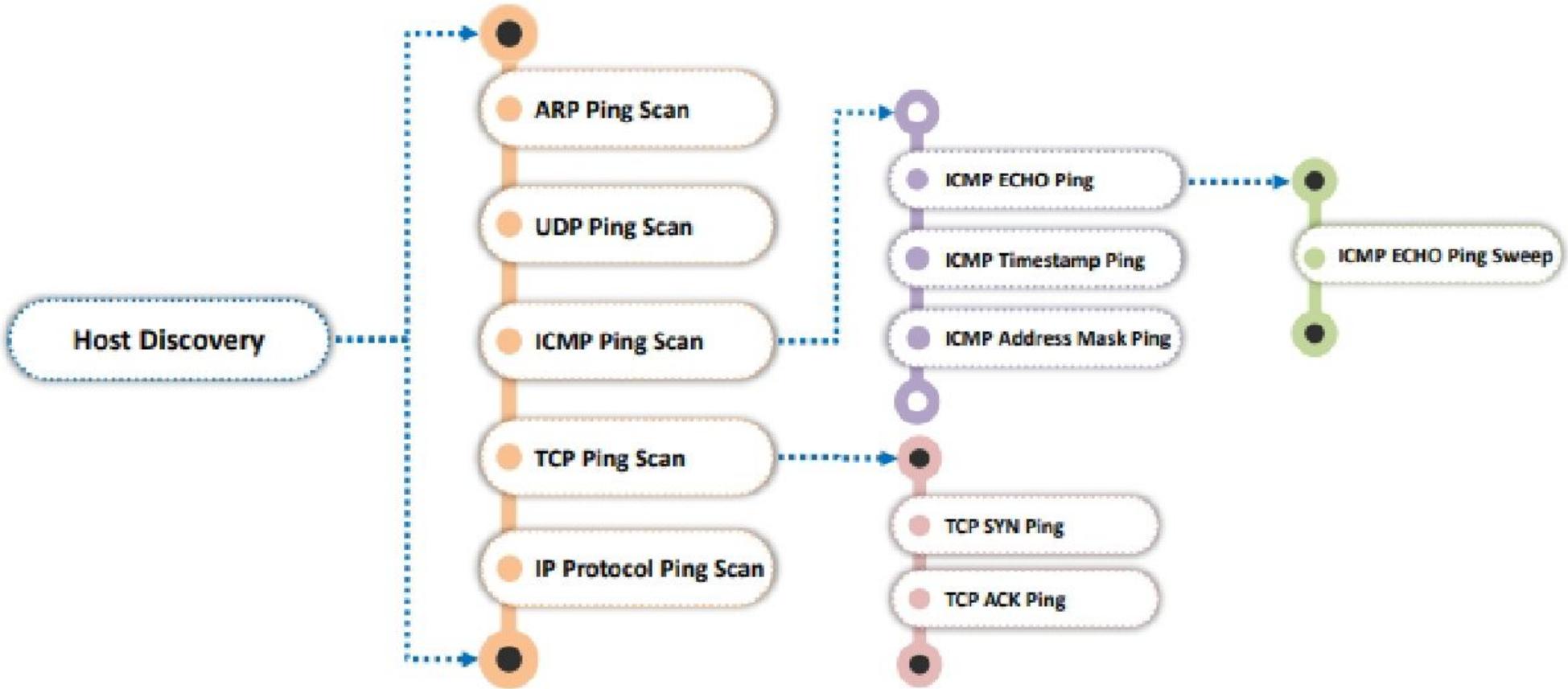
https://www.fing.io

Network Scanner

https://play.google.com

Host Discovery Techniques

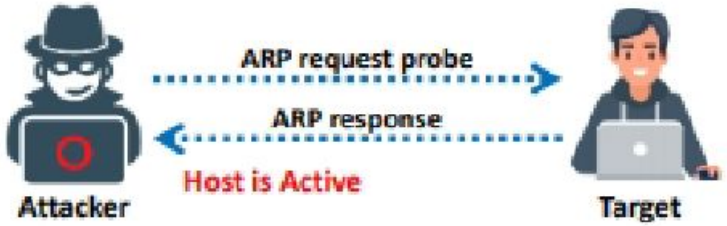
- Host discovery techniques are used to **identify the active/live systems** in the network

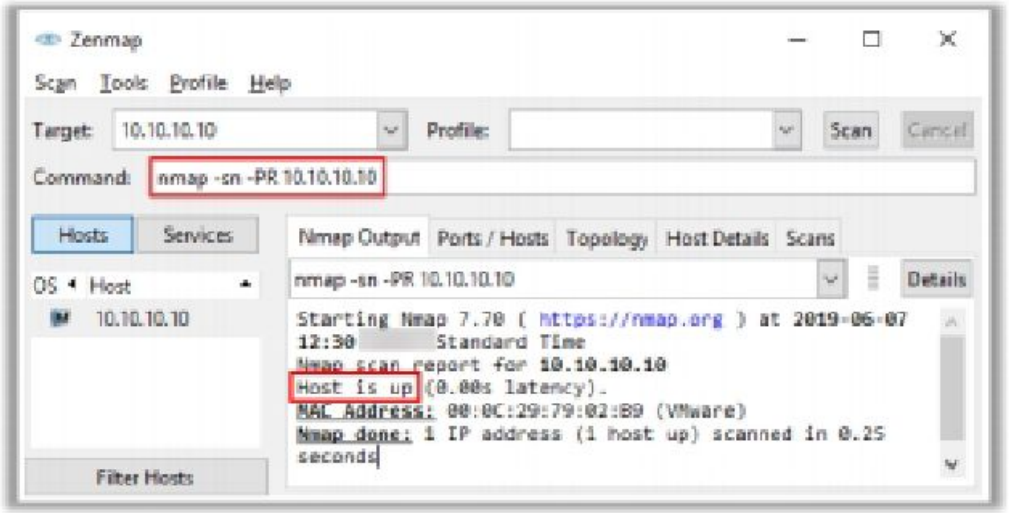


ARP Ping Scan and UDP Ping Scan

ARP Ping Scan

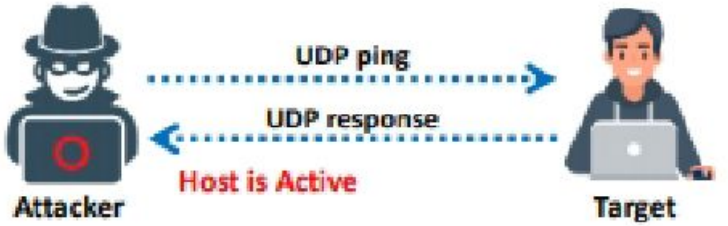
- Attackers send **ARP request probes** to target hosts, and an **ARP** response indicates that the **host is active**

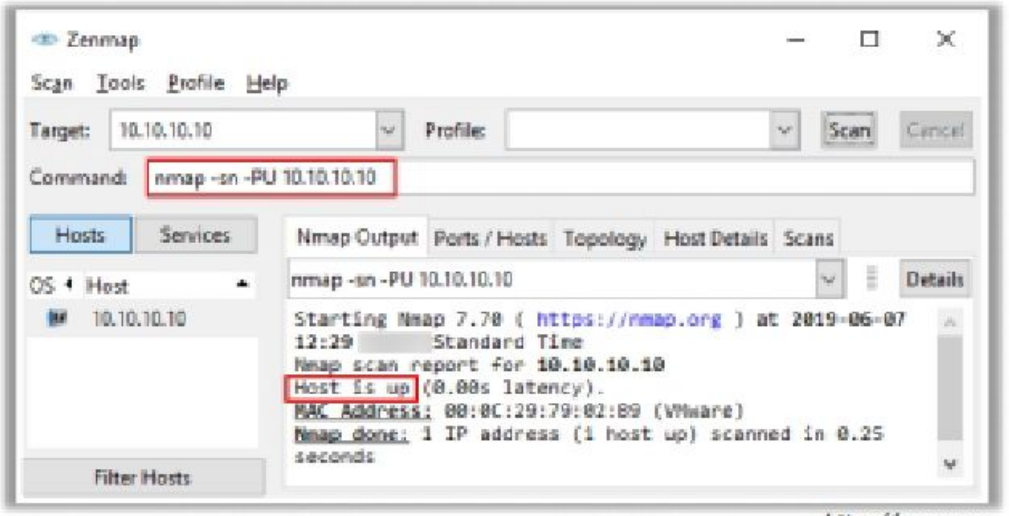




UDP Ping Scan

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



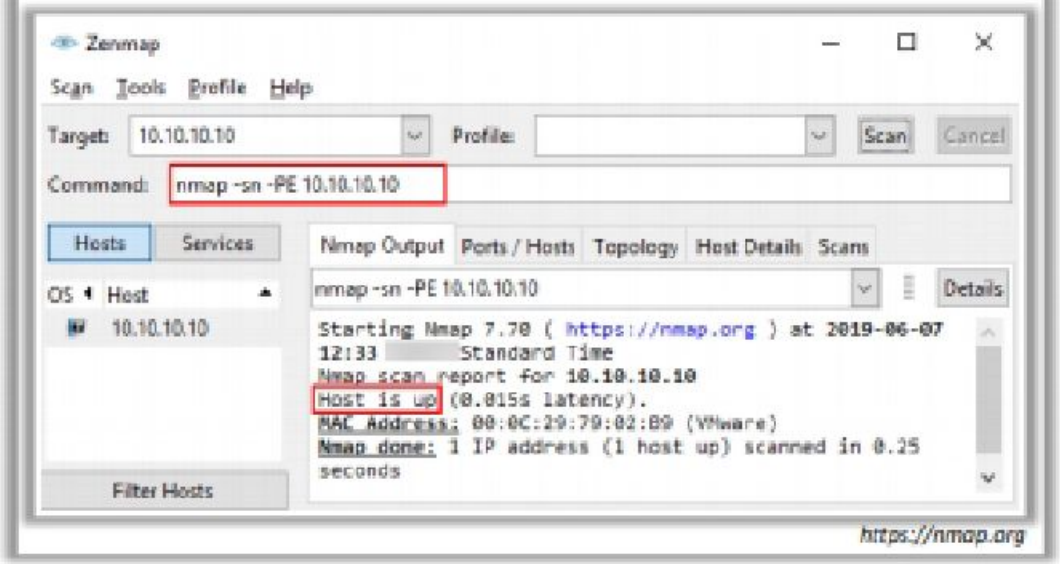


https://nmap.org

ICMP ECHO Ping Scan

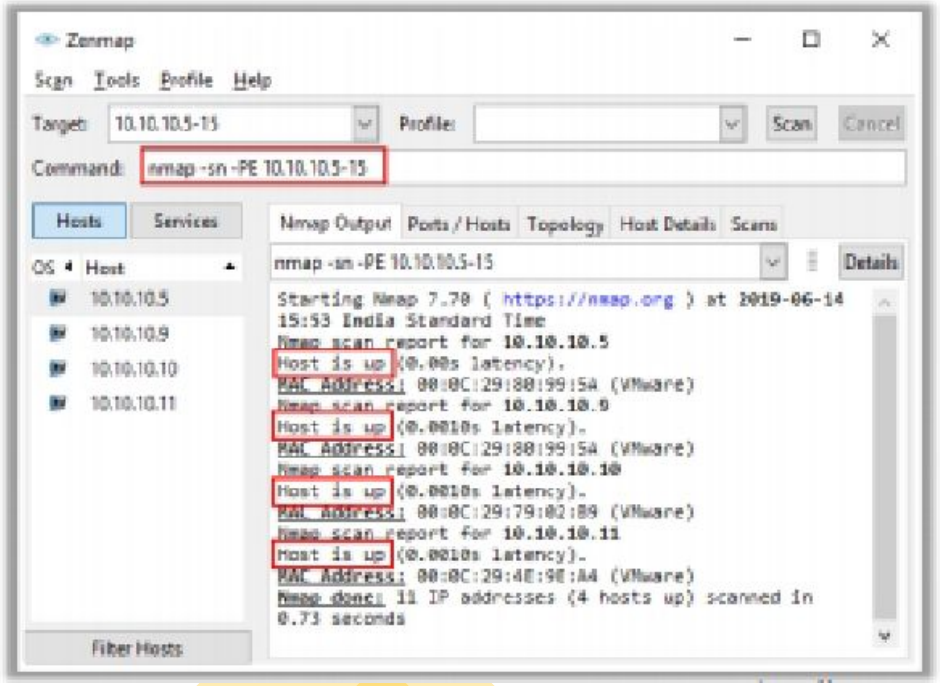
- ICMP ECHO ping scans involve sending **ICMP ECHO requests** to a host. If the host is live, it will return an ICMP ECHO reply
- This scan is useful for **locating active devices** or determining if the **ICMP is passing through a firewall**

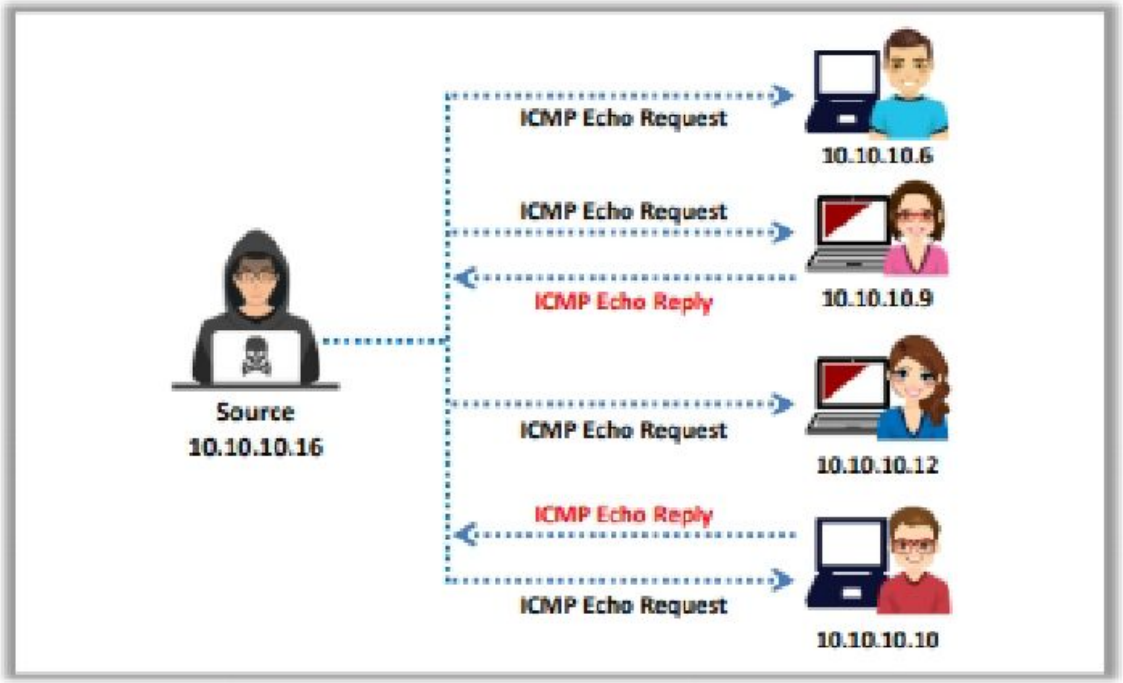
ICMP Echo ping scan output using Zenmap



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





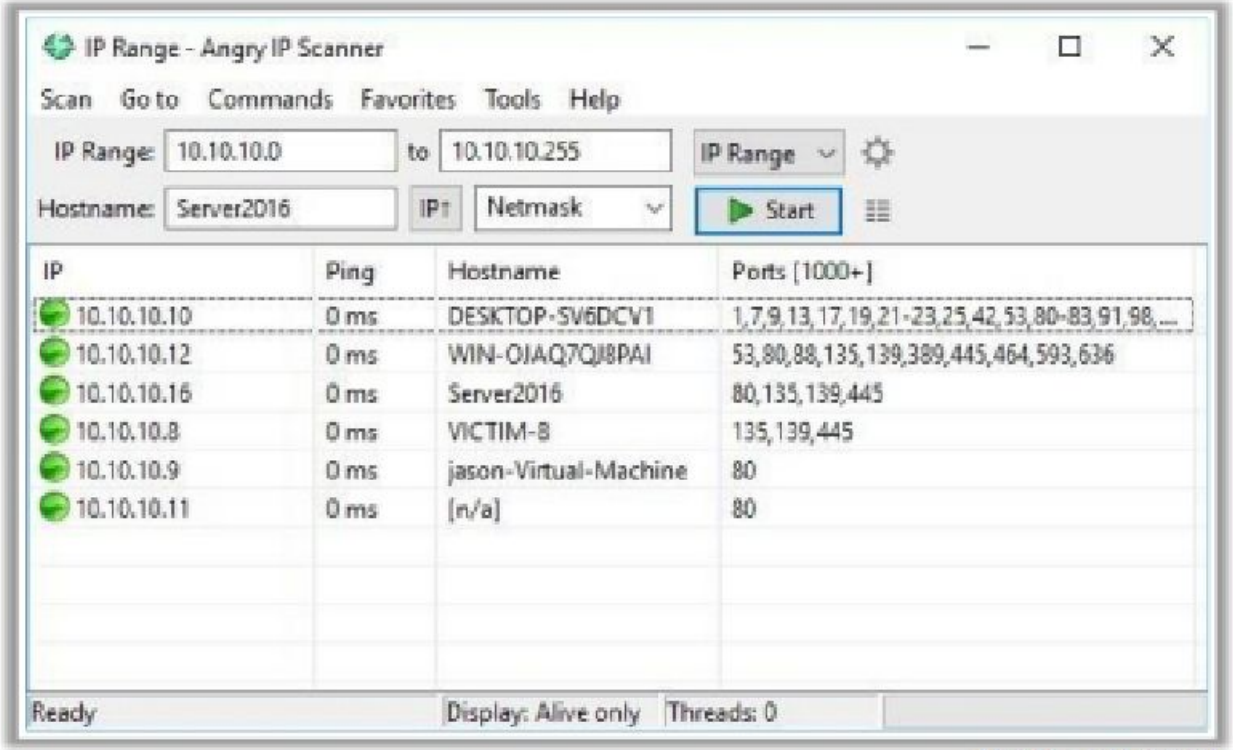
Ping Sweep Tools

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*)



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**

ICMP Timestamp Ping Scan

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

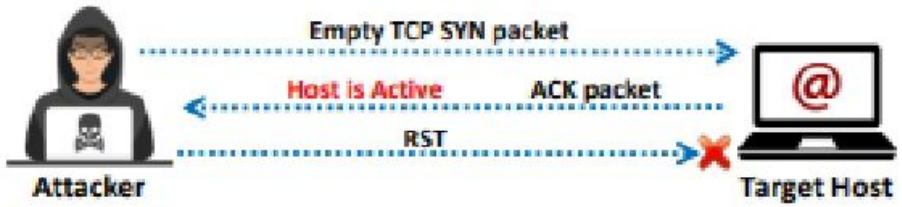
ICMP Address Mask Ping Scan

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

TCP SYN Ping Scan

- Attackers send **empty TCP SYN packets** to a target host, and an **ACK** response means that the **host is active**

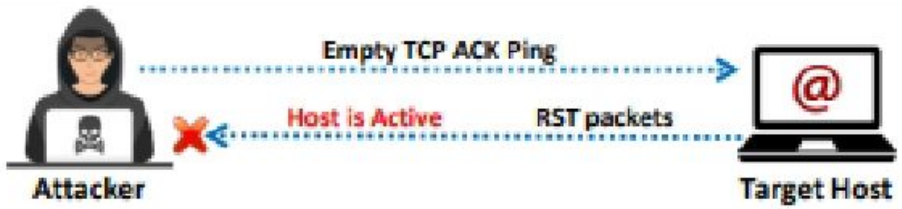
```
# nmap -sn -PS <target IP address>
```



TCP ACK Ping Scan

- Attackers send **empty TCP ACK packets** to a target host, and an **RST** response means that the **host is active**

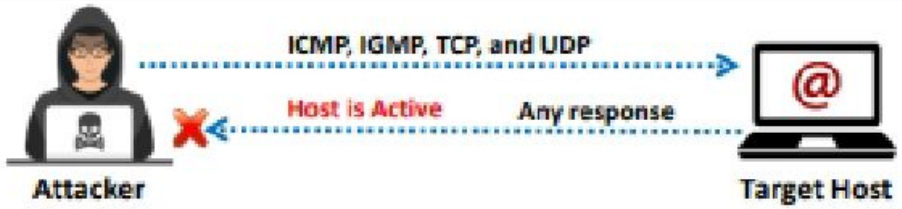
```
# nmap -sn -PA <target IP address>
```



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

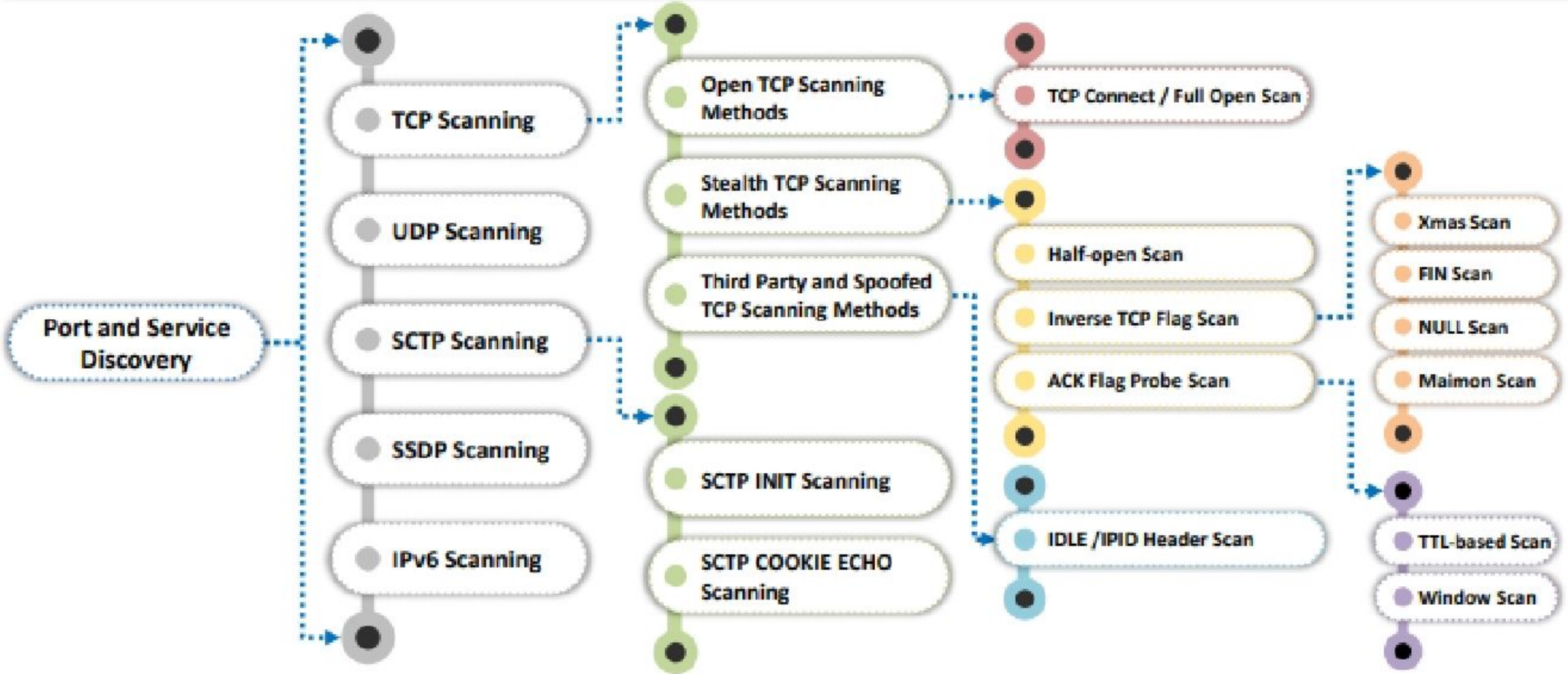
```
# nmap -sn -PO <target IP address>
```



20	FTP File Transfer Protocol (FTP) Data Transfer
21	FTP File Transfer Protocol (FTP) Command Control
22	SSH Secure Shell (SSH)
23	Telnet - Remote login service, unencrypted text messages
25	SMTP Simple Mail Transfer Protocol (SMTP) E-mail Routing
53	DNS Domain Name System (DNS) service
80	HTTP Hypertext Transfer Protocol (HTTP) used in World Wide Web
88	Kerberos - network authentication protocol
110	POP3 Post Office Protocol (POP3) used by e-mail clients to retrieve e-mail from a server
119	NNTP Network News Transfer Protocol (NNTP)
123	NTP Network Time Protocol (NTP)
143	IMAP Internet Message Access Protocol (IMAP) Management of Digital Mail
161	SNMP Simple Network Management Protocol (SNMP)
194	IRC Internet Relay Chat (IRC)
389	LDAP - Lightweight directory access protocol
443	HTTP Secure (HTTPS) HTTP over TLS/SSL
445	SMB over IP (Microsoft DS) (server message block)
3389	RDP (remote desktop protocol)

Port Scanning Techniques

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



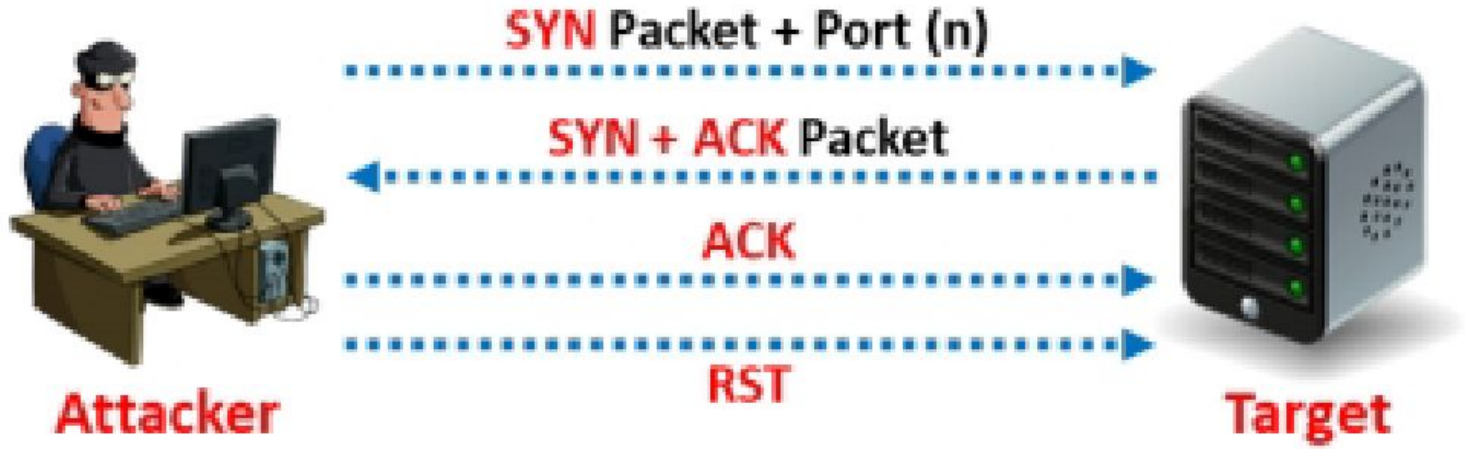
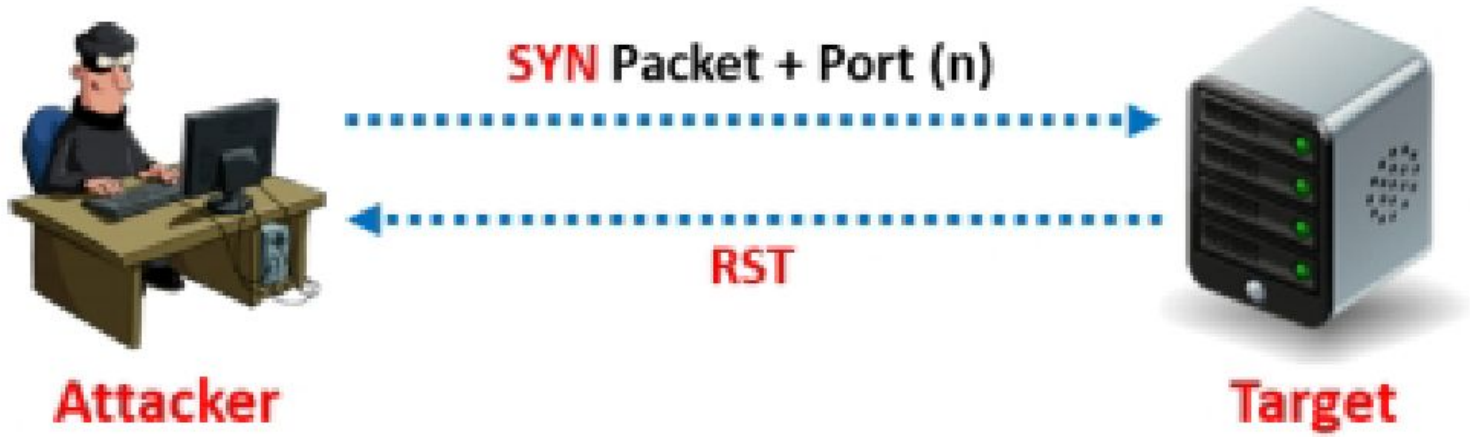
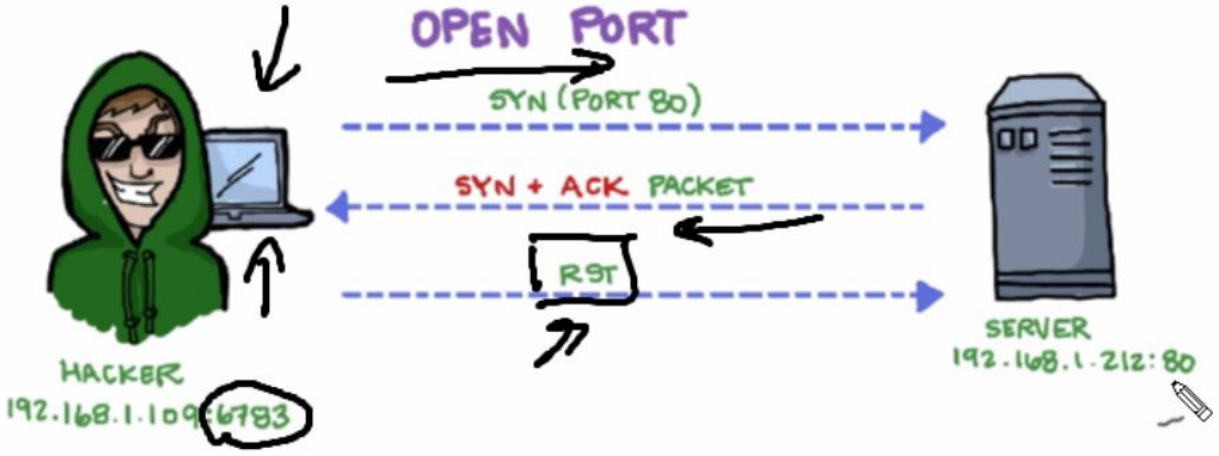


Figure 3.34: Scan result when a port is open

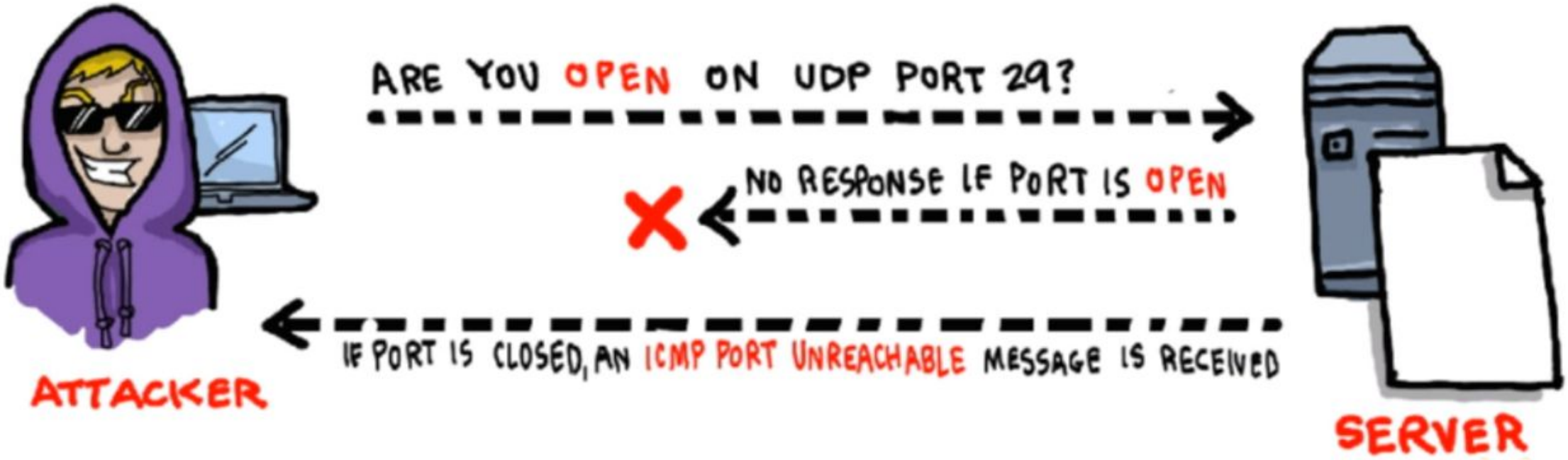


TCP SCAN

By Knocking at the Door



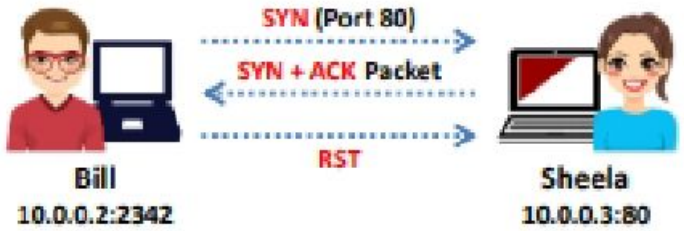
UDP SCAN


ARE YOU OPEN ON UDP PORT 29?
NO RESPONSE IF PORT IS OPEN
IF PORT IS CLOSED, AN ICMP PORT UNREACHABLE MESSAGE IS RECEIVED
ATTACKER
SERVER

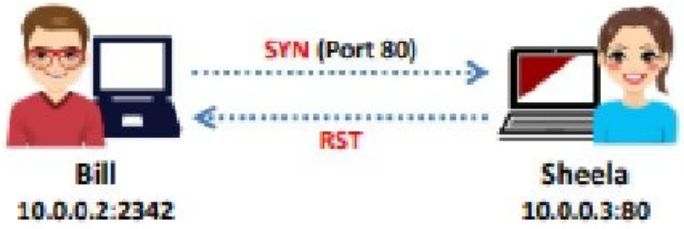
Stealth Scan (Half-open Scan)

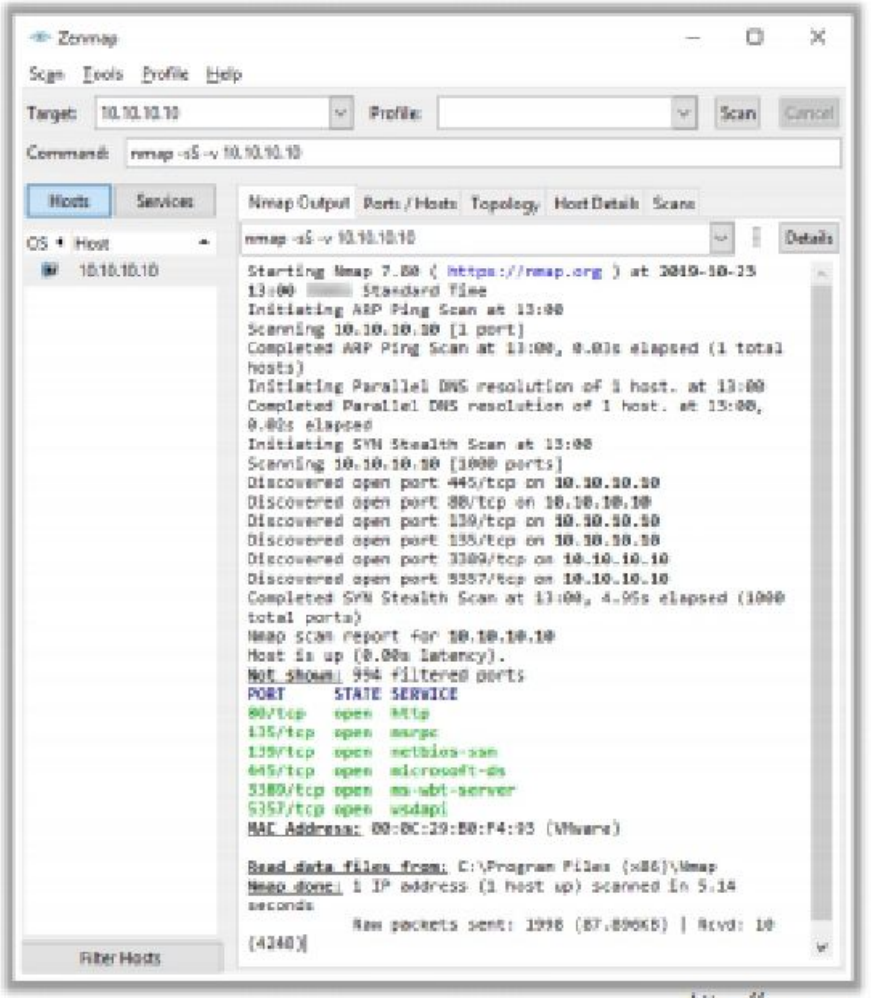
- Stealth scanning involves abruptly resetting the TCP connection between the client and server before the completion of **three-way handshake signals**, thus leaving the connection half-open
- Attackers use stealth scanning techniques to **bypass firewall rules** as well as **logging mechanisms**, and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

Scan result when a port is open



Scan result when a port is closed

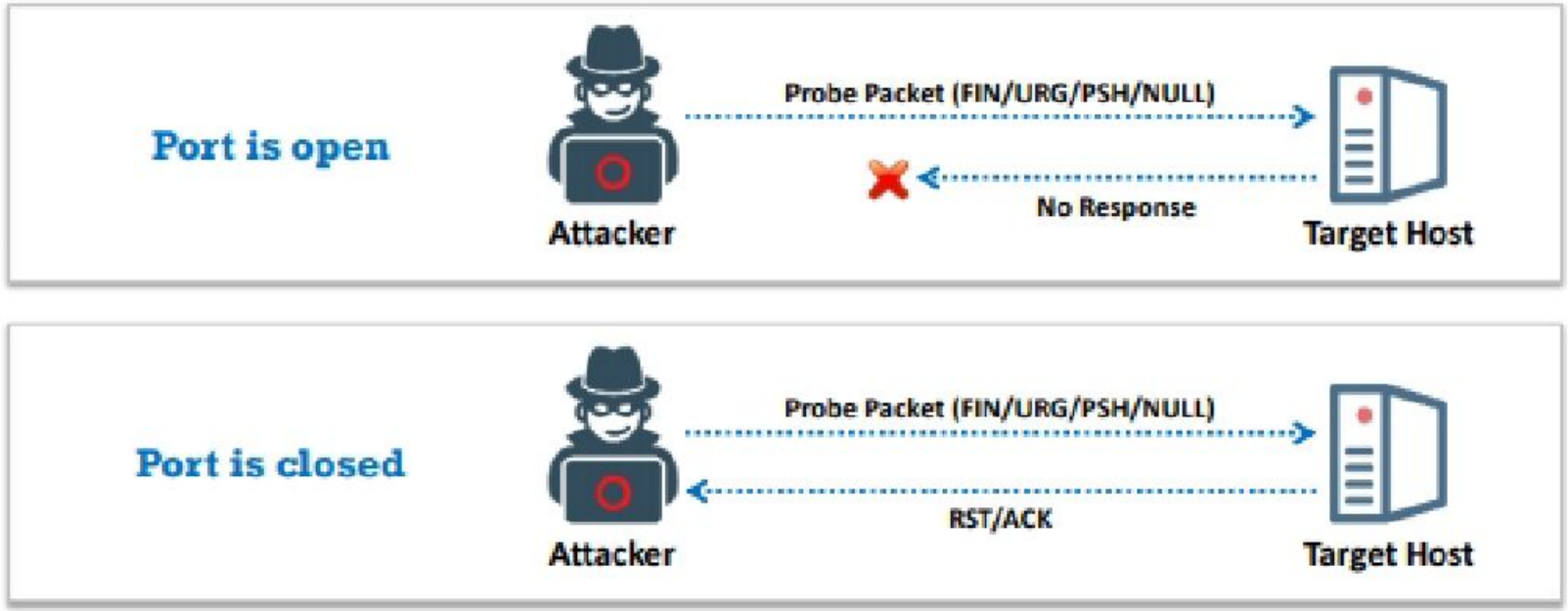




https://nmap.org

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

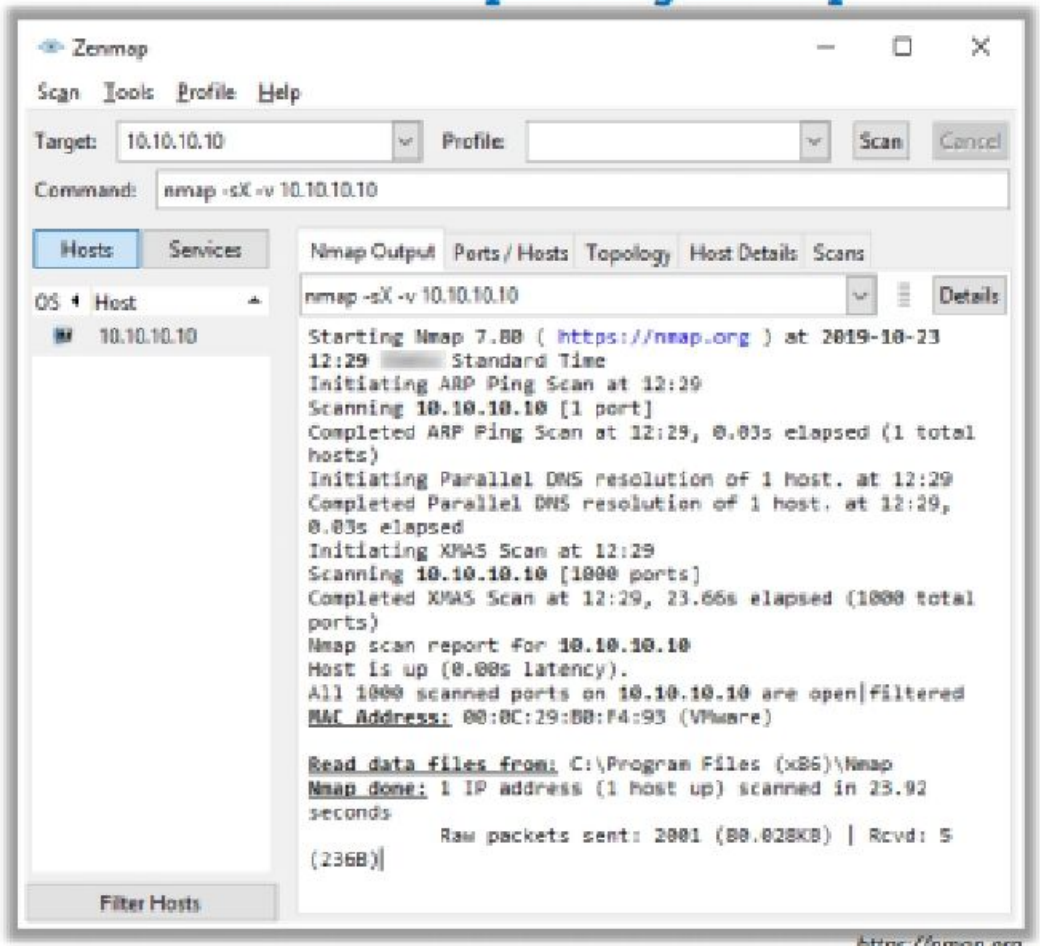


Note: Inverse TCP flag scanning is known as FIN, URG, PSH scanning based on the flag set in the probe packet. It is known as null scanning if there is no flag set

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**

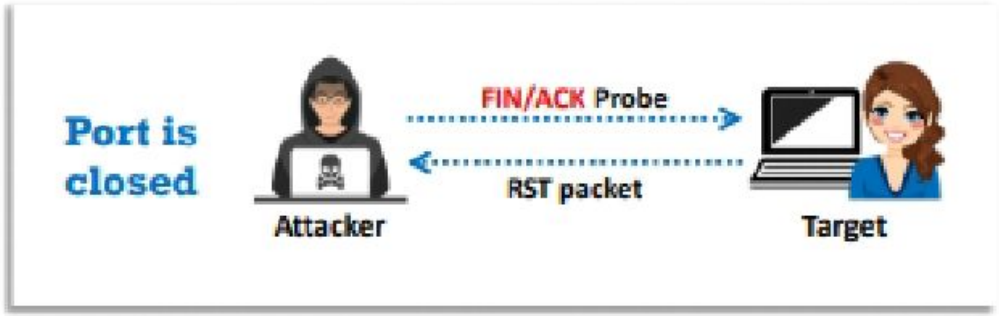
Xmas scan output using Zenmap

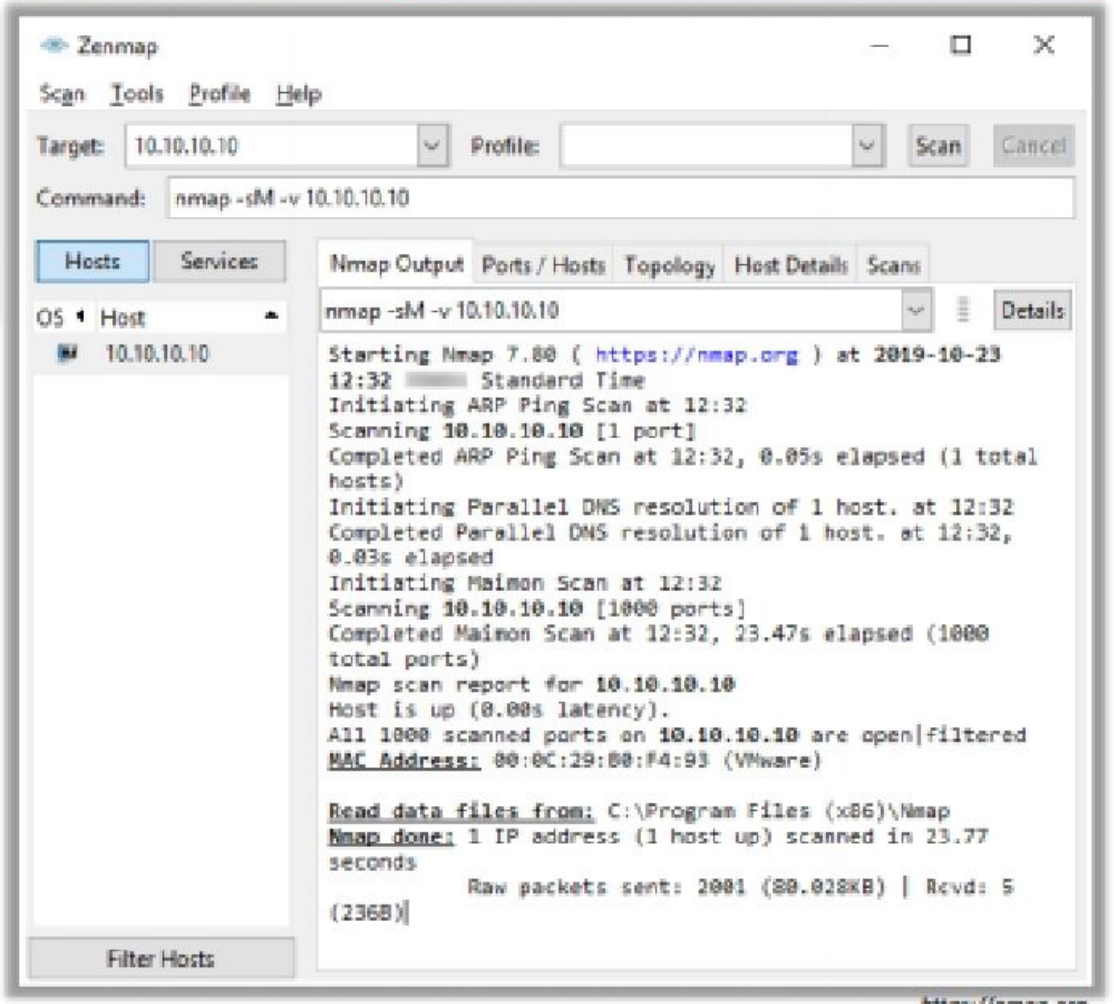


https://nmap.org

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**



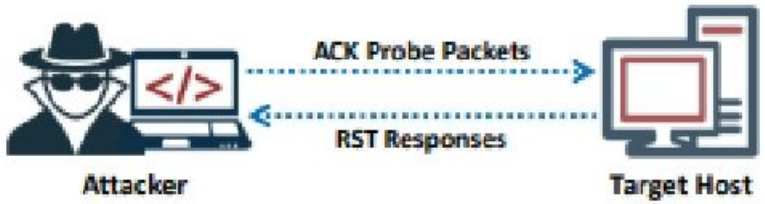


https://nmap.org

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**

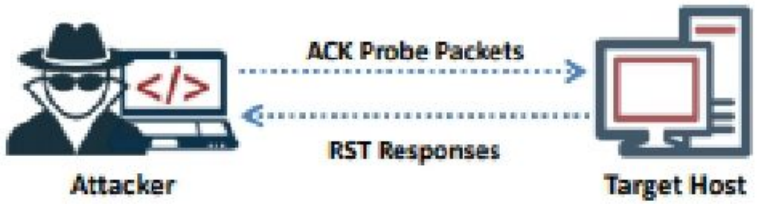
TTL-based ACK Flag Probe scanning




```
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

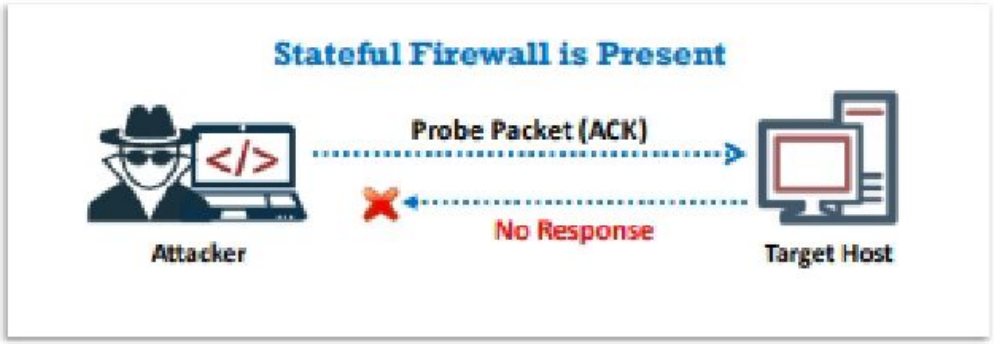


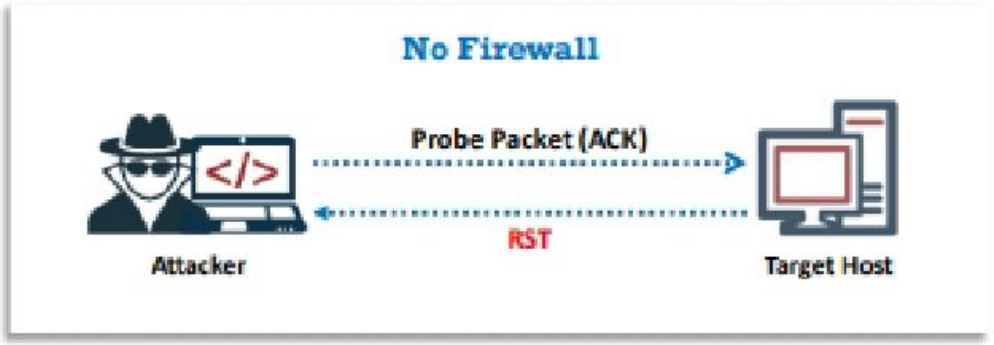

```
1: host 10.2.2.12 port 20: F:RST -> ttl: 64 win: 0
2: host 10.2.2.12 port 21: F:RST -> ttl: 64 win: 0
3: host 10.2.2.12 port 22: F:RST -> ttl: 64 win: 512
4: host 10.2.2.12 port 23: F:RST -> ttl: 64 win: 0
```

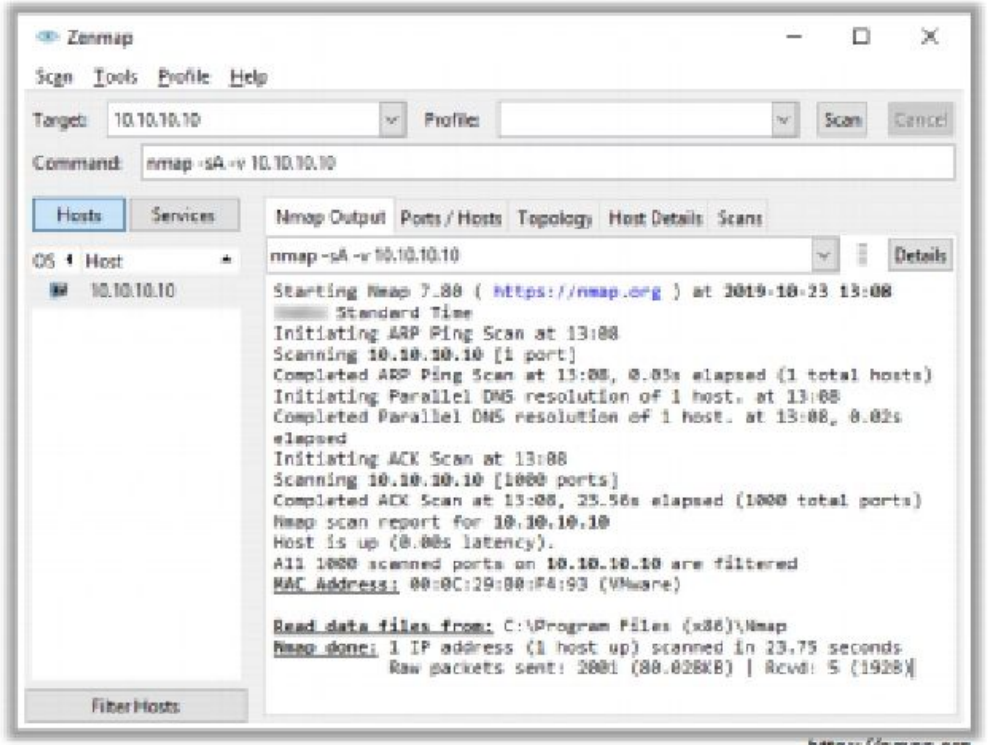
If the **window value of the RST packet** on a particular port has a **non-zero value**, then that **port is open**

ACK Flag Probe Scan (Cont'd)

- 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**





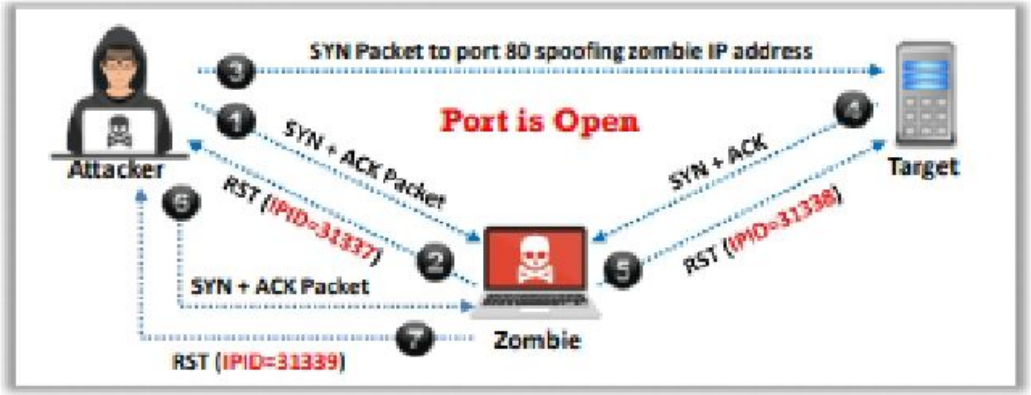


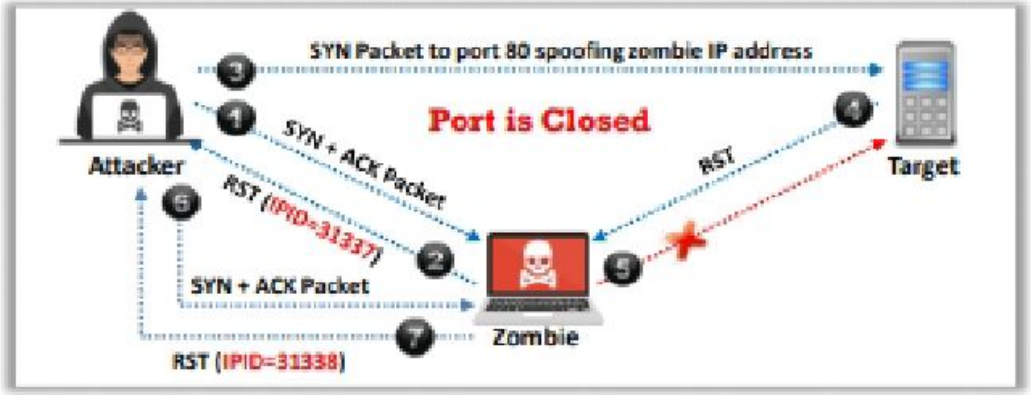
https://nmap.org

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

1. Send SYN + ACK packet to the zombie machine to **probe its IPID number**
2. 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**
3. Send a SYN packet to the **target machine (port 80)** to spoof the IP address of the "zombie"
4. 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
5. If the port is closed, the target will send an **RST to the zombie**, but the zombie will not send anything back
6. 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**



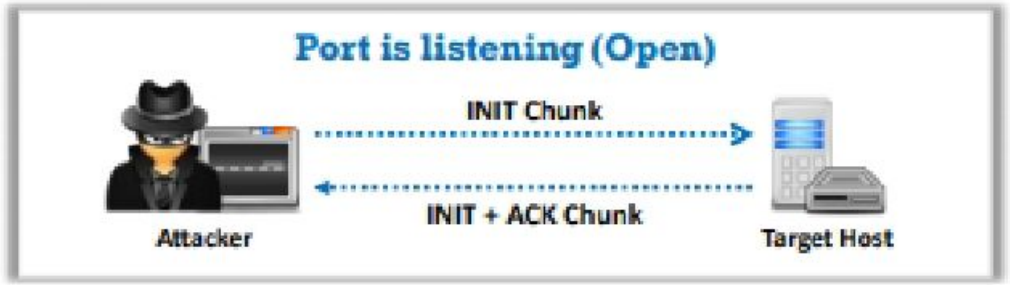


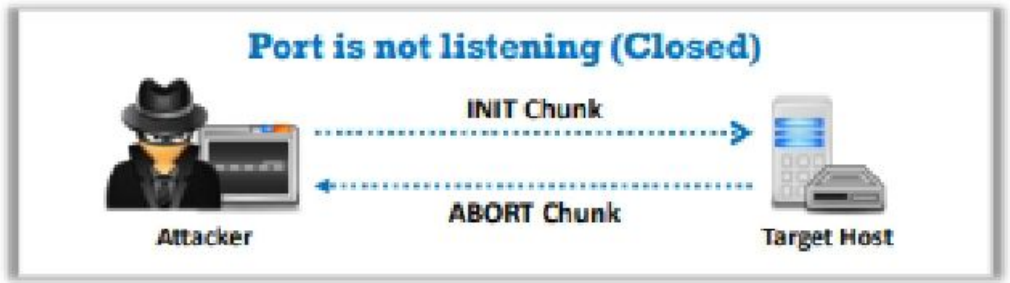
nmap -Pn -p- -sI zombie_ip victim_ip

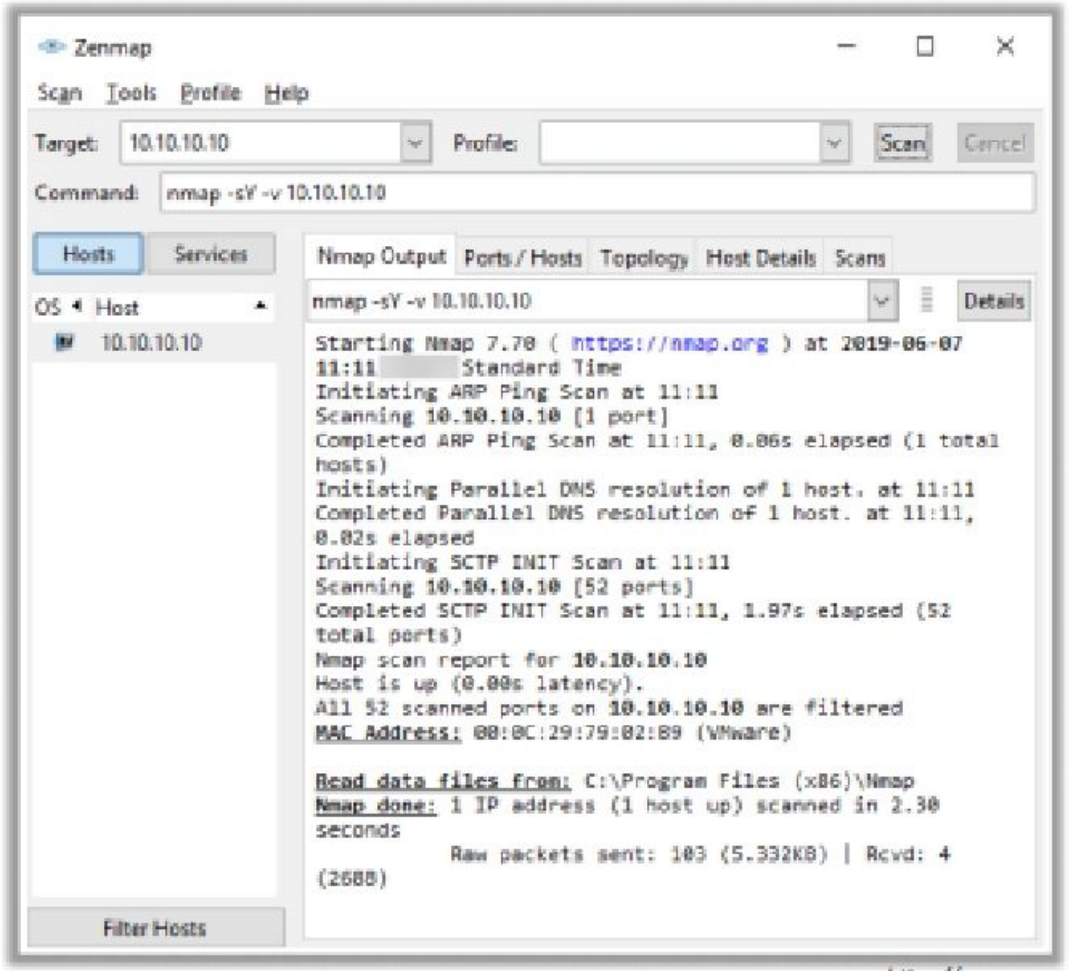

Command Prompt
C:\>nmap -Pn -p- -sI www.eccouncil.org www.certifiedhacker.com
Starting Nmap (http://nmap.org)
Idlescan using zombie www.eccouncil.org (192.130.18.124:80); Class: Incremental
Nmap scan report for 198.182.30.110
(The 40321 ports scanned but not shown below are in state: closed)
Port State Service
21/tcp open ftp
25/tcp open smtp
80/tcp open http
Nmap done: 1 IP address (1 host up) scanned in 1931.23 seconds

SCTP INIT Scanning

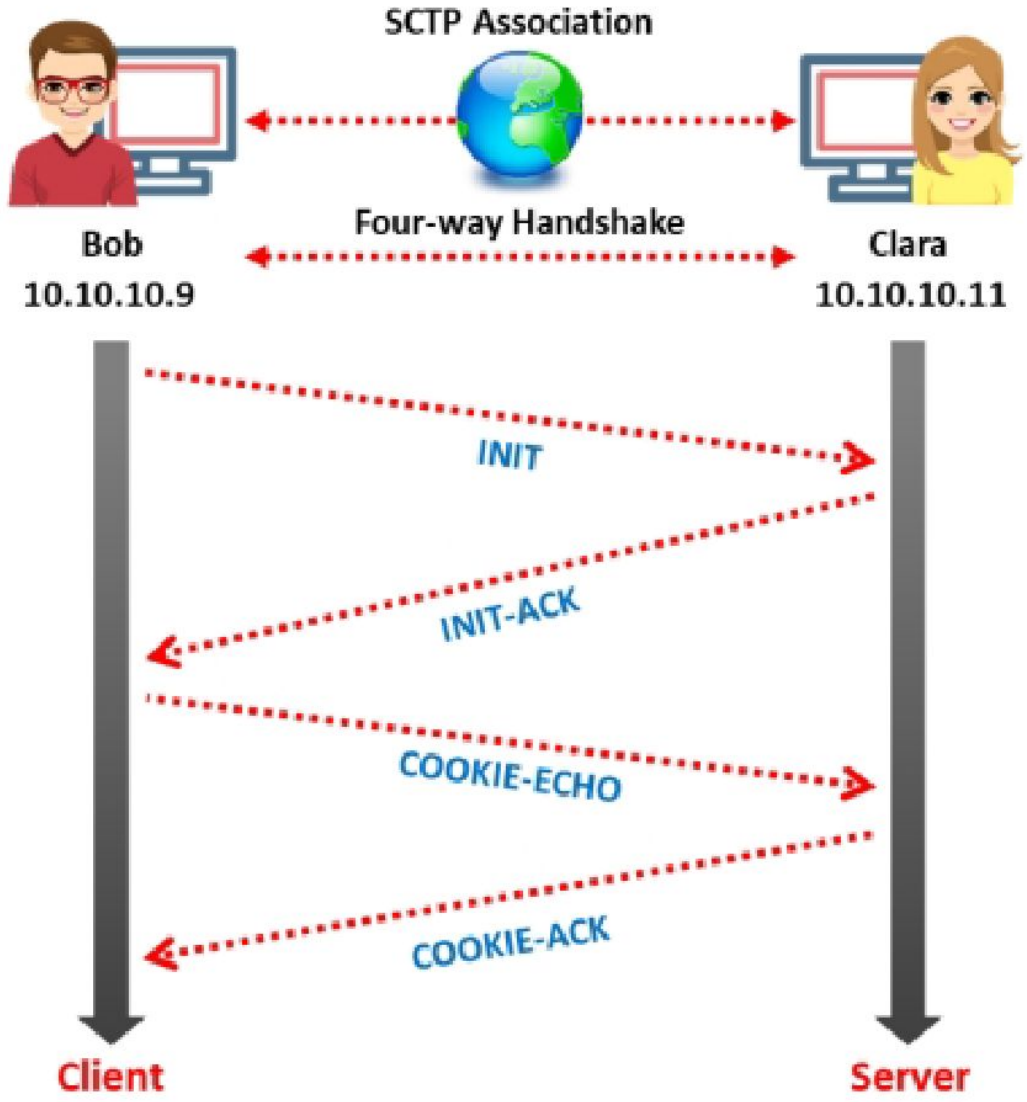
- Attackers send an **INIT chunk** to the target host, and an **INIT+ACK chunk** response implies that the **port is open**, whereas an **ABORT Chunk** response means that the **port is closed**
- **No response** from the target, or a response of an **ICMP unreachable exception** indicates that the port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







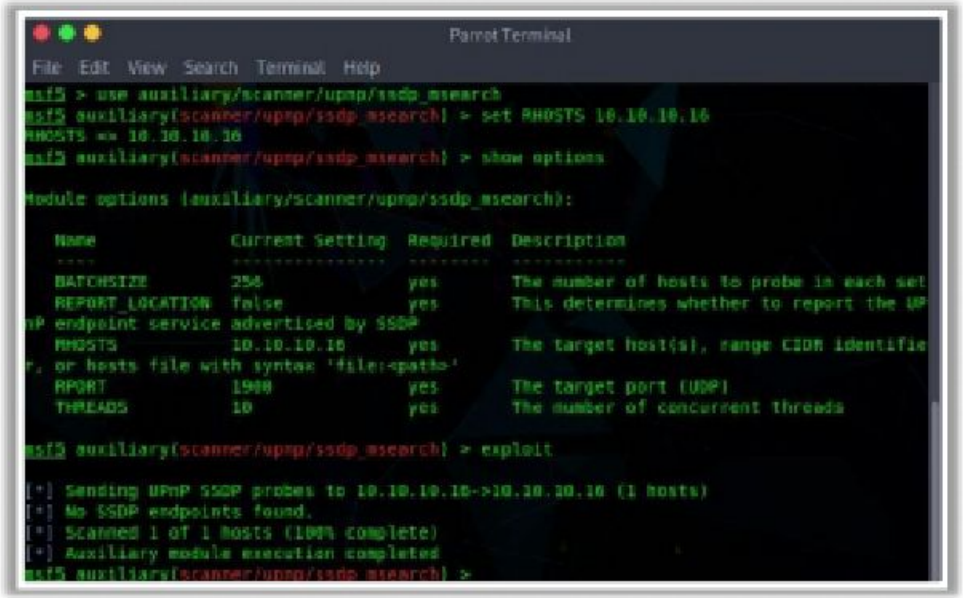
https://nmap.org


SCTP Association
Four-way Handshake
Bob
10.10.10.9
Clara
10.10.10.11
INIT
INIT-ACK
COOKIE-ECHO
COOKIE-ACK
Client
Server

SSDP and List Scanning

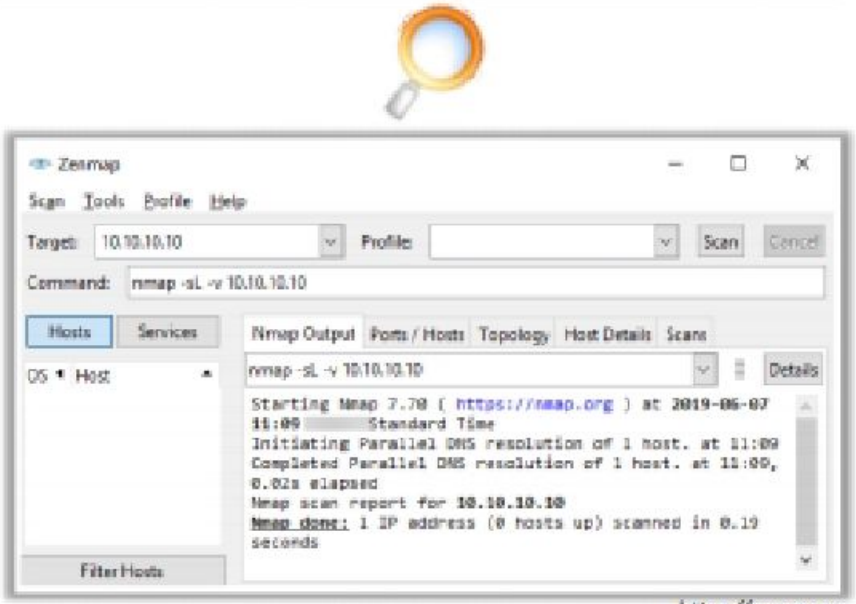
SSDP Scanning

- 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

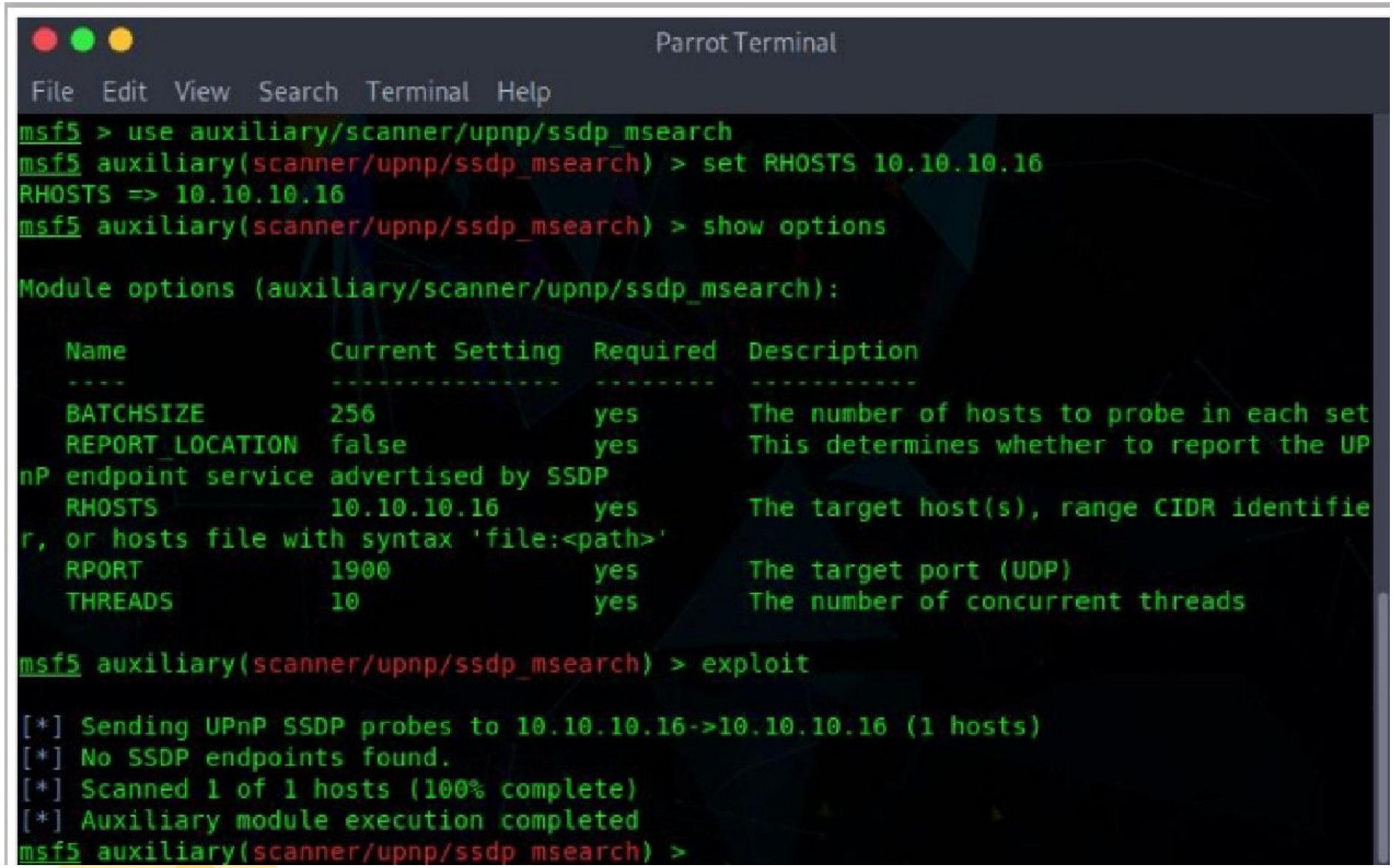


List Scanning

- 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



https://nmap.org


Parrot Terminal
File Edit View Search Terminal Help
msf5 > use auxiliary/scanner/upnp/ssdp_msearch
msf5 auxiliary(scanner/upnp/ssdp_msearch) > set RHOSTS 10.10.10.16
RHOSTS => 10.10.10.16
msf5 auxiliary(scanner/upnp/ssdp_msearch) > show options
Module options (auxiliary/scanner/upnp/ssdp_msearch):
Name Current Setting Required Description
BATCHSIZE 256 yes The number of hosts to probe in each set
REPORT_LOCATION false yes This determines whether to report the UP
nP endpoint service advertised by SSDP
RHOSTS 10.10.10.16 yes The target host(s), range CIDR identifie
r, or hosts file with syntax 'file:<path>'
RPORT 1900 yes The target port (UDP)
THREADS 10 yes The number of concurrent threads
msf5 auxiliary(scanner/upnp/ssdp_msearch) > exploit
[*] Sending UPnP SSDP probes to 10.10.10.16->10.10.10.16 (1 hosts)
[*] No SSDP endpoints found.
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf5 auxiliary(scanner/upnp/ssdp_msearch) >

IPv6 Scanning

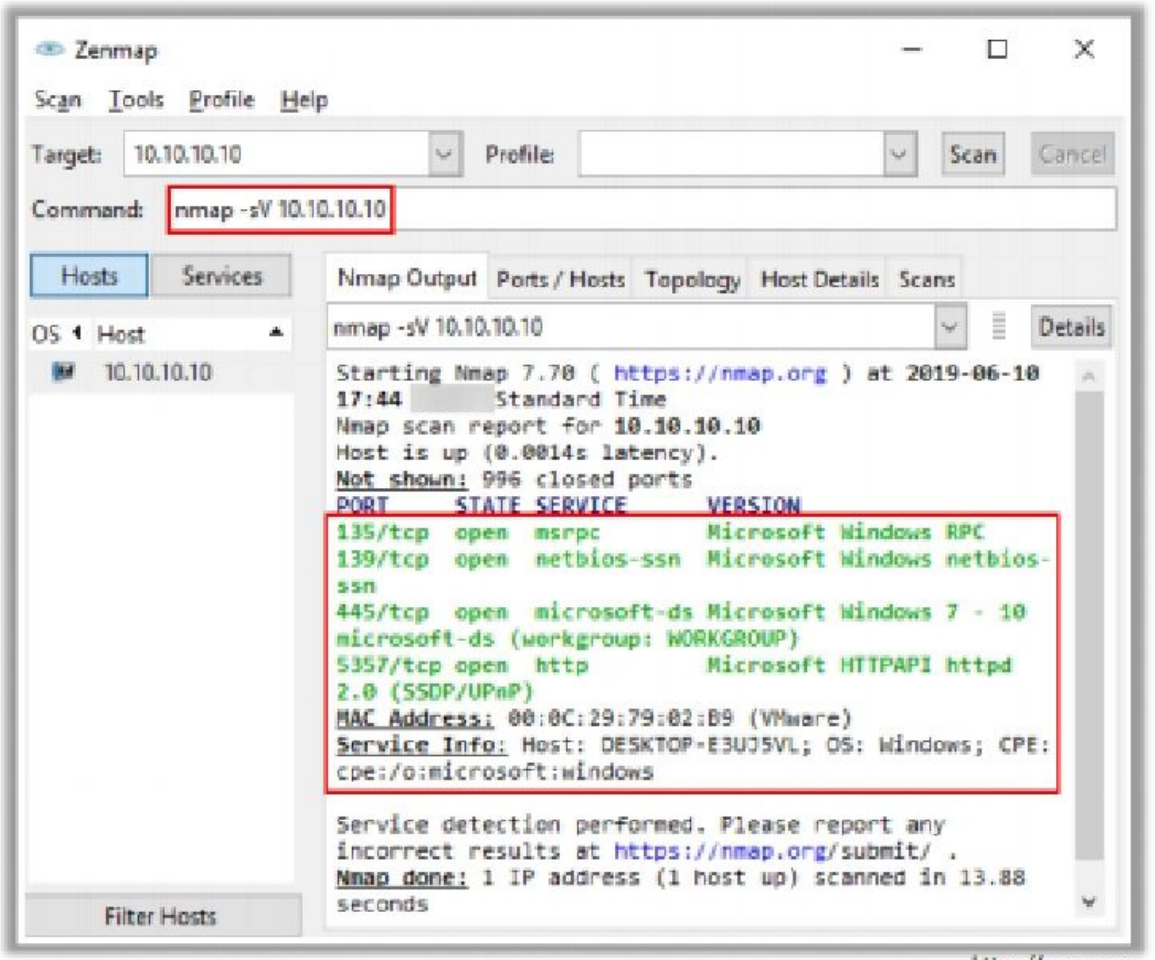
- IPv6 increases the IP address size from **32 bits** to **128 bits** to support more levels of address hierarchy
- Attackers need to harvest IPv6 addresses from **network traffic**, **recorded logs**, or **Received from**: header lines in archived emails
- Attackers can use the **-6** option in Zenmap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

```
root@[redacted]:~# nmap -6 scanme.nmap.org

Starting Nmap [redacted] ( http://nmap.org ) at [redacted] 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

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**
- 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



https://nmap.org

Nmap Scan Time Reduction Techniques

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

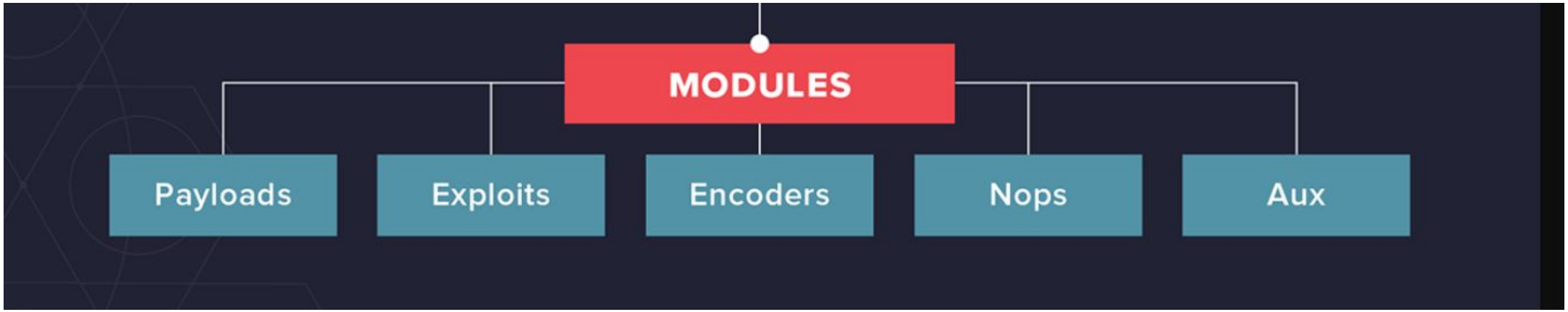
Scan Time Reduction Techniques

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


METASPLOIT
1496
Metasploit
Learn to use Metasploit, a tool to probe and exploit vulnerabilities on networks and servers.
Start AttackBox
Awards
Help
Options


MODULES
Payloads
Exploits
Encoders
Nops
Aux

1496

Metasploit

Learn to use Metasploit, a tool to probe and exploit vulnerabilities on networks and servers.

Start AttackBox | Awards | Help | Options

href="/room/kali">Kali Machine

- Enroll in all learning paths
- Private OpenVPN VIP server
- Deploy machines faster

Learn More

100%

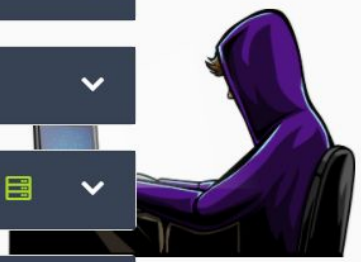
Task 1 Intro

Task 2 Initializing...

Task 3 Rock 'em to the Core [Commands]

Task 4 Modules for Every Occasion!

Task 5 Move that shell!



Task 6 We're in, now what?

Task 7 Makin' Cisco Proud

Created by DarkStar7471

OS Discovery (Banner Grabbing/OS Fingerprinting)

An attacker uses OS discovery or banner grabbing techniques to identify network hosts running application and OS versions with known exploits. This section introduces you to banner grabbing, its types, and its tools, as well as useful countermeasures that you can adopt against it.

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**

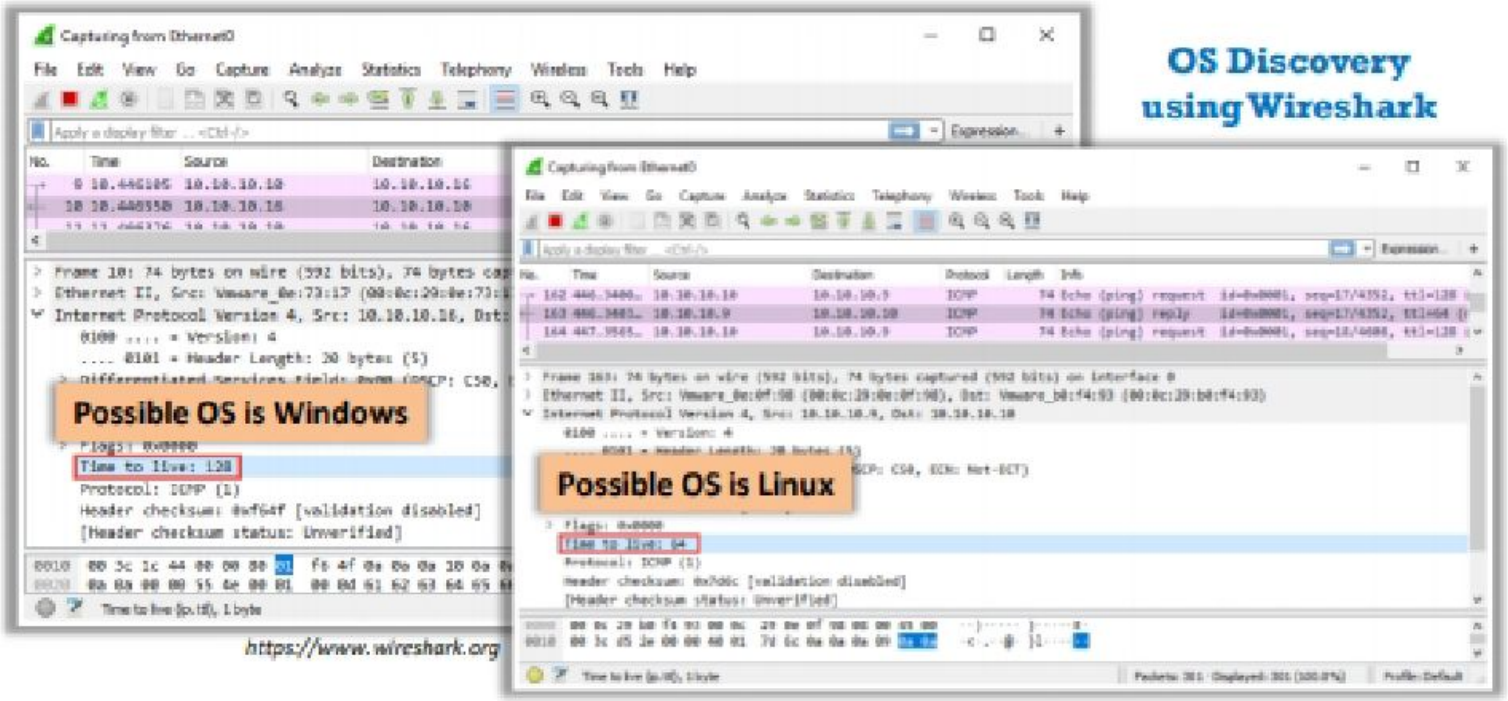
  Looking for an extension in the URL may assist in determining the application's version.

  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



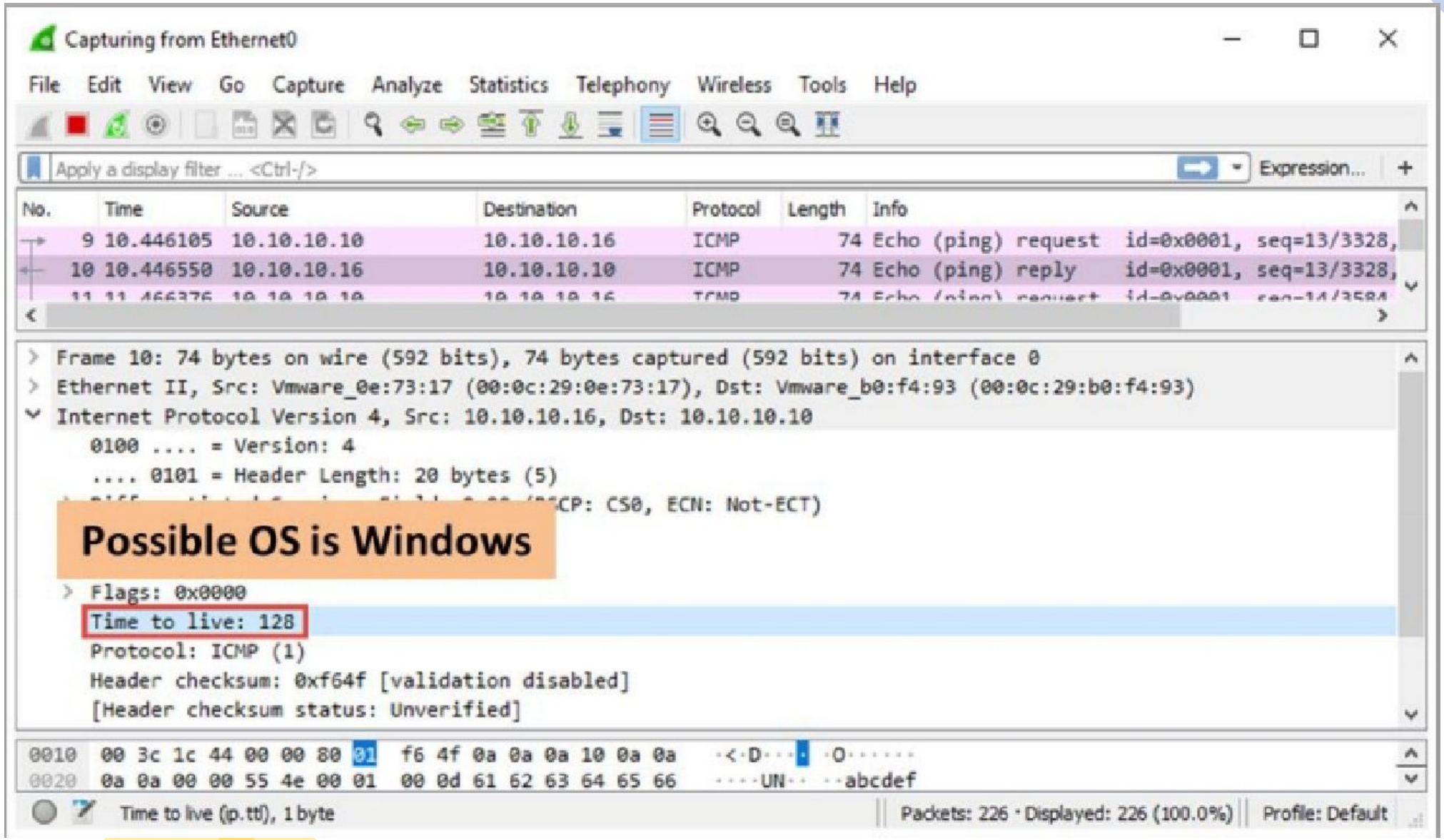
https://www.wireshark.org

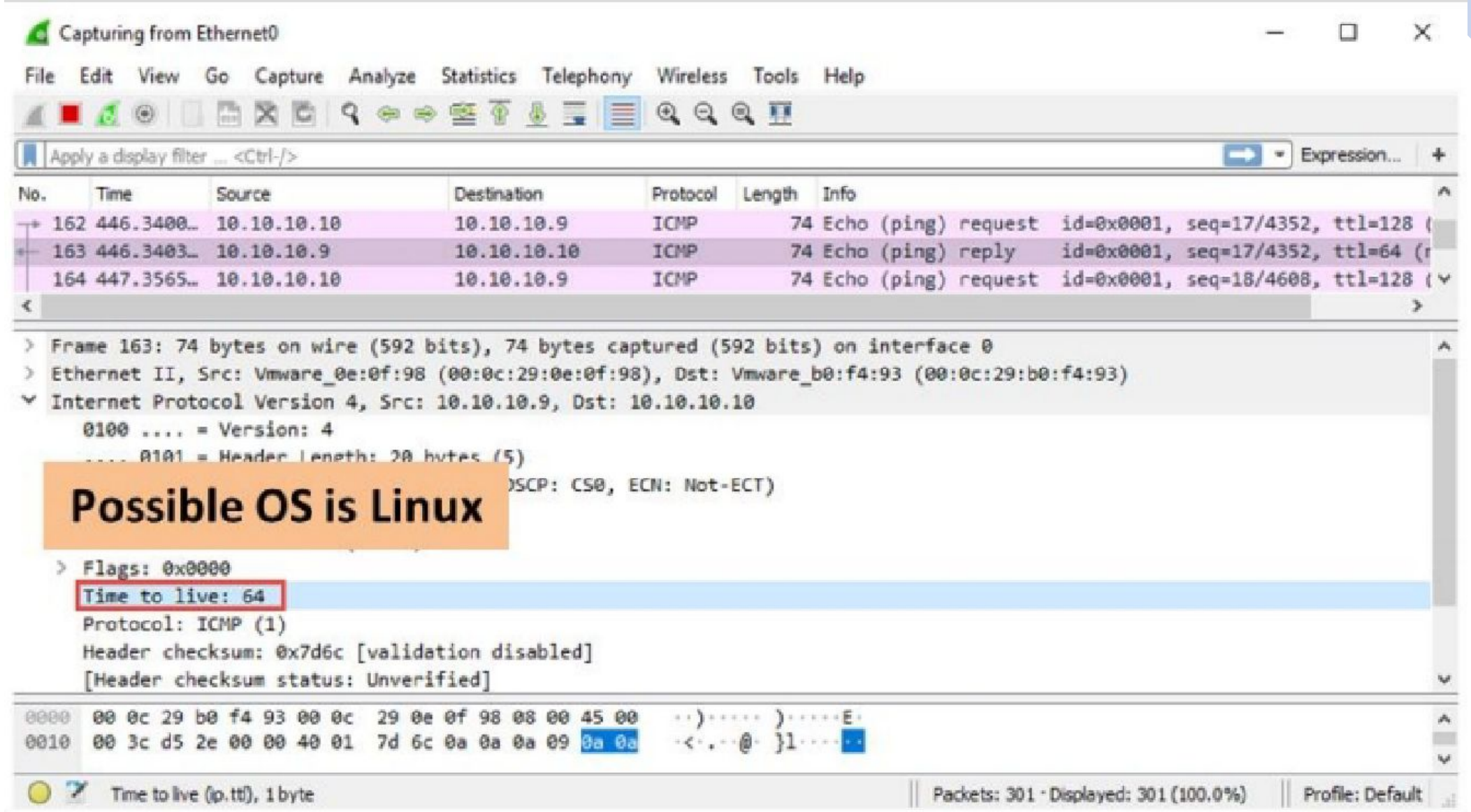
OS Discovery using Wireshark

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

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

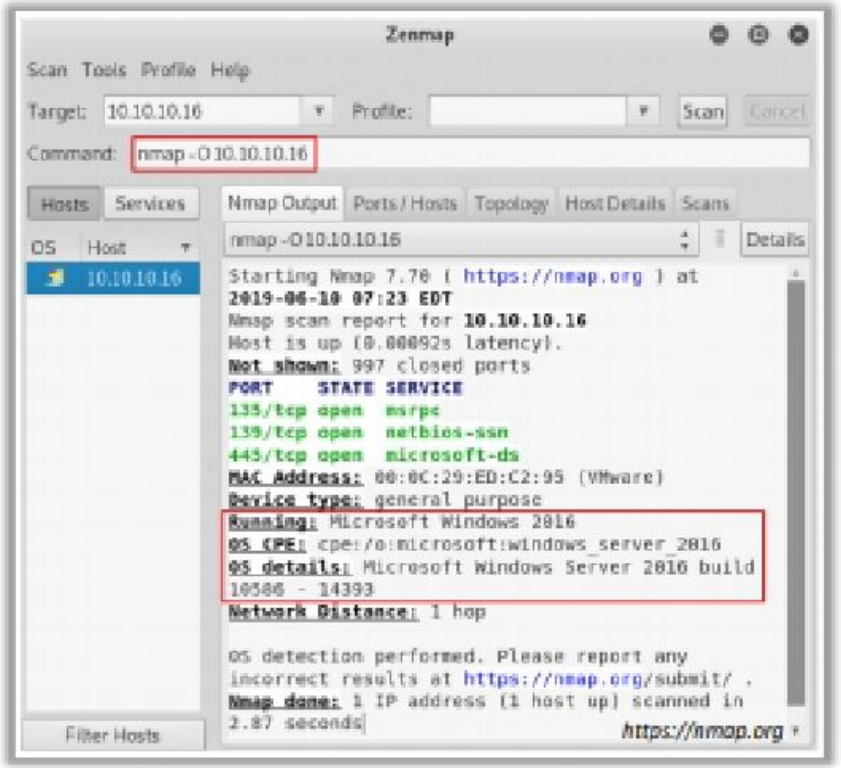

Capturing from Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
9 10.446105 10.10.10.10 10.10.10.16 ICMP 74 Echo (ping) request id=0x0001, seq=13/3328,
10 10.446550 10.10.10.16 10.10.10.10 ICMP 74 Echo (ping) reply id=0x0001, seq=13/3328,
Frame 10: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Vmware_0e:73:17 (00:0c:29:0e:73:17), Dst: Vmware_b0:f4:93 (00:0c:29:b0:f4:93)
Internet Protocol Version 4, Src: 10.10.10.16, Dst: 10.10.10.10
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Possible OS is Windows
Flags: 0x0000
Time to live: 128
Protocol: ICMP (1)
Header checksum: 0xf64f [validation disabled]
[Header checksum status: Unverified]
0010 00 3c 1c 44 00 00 80 01 f6 4f 0a 0a 0a 10 0a 0a
0020 0a 0a 00 00 55 4e 00 01 00 0d 61 62 63 64 65 66
Time to live (p.ttl), 1 byte
Packets: 226 · Displayed: 226 (100.0%)
Profile: Default

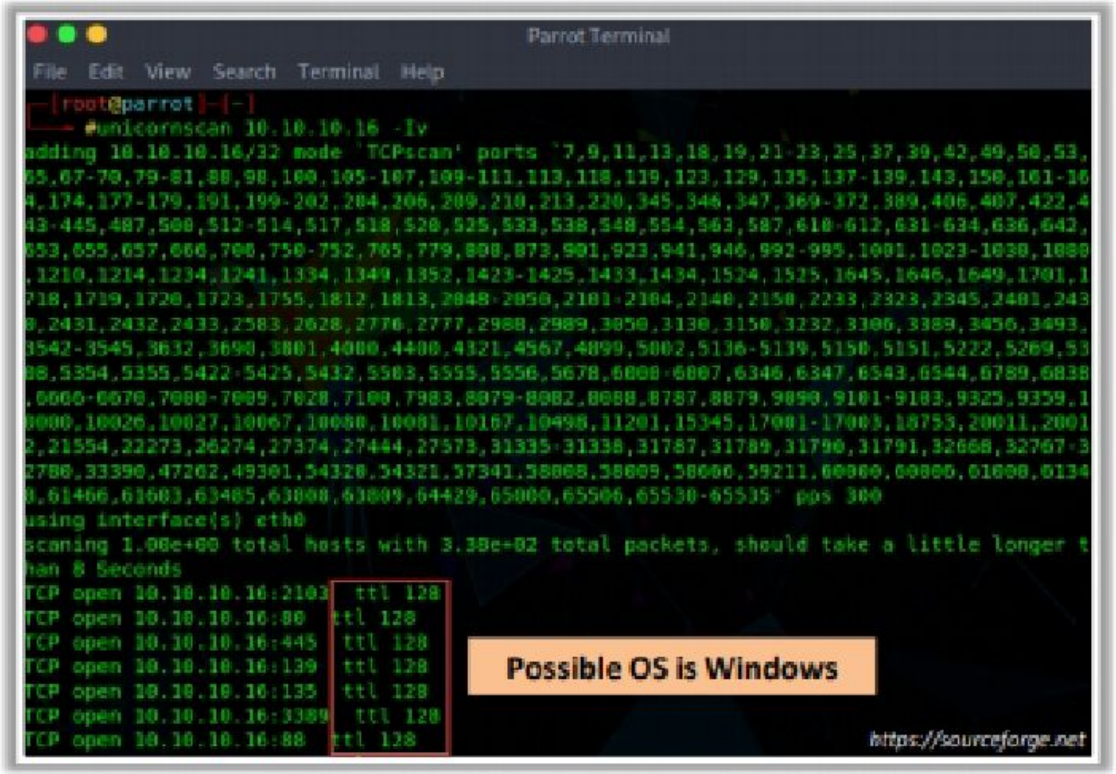

Capturing from Ethernet0
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
No. Time Source Destination Protocol Length Info
162 446.3400… 10.10.10.10 10.10.10.9 ICMP 74 Echo (ping) request id=0x0001, seq=17/4352, ttl=128
163 446.3403… 10.10.10.9 10.10.10.10 ICMP 74 Echo (ping) reply id=0x0001, seq=17/4352, ttl=64
164 447.3565… 10.10.10.10 10.10.10.9 ICMP 74 Echo (ping) request id=0x0001, seq=18/4608, ttl=128
Frame 163: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface 0
Ethernet II, Src: Vmware_0e:0f:98 (00:0c:29:0e:0f:98), Dst: Vmware_b0:f4:93 (00:0c:29:b0:f4:93)
Internet Protocol Version 4, Src: 10.10.10.9, Dst: 10.10.10.10
0100 .... = Version: 4
Possible OS is Linux
Flags: 0x0000
Time to live: 64
Protocol: ICMP (1)
Header checksum: 0x7d6c [validation disabled]
[Header checksum status: Unverified]
0000 00 0c 29 b0 f4 93 00 0c 29 0e 0f 98 08 00 45 00
0010 00 3c d5 2e 00 00 40 01 7d 6c 0a 0a 0a 09 0a 0a
Time to live (ip.ttl), 1 byte
Packets: 301 · Displayed: 301 (100.0%)
Profile: Default

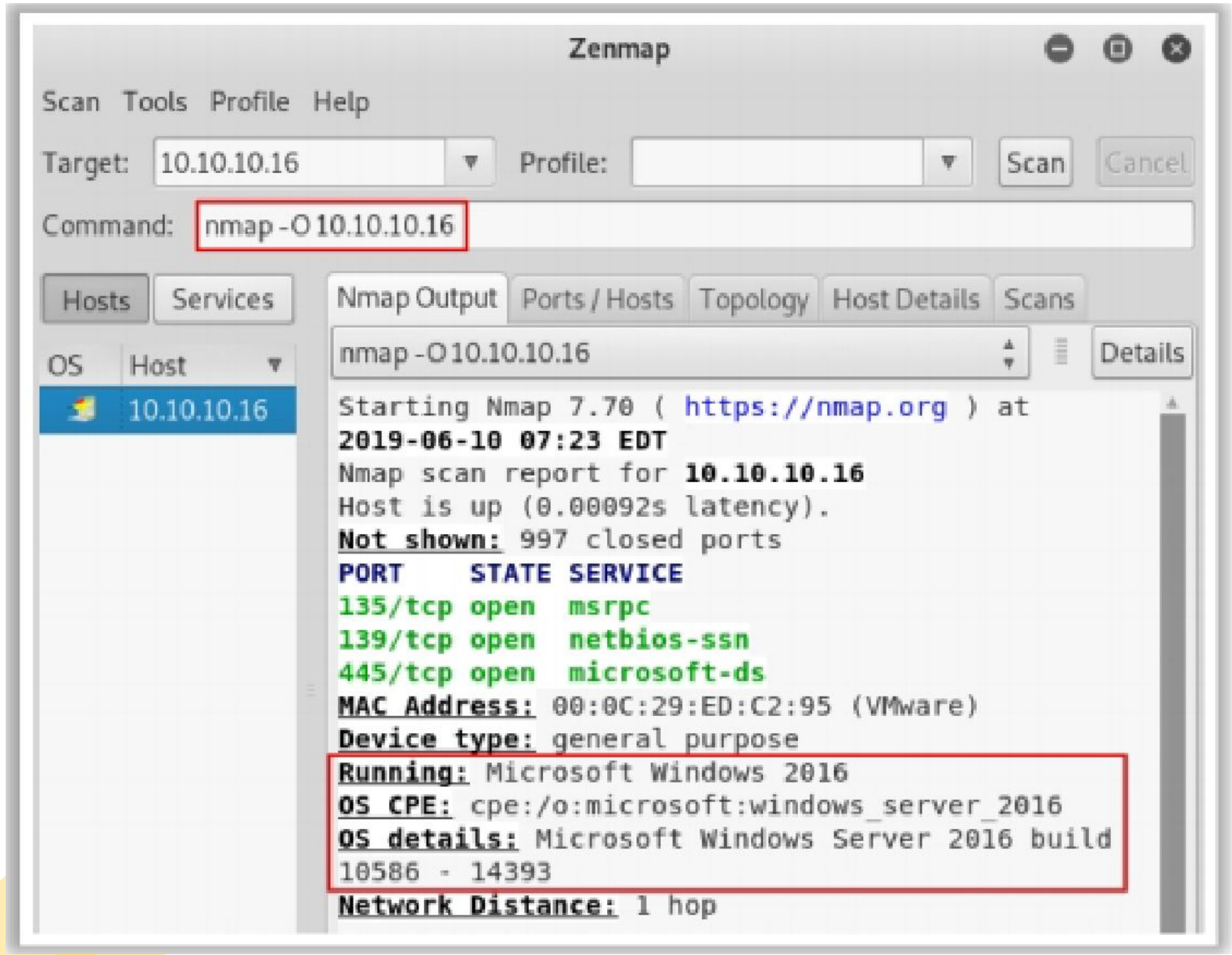
OS Discovery using Nmap and Unicornscan

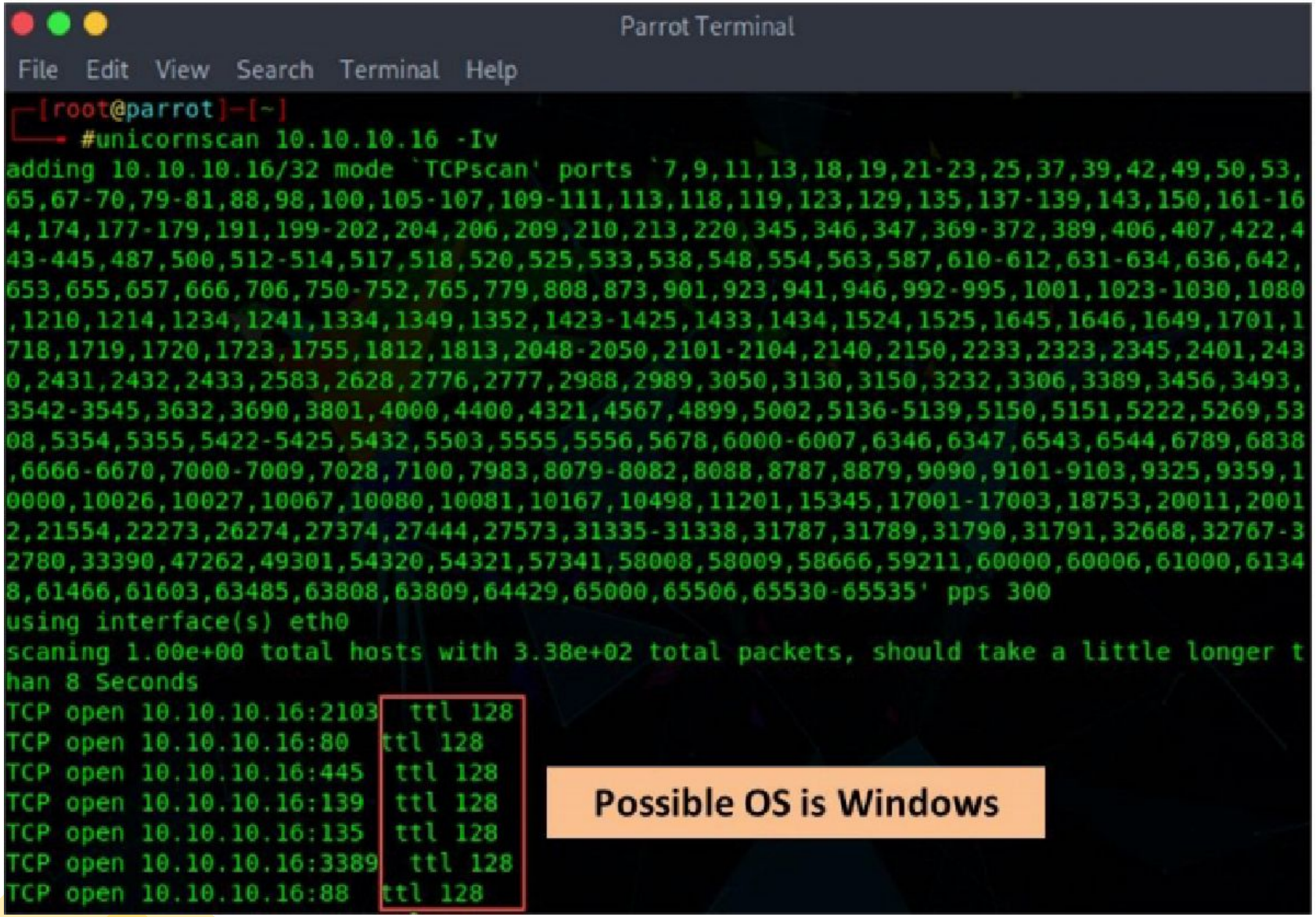
- In **Nmap**, the **-O** option is used to perform OS discovery, providing OS details of the target machine

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



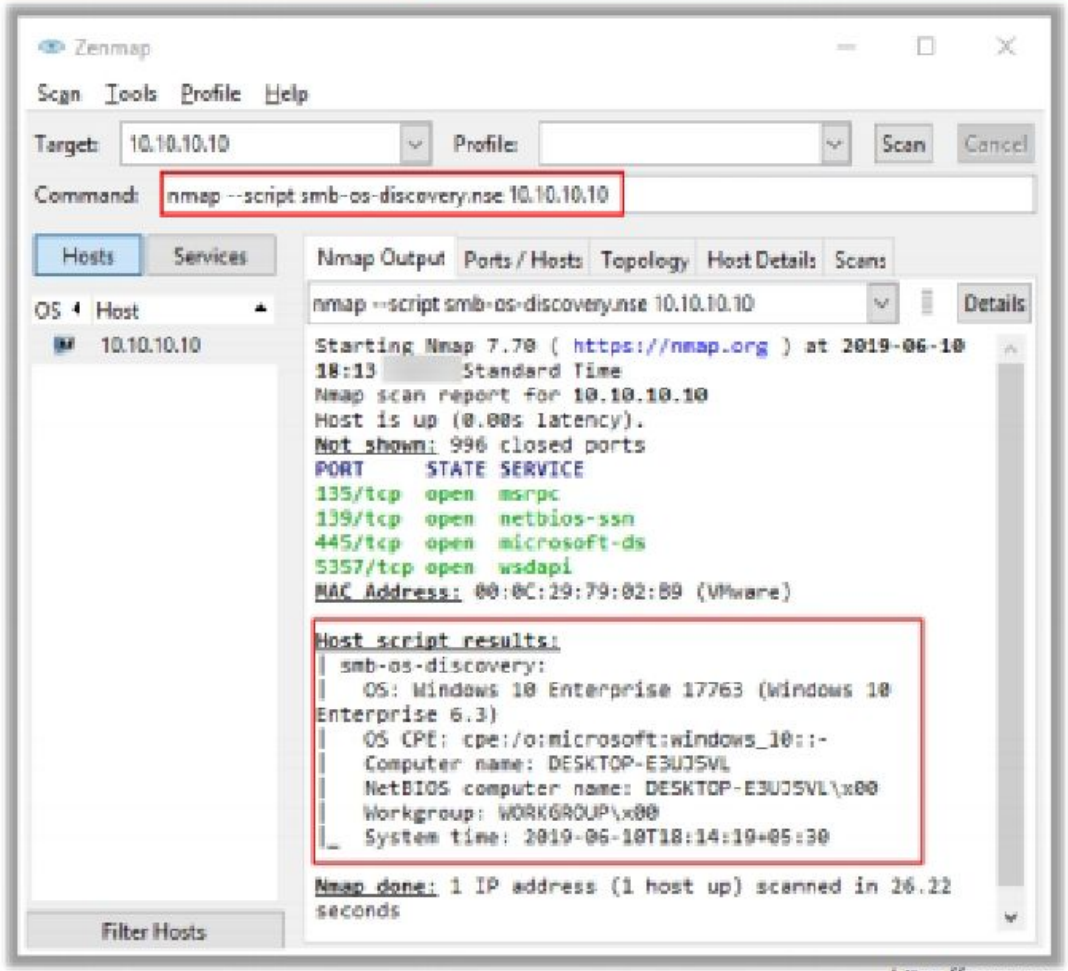



Zenmap
Scan Tools Profile Help
Target: 10.10.10.16
Profile:
Scan
Cancel
Command: nmap -O 10.10.10.16
Hosts
Services
OS
Host
10.10.10.16
Nmap Output
Ports / Hosts
Topology
Host Details
Scans
nmap -O 10.10.10.16
Details
Starting Nmap 7.70 (https://nmap.org) at
2019-06-10 07:23 EDT
Nmap scan report for 10.10.10.16
Host is up (0.00092s latency).
Not shown: 997 closed ports
PORT STATE SERVICE
135/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:/o:microsoft:windows_server_2016
OS details: Microsoft Windows Server 2016 build
10586 - 14393
Network Distance: 1 hop


Parrot Terminal
File Edit View Search Terminal Help
[root@parrot]-[~]
#unicornscan 10.10.10.16 -Iv
adding 10.10.10.16/32 mode `TCPscan' ports `7,9,11,13,18,19,21-23,25,37,39,42,49,50,53,65,67-70,79-81,88,98,100,105-107,109-111,113,118,119,123,129,135,137-139,143,150,161-164,174,177-179,191,199-202,204,206,209,210,213,220,345,346,347,369-372,389,406,407,422,443-445,487,500,512-514,517,518,520,525,533,538,548,554,563,587,610-612,631-634,636,642,653,655,657,666,706,750-752,765,779,808,873,901,923,941,946,992-995,1001,1023-1030,1080,1210,1214,1234,1241,1334,1349,1352,1423-1425,1433,1434,1524,1525,1645,1646,1649,1701,1718,1719,1720,1723,1755,1812,1813,2048-2050,2101-2104,2140,2150,2233,2323,2345,2401,2430,2431,2432,2433,2583,2628,2776,2777,2988,2989,3050,3130,3150,3232,3306,3389,3456,3493,3542-3545,3632,3690,3801,4000,4400,4321,4567,4899,5002,5136-5139,5150,5151,5222,5269,5308,5354,5355,5422-5425,5432,5503,5555,5556,5678,6000-6007,6346,6347,6543,6544,6789,6838,6666-6670,7000-7009,7028,7100,7983,8079-8082,8088,8787,8879,9090,9101-9103,9325,9359,10000,10026,10027,10067,10080,10081,10167,10498,11201,15345,17001-17003,18753,20011,20012,21554,22273,26274,27374,27444,27573,31335-31338,31787,31789,31790,31791,32668,32767-32780,33390,47262,49301,54320,54321,57341,58008,58009,58666,59211,60000,60006,61000,61348,61466,61603,63485,63808,63809,64429,65000,65506,65530-65535' pps 300
using interface(s) eth0
scaning 1.00e+00 total hosts with 3.38e+02 total packets, should take a little longer than 8 Seconds
TCP open 10.10.10.16:2103 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

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



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

Scanning Beyond IDS and Firewall

Intrusion detection systems (IDS) and firewalls are security mechanisms intended to prevent an attacker from accessing a network. However, even IDS and firewalls have some security limitations. Attackers try to launch attacks to exploit these limitations. This section highlights various IDS/firewall evasion techniques such as packet fragmentation, source routing, IP address spoofing, etc.

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 **evading 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

IDS/Firewall Evasion Techniques

Although firewalls and IDS can prevent malicious traffic (packets) from entering a network, attackers can send intended packets to the target that evade the IDS/firewall by implementing the following techniques:

- **Packet Fragmentation**: The attacker sends fragmented probe packets to the intended target, which reassembles the fragments after receiving all of them.
- **Source Routing**: The attacker specifies the routing path for the malformed packet to reach the intended target.
- **Source Port Manipulation**: The attacker manipulates the actual source port with the common source port to evade the IDS/firewall.
- **IP Address Decoy**: The attacker generates or manually specifies IP addresses of decoys so that the IDS/firewall cannot determine the actual IP address.
- **IP Address Spoofing**: The attacker changes the source IP addresses so that the attack appears to be coming from someone else.
- **Creating Custom Packets**: The attacker sends custom packets to scan the intended target beyond the firewalls.
- **Randomizing Host Order**: The attacker scans the number of hosts in the target network in a random order to scan the intended target that lies beyond the firewall.
- **Sending Bad Checksums**: The attacker sends packets with bad or bogus TCP/UPD checksums to the intended target.

- **Proxy Servers**: The attacker uses a chain of proxy servers to hide the actual source of a scan and evade certain IDS/firewall restrictions.
- **Anonymizers**: The attacker uses anonymizers, which allows them to bypass Internet censors and evade certain IDS and firewall rules.

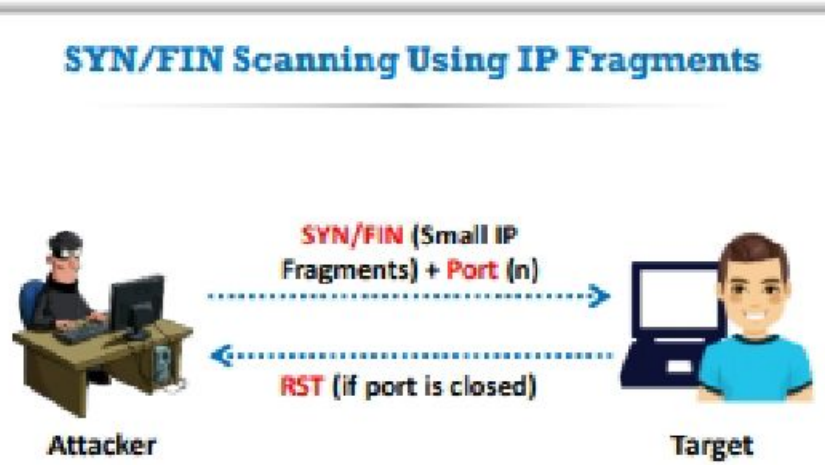
Packet Fragmentation

- 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

Command Prompt

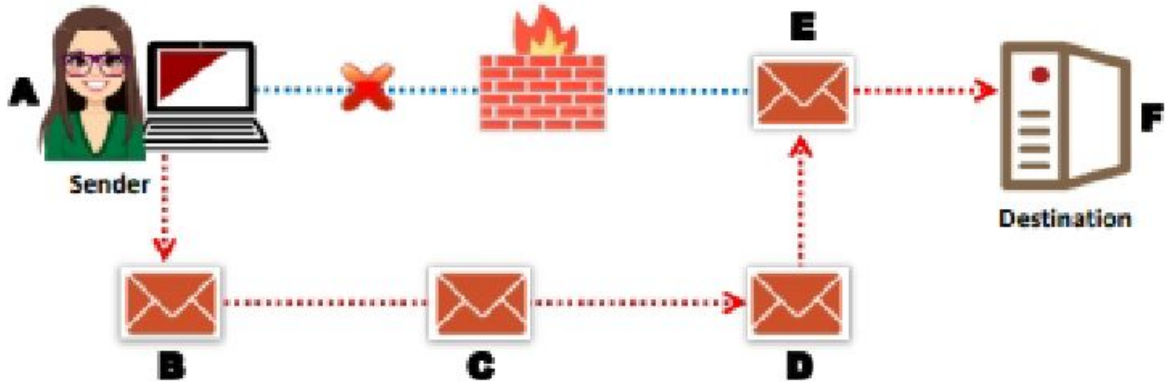
```
C:\>nmap -sS -T4 -A -f -v 10.10.10.10

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, 4.75s elapsed (1000
total ports)
```



Source Routing

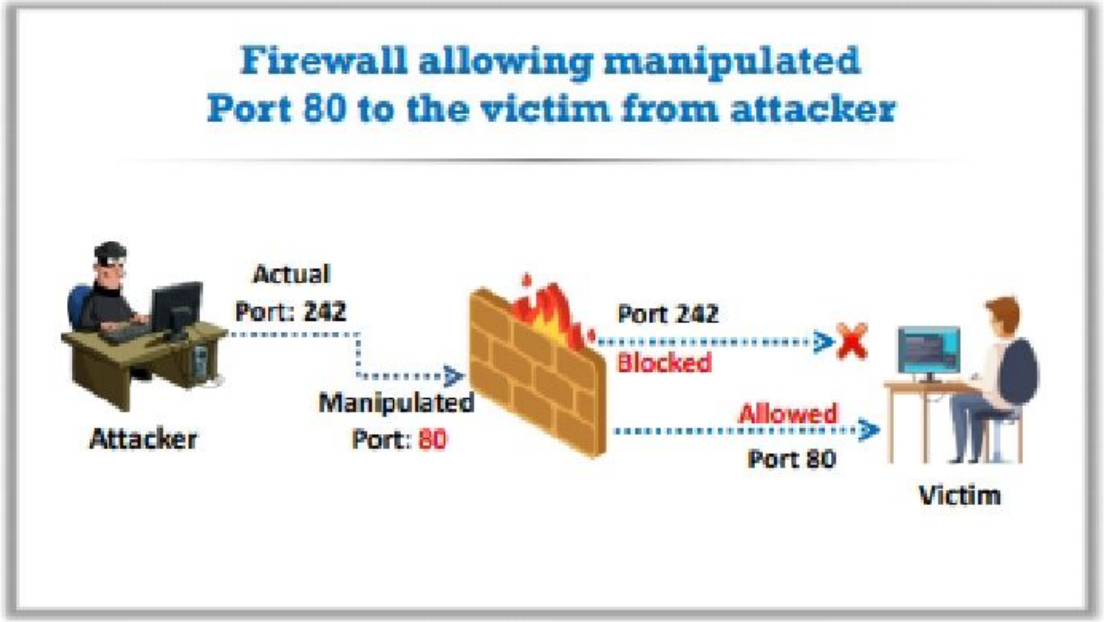
- As the packet travels through the nodes in the network, each **router examines** the destination IP address and **chooses the next hop** to direct the packet to the destination
- Source routing refers to sending a packet to the intended destination with a partially or completely **specified route** (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

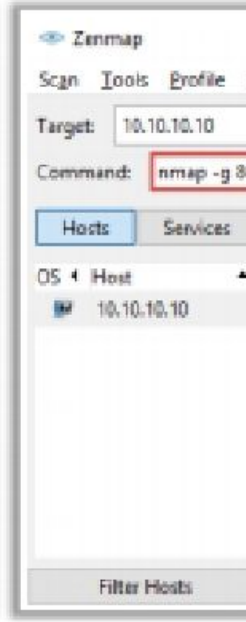
This figure shows source routing, where the originator dictates the eventual route of the traffic



Source Port Manipulation

- Source port manipulation refers to **manipulating actual port numbers wit** evade an IDS or firewall
- It occurs when a firewall is **configured to allow packets** from well-known p
- **Nmap** uses the **-g** or **--source-port** options to perform source port manipu





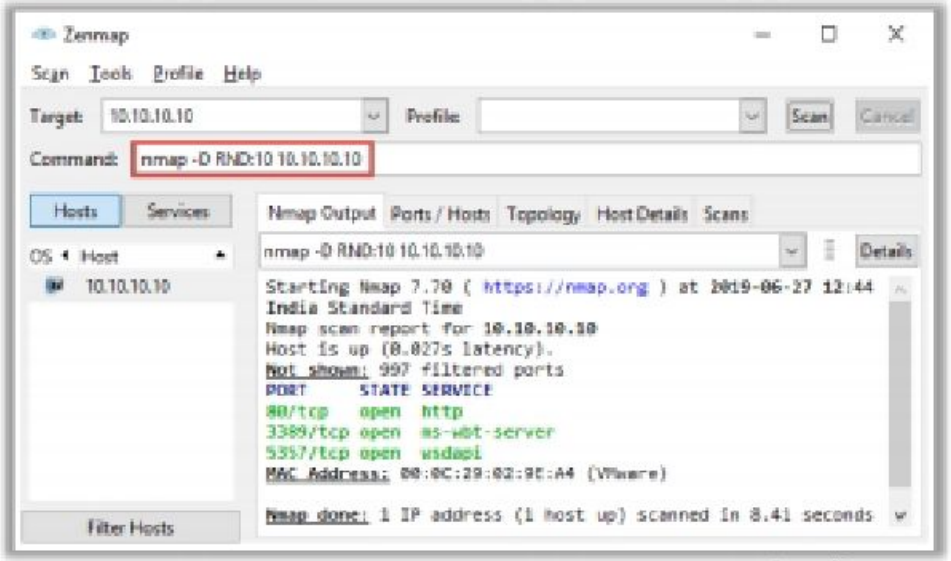
IP Address Decoy

- 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)



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

IP spoofing using Hping3:

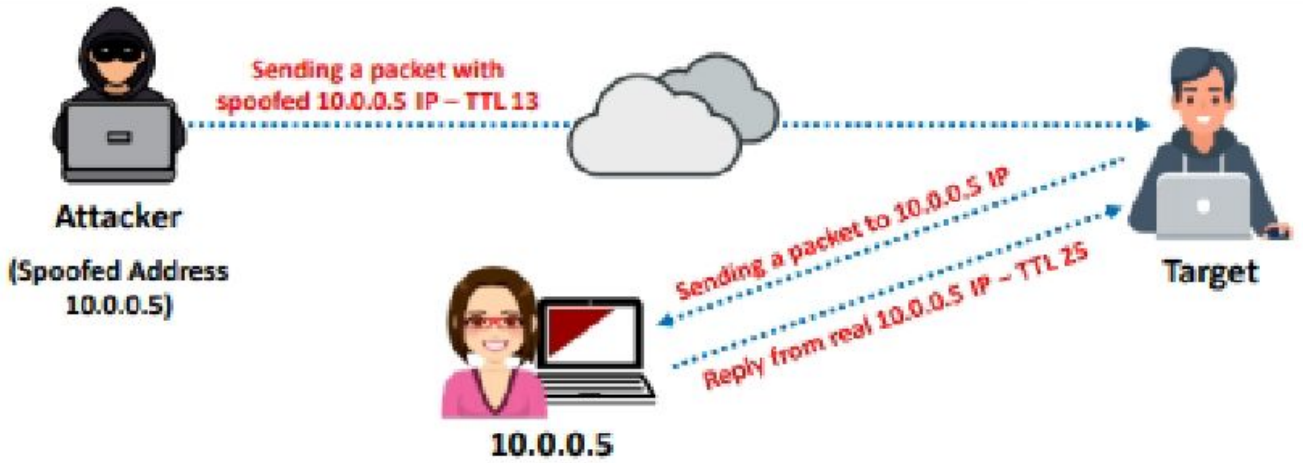
```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```

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

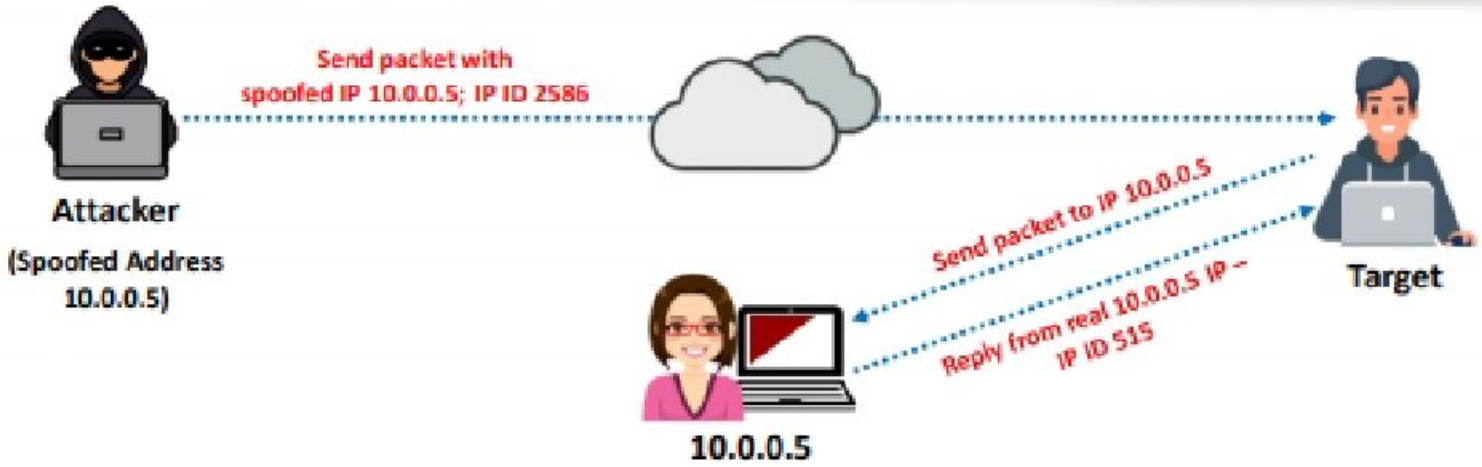
2. This technique is successful when the attacker is in a **different subnet** from that of the victim



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

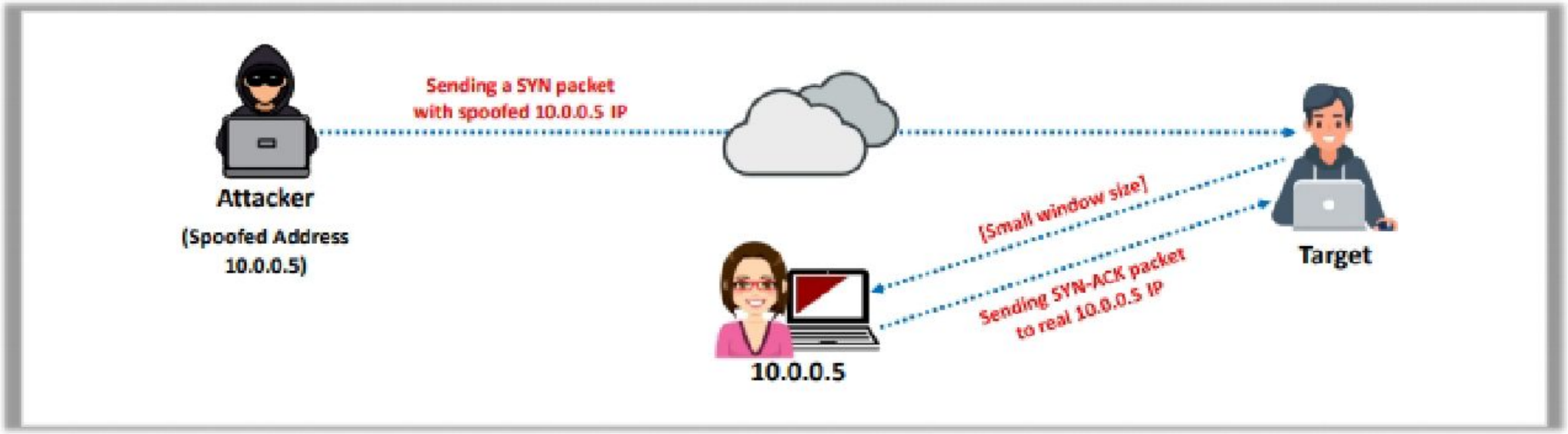
IP Spoofing Detection Techniques: IP Identification Number

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**



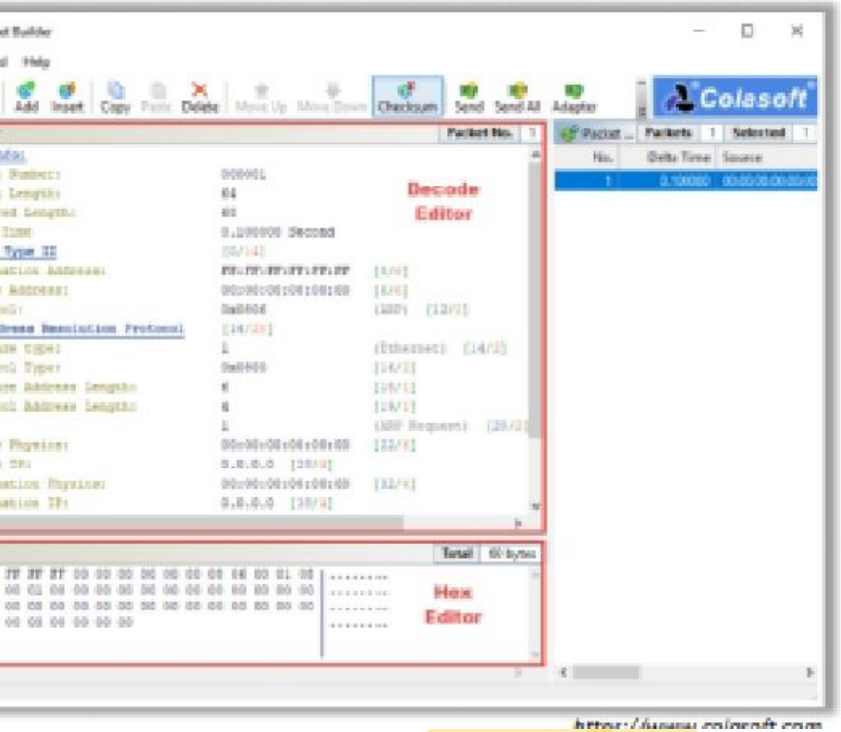
IP Spoofing Countermeasures

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

ating Custom Packets

ing Custom Packets by using Packet Crafting Tools

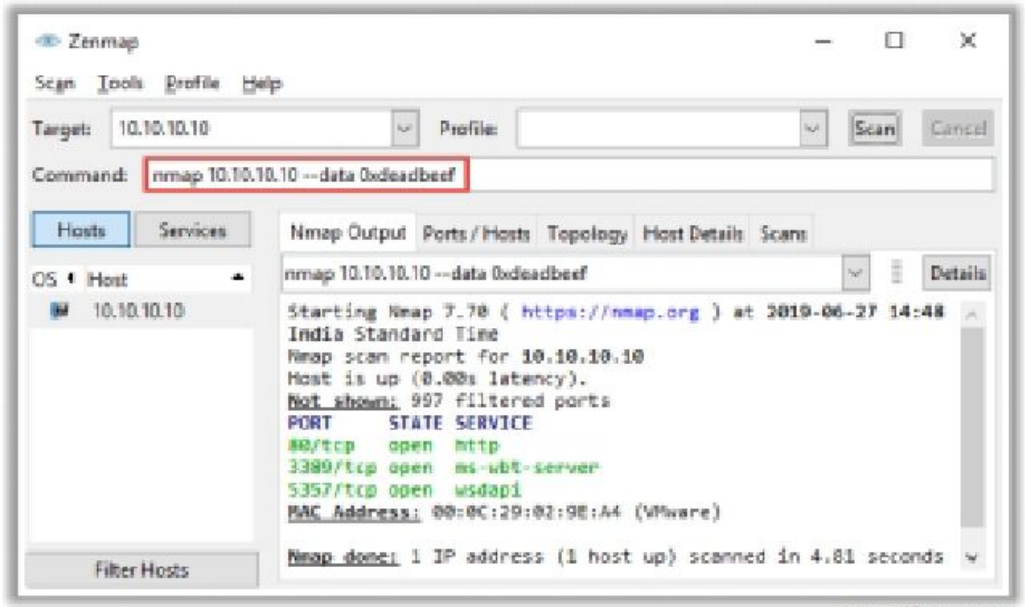
ackers **create custom TCP packets** using various
cket crafting tools like **Colasoft Packet Builder**,
tScanTools Pro, etc. to scan a target beyond a firewall



https://www.colasoft.com

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 0xdeadbeef**

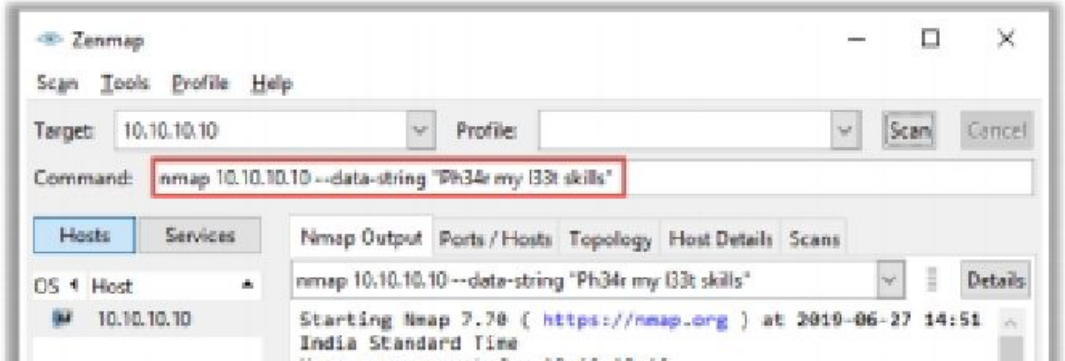


https://nmap.org

Creating Custom Packets (Cont'd)

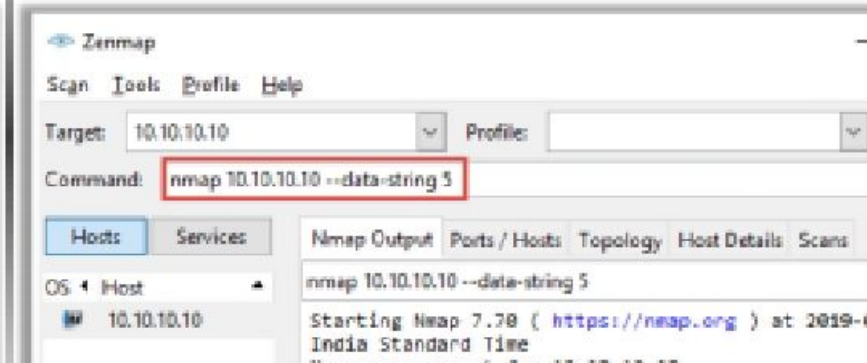
Creating Custom Packets by Appending Custom String

- 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"**



Creating Custom Packets by Appending Rand

- Attackers **append a number of random data byte** most of the packets sent without any protocol-sp payloads
- Example: **--data-string 5**



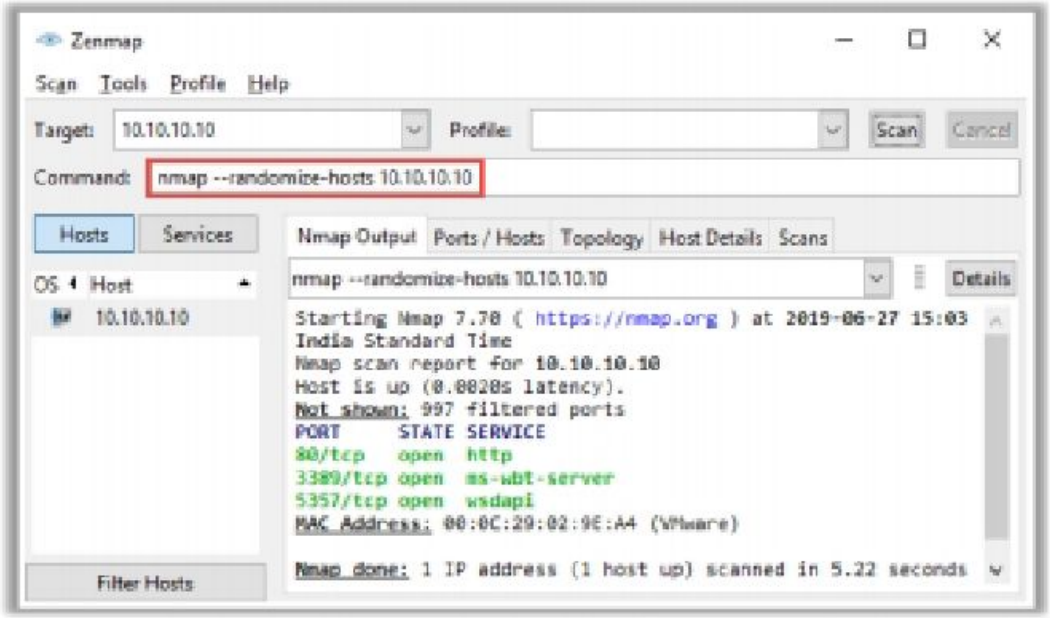
Randomizing Host Order and Sending Bad Checksums

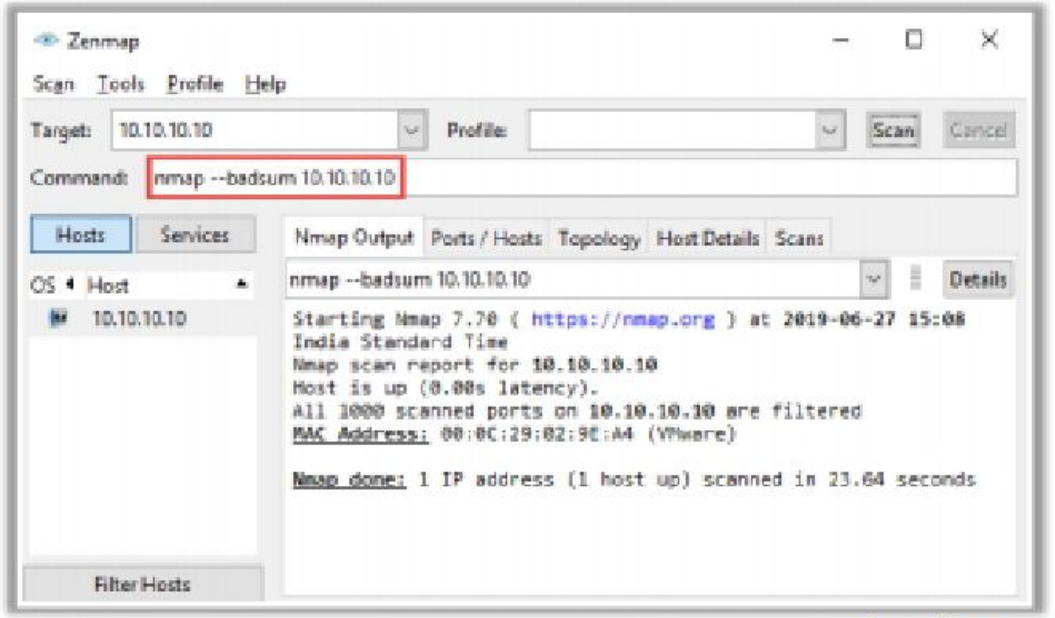
Randomizing Host Order

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

Sending Bad Checksums

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





https://nmap.org

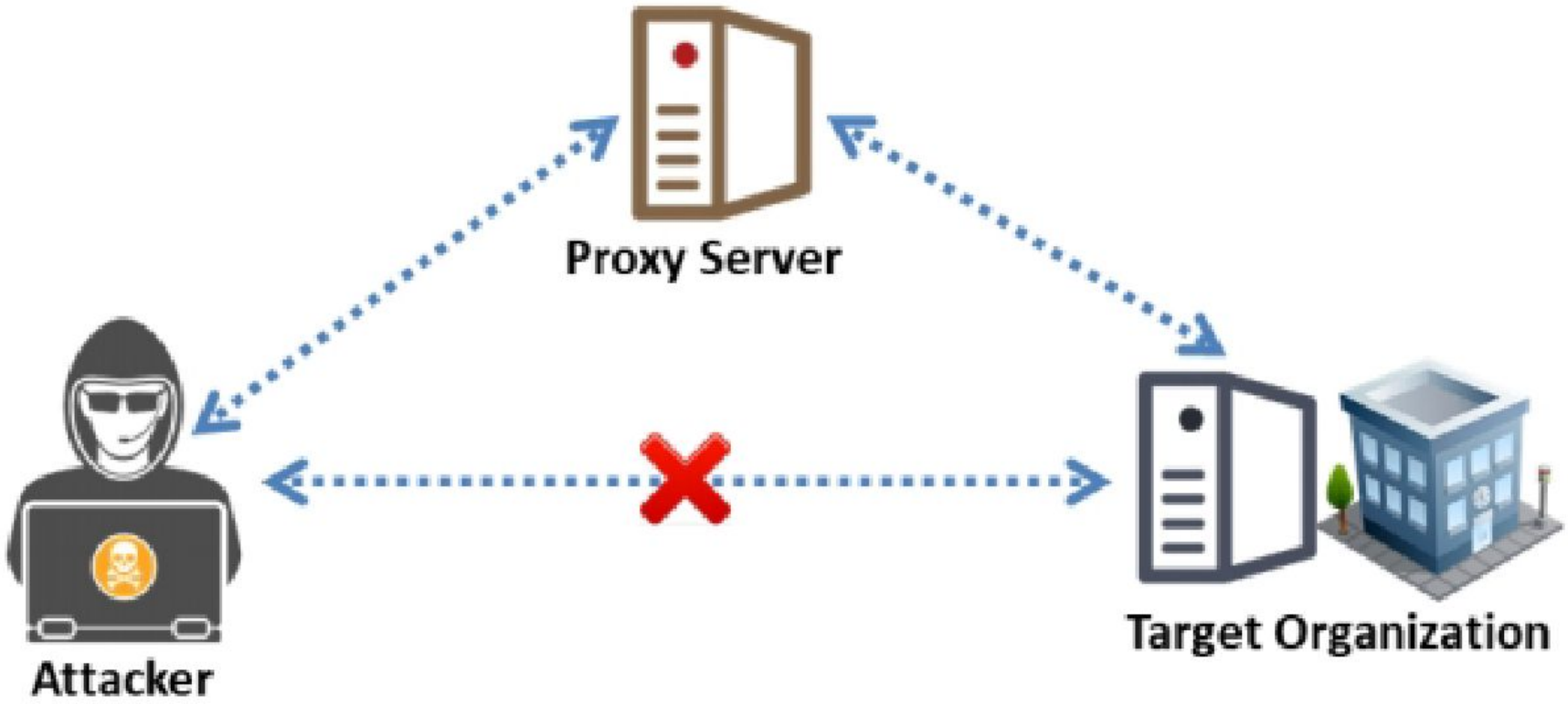
Proxy Servers

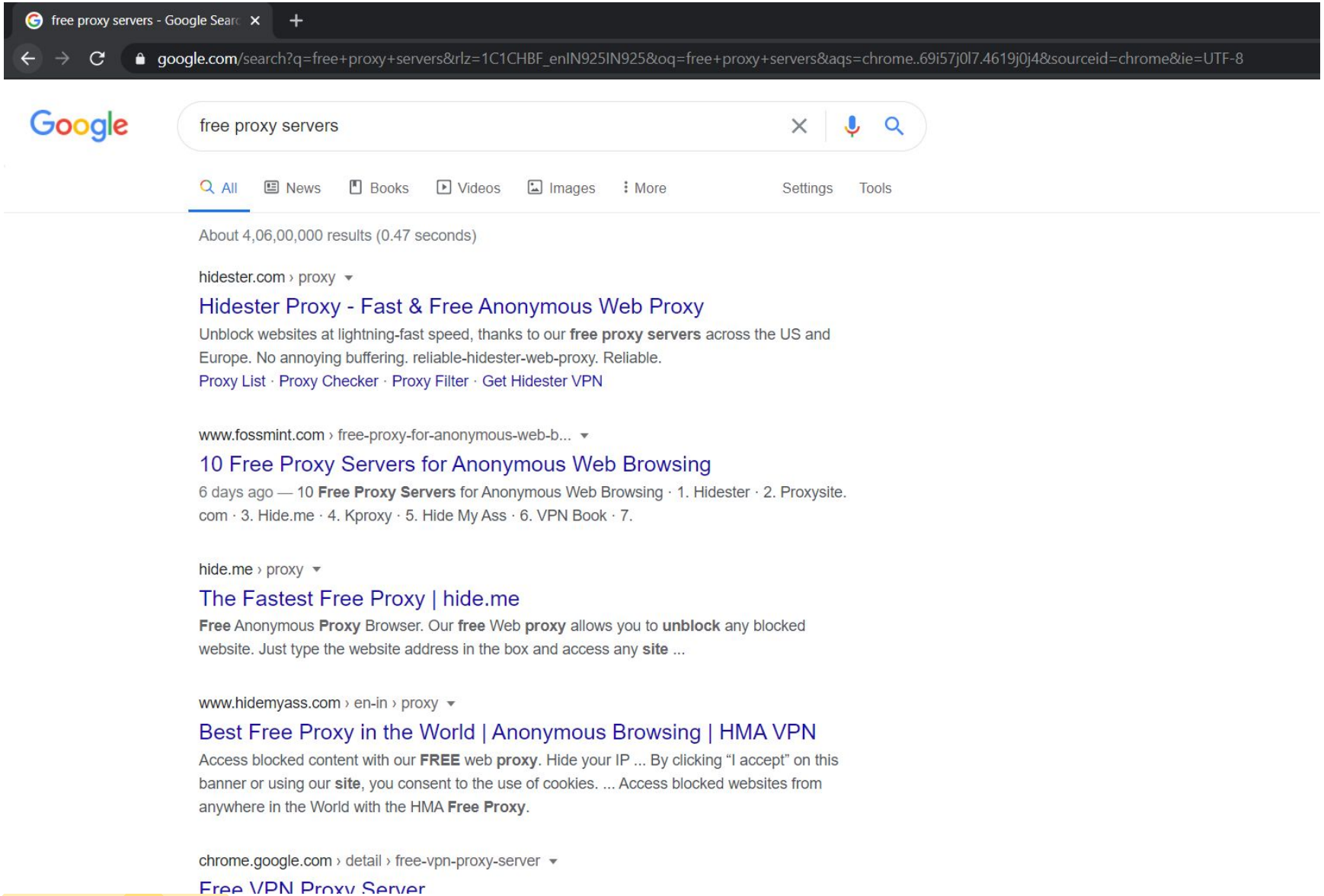
A proxy server is an application that can **serve as an intermediary** for connecting with other computers

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

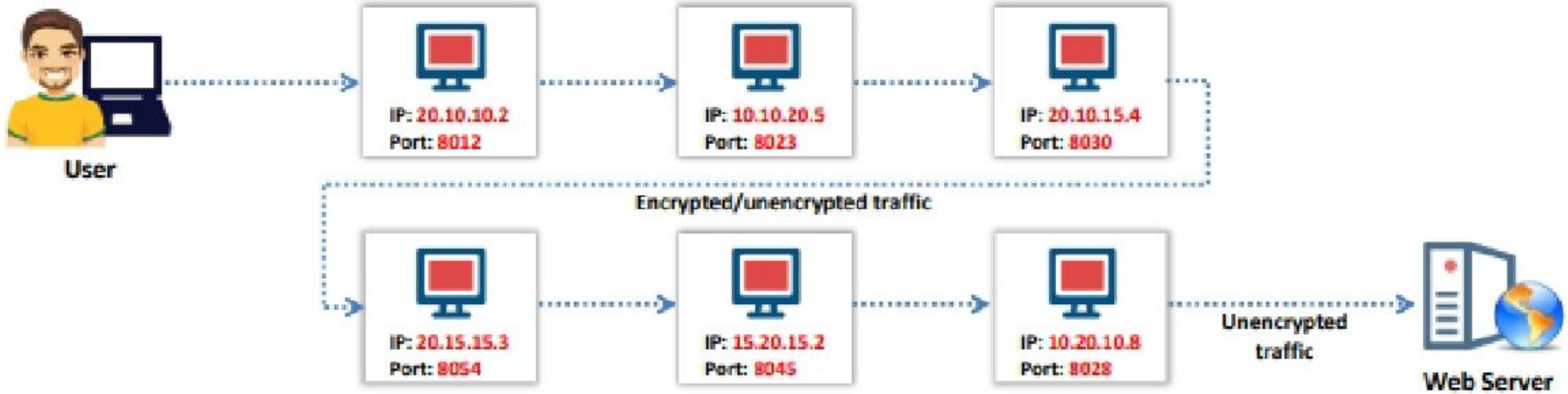
Note: A search in **Google** will list thousands of **free proxy servers**


Proxy Server
Attacker
Target Organization


free proxy servers - Google Sear
google.com/search?q=free+proxy+servers&rlz=1C1CHBF_enIN925IN925&oq=free+proxy+servers&aqs=chrome..69i57j0l7.4619j0j4&sourceid=chrome&ie=UTF-8
Google
free proxy servers
All
News
Books
Videos
Images
More
Settings
Tools
About 4,06,00,000 results (0.47 seconds)
hidester.com › proxy
Hidester Proxy - Fast & Free Anonymous Web Proxy
Unblock websites at lightning-fast speed, thanks to our free proxy servers across the US and Europe. No annoying buffering. reliable-hidester-web-proxy. Reliable.
Proxy List · Proxy Checker · Proxy Filter · Get Hidester VPN
www.fossmint.com › free-proxy-for-anonymous-web-b...
10 Free Proxy Servers for Anonymous Web Browsing
6 days ago — 10 Free Proxy Servers for Anonymous Web Browsing · 1. Hidester · 2. Proxysite.com · 3. Hide.me · 4. Kproxy · 5. Hide My Ass · 6. VPN Book · 7.
hide.me › proxy
The Fastest Free Proxy | hide.me
Free Anonymous Proxy Browser. Our free Web proxy allows you to unblock any blocked website. Just type the website address in the box and access any site ...
www.hidemyass.com › en-in › proxy
Best Free Proxy in the World | Anonymous Browsing | HMA VPN
Access blocked content with our FREE web proxy. Hide your IP ... By clicking "I accept" on this banner or using our site, you consent to the use of cookies. ... Access blocked websites from anywhere in the World with the HMA Free Proxy.
chrome.google.com › detail › free-vpn-proxy-server
Free VPN Proxy Server

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 Chaining

Proxy chaining helps an attacker to increase his/her Internet anonymity. Internet anonymity depends on the number of proxies used for fetching the target application; the larger the number of proxy servers used, the greater is the attacker's anonymity.

The proxy chaining process is described below:

- The user requests a resource from the destination.
- A proxy client in the user's system connects to a proxy server and passes the request to the proxy server.
- The proxy server strips the user's identification information and passes the request to the next proxy server.
- This process is repeated by all the proxy servers in the chain.
- Finally, the unencrypted request is passed to the web server.

Configure proxy chains using tor

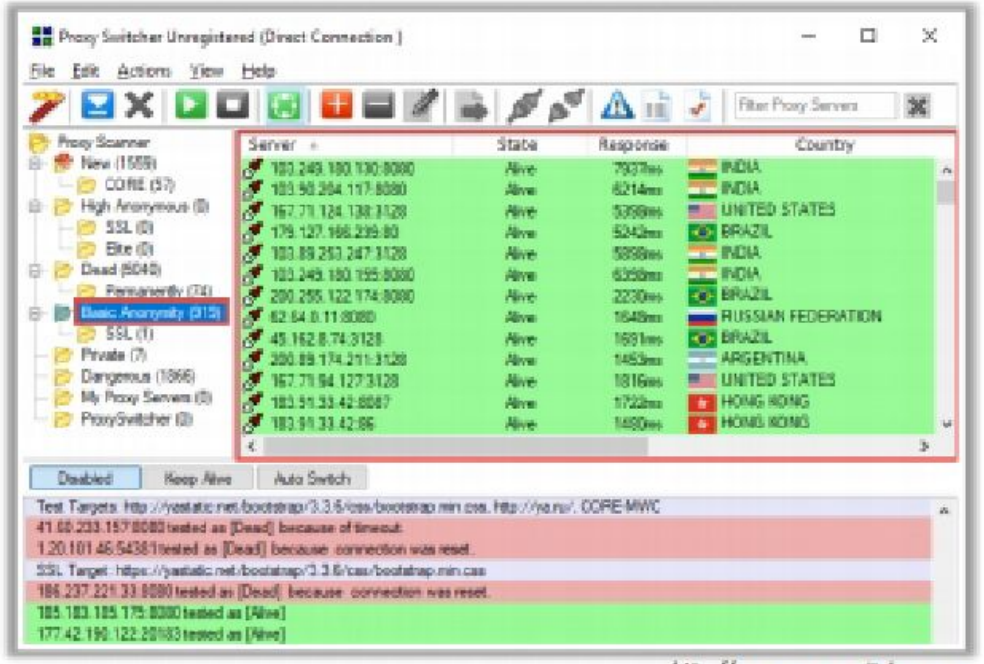
Install tor package – apt install tor
Service tor start / systemctl start tor
Service tor status
Edit the config file of proxychains
Vim /etc/proxychains.config
Remove Dynamic chain from comment
comment Strict chain and Random chain
Remove proxy DNS from comment
write socks5 127.0.0.1 9050 in last line of proxy list

Service tor restart
Proxychains firefox dnsleaktest.com

Proxy Tools

Proxy Switcher

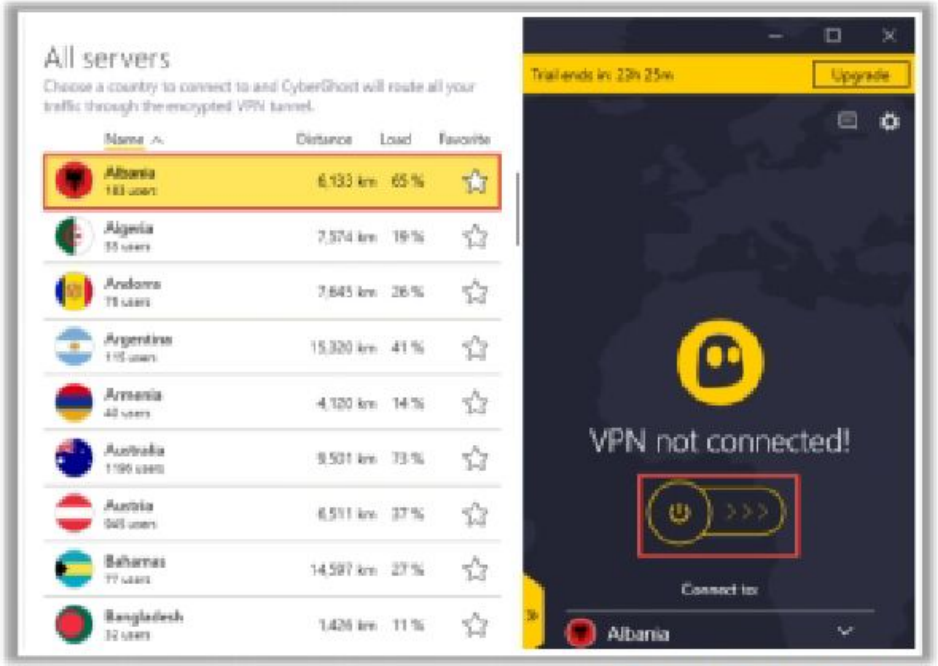
Proxy Switcher allows you to **surf anonymously on the Internet** without disclosing your IP address



http://www.proxyswitcher.com

CyberGhost VPN

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



https://www.cyberghostvpn.com

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

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?

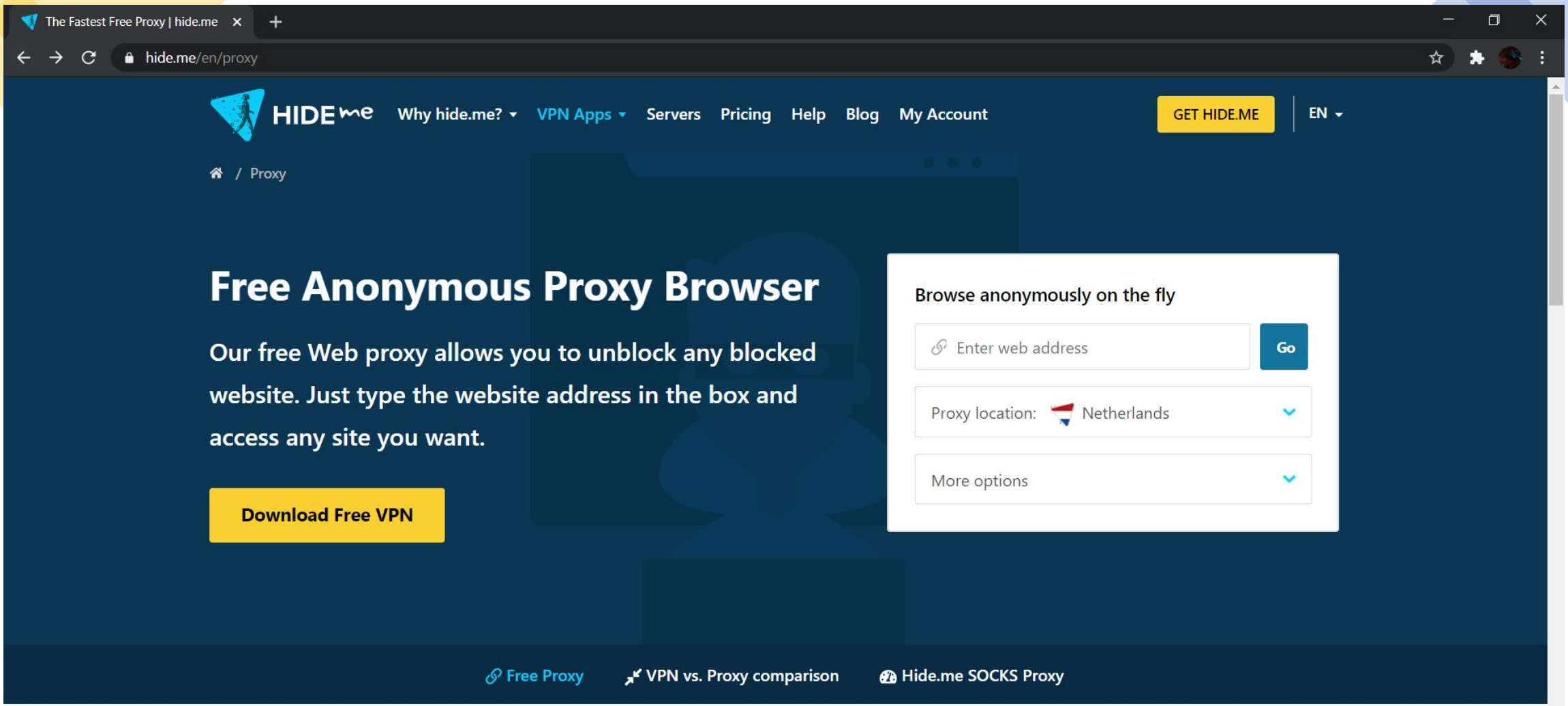
1. **Privacy and anonymity**
2. **Protection against online attacks**
3. **Access restricted content**
4. **Bypass IDS and Firewall rules**

https://hide.me/en/proxy

Anonymizers

An anonymizer is an intermediate server placed between you as the end user and the website to access the website on your behalf and make your web surfing activities untraceable. Anonymizers allows you to bypass Internet censors. An anonymizer eliminates all the identifying information (IP address) from your system while you are surfing the Internet, thereby ensuring privacy. Most anonymizers can anonymize the web (HTTP:), file transfer protocol (FTP:), and gopher (gopher:) Internet services.

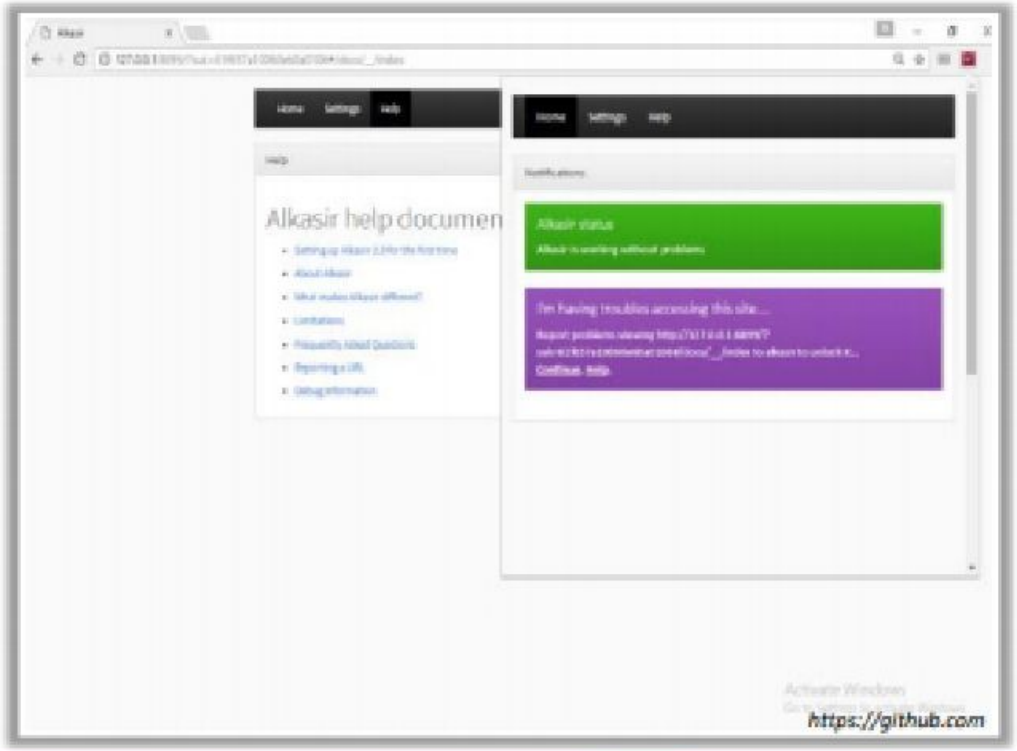
To visit a page anonymously, you can visit your preferred anonymizer site and enter the name of the target website in the anonymization field. Alternatively, you can set your browser home page to point to an anonymizer to anonymize subsequent web access. In addition, you can choose to anonymously provide passwords and other information to sites without revealing any additional information, such as your IP address. Attackers may configure an anonymizer as a permanent proxy server by making the site name the setting for the HTTP, FTP, Gopher, and other proxy options in their application configuration menu, thereby cloaking their malicious activities.


The Fastest Free Proxy | hide.me
hide.me/en/proxy
HIDE me
Why hide.me?
VPN Apps
Servers
Pricing
Help
Blog
My Account
GET HIDE.ME
EN
/ Proxy
Free Anonymous Proxy Browser
Our free Web proxy allows you to unblock any blocked website. Just type the website address in the box and access any site you want.
Download Free VPN
Browse anonymously on the fly
Enter web address
Go
Proxy location: Netherlands
More options
Free Proxy
VPN vs. Proxy comparison
Hide.me SOCKS Proxy

Censorship Circumvention Tools: Alkasir and Tails

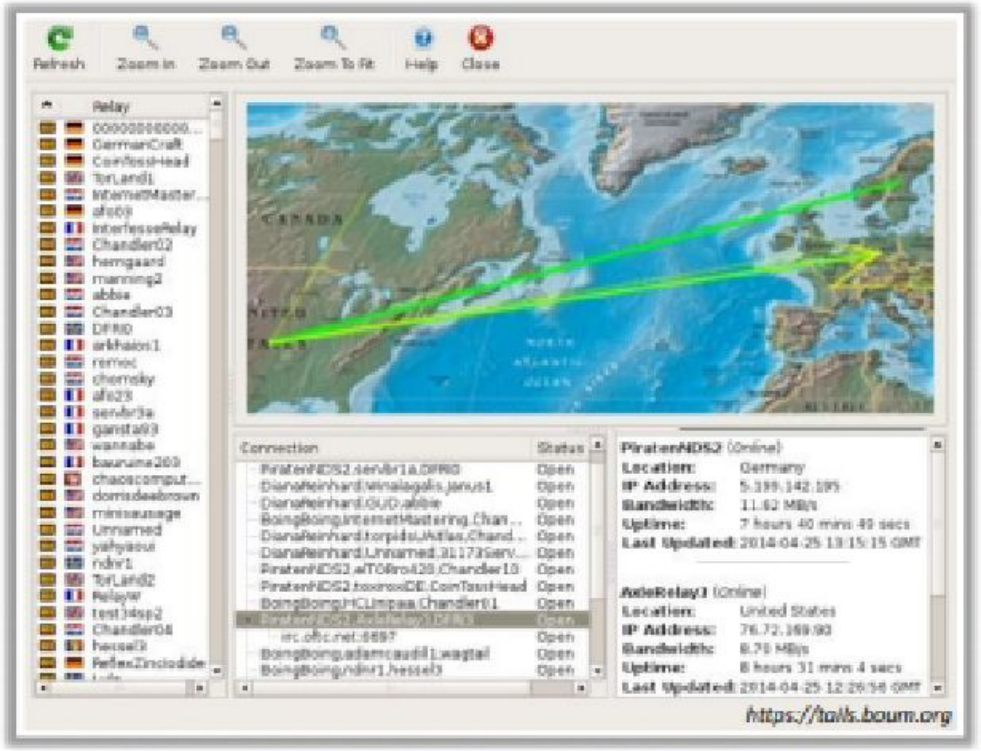
Alkasir

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



Tails

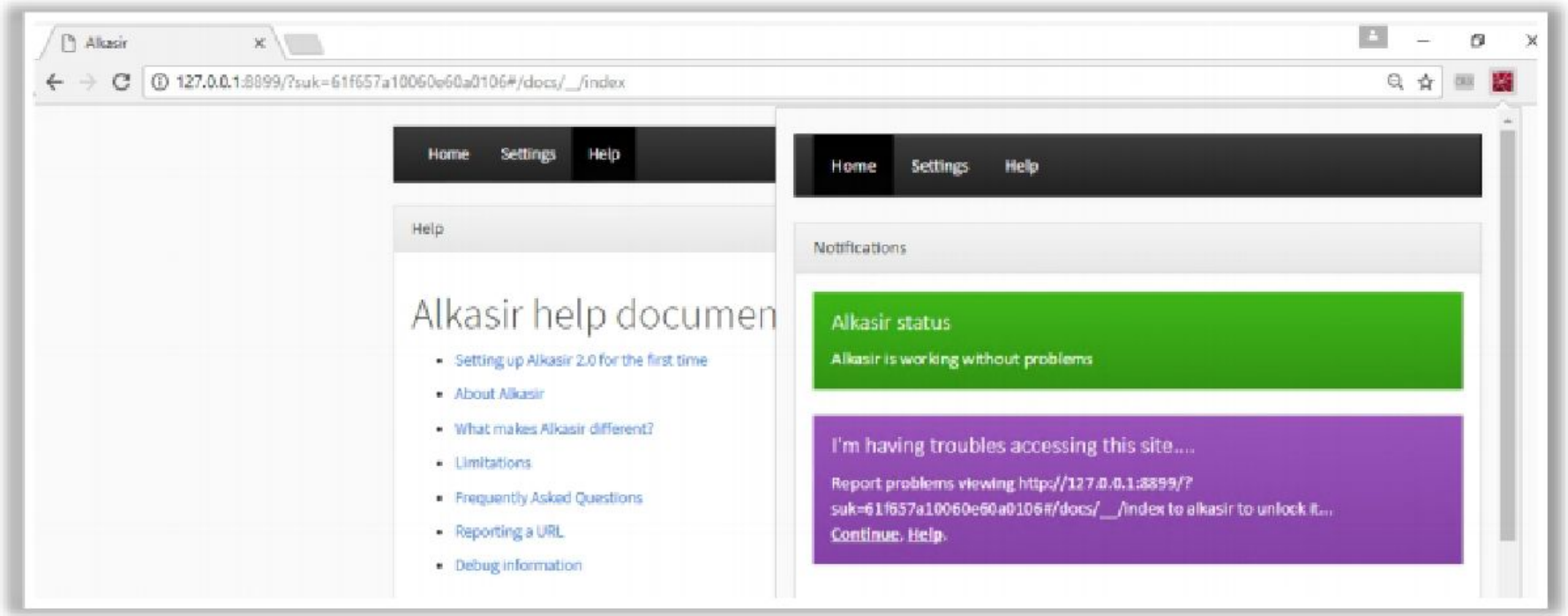
Tails is a **live operating system** that a user can start on any computer from a DVD, USB stick, or SD card



- **Alkasir**

  Source: *https://github.com*

  Alkasir is a cross-platform, open-source, and robust website censorship circumvention tool that also maps censorship patterns around the world. Alkasir enables attackers to identify censored links. It keeps them informed about links that are still blocked and links that are not blocked.



- **Tails**

  Source: *https://tails.boum.org*

  Tails is a live OS that users can run on any computer from a DVD drive, USB stick, or SD card. It uses state-of-the-art cryptographic tools to encrypt files, emails, and instant messaging. It allows attackers to use the Internet anonymously and circumvent censorship. It leaves no trace on the computer.

Anonymizers for Mobile

Orbot

https://guardianproject.info

Psiphon

https://psiphon.ca

OpenDoor

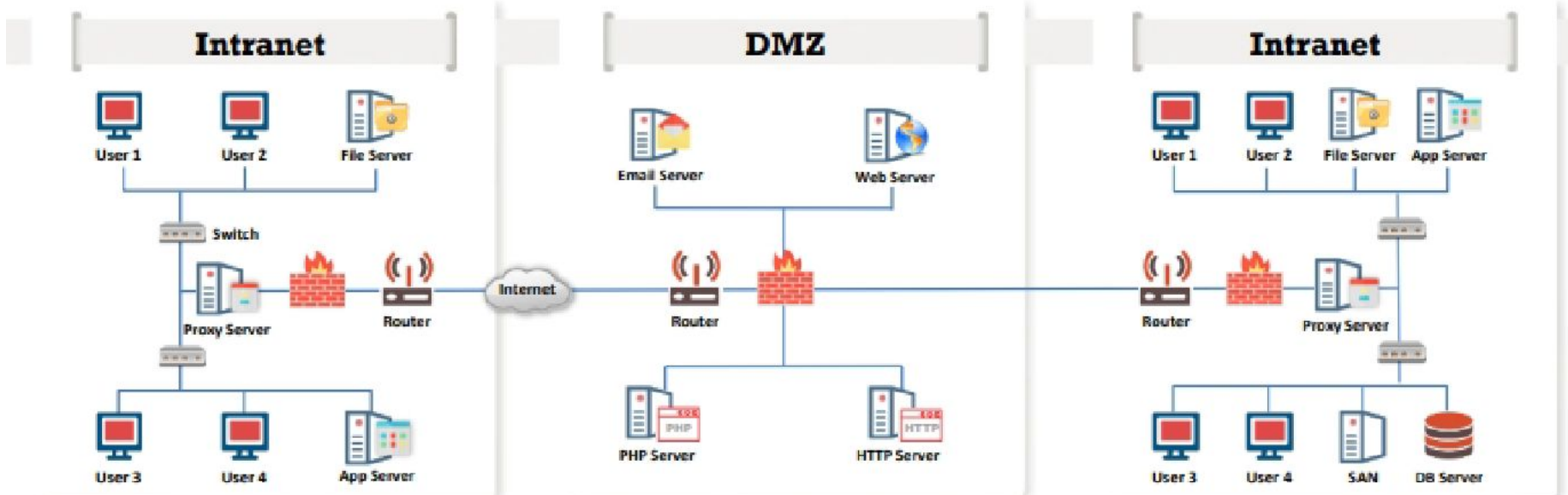
https://www.apple.com

Draw Network Diagrams

A network diagram helps in analyzing the complete network topology. This section highlights the importance of network diagrams, how to draw them, how an attacker uses them to launch an attack, and the tools used for drawing them.

Drawing Network Diagrams

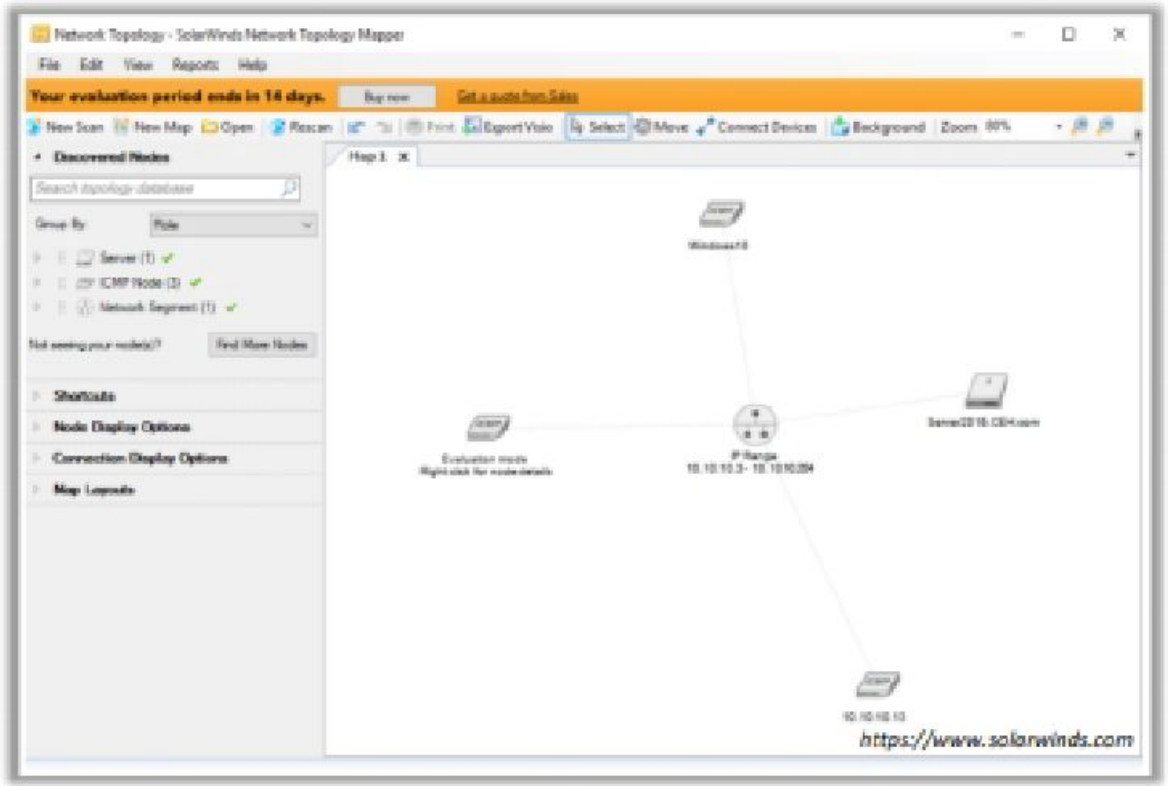
- A diagram of a target network provides an attacker with valuable information about the **network and its architecture**
- Network diagrams show **logical or physical paths** to a potential target



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