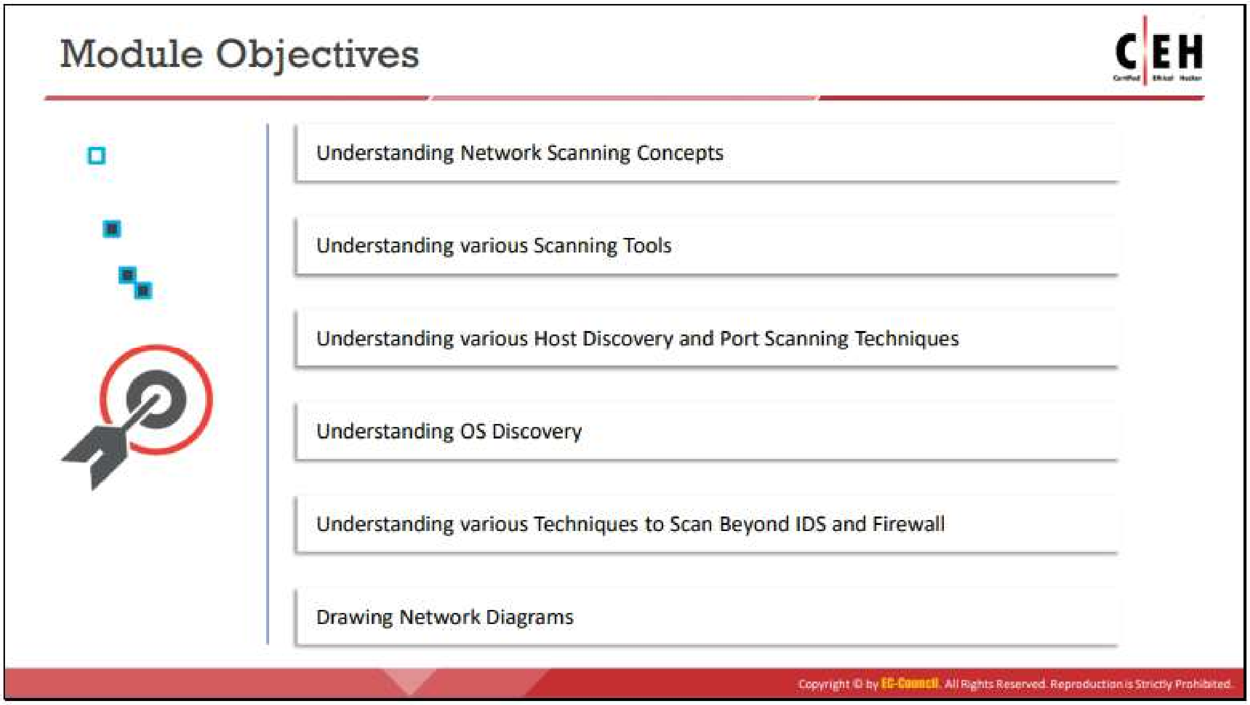
C|EH™

Certified Ethical Hacker

Module 03:
Scanning Networks



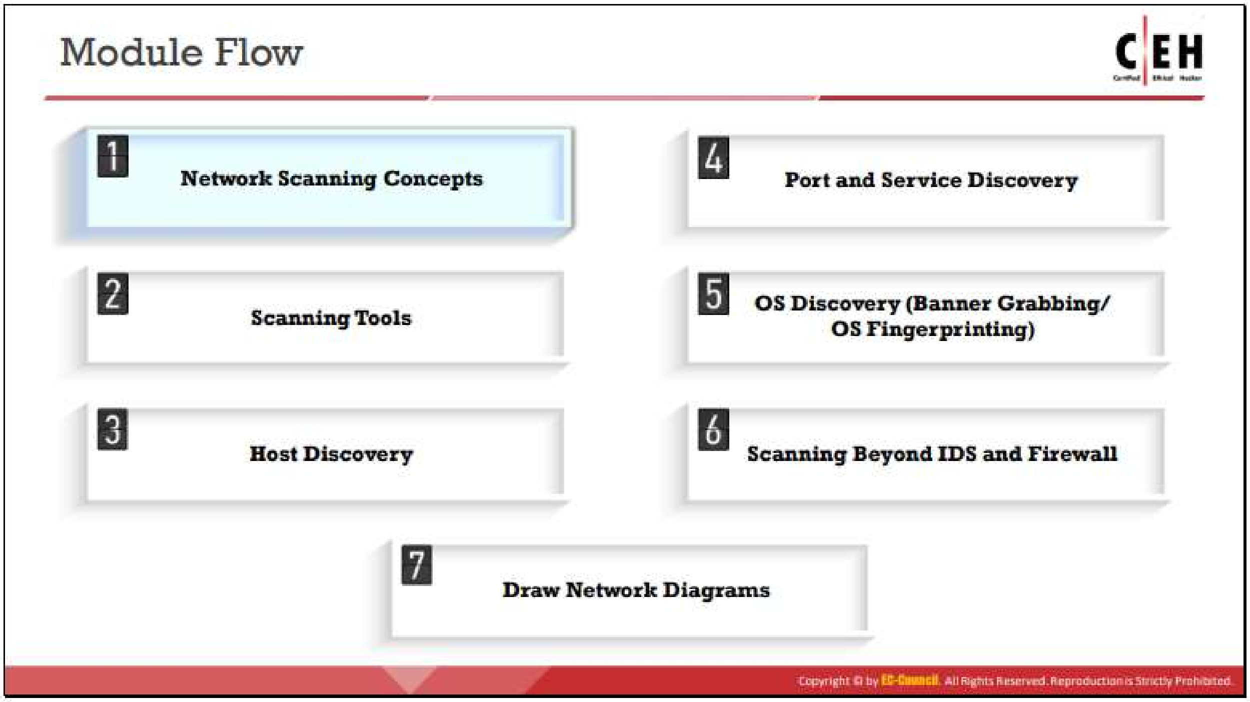
Module Objectives

After identifying the target and performing the initial reconnaissance, as discussed in the Footprinting and Reconnaissance module, attackers begin to search for an entry point into the target system. Attackers should determine whether the target systems are active or inactive to reduce the time spent on scanning. Notably, the scanning itself is not the actual intrusion but an extended form of reconnaissance in which the attacker learns more about his/her target, including information about OSs, services, and any configuration lapses. The information gleaned from such reconnaissance helps the attacker select strategies for attacking the target system or network.

This module starts with an overview of network scanning and provides insights into various host discovery techniques that can be used to check for live and active systems. Furthermore, it discusses various port and service discovery techniques, operating system discovery techniques, and techniques for scanning beyond IDS and firewalls. Finally, it ends with an overview of drawing network diagrams.

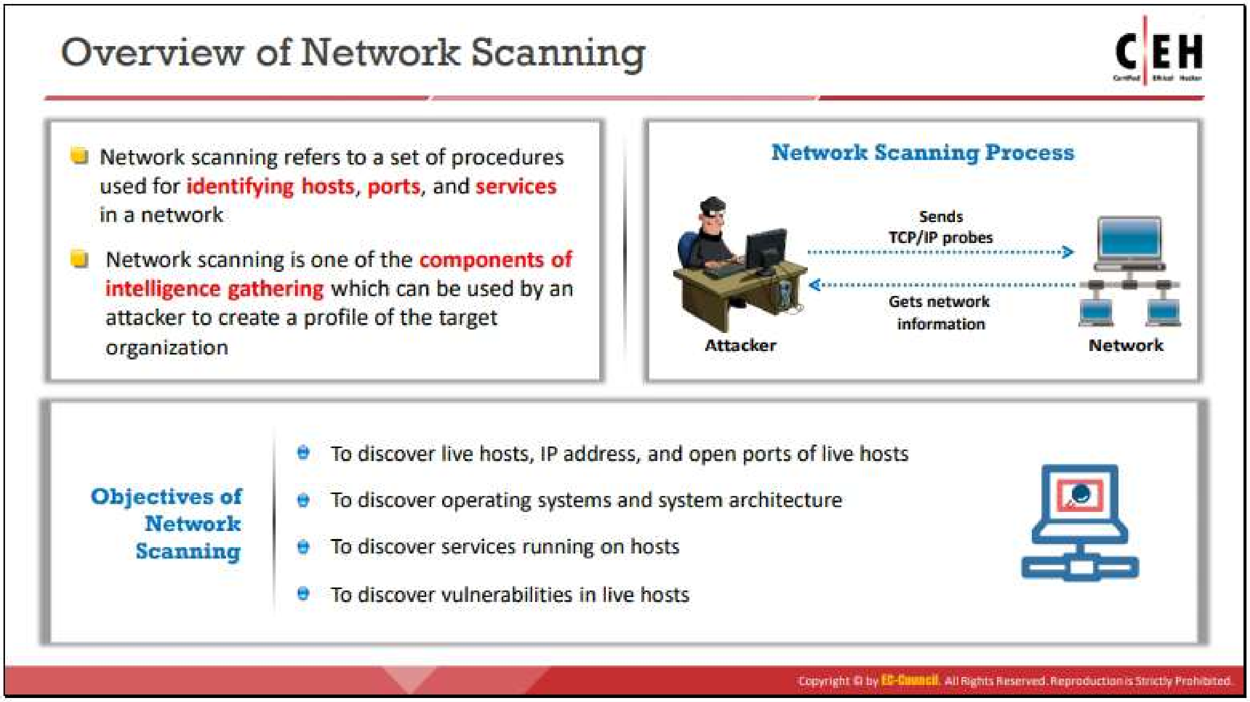
At the end of this module, you will be able to:

- Describe the network scanning concepts
- Use various scanning tools
- Perform host discovery to check for live systems
- Perform port and service discovery using various scanning techniques
- Scan beyond intrusion detection systems (IDS) and firewalls
- Perform operating system (OS) discovery
- Draw network diagrams using network discovery tools



Network Scanning Concepts

As already discussed, footprinting is the first phase of hacking, in which the attacker gains primary information about a potential target. He/she then uses this information in the scanning phase to gather more details about the target.



Overview of Network Scanning

Scanning is the process of gathering additional detailed information about the target using highly complex and aggressive reconnaissance techniques. Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network. Network scanning is also used for discovering active machines in a network and identifying the OS running on the target machine. It is one of the most important phases of intelligence gathering for an attacker, which enables him/her to create a profile of the target organization. In the process of scanning, the attacker tries to gather information, including the specific IP addresses that can be accessed over the network, the target's OS and system architecture, and the ports along with their respective services running on each computer.

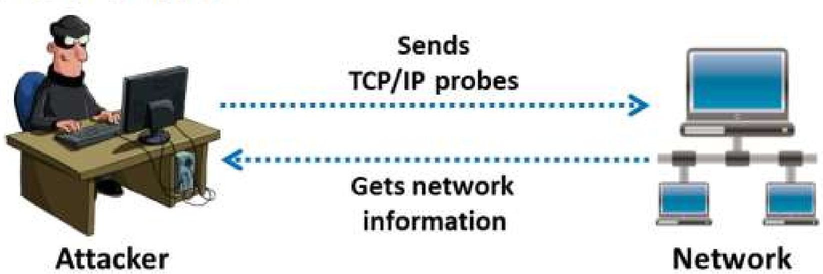


Figure 3.1: Network scanning process

The purpose of scanning is to discover exploitable communications channels, probe as many listeners as possible, and track the ones that are responsive or useful to an attacker's particular needs. In the scanning phase of an attack, the attacker tries to find various ways to intrude into a target system. The attacker also tries to discover more information about the target system to determine the presence of any configuration lapses. The attacker then uses the information obtained to develop an attack strategy.

Types of Scanning

- **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.

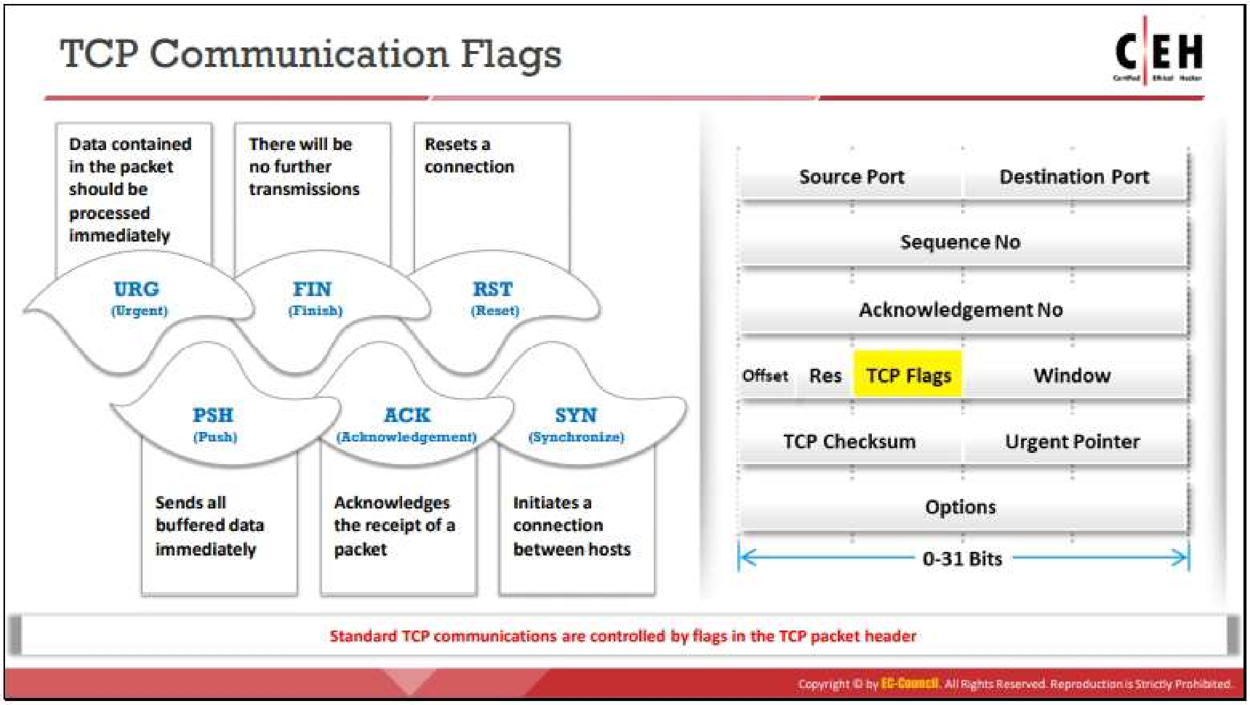
A thief who wants to break into a house looks for access points such as doors and windows. These are usually the house's points of vulnerability, as they are easily accessible. When it comes to computer systems and networks, ports are the doors and windows of a system that an intruder uses to gain access. 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.

Objectives of Network Scanning

The more the information at hand about a target organization, the higher are the chances of knowing a network's security loopholes, and, consequently, for gaining unauthorized access to it.

Some objectives for scanning a network are as follows:

- Discover the network's live hosts, IP addresses, and open ports of the live hosts. Using the open ports, the attacker will determine the best means of entering into the system.
- Discover the OS and system architecture of the target. This is also known as fingerprinting. An attacker can formulate an attack strategy based on the OS's vulnerabilities.
- Discover the services running/listening on the target system. Doing so gives the attacker an indication of the vulnerabilities (based on the service) that can be exploited for gaining access to the target system.
- Identify specific applications or versions of a particular service.
- Identify vulnerabilities in any of the network systems. This helps an attacker to compromise the target system or network through various exploits.



TCP Communication Flags

The TCP header contains various flags that control the transmission of data across a TCP connection. Six TCP control flags manage the connection between hosts and give instructions to the system. Four of these flags (SYN, ACK, FIN, and RST) govern the establishment, maintenance, and termination of a connection. The other two flags (PSH and URG) provide instructions to the system. The size of each flag is 1 bit. As there are six flags in the TCP Flags section, the size of this section is 6 bits. When a flag value is set to "1," that flag is automatically turned on.

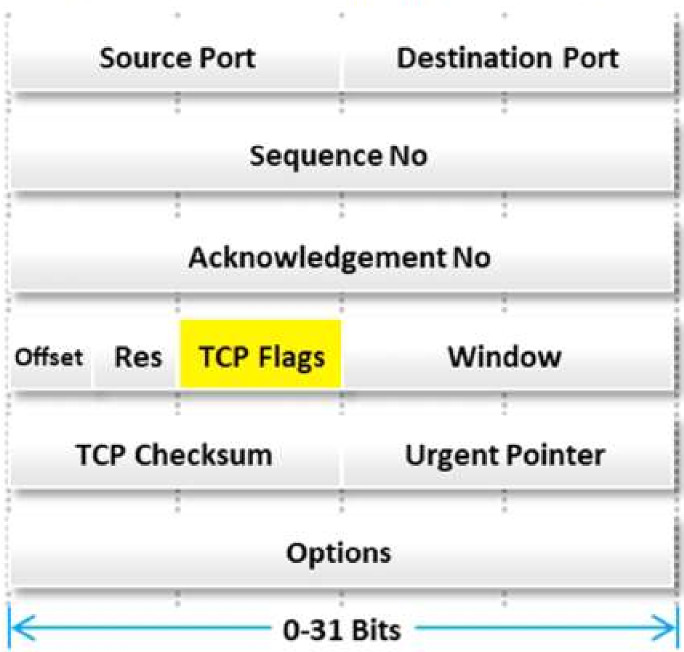


Figure 3.2: TCP header format

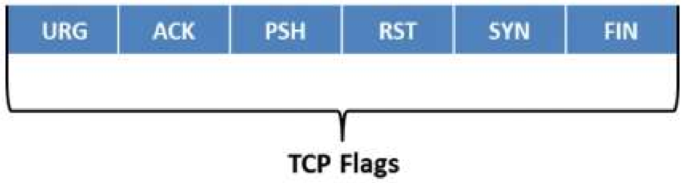
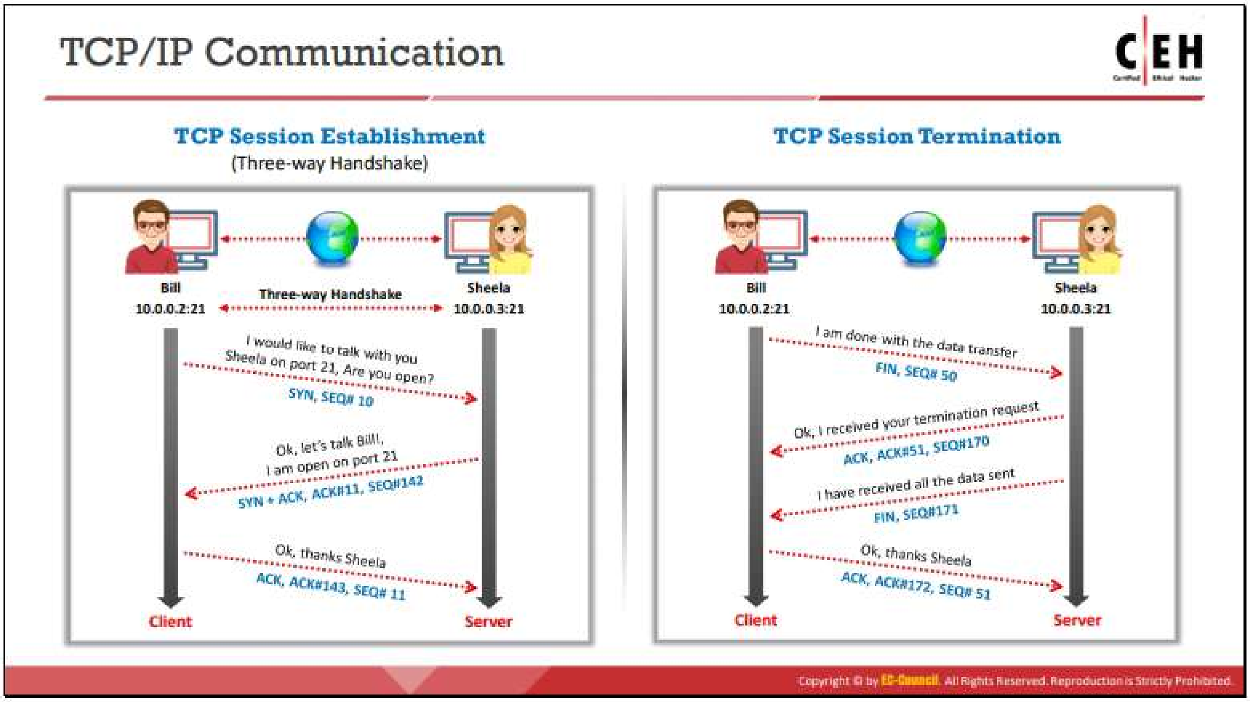


Figure 3.3: TCP communication flags

The following are the TCP communication flags:

- Synchronize or "SYN": It notifies the transmission of a new sequence number. This flag generally represents the establishment of a connection (three-way handshake) between two hosts.
- Acknowledgement or "ACK": It confirms the receipt of the transmission and identifies the next expected sequence number. When the system successfully receives a packet, it sets the value of its flag to "1," thus implying that the receiver should pay attention to it.
- Push or "PSH": When it is set to "1," it indicates that the sender has raised the push operation to the receiver; this implies that the remote system should inform the receiving application about the buffered data coming from the sender. The system raises the PSH flag at the start and end of data transfer and sets it on the last segment of a file to prevent buffer deadlocks.
- Urgent or "URG": It instructs the system to process the data contained in packets as soon as possible. When the system sets the flag to "1," priority is given to processing the urgent data first and all the other data processing is stopped.
- Finish or "FIN": It is set to "1" to announce that no more transmissions will be sent to the remote system and the connection established by the SYN flag is terminated.
- Reset or "RST": When there is an error in the current connection, this flag is set to "1" and the connection is aborted in response to the error. Attackers use this flag to scan hosts and identify open ports.

SYN scanning mainly deals with three flags: SYN, ACK, and RST. You can use these three flags for gathering illegal information from servers during enumeration.



TCP/IP Communication

TCP is connection oriented, i.e., it prioritizes connection establishment before data transfer between applications. This connection between protocols is possible through the three-way handshake.

A TCP session initiates using a three-way handshake mechanism:

- To launch a TCP connection, the source (10.0.0.2:21) sends a SYN packet to the destination (10.0.0.3:21).
- On receiving the SYN packet, the destination responds by sending a SYN/ACK packet back to the source.
- The ACK packet confirms the arrival of the first SYN packet to the source.
- Finally, the source sends an ACK packet for the ACK/SYN packet transmitted by the destination.
- This triggers an "OPEN" connection, thereby allowing communication between the source and destination, which continues until one of them issues a "FIN" or "RST" packet to close the connection.

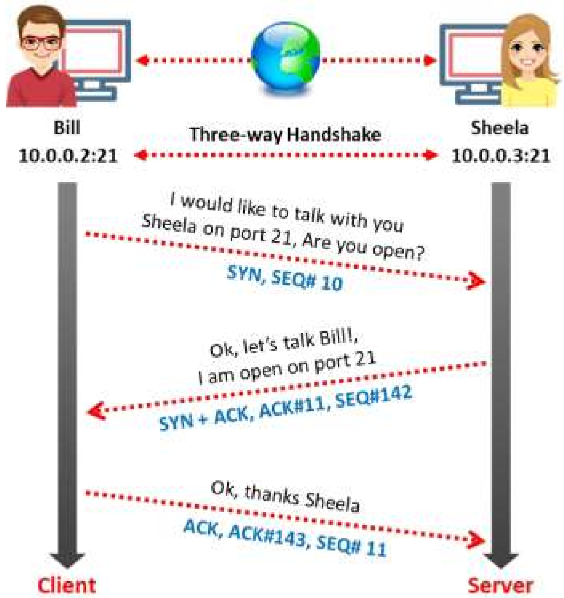


Figure 3.4: TCP session establishment

The TCP protocol maintains stateful connections for all connection-oriented protocols throughout the Internet and works similarly to ordinary telephone communication, in which one picks up a telephone receiver, hears a dial tone, and dials a number that triggers ringing at the other end until someone picks up the receiver and says, "Hello."

The system terminates the established TCP session as follows:

After completing all the data transfers through the established TCP connection, the sender sends the connection termination request to the receiver through a FIN or RST packet. Upon receiving the connection termination request, the receiver acknowledges the termination request by sending an ACK packet to the sender and finally sends its own FIN packet. Then, the system terminates the established connection.

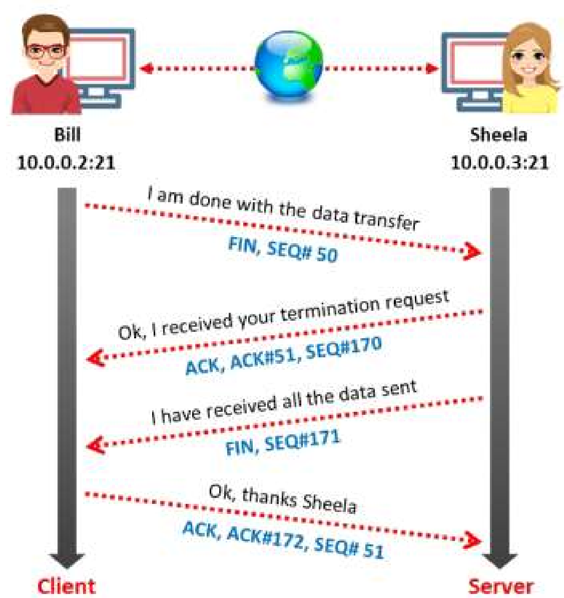


Figure 3.5: TCP session termination

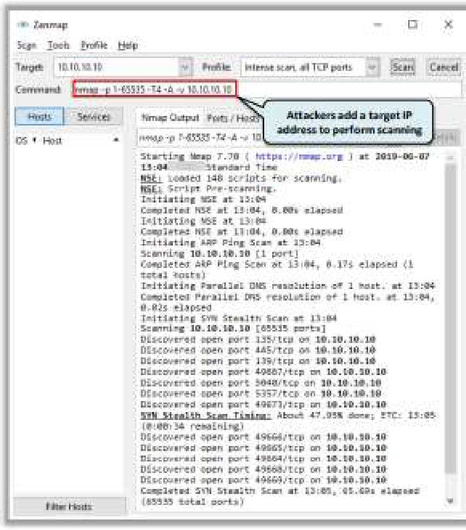
Module Flow

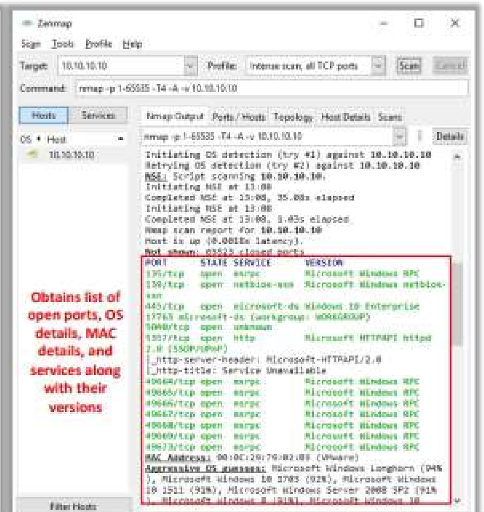
1. Network Scanning Concepts
2. **Scanning Tools**
3. Host Discovery
4. Port and Service Discovery
5. OS Discovery (Banner Grabbing/ OS Fingerprinting)
6. Scanning Beyond IDS and Firewall
7. Draw Network Diagrams

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**

NMAP





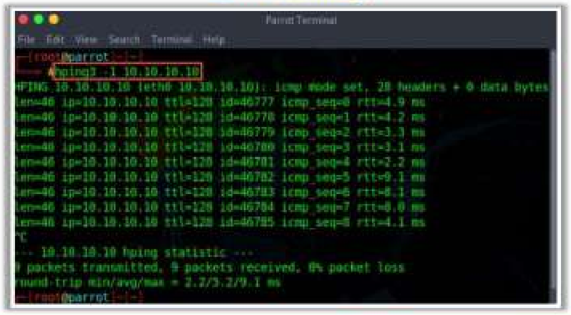
https://nmap.org

Scanning Tools: Hping2/Hping3

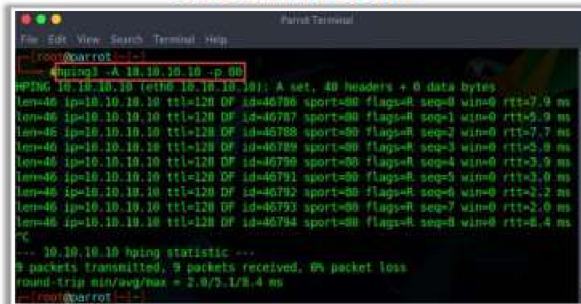
1. Command line **network scanning** and **packet crafting** tool for the TCP/IP protocol

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



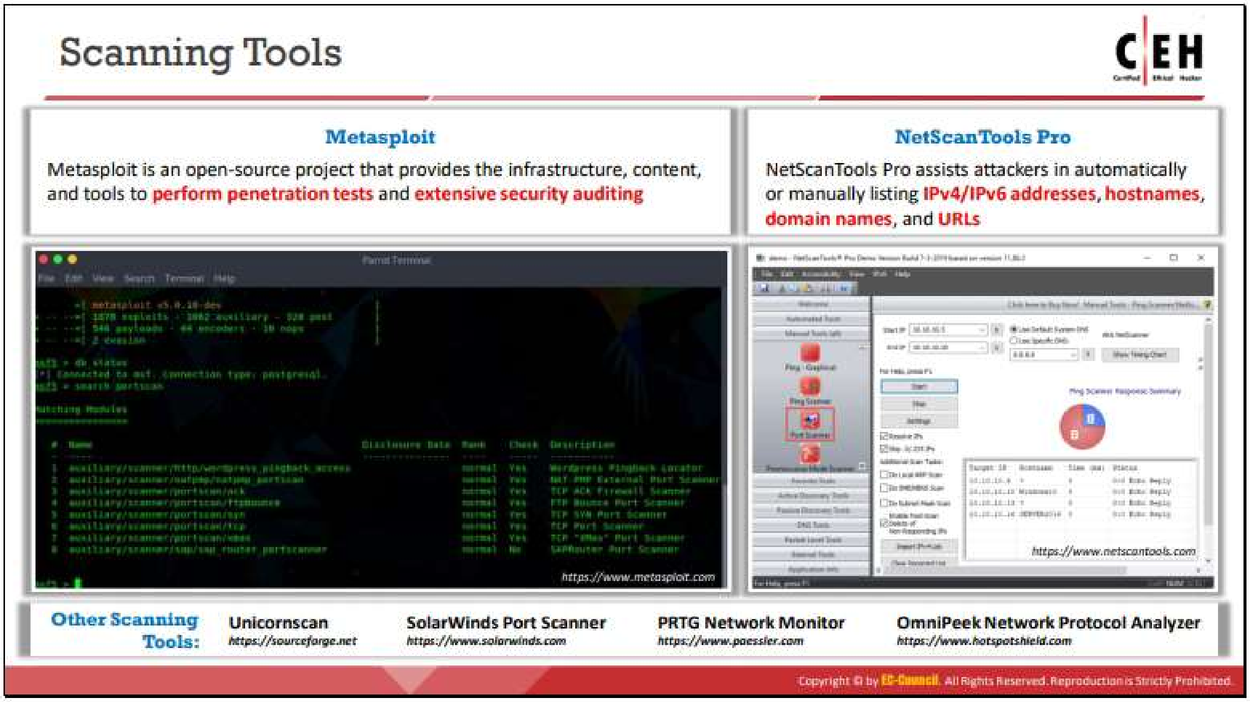
ACK Scanning on port 80



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

Scanning tools are used to scan and identify live hosts, open ports, running services on a target network, location info, NetBIOS info, and information about all TCP/IP and UDP open ports. The information obtained from these tools will help an ethical hacker in creating the profile of the target organization and scanning the network for open ports of the devices connected.

- **Nmap**

  Source: *https://nmap.org*

  Nmap ("Network Mapper") is a security scanner for network exploration and hacking. It allows you to discover hosts, ports, and services on a computer network, thus creating a "map" of the network. It sends specially crafted packets to the target host and then analyzes the responses to accomplish its goal. It scans vast networks of literally hundreds of thousands of machines. Nmap includes many mechanisms for port scanning (TCP and UDP), OS detection, version detection, ping sweeps, and so on.

  Either a network administrator or an attacker can use this tool for their specific needs. Network administrators can use Nmap for network inventory, 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), type of packet filters/firewalls, MAC details, and OSs along with their versions.

  Syntax: `# nmap <options> <Target IP address>`

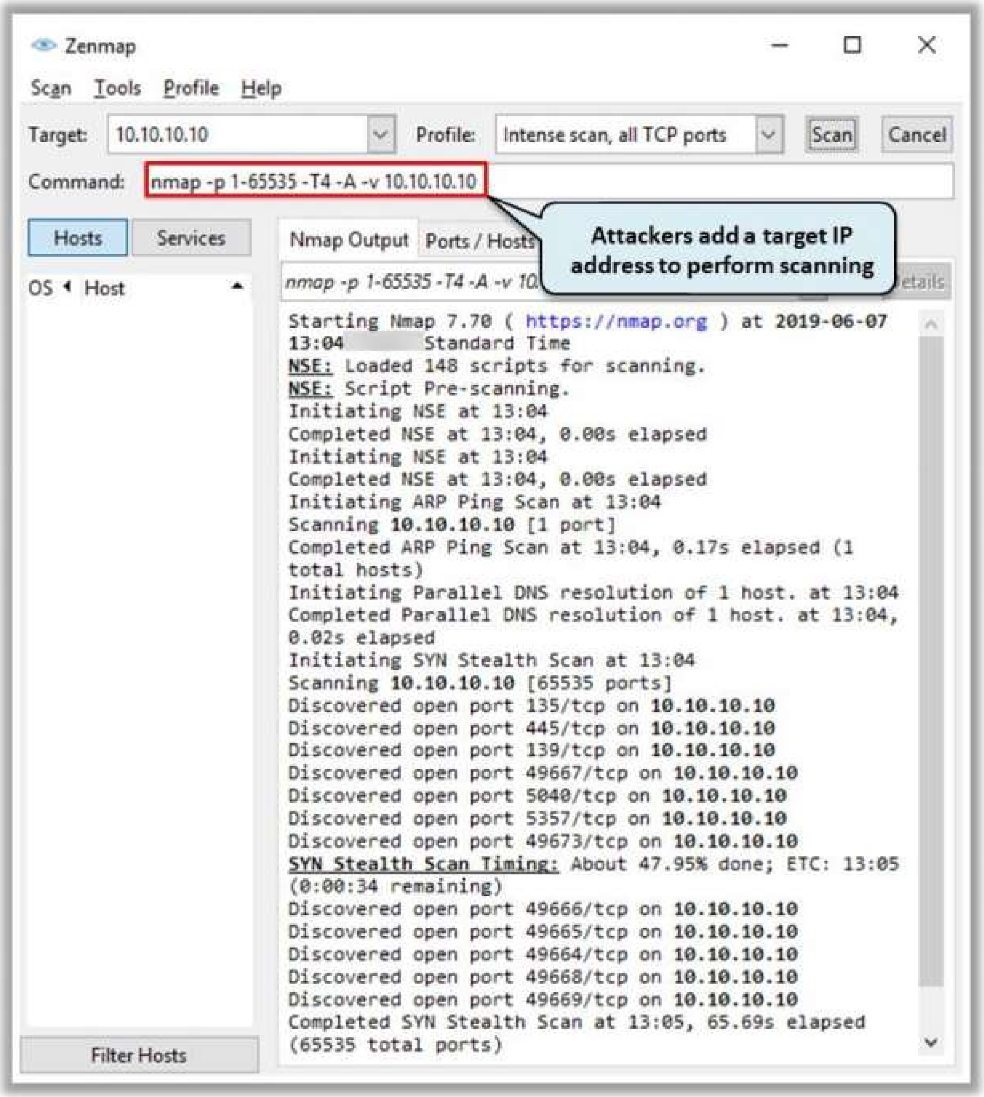


Figure 3.6: Screenshot displaying Nmap scan

Figure 3.7: Screenshot displaying Nmap scan result

- **Hping2/Hping3**

  Source: *http://www.hping.org*

  Hping2/Hping3 is a command-line-oriented network scanning and packet crafting tool for the TCP/IP protocol that sends ICMP echo requests and supports TCP, UDP, ICMP, and raw-IP protocols. It performs network security auditing, firewall testing, manual path MTU discovery, advanced traceroute, remote OS fingerprinting, remote uptime guessing, TCP/IP stacks auditing, and other functions. It can send custom TCP/IP packets and display target replies similarly to a ping program with ICMP replies. It handles fragmentation as well as arbitrary packet body and size, and it can be used to transfer encapsulated files under the supported protocols. It also supports idle host scanning. IP spoofing and network/host scanning can be used to perform an anonymous probe for services. Hping2/Hping3 also has a Traceroute mode, which enables attackers to send files between covert channels. It also determines whether the host is up even when the host blocks ICMP packets. Its firewalk-like usage allows the discovery of open ports behind firewalls. It performs manual path MTU discovery and enables attackers to perform remote OS fingerprinting.

Using Hping, an attacker can study the behavior of an idle host and gain information about the target, such as the services that the host offers, the ports supporting the services, and the OS of the target. This type of scan is a predecessor to either heavier probing or outright attacks.

Syntax: **# hping <options> <Target IP address>**

ICMP Scanning

A ping sweep or Internet Control Message Protocol (ICMP) scanning is a process of sending an ICMP request or ping to all the hosts on the network to determine the ones that are up.

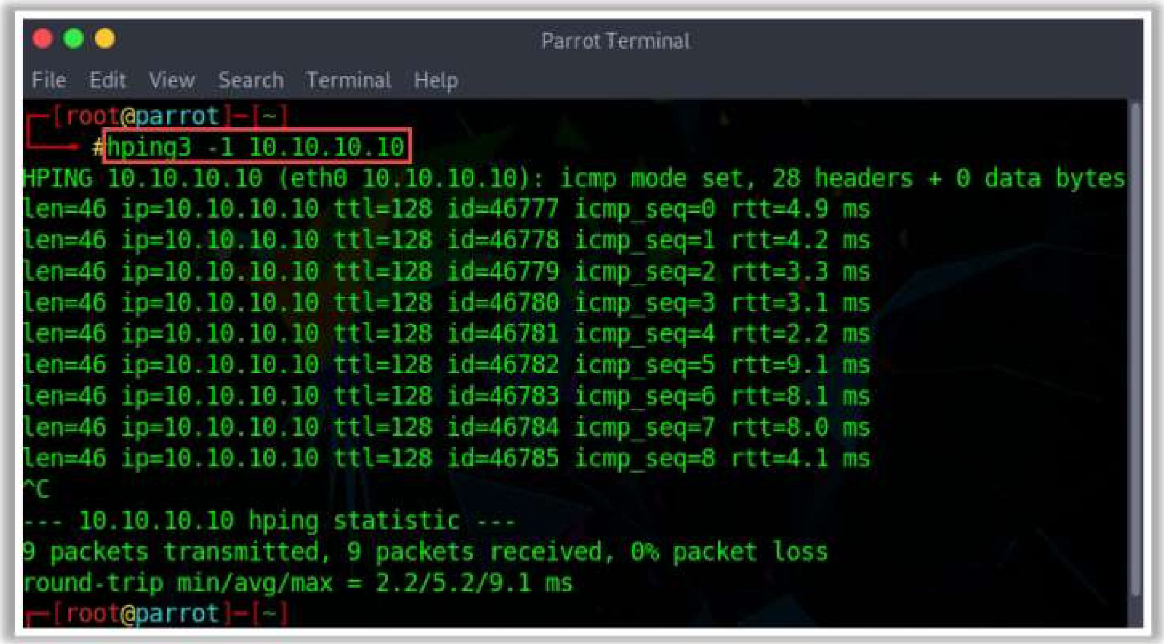


Figure 3.8: ICMP scanning

The OS, router, switch, and IP-based devices use this protocol via the ping command for echo request and echo response as a connectivity tester between different hosts.

ACK Scanning on Port 80

This scanning technique can be used to probe the existence of a firewall and its rule sets. Simple packet filtering allows the establishment of a connection (packets with the ACKbitset), whereas a sophisticated stateful firewall does not allow the establishment of a connection.

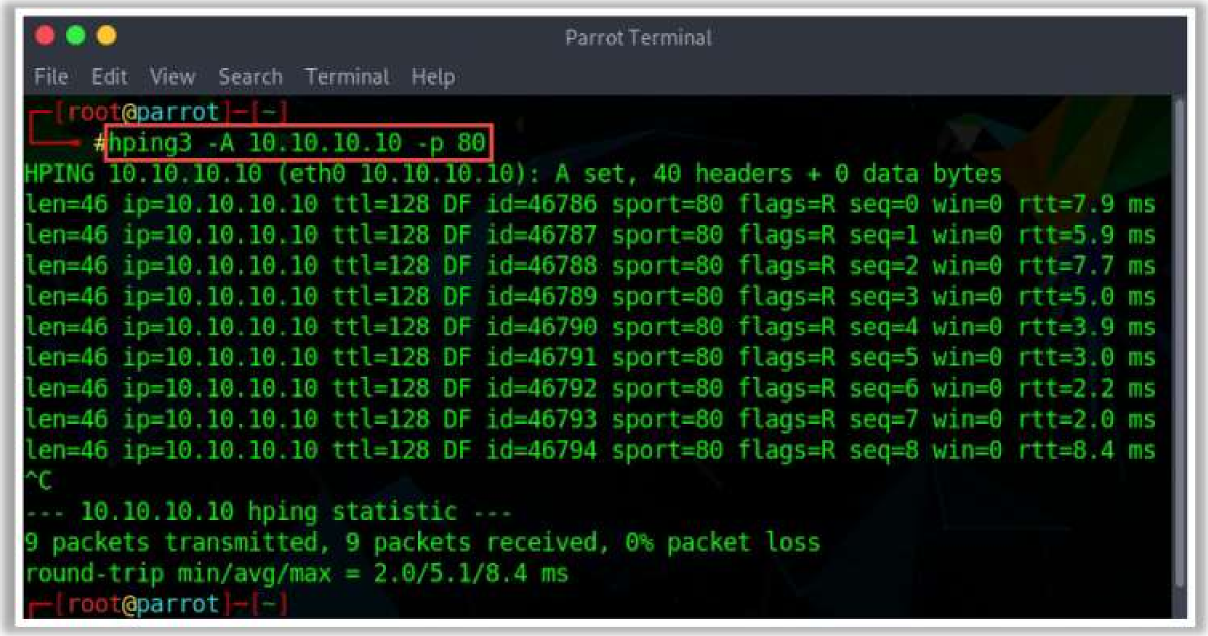


Figure 3.9: ACK scanning on port 80

Hping Commands

The various Hping commands are as follows:

- **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.

By issuing the above command, Hping sends UDP packets to port 80 on the host (10.0.0.25). It returns an ICMP port unreachable message if it finds the port closed and does not return a message if the port is open.

- **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.

The following table lists the various scanning methods and their respective Hping commands:

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`

Table 3.1: Hping command and its respective function

- **Metasploit**

  Source: *https://www.metasploit.com*

  Metasploit is an open-source project that provides the infrastructure, content, and tools to perform penetration tests and extensive security auditing. It provides information about security vulnerabilities and aids in penetration testing and IDS signature development. It facilitates the tasks of attackers, exploits writers, and payload writers. A major advantage of the framework is the modular approach, i.e., allowing the combination of any exploit with any payload.

  It enables you to automate the process of discovery and exploitation and provides you with the necessary tools to perform the manual testing phase of a penetration test. You can use Metasploit Pro to scan for open ports and services, exploit vulnerabilities, pivot further into a network, collect evidence, and create a report of the test results.

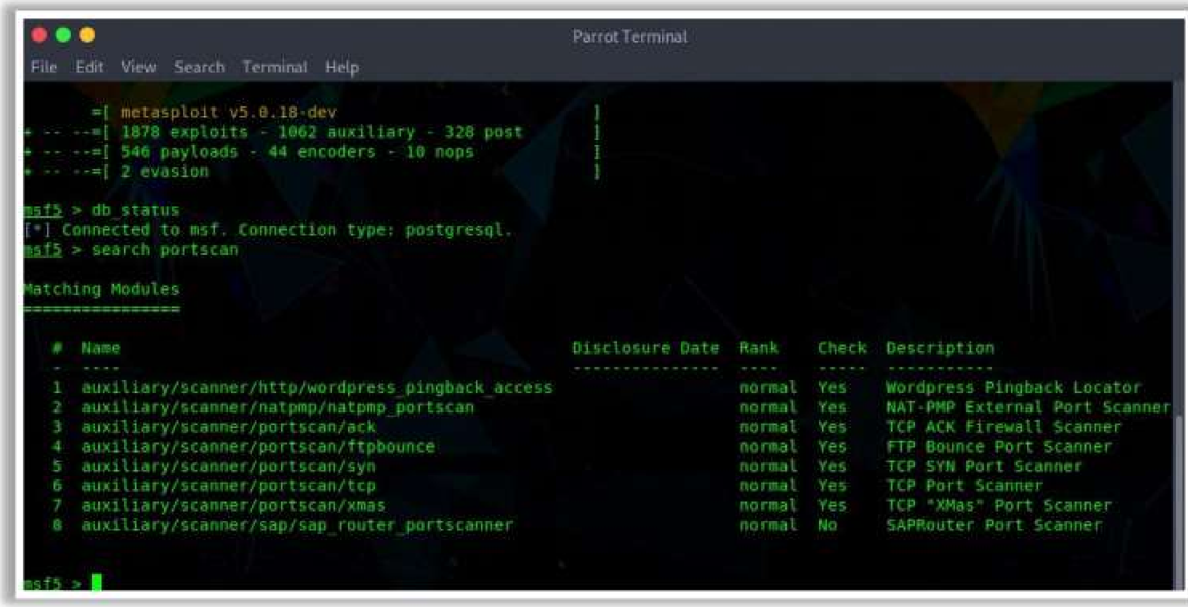


Figure 3.10: Screenshot displaying various Metasploit port scan modules

- **NetScanTools Pro**

  Source: *https://www.netscantools.com*

  NetScanTools Pro is an investigation tool that allows you to troubleshoot, monitor, discover, and detect devices on your network. Using this tool, you can easily gather information about the local LAN as well as Internet users, IP addresses, ports, and so on. Attackers can find vulnerabilities and exposed ports in the target system. It helps the attackers to list IPv4/IPv6 addresses, hostnames, domain names, email addresses, and URLs automatically or manually (using manual tools). NetScanTools Pro combines many network tools and utilities categorized by their functions, such as active, passive, DNS, and local computer.

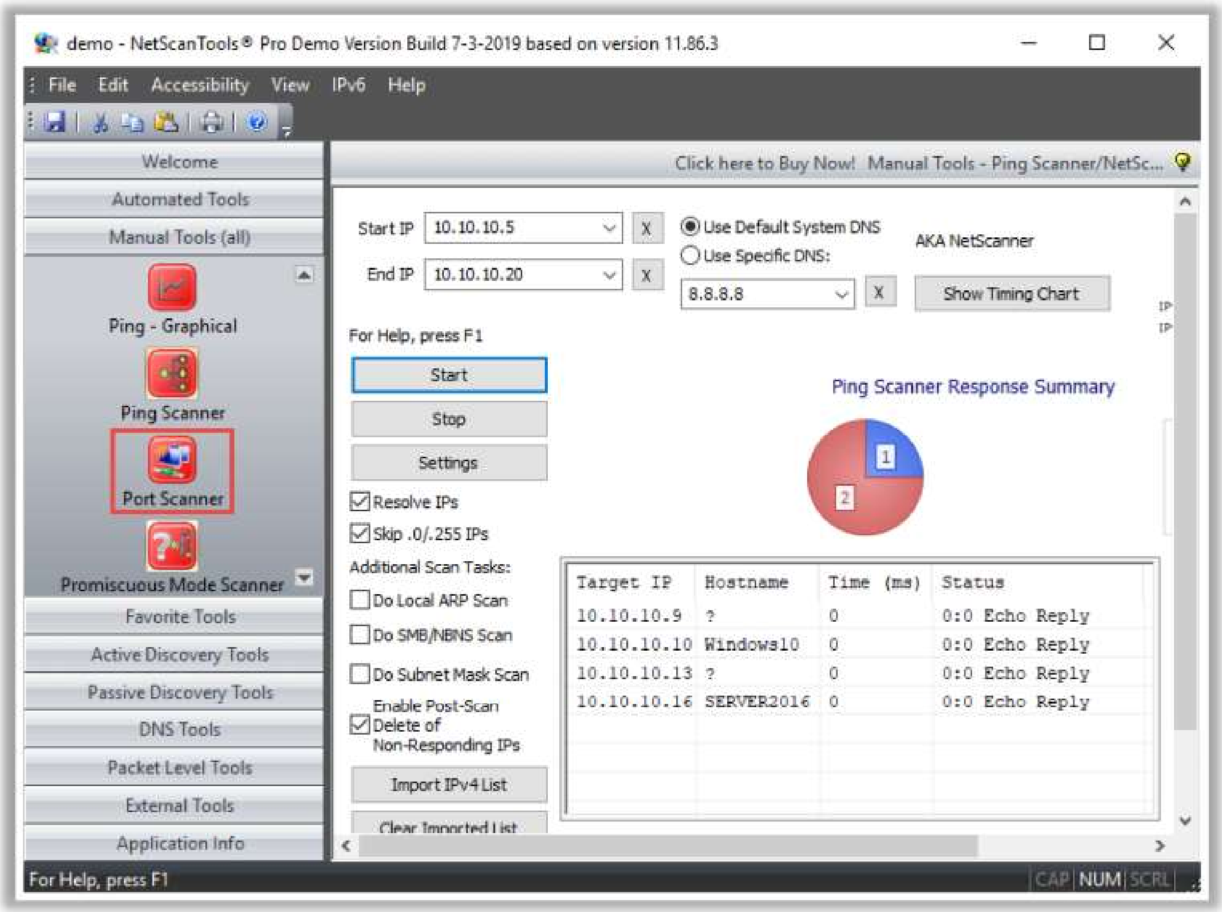
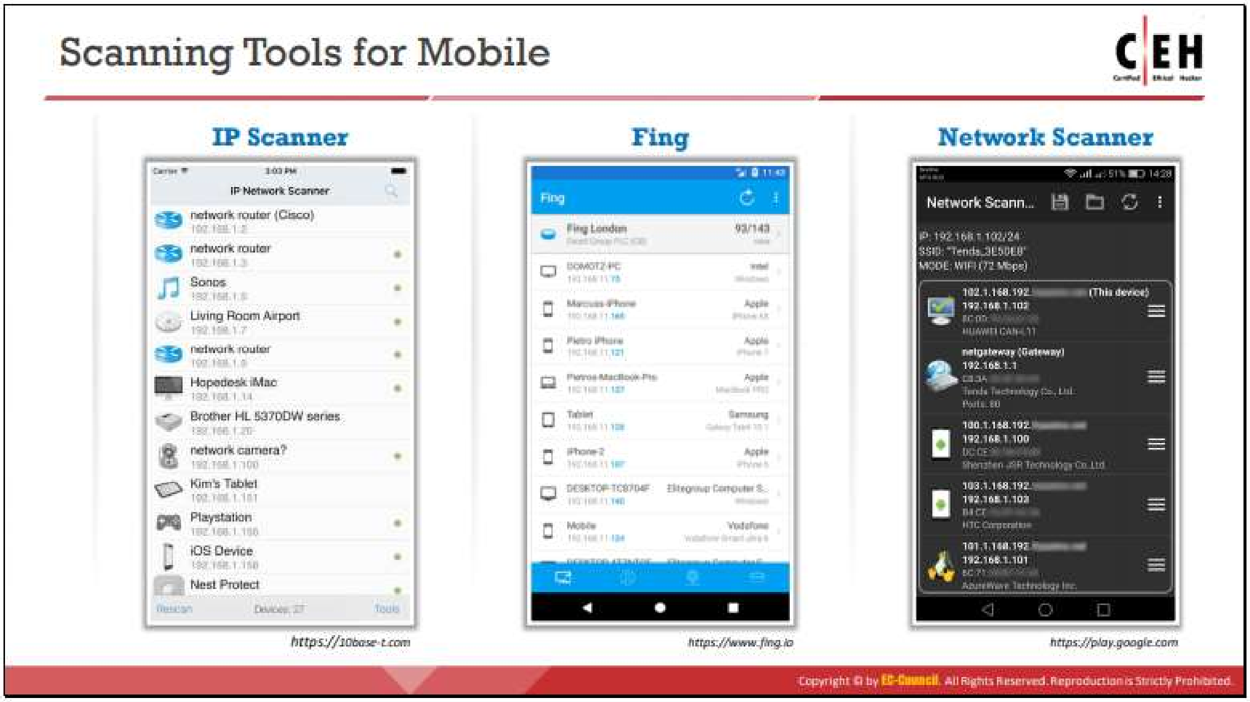


Figure 3.11: Screenshot of NetScanTools Pro

Some additional scanning tools are listed below:

- Unicornscan (*https://sourceforge.net*)
- SolarWinds Port Scanner (*https://www.solarwinds.com*)
- PRTG Network Monitor (*https://www.paessler.com*)
- OmniPeek Network Protocol Analyzer (*https://www.savvius.com*)



Scanning Tools for Mobile

- **IP Scanner**

  Source: *https://10base-t.com*

  IP Scanner for iOS scans your local area network to determine the identity of all its active machines and Internet devices. It allows attackers to perform network scanning activities along with ping and port scans.

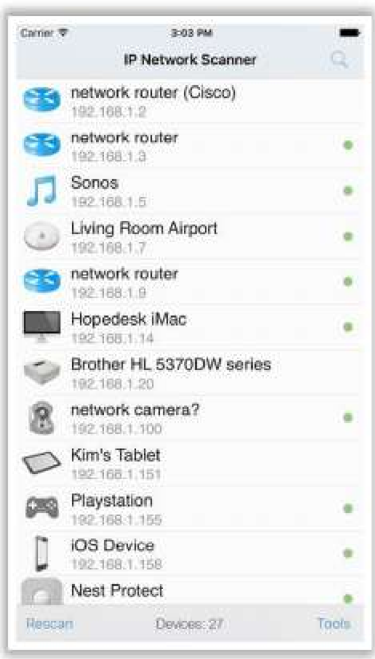


Figure 3.12: Screenshot of IP Scanner

- **Fing**

  Source: *https://www.fing.io*

  Fing is a mobile app for Android and iOS that scans and provides complete network information, such as IP address, MAC address, device vendor, and ISP location. It allows attackers to discover all devices connected to a Wi-Fi network along with their IP and MAC address as well as the name of the vendor/device manufacturer. It also allows attackers to perform network pinging and traceroute activities through specific ports such as SSH, FTP, NetBIOS, etc.

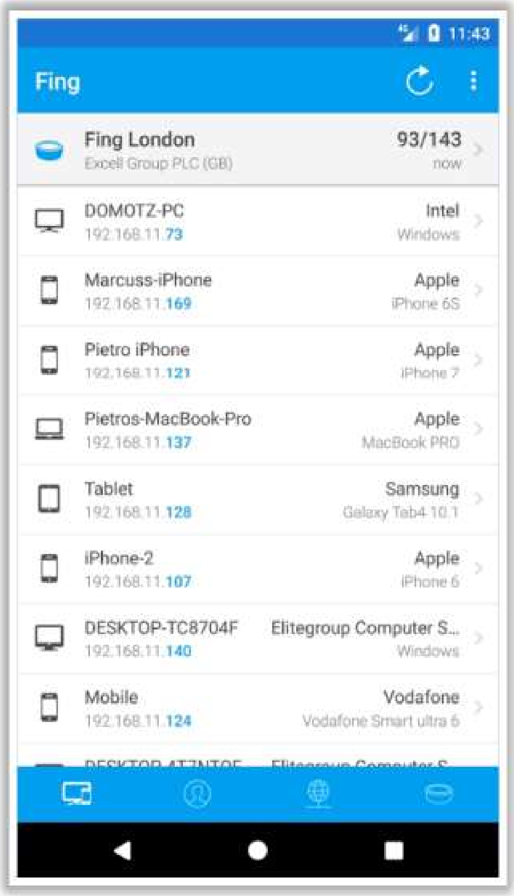


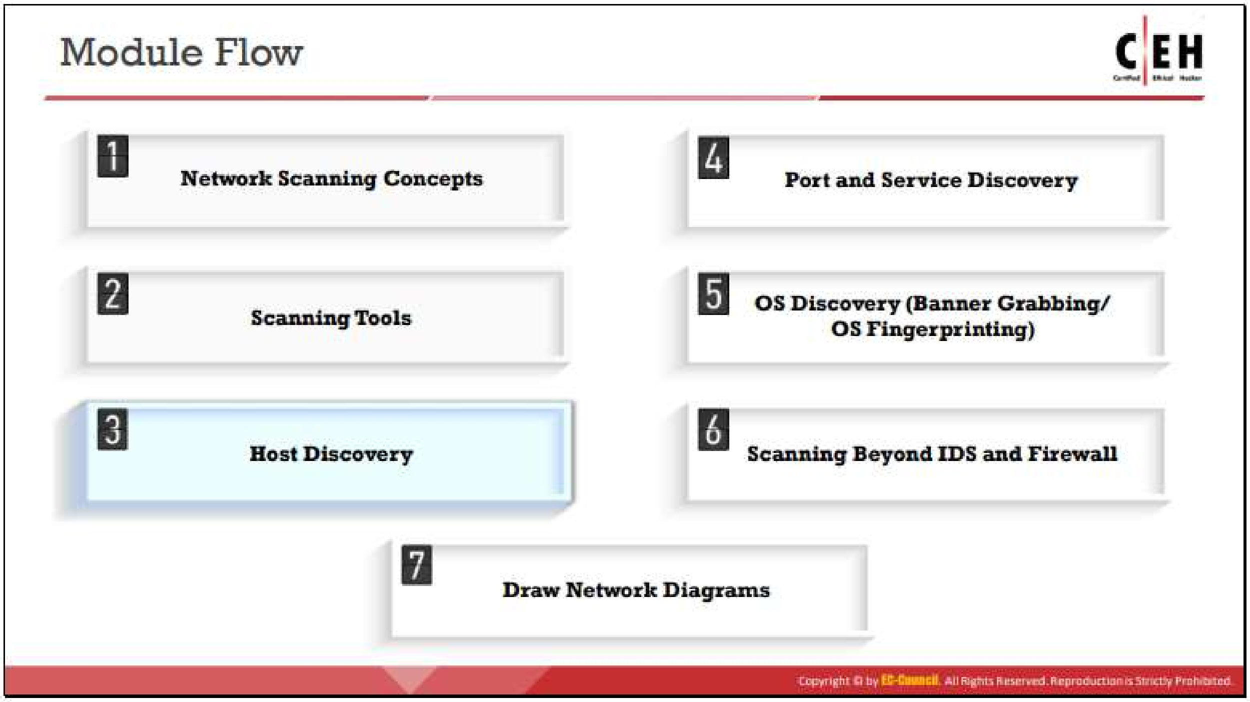
Figure 3.13: Screenshot of Fing

- **Network Scanner**

  Source: *https://play.google.com*

  Network Scanner is an Android mobile application that allows attackers to identify the active host in the range of possible addresses in a network. It also displays IP addresses, MAC addresses, host names, and vendor details of all the available devices in the network. This tool also allows attackers to port scan targets with specific port numbers.

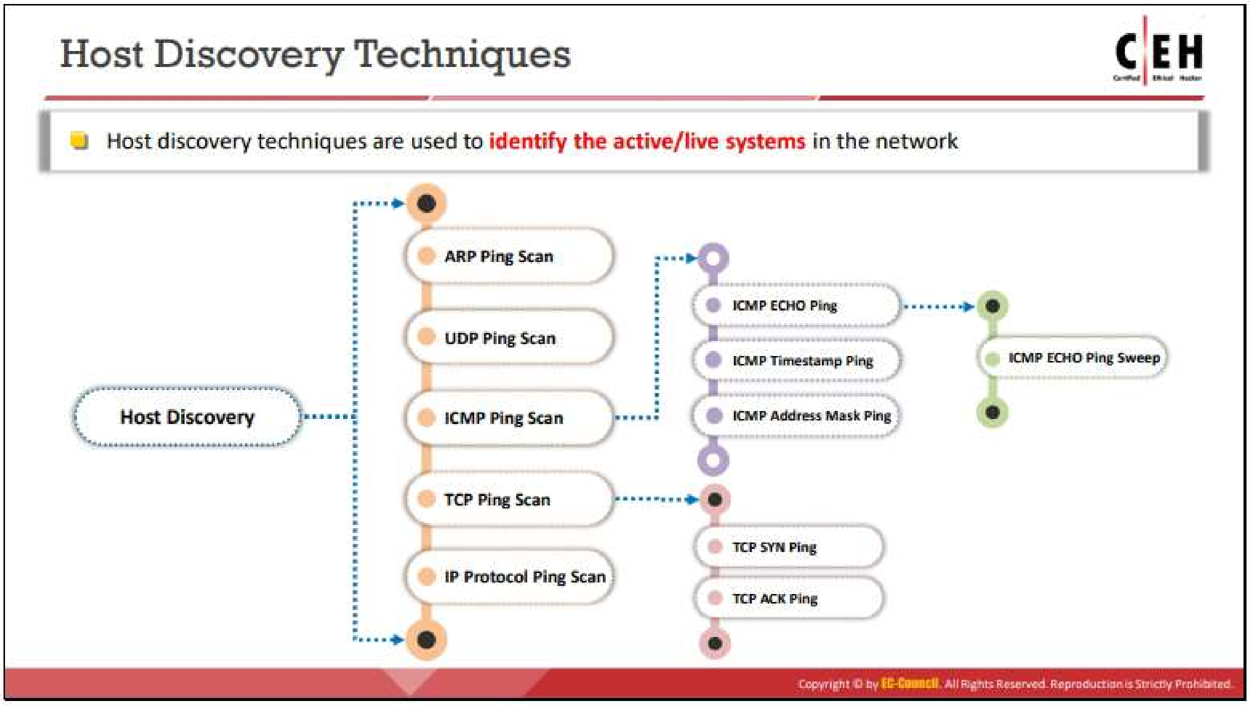
Figure 3.14: Screenshot of Network Scanner



Host Discovery

Scanning is the process of gathering information about systems that are "alive" and responding on the network. Host discovery is considered as the primary task in the network scanning process. To perform a complete scan and identify open ports and services, it is necessary to check for live systems. Host discovery provides an accurate status of the systems in the network, which enables an attacker to avoid scanning every port on every system in a sea of IP addresses to identify whether the target host is up.

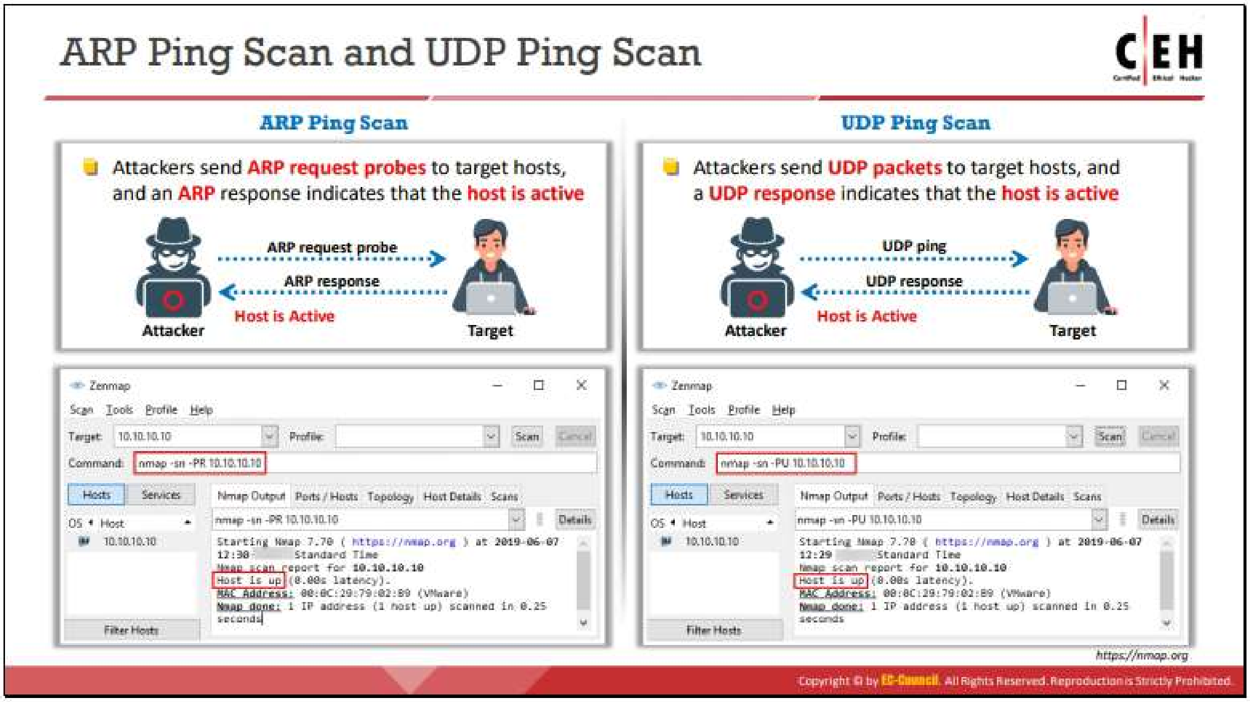
Host discovery is the first step in network scanning. This section highlights how to check for live systems in a network using various ping scan techniques. It also discusses how to ping sweep a network to detect live hosts/systems along with various ping sweep tools.



Host Discovery Techniques

Host discovery techniques can be adopted to discover the active/live hosts in the network. As an ethical hacker, you must be aware of the various types of host discovery techniques. Some host discovery techniques are listed below:

- ARP Ping Scan
- UDP Ping Scan
- ICMP Ping Scan
  - ICMP ECHO Ping
    - ICMP ECHO Ping Sweep
  - ICMP Timestamp Ping
  - ICMP Address Mask Ping
- TCP Ping Scan
  - TCP SYN Ping
  - TCP ACK Ping
- IP Protocol Scan



ARP Ping Scan and UDP Ping Scan

ARP Ping Scan

In the ARP ping scan, the ARP packets are sent for discovering all active devices in the IPv4 range even though the presence of such devices is hidden by restrictive firewalls. In most networks, many IP addresses are unused at any given time, specifically in the private address ranges of the LAN. Hence, when the attackers try to send IP packets such as ICMP echo request to the target host, the OS must determine the hardware destination address (ARP) corresponding to the target IP for addressing the ethernet frame correctly. For this purpose, a series of ARP requests are issued. ARP scan is used to show the MAC address of the network interface on the device, and it can also show the MAC addresses of all devices sharing the same IPv4 address on the LAN. If the host IP with the respective hardware destination address is active, then the ARP response will be generated by the host; otherwise, after a certain number of ping attempts, the original OS gives up on the host. In other words, when attackers send ARP request probes to the target host, if they receive any ARP response, then the host is active. In case the destination host is found to be unresponsive, the source host adds an incomplete entry to the destination IP in its kernel ARP table.

Attackers use the Nmap tool to perform ARP ping scan for discovering live hosts in the network. In Zenmap, the `-PR` option is used to perform ARP ping scan.

Note: `-sn` is the Nmap command to disable the port scan. Since Nmap uses ARP ping scan as the default ping scan, to disable it and perform other desired ping scans, you can use `--disable-arp-ping`.

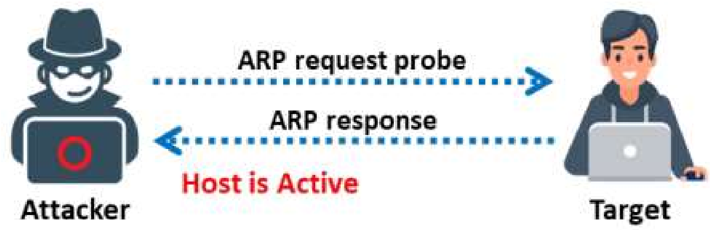


Figure 3.15: ARP ping scan

Advantages:

- ARP ping scan is considered to be more efficient and accurate than other host discovery techniques
- ARP ping scan automatically handles ARP requests, retransmission, and timeout at its own discretion
- ARP ping scan is useful for system discovery, where you may need to scan large address spaces
- ARP ping scan can display the response time or latency of a device to an ARP packet

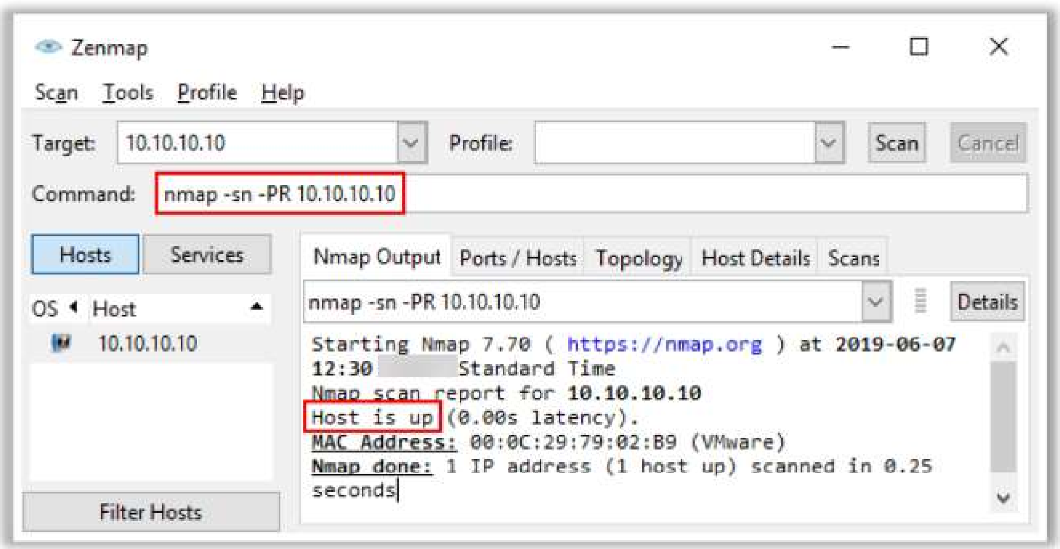


Figure 3.16: ARP scan in Zenmap

UDP Ping scan

UDP ping scan is similar to TCP ping scan; however, in the UDP ping scan, Nmap sends UDP packets to the target host. The default port number used by Nmap for the UDP ping scan is 40,125. This highly uncommon port is used as the default for sending UDP packets to the target. This default port number can be configured using `DEFAULT_UDP_PROBE_PORT_SPEC` during compile time in Nmap.

Attackers send UDP packets to the target host, and a UDP response means that the target host is active. If the target host is offline or unreachable, various error messages such as host/network unreachable or TTL exceeded could be returned. In Zenmap, the **-PU** option is used to perform the UDP ping scan.

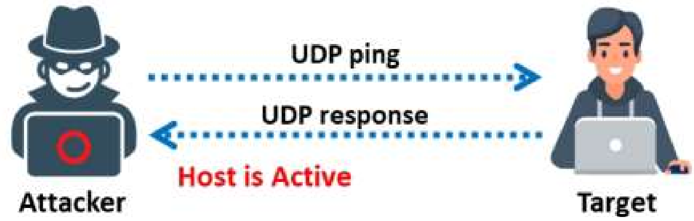


Figure 3.17: UDP ping scan to determine if the host is active

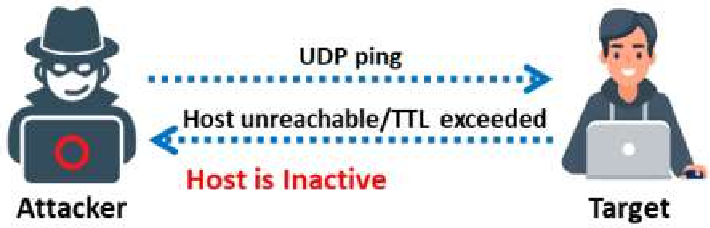


Figure 3.18: UDP ping scan to determine if the host is offline

Advantages:

- UDP ping scans have the advantage of detecting systems behind firewalls with strict TCP filtering, leaving the UDP traffic forgotten.

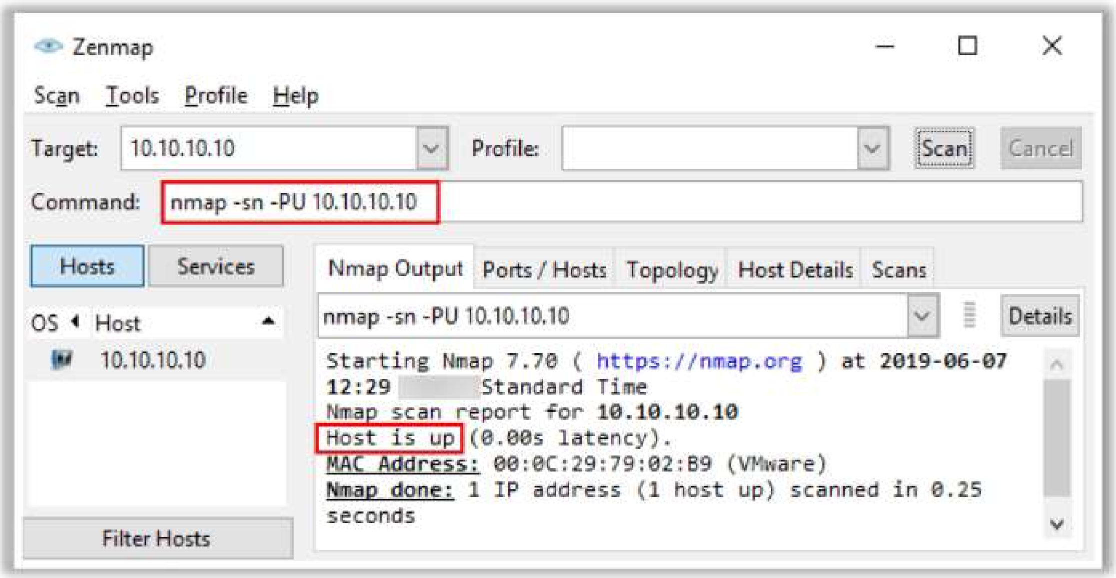
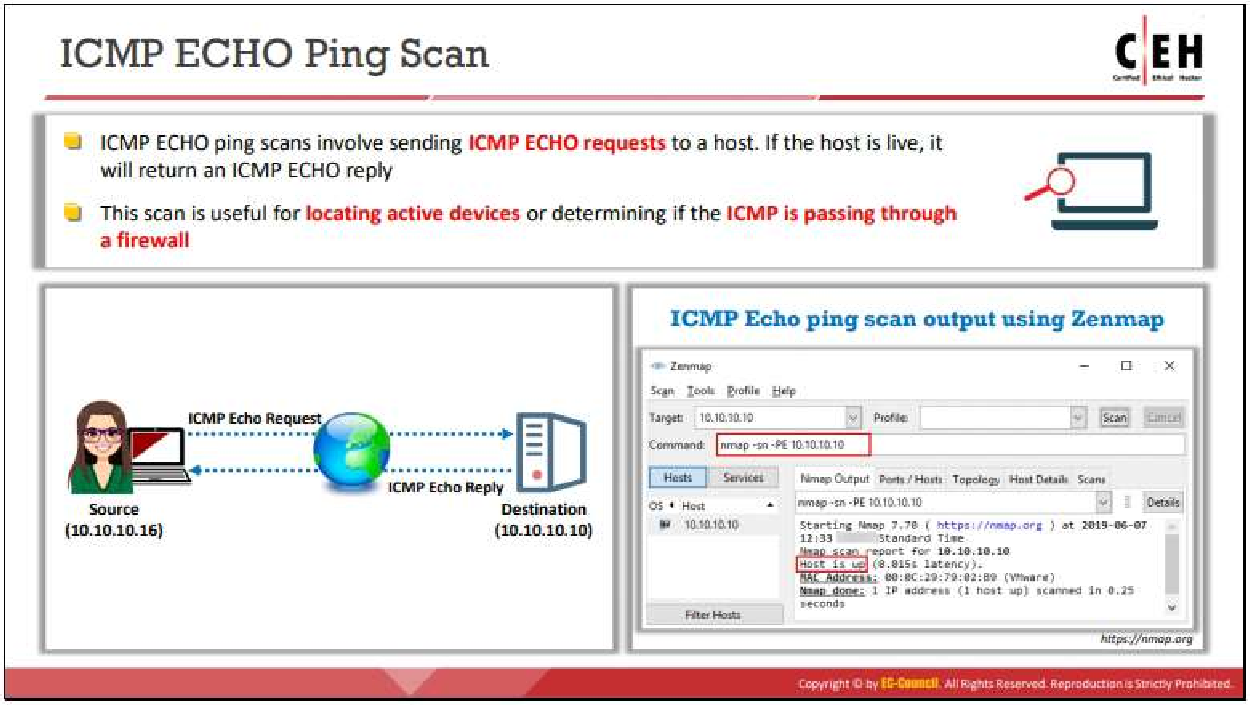


Figure 3.19: UDP ping scan in Zenmap



ICMP ECHO Ping Scan

Attackers use the ICMP ping scan to send ICMP packets to the destination system to gather all necessary information about it. This is because ICMP does not include port abstraction, and it is different from port scanning. However, it is useful to determine what hosts in a network are running by pinging them all.

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

Figure 3.20: ICMP echo request and reply

UNIX/Linux and BSD-based machines use ICMP echo scanning; the TCP/IP stack implementations in these OSs respond to the ICMP echo requests to the broadcast addresses. This technique does not work on Windows-based networks, as their TCP/IP stack implementation does not reply to ICMP probes directed at the broadcast address.

Nmap uses the `-P` option to ICMP scan the target. The user can also increase the number of pings in parallel using the `-L` option. It may also be useful to tweak the ping timeout value using the `-T` option.

In Zenmap, the **`-PE`** option is used to perform the ICMP ECHO ping scan. Active hosts are displayed as "**`Host is up,`**" as shown in the screenshot.

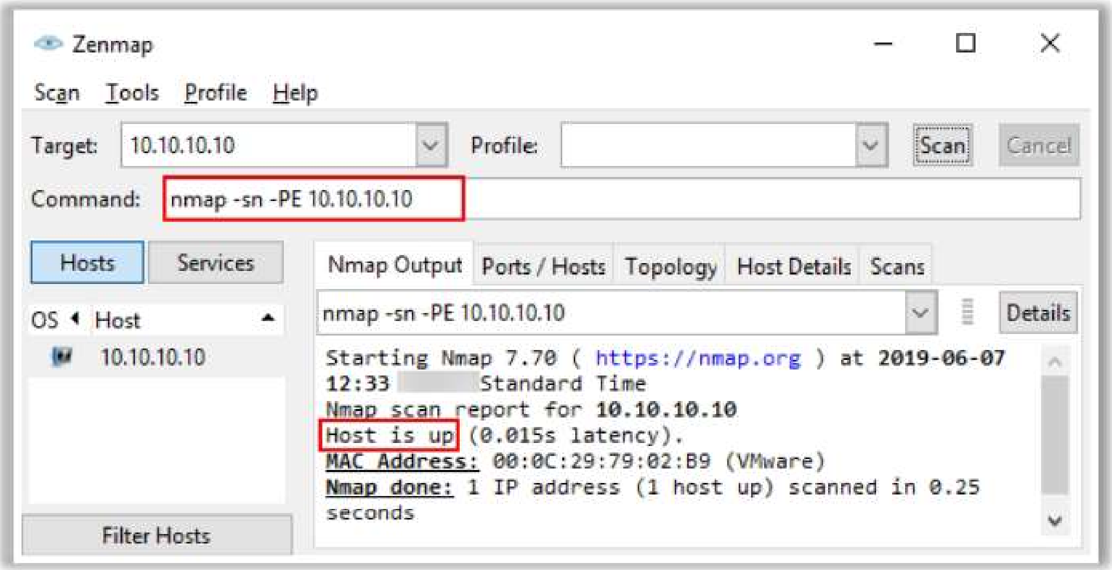
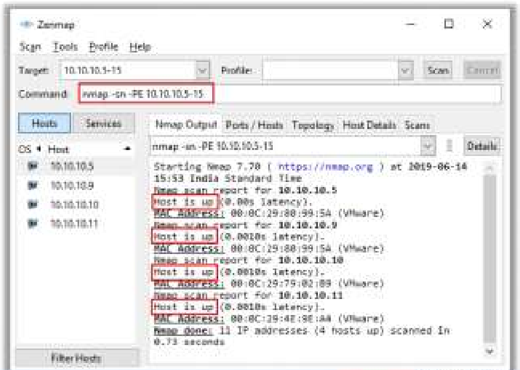


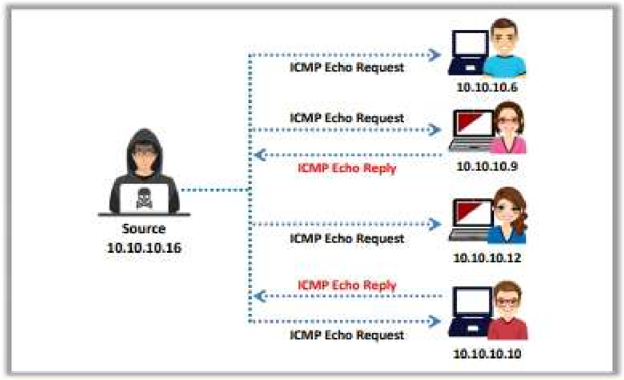
Figure 3.21: ICMP Echo ping scan output using Zenmap

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



https://nmap.org



ICMP ECHO Ping Sweep

A ping sweep (also known as an ICMP sweep) is a basic network scanning technique that is adopted to determine the range of IP addresses that map to live hosts (computers). Although a single ping will tell the user whether a specified host computer exists on the network, a ping sweep consists of ICMP ECHO requests sent to multiple hosts. If a specified host is active, it will return an ICMP ECHO reply.

Ping sweeps are among the oldest and slowest methods used to scan a network. This utility is distributed across nearly all platforms, and it acts as a roll call for systems; a system that is active on the network answers the ping query that another system sends out.

ICMP echo scanning pings all the machines in the target network to discover live machines. Attackers send ICMP probes to the broadcast or network address, which relays to all the host addresses in the subnet. The live systems will send the ICMP echo reply message to the source of the ICMP echo probe.

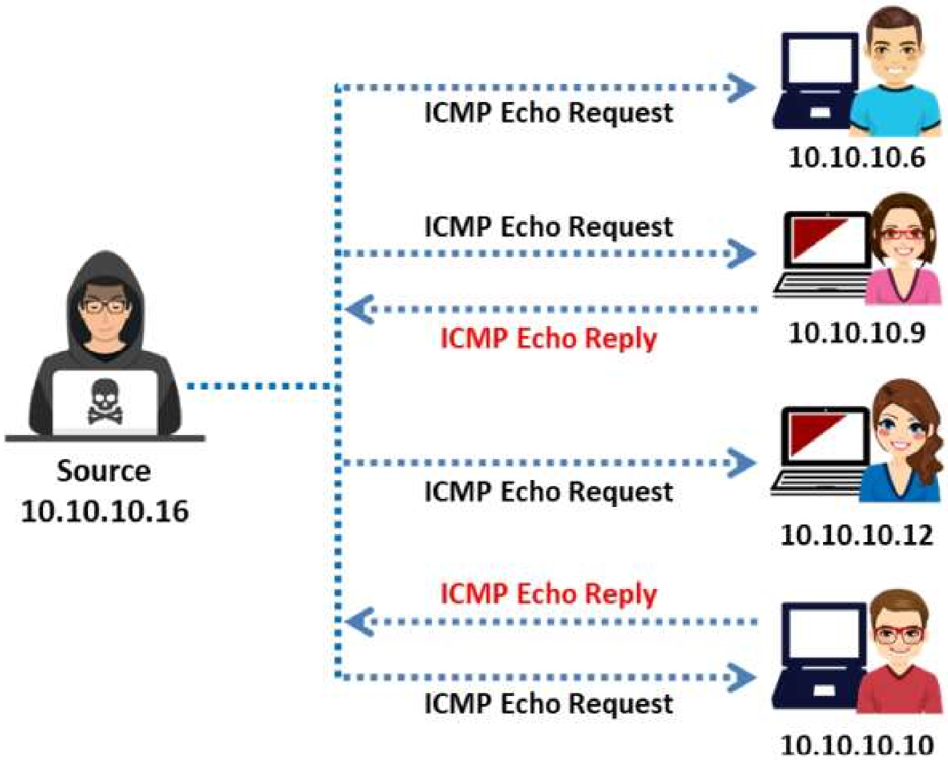


Figure 3.22: ICMP ECHO Ping Sweep

To understand pings better, one should be able to understand the TCP/IP packet. When a system pings, it sends a single packet across the network to a specific IP address. This packet contains 64 bytes (56 data bytes and 8 bytes of protocol header information). The sender then waits or listens for a return packet from the target system. If the connections are good and the target computer is "alive," a good return packet is expected. However, this will not be the case if there is a disruption in communication. Pings also detail the time taken for a packet to make a complete trip, called the "round-trip time." They also help in resolving hostnames. In this case, if the packet bounces back when sent to the IP address, but not when sent to the name, then the system is unable to reconcile the name with the specific IP address.

Attackers calculate subnet masks using subnet mask calculators to identify the number of hosts that are present in the subnet. They subsequently use ping sweep to create an inventory of live systems in the subnet.

ICMP ECHO Ping Sweep Using Nmap

Source: *https://nmap.org*

Nmap helps an attacker to perform a ping sweep that determines live hosts from a range of IP addresses. In Zenmap, the **-PE** option with a list of IP addresses is used to perform ICMP ECHO ping sweep.

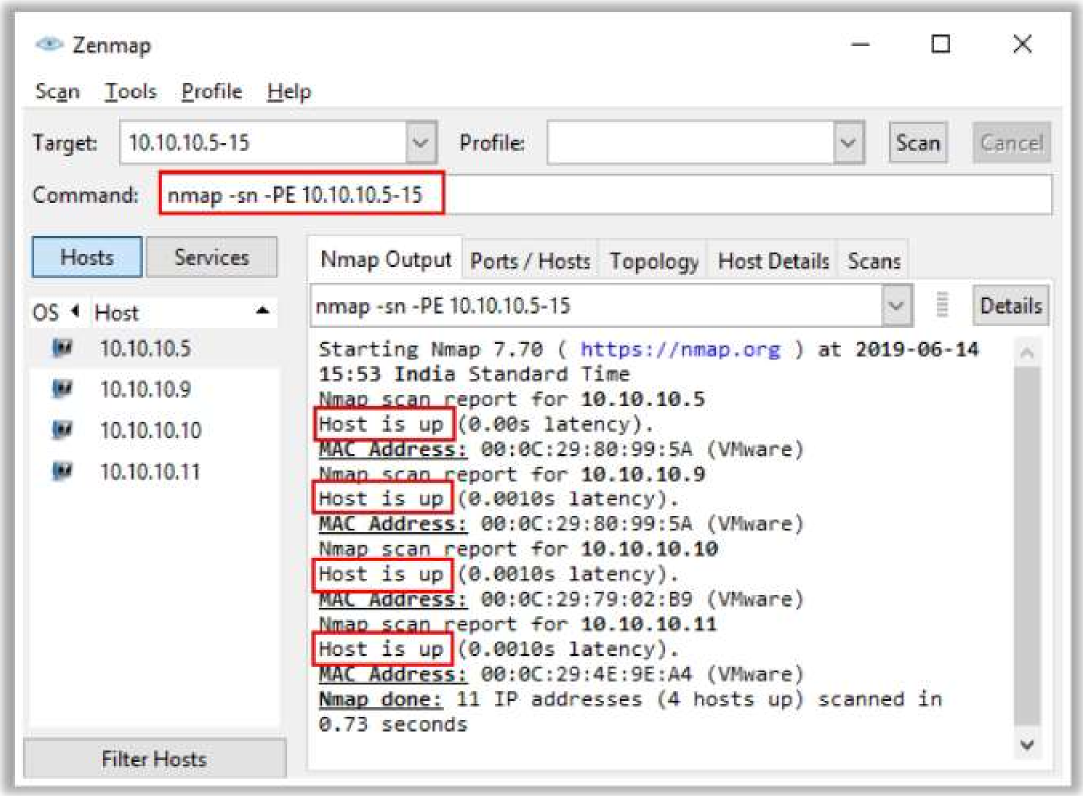


Figure 3.23: Ping Sweep output using Zenmap

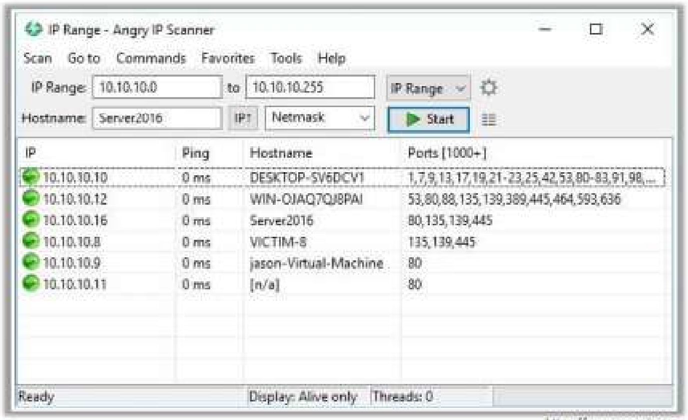
Ping Sweep Tools

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*)



https://www.angryip.org

Ping Sweep Tools

Ping sweep tools ping an entire range of network IP addresses to identify the live systems. The following are ping sweep tools that enable one to determine live hosts on the target network by sending multiple ICMP ECHO requests to various hosts on the network at a time.

- **Angry IP Scanner**

  Source: https://www.angryip.org

  Angry IP scanner is an IP address and port scanner. It can scan IP addresses in any range as well as any of their ports. It pings each IP address to check if it is alive; then, it optionally resolves its hostname, determines the MAC address, scans ports, and so on. The amount of data gathered about each host increases with plugins. Angry IP scanner has additional features, such as NetBIOS information (computer name, workgroup name, and currently logged in Windows user), favorite IP address ranges, web server detection, and customizable openers. The tool allows the user to save the scanning results to CSV, TXT, XML, or IP-Port list files. To increase the scanning speed, it uses a multithreaded approach: a separate scanning thread is created for each scanned IP address.

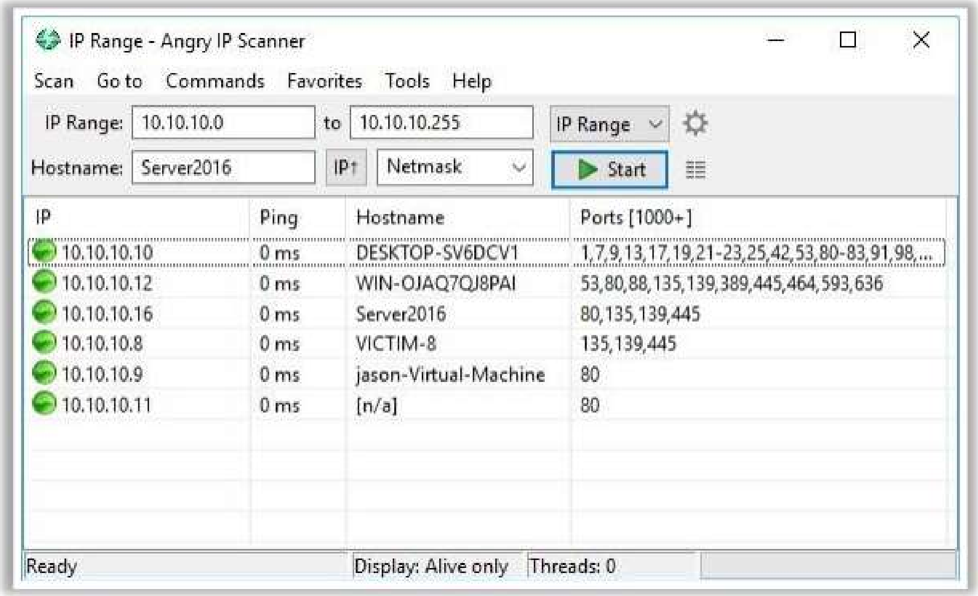


Figure 3.24: Screenshot of Angry IP Scanner showing live hosts

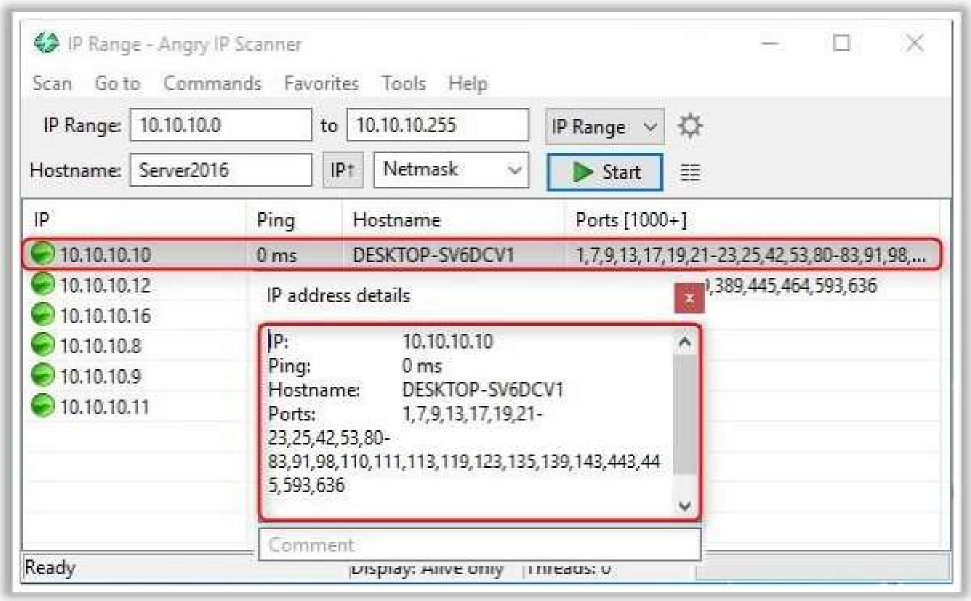


Figure 3.25: Screenshot of Angry IP Scanner showing complete details of live hosts

Some additional ping sweep tools that an attacker uses to determine live hosts on the target network are listed below:

- 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

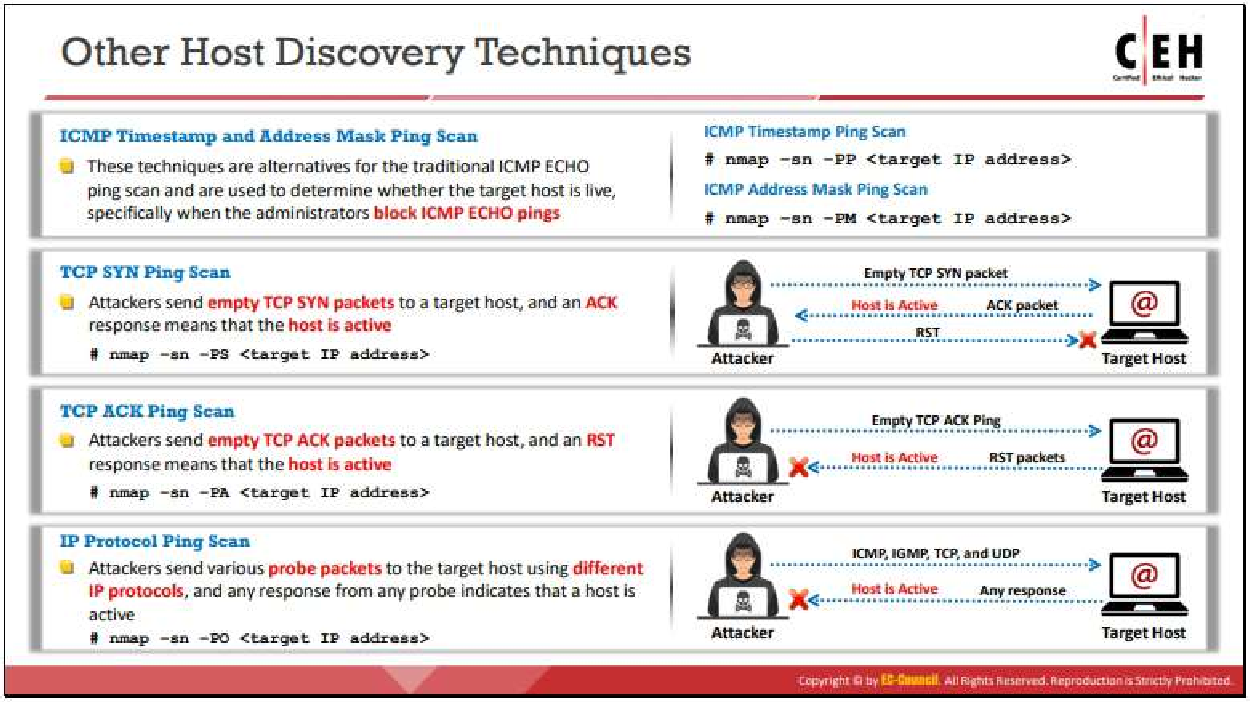
C|EH Certified Ethical Hacker

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

Ping Sweep Countermeasures

Some countermeasures for avoiding ping sweep are as follows:

- Configure the firewall to detect and prevent ping sweep attempts instantaneously
- Use intrusion detection systems and intrusion prevention systems such as Snort (*https://www.snort.org*) to detect and prevent ping sweep attempts
- Carefully evaluate the type of ICMP traffic flowing through the enterprise networks
- Terminate the connection with any host that is performing more than 10 ICMP ECHO requests
- Use DMZ and allow only commands such as `ICMP ECHO_REPLY`, `HOST UNREACHABLE`, and `TIME EXCEEDED` in DMZ Zone
- Limit the ICMP traffic with Access Control Lists (ACLs) to your ISP's specific IP addresses



Other Host Discovery Techniques

ICMP Timestamp Ping Scan

Besides the traditional ICMP ECHO ping, there are some other types of ICMP pinging techniques such as ICMP timestamp ping scan and ICMP address mask ping scan, which an attacker can adopt in specific conditions.

ICMP timestamp ping is an optional and additional type of ICMP ping whereby the attackers query a timestamp message to acquire the information related to the current time from the target host machine. The target machine responds with a timestamp reply to each timestamp query that is received. However, the response from the destination host is conditional, and it may or may not respond with the time value depending on its configuration by the administrator at the target's end. This ICMP timestamp pinging is generally used for time synchronization. Such a ping method is effective in identifying whether the destination host machine is active, specifically in the condition where the administrator blocks the traditional ICMP ECHO ping requests. In Zenmap, the `-PP` option is used to perform an ICMP timestamp ping scan.

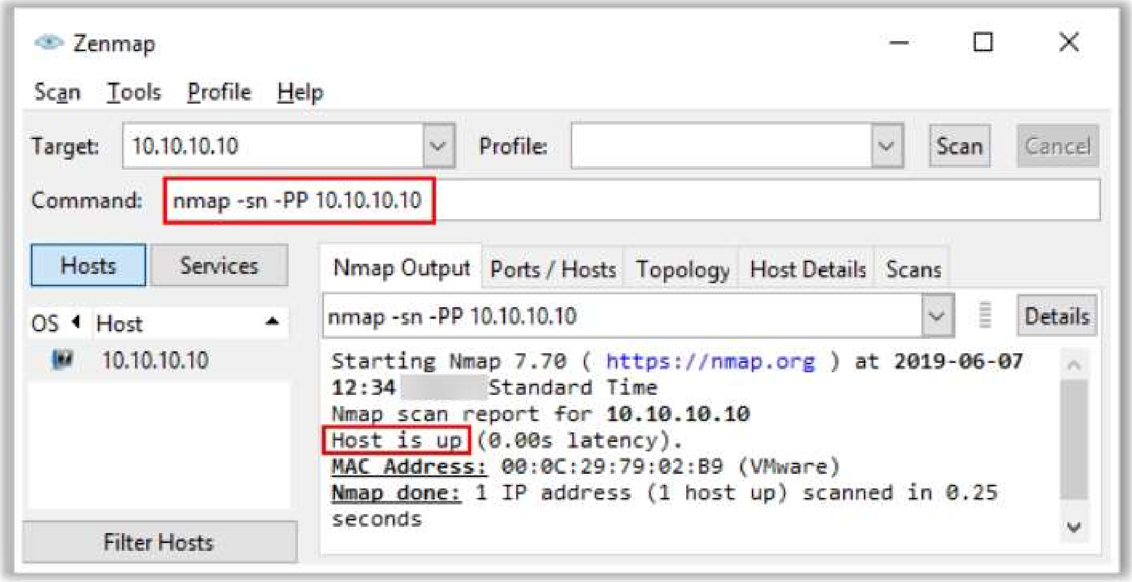


Figure 3.26: ICMP timestamp ping in Zenmap

ICMP Address Mask Ping Scan

ICMP address mask ping is another alternative to the traditional ICMP ECHO ping, where the attackers send an ICMP address mask query to the target host to acquire information related to the subnet mask. However, the address mask response from the destination host is conditional, and it may or may not respond with the appropriate subnet value depending on its configuration by the administrator at the target's end. This type of ping method is also effective in identifying the active hosts similarly to the ICMP timestamp ping, specifically when the administrator blocks the traditional ICMP Echo ping. In Zenmap, the **-PM** option is used to perform an ICMP address mask ping scan.

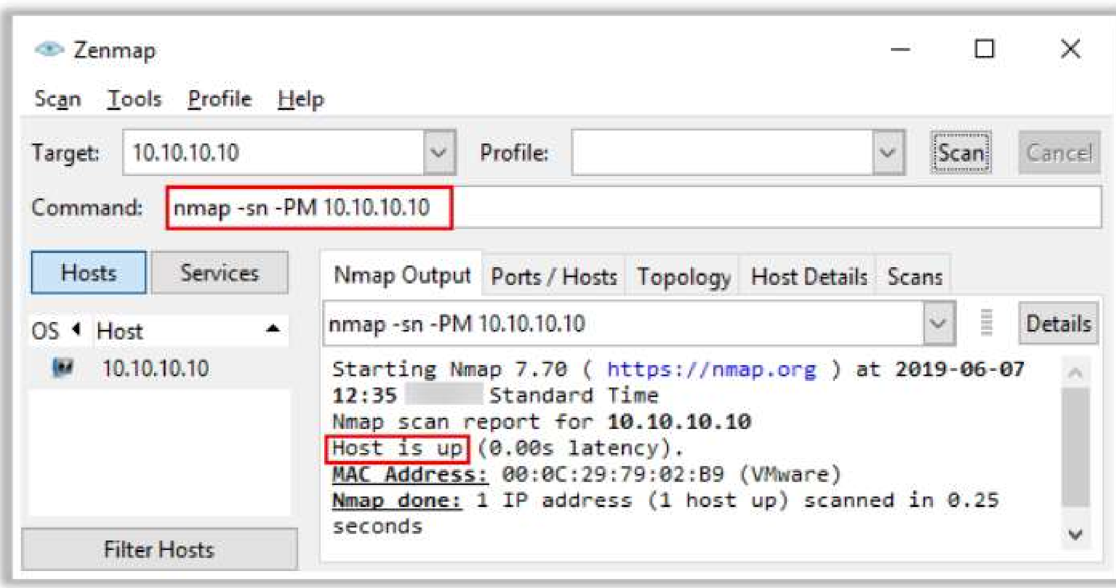


Figure 3.27: ICMP address mask ping in Zenmap

TCP SYN Ping Scan

TCP SYN ping is a host discovery technique for probing different ports to determine if the port is online and to check if it encounters any firewall rule sets. In this type of host discovery technique, an attacker uses the Nmap tool to initiate the three-way handshake by sending the empty TCP SYN flag to the target host. After receiving SYN, the target host acknowledges the receipt with an ACK flag. After reception of the ACK flag, the attacker confirms that the target host is active and terminates the connection by sending an RST flag to the target host machine (since his/her objective of host discovery is accomplished). Port 80 is used as the default destination port. A range of ports can also be specified in this type of pinging format without inserting a space between -PS and the port number (e.g., **PS22-25,80,113,1050,35000**), where the probe will be performed against each port parallelly. In Zenmap, the **-PS** option is used to perform a TCP SYN ping scan.

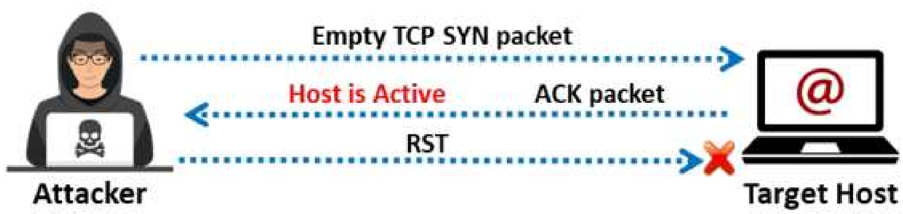


Figure 3.28: TCP SYN ping scan for host discovery

Advantages:

- As the machines can be scanned parallelly, the scan never gets the time-out error while waiting for the response.
- TCP SYN ping can be used to determine if the host is active without creating any connection. Hence, the logs are not recorded at the system or network level, enabling the attacker to leave no traces for detection.

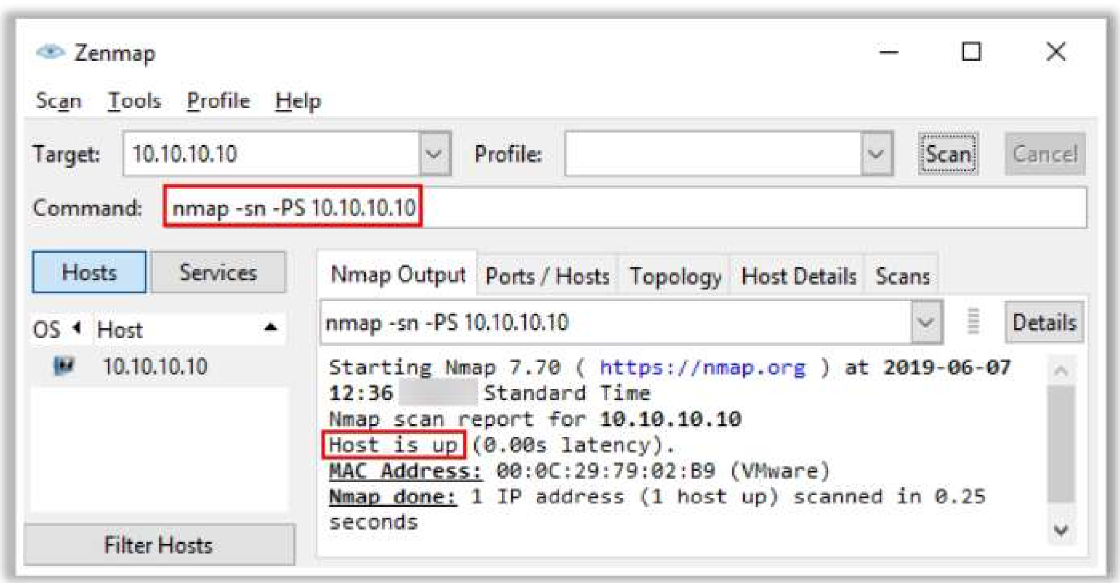


Figure 3.29: TCP SYN ping scan in Zenmap

TCP ACK Ping Scan

TCP ACK ping is similar to TCP SYN ping, albeit with minor variations. TCP ACK ping also uses the default port 80. In the TCP ACK ping technique, the attackers send an empty TCP ACK packet to the target host directly. Since there is no prior connection between the attacker and the target host, after receiving the ACK packet, the target host responds with an RST flag to terminate the request. The reception of this RST packet at the attacker's end indicates that the host inactive. In Zenmap, the **-PA** option is used to perform a TCP ACK ping scan.

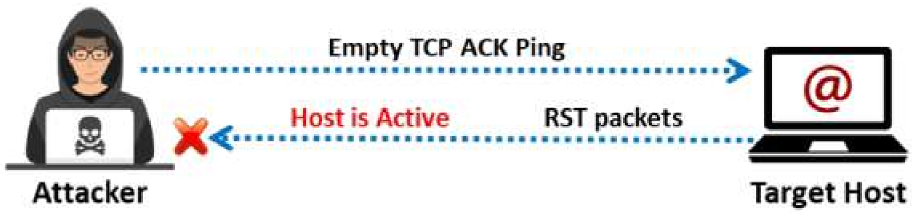


Figure 3.30: TCP ACK ping scan for host discovery

Advantages:

- Both the SYN and the ACK packet can be used to maximize the chances of bypassing the firewall. However, firewalls are mostly configured to block the SYN ping packets, as they are the most common pinging technique. In such cases, the ACK probe can be effectively used to bypass these firewall rule sets easily.

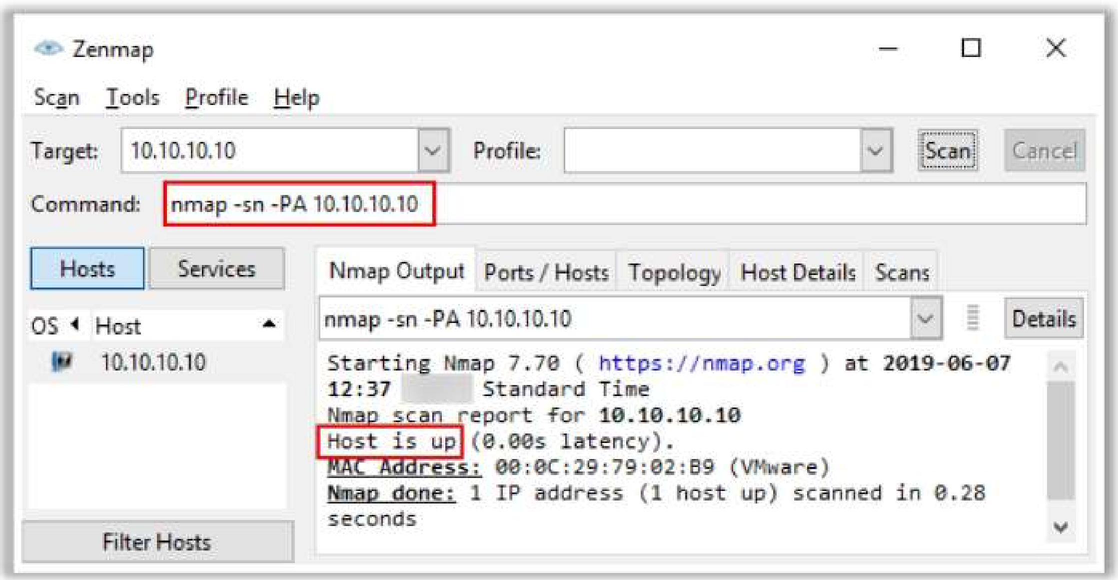


Figure 3.31: TCP ACK ping scan in Zenmap

IP Protocol Ping Scan

IP protocol ping is the latest host discovery option that sends IP ping packets with the IP header of any specified protocol number. It has the same format as the TCP and UDP ping. This technique tries to send different packets using different IP protocols, hoping to get a response indicating that a host is online.

Multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4) are sent by default when no protocols are specified. For configuring the default protocols, change **`DEFAULT_PROTO_PROBE_PORT_SPEC`** in **`nmap.h`** during compile time. For specific protocols such as ICMP, IGMP, TCP (protocol 6), and UDP (protocol 17), the packets are to be sent with proper protocol headers, and for the remaining protocols, only the IP header data is to be sent with the packets.

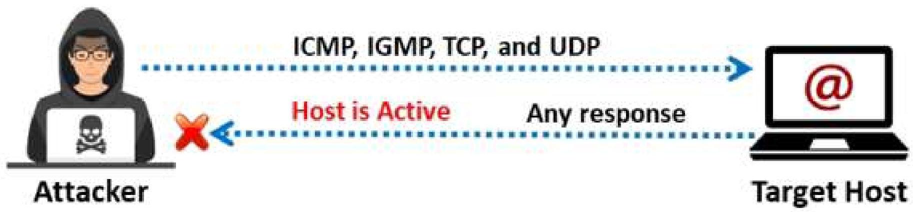


Figure 3.32: IP protocol ping scan for host discovery

In a nutshell, attackers send different probe packets of different IP protocols to the target host; any response from any probe indicates that a host is online. In Zenmap, the **`-PO`** option is used to perform an IP protocol ping scan.

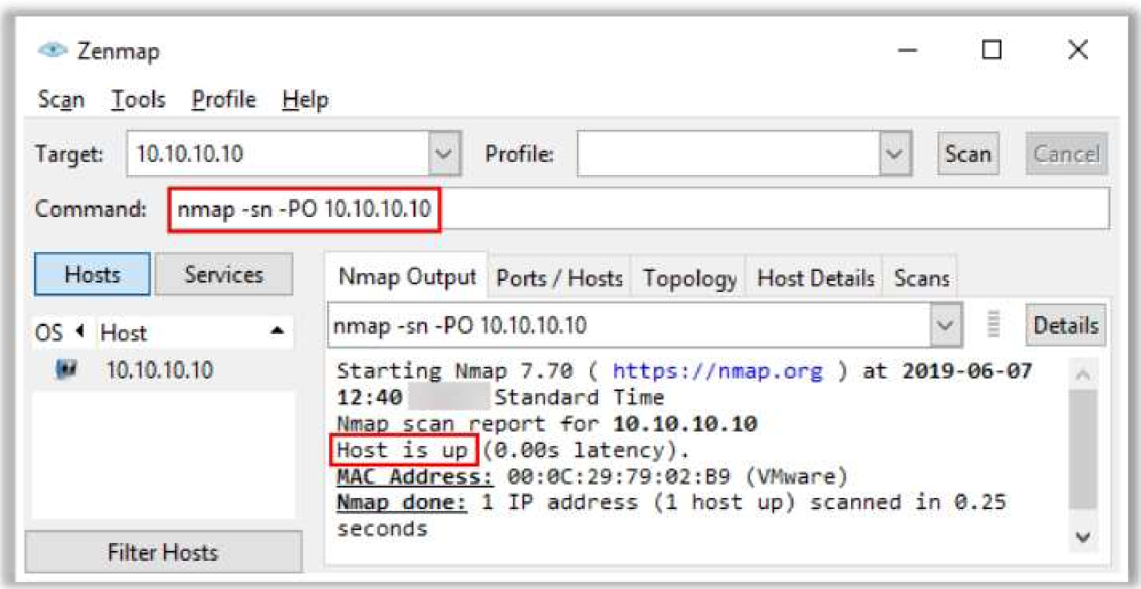
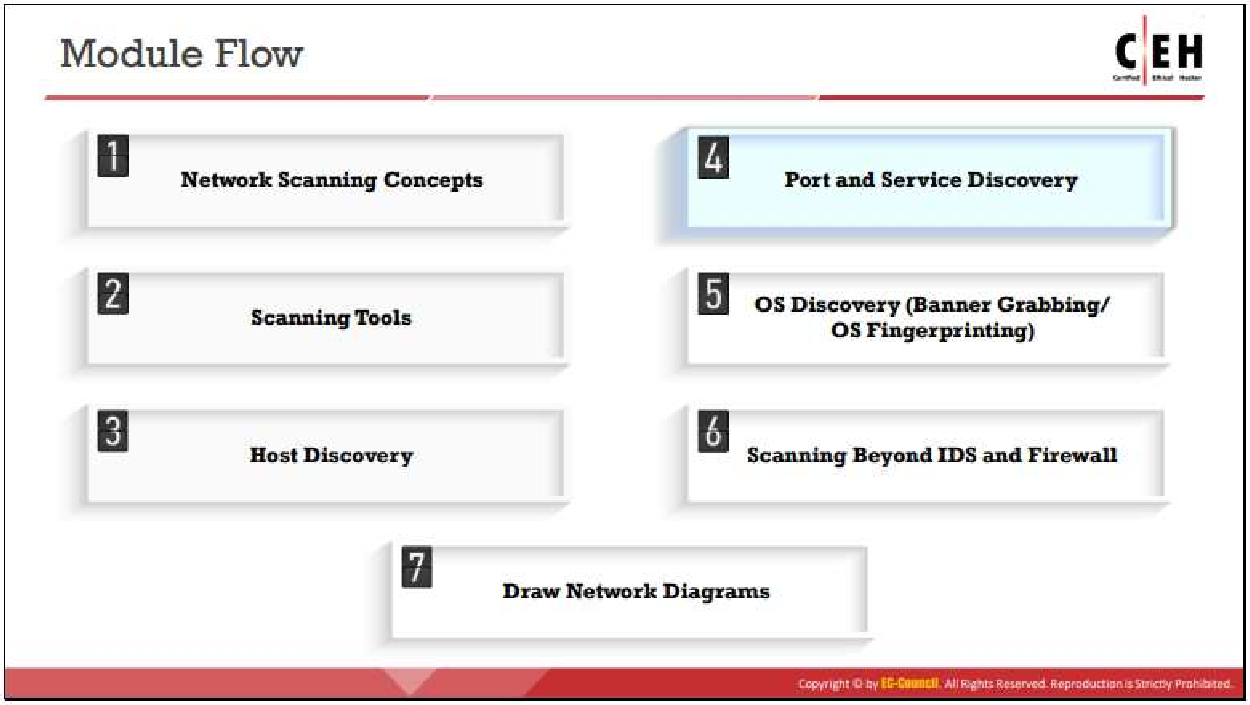


Figure 3.33: IP protocol ping scan in Zenmap



Port and Service Discovery

The next step in the network scanning process involves checking the open ports and services in live systems. After performing a ping scan, once attackers detect the live systems in the target network, they try to find open ports and services in the discovered live systems. This discovery of open ports and services can be performed via various port scanning techniques. Administrators often use port scanning techniques to verify the security policies of their networks, whereas attackers use them to identify open ports and running services on a host with the intent of compromising the network. Moreover, sometimes, users unknowingly keep unnecessary open ports on their systems. An attacker takes advantage of such open ports to launch attacks.

This section describes the common ports and corresponding services along with various port scanning techniques and tools used by the attacker to perform port scanning.

List of Common Ports and Services

The important reserved ports are listed below:

Name	Port/Protocol	Service Description
echo	7/tcp	
echo	7/udp	
discard	9/tcp	sink null
discard	9/udp	sink null
systat	11/tcp	Users
daytime	13/tcp	

daytime	13/udp	
netstat	15/tcp	
qotd	17/tcp	Quote
chargen	19/tcp	ttytst source
chargen	19/udp	ttytst source
ftp-data	20/tcp	ftp data transfer
ftp	21/tcp	ftp command
ssh	22/tcp	Secure Shell
telnet	23/tcp	
SMTP	25/tcp	Mail
time	37/tcp	Timeserver
time	37/udp	Timeserver
rlp	39/udp	resource location
nickname	43/tcp	who is
domain	53/tcp	domain name server
domain	53/udp	domain name server
sql*net	66/tcp	Oracle SQL*net
sql*net	66/udp	Oracle SQL*net
bootps	67/tcp	bootp server
bootps	67/udp	bootp server
bootpc	68/tcp	bootp client
bootpc	68/udp	bootp client
tftp	69/tcp	Trivial File Transfer
tftp	69/udp	Trivial File Transfer
gopher	70/tcp	gopher server
finger	79/tcp	Finger
www-http	80/tcp	WWW
www-http	80/udp	WWW
kerberos	88/tcp	Kerberos
kerberos	88/udp	Kerberos
pop2	109/tcp	PostOffice V.2
Pop3	110/tcp	PostOffice V.3
sunrpc	111/tcp	RPC 4.0 portmapper
sunrpc	111/udp	RPC 4.0 portmapper
auth/ident	113/tcp	Authentication Service
auth	113/udp	Authentication Service

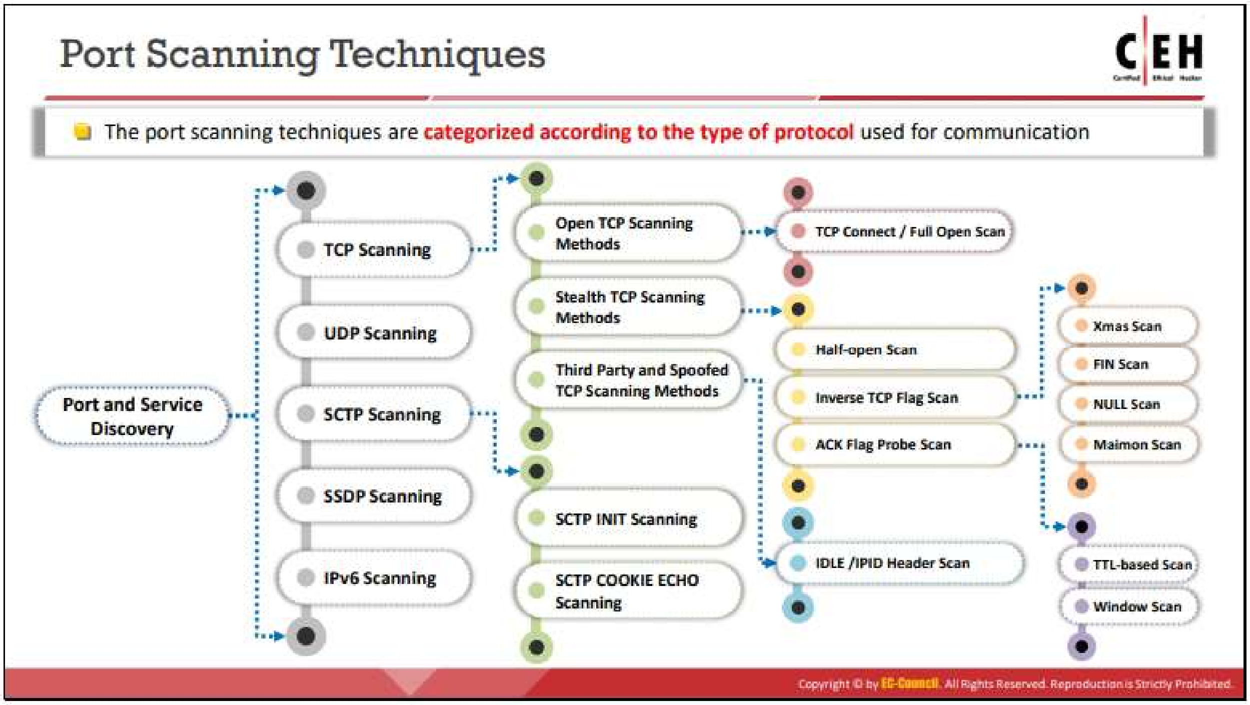
audionews	114/tcp	Audio News Multicast
audionews	114/udp	Audio News Multicast
nntp	119/tcp	Usenet Network News Transfer
nntp	119/udp	Usenet Network News Transfer
ntp	123/tcp	Network Time Protocol
Name	Port/Protocol	Description
ntp	123/udp	Network Time Protocol
netbios-ns	137/tcp	NETBIOS Name Service
netbios-ns	137/udp	NETBIOS Name Service
netbios-dgm	138/tcp	NETBIOS Datagram Service
netbios-dgm	138/udp	NETBIOS Datagram Service
netbios-ssn	139/tcp	NETBIOS Session Service
netbios-ssn	139/udp	NETBIOS Session Service
imap	143/tcp	Internet Message Access Protocol
imap	143/udp	Internet Message Access Protocol
sql-net	150/tcp	SQL-NET
sql-net	150/udp	SQL-NET
sqlsrv	156/tcp	SQL Service
sqlsrv	156/udp	SQL Service
snmp	161/tcp	
snmp	161/udp	
snmp-trap	162/tcp	
snmp-trap	162/udp	
cmip-man	163/tcp	CMIP/TCP Manager
cmip-man	163/udp	CMIP
cmip-agent	164/tcp	CMIP/TCP Agent
cmip-agent	164/udp	CMIP
irc	194/tcp	Internet Relay Chat
irc	194/udp	Internet Relay Chat
at-rtmp	201/tcp	AppleTalk Routing Maintenance
at-rtmp	201/udp	AppleTalk Routing Maintenance
at-nbp	202/tcp	AppleTalk Name Binding
at-nbp	202/udp	AppleTalk Name Binding
at-3	203/tcp	AppleTalk
at-3	203/udp	AppleTalk
at-echo	204/tcp	AppleTalk Echo

at-echo	204/udp	AppleTalk Echo
at-5	205/tcp	AppleTalk
at-5	205/udp	AppleTalk
at-zis	206/tcp	AppleTalk Zone Information
at-zis	206/udp	AppleTalk Zone Information
at-7	207/tcp	AppleTalk
at-7	207/udp	AppleTalk
at-8	208/tcp	AppleTalk
at-8	208/udp	AppleTalk
ipx	213/tcp	Novel
ipx	213/udp	Novel
imap3	220/tcp	Interactive Mail Access Protocol v3
imap3	220/udp	Interactive Mail Access Protocol v3
aurp	387/tcp	AppleTalk Update-Based Routing
aurp	387/udp	AppleTalk Update-Based Routing
netware-ip	396/tcp	Novell Netware over IP
netware-ip	396/udp	Novell Netware over IP
Name	Port/Protocol	Description
rmt	411/tcp	Remote mt
rmt	411/udp	Remote mt
kerberos-ds	445/tcp	Microsoft DS
kerberos-ds	445/udp	Microsoft DS
isakmp	500/udp	ISAKMP/IKE
fcp	510/tcp	First Class Server
exec	512/tcp	BSD rexecd(8)
comsat/biff	512/udp	used by mail system to notify users
login	513/tcp	BSD rlogind(8)
who	513/udp	whod BSD rwhod(8)
shell	514/tcp	cmd BSD rshd(8)
syslog	514/udp	BSD syslogd(8)
printer	515/tcp	spooler BSD lpd(8)
printer	515/udp	Printer Spooler
talk	517/tcp	BSD talkd(8)
talk	517/udp	Talk
ntalk	518/udp	New Talk (ntalk)
ntalk	518/udp	SunOS talkd(8)

netnews	532/tcp	Readnews
uucp	540/tcp	uucpd BSD uucpd(8)
uucp	540/udp	uucpd BSD uucpd(8)
klogin	543/tcp	Kerberos Login
klogin	543/udp	Kerberos Login
kshell	544/tcp	Kerberos Shell
kshell	544/udp	Kerberos Shell
ekshell	545/tcp	krcmd Kerberos encrypted remote shell –kfall
pcserver	600/tcp	ECD Integrated PC board srvr
mount	635/udp	NFS Mount Service
pcnfs	640/udp	PC-NFS DOS Authentication
bwnfs	650/udp	BW-NFS DOS Authentication
flexlm	744/tcp	Flexible License Manager
flexlm	744/udp	Flexible License Manager
kerberos-adm	749/tcp	Kerberos Administration
kerberos-adm	749/udp	Kerberos Administration
kerberos	750/tcp	kdc Kerberos authentication—tcp
kerberos	750/udp	Kerberos
kerberos_master	751/udp	Kerberos authentication
kerberos_master	751/tcp	Kerberos authentication
krb_prop	754/tcp	Kerberos slave propagation
	999/udp	Applixware
socks	1080/tcp	
socks	1080/udp	
kpop	1109/tcp	Pop with Kerberos
ms-sql-s	1433/tcp	Microsoft SQL Server
ms-sql-s	1433/udp	Microsoft SQL Server
ms-sql-m	1434/tcp	Microsoft SQL Monitor
ms-sql-m	1434/udp	Microsoft SQL Monitor
pptp	1723/tcp	Pptp
pptp	1723/udp	Pptp
nfs	2049/tcp	Network File System
nfs	2049/udp	Network File System
eklogin	2105/tcp	Kerberos encrypted rlogin

rkinit	2108/tcp	Kerberos remote kinit
kx	2111/tcp	X over Kerberos
kauth	2120/tcp	Remote kauth
lyskom	4894/tcp	LysKOM (conference system)
sip	5060/tcp	Session Initiation Protocol
sip	5060/udp	Session Initiation Protocol
x11	6000-6063/tcp	X Window System
x11	6000-6063/udp	X Window System
irc	6667/tcp	Internet Relay Chat
afs	7000-7009/udp	Andrew File System
afs	7000-7009/udp	Andrew File System

Table 3.2: Reserved ports table



Port Scanning Techniques

Port scanning techniques are further categorized as described below. This categorization is based on the type of protocol used for communication in the network.

TCP Scanning:

- Open TCP Scanning Methods
  - TCP Connect/Full Open Scan
- Stealth TCP Scanning Methods
  - Half-open Scan
  - Inverse TCP Flag Scan
    - Xmas Scan
    - FIN Scan
    - NULL Scan
    - Maimon Scan
  - ACK Flag Probe Scan
    - TTL-Based Scan
    - Window Scan
- Third Party and Spoofed TCP Scanning Methods
  - IDLE/IP ID Header Scan

UDP Scanning:

- UDP Scanning

SCTP Scanning:

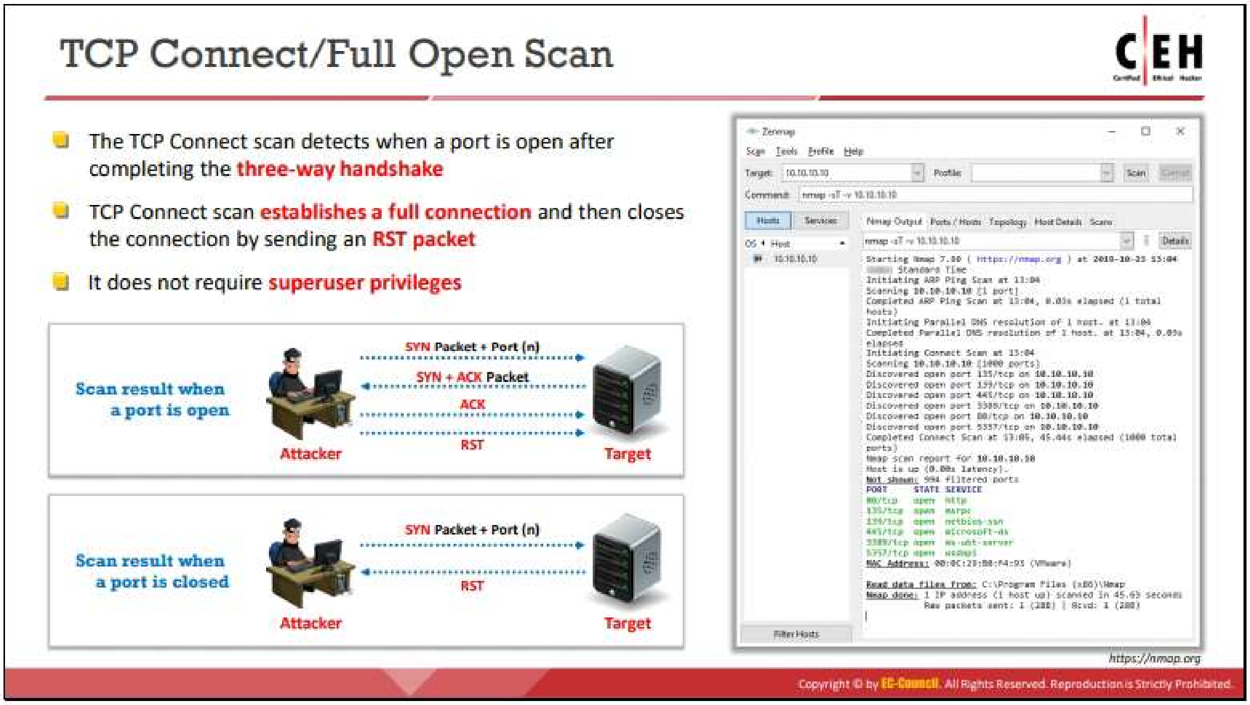
- SCTP INIT Scanning
- SCTP COOKIE/ECHO Scanning

SSDP Scanning:

- SSDP and List Scanning

IPv6 Scanning:

- IPv6 Scanning



TCP Connect/Full Open Scan

Source: *http://insecure.org*

TCP Connect/Full Open Scan is one of the most reliable forms of TCP scanning. In TCP Connect scanning, the OS's TCP `connect()` system call tries to open a connection to every port of interest on the target machine. If the port is listening, the `connect()` call will result in a successful connection with the host on that particular port; otherwise, it will return an error message stating that the port is not reachable.

TCP Connect scan completes a three-way handshake with the target machine. In the TCP three-way handshake, the client sends a SYN packet, which the recipient acknowledges with a SYN+ACK packet. Then, the client acknowledges the SYN+ACK packet with an ACK packet to complete the connection. Once the handshake is completed, the scanner sends an RST packet to end the connection.

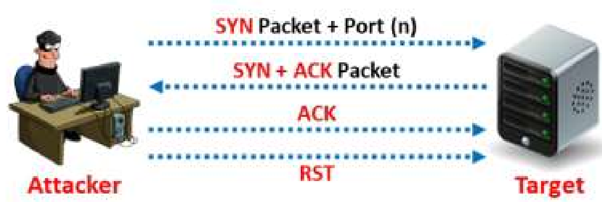


Figure 3.34: Scan result when a port is open

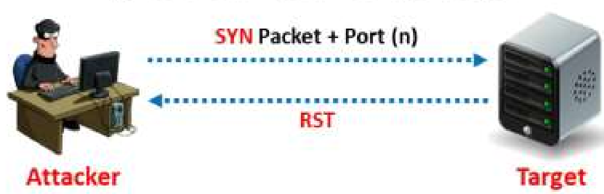


Figure 3.35: Scan result when a port is closed

Making a separate **`connect()`** call for every targeted port in a linear manner would take a long time over a slow connection. The attacker can accelerate the scan using many sockets in parallel. Using non-blocking, I/O allows the attacker to set a short time-out period and watch all the sockets simultaneously. In Zenmap, the **`-sT`** option is used to perform TCP Connect/full open scan.

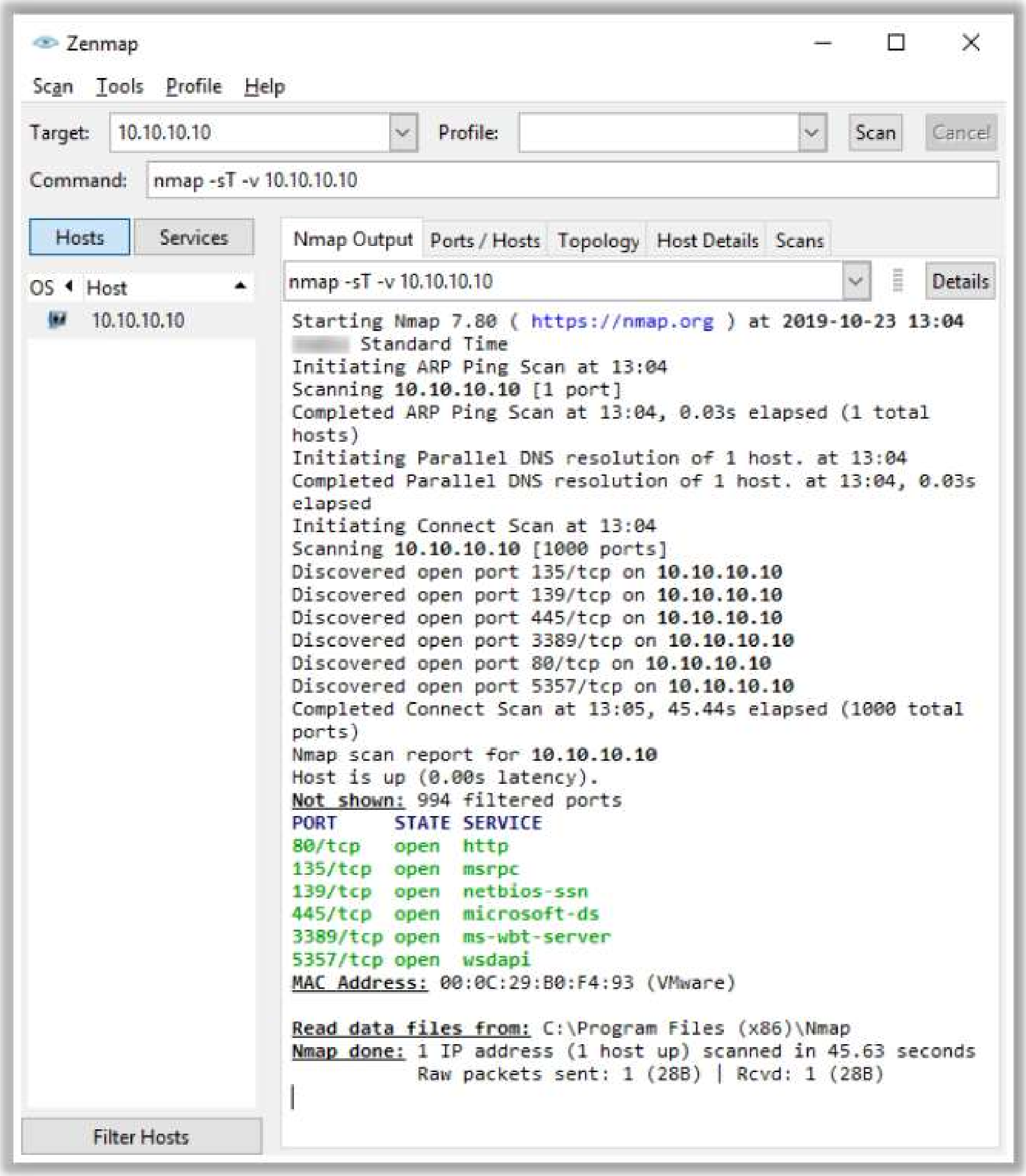
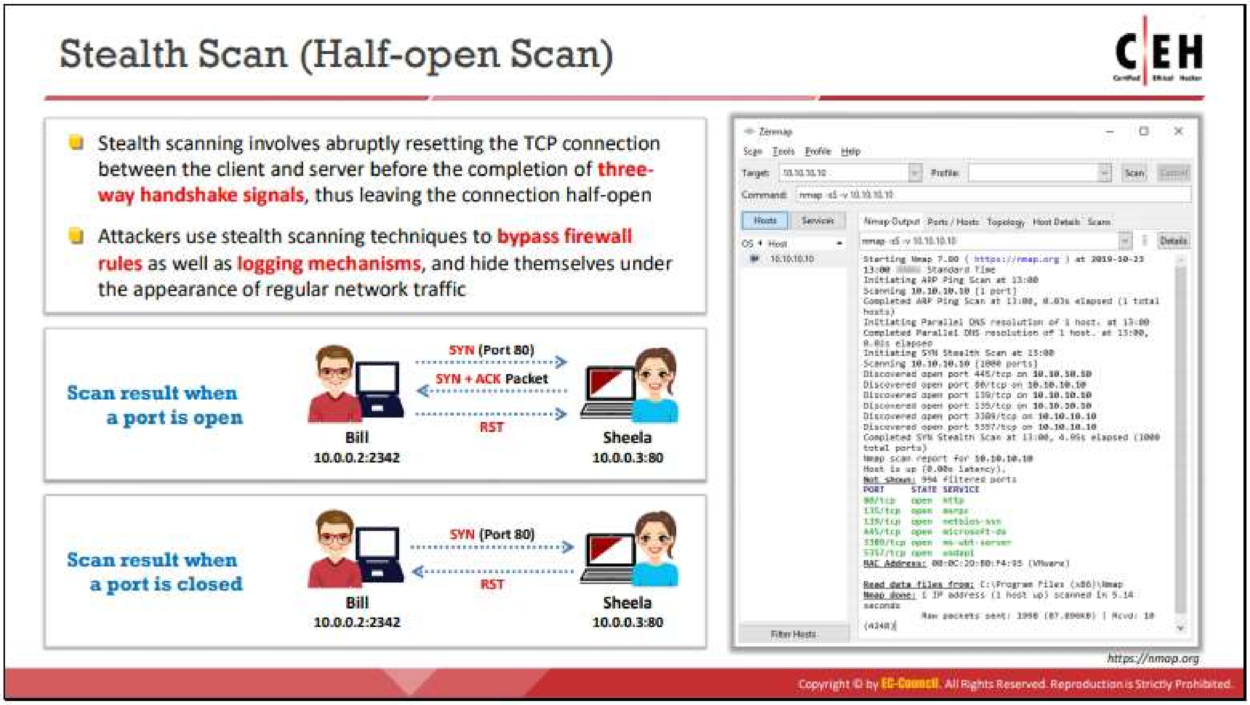


Figure 3.36: TCP Connect/Full Open scan using Zenmap

The drawback of this type of scan is that it is easily detectable and filterable. The logs in the target system will disclose the connection. Such scanning does not require superuser privileges.



Stealth Scan (Half-open Scan)

The stealth scan involves resetting the TCP connection between the client and the server abruptly before completion of the three-way handshake signals, hence making the connection half-open. A stealth scan sends a single frame to a TCP port without any TCP handshaking or additional packet transfers. This type of scan sends a single frame with the expectation of a single response. The half-open scan partially opens a connection but stops halfway through. The stealth scan is also called a “SYN scan,” because it only sends the SYN packet. This prevents the service from notifying the incoming connection. TCP SYN or half-open scanning is a stealth method of port scanning.

The stealth scan also implements the three-way handshake methodology. In the last stage, it examines the packets entering the interface and terminates the connection before triggering a new initialization to identify remote ports. The stealth scan process is described below.

- The client sends a single SYN packet to the server on the appropriate port.
- If the port is open, the server subsequently responds with a SYN/ACK packet.
- If the server responds with an RST packet, then the remote port is in the "closed” state.
- The client sends the RST packet to close the initiation before a connection can be established.

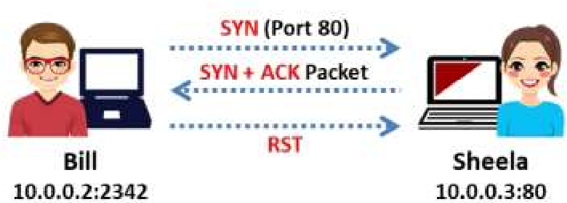


Figure 3.37: Port is open

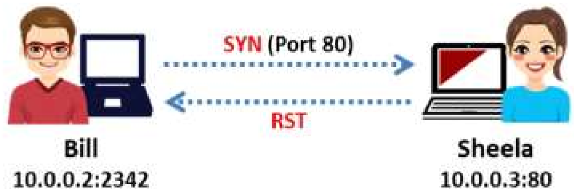


Figure 3.38: Port is closed

Attackers use stealth scanning techniques to bypass firewall rules and logging mechanisms, and they hide themselves as usual under network traffic. In Zenmap, the **-sS** option is used to perform a stealth scan/TCP half-open scan.

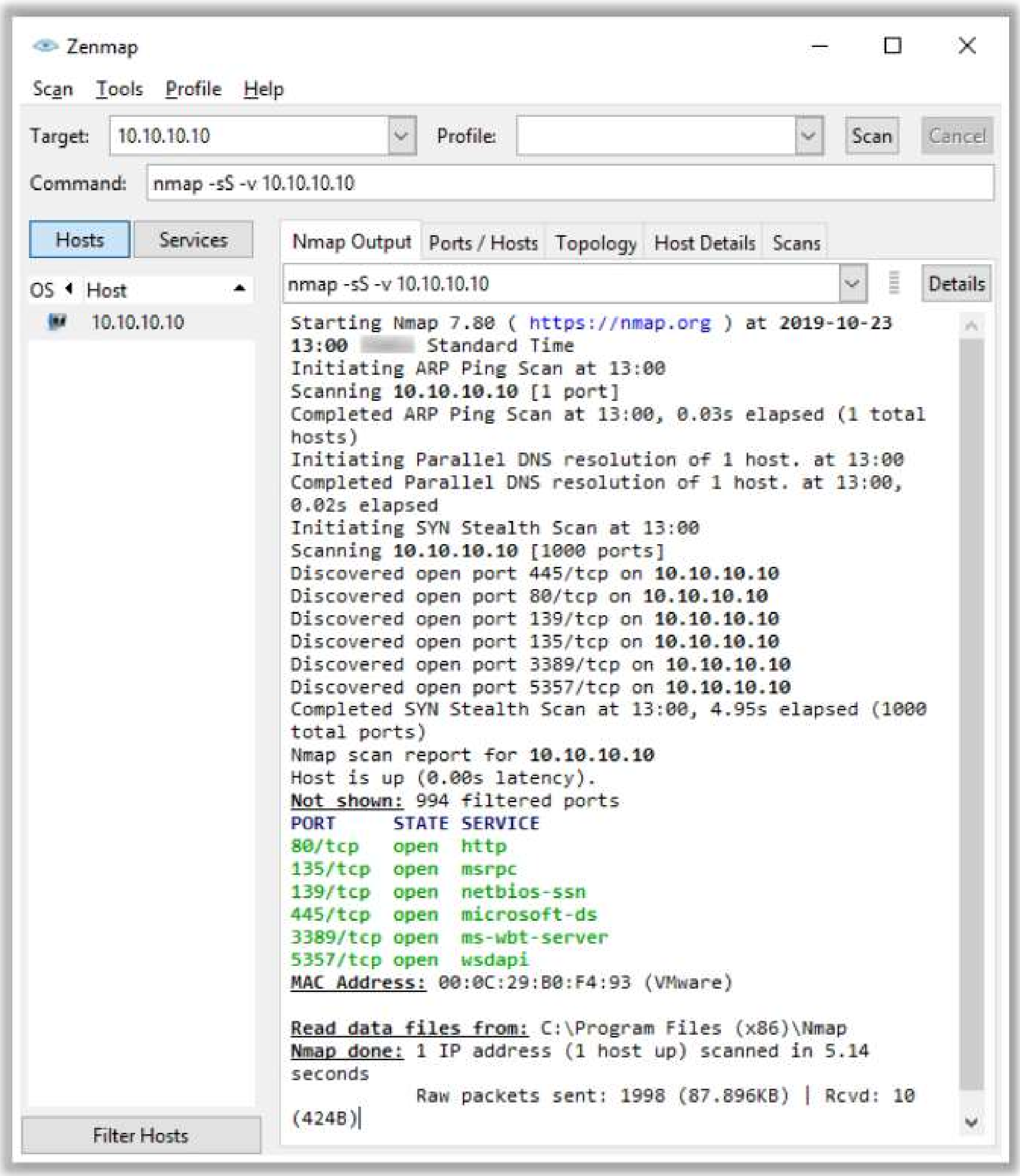
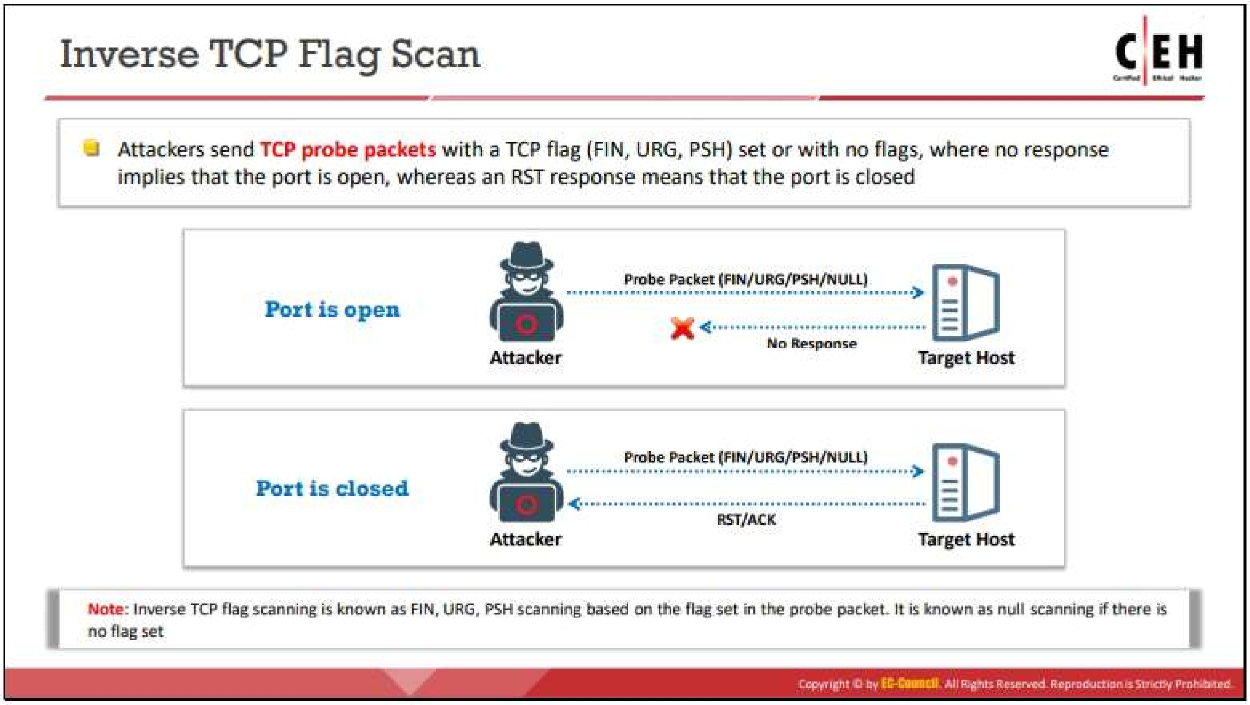


Figure 3.39: TCP Stealth/Half Open scan using Zenmap



Inverse TCP Flag Scan

Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags. When the port is open, the attacker does not get any response from the host, whereas when the port is closed, he or she receives the RST from the target host.

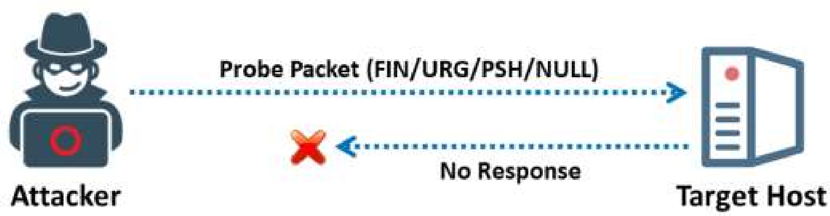


Figure 3.40: Inverse TCP flag scan when port is open

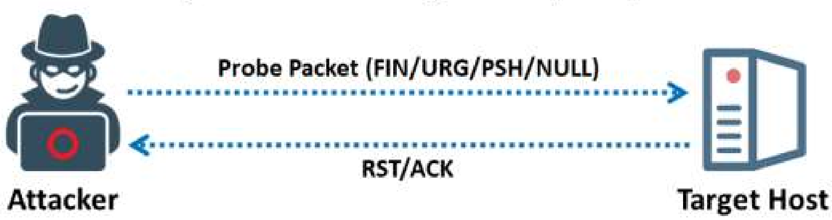


Figure 3.41: Inverse TCP flag scan when port is closed

Security mechanisms such as firewalls and IDS detect the SYN packets sent to the sensitive ports of the targeted hosts. Programs such as Synlogger and Courtney are available to log half-open SYN flag scan attempts. At times, the probe packets enabled with TCP flags can pass through filters undetected, depending on the security mechanisms installed.

An inverted technique involves probing a target using a half-open SYN flag because the closed ports can only send the response back. According to RFC 793, an RST/ACK packet is sent for connection reset when the host closes a port. Attackers take advantage of this feature to send TCP probe packets to each port of the target host with various TCP flags set.

Common flag configurations used for a probe packet include:

- A FIN probe with the FIN TCP flag set
- An Xmas probe with the FIN, URG, and PUSH TCP flags set
- A NULL probe with no TCP flags set
- A SYN/ACK probe

All closed ports on the targeted host will send an RST/ACK response. Since OSs such as Windows completely ignore the RFC 793 standard, you cannot see the RST/ACK response when connected to a closed port on the target host. However, this technique is effective when used with UNIX-based OSs.

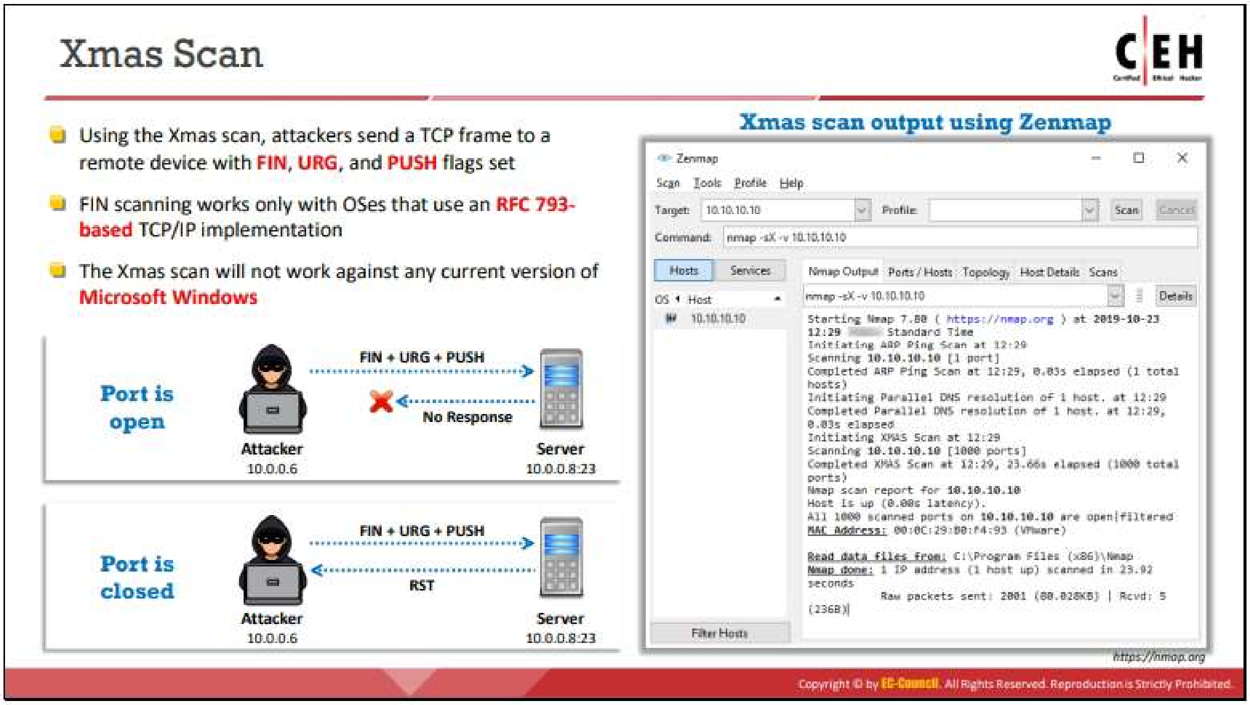
Advantages

- Avoids many IDS and logging systems; highly stealthy

Disadvantages

- Needs raw access to network sockets, thus requiring super-user privileges
- Mostly effective against hosts using a BSD-derived TCP/IP stack (not effective against Microsoft Windows hosts, in particular).

Note: Inverse TCP flag scanning is known as FIN, URG, and PSH scanning based on the flag set in the probe packet. If there is no flag set, it is known as NULL scanning. If only the FIN flag is set, it is known as FIN scanning, and if all of FIN, URG, and PSH are set, it is known as Xmas scanning.



Xmas Scan

Xmas scan is a type of inverse TCP scanning technique with the FIN, URG, and PUSH flags set to send a TCP frame to a remote device. If the target has opened the port, then you will receive no response from the remote system. If the target has closed the port, then you will receive a remote system reply with an RST. You can use this port scanning technique to scan large networks and find which host is up and what services it is offering. This technique describes all TCP flag sets. When all flags are set, some systems hang; hence, the flags are often set in the nonsense pattern URG-PSH-FIN. Attackers use the TCP Xmas scan to determine if ports are closed on the target machine via the RST packet. This scan only works when systems are compliant with RFC 793-based TCP/IP implementation. It will not work against any current version of Microsoft Windows.

Figure 3.42: Xmas scan when the port is open

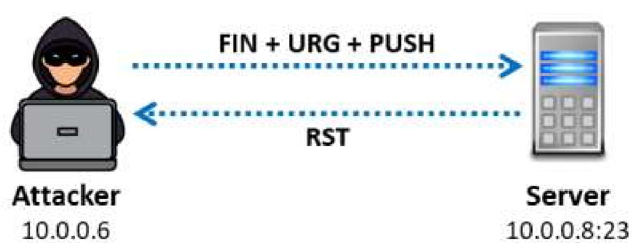


Figure 3.43: Xmas scan when the port is closed

BSD Networking Code

This method relies on the BSD networking code. Thus, you can use this only for UNIX hosts; it does not support Windows NT. If the user scans any Microsoft system, it will show that all the ports on the host are open.

Transmitting Packets

You can initialize all the flags when transmitting the packet to a remote host. If the target system accepts the packet and does not send any response, it means that the port is open. If the target system sends an RST flag, then it implies that the port is closed.

Advantages

- It avoids IDS and TCP three-way handshake.

Disadvantages

- It works on the UNIX platform only.

In Zenmap, the **-sX** option is used to perform Xmas scan whereas the **-sF** and **-sN** options are used to perform FIN scan and NULL scan, respectively.

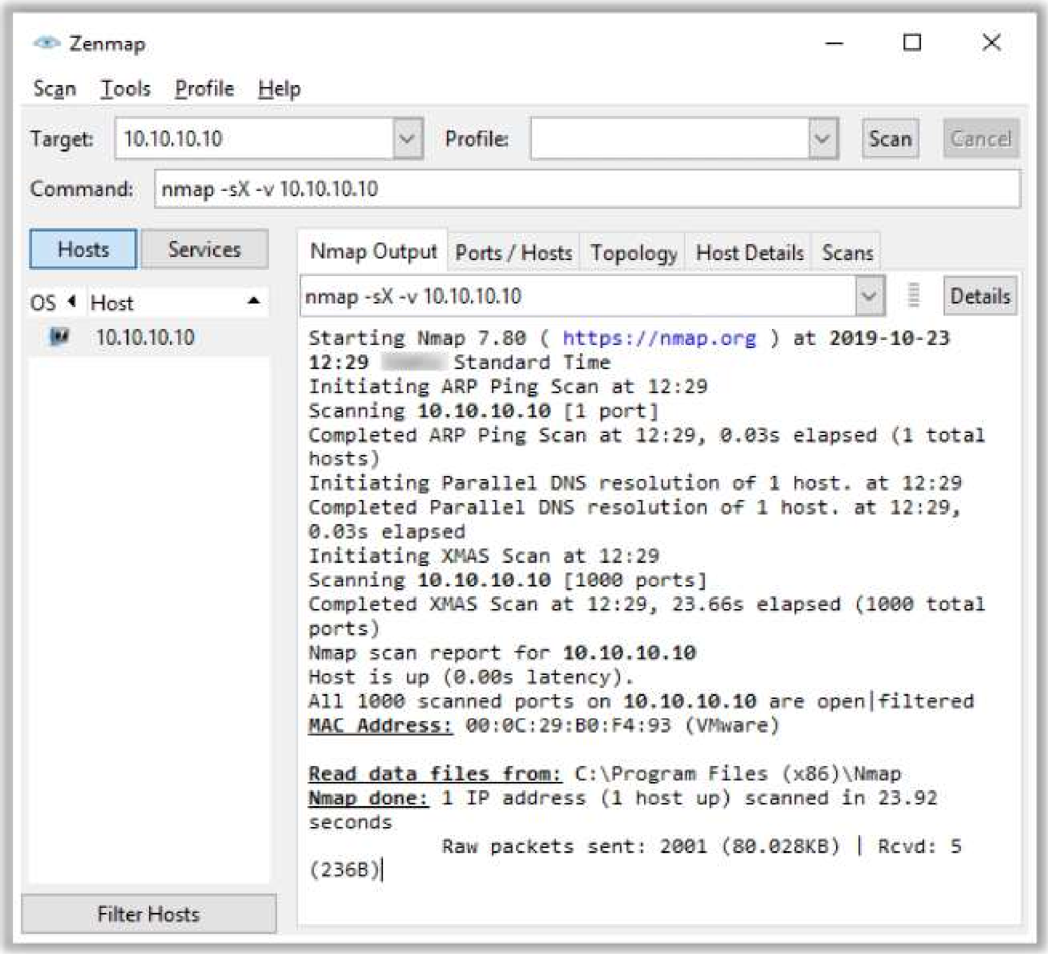
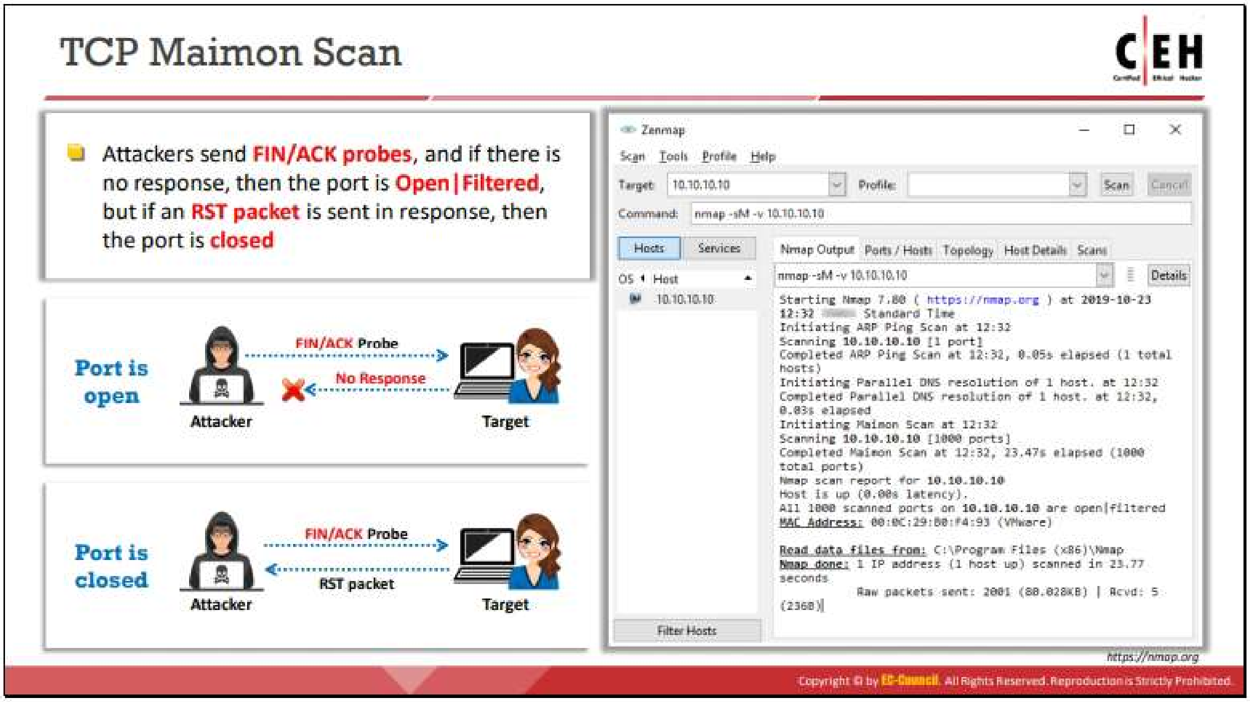


Figure 3.44: Xmas scan output using Zenmap



TCP Maimon scan

This scan technique is very similar to NULL, FIN, and Xmas scan, but the probe used here is FIN/ACK. In most cases, to determine if the port is open or closed, the RST packet should be generated as a response to a probe request. However, in many BSD systems, the port is open if the packet gets dropped in response to a probe.

Nmap interprets a port as open|filtered when there is no response from the Maimon scan probe even after many retransmissions. The port is closed if the probe gets a response as an RST packet. The port is filtered when the ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is returned from the target host. In Zenmap, the **-sM** option is used to perform the TCP Maimon scan.

Figure 3.45: TCP Maimon scan result of open port

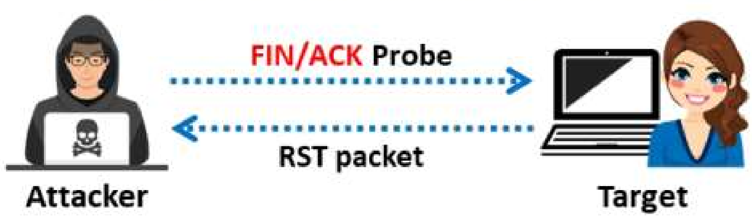


Figure 3.46: TCP Maimon scan result of closed port

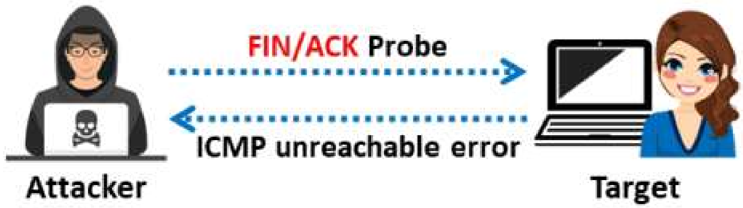


Figure 3.47: TCP Maimon scan result of filtered port

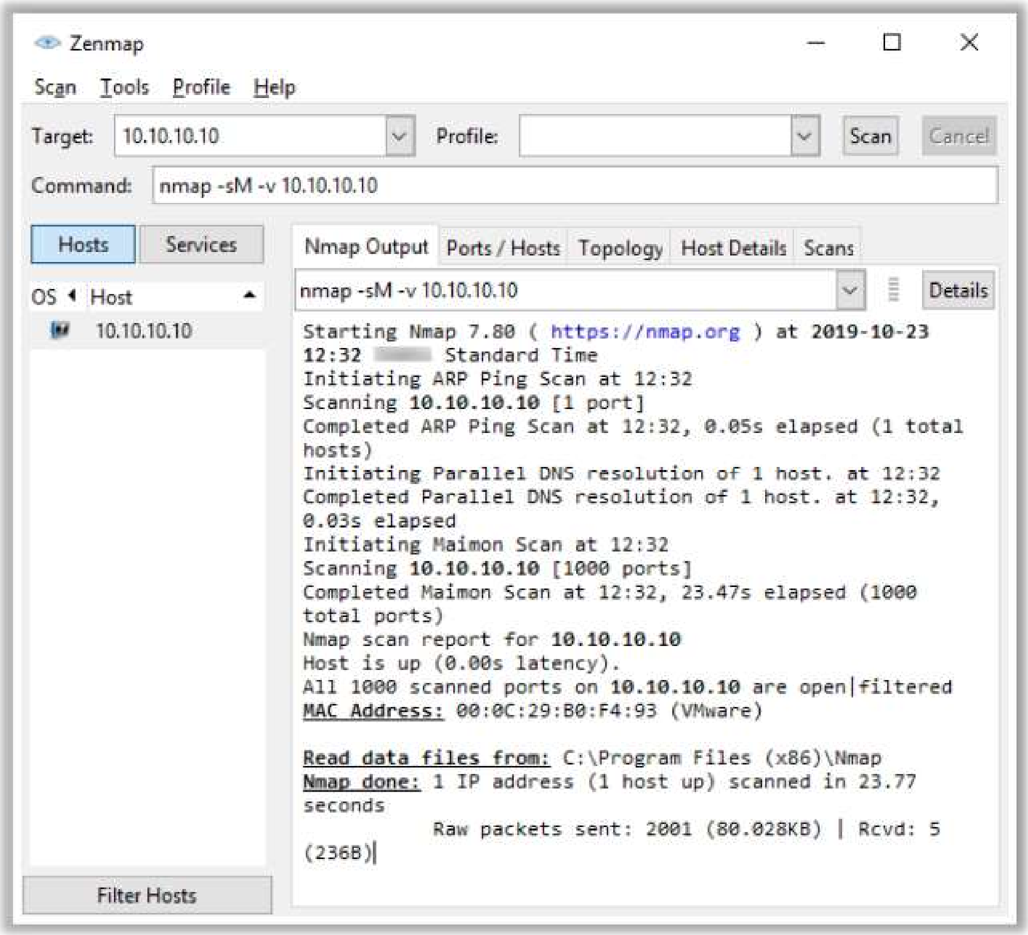
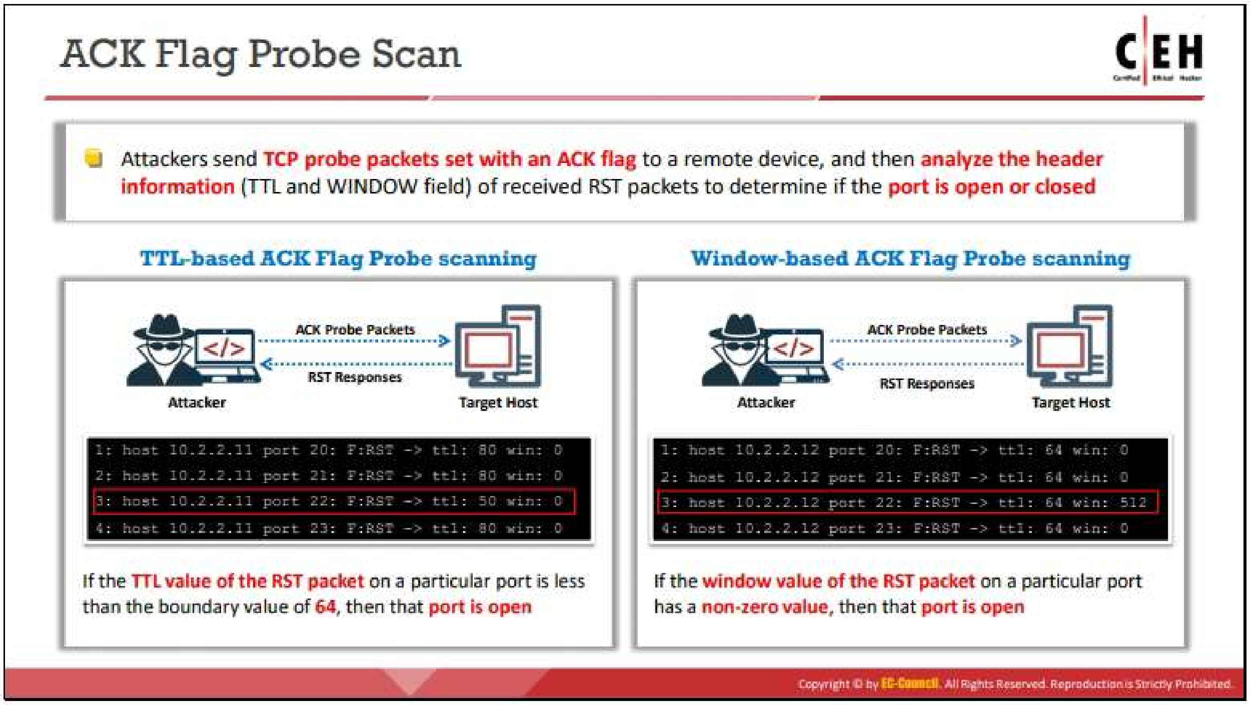
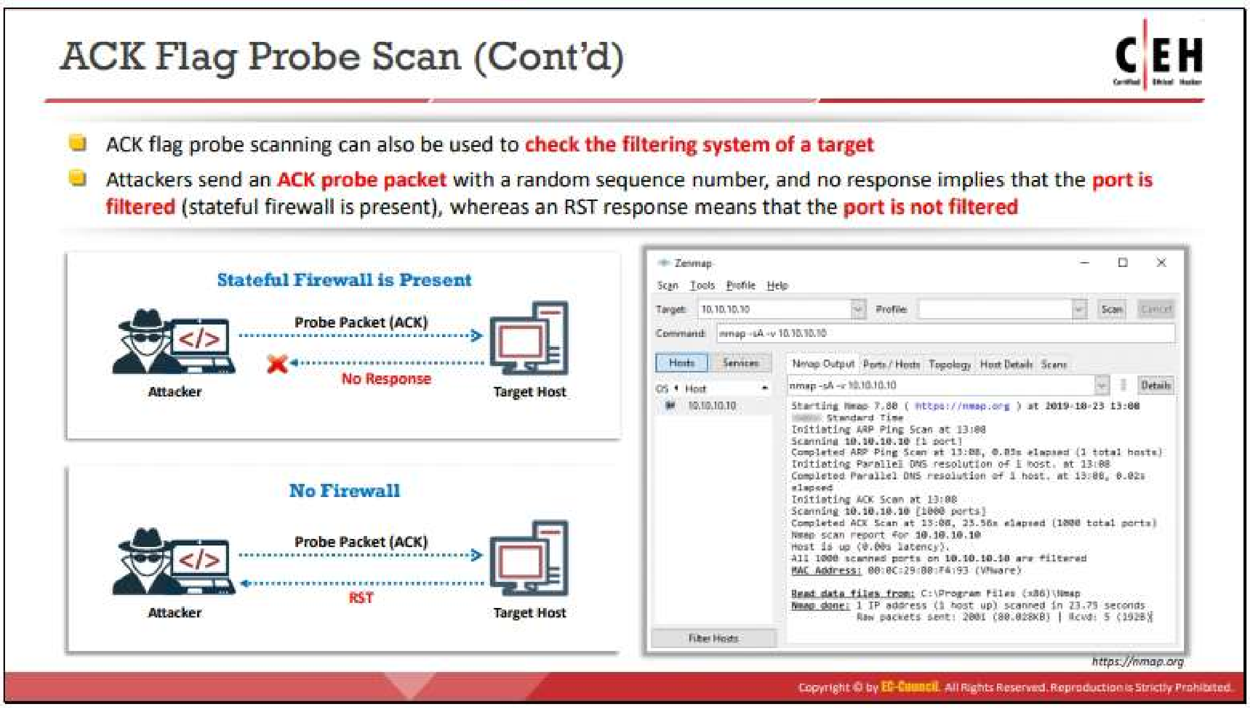


Figure 3.48: TCP Maimon scan displaying port state in Zenmap





ACK Flag Probe Scan

Attackers send TCP probe packets with the ACK flag set to a remote device and then analyze the header information (TTL and WINDOW field) of the received RST packets to find out if the port is open or closed. The ACK flag probe scan exploits the vulnerabilities within the BSD-derived TCP/IP stack. Thus, such scanning is effective only on those OSs and platforms on which the BSD derives TCP/IP stacks.

Categories of ACK flag probe scanning include:

- **TTL-based ACK Flag Probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (several thousands) to different TCP ports and then analyze the TTL field value of the RST packets received. In Zenmap, the syntax `nmap -ttl [time] [target]` is used to perform TTL-based scan.

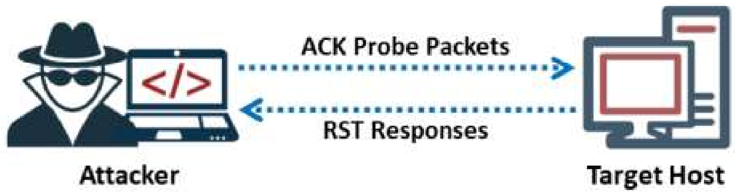


Figure 3.49: TTL-based ACK flag probe scanning

  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. An example showing a log of the first four RST packets received is presented below:

```
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
```

Figure 3.50: Screenshot showing the open port based on the TTL value of the RST packet

  In this example, port 22 returned a TTL value of 50, which is less than 64; all other ports returned a TTL value of 80, which is greater than 64. Therefore, port 22 is open.

- **Window-based ACK Flag Probe scanning**

  In this scanning technique, you will first need to send ACK probe packets (several thousands) to different TCP ports and then analyze the window field value of the received RST packets. The user can use this scanning technique when all the ports return the same TTL value. In Zenmap, the `-sW` option is used to perform a window scan.

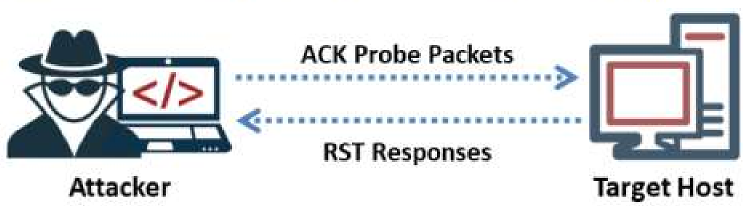


Figure 3.51: Window-based ACK flag probe scanning

  If the window value of the RST packet on a particular port is non-zero, then that port is open. An example showing a log of the first four RST packets received is presented below:

```
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
```

Figure 3.52: Screenshot showing the open port based on the window value of the RST packet

The above figure shows that the TTL value returned for each packet is the same; hence, you cannot perform TTL-based ACK flag probe scanning to find the open ports. Therefore, when you observe the window value, the third packet has a non-zero window value, which means that the port is open. When the returned RST value is zero, then the port is closed. If there is no response even after many retransmissions and an ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is returned, then the port is inferred to be a filtered port.

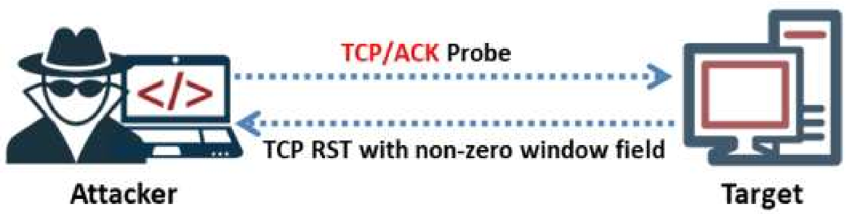


Figure 3.53: TCP Window scan result of an open port

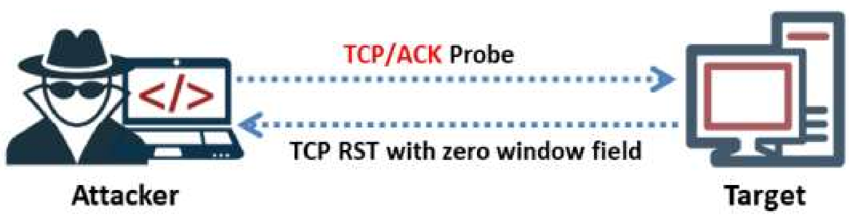


Figure 3.54: TCP Window scan result of a closed port

Figure 3.55: TCP Window scan result of a filtered port

Advantages:

- This type of scan can evade IDS in most cases.

Disadvantages:

- It is extremely slow and can exploit only older OSs with vulnerable BSD-derived TCP/IP stacks.

Checking the Filtering Systems of Target Networks

The ACK flag probe scanning technique also helps in checking the filtering systems of target networks. The attacker sends an ACK probe packet to check the filtering mechanism (firewalls) of packets employed by the target network.

Sending an ACK probe packet with a random sequence number and getting no response from the target means that the port is filtered (stateful firewall is present); an RST response from the target means that the port is not filtered (no firewall is present).

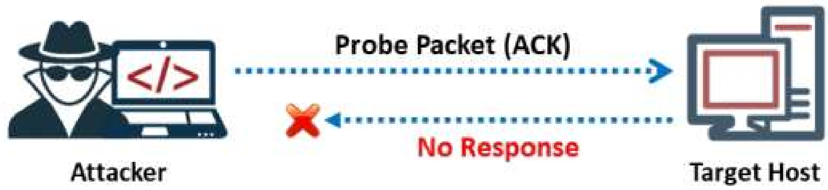


Figure 3.56: Stateful Firewall is present

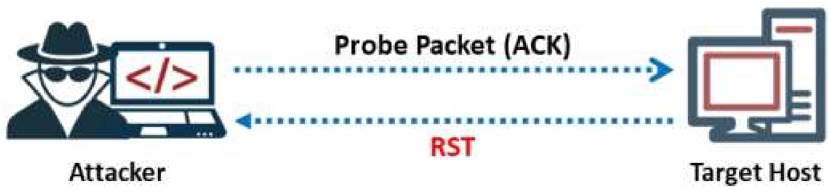


Figure 3.57: No Firewall

ACK Flag Probe Scanning using Nmap

In Zenmap, the **`-sA`** option is used to perform an ACK flag probe scan.

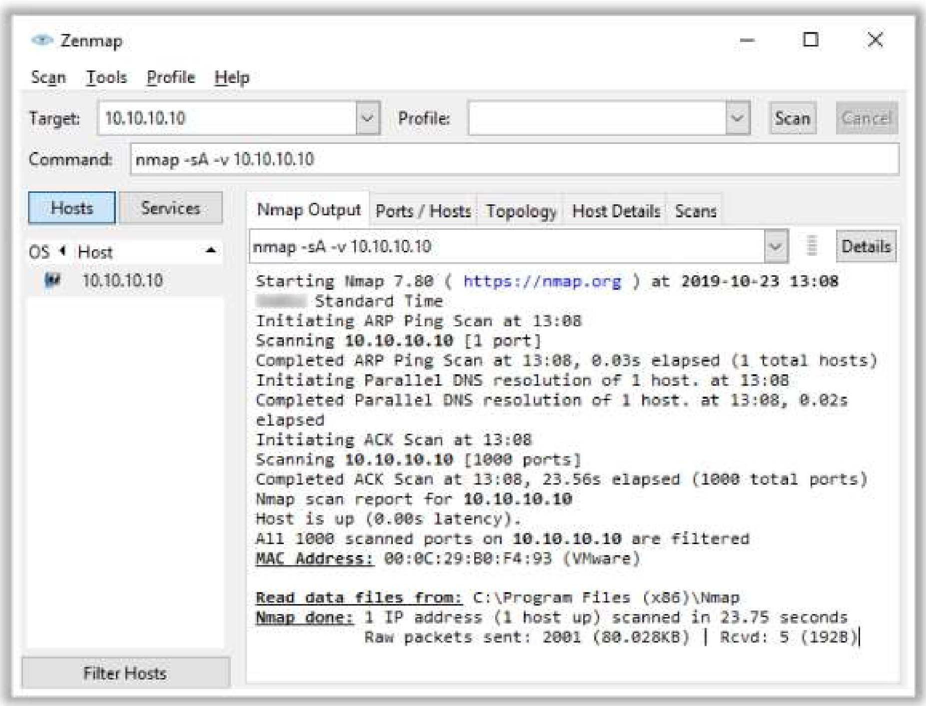
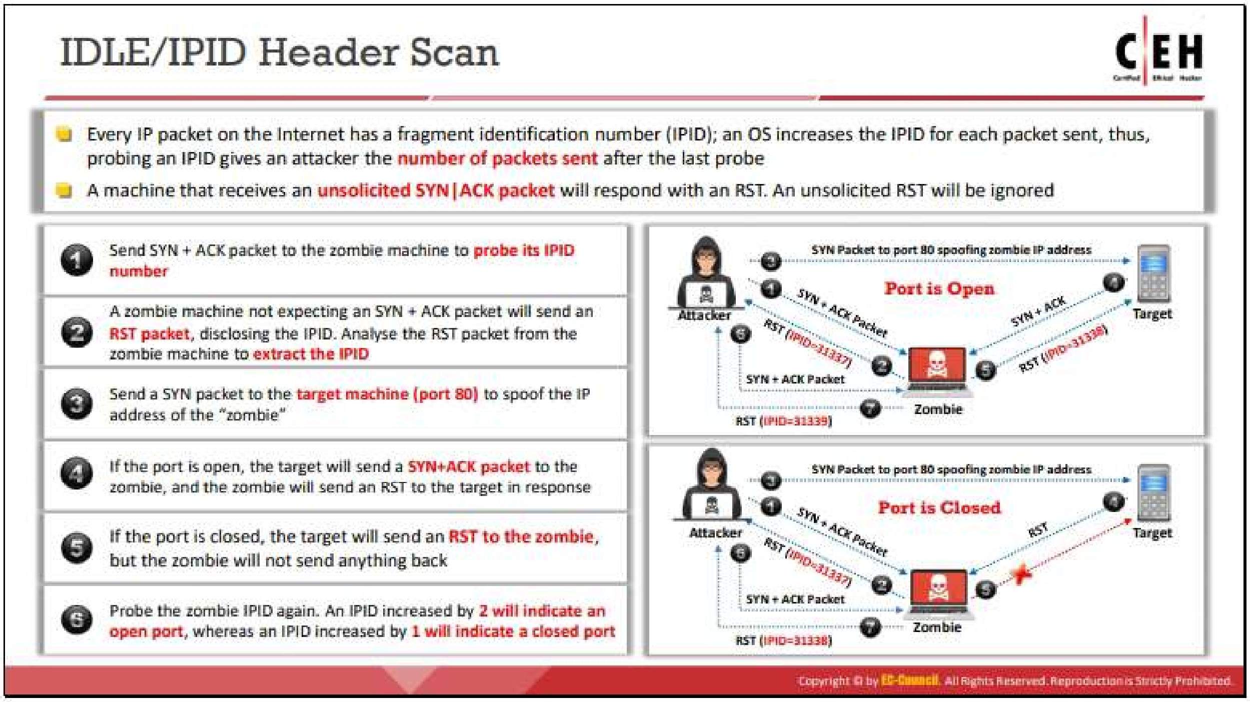


Figure 3.58: ACK Flag Probe scanning using Zenmap



IDLE/IPID Header Scan

The IDLE/IPID Header scan is a TCP port scan method that you can use to send a spoofed source address to a computer to find out what services are available. It offers complete blind scanning of a remote host. Most network servers listen on TCP ports, such as web servers on port 80 and mail servers on port 25. A port is considered "open" if an application is listening on the port. One way to determine whether a port is open is to send a "SYN" (session establishment) packet to the port. The target machine will send back a "SYN|ACK" (session request acknowledgement) packet if the port is open or an "RST" (Reset) packet if the port is closed. A machine that receives an unsolicited SYN|ACK packet will respond with an RST. An unsolicited RST will be ignored. Every IP packet on the Internet has a "fragment identification" number (IPID). The OS increases the IPID for each packet sent; thus, probing an IPID gives an attacker the number of packets sent since the last probe. In Zenmap, the **-sI** option is used to perform the IDLE scan.

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
```

Figure 3.59: IDLE/IPID Header scan using Zenmap

The attacker performs this scan by impersonating another computer via spoofing. The attacker does not send a packet from her/his IP address; instead, he/she uses another host, often called a "zombie," to scan the remote host and identify any open ports. In this attack, the attacker expects the sequence numbers of the zombie host, and if the remote host checks the IP of the scanning party, the IP of the zombie machine will be displayed.

IDLE Scan

Every IP packet on the Internet has a fragment Internet protocol identification (IPID) number that uniquely identifies fragments of an original IP datagram. As many OSs simply increase this number for each packet that they send, probing the IPID can tell an attacker how many packets the user sent since the last probe.

- **Step 1**

  The first step in performing an idle scan is to find an appropriate zombie. A zombie that assigns IPID packets incrementally on a global basis is an appropriate or idle zombie for performing the idle scan. The shorter the time interval for request/response between the attacker-zombie and the zombie-target, the faster is the scan.

  Choose a "Zombie" and Probe its Current IP Identification (IPID) Number

  In the first step, you will send the SYN+ACK packet to the zombie machine to probe its IPID number. Here, the SYN+ACK packet is sent to probe the IPID number but not establish a TCP connection (three-way handshake).

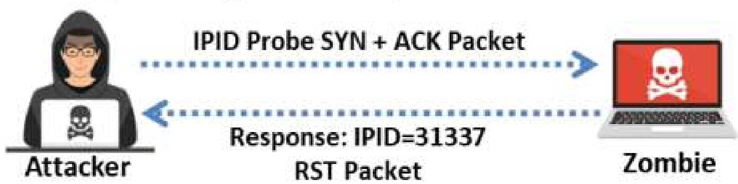


Figure 3.60: IDLE scan: Step 1

  As the zombie does not expect a SYN+ACK packet, it will deny the connection by sending back an RST packet. Analyze the RST packet sent by the zombie machine to extract the IPID. In the diagram shown in the slide above, assume that the zombie responds with `IPID=31337`. Furthermore, assume that this IPID is X.

- **Step 2**

  The attacker sends a SYN packet to the target machine on port 80, spoofing the IP address of the zombie.

  Idle Scan: Step 2.1 (Open Port)

  If the port is open, the target will send the SYN+ACK packet to the zombie (as the IP address was spoofed) to proceed with the three-way handshake. Since the zombie did not expect a SYN+ACK packet from the target machine, it will respond with an RST packet.

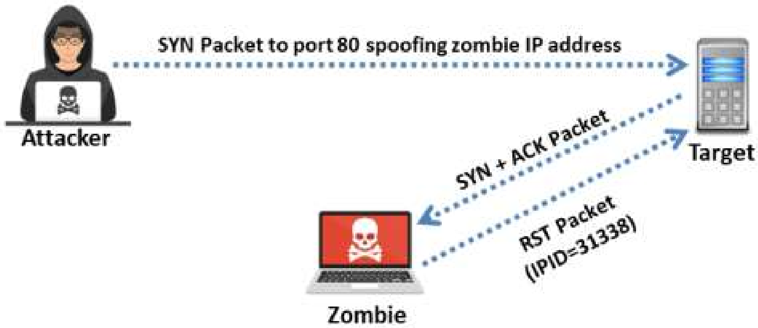


Figure 3.61: Port is open

Since every IP packet has a "fragment identification" number, which increases by one for every packet transmission, the zombie will now use its next available IPID, i.e., 31338 (X + 1).

Idle Scan: Step 2.2 (Closed Port)

Assume that the port on the target is closed. Subsequently, on receiving the SYN packet from the attacker (you), the target will respond with an RST, and the zombie will remain idle without taking any further action.

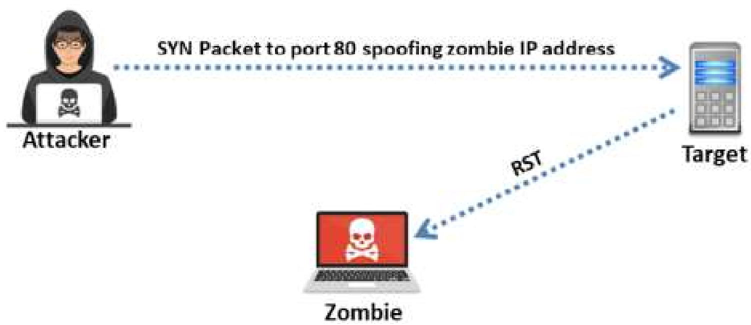


Figure 3.62: Port is closed

- **Step 3**

  Now, follow step 1 again to probe the IPID number.

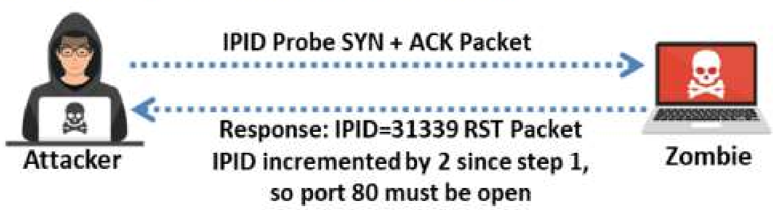
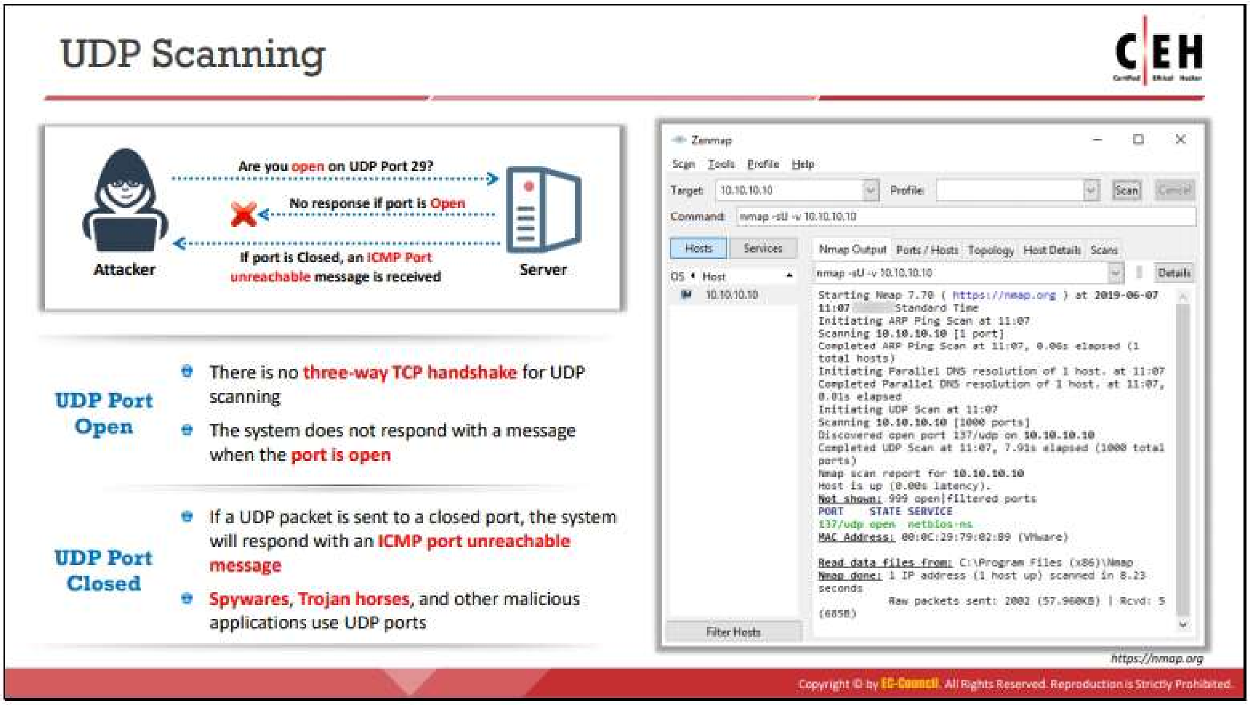


Figure 3.63: Idle scan: Step 3

Send a SYN+ACK packet to the zombie, and it will respond with an RST packet containing the IPID. Assume that the port on the target was open and that the zombie has already sent an RST packet to the target; then, the IPID number is increased by 1. Now, the zombie responds with an RST packet to the attacker using its next IPID, i.e., 31339 (X + 2). Consequently, the IPID is increased by 2, which implies that the port on the target machine was open. Thus, using an idle scan, an attacker can find out the open ports and services on the target machine by spoofing his/her IP address with a zombie's IP address.



UDP Scanning

UDP Raw ICMP Port Unreachable Scanning

UDP port scanners use the UDP protocol instead of TCP. There is no three-way handshake for the UDP scan. The UDP protocol can be more challenging to use than TCP scanning because you can send a packet but you cannot determine whether the host is alive, dead, or filtered. However, you can use one ICMP that checks for open or closed ports. If you send a UDP packet to a port without an application bound to it, the IP stack will return an ICMP port unreachable packet. If any port returns an ICMP error, it will be closed, leaving the ports that did not answer if they are open or filtered through the firewall.

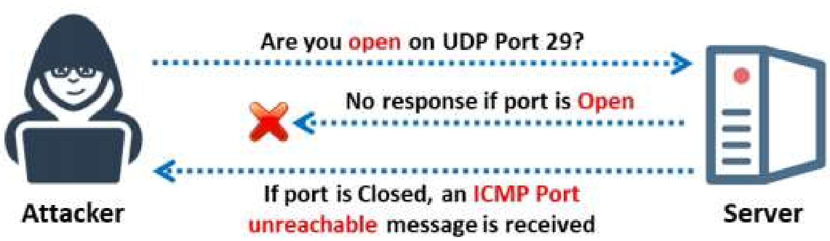


Figure 3.64: UDP scanning

This happens because open ports do not have to send an acknowledgement in response to a probe, and closed ports are not even required to send an error packet.

UDP Packets

Source: *https://nmap.org*

When you send a packet to a closed UDP port, most of the hosts send an `ICMP_PORT_UNREACH` error. Thus, you can determine whether a port is not open if UDP packets or ICMP errors are not guaranteed to arrive. Thus, UDP scanners of this type must implement retransmission of packets that appear lost. UDP scanners interpret lost traffic as open ports. In Zenmap, the **`-sU`** option is used to perform a UDP scan.

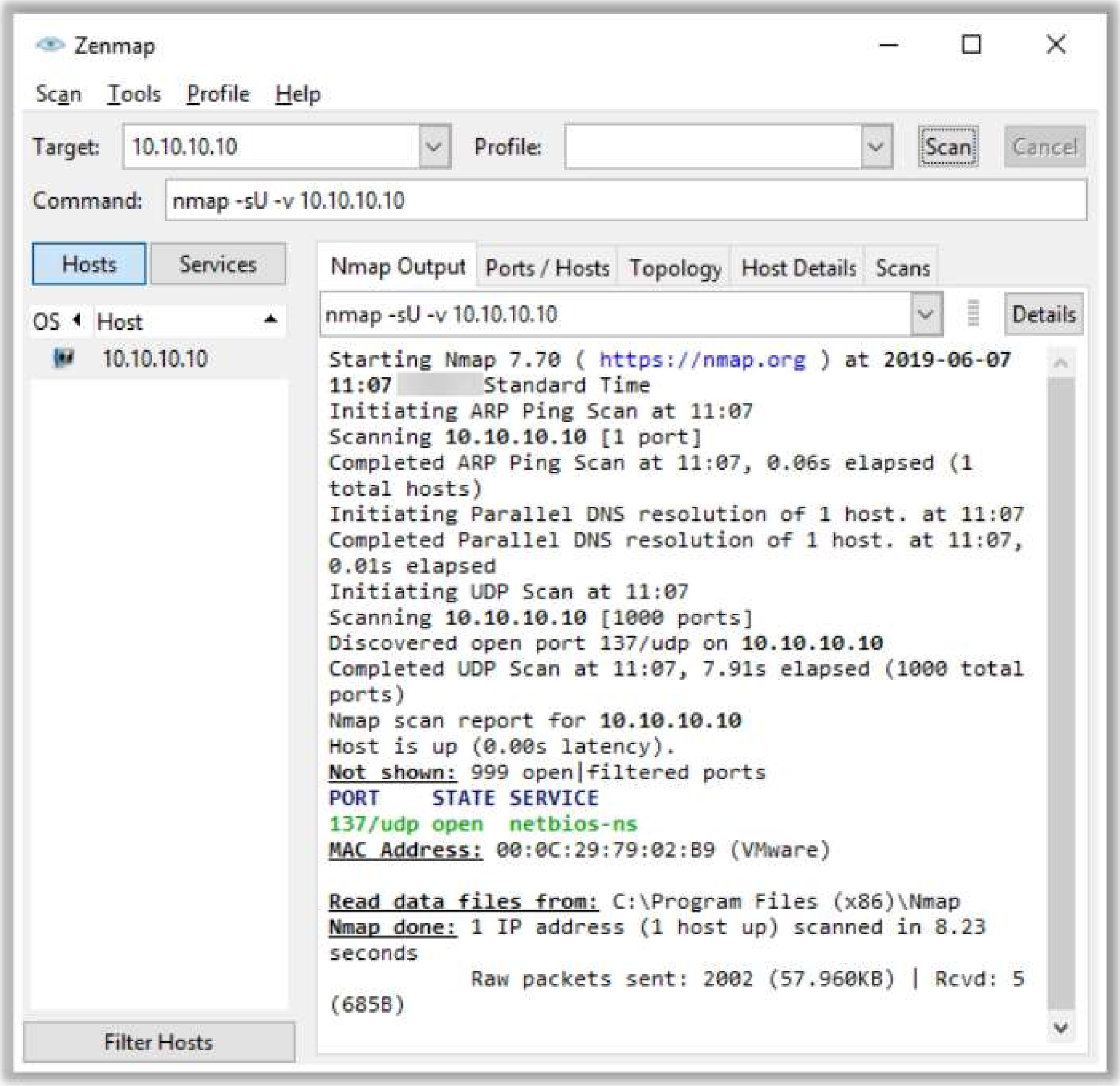


Figure 3.65: UDP scanning using Zenmap

In addition, this scanning technique is slow because it limits the ICMP error message rate as a form of compensation to machines that apply RFC 1812 section 4.3.2.8. A remote host will require access to the raw ICMP socket to distinguish closed ports from unreachable ports.

UDP RECVFROM () and WRITE () Scanning

Although non-root users cannot read unreachable port errors directly, Linux informs you indirectly when it receives messages.

- **Example:**

  For example, a second **`write ()`** call to a closed port will usually fail. Various scanners, such as Netcat and Pluvial **`pscan.c`**, perform **`recvfrom ()`** on non-blocking UDP sockets, and they usually return **`EAGAIN`** ("**`Try Again`**," **`errno 13`**) if the ICMP error has not been received or `ECONNREFUSED` (`"Connection refused,"` `errno 111`) otherwise. This technique is used for determining open ports when non-root users use `-u` (UDP). Root users can also use the `-l` (lamer UDP scan) option to force this process.

Advantage:

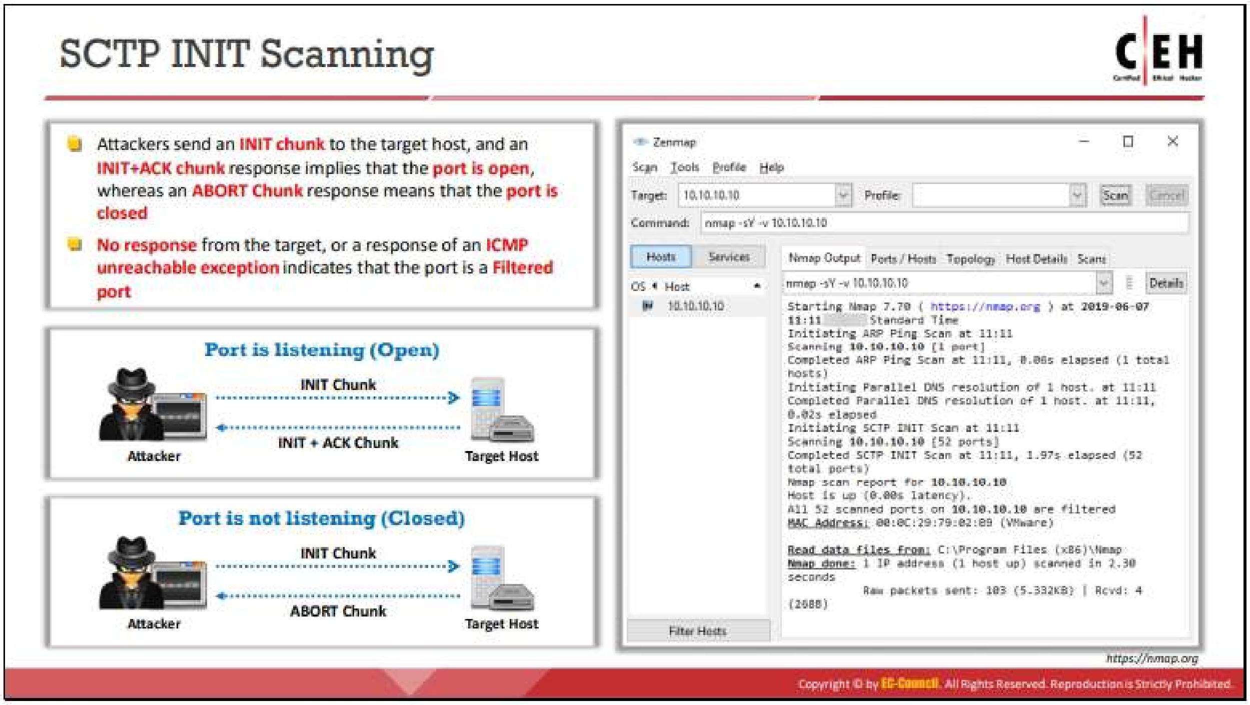
The UDP scan is less informal with regard to an open port because there is no overhead of a TCP handshake. However, if ICMP is responding to each unavailable port, the total number of frames can exceed that from a TCP scan. Microsoft-based OSs do not usually implement any ICMP rate limiting; hence, this scan operates very efficiently on Windows-based devices.

Disadvantage:

The UDP scan provides port information only. If additional information of the version is needed, the scan must be supplemented with a version detection scan (`-sV`) or the OS fingerprinting option (`-O`).

The UDP scan requires privileged access; hence, this scan option is only available on systems with the appropriate user permissions.

Most networks have massive amounts of TCP traffic; as a result, the efficiency of the UDP scan is low. The UDP scan will locate open ports and provide the security manager with valuable information for identifying successful attacker invasions on open UDP ports owing to spyware applications, Trojan horses, and other malicious software.



SCTP INIT Scanning

Stream Control Transport Protocol (SCTP) is a reliable message-oriented transport layer protocol. It is used as an alternative to the TCP and UDP protocols, as its characteristics are similar to those of TCP and UDP. SCTP is specifically used to perform multi-homing and multi-streaming activities. Some SCTP applications include discovering VoIP, IP telephony, and Signaling System 7/SIGnaling TRANsport (SS7/SIGTRAN)-related services. SCTP association comprises a four-way handshake method, as shown in the screenshot below.

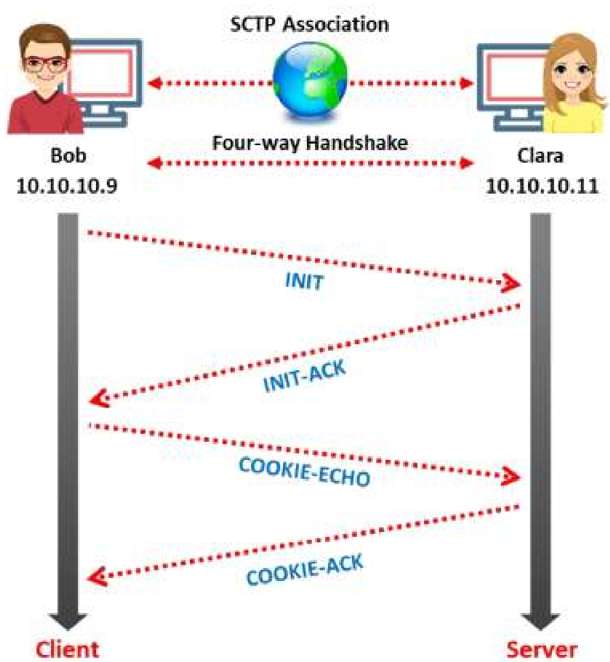


Figure 3.66: SCTP Association four-way handshake

In SCTP, the INIT scan is performed quickly by scanning thousands of ports per second on a fast network not obstructed by a firewall offering a stronger sense of security. The SCTP INIT scan is very similar to the TCP SYN scan; comparatively, it is also stealthy and unobtrusive, as it cannot complete SCTP associations, hence making the connection half-open.

Attackers send INIT chunk to the target host. If the port is listening or open, it sends an acknowledgement as an INIT+ACK chunk.

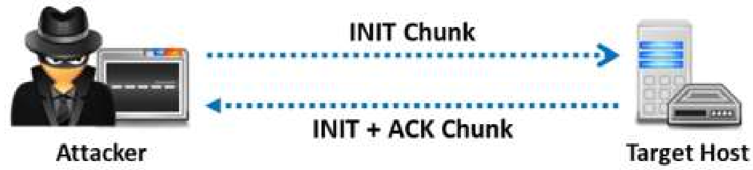


Figure 3.67: SCTP INIT scan result when a port is listening (Open)

If the target is inactive and it is not listening, then it sends an acknowledgement as an ABORT chunk.

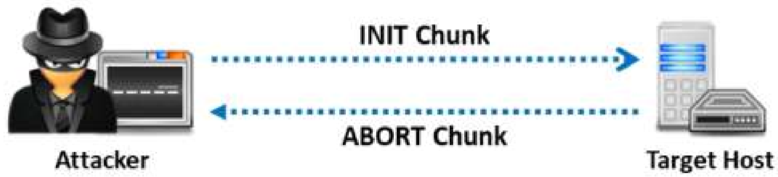


Figure 3.68: SCTP INIT scan result when a port is not listening (Closed)

After several retransmissions, if there is no response, then the port is indicated as a filtered port. The port is also indicated as a filtered port if the target server responds with an ICMP unreachable exception (type 3, code 0, 1, 2, 3, 9, 10, or 13). In Zenmap, the **`-sY`** option is used to perform the SCTP INIT scan.

Advantages:

- INIT scan can clearly differentiate between various ports such as open, closed, and filtered states

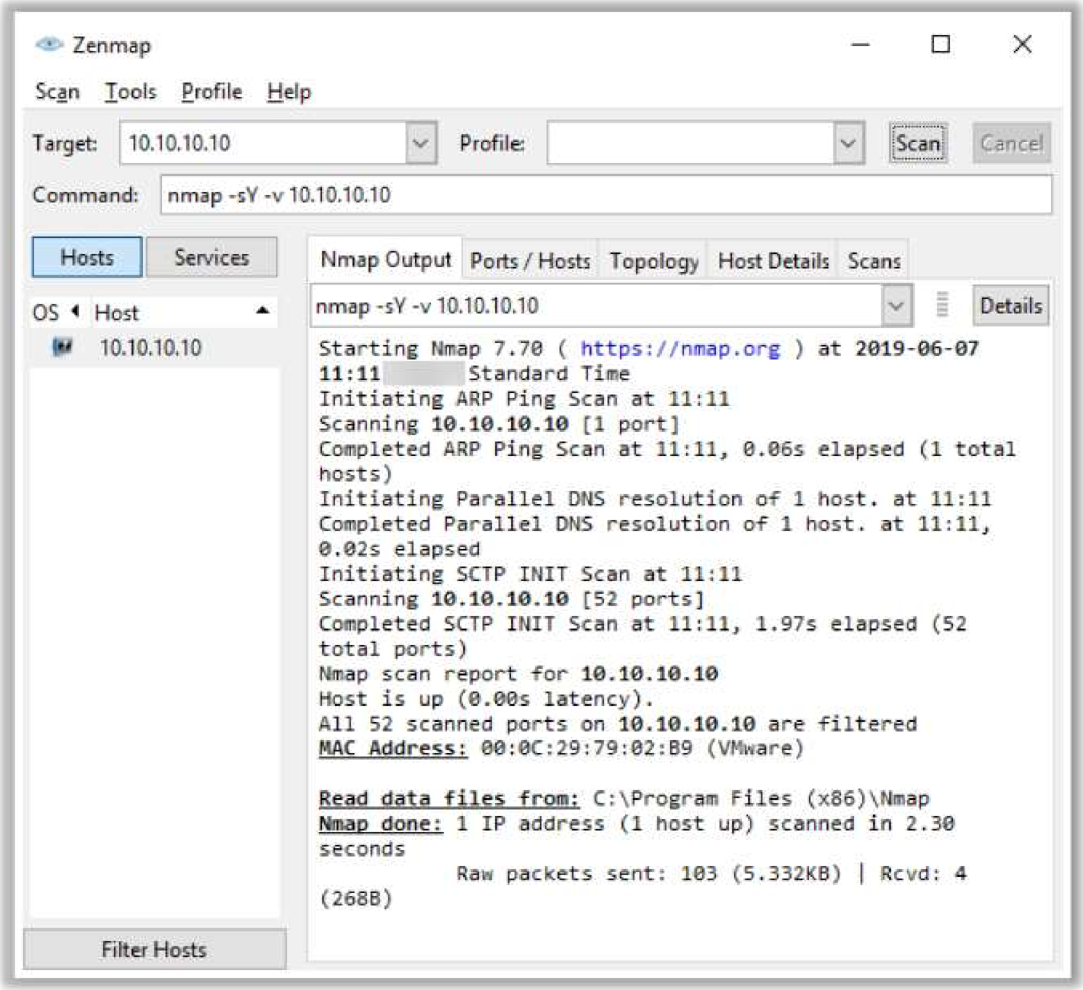
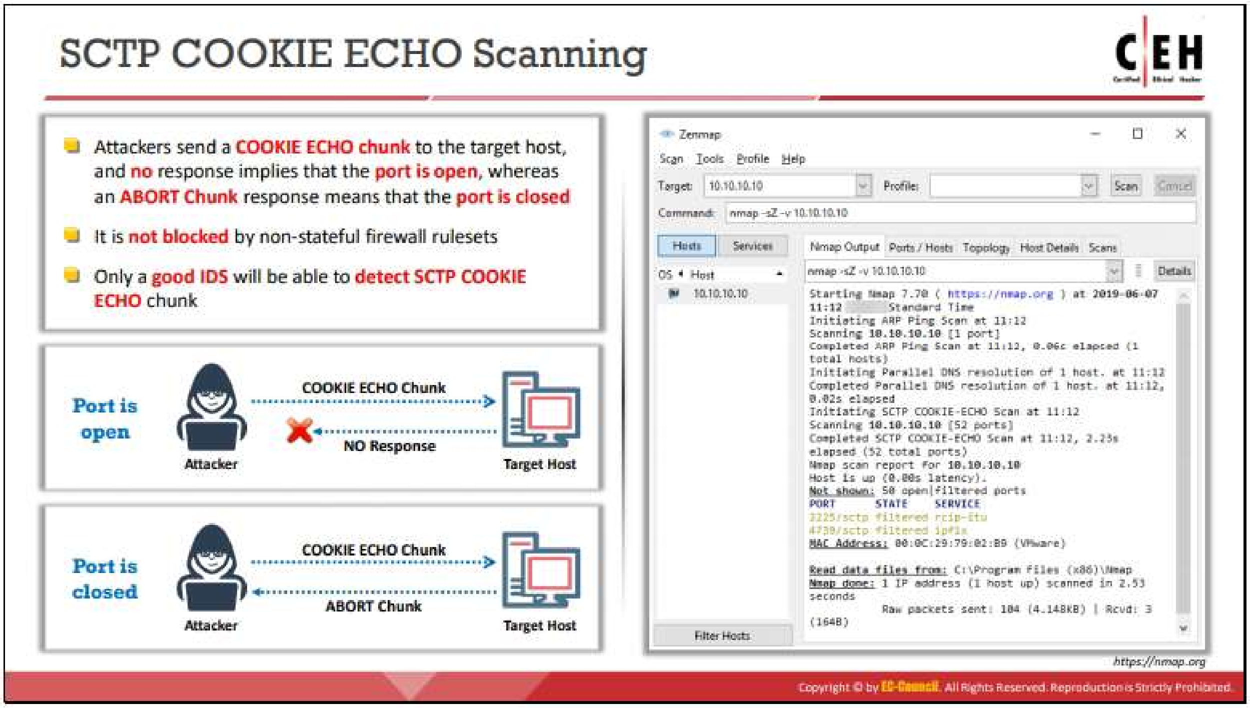


Figure 3.69: SCTP INIT scan in Zenmap



SCTP COOKIE ECHO Scanning

SCTP COOKIE ECHO scan is a more advanced type of scan. In this type of scan, attackers send the COOKIE ECHO chunk to the target, and if the target port is open, it will silently drop the packets onto the port and you will not receive any response from the target. If the target sends back the ABORT chunk response, then the port is considered as a closed port. The COOKIE ECHO chunk is not blocked by non-stateful firewall rule sets as in the INIT scan. Only an advanced IDS can detect the SCTP COOKIE ECHO scan. In Zenmap, the **`-sZ`** option is used to perform the SCTP COOKIE ECHO scan.

Figure 3.70: SCTP COOKIE ECHO scan result when port is open

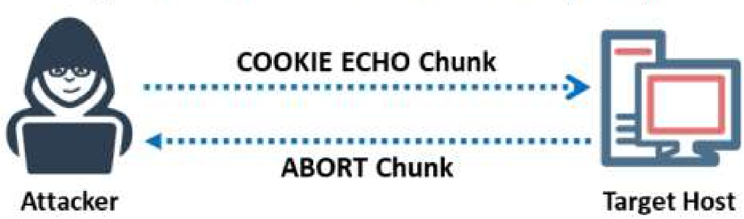


Figure 3.71: SCTP COOKIE ECHO scan result when port is closed

Advantages:

- The port scan is not as conspicuous as the INIT scan.

Disadvantages:

- SCTP COOKIE ECHO scan cannot differentiate clearly between open and filtered ports, and it shows the output as open|filtered in both cases.

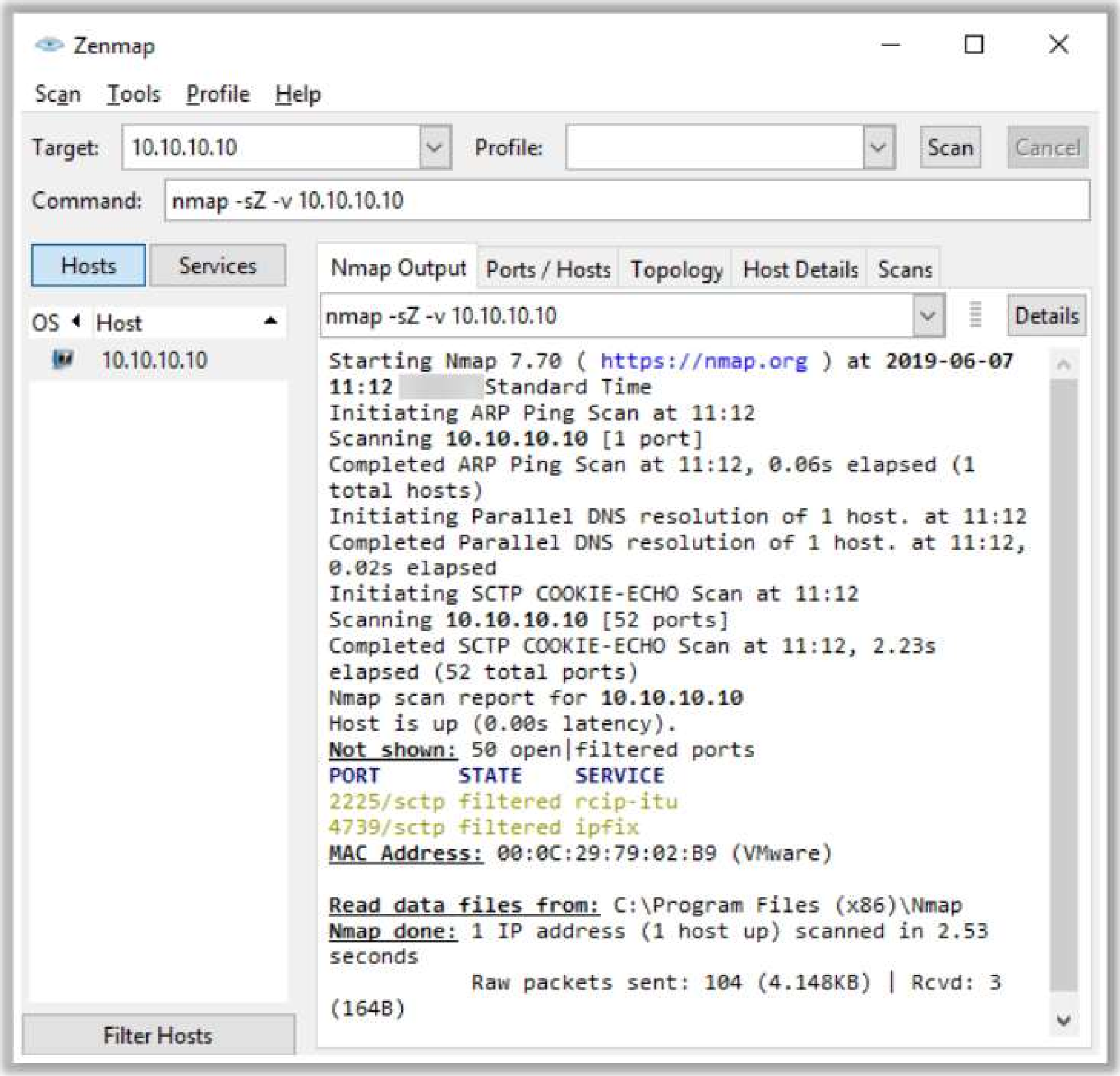
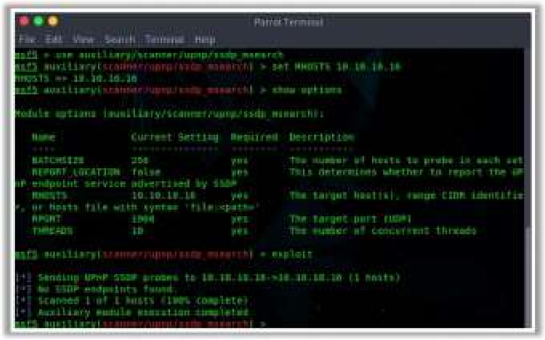


Figure 3.72: SCTP COOKIE-ECHO scan in Zenmap

SSDP and List Scanning

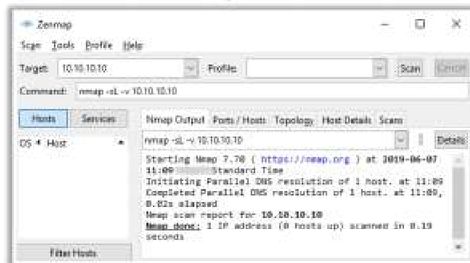
SSDP Scanning

- 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

SSDP and List Scanning

SSDP Scanning

Simple Service Discovery Protocol (SSDP) is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses. The SSDP service controls communication for the Universal Plug and Play (UPnP) feature. It generally works when the machine is not firewalled; however, it can sometimes work through a firewall. The SSDP service will respond to a query sent over IPv4 or IPv6 broadcast addresses. This response includes information about the UPnP feature associated with it. The attacker uses SSDP scanning to detect UPnP vulnerabilities that may allow him/her to launch buffer overflow or DoS attacks.

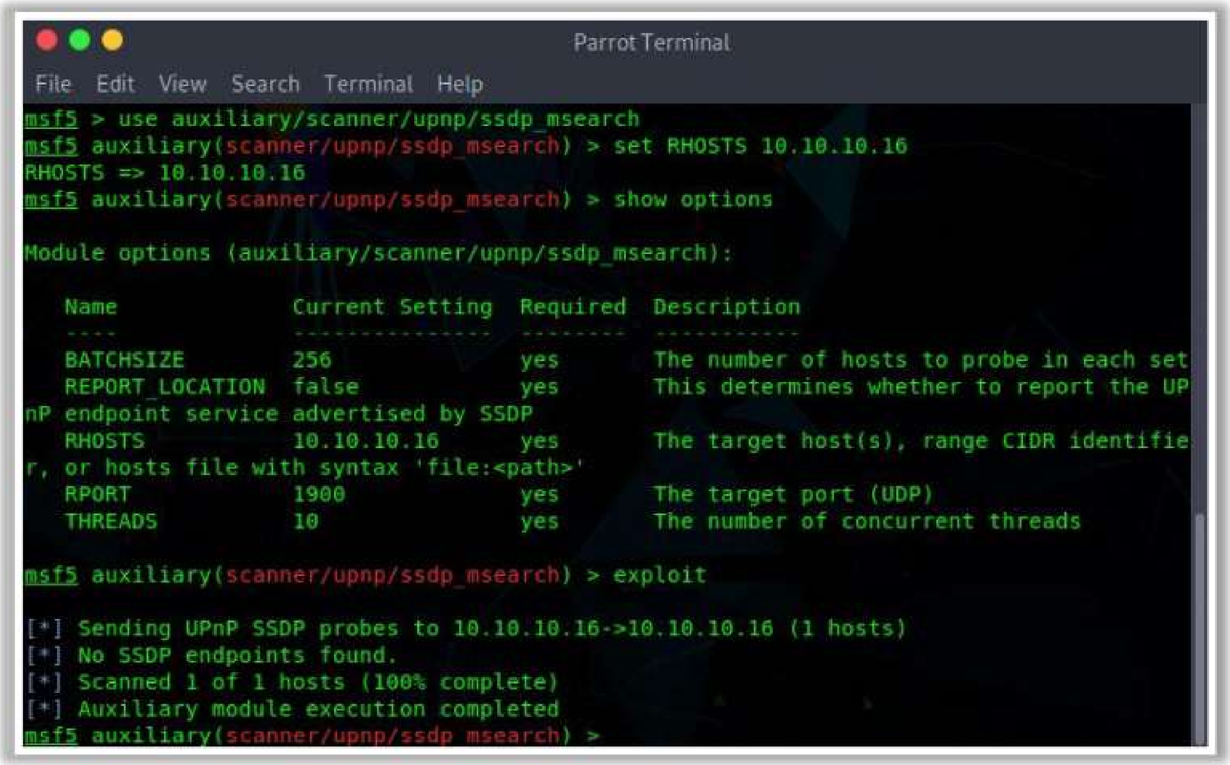


Figure 3.73: UPnP SSDP M-SEARCH in Parrot Security

The attacker may use the UPnP SSDP M-SEARCH information discovery tool to check whether the machine is vulnerable to UPnP exploits. The UPnP SSDP M-SEARCH information discovery tool gleans information from UPnP-enabled systems, as shown in the figure.

List Scanning

In a list scan, the discovery of the active network host is indirect. A list scan simply generates and prints a list of IPs/Names without actually pinging or scanning the hosts. As a result, the list scan shows all IP addresses as "not scanned" (0 hosts up). By default, a reverse DNS resolution is still carried out on each host by Nmap to learn their names. In Zenmap, the `-sL` option is used to perform a list scan.

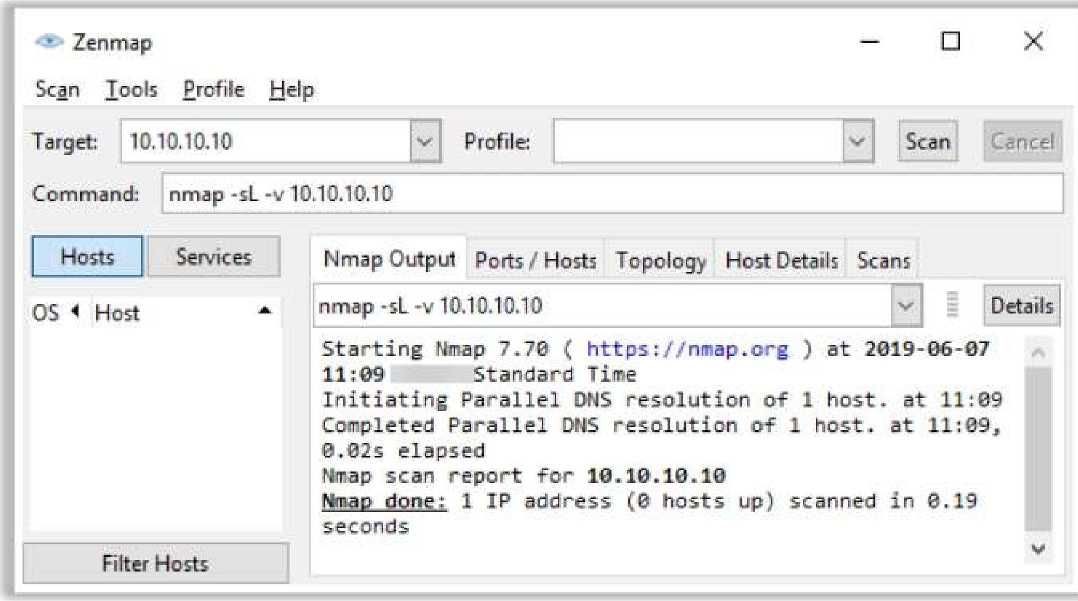
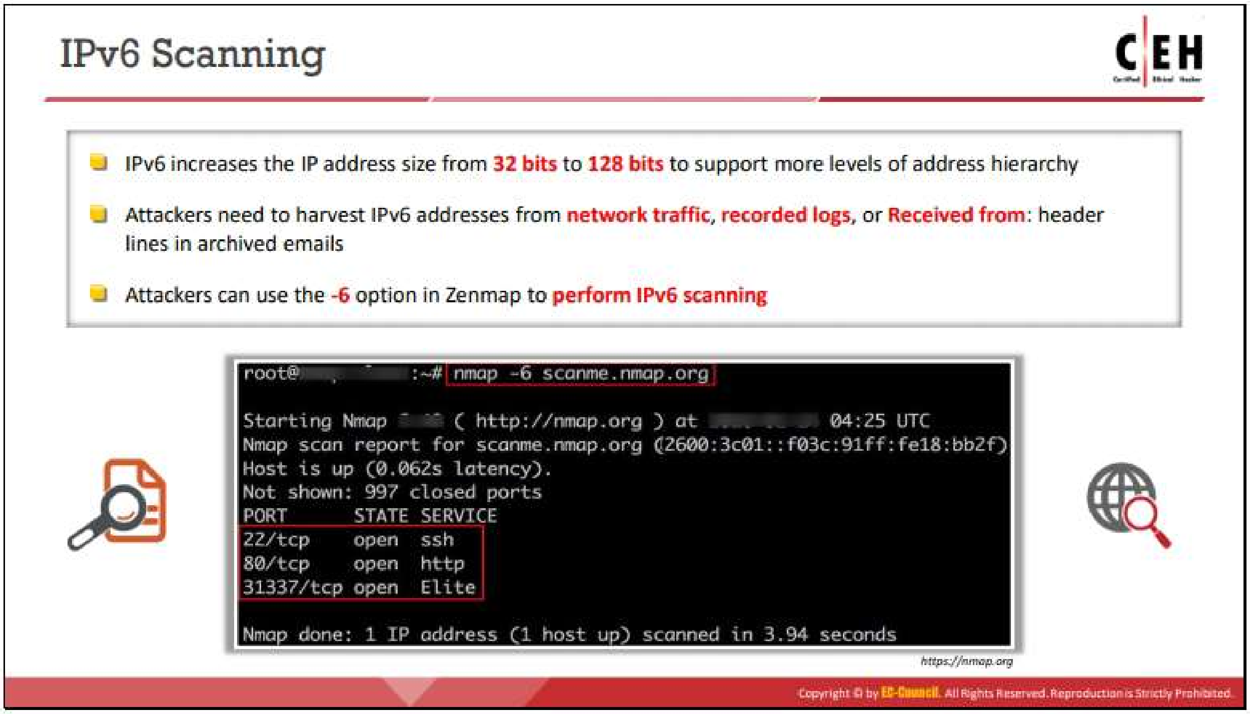


Figure 3.74: List scan using Zenmap

Advantages:

- A list scan can perform a good sanity check.
- The list scan detects incorrectly defined IP addresses in the command line or in an option file. It primarily repairs the detected errors to run any "active" scan.



IPv6 Scanning

IPv6 increases the size of the IP address space from 32 bits to 128 bits to support higher levels of the addressing hierarchy. Traditional network scanning techniques are computationally less feasible because of the larger search space (64 bits of host address space, or 2^{64} addresses) provided by IPv6 in a subnet. Scanning the IPv6 network is more difficult and complex compared to IPv4. Additionally, a number of scanning tools do not support ping sweeps on IPv6 networks. Attackers need to harvest IPv6 addresses from network traffic, recorded logs, or "Received from" and other header lines in archived email or Usenet news messages to identify IPv6 addresses for subsequent port scanning. However, scanning an IPv6 network provides a large number of hosts in a subnet; if an attacker can compromise one subnet host, he/she can probe the "all hosts" link local multicast address if the hosts numbers are sequential or use any regular scheme. An attacker needs to analyze 2^{64} addresses to verify if a particular open service is running on a host in that subnet. At a conservative rate of one probe per second, such a scan would take about 5 billion years to complete. Attackers can use Nmap to perform IPv6 scanning. In Zenmap, the **-6** option is used to perform the IPv6 scan.

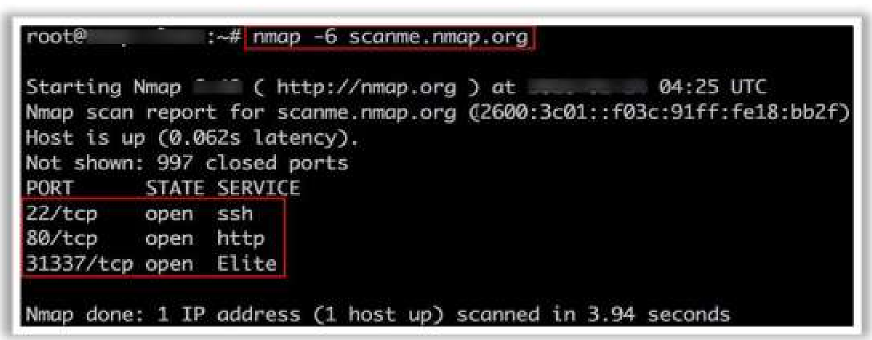
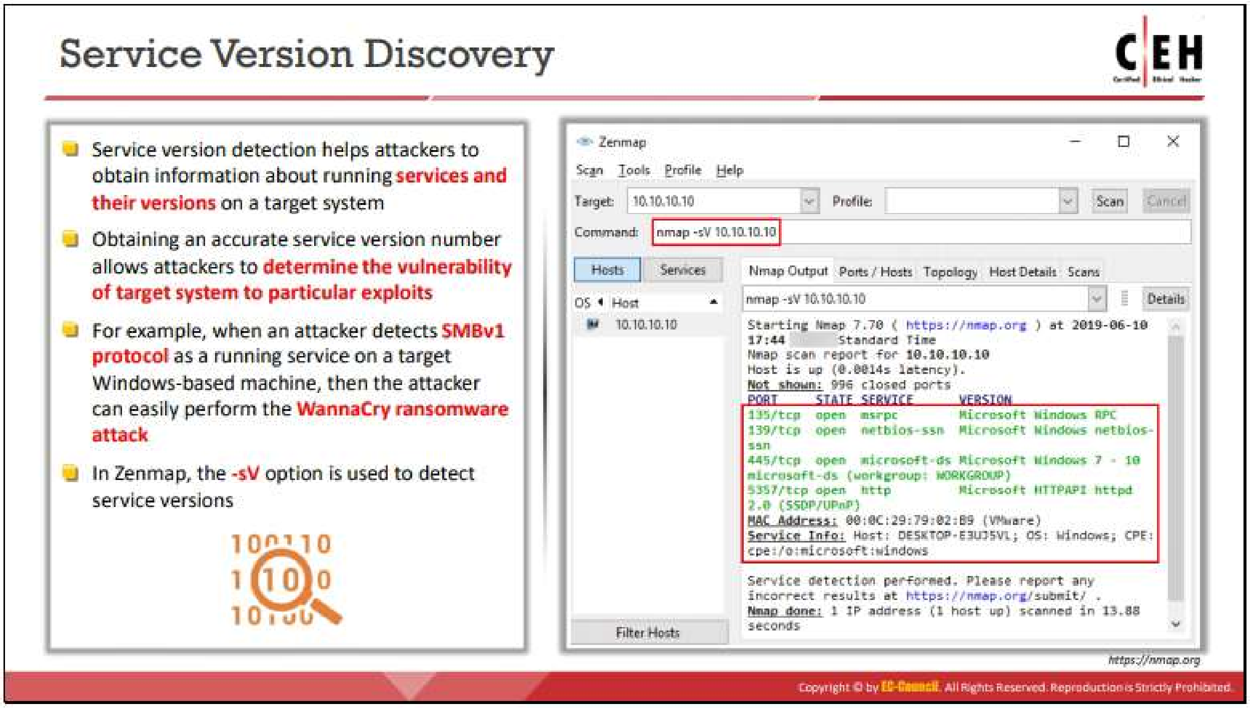


Figure 3.75: IPv6 Scan in Zenmap



Service Version Discovery

Every port is assigned a specific service, and every service has its own version. Some versions of the protocols are insecure, and they can allow attackers to compromise the machine by exploiting this vulnerability. Service version detection helps attackers to obtain information about the running services and their versions on a target system. By obtaining accurate service version numbers, an attacker can determine which exploits the target system is vulnerable to. For example, when the attacker detects the SMBv1 protocol as a running service on the target Windows machine, then he/she can easily perform a WannaCry ransomware attack with the help of the eternalblue and doublepulsar backdoor combination in Metasploit.

The version detection technique is nothing but examination of the TCP and UDP ports. The probes from the Nmap `service-probes` database are used for querying various services and matching expressions for recognizing and parsing responses. In Zenmap, the **-sV** option is used to detect service versions.

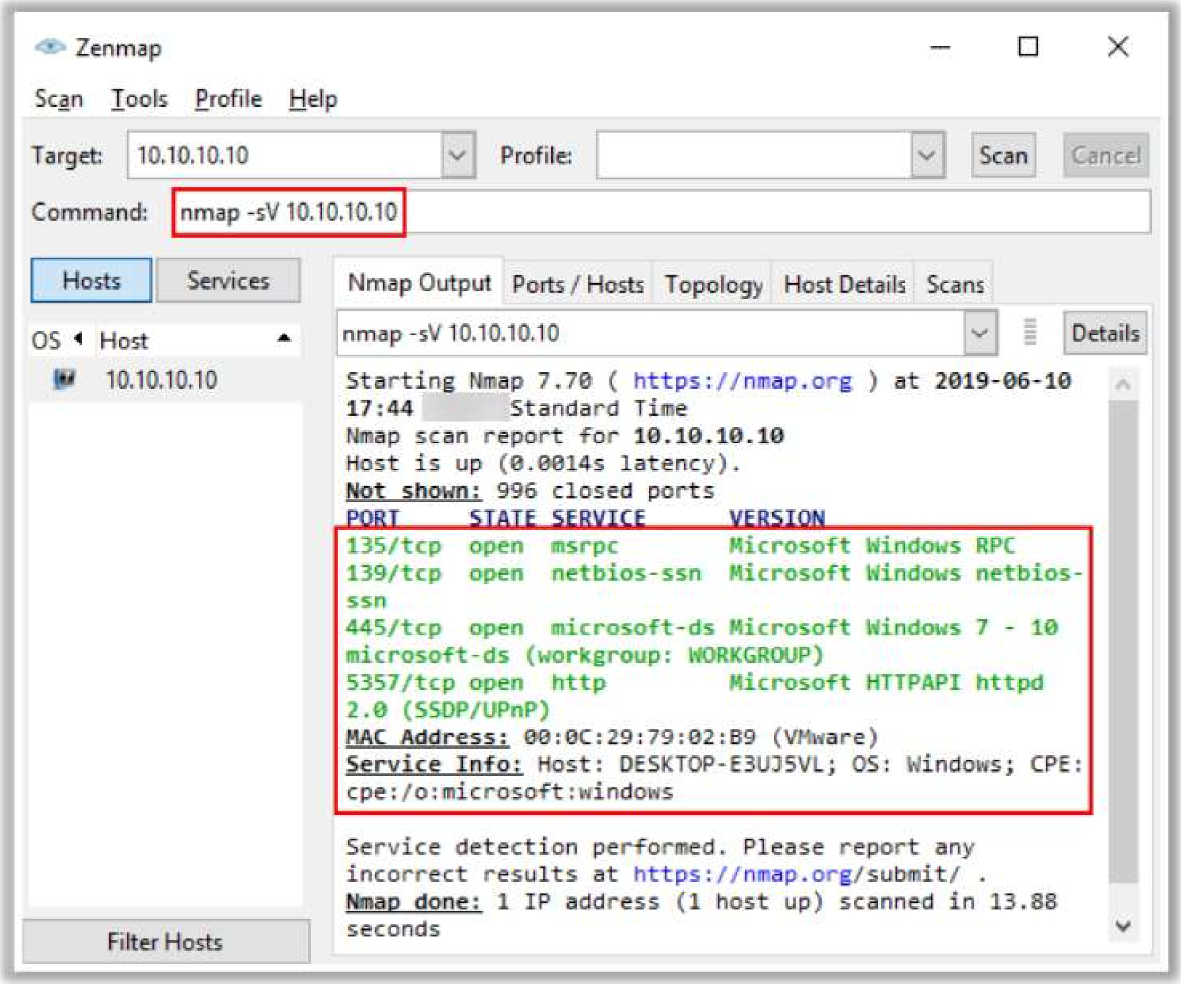
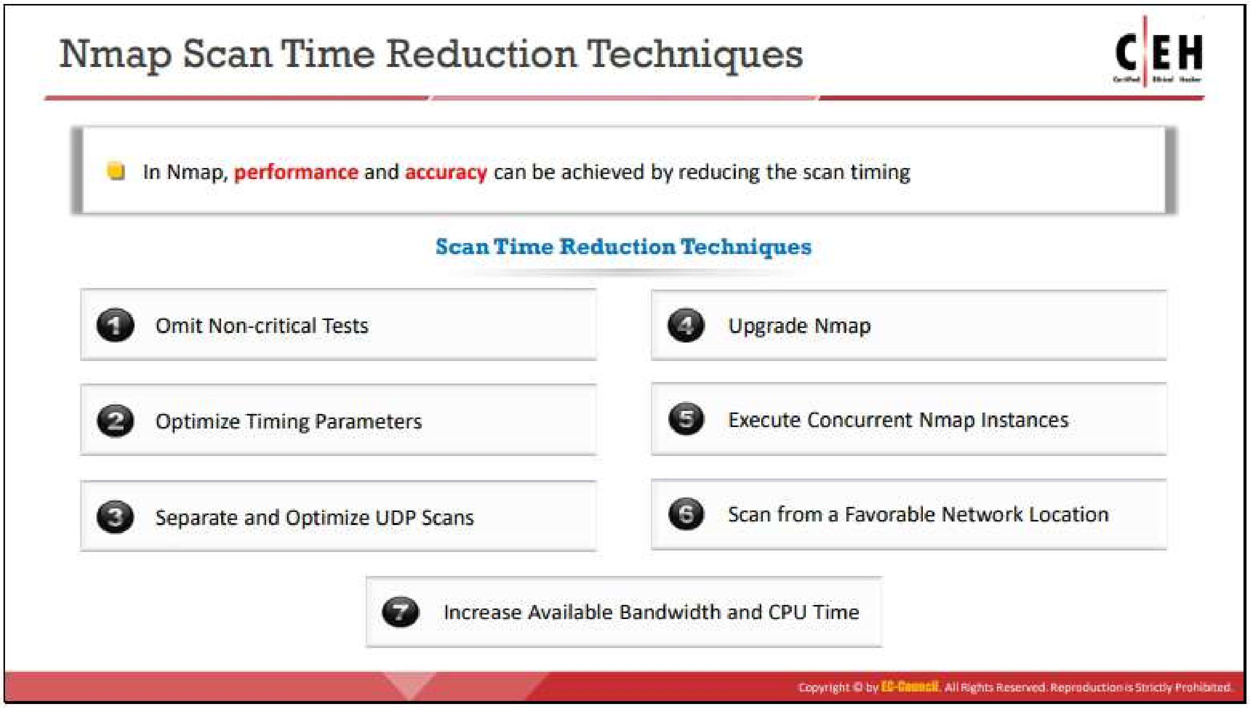


Figure 3.76: Service version discovery in Zenmap



Nmap Scan Time Reduction Techniques

In Nmap, performance and accuracy take high priority, and this only be achieved only by reducing the long scan time. The important techniques for reducing the scan time are as follows:

- **Omit Non-critical Tests**

  While performing the Nmap scan, the time complexity can be reduced by the following methods:

  - Avoiding an intense scan if only a minimal amount of information is required.
  - The number of ports scanned can be limited using specific commands.
  - The port scan (`-sn`) can be skipped if and only if one has to check whether the hosts are online or not.
  - Advanced scan types (`-sC`, `-sV`, `-O`, `--traceroute`, and `-A`) can be avoided.
  - The DNS resolution should be turned on only when it is necessary.

- **Optimize Timing Parameters**

  To control the scan activity, Nmap provides the `-T` option for scanning ranging from high-level to low-level timing aggressiveness. This can be extremely useful for scanning highly filtered networks.

- **Separate and Optimize UDP Scans**

  As many vulnerable services use the UDP protocol, scanning the UDP protocol is vital, and it should be scanned separately, as TCP scans have different performance requirements and timing characteristics. Moreover, the UDP scan is more affected by the ICMP error rate-limiting compared to the TCP scan.

- **Upgrade Nmap**

  It is always advisable to use the upgraded version of Nmap as it contains many bug fixes, important algorithmic enhancements, and high-performance features such as local network ARP scanning.

- **Execute Concurrent Nmap Instances**

  Running Nmap against the whole network usually makes the system slower and less efficient. Nmap supports parallelization and it can also be customized according to specific needs. It becomes very efficient by getting an idea of the network reliability while scanning a larger group. The overall speed of the scan can be improved by dividing it into many groups and running them simultaneously.

- **Scan from a Favorable Network Location**

  It is always advisable to run Nmap from the host's local network to the target while in the internal network, as it offers defense-in-depth security. External scanning is obligatory when performing firewall testing or when the network should be monitored from the external attacker's viewpoint.

- **Increase Available Bandwidth and CPU Time**

  By increasing the available bandwidth or CPU power, the Nmap scan time can be reduced. This can be done by installing a new data line or stopping any running applications. Nmap is controlled by its own congestion control algorithms, so that network flooding can be prevented. This improves its accuracy. The Nmap bandwidth usage can be tested by running it in the verbose mode `-v`.

Port Scanning Countermeasures

CEH

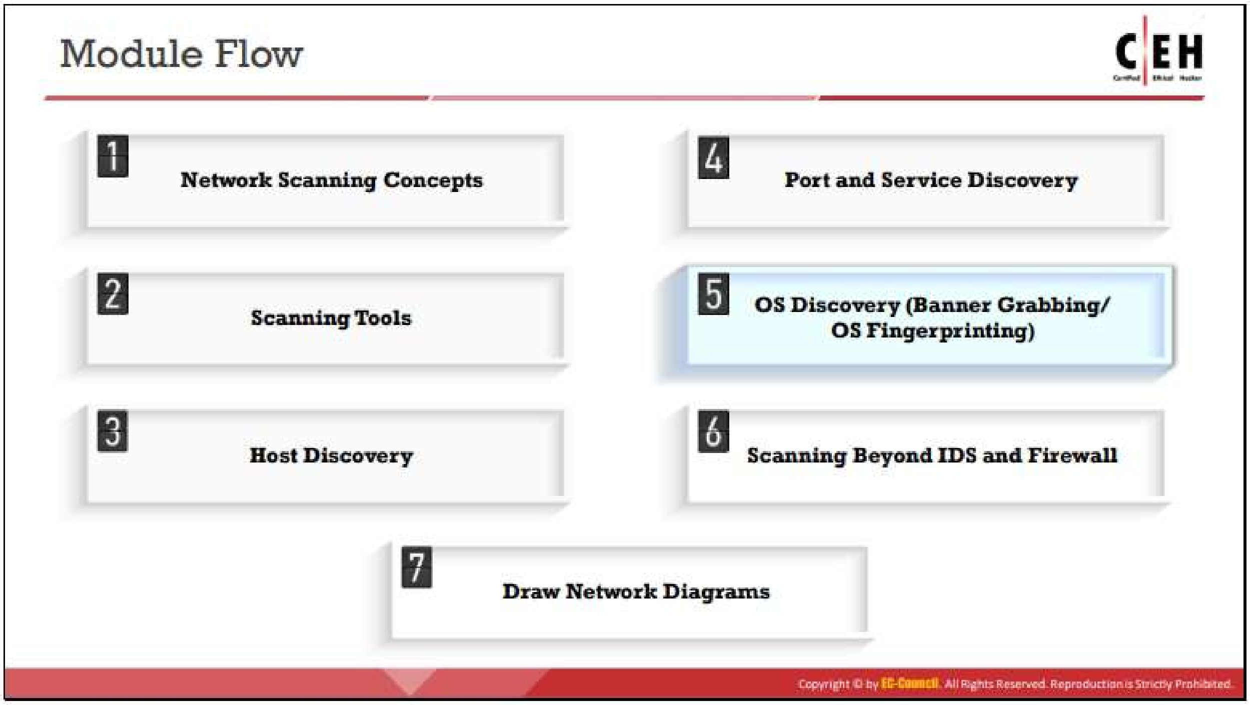
1. Configure firewall and IDS rules to detect and block probes
2. Run port scanning tools against hosts on the network to determine whether the firewall properly detects port scanning activity
3. Ensure that the mechanisms used for routing by routers and for filtering by firewalls cannot be bypassed 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

Port Scanning Countermeasures

As discussed previously, port scanning provides a large amount of useful information to the attacker, such as IP addresses, host names, open ports, and services running on ports. Open ports specifically offer an easy means for the attacker to break into the network. However, there is no cause for concern, provided that you secure your system or network against port scanning by adopting the following countermeasures:

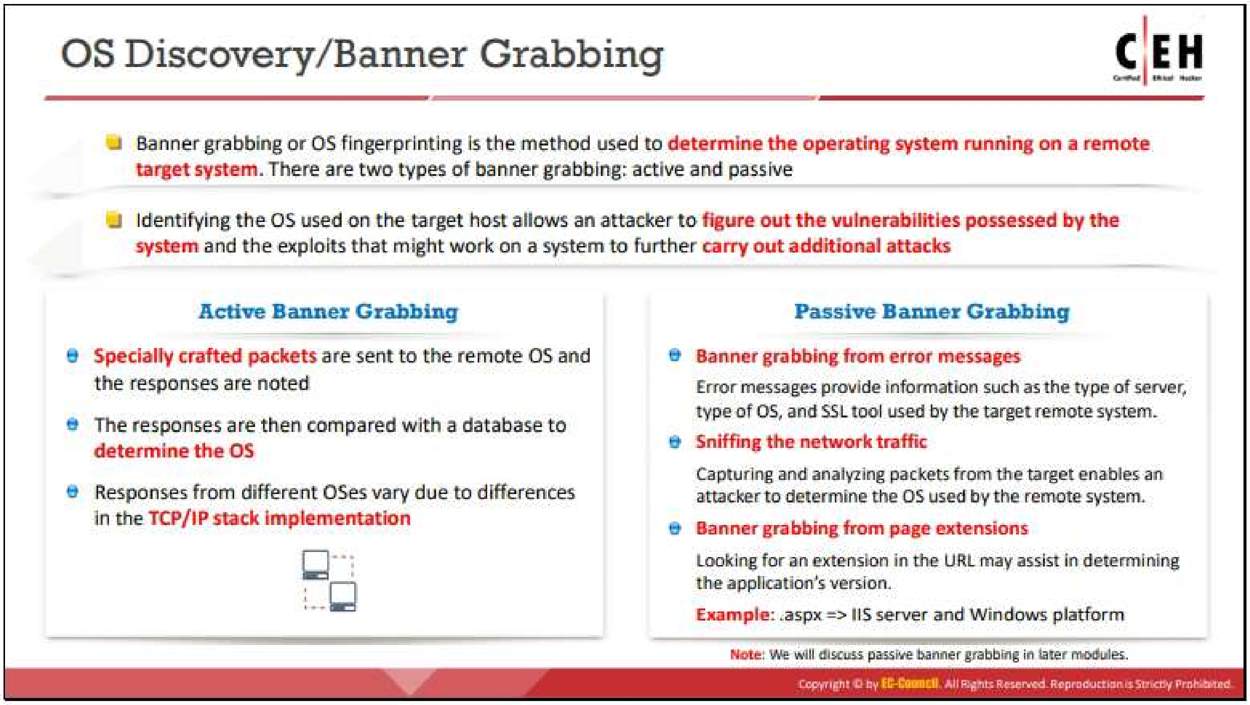
- Configure firewall and IDS rules to detect and block probes.
- The firewall should be capable of detecting probes sent by the attackers using port scanning tools. It should not allow traffic to pass through it after simply inspecting the TCP header. The firewall should be able to examine the data contained in each packet before allowing the traffic to pass through it.
- Run the port scanning tools against hosts on the network to determine whether the firewall accurately detects the port scanning activity.
- Some firewalls do a better job than others in terms of detecting stealth scans. For example, many firewalls have specific options to detect SYN scans, while others completely ignore FIN scans.
- Ensure that the router, IDS, and firewall firmware are updated with their latest releases/versions.
- Configure commercial firewalls to protect your network against fast port scans and SYN floods. You can run tools such as port entry to detect and stop port scan attempts on Linux/UNIX systems.

- Hackers use tools such as Nmap and perform OS detection to sniff the details of a remote OS. Thus, it is important to employ intrusion detection systems in such cases. Snort (*https://www.snort.org*) is an intrusion detection and prevention technology that is very useful, mainly because signatures are frequently available from the public authors.
- Keep as few ports open as possible and filter the rest, as the intruder will try to enter through any open port. Use a custom rule set to lock down the network, block unwanted ports at the firewall, and filter the following ports: 135–159, 256–258, 389, 445, 1080, 1745, and 3268.
- Block unwanted services running on the ports and update the service versions.
- Ensure that the versions of services running on the ports are non-vulnerable.
- Block inbound ICMP message types and all outbound ICMP type-3 unreachable messages at border routers arranged in front of a company's main firewall.
- Attackers try to perform source routing and send packets to the targets (which may not be reachable via the Internet) using an intermediate host that can interact with the target. Hence, it is necessary to ensure that your firewall and router can block such source-routing techniques.
- Ensure that the mechanism used for routing and filtering at the routers and firewalls, respectively, cannot be bypassed using a particular source port or source-routing methods.
- Test your IP address space using TCP and UDP port scans as well as ICMP probes to determine the network configuration and accessible ports.
- Ensure that the anti-scanning and anti-spoofing rules are configured.
- If a commercial firewall is in use, then ensure that:
    - It is patched with the latest updates
    - It has correctly defined antispoofing rules
    - Its fastmode services are unusable in Check Point Firewall-1 environments



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 a method used to determine the OS that is running on a remote target system. It is an important scanning method, as the attacker will have a higher probability of success if the OS of the target system is known (many vulnerabilities are OS-specific). The attacker can then formulate an attack strategy based on the OS of the target system.

There are two methods for banner grabbing: spotting the banner while trying to connect to a service, such as an FTP site, and downloading the binary file/bin/ls to check the system architecture.

A more advanced fingerprinting technique depends on stack querying, which transfers the packets to the network host and evaluates them by the reply. The first stack-querying method designed with regard to the TCP mode of communication evaluates the response to connection requests.

The next method, known as initial sequence number (ISN) analysis, identifies the differences in random number generators found in the TCP stack.

ICMP response analysis is another method used to fingerprint an OS. It consists of sending ICMP messages to a remote host and evaluating the reply.

Two types of banner grabbing techniques are described below:

- **Active Banner Grabbing**

  Active banner grabbing applies the principle that an OS's IP stack has a unique way of responding to specially crafted TCP packets. This happens because of different interpretations that vendors apply while implementing the TCP/IP stack on a particular

OS. In active banner grabbing, the attacker sends a variety of malformed packets to the remote host, and the responses are compared with a database. Responses from different OS vary because of differences in TCP/IP stack implementation.

For instance, the scanning utility Nmap uses a series of nine tests to determine an OS fingerprint or banner grabbing. The tests listed below provide some insights into an active banner grabbing attack, as described at *www.packetwatch.net*:

- **Test 1:** A TCP packet with the SYN and ECN-Echo flags enabled is sent to an open TCP port.
- **Test 2:** A TCP packet with no flags enabled is sent to an open TCP port. This type of packet is a NULL packet.
- **Test 3:** A TCP packet with the URG, PSH, SYN, and FIN flags enabled is sent to an open TCP port.
- **Test 4:** A TCP packet with the ACK flag enabled is sent to an open TCP port.
- **Test 5:** A TCP packet with the SYN flag enabled is sent to a closed TCP port.
- **Test 6:** A TCP packet with the ACK flag enabled is sent to a closed TCP port.
- **Test 7:** A TCP packet with the URG, PSH, and FIN flags enabled is sent to a closed TCP port.
- **Test 8 PU (Port Unreachable):** A UDP packet is sent to a closed UDP port. The objective is to extract an "ICMP port unreachable" message from the target machine.
- **Test 9 TSeq (TCP Sequence ability test):** This test tries to determine the sequence generation patterns of the TCP initial sequence numbers (also known as TCP ISN sampling), the IP identification numbers (also known as IPID sampling), and the TCP timestamp numbers. It sends six TCP packets with the SYN flag enabled to an open TCP port.

The objective of these tests is to find patterns in the initial sequence of numbers that the TCP implementations chose while responding to a connection request. They can be categorized into groups, such as traditional 64K (many old UNIX boxes), random increments (newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others), or true random (Linux 2.0.*, OpenVMS, newer AIX, etc.). Windows boxes use a "time-dependent" model in which the ISN is incremented by a fixed amount for each occurrence.

- **Passive Banner Grabbing**

  Source: *https://www.symantec.com*

  Like active banner grabbing, passive banner grabbing also depends on the differential implementation of the stack and the various ways in which an OS responds to packets. However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host via sniffing to study telltale signs that can reveal an OS.

Passive banner grabbing includes:

- **Banner grabbing from error messages**: Error messages provide information, such as type of server, type of OS, and SSL tools 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 help in determining the application version. For example, .aspx => IIS server and Windows platform.

The four areas that typically determine the OS are given below:

- TTL (time to live) of the packets: What does the OS sets as the Time To Live on the outbound packet?
- Window Size: What is the Window size set by the OS?
- Whether the DF (Don't Fragment) bit is set: Does the OS set the DF bit?
- TOS (Type of Service): Does the OS set the TOS, and if so, what setting is it?

Passive fingerprinting is neither fully accurate nor limited to these four signatures. However, one can improve its accuracy by looking at several signatures and combining the information. The following is an analysis of a sniffed packet described by Lance Spitzner in his paper on passive fingerprinting (*https://www.symantec.com/connect/articles/passive-fingerprinting*):

```
04/20-21:41:48.129662 129.142.224.3:659 -> 172.16.1.107:604
TCP TTL:45 TOS:0x0 ID:56257
***F**A* Seq: 0x9DD90553
Ack: 0xE3C65D7 Win: 0x7D78
```

According to the four criteria, the following are identified:

- TTL: 45
- Window Size: 0x7D78 (or 32120 in decimal)
- DF: The DF bit is set
- TOS: 0x0

Compare this information with a database of signatures.

TTL: The TLL from the analysis is 45. The original packet went through 19 hops to get to the target, so it sets the original TTL to 64. Based on this TTL, it appears that the user sent the packet from a Linux or FreeBSD box (however, more system signatures need to be added to the database). This TTL confirms it by implementing a traceroute to the remote host. If the trace needs to be performed stealthily, the traceroute TTL (default 30 hops) can be set to one or two hops fewer than the remote host (-m option). Setting the traceroute in this manner reveals the path information (including the upstream provider) without actually contacting the remote host.

Window Size: In this step, the window sizes are compared. The window size is another effective tool for determining precisely what window size is used and how often it is changed. In the previous signature, the window size is set at 0x7D78, which is the default window size used by Linux. In addition, FreeBSD and Solaris tend to maintain the same window size throughout a session. However, Cisco routers and Microsoft Windows NT window sizes constantly change. The window size is more accurate when measured after the initial three-way handshake (due to TCP slow start).

DF bit: Most systems use the DF bit set; hence, this is of limited value. However, this makes it easier to identify a few systems that do not use the DF flag (such as SCO or OpenBSD).

TOS: TOS is also of limited value, as it seems to be more session-based than OS-based. In other words, it is not so much the OS as the protocol used that determines the TOS to a large extent.

Using the information obtained from the packet, specifically the TTL and the window size, one can compare the results with the database of signatures and determine the OS with some degree of confidence (in this case, Linux kernel 2.2.x).

Passive fingerprinting, like active fingerprinting, has some limitations. First, applications that build their own packets (e.g., Nmap, Hunt, Nemesis, etc.) will not use the same signatures as the OS. Second, it is relatively simple for a remote host to adjust the TTL, window size, DF, or TOS setting on the packets.

Passive fingerprinting has several other uses. For example, attackers can use stealthy fingerprinting to determine the OS of a potential target such as a web server. A user only needs to request a web page from the server and then analyze the sniffer traces. This bypasses the need for using an active tool that various IDS systems can detect. Passive fingerprinting also helps in identifying remote proxy firewalls. It may be possible to ID proxy firewalls from the signatures as discussed above, simply because proxy firewalls rebuild connections for clients. Similarly, passive fingerprinting can be used to identify rogue systems.

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

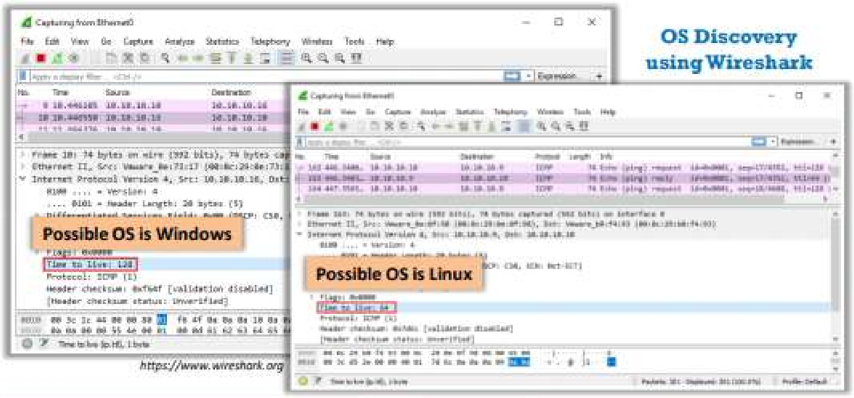
Why Banner Grabbing?

An attacker uses banner grabbing to identify the OS used on the target host and thus determine the system vulnerabilities and exploits that might work on that system to carry out further attacks.

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

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

How to Identify Target System OS

Identifying the target OS is one of the important tasks for an attacker to compromise the target network/machine. In a network, various standards are implemented to allow different OSs to communicate with each other. These standards govern the functioning of various protocols such as IP, TCP, UDP, etc. By analyzing certain parameters/fields in these protocols, one can reveal the details of the OS. Parameters such as Time to Live (TTL) and TCP window size in the IP header of the first packet in a TCP session help identify the OS running on the target machine. The TTL field determines the maximum time that a packet can remain in a network, and the TCP window size determines the length of the packet reported. These values vary among OSs, as described in the following table:

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

Table 3.3: TTL and TCP Window size values for OS

Attackers can use various tools to perform OS discovery on the target machine, including Wireshark, Nmap, Unicornscan, and Nmap Script Engine. Attackers can also adopt the IPv6 fingerprinting method to grab the target OS details.

OS Discovery using Wireshark

Source: *https://www.wireshark.org*

To identify the target OS, sniff/capture the response generated from the target machine to the request-originated machine using packet-sniffing tools such as Wireshark, etc., and observe the TTL and TCP window size fields in the first captured TCP packet. By comparing these values with those in the above table, you can determine the target OS that has generated the response.

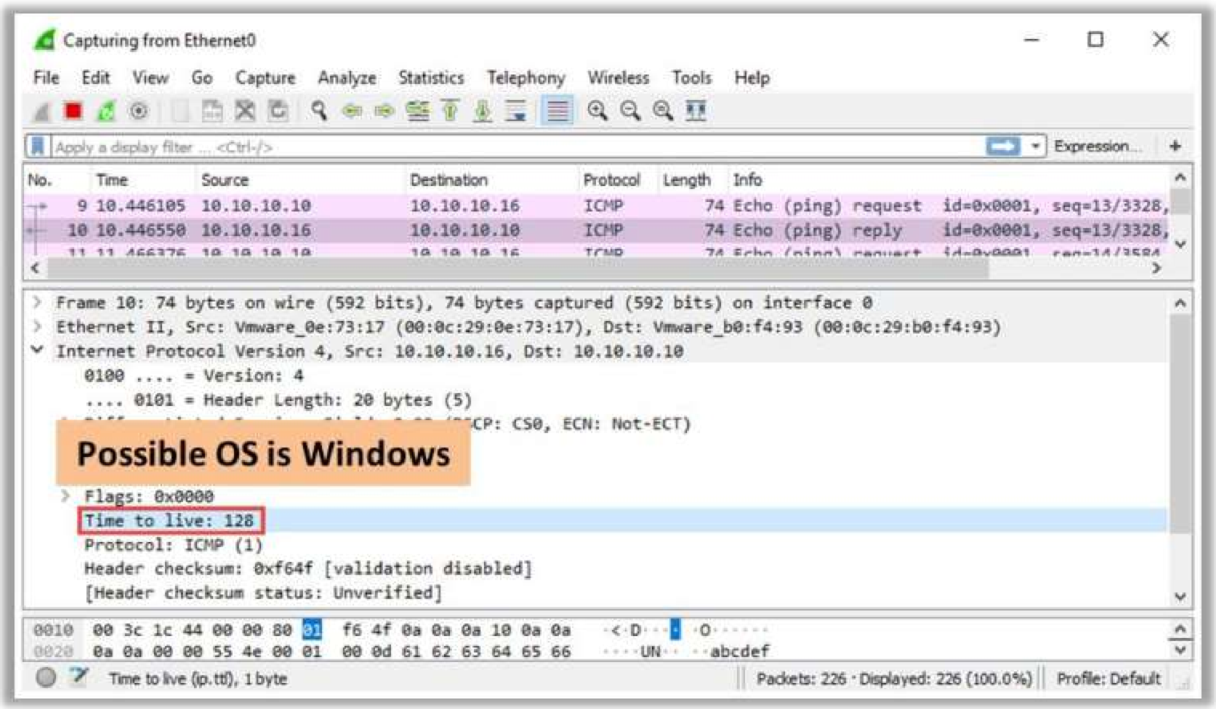


Figure 3.77: Wireshark screenshot showing TTL value (Possible OS is Windows)

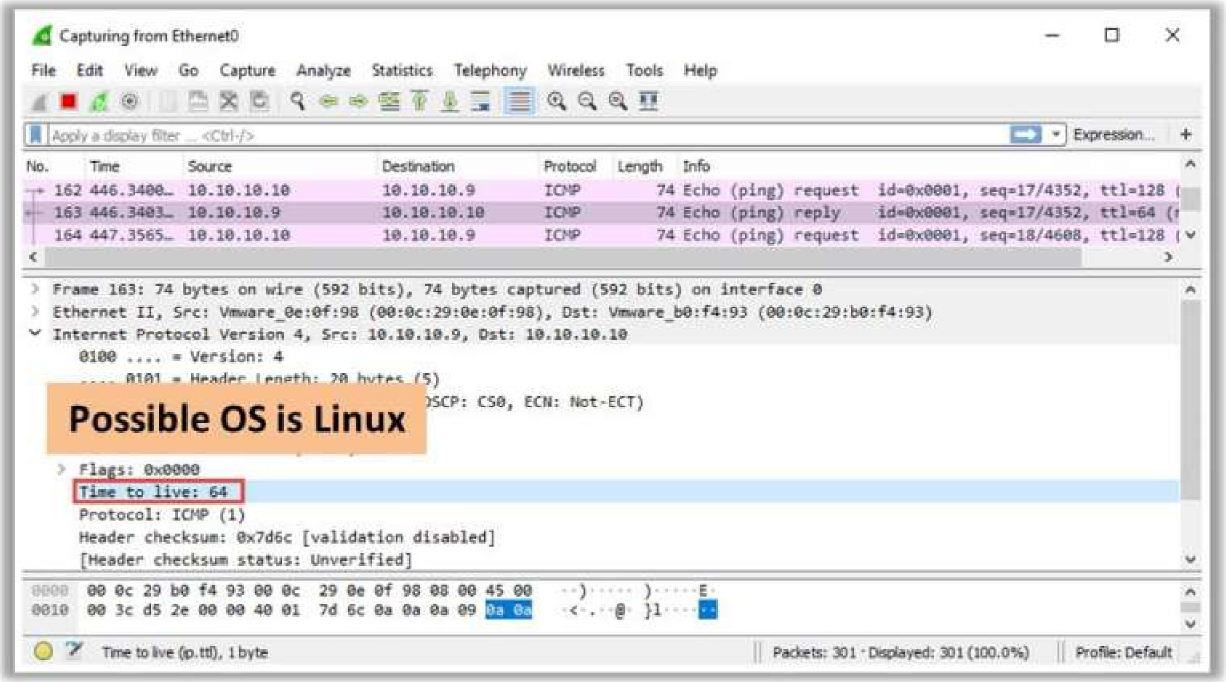
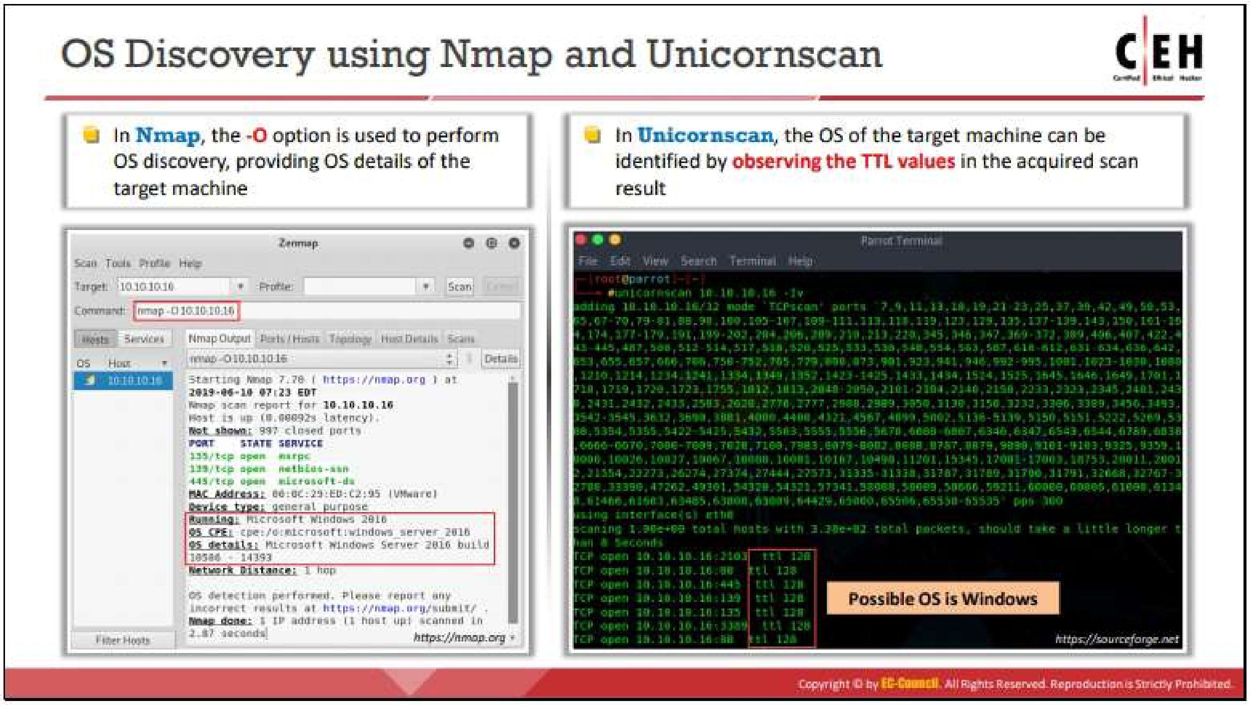


Figure 3.78: Wireshark screenshot showing TTL value (Possible OS is Linux)



OS Discovery using Nmap and Unicornscan

OS Discovery using Nmap

Source: *https://nmap.org*

To exploit the target, it is highly essential to identify the OS running on the target machine. Attackers can employ various tools to acquire the OS details of the target. Nmap is one of the effective tools for performing OS discovery activities. In Zenmap, the -O option is used to perform OS discovery, which displays the OS details of the target machine.

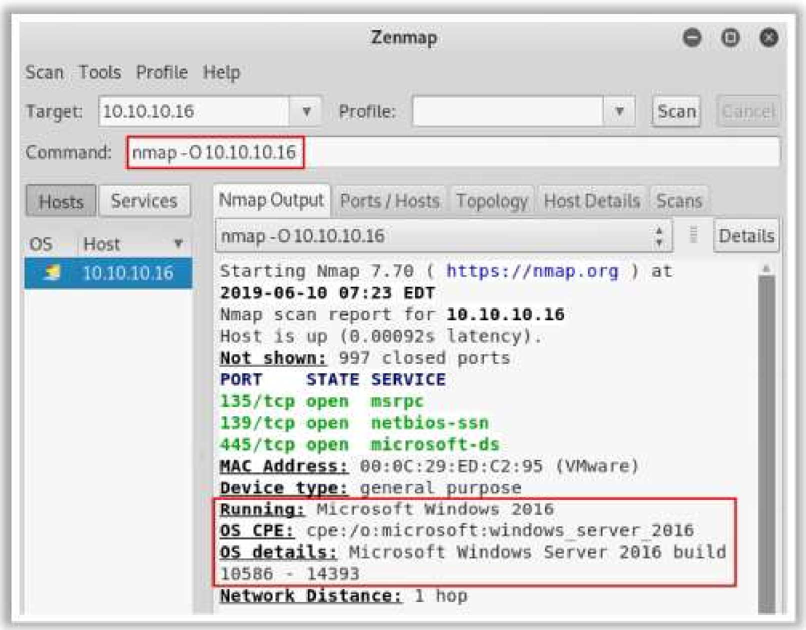


Figure 3.79: OS Discovery using Zenmap

OS Discovery using Unicornscan

Source: *https://sourceforge.net*

In Unicornscan, the OS of the target machine can be identified by observing the TTL values in the acquired scan result. To perform Unicornscan, the syntax **`#unicornscan <target IP address>`** is used. As shown in the screenshot, the **`ttl`** value acquired after the scan is **`128`**; hence, the OS is possibly Microsoft Windows (Windows 7/8/8.1/10 or Windows Server 2008/12/16).

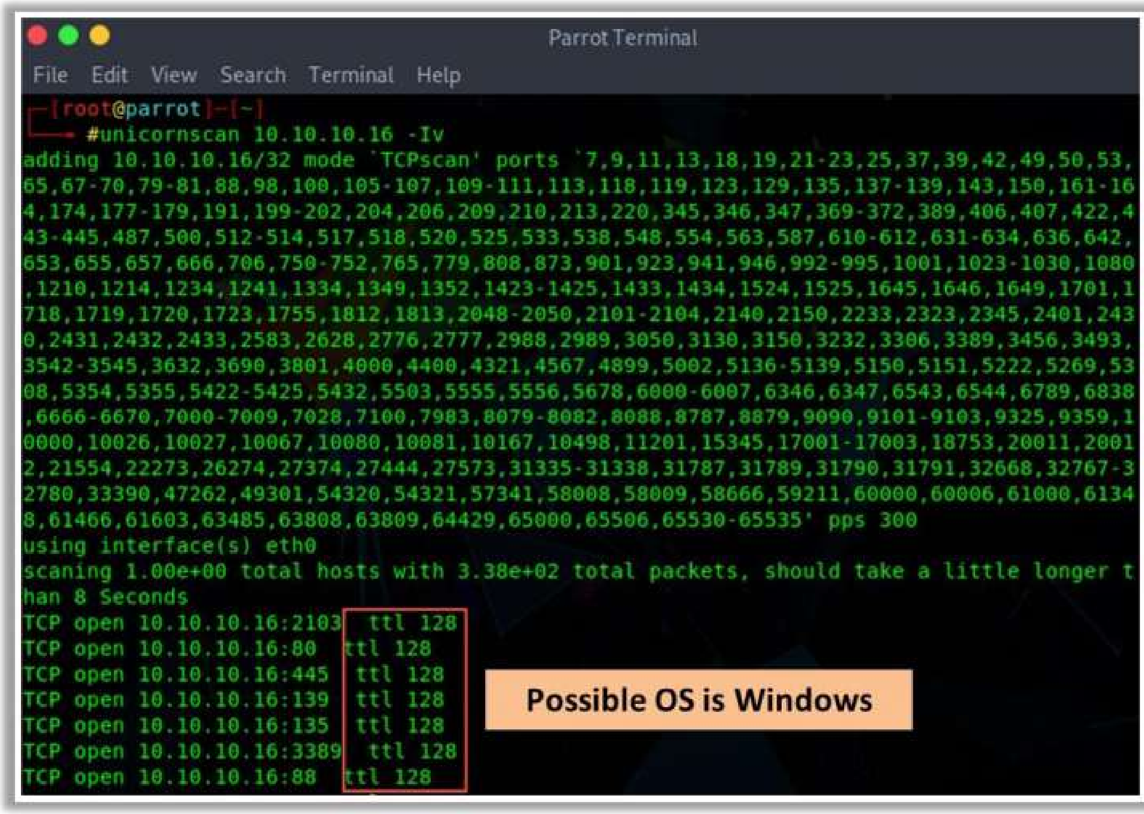
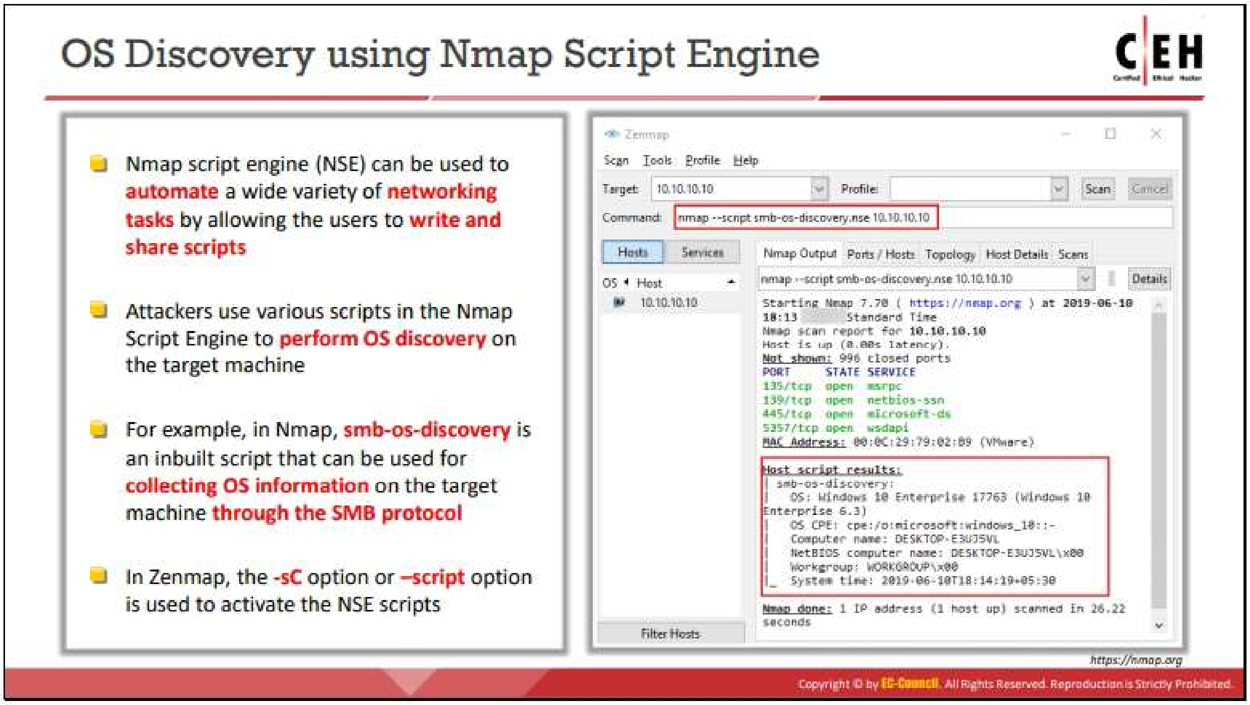


Figure 3.80: OS Discovery using Unicornscan



OS Discovery using Nmap Script Engine

Source: *https://nmap.org*

Nmap Scripting Engine (NSE) in Nmap can be used to automate a wide variety of networking tasks by allowing users to write and share scripts. These scripts can be executed parallelly with the same efficiency and speed as Nmap. Attackers can also use various scripts in the Nmap Script Engine for performing OS discovery on the target machine. For example, in Nmap, **`smb-os-discovery`** is an inbuilt script used for collecting OS information on the target machine through the SMB protocol.

In Zenmap, NSE can be generally activated using the **`-sC`** option. If the custom scripts are to be specified, then attackers can use the **`--script`** option. The NSE results will be displayed with both the Nmap normal and XML outputs.

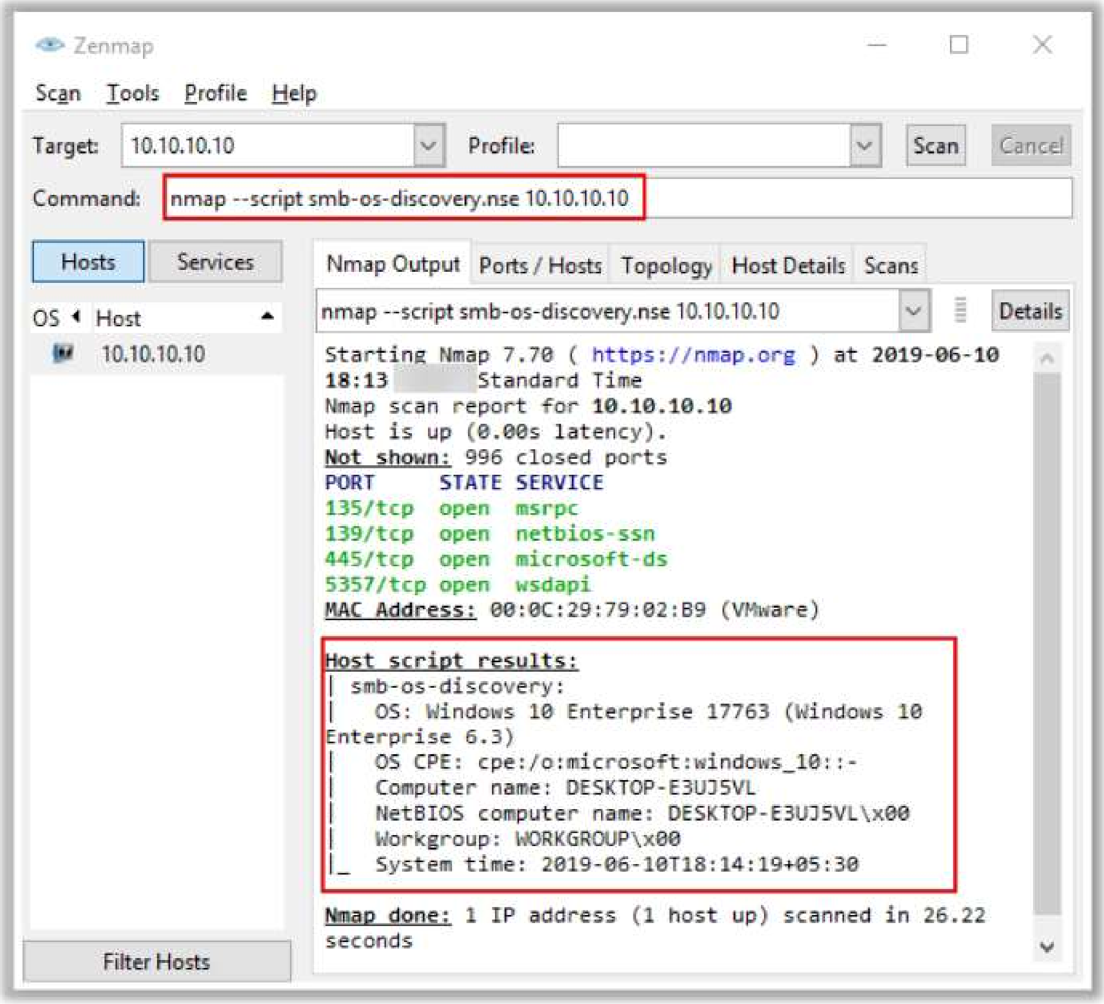
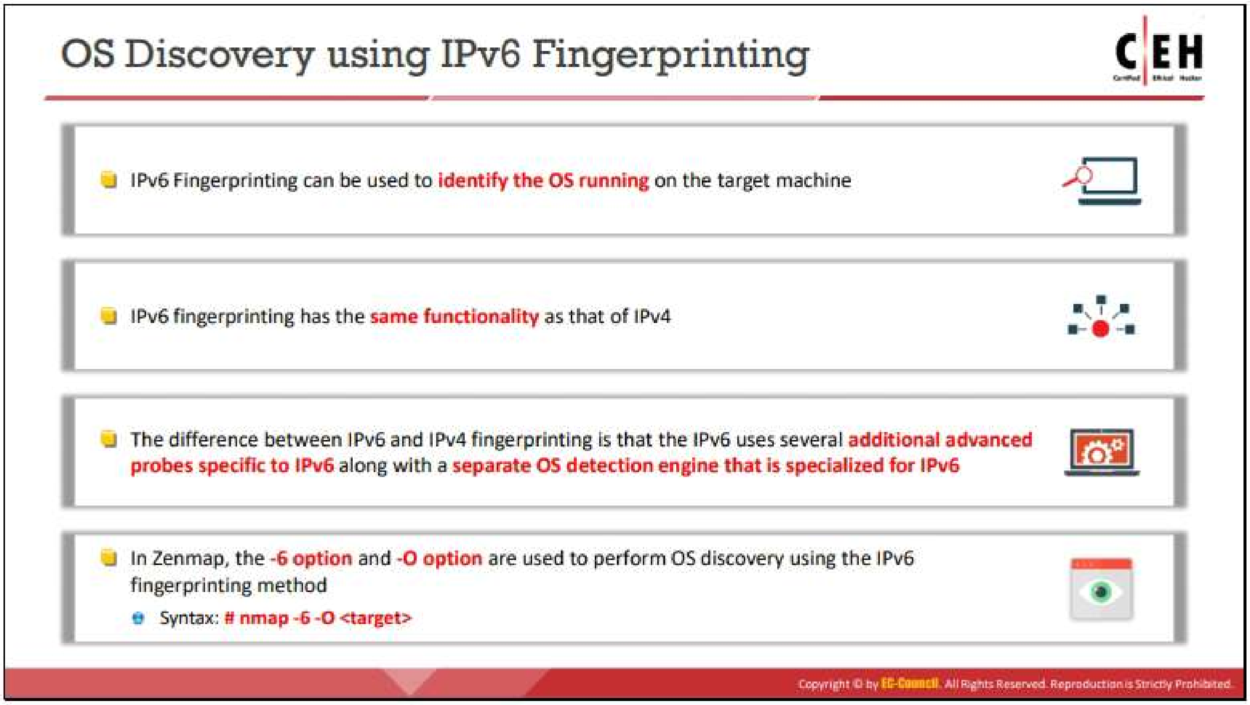


Figure 3.81: OS Discovery using Nmap Script Engine



OS Discovery using IPv6 Fingerprinting

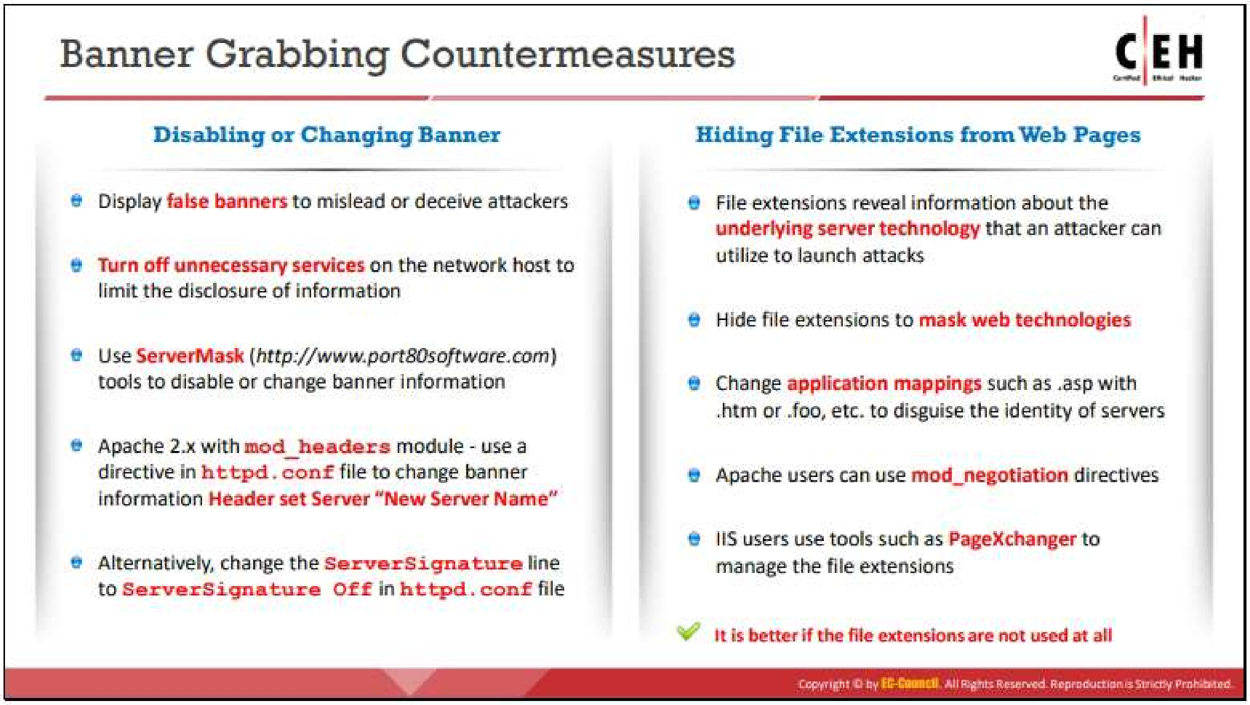
Source: *https://nmap.org*

IPv6 Fingerprinting is another technique used to identify the OS running on the target machine. It has the same functionality as IPv4, such as sending probes, waiting and collecting the responses, and matching them with the database of fingerprints. The difference between IPv6 and IPv4 fingerprinting is that IPv6 uses several additional advanced IPv6-specific probes along with a separate IPv6-specifc OS detection engine. Nmap sends nearly 18 probes in the following order to identify the target OS using the IPv6 fingerprinting method.

- Sequence generation (S1–S6)
- ICMPv6 echo (IE1)
- ICMPv6 echo (IE2)
- Node Information Query (NI)
- Neighbor Solicitation (NS)
- UDP (U1)
- TCP explicit congestion notification (TECN)
- TCP (T2–T7)

In Zenmap, the `-6` option along with `-O` option is used to perform OS discovery using the IPv6 fingerprinting method.

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



Banner Grabbing Countermeasures

- **Disabling or Changing Banner**

  Whenever a port is open, it implies that a service/banner is running on it. When attackers connect to the open port using banner grabbing techniques, the system presents a banner containing sensitive information such as OS, server type, and version. Using the information gathered, the attacker identifies specific vulnerabilities to exploit and then launches attacks. The countermeasures against banner grabbing attacks are as follows:

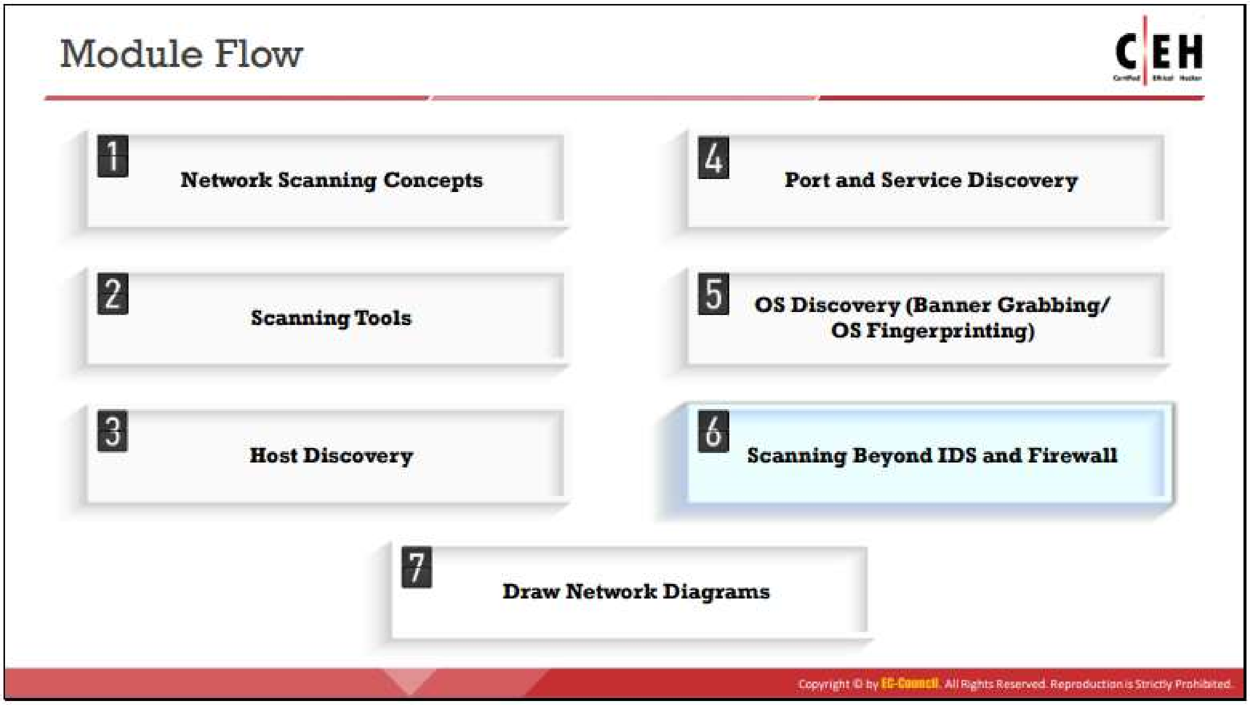
  - Display false banners to mislead or deceive attackers.
  - Turn off unnecessary services on the network host to limit information disclosure.
  - Use ServerMask (*https://www.port80software.com*) tools to disable or change banner information.
  - ServerMask removes unnecessary HTTP header and response data and camouflages the server by providing false signatures. It also provides you with the option of eliminating file extensions such as `.asp` or `.aspx`, and it clearly indicates that a site is running on a Microsoft server.
  - Apache 2.x with `mod_headers` module: use a directive in the `httpd.conf` file to change the banner information header and set the server as "`New Server Name`".
  - Alternatively, change the `ServerSignature` line to `ServerSignatureOff` in the `httpd.conf` file.
  - The details of the vendor and version in the banners should be disabled.

- **Hiding File Extensions from Web Pages**

  File extensions reveal information about the underlying server technology that an attacker can use to launch attacks. The countermeasures against such banner grabbing attacks are as follows:

  - Hide file extensions to mask the web technology.
  - Replace application mappings such as .asp with .htm or .foo, etc., to disguise the identity of the servers.
  - Apache users can use `mod_negotiation` directives.
  - IIS users can use tools such as PageXchanger to manage the file extensions.

Note: It would be 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:

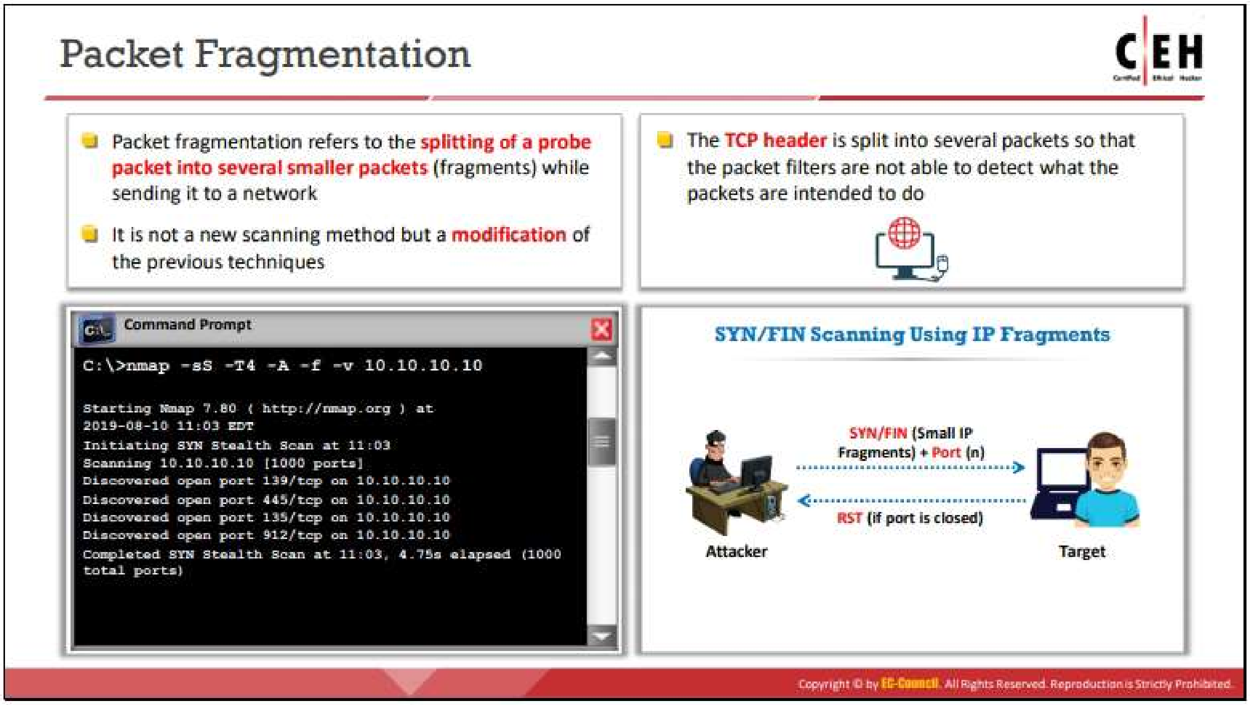
1. **Packet Fragmentation**
2. **Source Routing**
3. **Source Port Manipulation**
4. **IP Address Decoy**
5. **IP Address Spoofing**
6. **Creating Custom Packets**
7. **Randomizing Host Order**
8. **Sending Bad Checksums**
9. **Proxy Servers**
10. **Anonymizers**

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. When these packets reach a host, the IDS and firewalls behind the host generally queue all of them and process them one by one. However, since this method of processing involves greater CPU and network resource consumption, the configuration of most IDS cause them to skip fragmented packets during port scans.

Therefore, attackers use packet fragmentation tools such as Nmap and fragroute to split the probe packet into smaller packets that circumvent the port-scanning techniques employed by IDS. Once these fragments reach the destined host, they are reassembled to form a single packet.

SYN/FIN Scanning Using IP Fragments

SYN/FIN scanning using IP fragments is not a new scanning method but a modification of previous techniques. This process of scanning was developed to avoid false positives generated by other scans because of a packet filtering device on the target system. The TCP header splits into several packets to evade the packet filter. For any transmission, every TCP header must have the source and destination port for the initial packet (8-octet, 64-bit). The initialized flags in the next packet allow the remote host to reassemble the packets upon receipt via an Internet protocol module that detects the fragmented data packets using field-equivalent values of the source, destination, protocol, and identification.

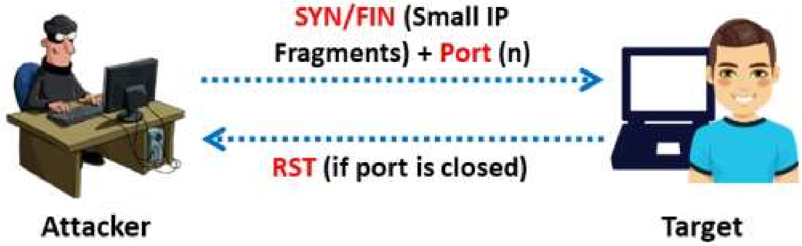


Figure 3.82: SYN/FIN scanning

In this scan, the system splits the TCP header into several fragments and transmits them over the network. However, IP reassembly on the server side may result in unpredictable and abnormal results, such as fragmentation of the IP header data. Some hosts may fail to parse and reassemble the fragmented packets, which may lead to crashes, reboots, or even network device monitoring dumps.

Some firewalls might have rule sets that block IP fragmentation queues in the kernel (e.g., CONFIG_IP_ALWAYS_DEFRAG option in the Linux kernel), although this is not widely implemented because of its adverse effects on performance. Since many IDS use signature-based methods to indicate scanning attempts on IP and/or TCP headers, the use of fragmentation will often evade this type of packet filtering and detection, resulting in a high probability of causing problems on the target network. Attackers use the SYN/FIN scanning method with IP fragmentation to evade this type of filtering and detection.

The screenshot below shows the SYN/FIN scan using the Nmap tool.

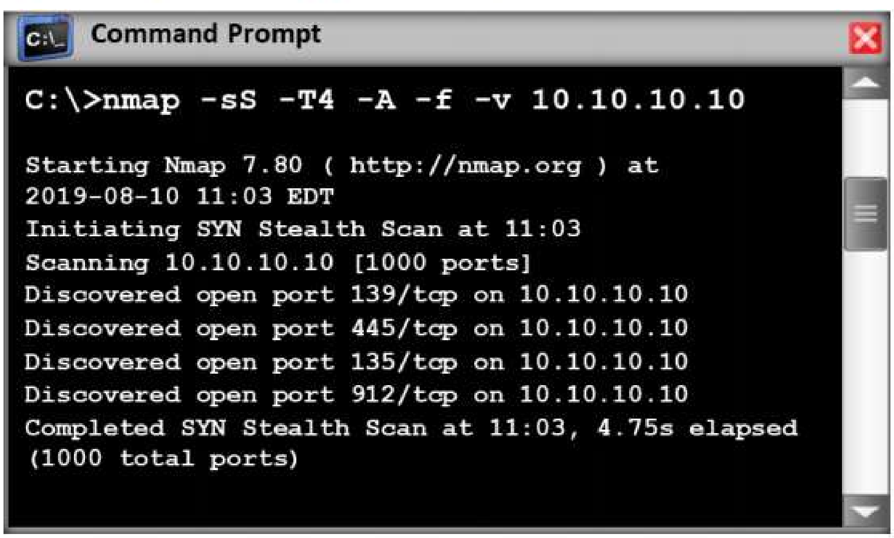
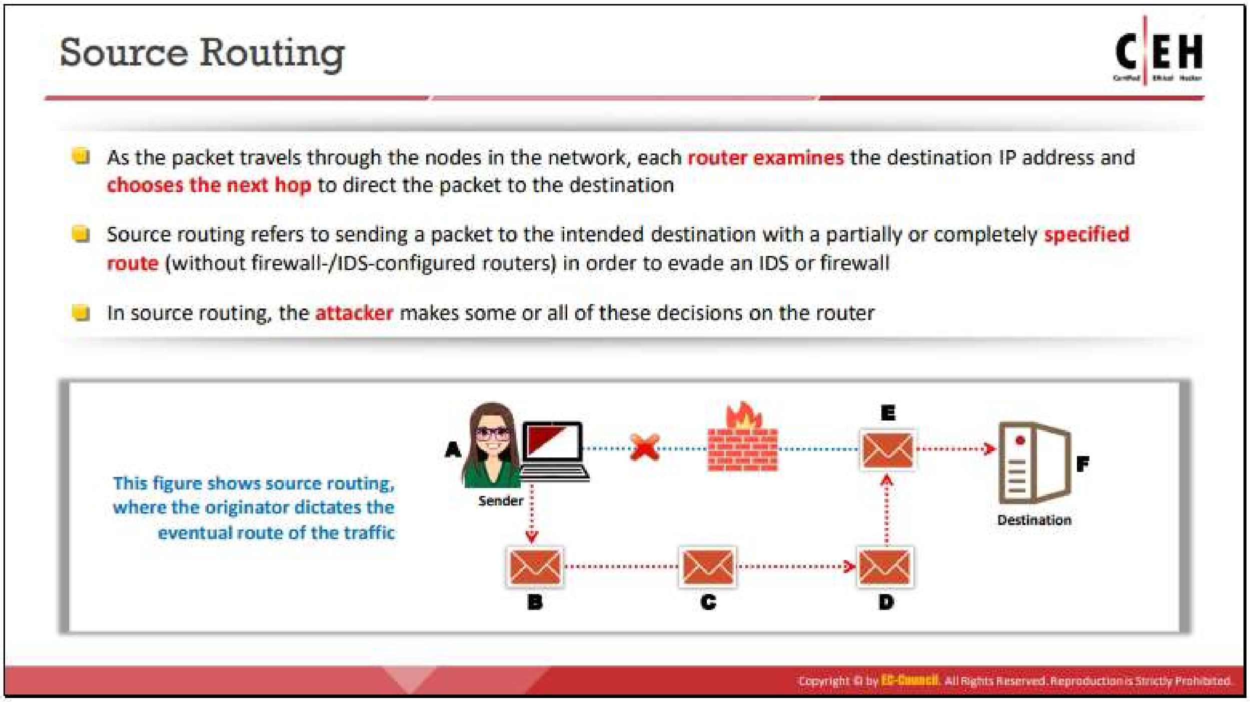


Figure 3.83: SYN/FIN scan using Nmap



Source Routing

An IP datagram contains various fields, including the IP options field, which stores source routing information and includes a list of IP addresses through which the packet travels to its destination. 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.

When attackers send malformed packets to a target, these packets hop through various routers and gateways to reach the destination. In some cases, the routers in the path might include configured firewalls and IDS that block such packets. To avoid them, attackers enforce a loose or strict source routing mechanism, in which they manipulate the IP address path in the IP options field so that the packet takes the attacker-defined path (without firewall-/IDS-configured routers) to reach the destination, thereby evading firewalls and IDS.

The figure below shows source routing, where the originator dictates the eventual route of the traffic.

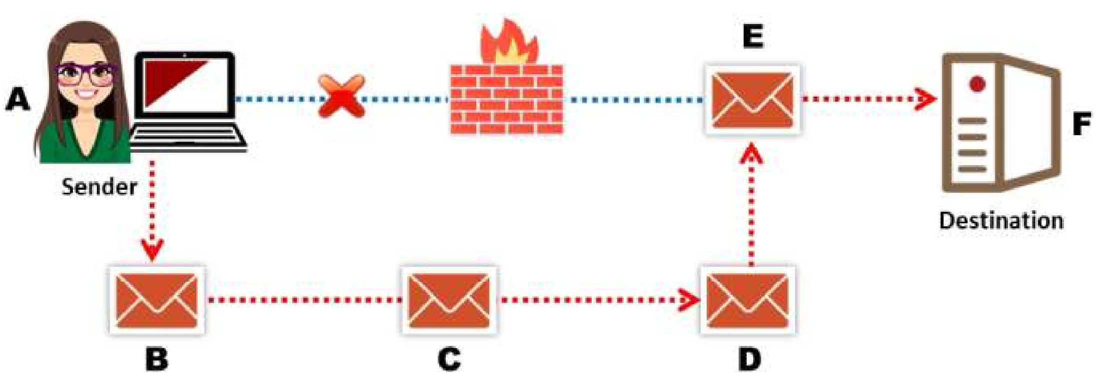
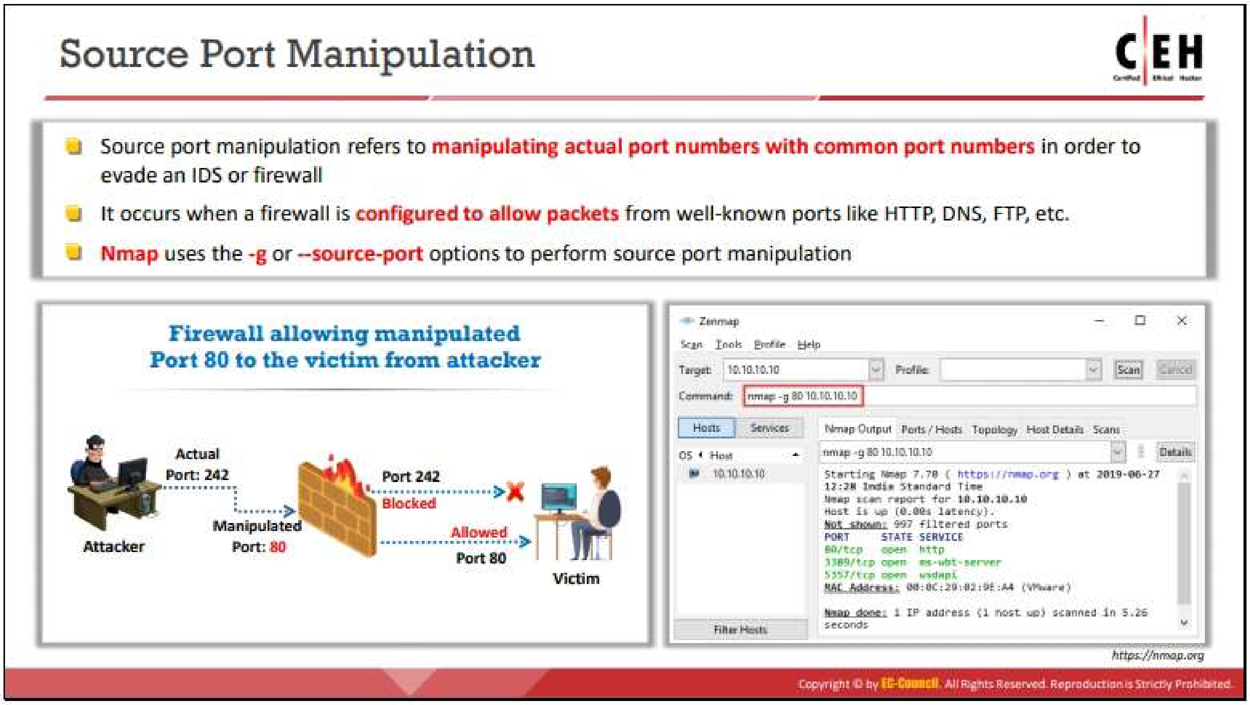


Figure 3.84: Source Routing



Source Port Manipulation

Source port manipulation is a technique used for bypassing the IDS/firewall, where the actual port numbers are manipulated with common port numbers for evading certain IDS and firewall rules. The main security misconfigurations occur because of blindly trusting the source port number. The administrator mostly configures the firewall by allowing the incoming traffic from well-known ports such as HTTP, DNS, FTP, etc. The firewall can simply allow the incoming traffic from the packets sent by the attackers using such common ports.

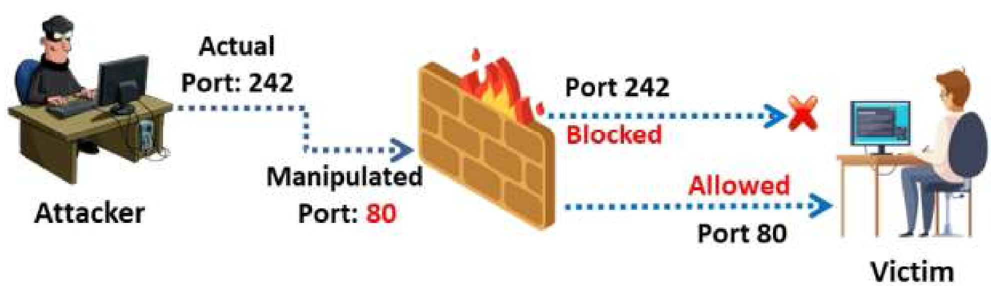


Figure 3.85: Firewall allowing manipulated port 80 to the victim from attacker

Although the firewalls can be made secure using application-level proxies or protocol-parsing firewall elements, this technique helps the attacker to bypass the firewall rules easily. The attacker tries to manipulate the original port number with the common port numbers, which can easily bypass the IDS/firewall. In Zenmap, the `-g` or `--source-port` option is used to perform source port manipulation.

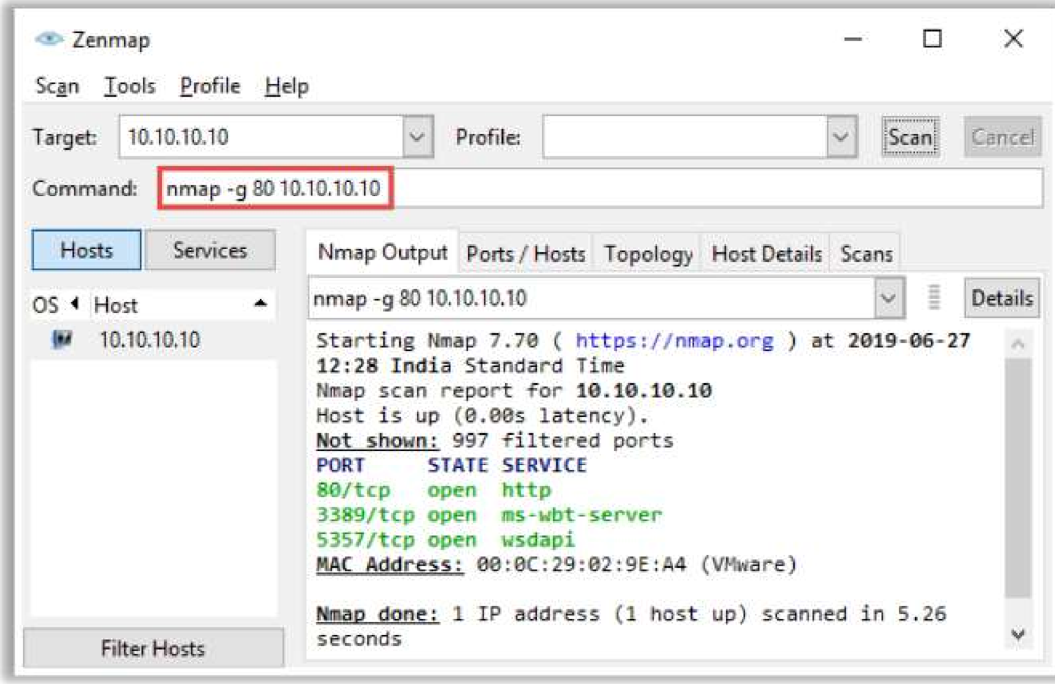


Figure 3.86: Scanning over Firewall using Nmap

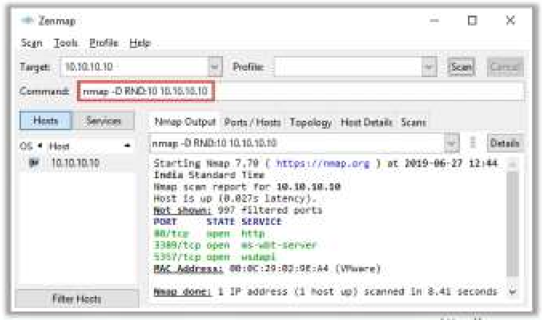
IP Address Decoy

- 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 Decoy

The IP address decoy technique refers to generating or manually specifying IP addresses of the decoys to evade IDS/firewalls. 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/firewall to determine which IP address is actually scanning the network and which IP addresses are decoys.

The Nmap scanning tool comes with a built-in scan function called a decoy scan, which cloaks a scan with decoys. This technique generates multiple IP addresses to perform a scan, thus making it difficult for the target security mechanisms such as IDS, firewalls, etc., to identify the original source from the registered logs. The target IDS might report scanning from 5– 0 IP addresses; however, it cannot differentiate between the actual scanning IP address and the innocuous decoy IPs.

You can perform two types of decoy scans using Nmap:

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

  Using this command, Nmap automatically generates a random number of decoys for the scan and randomly positions the real IP address between the decoy IPs.

  Ex. Assume that 10.10.10.10 is the target IP address to be scanned. Thus, the Nmap decoy scan command will be:

  ```
  # nmap -D RND:10 10.10.10.10
  ```

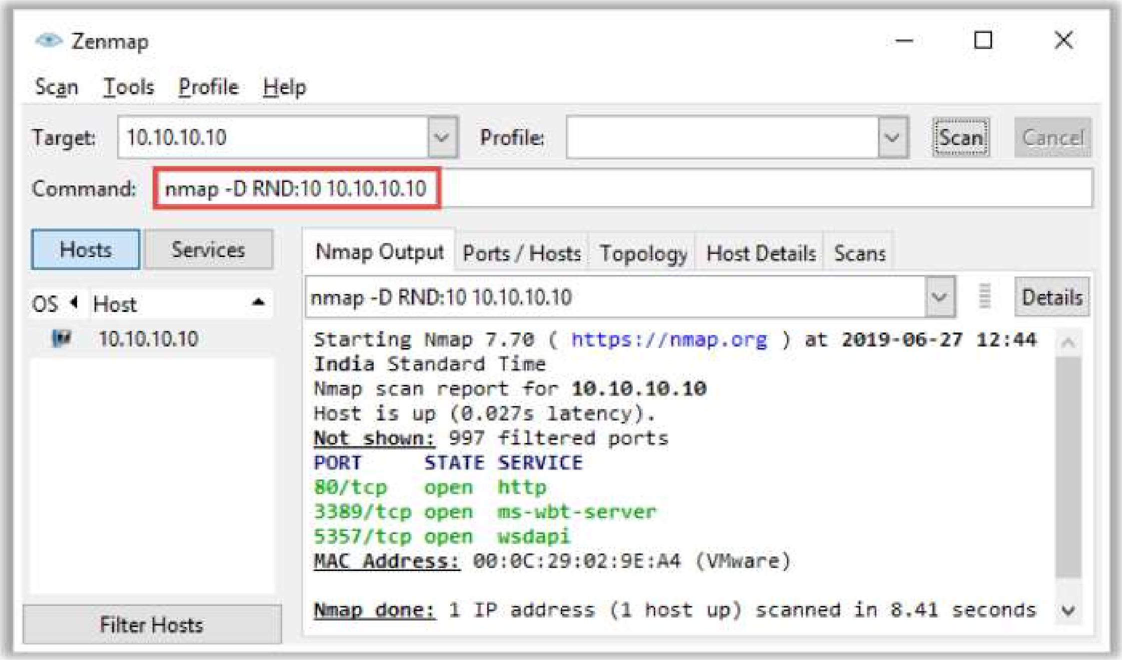


Figure 3.87: Decoy using Nmap RND option

- **nmap -D decoy1,decoy2,decoy3,...,ME,... [target]**

  Using this command, you can manually specify the IP addresses of the decoys to scan the victim's network. Here, you have to separate each decoy IP with a comma (,) and you can optionally use the ME command to position your real IP in the decoy list. If you place ME in the 4th position of the command, your real IP will be positioned at the 4th position accordingly. This is an optional command, and if you do not mention ME in your scan command, then Nmap will automatically place your real IP in any random position.

  For example, assume that 10.10.10.16 is the real source IP and 10.10.10.10 is the target IP address to be scanned. Then, the Nmap decoy command will be:

  Syntax:

  ```
  # nmap -D 192.168.0.1,172.120.2.8,192.168.2.8,10.10.10.16,10.10.10.5 10.10.10.10
  ```

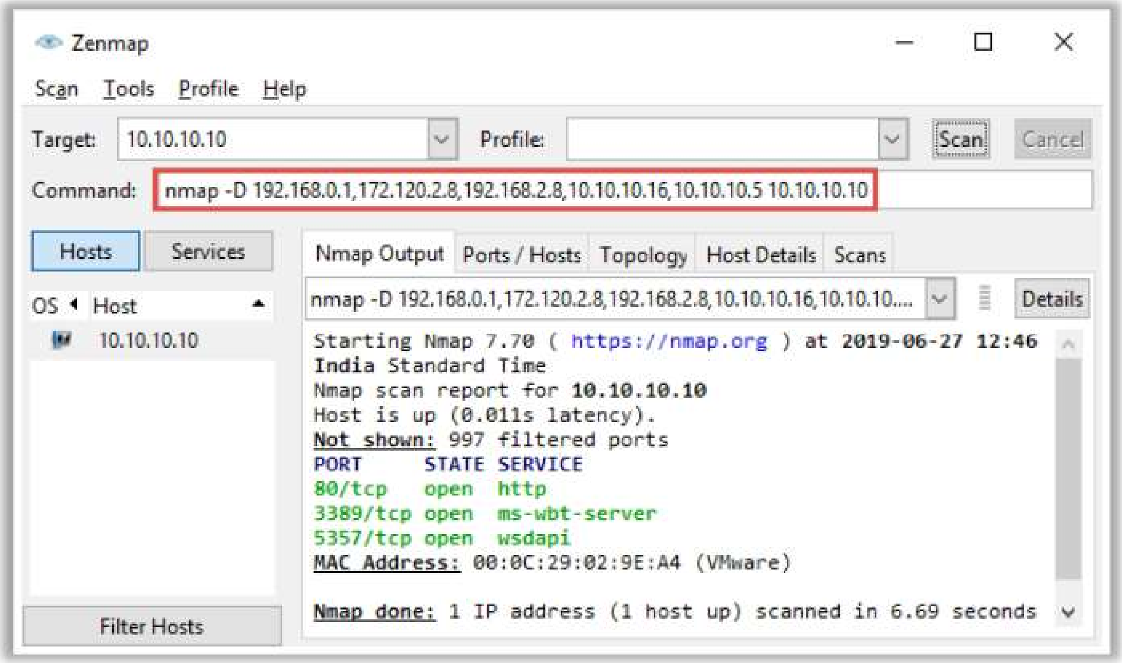
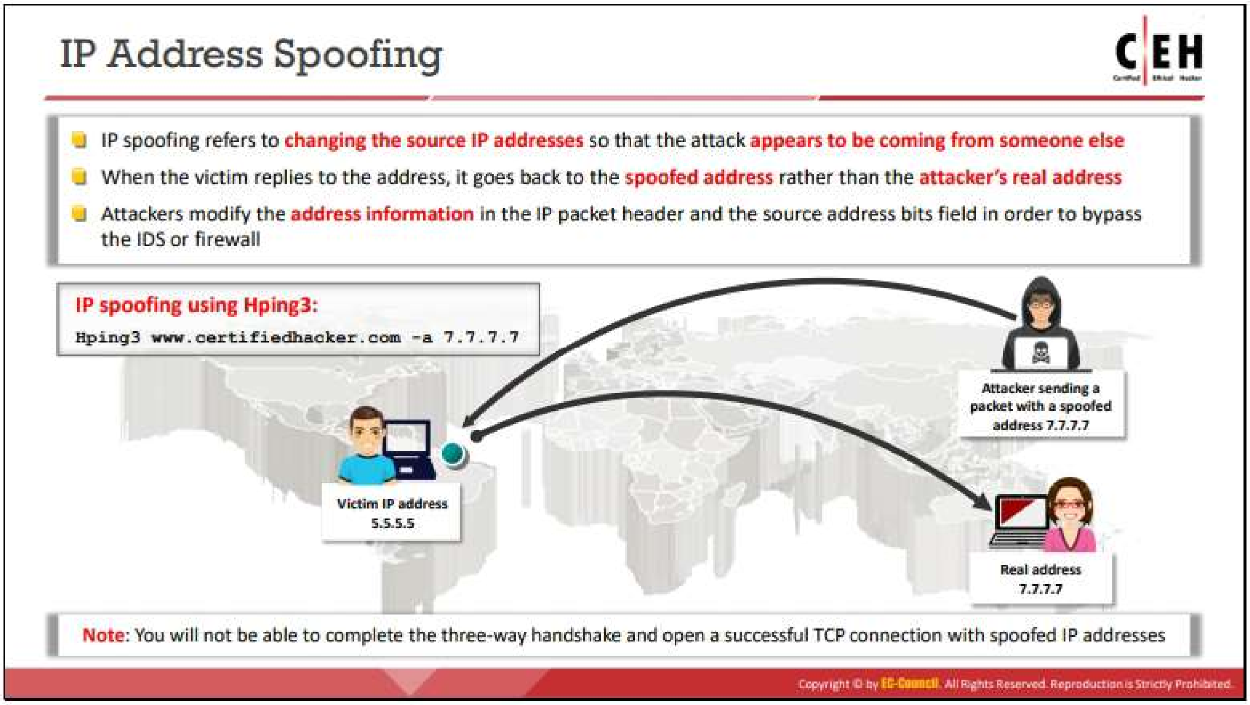


Figure 3.88: Decoy using Nmap with manual decoy list

These decoys can be generated in both initial ping scans such as ICMP, SYN, ACK, etc., and during the actual port scanning phase.

IP address decoy is a useful technique for hiding your IP address. However, it will not be successful if the target employs active mechanisms such as router path tracing, response dropping, etc. Moreover, using many decoys can slow down the scanning process and affect the accuracy of the scan.



IP Address Spoofing

Most firewalls filter packets based on the source IP address. These firewalls examine the source IP address and determine whether the packet is coming from a legitimate source or an illegitimate source. The IDS filters packets from illegitimate sources. Attackers use IP spoofing technique to bypass such IDS/firewalls.

IP address spoofing is a hijacking technique in which an attacker obtains a computer's IP address, alters the packet headers, and sends request packets to a target machine, pretending to be a legitimate host. The packets appear to be sent from a legitimate machine but are actually sent from the attacker's machine, while his/her machine's IP address is concealed. When the victim replies to the address, it goes back to the spoofed address and not to the attacker's real address. Attackers mostly use IP address spoofing to perform DoS attacks.

When the attacker sends a connection request to the target host, the target host replies to the spoofed IP address. When spoofing a nonexistent address, the target replies to a nonexistent system and then hangs until the session times out, thus consuming a significant amount of its own resources.

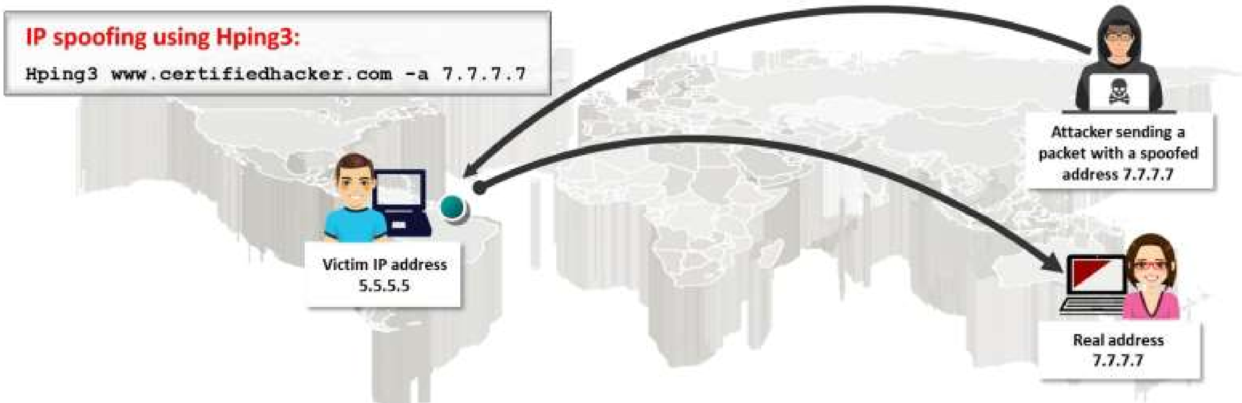


Figure 3.89: IP Spoofing using Hping3

IP spoofing using Hping3:

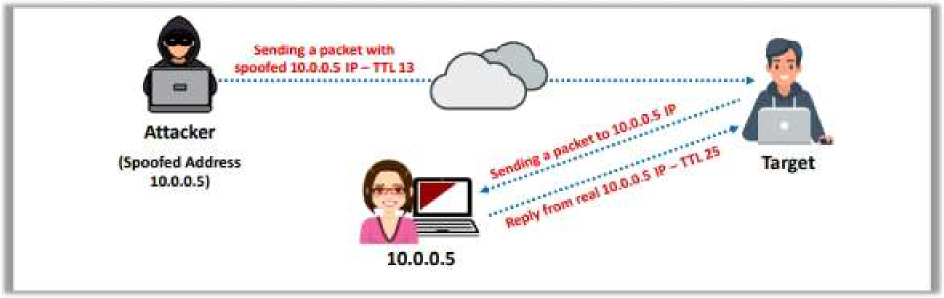
```
Hping3 www.certifiedhacker.com -a 7.7.7.7
```

You can use Hping3 to perform IP spoofing. The above command helps you to send arbitrary TCP/IP packets to network hosts.

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



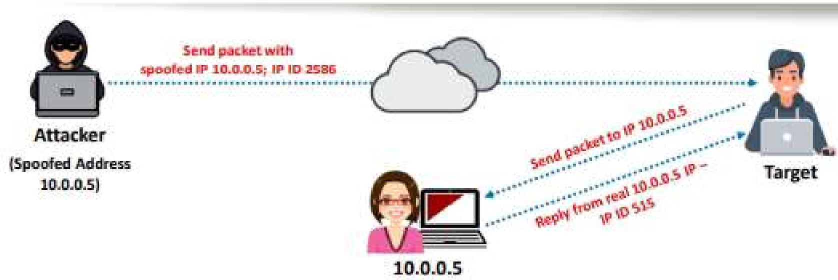
Note: Normal traffic from one host can contrast TTLs depending on traffic patterns

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

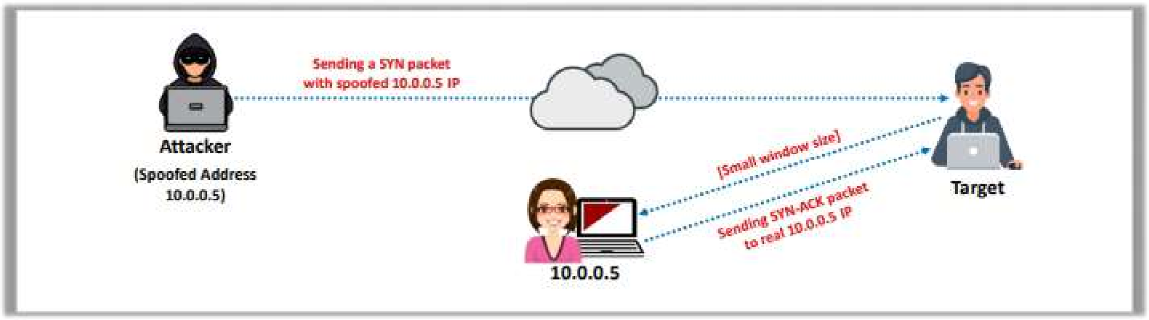
02 If the IPIDs are **not close in value** to the packet being checked, then the suspected traffic is spoofed

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 Detection Techniques

- **Direct TTL Probes**

  In this technique, you initially send a packet (ping request) to the legitimate host and wait for a reply. Check whether the TTL value in the reply matches with that of the packet you are checking. Both will have the same TTL if they are using the same protocol. Although the initial TTL values vary according to the protocol used, a few initial TTL values are commonly used. For TCP/UDP, the values are 64 and 128; for ICMP, they are 128 and 255.

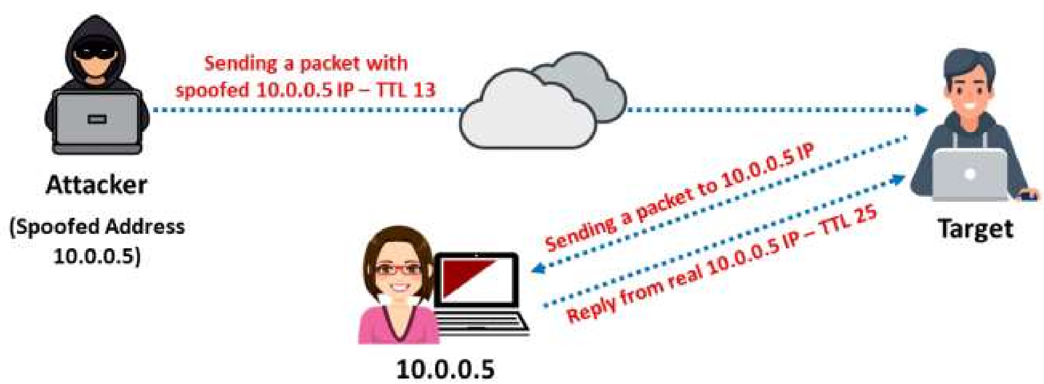


Figure 3.90: IP Spoofing detection technique: Direct TTL Probes

If the reply is from a different protocol, then you should check the actual hop count to detect the spoofed packets. Deduct the TTL value in the reply from the initial TTL value to determine the hop count. The packet is a spoofed packet if the reply TTL does not match the TTL of the packet. It will be very easy to launch an attack if the attacker knows the hop count between the source and the host. In this case, the test result is a false negative.

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 Identification Number**

  Users can identify spoofed packets by monitoring the IP identification (IPID) number in the IP packet headers. The IPID increases incrementally each time a system sends a packet. Every IP packet on the network has a "fragment identification" number, which is increased by one for every packet transmission. To identify whether a packet is spoofed, send a probe packet to the source IP address of the packet and observe the IPID number in the reply. The IPID value in the response packet must be close to but slightly greater than the IPID value of the probe packet. The source address of the IP packet is spoofed if the IPID of the response packet is not close to that of the probe packet.

  This method is effective even when both the attacker and the target are on the same subnet.

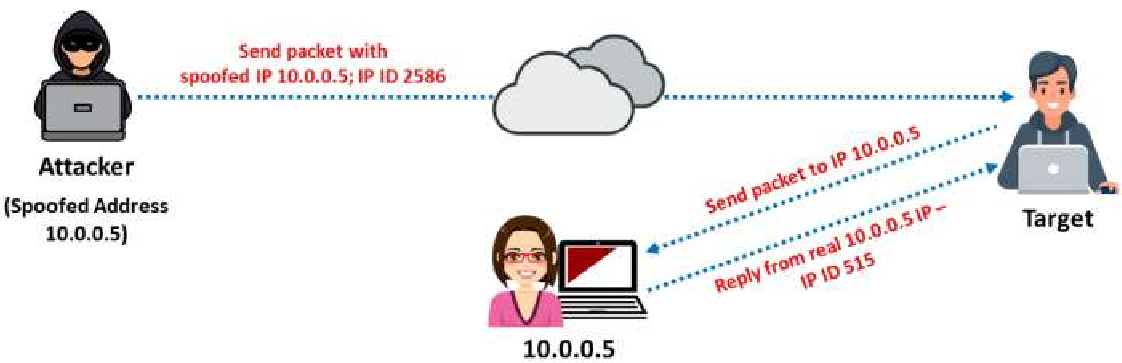


Figure 3.91: IP Spoofing detection technique: IP Identification Number

- **TCP Flow Control Method**

  The TCP can optimize the flow control on both the sender's and the receiver's end with its algorithm. The algorithm accomplishes flow control using the sliding window principle. The user can control the flow of IP packets by the window size field in the TCP header. This field represents the maximum amount of data that the recipient can receive and the maximum amount of data that the sender can transmit without acknowledgement. Thus, this field helps to control data flow. The sender should stop sending data whenever the window size is set to zero.

  In general flow control, the sender should stop sending data once the initial window size is exhausted. The attacker, who is unaware of the ACK packet containing window size information, might continue to send data to the victim. If the victim receives data packets beyond the window size, they are spoofed packets. For effective flow control and early detection of spoofing, the initial window size must be very small.

  Most spoofing attacks occur during the handshake, as it is challenging to build multiple spoofing replies with the correct sequence number. Therefore, apply the flow control spoofed packet detection method to the handshake. In a TCP handshake, the host sending the initial SYN packet waits for SYN-ACK before sending the ACK packet. To check whether you are getting the SYN request from a genuine client or a spoofed one, set SYN-ACK to zero. If the sender sends an ACK with any data, it means that the sender is a spoofed one. This is because when SYN-ACK is set to zero, the sender must respond to it only with the ACK packet, without additional data.

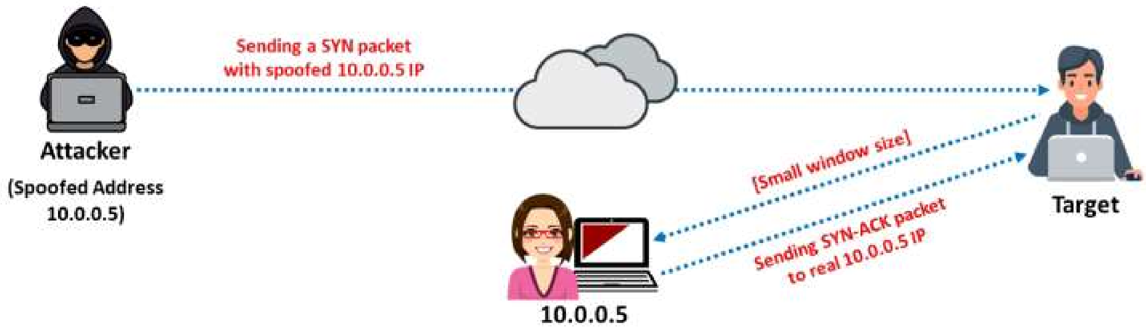


Figure 3.92: IP Spoofing detection technique: TCP Flow Control Method

Attackers sending spoofed TCP packets will not receive the target's SYN-ACK packets. Attackers cannot respond to changes in the congestion window size. When the 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

IP Spoofing Countermeasures

In ethical hacking, the ethical hacker, also known as the "pen tester," has to perform an additional task that a normal hacker does not follow (i.e., adopting countermeasures against the respective vulnerabilities determined through hacking). This is essential because knowing security loopholes in your network is worthless unless you adopt measures to protect them against real hackers. As mentioned previously, IP spoofing is one of the techniques that a hacker adopts to break into the target network. Therefore, to protect your network from external hackers, you should apply IP spoofing countermeasures to your network security settings. Some IP spoofing countermeasures that you can apply are as follows:

- **Avoid trust relationships**

  Do not rely on IP-based authentication. Attackers may spoof themselves as trusted hosts and send malicious packets to you. If you accept these packets under the assumption that they are "clean" because they are from your trusted host, the malicious code will infect your system. Therefore, it is advisable to test all packets, even when they come from one of your trusted hosts. You can avoid this problem by implementing password authentication along with trust-relationship-based authentication.

- **Use firewalls and filtering mechanisms**

  As stated above, you should filter all the incoming and outgoing packets to avoid attacks and sensitive information loss. A firewall can restrict malicious packets from entering your private network and prevent severe data loss. You can use access control lists (ACLs) to block unauthorized access. At the same time, there is a possibility of an insider attack. Inside attackers can send sensitive information about your business to your competitors, which could lead to monetary loss and other issues. Another risk of outgoing packets is that an attacker will succeed in installing a malicious sniffing program running in a hidden mode on your network. These programs gather and send all your network information to the attacker without any notification after filtering the outgoing packets. Therefore, you should assign the same importance to the scanning of outgoing packets as you would to that of incoming packets.

- **Use random initial sequence numbers**

  Most devices choose their ISN based on timed counters. This makes the ISNs predictable, as it is easy for an attacker to determine the concept of generating the ISN. The attacker can determine the ISN of the next TCP connection by analyzing the ISN of the current session or connection. If the attacker can predict the ISN, then he/she can establish a malicious connection to the server and sniff out your network traffic. To avoid this risk, use random initial sequence numbers.

- **Ingress filtering**

  Ingress filtering prevents spoofed traffic from entering the Internet. It is applied to routers because it enhances the functionality of the routers and blocks spoofed traffic. Configuring and using ACLs that drop packets with the source address outside the defined range is one method of implementing ingress filtering.

- **Egress filtering**

  Egress filtering refers to a practice that aims to prevent IP spoofing by blocking outgoing packets with a source address that is not inside.

- **Use encryption**

  If you want to attain maximum network security, then use strong encryption for all the traffic placed onto the transmission media without considering its type and location. This is the best way to prevent IP spoofing attacks. IPsec can be used to reduce the IP spoofing risk drastically, as it provides data authentication, integrity, and confidentiality. Furthermore, ACLs can be used for blocking private IP addresses at the downstream interfaces. Encryption sessions should be enabled on the router so that trusted hosts can communicate securely with local hosts. Attackers tend to focus on easy-to-compromise targets. If an attacker wants to break into the encrypted network, he or she has to decrypt a whole slew of encrypted packets, which is a difficult task. Therefore, the attacker is likely to move on and try to find another target that is easy to compromise or simply abort the attempt. Moreover, use the latest encryption algorithms that provide strong security.

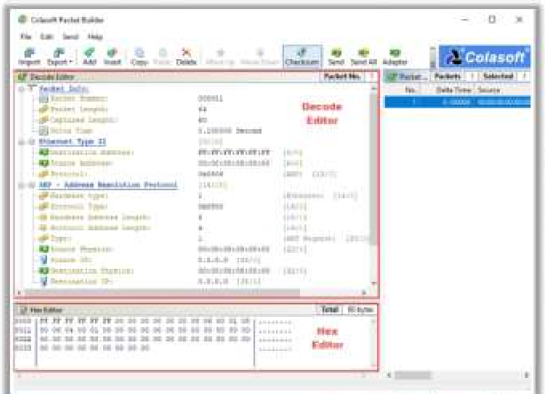
- **SYN flooding countermeasures**

  Countermeasures against SYN flooding attacks can also help you to avoid IP spoofing attacks.

Creating Custom Packets

Creating Custom Packets by using Packet Crafting Tools

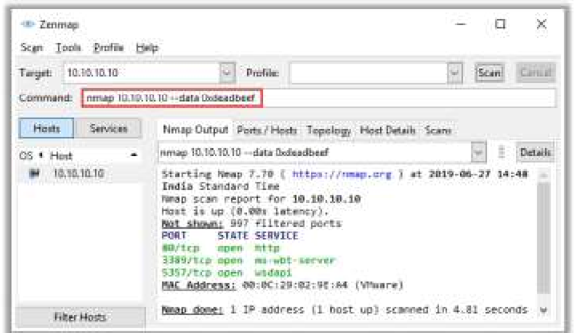
- Attackers **create custom TCP packets** using various packet crafting tools like **Colasoft Packet Builder**, **NetScanTools Pro**, etc. to scan a target beyond a firewall



https://www.colasoft.com

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**

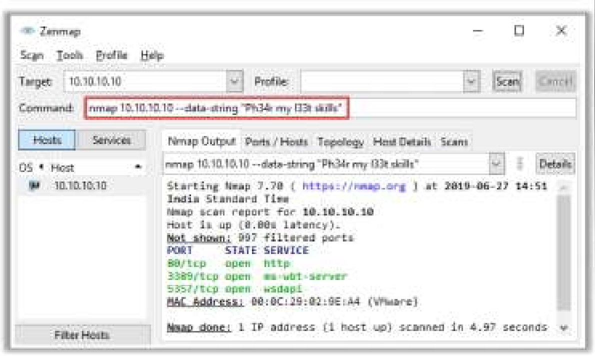


https://nmap.org

Creating Custom Packets (Cont'd)

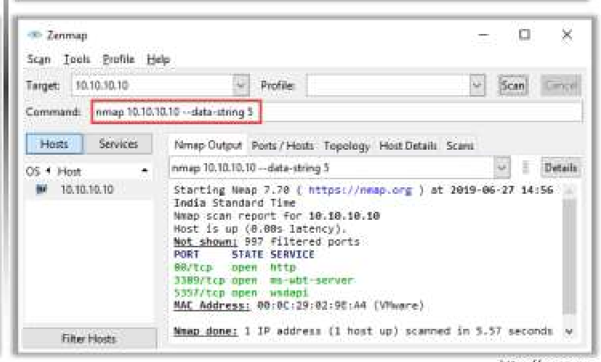
Creating Custom Packets by Appending Custom String

- 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 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**



https://nmap.org

Creating Custom Packets

The attacker creates and sends custom packets to scan the intended target beyond the IDS/firewalls. Various techniques are used to create custom packets. Some of them are mentioned below:

- **Creating Custom Packets by using Packet Crafting Tools**

Attackers create custom TCP packets to scan the target by bypassing the firewalls. Attackers use various packet crafting tools such as Colasoft packet builder (*https://www.colasoft.com*), NetScanTools Pro (*https://www.netscantools.com*), etc., to scan the target that is beyond the firewall. Packet crafting tools craft and send packet streams (custom packets) using different protocols at different transfer rates.

- **Colasoft Packet Builder**

  Source: *https://www.colasoft.com*

  Colasoft Packet Builder is a tool that allows an attacker to create custom network packets and helps security professionals assess the network. The attacker can select a TCP packet from the provided templates and change the parameters in the decoder, hexadecimal, or ASCII editor to create a packet. In addition to building packets, Colasoft Packet Builder supports saving packets to packet files and sending packets to the network.

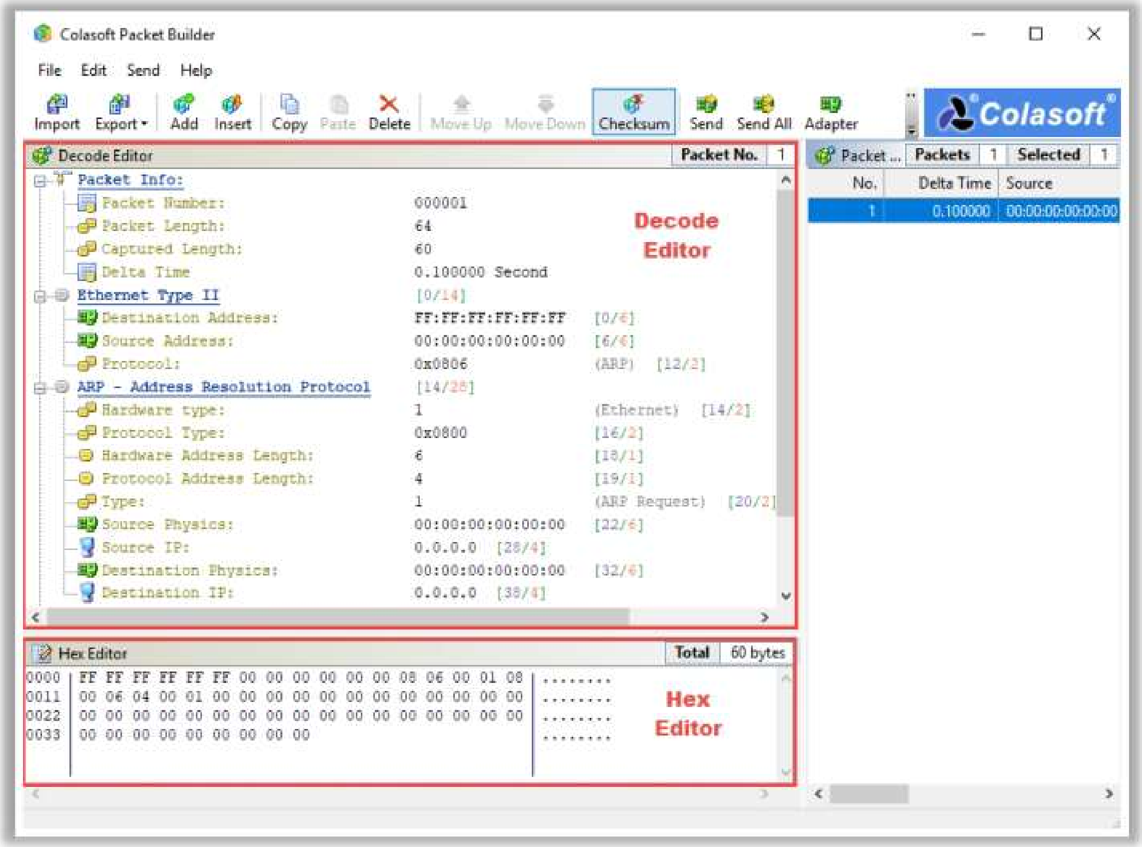


Figure 3.93: Screenshot of Colasoft Packet Builder

There are three views in the Packet Builder: Packet List, Decode Editor, and Hex Editor.

- Packet List displays all the constructed packets. When you select one or more packets in Packet List, the first highlighted packet is displayed in both Decode Editor and Hex Editor for editing.

- In Hex Editor, the data of the packet are represented as hexadecimal values and ASCII characters; nonprintable characters are represented by a dot (".") in the ASCII section. You can edit either the hexadecimal values or the ASCII characters.
- Decode Editor allows the attacker to edit packets without remembering the value length, byte order, and offsets. You can select a field and change the value in the edit box.

For creating a packet, you can use the add or insert packet command in the Edit menu or the Toolbar to create a new packet.

The attacker can send a constructed packet to wire directly and control how Colasoft Packet Builder sends the packets, specifying, for example, the interval between packets, loop times, and delay between loops.

This packet builder audits networks and checks the network protection against attacks and intruders. Attackers may use this packet builder to create fragmented packets to bypass network firewalls and IDS systems. They can also create packets and flood the victim with a very large number of packets, which could result in DoS attacks.

- **Creating Custom Packets by Appending Custom Binary Data**

  Attackers send binary data (0's and 1's) as payloads in the packets sent to the target machine present behind the firewall. The option used by Nmap for appending custom binary data to the sent packets is **--data <hex string>**. Any **<hex string>** is specified in the formats **0xAABBCCDDEEFF<...>**, **AABBCCDDEEFF<...>**, or **\xAA\xBB\xCC\xDD\xEE\xFF<...>**. To perform a byte-order conversion, the specified information should be based on the receiver's expectations. Attackers can use this technique to scan the target by manipulating the firewalls by appending custom binary or hex data to the sent packets.

  Example: **--data 0xdeadbeef** (or) **--data \xCA\xFE\x09**

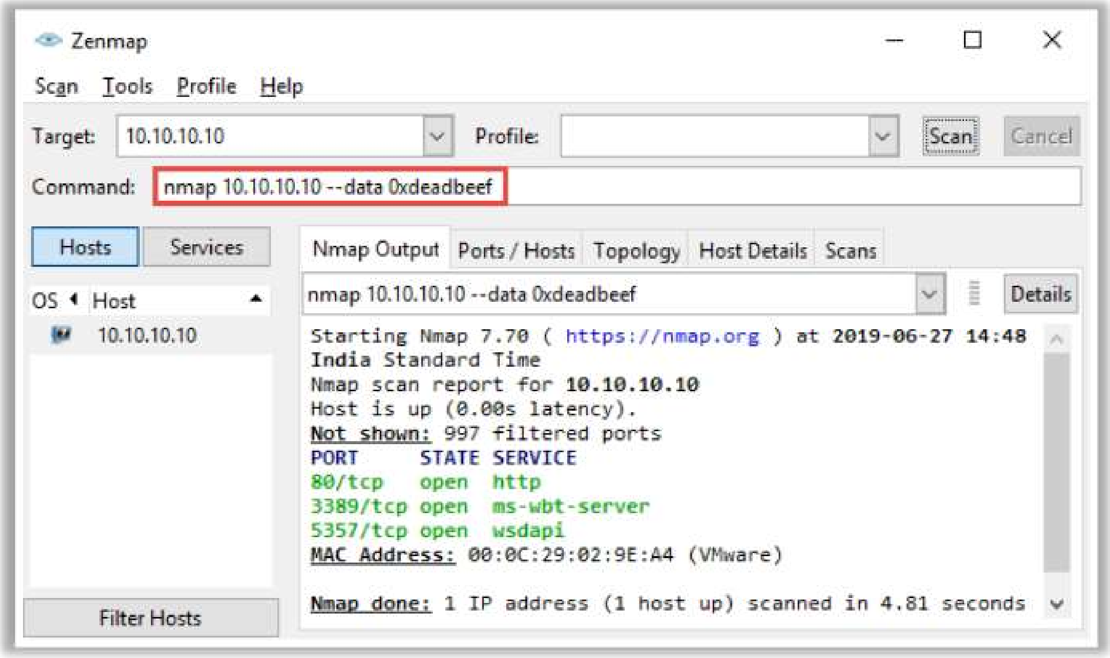


Figure 3.94: Screenshot of appending binary string in Zenmap

- **Creating Custom Packets by Appending Custom String**

  Attackers send regular strings as payloads in the packets sent to the target machine for scanning beyond the firewall. The option used by Nmap for appending a custom string to the sent packets is **`--data-string <string>`**. The **`<string>`** can contain any string and a few characters depend on the system's location; however, it is not guaranteed whether the same information is retrieved. The string is enclosed with double quotes ("") and special characters from the shell are not used. Attackers can use this technique to scan the target by manipulating the firewalls by appending custom string data to the sent packets.

  Example: **`--data-string "Scan conducted by Security Ops, extension 7192"`** (or) **`--data-string "Ph34r my l33t skills"`**.

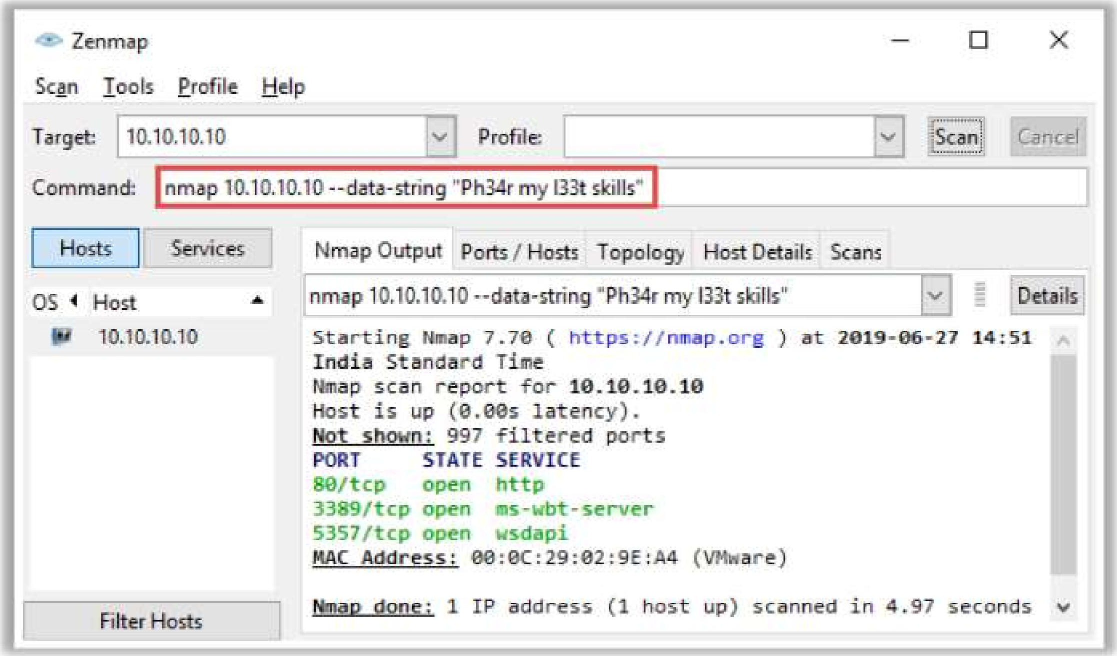


Figure 3.95: Screenshot of appending custom string in Zenmap

- **Creating Custom Packets by Appending Random Data**

  Attackers append a number of random data bytes to most packets sent without using any protocol-specific payloads. The option used by Nmap for appending random data to the sent packets is **`--data-length <number>`**. For protocol-specific and no random payloads, **`--data-length 0`** is used. The (-O) OS detection packets are not usually affected, as probe consistency is needed for it to be accurate. By default, a few UDP ports and IP protocols get a custom payload. Attackers can use this technique to scan the target by manipulating the firewalls by appending random data or numbers to the sent packets.

  Example: **`--data-string 1`** (or) **`--data-string 5`**

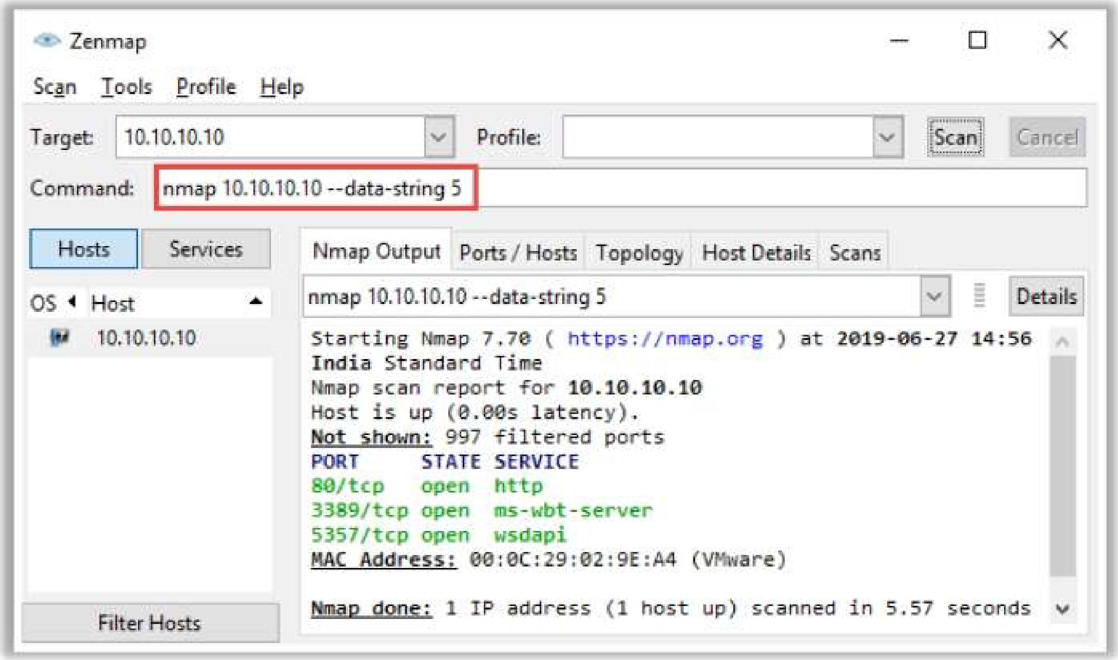
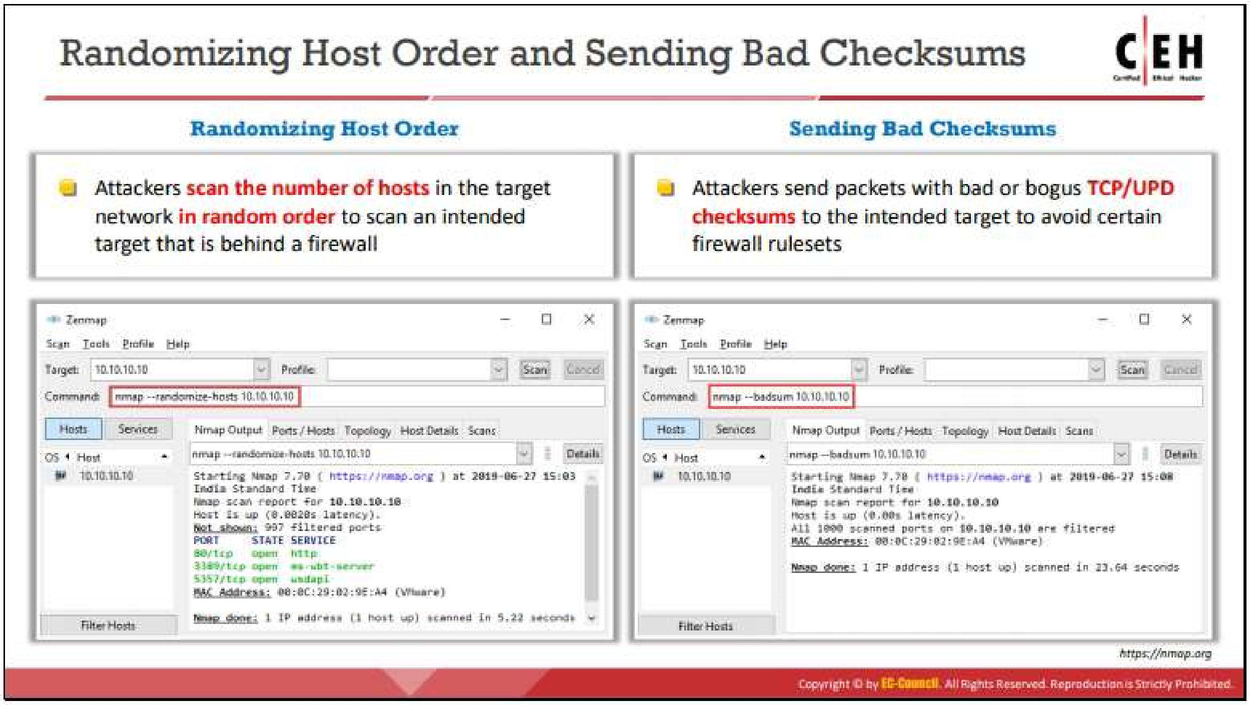


Figure 3.96: Screenshot of appending random string in Zenmap



Randomizing Host Order and Sending Bad Checksums

Randomizing Host Order

The attacker scans the number of hosts in the target network in a random order to scan the intended target that is lying beyond the firewall. The option used by Nmap to scan with a random host order is `--randomize-hosts`.

This technique instructs Nmap to shuffle each group of 16384 hosts before scanning with slow timing options, thus making the scan less notable to network monitoring systems and firewalls. If larger group sizes are randomized, the `PING_GROUP_SZ` should be increased in `nmap.h` and it should be compiled again. Another method can be followed by generating the target IP list with the list scan command `-sL -n -oN <filename>` and then randomizing it with a Perl script and providing the whole list to Nmap using the `-iL` command.

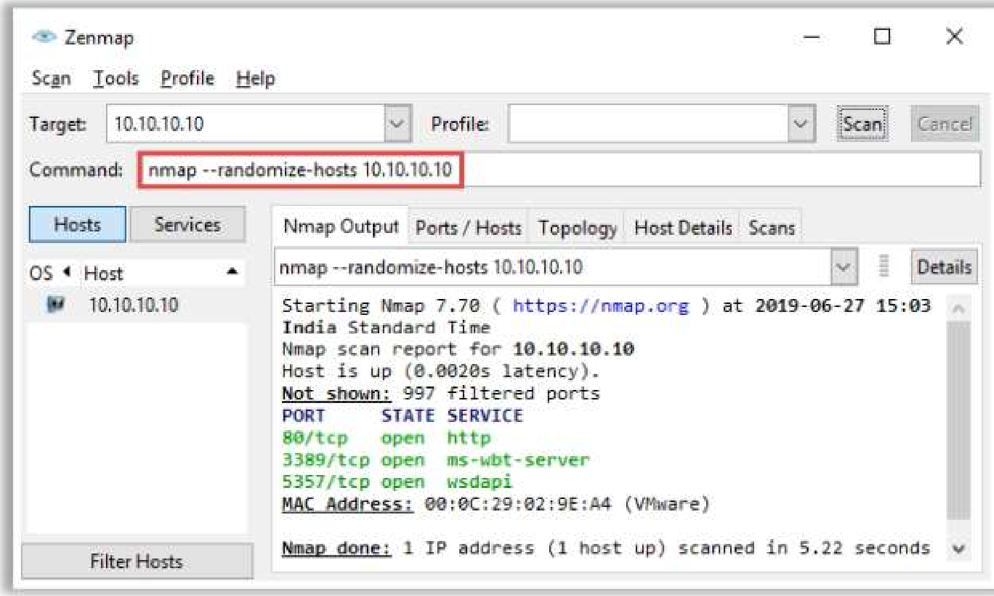


Figure 3.97: Screenshot of randomizing hosts in Zenmap

Sending Bad Checksums

The attacker sends packets with bad or bogus TCP/UPD checksums to the intended target to avoid certain firewall rule sets. TCP/UPD checksums are used to ensure data integrity. Sending packets with incorrect checksums can help attackers to acquire information from improperly configured systems by checking for any response. If there is a response, then it is from the IDS or firewall, which did not verify the obtained checksum. If there is no response or the packets are dropped, then it can be inferred that the system is configured. This technique instructs Nmap to send packets with invalid TCP, UDP, or SCTP checksums to the target host. The option used by Nmap is **`--badsum`**.

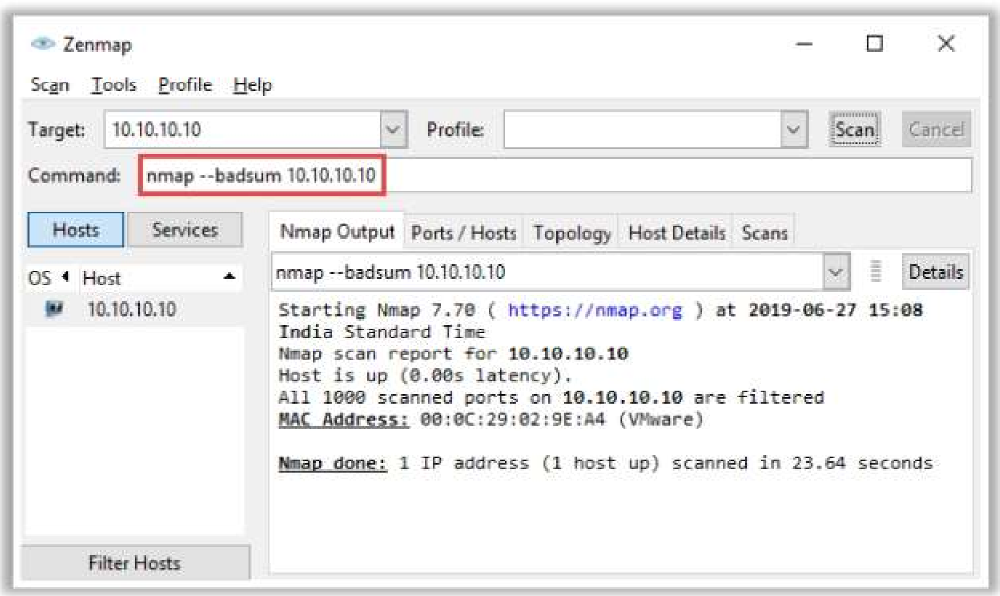
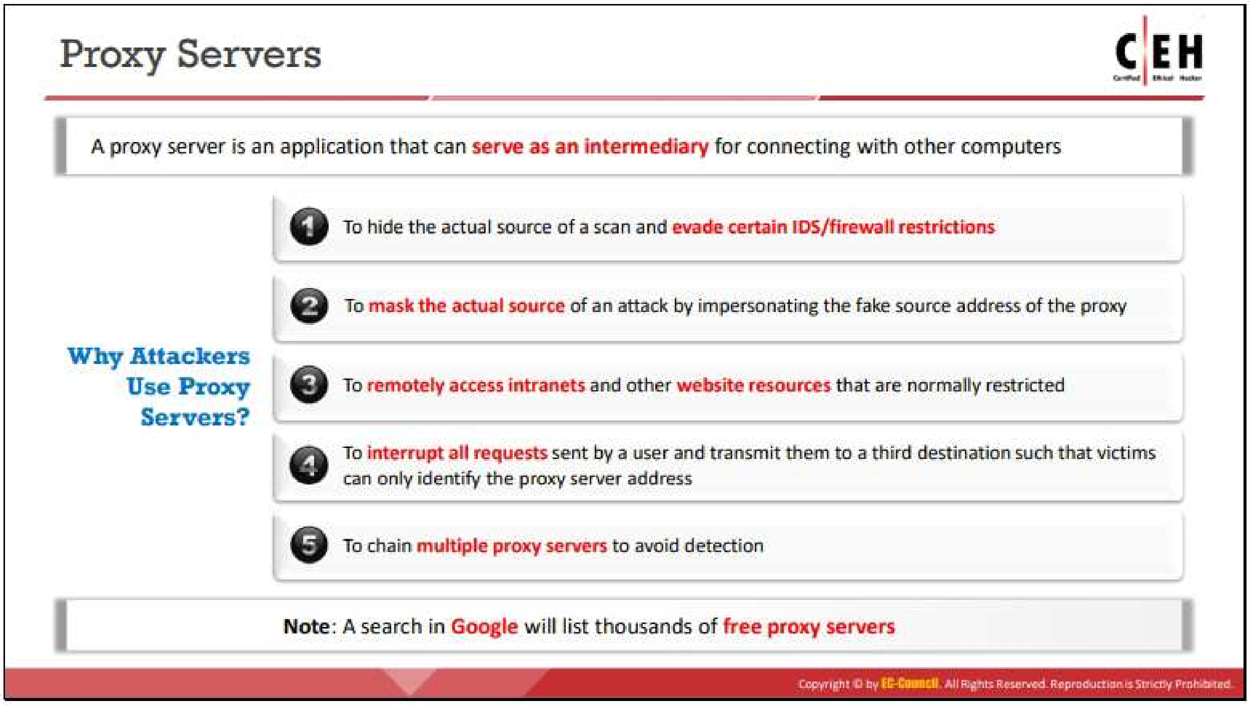


Figure 3.98: Screenshot of scanning by sending bad checksums in Zenmap



Proxy Servers

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

A proxy server is used:

- As a firewall and to protect the local network from external attacks.
- As an IP address multiplexer that allows several computers to connect to the Internet when you have only one IP address (NAT/PAT).
- To anonymize web surfing (to some extent).
- To extract unwanted content, such as ads or “unsuitable” material (using specialized proxy servers).
- To provide some protection against hacking attacks.
- To save bandwidth.

How does a proxy server work?

Initially, when you use a proxy to request a particular web page on an actual server, the proxy server receives it. The proxy server then sends your request to the actual server on your behalf. It mediates between you and the actual server to transmit and respond to the request, as shown in the figure below.

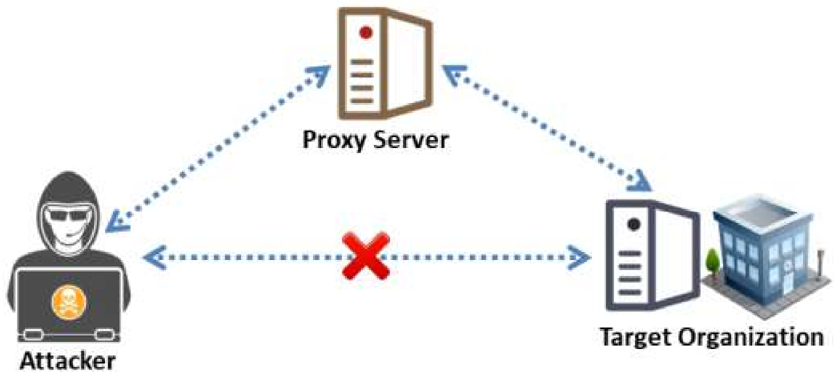


Figure 3.99: Attacker using a proxy server for connecting to the target

In this process, the proxy receives the communication between the client and the destination application. To take advantage of a proxy server, an attacker must configure client programs so that they can send their requests to the proxy server instead of the final destination.

Why Attackers Use Proxy Servers?

It is easier for an attacker to attack or hack a particular system than to conceal the attack source. Therefore, the primary challenge for an attacker is to hide his/her identity so that he/she cannot be traced. Thus, the attacker uses a proxy server to avoid attack detection by masking his/her IP address. When the attacker uses a proxy to connect to the target system, the server logs will record the proxy's source address rather than the attacker's source address.

Proxy sites help the attacker to browse the Internet anonymously and access blocked sites (i.e., evade firewall restrictions). Thus, the attacker can surf restricted sites anonymously without using the source IP address.

Attackers use proxy servers:

- To hide the actual source of a scan and evade certain IDS/firewall restrictions.
- To hide the source IP address so that they can hack without any legal corollary.
- To mask the actual source of the attack by employing a fake source address of the proxy.
- To remotely access intranets and other website resources that are normally off limits.
- To interrupt all the requests sent by a user and transmit them to a third destination; hence, victims will only be able to identify the proxy server address.
- To chain multiple proxy servers to avoid detection.

Free Proxy Servers

Some free proxy servers available on the Internet, which can help you to access restricted sites without revealing your IP address. In the **Google** search engine, type "**Free Proxy Servers**" to see a list of such servers. Select one from this list and download and install it to browse anonymously without revealing your legitimate IP address.

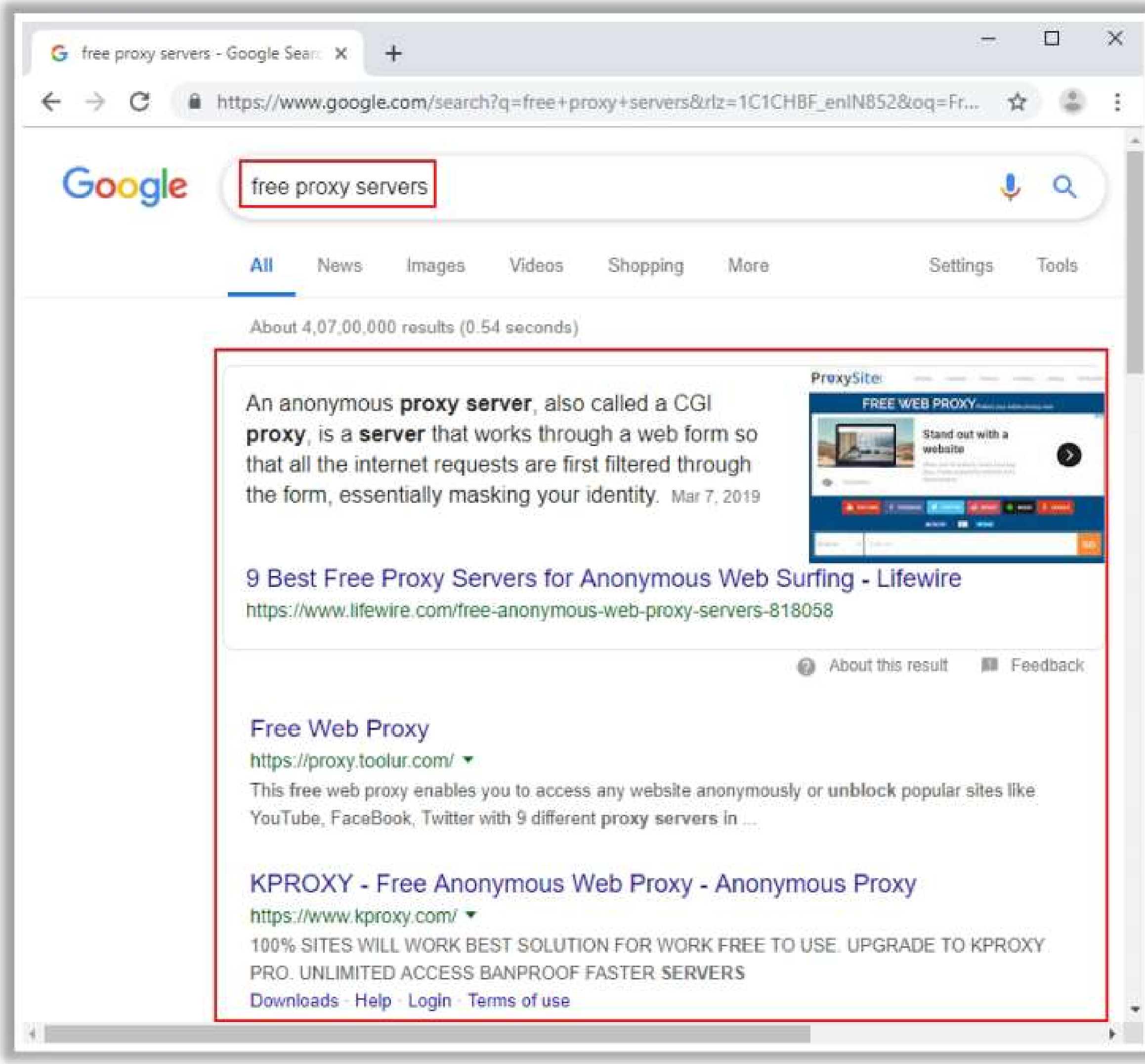
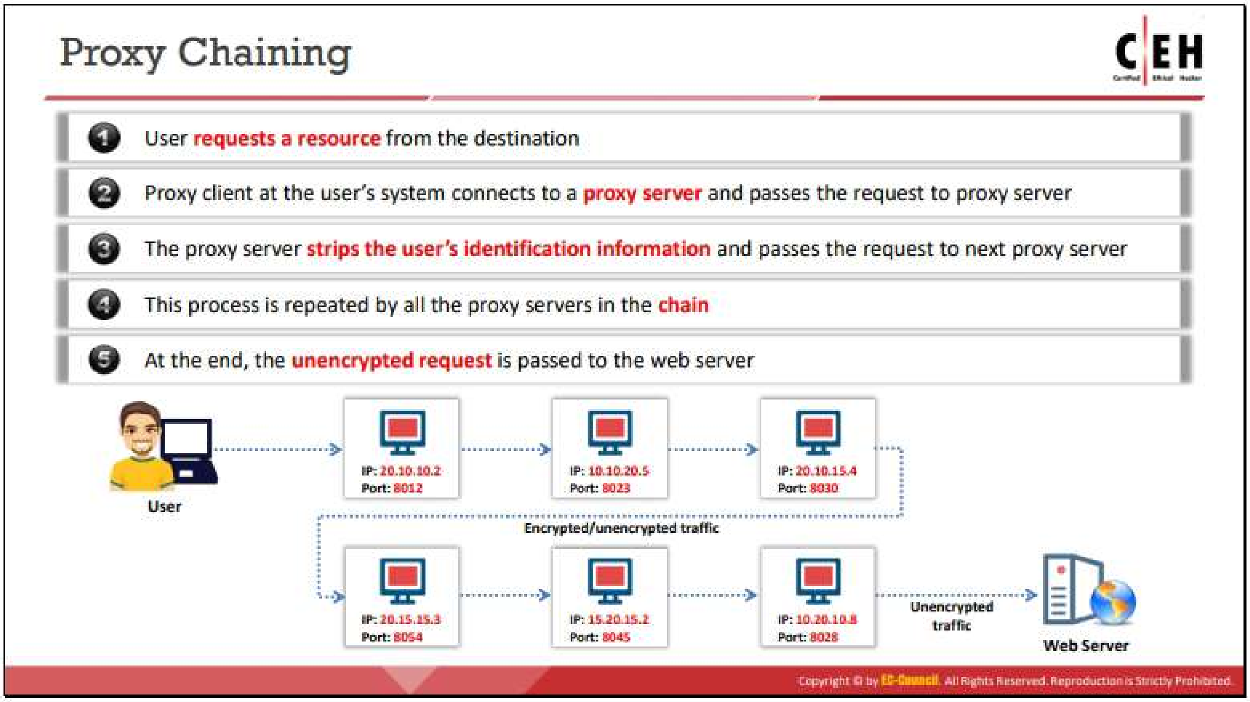


Figure 3.100: Free Proxy Servers



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.

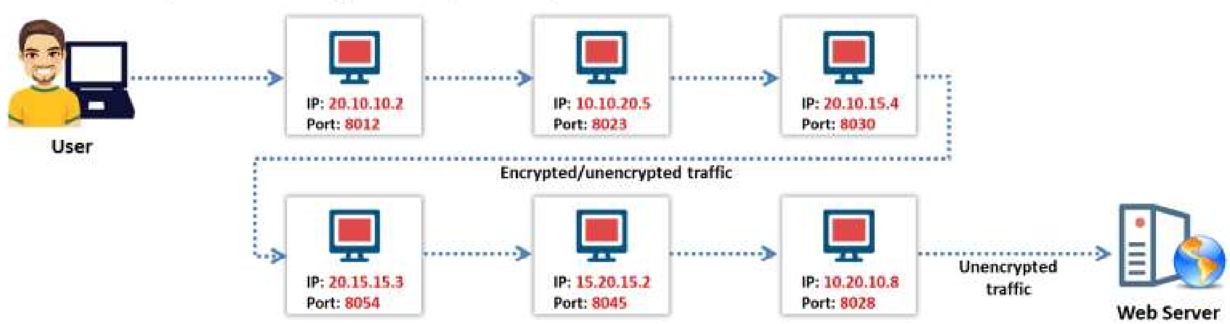
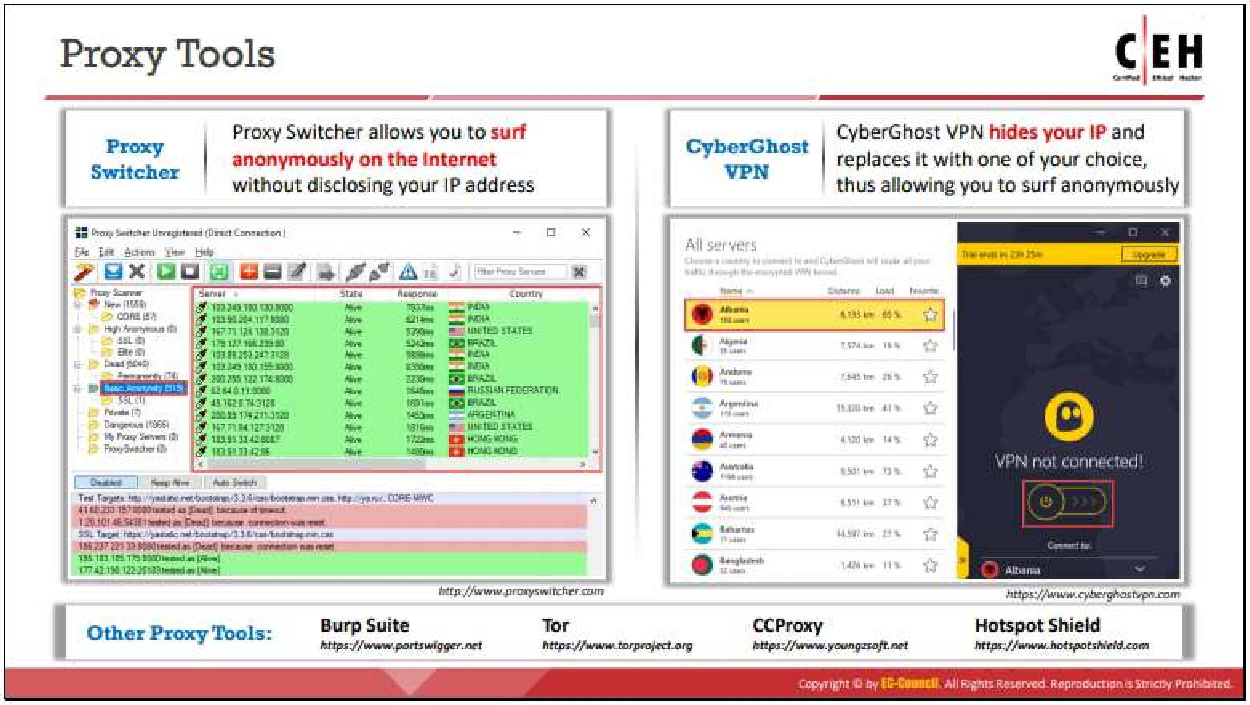


Figure 3.101: Proxy Chaining



Proxy Tools

Proxy tools are intended to allow users to surf the Internet anonymously by keeping their IP hidden through a chain of SOCKS or HTTP proxies. These tools can also act as HTTP, mail, FTP, SOCKS, news, telnet, and HTTPS proxy servers.

- **Proxy Switcher**

  Source: *http://www.proxyswitcher.com*

  Proxy Switcher allows attackers to surf the Internet anonymously without disclosing their IP address. It also helps attackers to access various blocked sites in the organization. In addition, it avoids all sorts of limitations imposed by target sites.

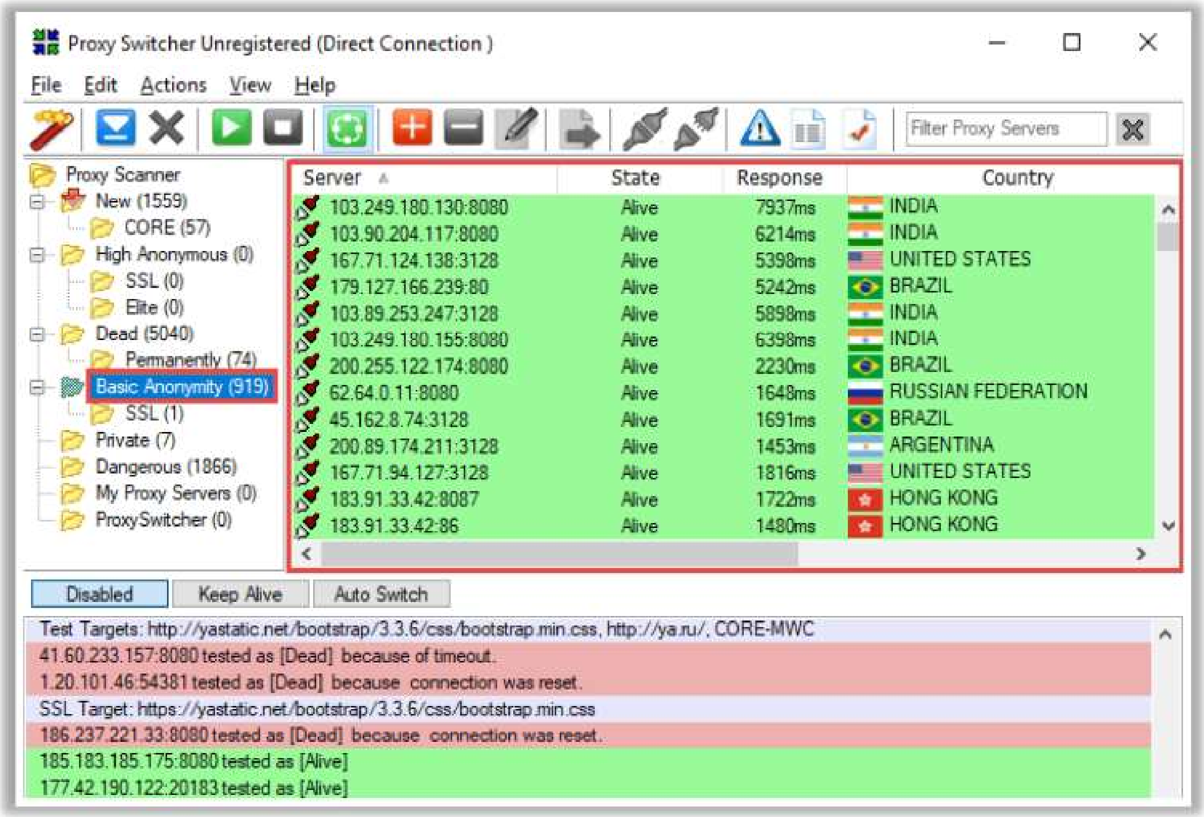


Figure 3.102: Screenshot of Proxy Switcher

- **CyberGhost VPN**

  Source: *https://www.cyberghostvpn.com*

  CyberGhost VPN hides the attacker's IP and replaces it with a selected IP, allowing him or her to surf anonymously and access blocked or censored content. It encrypts the connection and does not keep logs, thus securing data.

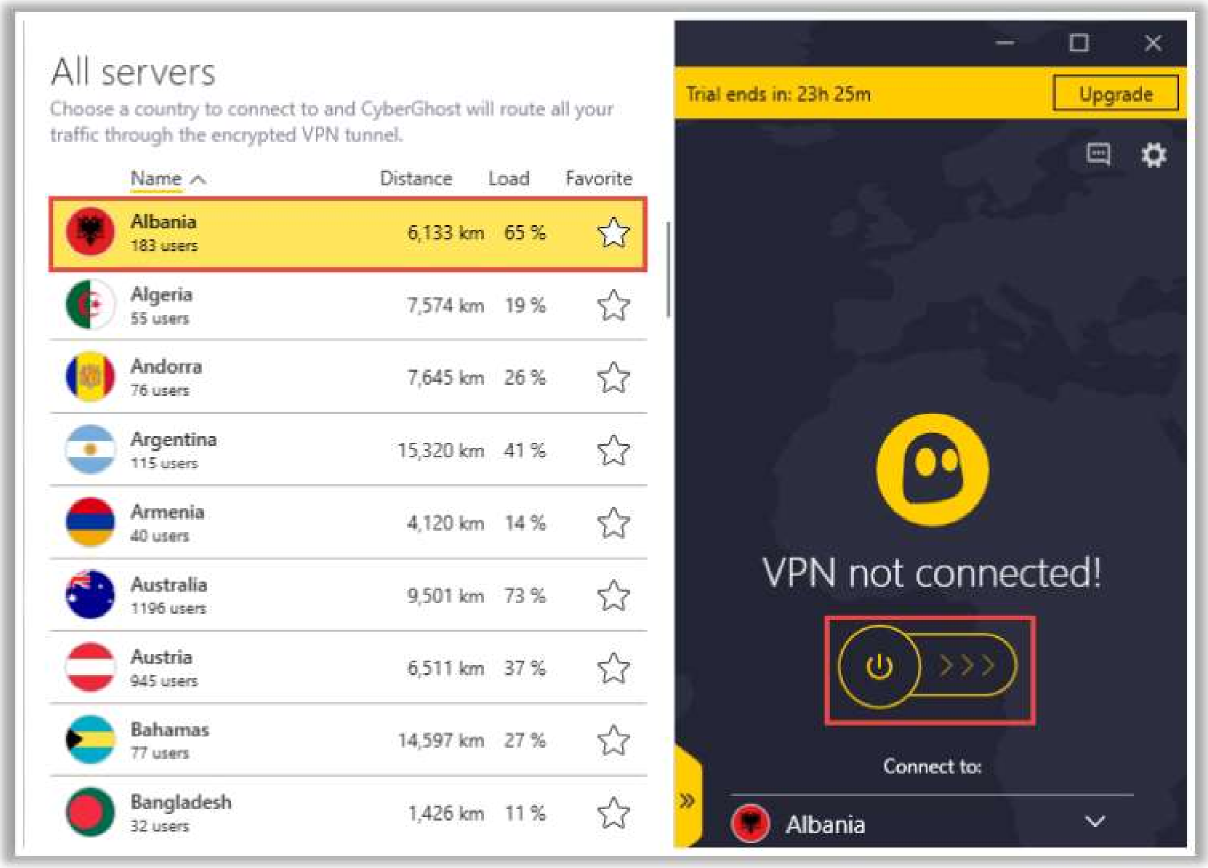
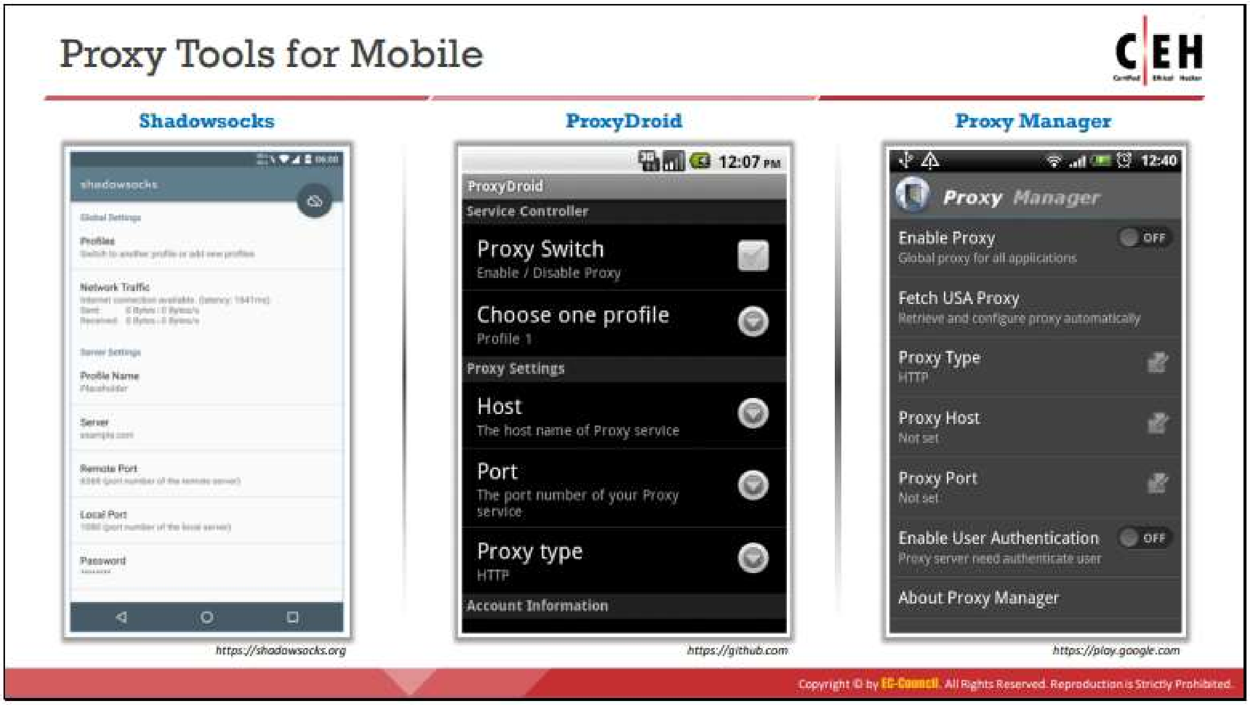


Figure 3.103: Screenshot of CyberGhost

In addition to the proxy tools mentioned above, there are many other proxy tools intended to allow users to surf the Internet anonymously. Some additional proxy tools are listed below:

- 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**

  Source: *https://shadowsocks.org*

  Shadowsocks is a high-performance, cross-platform secured socks5 proxy. It adopts bleeding-edge techniques with asynchronous I/O and event-driven programming. This tool is available on multiple platforms, including PC, MAC, mobile devices (Android and iOS), and routers (OpenWRT). It is a low-resource-consumption tool that is suitable for low-end boxes and embedded devices. It supports open-source implementations in python, node.js, golang, C#, and pure C.

  Shadowsocks help attackers to surf the Internet privately and securely.

Figure 3.104: Screenshot of Shadowsocks

- **ProxyDroid**

  Source: *https://github.com*

  ProxyDroid is an app that can help you to set the proxy (http/socks4/socks5) on your Android devices. It supports HTTP/HTTPS/SOCKS4/SOCKS5 proxy and also supports basic/NTLM/NTLMv2 authentication methods. Attackers can use this tool as a DNS proxy to access IP addresses that are beyond the firewalls.

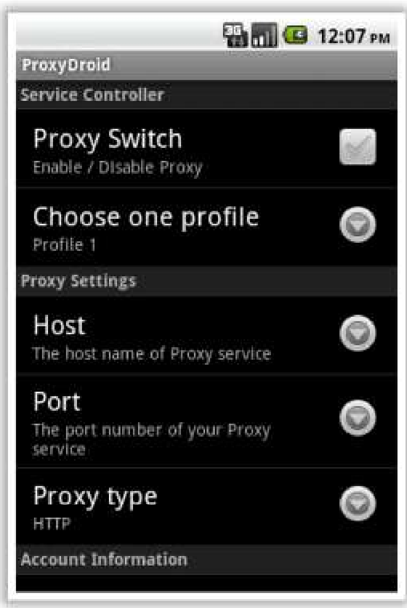


Figure 3.105: Screenshot of ProxyDroid

- **Proxy Manager**

  Source: *https://play.google.com*

  Proxy Manager is another Android-based proxy tool that supports HTTP/SOCKS4/SOCKS5 proxy and user authentication. It enables attackers to surf the Internet anonymously.

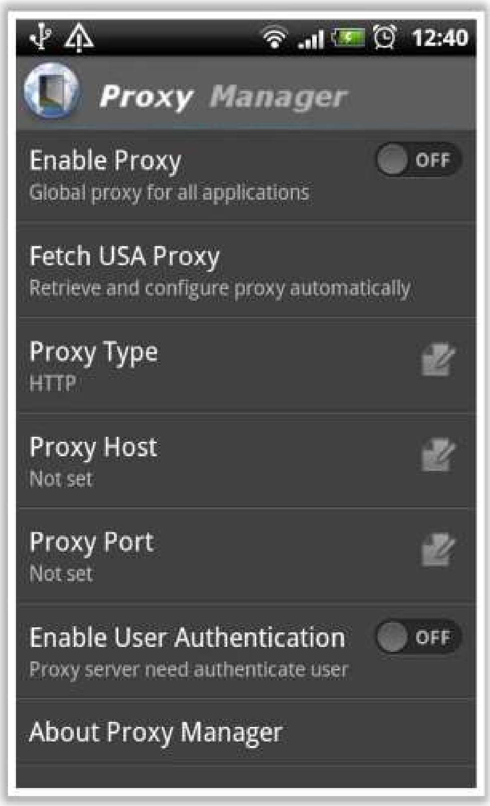


Figure 3.106: Screenshot of Proxy Manager

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**

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.

Why Use an Anonymizer?

The reasons for using anonymizers include:

- **Ensuring privacy**: Protect your identity by making your web navigation activities untraceable. Your privacy is maintained until and unless you disclose your personal information on the web, for example, by filling out forms.
- **Accessing government-restricted content**: Most governments prevent their citizens from accessing certain websites or content deemed inappropriate or sensitive. However, these sites can still be accessed using an anonymizer located outside the target country.

- **Protection against online attacks**: An anonymizer can protect you from all instances of online pharming attacks by routing all customer Internet traffic via its protected DNS server.
- **Bypassing IDS and firewall rules**: Firewalls are typically bypassed by employees or students accessing websites that they are not supposed to access. An anonymizer service gets around your organization's firewall by setting up a connection between your computer and the anonymizer service. Thus, firewalls see only the connection from your computer to the anonymizer's web address. The anonymizer will subsequently connect to any website (e.g., Twitter) with the help of an Internet connection and then direct the content back to you. To your organization, your system appears to be simply connected to the anonymizer's web address but not to the actual site that you are browsing.

In addition to protecting users' identities, anonymizers can also be used to attack a website without being traced.

Types of Anonymizers

An anonymizer is a service through which one can hide one's identity when using certain Internet services. It encrypts the data from your computer to the Internet service provider. Anonymizers are of two basic types: networked anonymizers and single-point anonymizers.

- **Networked Anonymizers**

  A networked anonymizer first transfers your information through a network of Internet-connected computers before passing it on to the website. Because the information passes through several Internet computers, it becomes cumbersome for anyone trying to track your information to establish the connection between you and the anonymizer.

  Example: If you want to visit any web page, you have to make a request. The request will first pass through A, B, and C Internet computers before going to the website.

  Advantage: Complication of the communications makes traffic analysis complex.

  Disadvantage: Any multi-node network communication incurs some degree of risk of compromising confidentiality at each node.

- **Single-Point Anonymizers**

  Single-point anonymizers first transfer your information through a website before sending it to the target website and then pass back the information gathered from the target website to you via the website to protect your identity.

  Advantage: Arms-length communication hides the IP address and related identifying information.

  Disadvantage: It offers less resistance to sophisticated traffic analysis.

Censorship Circumvention Tools

- **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.

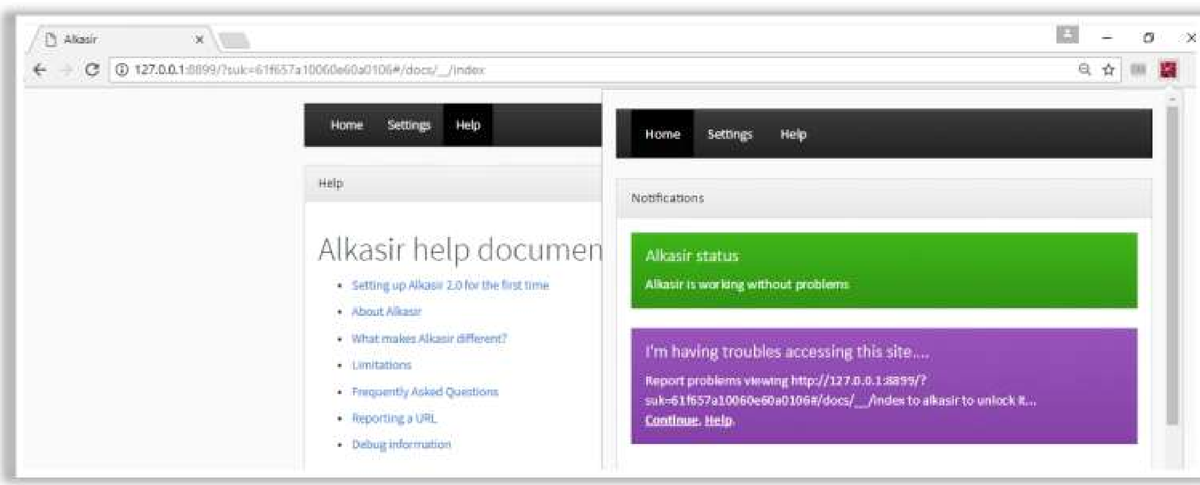


Figure 3.107: Screenshot of Alkasir

- **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.

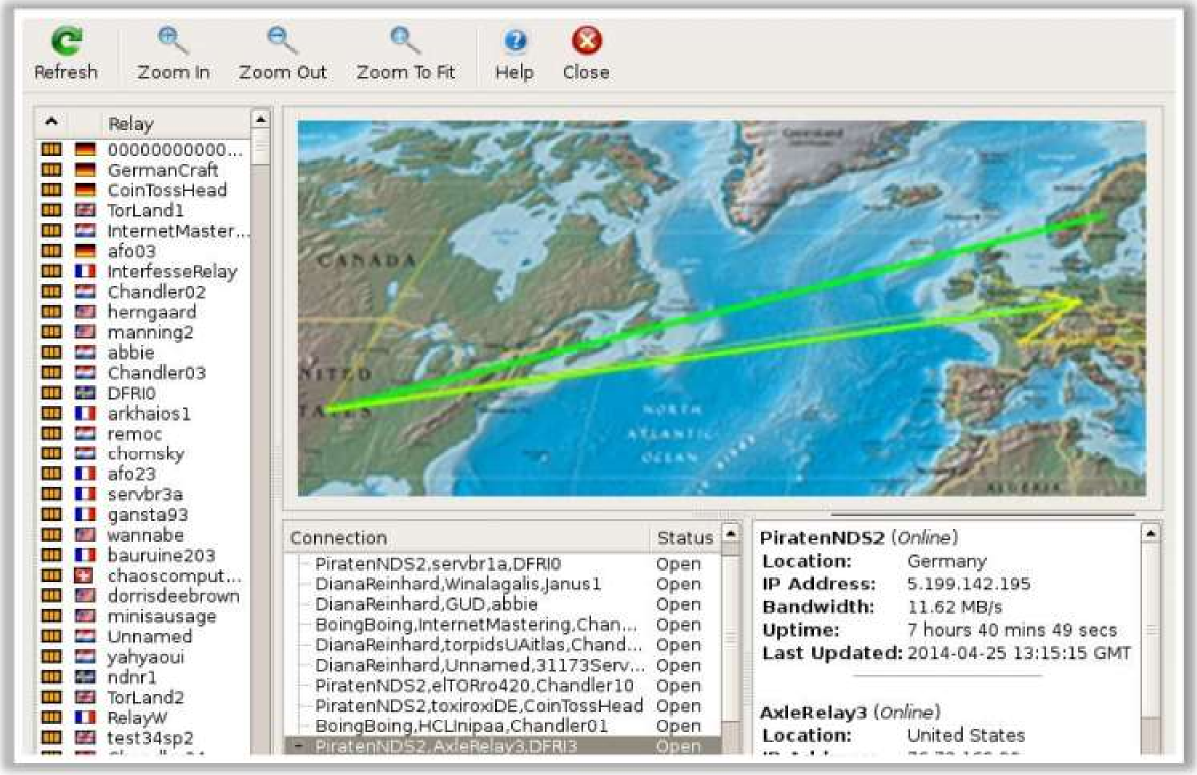
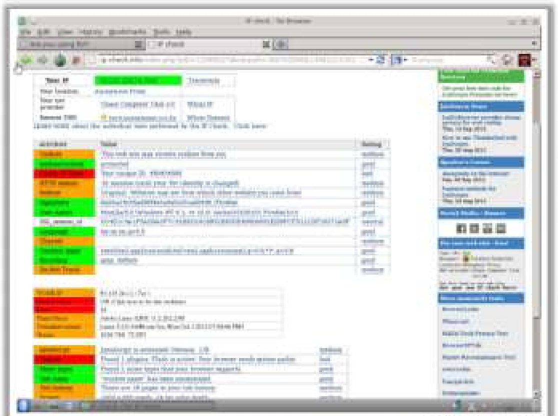


Figure 3.108: Screenshot of Tails

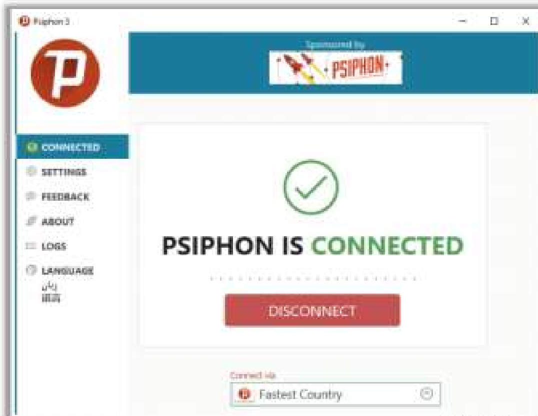
Anonymizers

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

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



https://www.whonix.org



https://psiphon.ca

Anonymizers

An anonymizer helps you to mask your IP address so that you can visit websites without being tracked or identified while keeping your activity and identity protected. It uses various techniques such as SSH, VPN, and HTTP proxies, which allow you to access blocked or censored content on the Internet with omitted advertisements.

- **Whonix**

  Source: *https://www.whonix.org*

  Whonix is a desktop OS designed for advanced security and privacy. It mitigates the threat of common attack vectors while maintaining usability. Online anonymity is realized via fail-safe, automatic, and desktop-wide use of the Tor network. It consists of a heavily reconfigured Debian base that is run inside multiple virtual machines, providing a substantial layer of protection from malware and IP address leaks.

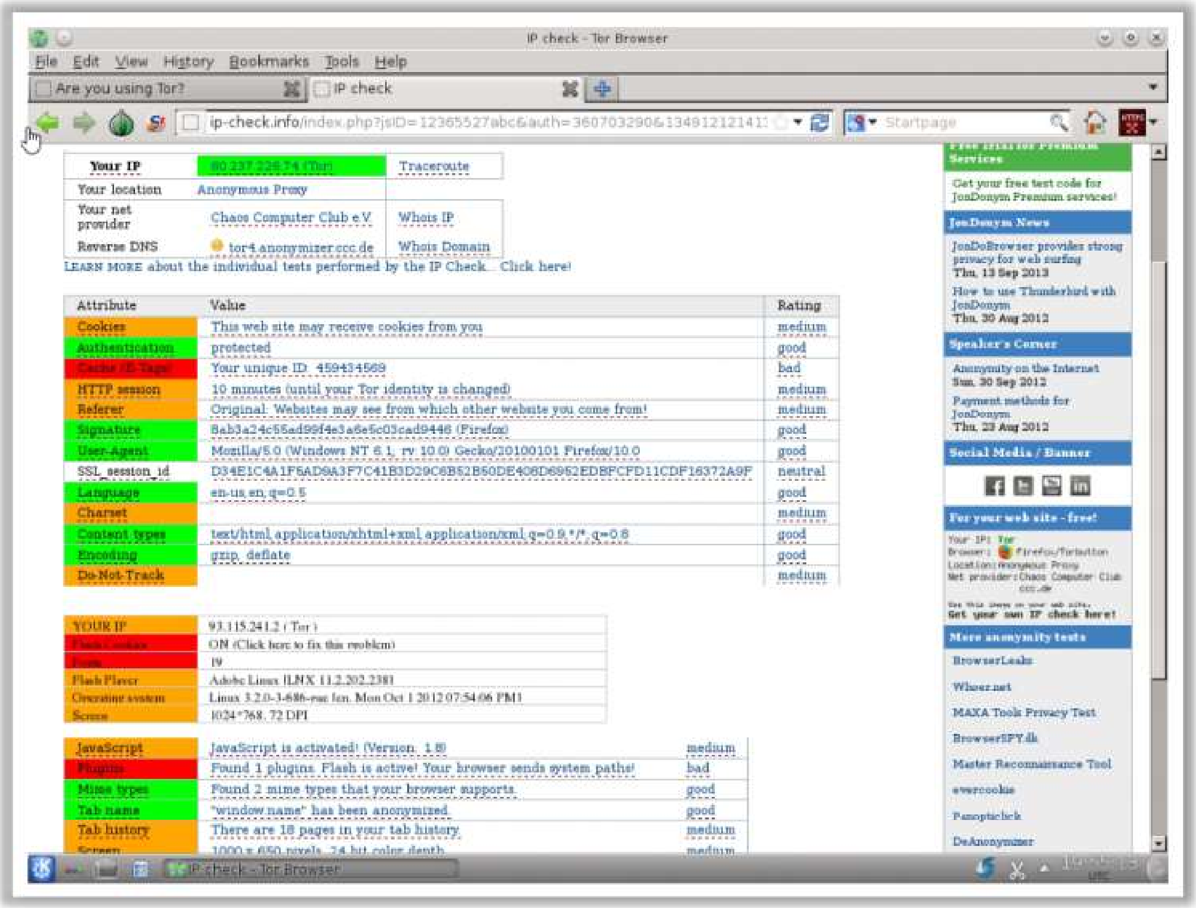


Figure 3.109: Screenshot of Whonix

- **Psiphon**

  Source: *https://psiphon.ca*

  Psiphon is an open-source anonymizer software that allows attackers to surf the Internet through a secure proxy. After installation, it will automatically configure the Windows machine's proxy configurations in such a way that the network traffic for the web applications and browsers that operate through these configurations will be tunneled through Psiphon.

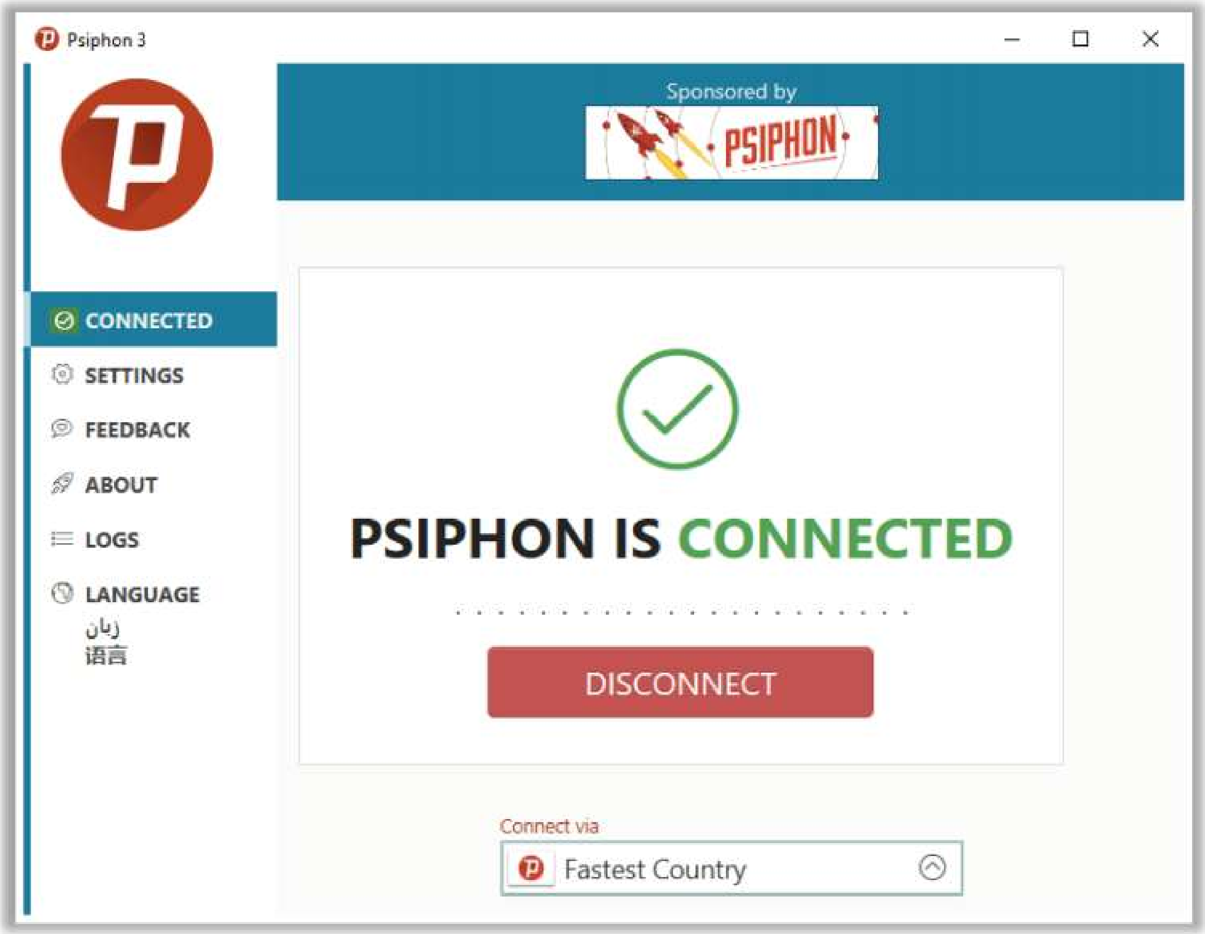


Figure 3.110: Screenshot of Psiphon

Anonymizers for Mobile

- **Orbot**

  Source: *https://guardianproject.info*

  Orbot is a proxy app that allows other apps to use the Internet more securely. It uses Tor to encrypt Internet traffic and then hides it by bouncing through a series of computers around the world. Tor is a free software that provides an open network to help defend your system against any form of network surveillance that may compromise personal freedom and privacy as well as confidential business activities and relationships through a type of state security monitoring known as "traffic analysis." Orbot creates a truly private Internet connection.

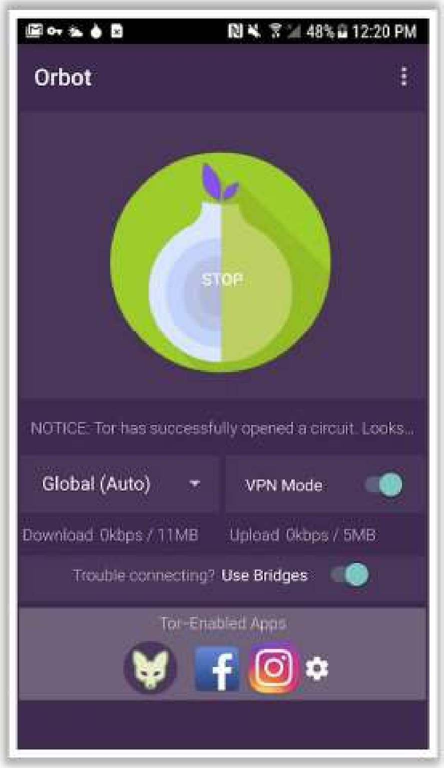


Figure 3.111: Screenshot of Orbot

- **Psiphon**

  Source: *https://psiphon.ca*

  Psiphon is a circumvention tool developed by Psiphon, Inc., which uses VPN, SSH, and HTTP proxy technology to provide you with open and uncensored access to Internet content. However, Psiphon does not increase online privacy and is not an online security tool.

  Features:

  - **Browser** or **VPN** (whole-device) **mode**: one can choose whether to tunnel everything or just the web browser.
  - **In-app stats**: This lets you know how much traffic you have been using.

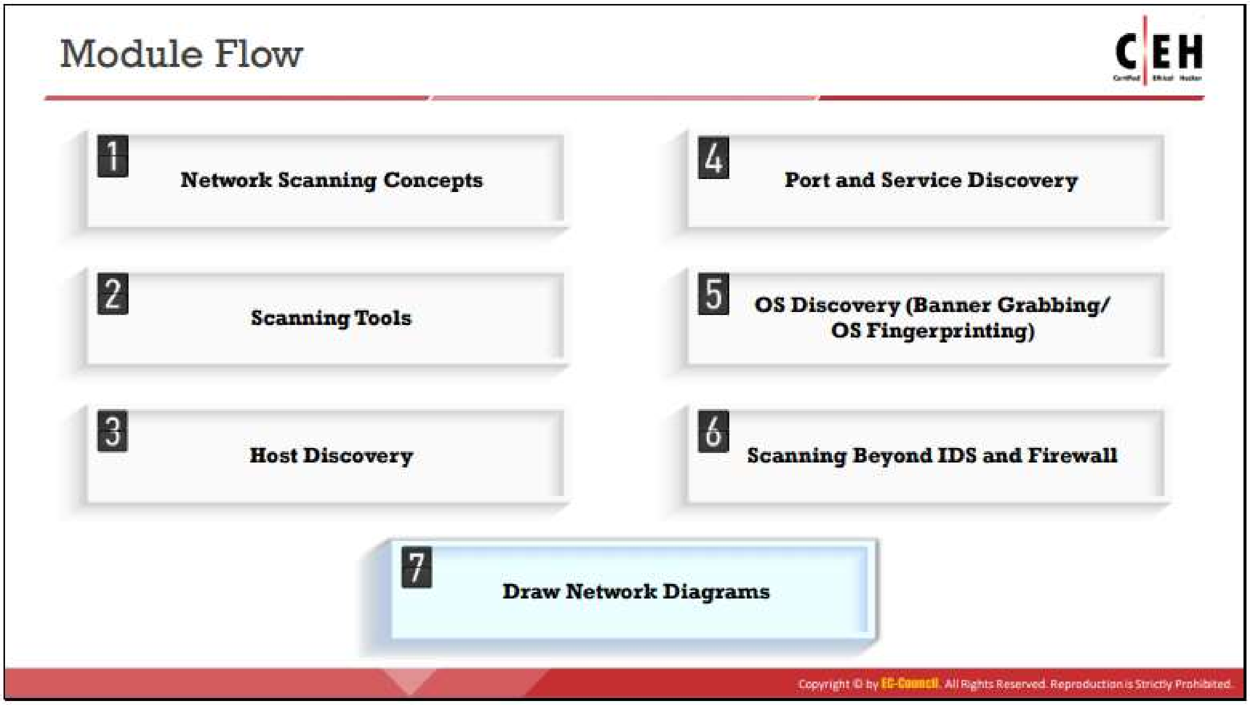
Figure 3.112: Screenshot of Psiphon

- **OpenDoor**

  Source: *https://www.apple.com*

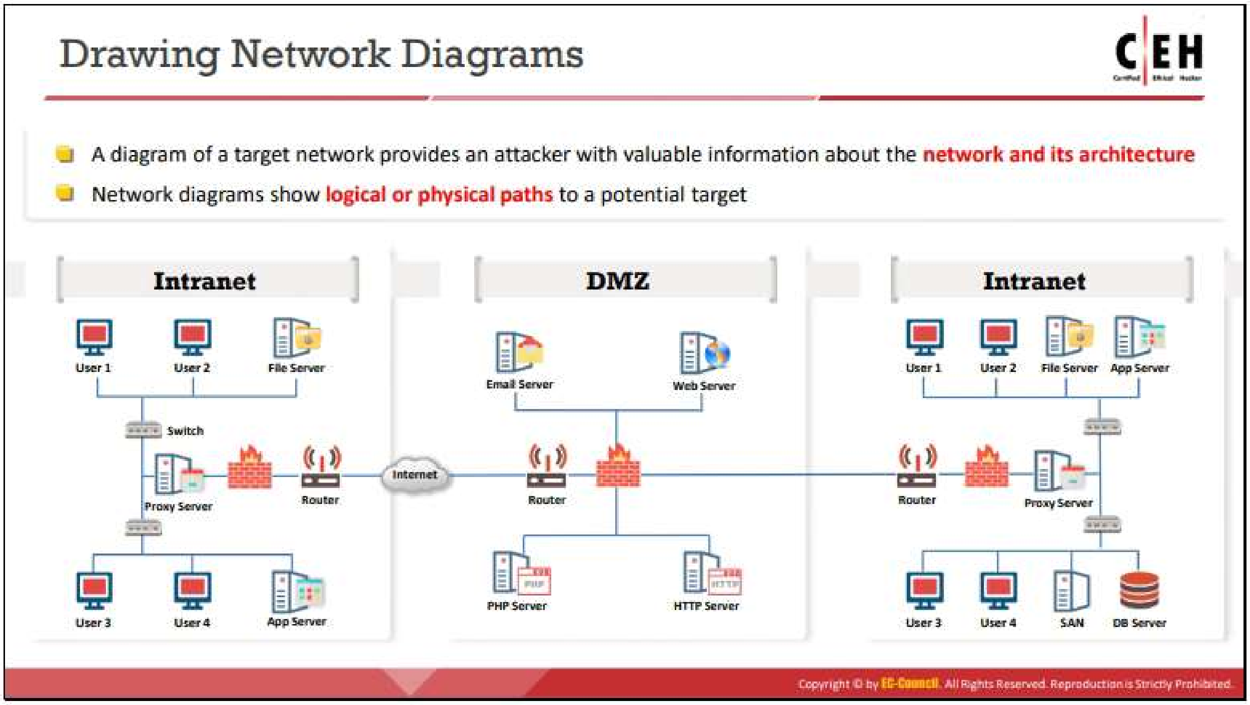
  OpenDoor is an app designed for both iPhone and iPad; it allows attackers to browse websites smoothly and anonymously.

Figure 3.113: Screenshot of OpenDoor



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

Drawing a network diagram helps an attacker to identify the topology or architecture of a target network. The network diagram also helps to trace the path to the target host in the network and enables the attacker to understand the positions of firewalls, IDS, routers, and other access control devices. Once the attacker has this information, he/she can try to find the vulnerabilities or weak points in these security mechanisms. Then, the attacker can exploit these weaknesses to find his/her way into the victim's network.

The network diagram also helps network administrators to manage their networks. Attackers use network discovery or mapping tools to draw network diagrams of target networks. An example of a network diagram is shown below.

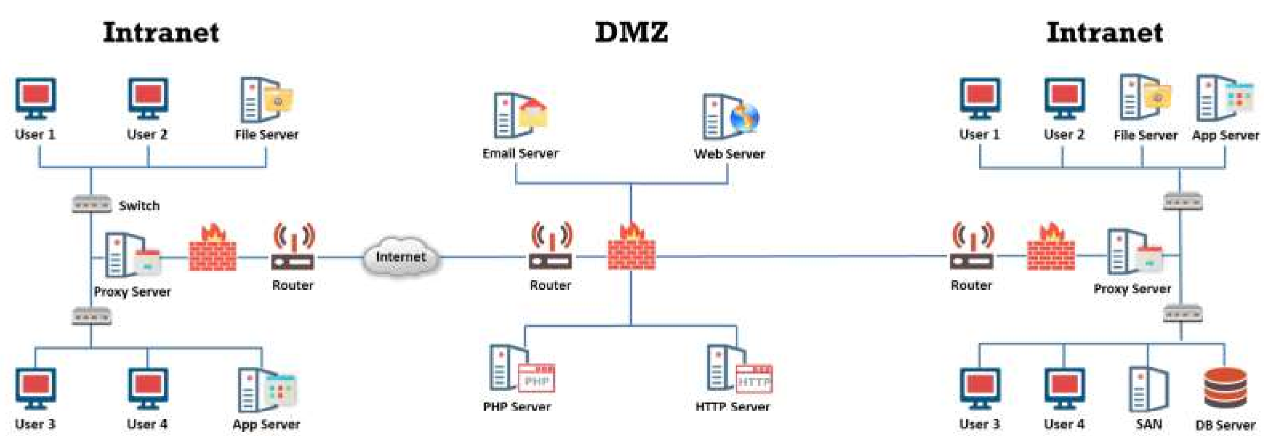
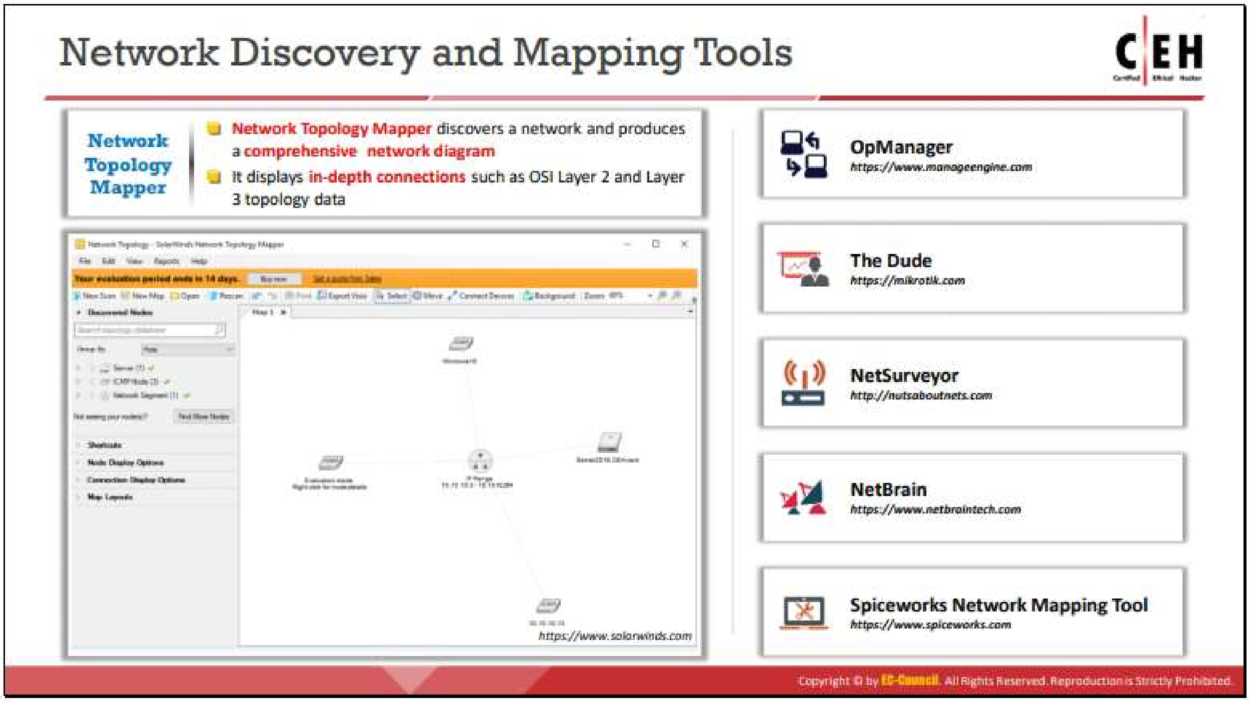


Figure 3.114: Example of Network Diagram



Network Discovery and Mapping Tools

Network discovery and mapping tools allow you to view the map of your network. They help you to detect rogue hardware and software violations and notify you whenever a particular host becomes active or goes down. Thus, you can also determine server outages or problems related to performance. An attacker can use the same tools to draw a diagram of the target network, analyze the topology, find the vulnerabilities or weak points, and launch an attack by exploiting these weak points.

- **Network Topology Mapper**

  Source: *https://www.solarwinds.com*

  The Network Topology Mapper tool allows one to automatically discover and create a network map of the target network. It can also display in-depth connections such as OSI Layer 2 and Layer 3 topology data (e.g., switch-to-switch, switch-to-node, and switch-to-router connections). It can keep track of network changes and allow the user to perform inventory management of hardware and software assets.

  Features:

  - **Network topology discovery and mapping**

    Automatically discovers the entire network and creates comprehensive and detailed network maps

  - **Export network diagrams to Visio**

    Exports network diagrams to Microsoft Office® Visio®, Orion Network Atlas, PDF, and PNG formats

- **Network mapping for regulatory compliance**

  Allows one to directly address PCI compliance and other regulations that require maintenance of an up-to-date network diagram

- **Multi-level network discovery**

  Performs multi-level network discovery to produce an integrated OSI Layer 2 and Layer 3 network map that includes detailed device information

- **Auto-detection of changes to network topology**

  Automatically detects new devices and changes to a network topology with scheduled network scanning

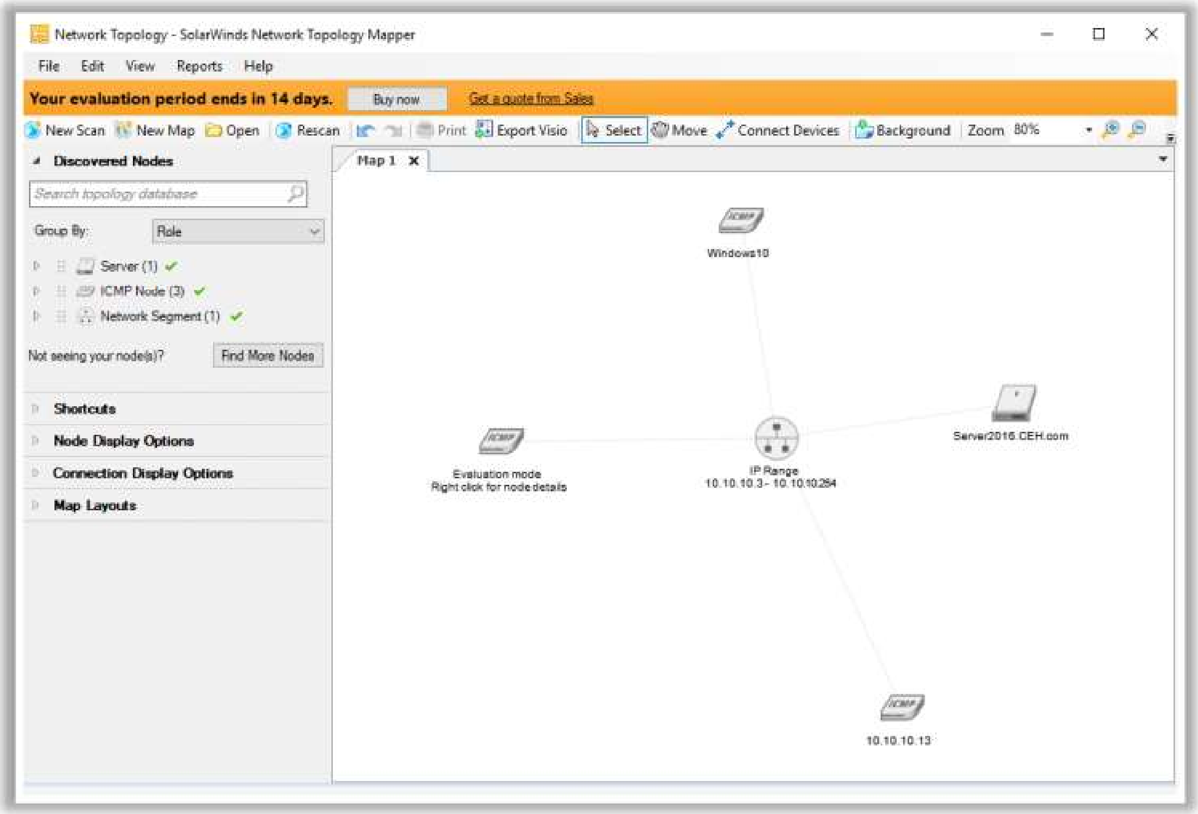
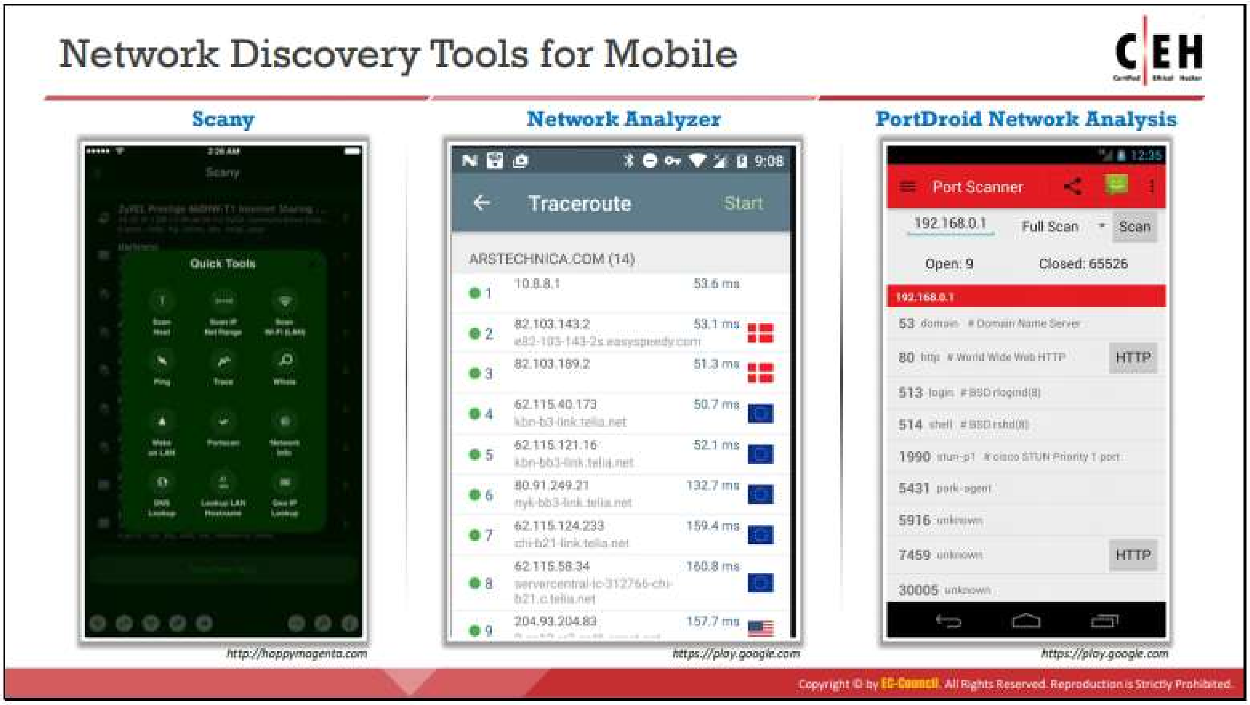


Figure 3.115: Screenshot of SolarWinds Network Topology Mapper

Some network discovery and mapping tools that an attacker can use to create a network map are listed below:

- OpManager (*https://www.manageengine.com*)
- The Dude (*https://www.mikrotik.com*)
- NetSurveyor (*http://nutsaboutnets.com*)
- NetBrain (*https://www.netbraintech.com*)
- Spiceworks Network Mapping Tool (*https://www.spiceworks.com*)



Network Discovery Tools for Mobile

Some network discovery tools for mobile devices are as follows:

- **Scany**

  Source: *http://happymagenta.com*

  Scany, a network scanner app for iPhone and iPad, scans LAN, Wi-Fi networks, websites, and open ports, discovers network devices, and digs network info. It supports several networking protocols and anti-stealth technologies. It is a multifunctional networking instrument for finding connected devices, looking up detailed device information, network troubleshooting, scanning ports, and testing network security and firewalls.

  Attackers use this tool to scan both the LAN and the Internet, scan any IP address or network range, perform hostname, device name, MAC address, and hardware vendor lookups, ping/trace hosts with integrated tools and WHOIS hostnames, IP addresses, ASNs, etc.

Figure 3.116: Screenshot of Scany

- **Network Analyzer**

  Source: *https://play.google.com*

  Network Analyzer can diagnose various problems in the Wi-Fi network setup or Internet connectivity, and it can also detect various issues in remote servers based on its wide range of in-built tools. Attackers can use it to perform ping, traceroute, port scanning, Whois, and DNS lookup activities.

Figure 3.117: Screenshot of Network Analyzer

- **PortDroid Network Analysis**

  Source: *https://play.google.com*

  Attackers can use PortDroid Network Analysis to perform local network discovery. It is also effective in analyzing the network and performing port scanning as well as banner grabbing using certain protocols, including ssh, telnet, http, https, ftp, smb, etc.

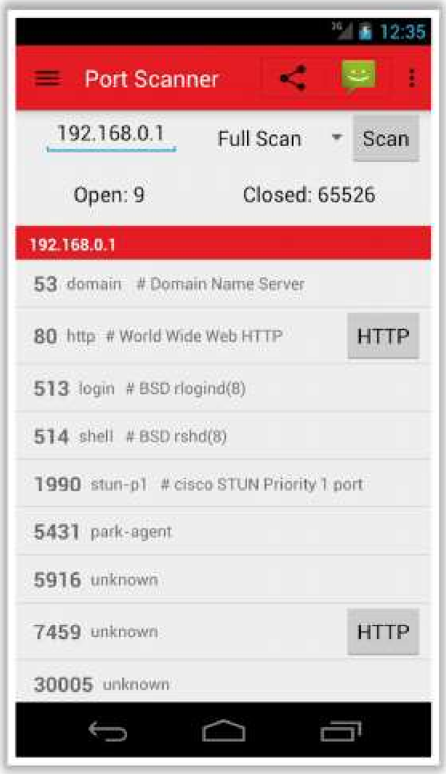


Figure 3.118: Screenshot of Network Analyzer

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

Module Summary

This module discussed how attackers determine live hosts from a range of IP addresses by sending various ping scan requests to multiple hosts. It also described how attackers perform different scanning techniques to determine open ports, services, service versions, etc., on the target system. Furthermore, it explained how attackers perform banner grabbing or OS fingerprinting to determine the OS running on a remote target system. It also illustrated various scanning techniques that attackers can adopt to bypass IDS/firewall rules and logging mechanisms and hide themselves as usual under network traffic. Finally, it ended with a detailed discussion on drawing the target's network diagram and its significance in providing valuable information about the 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.