

Chapter 03: Scanning Networks

Introduction

Cybercriminals are always searching for vulnerabilities, and even a small gap can be enough for them to strike. Network scanning is your first line of defense, helping identify active devices and how they interact with internal and external systems. By uncovering hidden vulnerabilities through specialized network protocols, it provides the insights needed to strengthen your defenses and keep your network secure.

In this chapter, you will explore:

- Network scanning concepts and their role in identifying vulnerabilities
- Key scanning tools for assessing network security
- Techniques for host discovery and finding active devices
- Methods for port and service discovery to detect open ports and services
- OS discovery using banner grabbing and OS fingerprinting
- Approaches for scanning beyond IDS and firewalls
- Network scanning countermeasures to protect against unauthorized scans

Network Scanning Concepts

The Scanning Network phase includes probing the target network to get information. When a user probes another user, the received reply can reveal very useful information. A comprehensive analysis of networks, ports, and active services contributes to the development of a network architecture, providing the attacker with a more detailed understanding of the target.

An Overview of Network Scanning

Scanning involves the collection of comprehensive and intricate information regarding a target through advanced and often aggressive reconnaissance methods. Network scanning involves a set of processes designed to detect hosts, ports, and services present within a network. This technique is also employed to detect active devices on a network and ascertain the operating system in use on the target machine. It represents a critical phase of information gathering for an attacker, facilitating the creation of a detailed profile of the target organization. During the scanning process, the attacker seeks to collect data such as accessible IP addresses within the network, the operating system and system architecture of the target, as well as the ports and their corresponding services operating on each device.

The objective of scanning is to uncover exploitable communication channels, assess as many active listeners as possible, and identify those that are responsive or beneficial to the attacker's specific requirements. In this phase of an attack, the attacker aims to explore various methods for breaching the target system. Additionally, the attacker attempt to gather further insights into the target system to identify any potential configuration weaknesses. The information acquired is subsequently utilized to formulate an effective attack strategy.

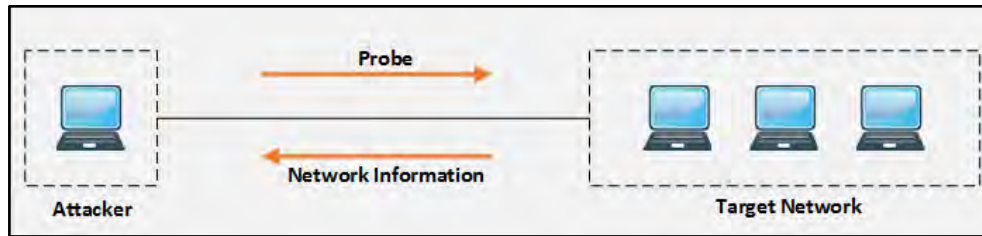


Figure 3-01: Network Scanning Process

Types of Scanning

There are several types of scanning techniques used in network security and penetration testing to gather information and identify vulnerabilities. Some of them include:

- **Port Scanning:** This involves examining the open ports and the services operational on a designated system. Port scanning, through the examination of TCP and UDP ports, assists in identifying whether services are actively listening, thereby disclosing details regarding the operating system and applications currently in operation. Active services that are listening may expose the system to unauthorized access or misconfigurations, potentially leading to vulnerabilities.
- **Network Scanning:** This process involves detecting active hosts and IP addresses within a network. It may be conducted with malicious intent or to evaluate the network's security.
- **Vulnerability Scanning:** This method identifies known weaknesses in a system, checking for exploitable vulnerabilities. A vulnerability scanner uses a scanning engine and a catalog of known vulnerabilities and exploits for various servers. The scanner may look for issues like backup files or directory traversal exploits. It helps ensure the security of a server by detecting common flaws that can typically be fixed through updated patches or proper configuration.

Objectives of Network Scanning

There are some key objectives of network scanning involved:

- Discover live hosts, IP addresses, and open ports, which helps an attacker identify potential entry points based on open services
- Identify the OS and system architecture, also known as fingerprinting, which can guide attackers in exploiting OS-specific vulnerabilities
- Identify running services, allowing attackers to pinpoint potential weaknesses in those services that can be exploited
- Find specific applications or service versions, helping attackers identify vulnerabilities tied to specific software versions
- Identify network vulnerabilities, as scanning reveals exploitable flaws that can be used to gain unauthorized access
- Map the network topology to understand the devices, routers, switches, and their connections, assisting attackers in planning their approach to the network's defenses

TCP Communication Flags

The TCP header contains numerous flags that control data transmission across a TCP connection. Six TCP control flags control communication between hosts and provide guidance to the system. Four of these flags control the setup, maintenance, and termination of a connection, while the other two flags give instructions to the system. Each flag is 1 bit in size. The TCP Flags section comprises six flags, resulting in a total size of 6. When the flag value is set to 1, it will be automatically turned on. The TCP header includes:

Flag	Use
SYN	Initiates a connection between two hosts to facilitate communication
ACK	Acknowledges the receipt of a packet
URG	Indicates that the data contained in the packet is urgent and should be processed immediately
PSH	Instructs the sending system to send all buffered data immediately
FIN	Informs the remote system when communication ends. In essence, this gracefully closes a connection.
RST	Resets a connection

Table 3-01: TCP Flags

TCP Communication

Internet Protocol (IP) traffic is primarily classified into two types: Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP operates on a connection-oriented basis, facilitating bidirectional communication following the successful establishment of a connection. UDP, on the other hand, is a simpler protocol that operates without establishing a connection. It transmits multiple messages as packets in segments. Unlike TCP, UDP does not incorporate features for reliability, flow control, or error recovery within its IP packets. Due to its simplicity, UDP headers are smaller in size, resulting in reduced network overhead compared to TCP.

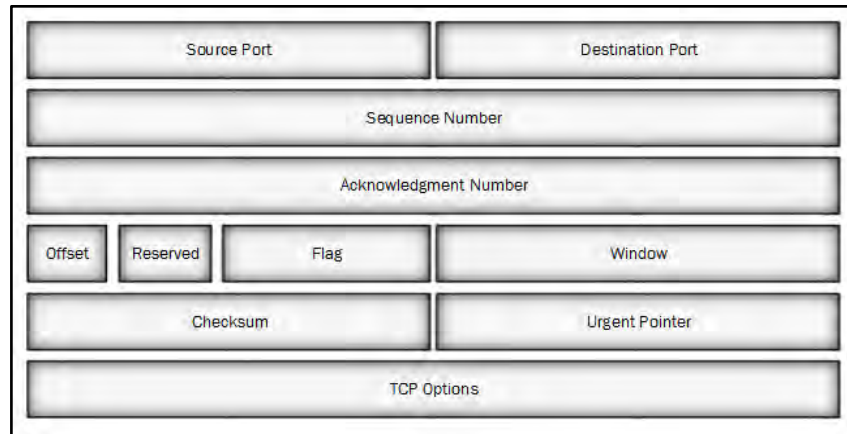


Figure 3-02: TCP Header

The process of establishing a TCP connection between hosts requires a three-step handshake. This handshake ensures successful, reliable, and connection-oriented sessions between hosts. The process of establishing a TCP connection includes three steps, as shown in Figure 3-03.



Figure 3-03: TCP Connection Handshake

When host A seeks to establish communication with host B, it initiates a TCP connection by transmitting a SYN packet to host B. Upon receiving the SYN packet, host B transmits a SYN+ACK packet back to host A. Subsequently, host A acknowledges this by sending an ACK packet upon receipt of the SYN+ACK packet from host B. A successful handshake results in the establishment of a TCP connection.



EXAM TIP: By mapping the network topology and analyzing TCP communication flags, attackers can gain a comprehensive understanding of the target network. This insight allows for a more effective and targeted attack strategy based on the network's architecture and weaknesses.

The U.S. Department of Defense introduced the TCP/IP model by integrating elements from the OSI Layer Model and its framework. The TCP and the IP are fundamental standards that form the foundation of internet connectivity. IP specifies the methodology for data exchange between computers over a network of interconnected routes, while TCP outlines the process for applications to establish dependable communication channels within that network. IP is responsible for addressing and routing, whereas TCP ensures that communication remains coherent and data integrity is maintained throughout the exchange.

Scanning Tools

Scanning tools play a crucial role in detecting active hosts, open ports, and services in operation, as well as in collecting diverse information regarding a target network. These tools provide critical data such as location details, NetBIOS information, and information about all TCP/IP and UDP open ports. The insights gained from these tools enable ethical hackers to build a detailed profile of the target organization and its network. By scanning the network for open ports on connected devices, ethical hackers can uncover potential vulnerabilities, assess the network's security posture, and identify entry points for further testing.



EXAM TIP: Tools like Nmap, Hping3, and Metasploit help gather detailed network information, such as open ports, services, and OS details. This data is critical for understanding vulnerabilities and planning penetration testing efforts.

Nmap

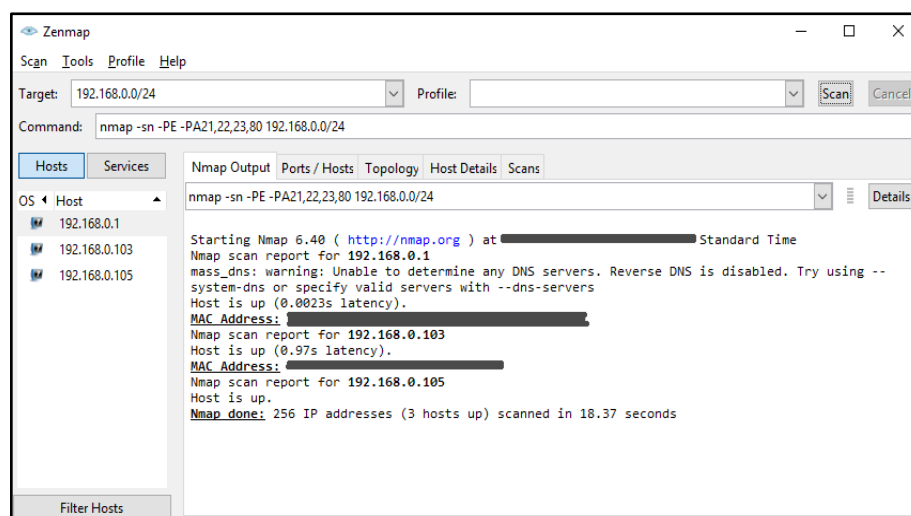
Another way to ping a host is to utilize Nmap to conduct a ping operation. To execute the following command, please use the command prompt available in either Windows or Linux.

```
nmap -sP -v <target IP address>
```

If the command successfully detects an active host and receives a response from the designated host, it will return a message confirming that the targeted host's IP address, Media Access Control (MAC) address, and network interface card vendor are operational.

In addition to ICMP echo request packets and ping sweeps, Nmap provides a quick scan option. To initiate a quick scan, please enter the following command:

```
nmap -sP -PE -PA<port numbers> <starting IP/ending IP>
```



Nmap, in a nutshell, offers host discovery, port discovery, service discovery, operating system version information, hardware address (MAC) information, service version detection, vulnerabilities, and exploit detection using the Nmap Scripting Engine (NSE).

Note: The Nmap Scripting engine is the most powerful engine for network discovery, version detection, vulnerability detection, and backdoor detection.

Hping3

Hping3 is a multifunctional command-line utility intended for network scanning and packet crafting within the TCP/IP protocol suite. It accommodates a range of protocols, including TCP, UDP, ICMP, and raw IP, rendering it an effective tool for activities such as network security assessments, firewall evaluations, manual path MTU discovery, advanced traceroute operations, remote operating system fingerprinting, uptime estimation, and TCP/IP stack evaluations. Hping3 transmits customized TCP/IP packets and analyzes responses from targets in a way similar to a conventional ping program that utilizes ICMP replies.

The tool is capable of handling packet fragmentation, arbitrary packet body customization, and size adjustments, allowing for the transfer of encapsulated files over supported protocols. It facilitates the scanning of idle hosts, the practice of IP spoofing, and the anonymous probing of services, thereby allowing for discreet network and host scanning. Hping3's Traceroute mode facilitates file transfers over covert channels and can verify host availability even when ICMP packets are blocked. Its firewall-like capabilities allow it to discover open ports behind firewalls. Additionally, it performs manual path MTU discovery and enables remote operating system fingerprinting, making it a comprehensive tool for both legitimate network diagnostics and potential exploitation activities.

Using Hping, an attacker can analyze the behavior of an idle host to gather valuable information about the target system. This includes details about the host's services, the ports supporting those services, and the target's operating system. Such scans often serve as precursors to more intensive probing or potential attacks.

The general syntax for conducting scans with Hping is as follows:

```
hping3 <options> <Target IP address>
```

Hping Scan with AI

Attackers can use AI-driven technologies to improve and automate their network scanning activities. By employing AI, they can efficiently conduct network scans with the Hping3 tool, thereby gaining critical information regarding their intended targets.

Example #1: ICMP Scanning with Hping3

An individual with malicious intent may use ChatGPT to accomplish this objective by employing a suitable prompt, such as:

"Use Hping3 to perform an ICMP scan on the target IP address 10.10.1.11 and stop after 10 iterations."

```

sgpt --chat scan --shell "Use Hping3 to perform ICMP scanning on the target IP address 10.
File Edit View Search Terminal Help
[root@parrot]~#
#sgpt --chat scan --shell "Use Hping3 to perform ICMP scanning on
the target IP address 10.10.1.11 and stop after 10 iterations"
hping3 --icmp --count 10 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
HPING 10.10.1.11 (eth0 10.10.1.11): icmp mode set, 28 headers + 0 data
bytes
len=28 ip=10.10.1.11 ttl=128 id=35726 icmp_seq=0 rtt=3.2 ms
len=28 ip=10.10.1.11 ttl=128 id=35727 icmp_seq=1 rtt=3.1 ms
len=28 ip=10.10.1.11 ttl=128 id=35728 icmp_seq=2 rtt=2.9 ms
len=28 ip=10.10.1.11 ttl=128 id=35729 icmp_seq=3 rtt=2.7 ms
len=28 ip=10.10.1.11 ttl=128 id=35730 icmp_seq=4 rtt=2.5 ms
len=28 ip=10.10.1.11 ttl=128 id=35731 icmp_seq=5 rtt=2.4 ms
len=28 ip=10.10.1.11 ttl=128 id=35732 icmp_seq=6 rtt=2.2 ms
len=28 ip=10.10.1.11 ttl=128 id=35733 icmp_seq=7 rtt=2.0 ms
len=28 ip=10.10.1.11 ttl=128 id=35734 icmp_seq=8 rtt=1.8 ms
len=28 ip=10.10.1.11 ttl=128 id=35735 icmp_seq=9 rtt=1.8 ms

--- 10.10.1.11 hping statistic ---
10 packets transmitted, 10 packets received, 0% packet loss
round-trip min/avg/max = 1.8/2.5/3.2 ms
  
```

Figure 3-04: Use Hping3 to Perform ICMP Scanning on the Target IP

hping3 --icmp --count 10 10.10.1.11

This table offers a detailed explanation of each option used in the command.

Command	Description
hping3	Invokes the Hping3 tool, a network scanning utility.
--icmp	Specifies that ICMP packets will be sent. ICMP packets are commonly used for network troubleshooting and diagnostics.
--count 10	Specifies that only 10 ICMP packets will be sent to the target IP address.
10.10.1.11	The target IP address for the ICMP packets.

Example #2: TCP ACK Scan with Hping3

An attacker has the ability to execute this task through ChatGPT by utilizing a suitable prompt, such as:

"Run a Hping3 ACK scan on port 80 of target IP 10.10.1.11."

```
[attacker@parrot]~$ sudo hping3 --ack -p 80 10.10.1.11
[Execute, [Describe, [Abort]: E
HPING 10.10.1.11 (eth0 10.10.1.11): A set, 40 headers + 0 data bytes
len=40 ip=10.10.1.11 ttl=128 DF id=50 sport=80 flags=R seq=0 win=0 rtt=6.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=51 sport=80 flags=R seq=1 win=0 rtt=6.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=52 sport=80 flags=R seq=2 win=0 rtt=6.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=53 sport=80 flags=R seq=3 win=0 rtt=2.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=54 sport=80 flags=R seq=4 win=0 rtt=2.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=55 sport=80 flags=R seq=5 win=0 rtt=6.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=56 sport=80 flags=R seq=6 win=0 rtt=6.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=57 sport=80 flags=R seq=7 win=0 rtt=9.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=58 sport=80 flags=R seq=8 win=0 rtt=2.6 ms
len=40 ip=10.10.1.11 ttl=128 DF id=59 sport=80 flags=R seq=9 win=0 rtt=2.5 ms
len=40 ip=10.10.1.11 ttl=128 DF id=60 sport=80 flags=R seq=10 win=0 rtt=2.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=61 sport=80 flags=R seq=11 win=0 rtt=9.0 ms
len=40 ip=10.10.1.11 ttl=128 DF id=62 sport=80 flags=R seq=12 win=0 rtt=2.3 ms
len=40 ip=10.10.1.11 ttl=128 DF id=63 sport=80 flags=R seq=13 win=0 rtt=2.2 ms
len=40 ip=10.10.1.11 ttl=128 DF id=64 sport=80 flags=R seq=14 win=0 rtt=2.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=65 sport=80 flags=R seq=15 win=0 rtt=5.4 ms
len=40 ip=10.10.1.11 ttl=128 DF id=66 sport=80 flags=R seq=16 win=0 rtt=1.9 ms
len=40 ip=10.10.1.11 ttl=128 DF id=67 sport=80 flags=R seq=17 win=0 rtt=5.2 ms
len=40 ip=10.10.1.11 ttl=128 DF id=68 sport=80 flags=R seq=18 win=0 rtt=5.1 ms
len=40 ip=10.10.1.11 ttl=128 DF id=69 sport=80 flags=R seq=19 win=0 rtt=5.0 ms
```

Figure 3-05: Run a Hping3 ACK Scan on Port 80 of Target IP

```
sudo hping3 --ack -p 80 10.10.1.11
```

This table offers a detailed explanation of each option used in the command.

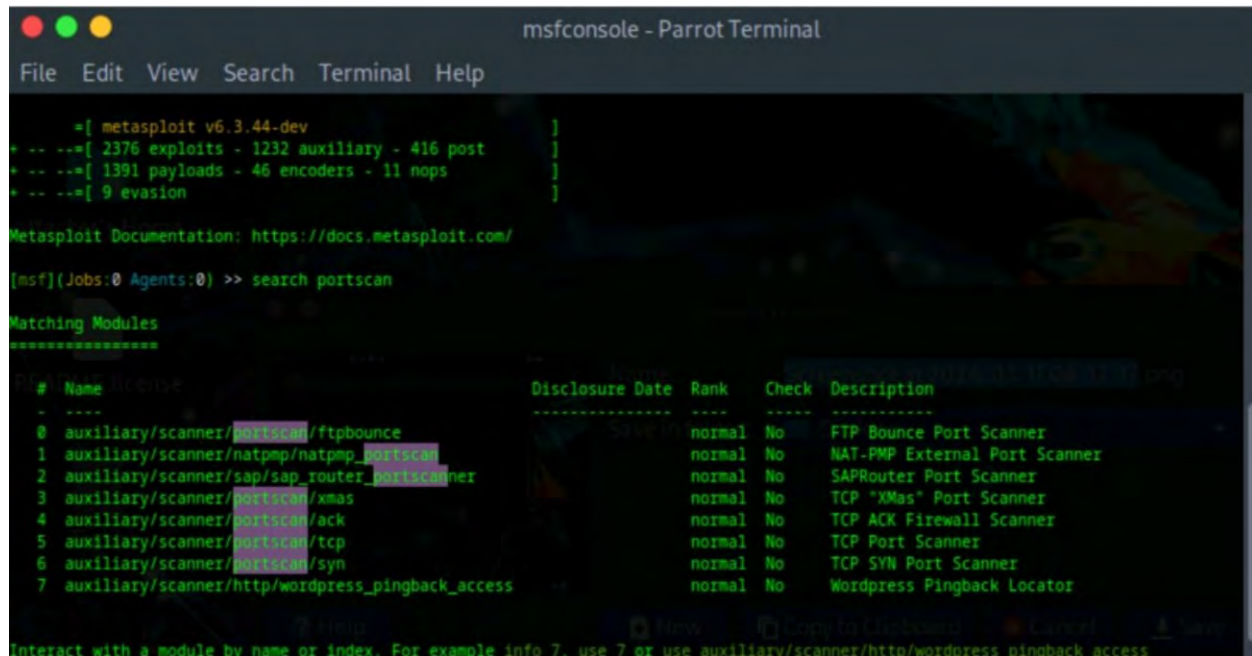
Command	Description
Sudo	Ensures the command is executed with administrative privileges.
Hping3	Invokes the Hping3 tool.
--ack	Specifies the TCP ACK scan mode. In this mode, Hping3 sends TCP packets with the ACK flag set.
-p 80	Specifies the destination port (in this case, port 80, commonly used for HTTP traffic).
10.10.1.11	The target IP address for the TCP ACK packets.

Metasploit

Metasploit is a widely used open-source project designed to provide the tools, infrastructure, and resources required for penetration testing and advanced security auditing. It serves as a valuable asset for identifying security vulnerabilities, aiding penetration testing efforts, and supporting IDS signature development. Metasploit simplifies the tasks of attackers, exploit developers, and payload creators with its versatile and user-friendly design.

One of Metasploit's standout features is its modular approach, which allows users to combine any exploit with any payload to achieve specific objectives. This flexibility makes it an essential tool for automating the discovery and exploitation process while also offering comprehensive support for

the manual testing phases of penetration tests. Metasploit Pro, the professional version, enhances its capabilities by enabling users to scan for open ports and services, exploit identified vulnerabilities, pivot through networks, gather evidence, and generate detailed test reports.



```

msfconsole - Parrot Terminal
File Edit View Search Terminal Help

=[ metasploit v6.3.44-dev ]
+ --=[ 2376 exploits - 1232 auxiliary - 416 post ]
+ --=[ 1391 payloads - 46 encoders - 11 nops ]
+ --=[ 9 evasion ]

Metasploit Documentation: https://docs.metasploit.com/

[msf](Jobs:0 Agents:0) >> search portscan

Matching Modules
=====
#  Name
-  -
0  auxiliary/scanner/portscan/ftpbounce
1  auxiliary/scanner/natpmp/natpmp_portscan
2  auxiliary/scanner/sap/sap_router_portscanner
3  auxiliary/scanner/portscan/xmas
4  auxiliary/scanner/portscan/ack
5  auxiliary/scanner/portscan/tcp
6  auxiliary/scanner/portscan/syn
7  auxiliary/scanner/http/wordpress_pingback_access

Disclosure Date  Rank  Check  Description
-----
0  normal No  FTP Bounce Port Scanner
1  normal No  NAT-PMP External Port Scanner
2  normal No  SAPRouter Port Scanner
3  normal No  TCP "XMas" Port Scanner
4  normal No  TCP ACK Firewall Scanner
5  normal No  TCP Port Scanner
6  normal No  TCP SYN Port Scanner
7  normal No  Wordpress Pingback Locator

Interact with a module by name or index. For example info 7, use 7 or use auxiliary/scanner/http/wordpress_pingback_access
  
```

Figure 3-06: Screenshot of Metasploit

Net ScanTools Pro

NetScanTools Pro is an all-encompassing investigative instrument tailored for troubleshooting, monitoring, discovering, and identifying devices within a network. It enables users to efficiently gather information about local LAN environments and Internet-connected systems, including details about IP addresses, ports, hostnames, domain names, email addresses, and URLs. This tool assists attackers in identifying vulnerabilities and exposed ports on target systems.

This tool provides both automatic and manual methods to list IPv4 and IPv6 addresses, domain details, and other relevant network information. It integrates a wide range of network utilities and tools, organized by functionality into categories such as active, passive, DNS, and local computer operations, making it a versatile tool for network analysis and reconnaissance.

Host Discovery

Host discovery scanning involves identifying systems that are 'alive' and responding on a network. It is an initial step in the network scanning process. This step is essential for conducting comprehensive scans to detect open ports and services while avoiding unnecessary scans on inactive systems. By determining which hosts are live, attackers or analysts can focus their efforts and streamline the scanning process.

Initially, you must know about the hosts that live in the targeted network. ICMP packets carry out the process of finding live hosts in a network. The target replies to ICMP echo packets with an ICMP echo reply. This response confirms that the host is currently active.

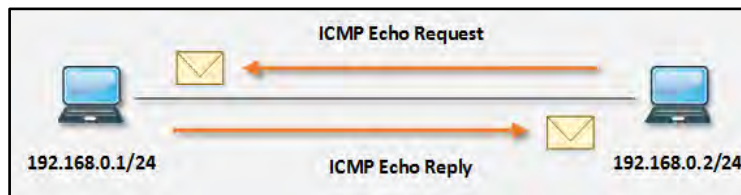
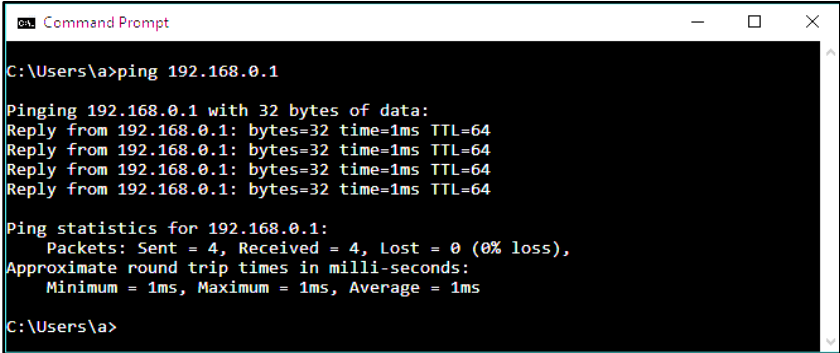


Figure 3-07: ICMP Echo Request & Reply Packets

The above figure shows that the host with IP address 192.168.0.2/24 is trying to identify whether the host 192.168.0.1/24 is live by sending the ICMP echo packets to the destination IP address 192.168.0.1.



```

C:\Users\A>ping 192.168.0.1

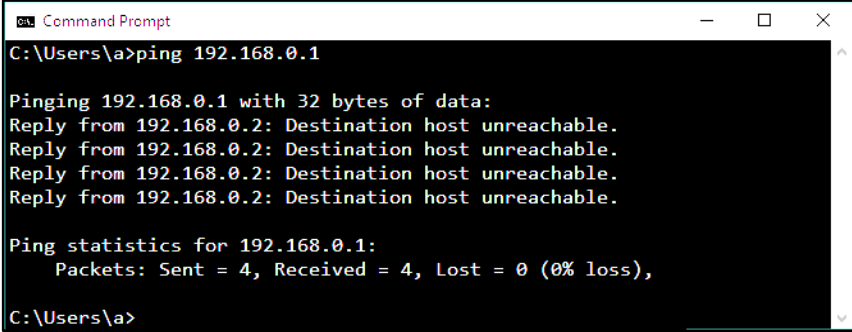
Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms

C:\Users\A>
```

Figure 3-08: ICMP Echo Reply Packets

The host is live if the destination host successfully responds to the ICMP echo packets. The following response of ICMP echo packets is observed when a destination host is down.



```

C:\Users\A>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:
Reply from 192.168.0.2: Destination host unreachable.
Reply from 192.168.0.2: Destination host unreachable.
Reply from 192.168.0.2: Destination host unreachable.
Reply from 192.168.0.2: Destination host unreachable.

Ping statistics for 192.168.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\A>
```

Figure 3-09: ICMP Echo Reply Packets

This section explores methods to identify live systems in a network using various ping scan techniques. It also covers ping sweep methods for detecting active hosts and discusses tools designed to perform efficient ping sweeps.

Host Discovery Techniques

Host discovery techniques are used to identify active or live hosts within a network. As an ethical hacker, it is essential to understand and utilize various types of host discovery methods to assess network activity effectively. In Figure 3-07, you have learned some commonly used host discovery techniques:

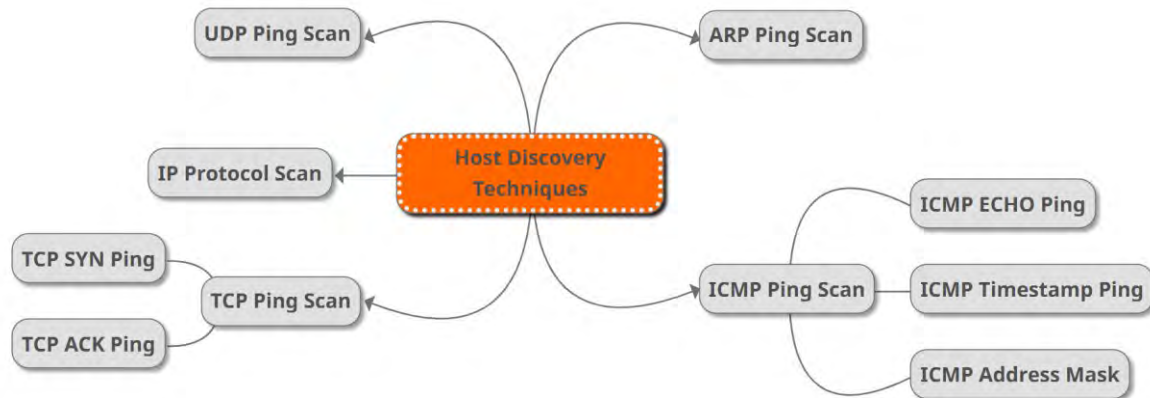


Figure 3-10: ARP Ping Scan

ARP Ping Scan

In an ARP ping scan, Address Resolution Protocol (ARP) packets are used to discover active devices within an IPv4 range, even in environments where firewalls attempt to hide their presence. This method is particularly effective in Local Area Networks (LANs) since ARP operates at the link layer, bypassing IP-layer firewall restrictions. Within most networks, a substantial number of IP addresses remain unused at any given time, especially in private address ranges. When an IP packet, like an ICMP echo request, is transmitted to a destination, the operating system is required to translate the target IP address into the appropriate hardware (MAC) address in order to format the Ethernet frame properly. This resolution process involves issuing a series of ARP requests.

An ARP scan reveals the MAC address of the network interface on the target device and can also identify MAC addresses of devices sharing the same IPv4 address within a LAN. If the target host is active, it responds with an ARP reply. In contrast, if the host is inactive after a predefined number of attempts, the source system stops further requests. For unresponsive hosts, the source device typically adds an incomplete ARP entry for the target IP address in its kernel ARP table. Attackers utilize this mechanism to determine which devices are live in a network by interpreting ARP responses to their probes.

Tools like Nmap are commonly used for ARP ping scans. The `-PR` option in Nmap specifically enables ARP ping scans, making it a popular choice for network discovery tasks. For users working with Zenmap, the graphical interface for Nmap, similar configurations can be applied to perform ARP-based discovery. By default, Nmap performs ARP ping scans as part of its process. However, users can disable it using the `--disable-arp-ping` flag to focus on other ping scan techniques. Additionally, the `-sn` option can be used to disable port scanning altogether during host discovery.

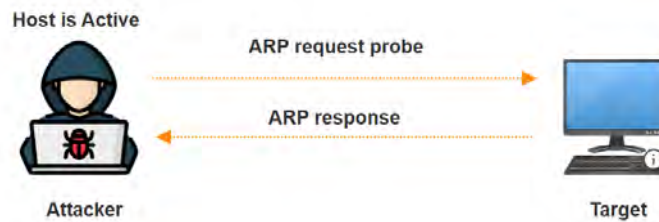


Figure 3-11: ARP Ping Scan

ARP ping scans are invaluable for identifying active hosts in a network environment, especially in LANs, due to their accuracy and ability to bypass traditional IP-layer restrictions. This makes them a critical tool for network administrators and ethical hackers conducting reconnaissance.

UDP Ping Scan

A UDP ping scan is similar to a TCP ping scan, but it uses the UDP protocol to identify active hosts in a network. In this method, Nmap transmits UDP packets to the target system in order to assess its status. By default, Nmap uses the uncommon port number 40,125 to send these UDP packets, though this can be customized during the compile time by modifying the `DEFAULT_UDP_PROBE_PORT_SPEC` parameter.

Attackers or network administrators utilize this method to detect live systems based on their responses. A UDP response from the target indicates that the host is active and reachable. Conversely, if the host is offline or unreachable, various error messages may be returned, such as host or network unreachable or TTL exceeded, providing clues about the host's status.

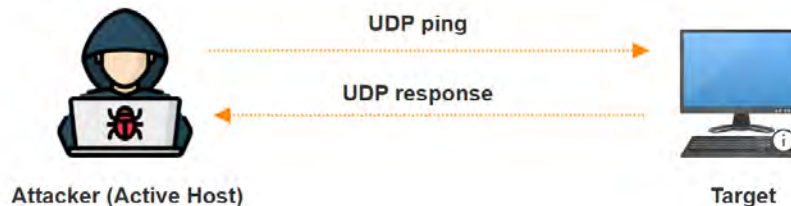


Figure 3-12: UDP Ping Scan to Find the Host Is Active

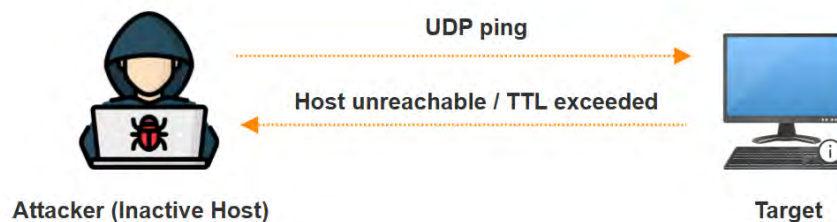


Figure 3-13: UDP Ping Scan Indicating the Host Is Offline.

In Zenmap, the graphical interface for Nmap, the `-PU` option is used to perform a UDP ping scan, enabling easy configuration for users. This technique is effective for identifying live systems that may not respond to other types of probes, making it an essential tool for network reconnaissance and host discovery.



EXAM TIP: When traditional ICMP or TCP methods fail, consider using a UDP ping scan. This can help detect live systems that do not respond to common protocols, especially when firewalls block other probes.

ICMP ECHO Ping Scan

An ICMP ECHO ping scan is a host discovery technique where attackers or network administrators send ICMP ECHO requests to a target system to determine its status. Unlike port scanning, which focuses on identifying open ports, this method is used to locate active hosts in a network by probing them. If a host is alive, it responds with an ICMP ECHO reply, confirming its presence on the network. This method is also valuable for assessing whether ICMP traffic can pass through firewalls, offering insights into the network's configuration and restrictions.



Figure 3-14: ICMP Echo Request and Reply

ICMP ECHO scanning is commonly employed on UNIX/Linux and BSD-based machines, as these systems' TCP/IP stack implementations are configured to respond to ICMP ECHO requests, even those directed at broadcast addresses. However, this technique is not effective in Windows-based environments because the TCP/IP stack implementation in these systems does not reply to ICMP probes aimed at broadcast addresses.

By leveraging ICMP ECHO ping scans, attackers and security professionals can efficiently identify live devices and assess network configurations, providing a foundation for further analysis or mitigation strategies.

Nmap facilitates ICMP ECHO ping scans using the `-P` option, enabling users to scan a target network and identify live hosts. For enhanced efficiency, the number of ICMP pings sent in parallel can be increased with the `-L` option, allowing users to adjust the intensity of the scan. Additionally, the `-T` option can be used to tweak the ping timeout value, offering greater control over scan responsiveness and timing.

In Zenmap, the graphical interface for Nmap, the -PE option performs the ICMP ECHO ping scan. The scan results display active hosts as "Host is up," providing clear feedback on the network's live devices. These tools and options allow for flexible and effective ICMP scanning, accommodating various network configurations and requirements.

ICMP ECHO Ping Sweep

An ICMP ECHO Ping Sweep is a fundamental network scanning technique used to identify which IP addresses within a range correspond to live hosts. Unlike a single ping, which checks the availability of a specific host, a ping sweep involves sending ICMP ECHO requests to multiple hosts across a network. Active hosts respond with ICMP ECHO replies, allowing the scanner to map live systems.

While effective, ping sweeps rank among the oldest and most time-consuming methods of network scanning. Their simplicity and utility make them widely available across nearly all platforms. Functioning like a "roll call" for systems on a network, active systems respond to ping queries from other systems.

ICMP echo scanning involves sending ICMP probes to the broadcast or network address, which distributes the probes to all host addresses within a subnet. Active hosts reply with ICMP ECHO responses, enabling attackers or network administrators to identify live systems within the network. Despite their limitations, ping sweeps remain a foundational approach in network discovery.

To understand pings fully, it is essential to grasp the concept of the TCP/IP packet. When a system sends a ping, it transmits a single packet over the network to a specified IP address. This packet typically contains 64 bytes—56 bytes of data and 8 bytes for protocol header information. The sender then listens for a return packet from the target system. If the network connections are stable and the target system is operational, a successful return packet is received. However, disruptions in communication or an inactive target system will result in no response.

Pings provide valuable insights, including the time taken for a packet to travel to the destination and return, known as the "round-trip time." They also assist in resolving hostnames. For example, if a ping to an IP address succeeds but fails when directed at a hostname, it indicates an issue with the system's ability to map the name to the corresponding IP address.

Attackers or network administrators can use subnet mask calculators to determine the number of hosts within a subnet.

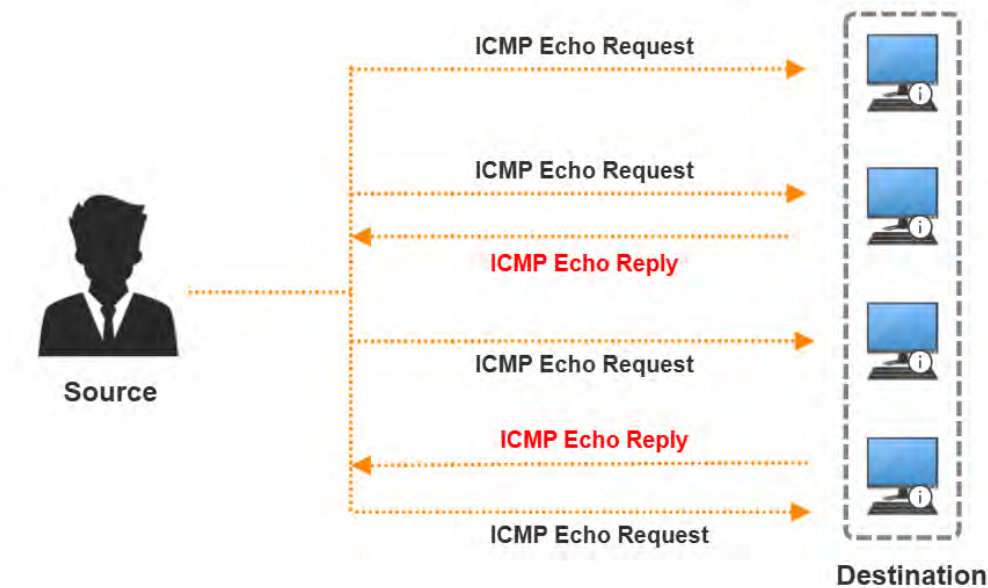


Figure 3-15: ICMP Echo Ping Sweep

ICMP Timestamp Ping Scan

Beyond the standard ICMP ECHO ping, attackers can employ alternative techniques like ICMP timestamp and ICMP address mask ping scans tailored for specific scenarios. The ICMP timestamp ping is an optional type of ICMP ping that allows an attacker to query a target host for the current time. The target machine responds with a timestamp reply, sending the time information back to the attacker.

The response from the destination host depends on specific conditions. Depending on the configuration set by the system administrator, the target machine may or may not reply with the time value. This makes the timestamp ping scan less reliable in some environments where the system is configured to block or ignore such queries. Despite this limitation, ICMP timestamp ping scanning can still be effective for identifying whether a destination host is active, particularly in situations where firewalls or security measures block traditional ICMP ECHO pings.

ICMP timestamp pings are commonly used for time synchronization between systems, and attackers can leverage this functionality to gather information about active hosts, even when other ping methods are restricted. In Zenmap, the -PP option initiates an ICMP timestamp ping scan. This technique provides an additional tool for network discovery in cases where standard ICMP ECHO scans are not feasible.

ICMP Address Mask Ping Scan

The ICMP address mask ping scan is another variation of the standard ICMP ECHO ping. In this approach, an attacker sends an ICMP address mask request to the target host to gather information

about the network's subnet mask, which is critical in identifying the network's size and the range of IP addresses within it.

As with the ICMP timestamp, the response to an address mask ping is conditional. The target host may or may not provide the appropriate subnet mask value, depending on how the system administrator configures it. In environments where traditional ICMP ECHO pings are blocked, the address mask ping scan can still be effective in identifying active hosts, as it bypasses some of the typical restrictions on standard ping methods.

The ICMP address mask ping scan is useful for attackers who want to gain information about the network structure, especially when other methods are being filtered or blocked. In Zenmap, the `-PM` option facilitates the execution of an ICMP address mask ping scan. This technique serves as an alternative method for network discovery in cases where more common ping techniques are not possible.

TCP SYN Ping Scan

The TCP SYN ping scan is a host discovery method that probes various target ports to verify system availability and assess the presence of firewall rules. In this method, an attacker uses a tool like Nmap to initiate the three-way handshake by sending an empty TCP SYN (synchronize) packet to the target host. If the target system is active, it responds with a SYN-ACK (synchronize-acknowledge) packet, indicating that the system is online. Once the attacker receives the SYN-ACK packet, the connection is terminated by sending a TCP RST (reset) packet to the target, as the objective—host discovery—has been completed.

Port 80 (the default HTTP port) is typically used for this scan, though a range of ports can be specified. For example, by using a format such as `-PS22-25,80,113,1050,35000` (with no spaces), the probe will be executed against each specified port in parallel. In Zenmap, the `-PS` option performs a TCP SYN ping scan to check for active hosts by sending SYN packets to specified ports.



Figure 3-16: TCP SYN Ping Scan for Discovering Hosts

The advantages of TCP SYN ping scans are significant. Firstly, as multiple machines can be scanned in parallel, the scan avoids time-out errors that often occur when waiting for responses from target systems. This parallel scanning increases efficiency and reduces delays. Secondly, TCP SYN ping scans allow attackers to determine if a host is active without establishing a full connection. Since the scan does not complete the three-way handshake and is terminated with a reset (RST) packet,

no persistent connection is logged at the system or network level. This makes the scan more stealthy, enabling attackers to carry out their discovery without leaving detectable traces.

TCP ACK Ping Scan

TCP ACK ping is a host discovery technique where the attacker sends an empty TCP ACK packet to the target, commonly on port 80, to check for system responsiveness. Since there is no established connection between the attacker and the target, the host responds to the incoming ACK packet with an RST flag to terminate the request. The attacker's receipt of this RST packet indicates that the target host is active. In Zenmap, the -PS option facilitates the execution of a TCP SYN ping scan.

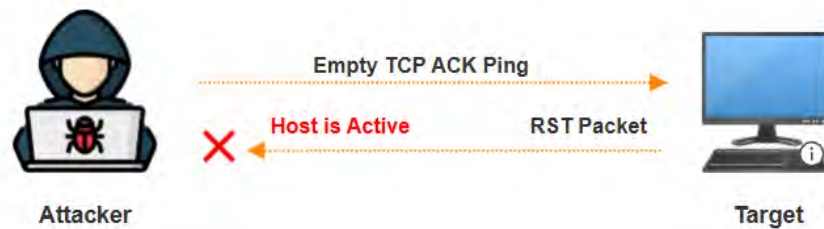


Figure 3-17: TCP ACK Ping Scan for Host Discovery

One of the key advantages of the TCP ACK ping technique is its ability to bypass firewall restrictions. Firewalls often block SYN packets, as they are commonly used for connection attempts, but ACK packets are less likely to be filtered. This makes the ACK ping a useful method for probing hosts, particularly in environments where firewalls are configured to block SYN ping packets.

IP Protocol Ping Scan

The IP Protocol Ping Scan is a host discovery method that involves sending IP ping packets using various IP protocols to determine whether a host is online. This technique works by sending packets with an IP header of a specified protocol number, much like TCP and UDP ping scans. By utilizing different IP protocols, the scan attempts to elicit a response from the target host, signaling its activity.

When no specific protocol is defined, the scan, by default, sends packets using the following protocols: ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4). These protocols are commonly used for network management and communication. However, when targeting specific protocols such as ICMP, IGMP, TCP (protocol 6), and UDP (protocol 17), the packets must include the appropriate protocol headers. For non-TCP/UDP protocols, the packet consists solely of IP header data.

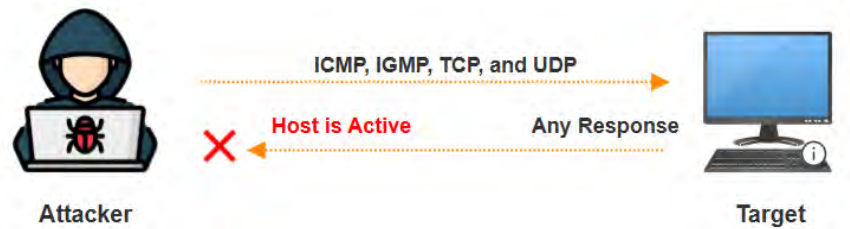


Figure 3-18: IP Protocol Ping Scan for Host Discovery

The IP Protocol Ping Scan offers flexibility for discovering hosts that might otherwise be concealed behind firewalls or filtering mechanisms that block common ping methods. By using a range of protocols, this scan increases the chances of successfully identifying active hosts. To customize the default protocols used during the scan, users can modify the `DEFAULT_PROTO_PROBE_PORT_SPEC` setting in the Nmap configuration file (`nmap.h`) during compile time.

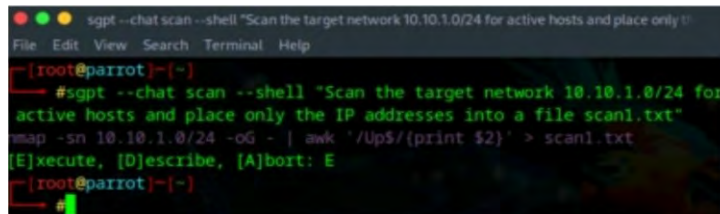
Host Discovery with AI

AI-powered technologies can significantly enhance and automate host discovery tasks, allowing attackers to identify live hosts on a target network quickly and efficiently. With AI's capabilities, attackers can streamline and automate the discovery process, removing much of the manual effort involved.

Example:

An attacker could use a tool like ChatGPT to generate a specific prompt such as:

“Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt.”



```

sgpt --chat scan --shell "Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt"
[root@parrot]~# sgpt --chat scan --shell "Scan the target network 10.10.1.0/24 for active hosts and place only the IP addresses into a file scan1.txt"
nmap -sn 10.10.1.0/24 -oG - | awk '/Up$/{print $2}' > scan1.txt
[E]xecute, [D]escribe, [A]bort: E
[root@parrot]~#
  
```

Figure 3-19: Scan the Target Network to Identify Active Hosts

To perform the task, the attacker would use a command like the following:

```
nmap -sn 10.10.1.0/24 -oG - | awk '/Up$/{print $2}' > scan1.txt
```

Table 3-2 provides a description of each option used in the command.

Command Breakdown	Description
nmap	Invokes Nmap is a robust network scanning tool used for various network discovery tasks.
-sn	This flag specifies a "ping scan" or "ping sweep," where Nmap sends ICMP echo requests to identify which hosts are live without probing ports.
10.10.1.0/24	Denotes the target IP range (from 10.10.1.0 to 10.10.1.255), defined using CIDR notation.
-OG -	Outputs the scan results in a "grepable" format, which other tools can easily process.
 awk '/Up\$/{print \$2}'	This pipes the output to the awk command, filtering for lines containing "Up" (which indicates the host is active) and extracting the second field, which is the host's IP address.
> scan1.txt	Directs the filtered output (the live IP addresses) to a text file named scan1.txt.

Table 3-2: Description of Nmap for identifying live host in a target server

Ping Sweep Tools

Ping Sweep is a method for detecting active devices within a vast network by sending ICMP echo requests across a series of IP addresses. Ping Sweep involves sending ICMP echo request packets to multiple IP addresses in a range, rather than targeting each address individually, to identify active hosts. Active hosts respond with ICMP echo reply packets, indicating that they are online and reachable. Consequently, rather than examining each IP address separately, you can assess a series of IPs through the use of a Ping Sweep. Various tools are accessible for conducting a Ping Sweep. You can ping the range of IP addresses using these ping sweep tools, such as the SolarWinds Ping Sweep tool or Angry IP Scanner. Additionally, they can perform the reverse DNS lookup, resolve hostnames, bring MAC addresses, and scan ports.

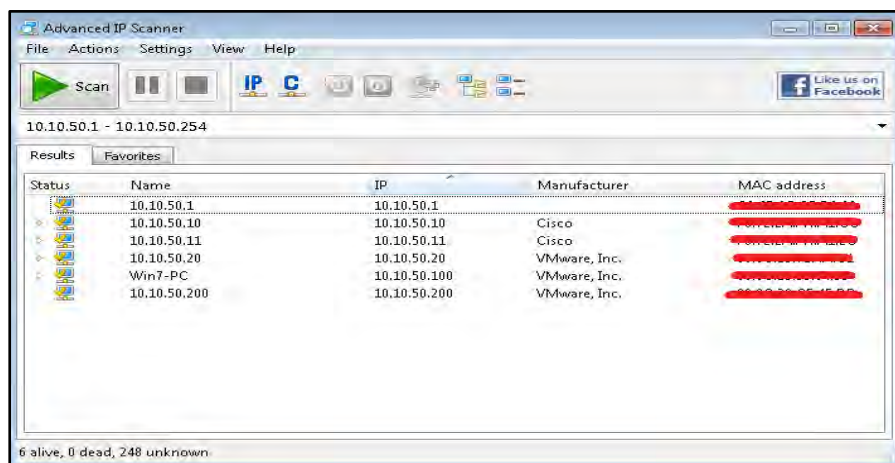


Figure 3-20: Ping Sweep using Advance IP Scanner Tool



EXAM TIP: Ping sweep tools such as SolarWinds and Angry IP Scanner check a range of IP addresses to discover active devices. They can also carry out DNS lookups and identify MAC addresses, although firewalls might prevent them from doing so.

Port and Service Discovery

The next step in network scanning involves discovering open ports and services on live systems. Administrators use port scanning techniques to verify network security policies, while attackers use them to identify open ports and running services with the aim of compromising the system. Additionally, users may unknowingly leave unnecessary open ports on their systems, which attackers can exploit to launch attacks. Port scanning techniques such as TCP connect scan, SYN scan, UDP scan, stealth scan, and FIN scan are commonly used to detect vulnerabilities and expose open ports. Regular port scanning is crucial to identifying security weaknesses and securing networks.

Port Scanning Techniques

Scanning techniques include UDP and TCP scanning. Figure 3-17 displays the different types of scanning techniques:

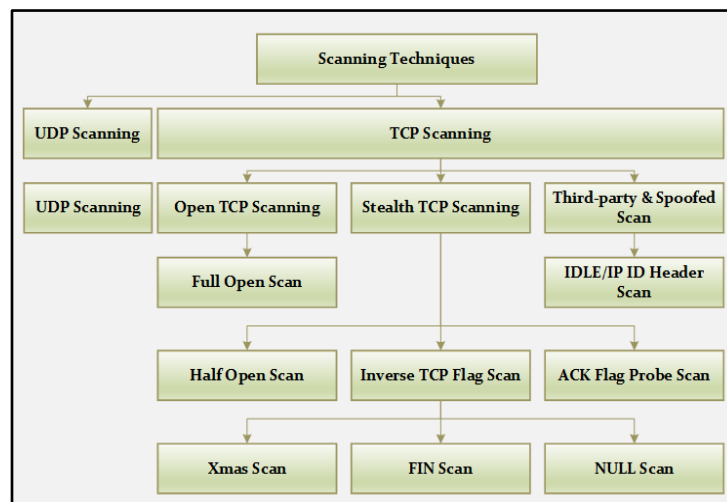


Figure 3-21: Scanning Techniques

TCP Connect/Full-Open Scan

In this scanning technique, a three-way handshake session is initiated and completed. Full Open Scanning guarantees that the targeted host is live by completing the full TCP handshake, ensuring the connection is successfully established. This is considered a major advantage of Full Open Scanning. However, security devices such as Firewalls and IDS can detect and log it. The TCP Connect/Full Open Scan does not need admin rights to execute.

If a closed port is encountered while using Full Open Scanning, the RST response is sent to the incoming request to terminate the attempt. To perform a Full Open Scan, you must use the `-sT` option for Connect Scan.

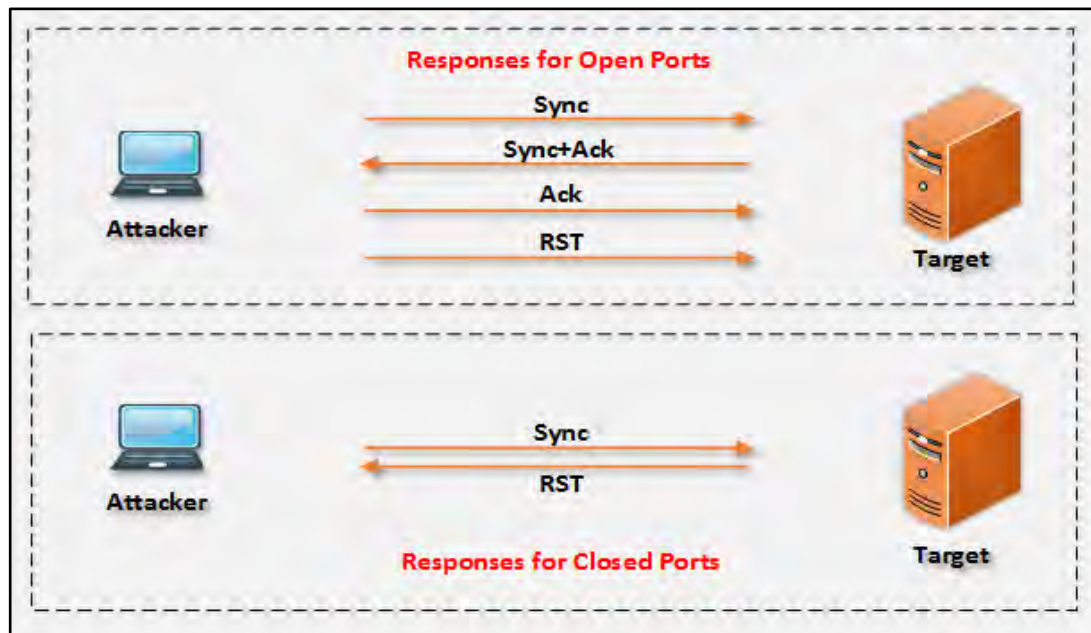


Figure 3-22: TCP Connection Responses

Type the command to execute Full Open Scan:

```
nmap -sT <ip address or range>
```

For example, consider the result illustrated in the figure 3-19 provided below. The Zenmap tool is used to perform a Full Open Scan.

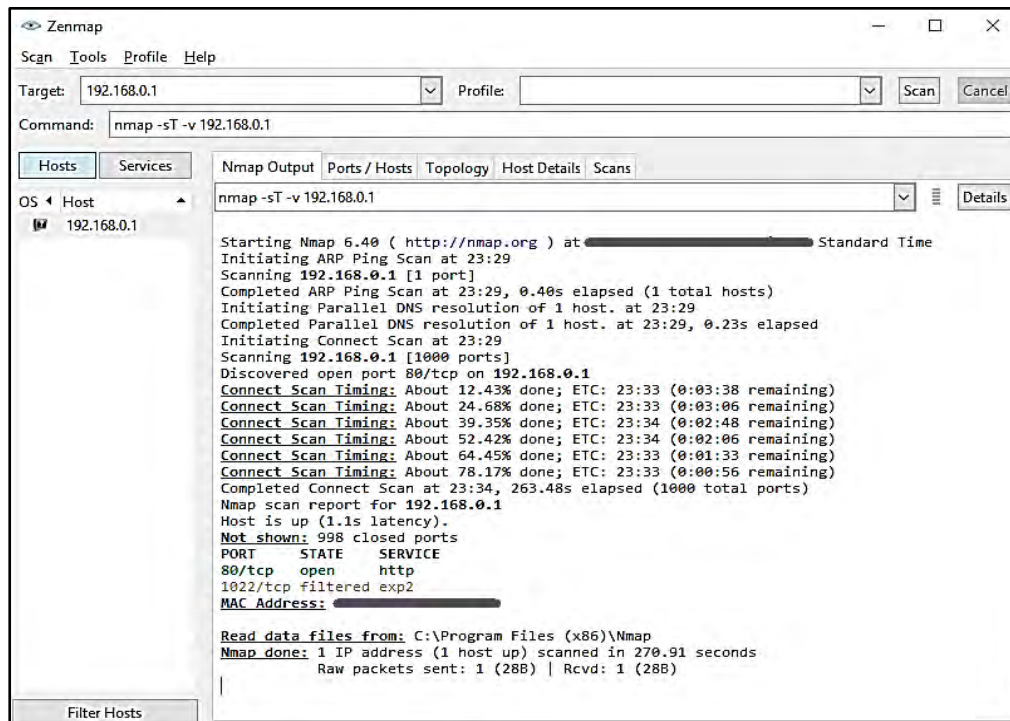


Figure 3-23: Full Open Scan

Stealth Scan (Half-Open Scan)

A Stealth Scan, also known as a Half-Open Scan, works as follows in a scenario with two hosts: Host A and Host B. Host A commences the TCP connection by transmitting a SYN packet to Host B. Host B transmits a SYN+ACK packet in reply. However, instead of completing the connection, Host A does not send the final ACK packet, leaving the connection "half-open."

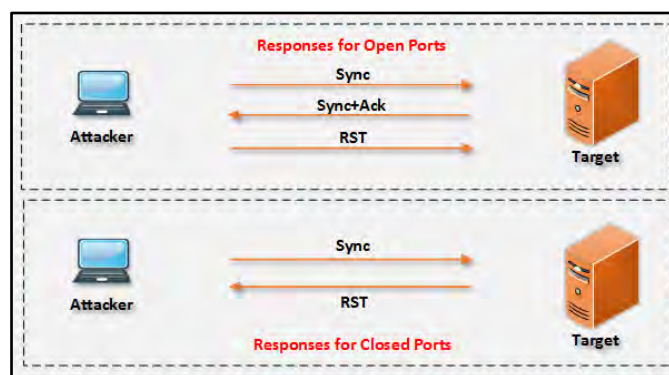


Figure 3-24: TCP Half Open Scan

You can execute this type of scan using the following syntax:

```
nmap -sS <ip address or range>
```

Observe the result in the figure 3-21 below:

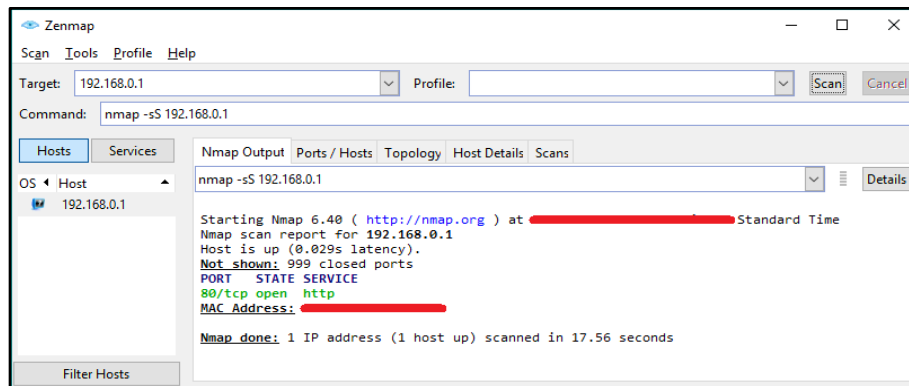


Figure 3-25: Half Open Scan

Inverse TCP Flag Scan

Inverse TCP Flag Scanning is a method used by attackers to scan network ports by sending TCP probes with specific flags or without any flags at all. Probes with flags such as FIN, URG, and PSF are known as Xmas Scanning, while probes with no flags are referred to as Null Scanning.



Figure 3-26: Inverse TCP Flag Scan When the Port is Open



Figure 3-27: Inverse TCP Flag Scan When the Port is Closed

In this technique, attackers send probe packets to specific target ports to gather information. If the port is open, the host typically does not respond, as it is configured to ignore unsolicited packets. However, when the port is closed, the host sends an RST (reset) packet back. This behavior is based on RFC 793, which states that a connection reset (RST/ACK) is returned when a port is closed.

To exploit this, attackers use different flag configurations for probing the target, including:

FIN probe	Only the FIN flag is set.
Xmas probe	The flags URG, FIN, and PUSH are set.
NULL probe	No flags are set.
SYN/ACK probe	SYN and ACK flags are set.

Operating systems like Windows may ignore the RFC 793 standard, meaning no response is sent for closed ports. This makes inverse TCP flag scanning more effective against UNIX-based systems, where the RST/ACK response is more predictable and easier to detect.

Xmas Scan

An Xmas Scan is a type of scan that activates multiple flags, including the URG, PSH, and FIN flags, creating an unusual condition for the receiver. When the target system receives such a packet, it must decide how to handle the situation. If the port is inaccessible, the system will reply with a single RST packet. If the port is open, some systems may respond as if the port is open, but modern systems typically ignore or drop the request because the combination of flags is invalid.

On the other hand, a FIN Scan works only with operating systems that follow TCP/IP configurations based on RFC-793. This scan method does not work with modern versions of Windows, such as Windows XP or Windows Vista, as they handle the FIN flag differently and do not respond as older systems would.



Figure 3-28: Xmas Scan

You can perform this type of scan by using the following command syntax:

```
nmap -sX -v <ip address or range>
```

TCP Maimon Scan

The TCP Maimon scan represents a modification of the NULL, FIN, and Xmas scanning techniques, employing a FIN/ACK probe to detect open ports. Usually, a reply in the form of an RST packet shows if a port is open or closed. However, on some BSD-based systems, if there is no response, the port is seen as open. In Nmap, a Maimon scan sends several probes to a port. When no response is detected, the port is identified as open|filtered. On the other hand, getting an RST packet indicates that the port is closed. Also, an ICMP unreachable error further confirms that the port is closed.



Figure 3-29: TCP Maimon Scan (Open Port)

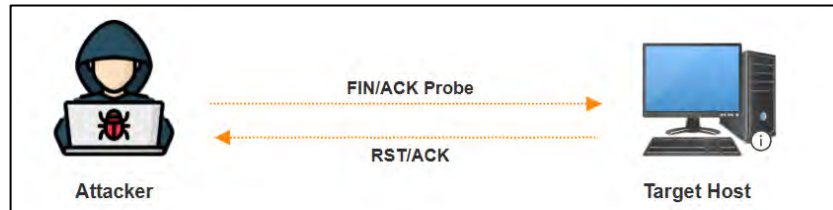


Figure 3-30: TCP Maimon Scan (Closed Port)

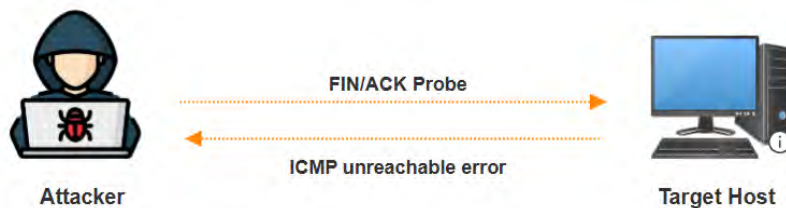


Figure 3-31: TCP Maimon Scan Result of Filtered Port

ACK Flag Probe Scan

The method involves sending a TCP packet to the target system with the ACK flag enabled. The sender analyzes the header information, specifically the Time to Live (TTL) and WINDOW fields, in the RST packet sent back by the target. The target responds with an RST packet, regardless of whether the port is open or closed. By examining these fields, the attacker can determine whether the port is open or closed.

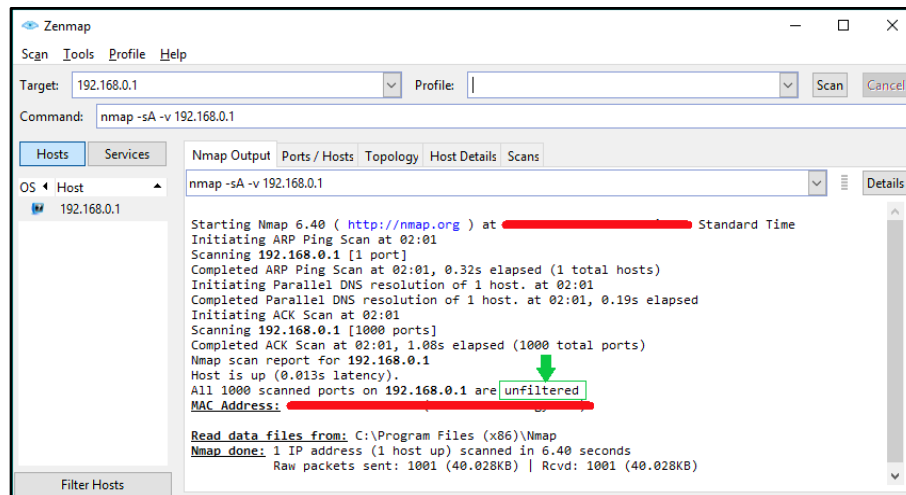


Figure 3-32: ACK Flag Probe Scanning

ACK Probe scanning also assists in detecting the existence of a filtering system. If an RST packet is received from the target, packets toward this port are not being filtered. If there is no response, it indicates that a stateful firewall is filtering the port, blocking the communication, and preventing the attacker from receiving any feedback.

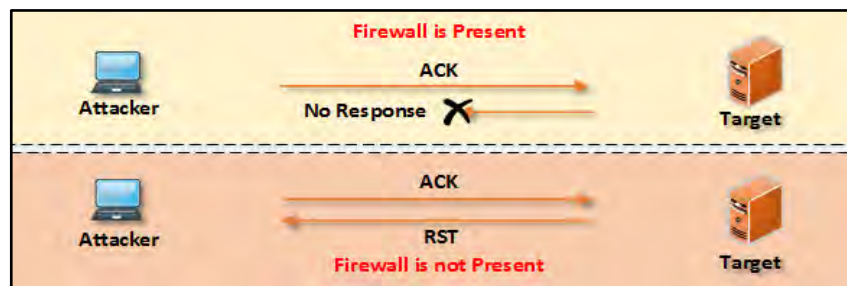


Figure 3-33: ACK Flag Probe Scanning Response

IDLE/IPID Header Scan

IDLE/IPID Header Scan is a distinct technique used to determine the port status of a target host while maintaining a low profile. This method allows the attacker to conceal their identity by relaying packets through a "zombie" system, effectively masking their actions.

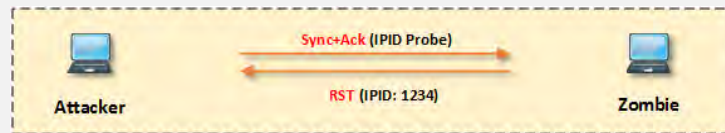
Before understanding the steps required for the IDLE/IPID Scan, you must keep the following important points in mind:

- Target Machine responds with the RST packet if the port is closed
- The unsolicited SYN+ACK packet is either disregarded or met with a RST response.
- A Fragment Identification Number (IPID) is assigned to each IP packet.
- OS increments IPID for each packet

Step: 01

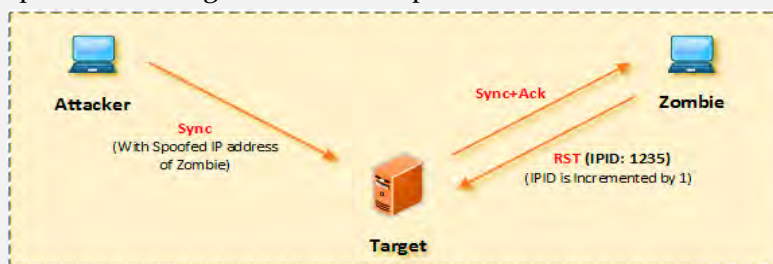
- Transmit a SYN+ACK packet to the Zombie in order to retrieve its IPID number.

- Zombie is not waiting for SYN+ACK; hence, it responds with the RST packet. Its reply discloses the IPID
- Extract IPID from the Packet

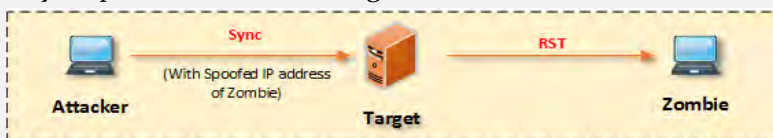


Step: 02

- This SYN packet is transmitted to the target using a forged IP address that appears to come from a Zombie system.
- When the target's IP port is open, it sends a SYN+ACK packet to the Zombie in response. The Zombie then replies to the target with an RST packet.

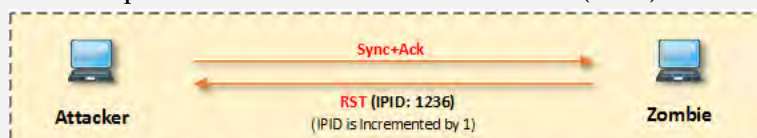


- If the port is closed, the target transmits an RST response to the Zombie; however, the Zombie does not relay any response back to the target. IPID of Zombie is not incremented



Step: 03

- Send the SYN+ACK packet to Zombie again to receive and compare its IPID Numbers to the IPID extracted in step 01 (i.e., 1234)
- Zombie responds with the RST packet. Its reply discloses the IPID
- Extract IPID from the Packet
- Compare the IPID
- The port is considered open if the IP Identification Number (IPID) increments by 2.



- If the IPID increases by 1, the port is seen as closed.

UDP Scan

Just like TCP-based scanning techniques, there are also methods for scanning UDP. However, it is important to note that UDP is a connectionless protocol, meaning it does not establish a formal

connection before sending data. UDP packets work with ports; no connection orientation is required. No response will be received if the targeted port is open; however, if the port is closed, a response message will be received stating, "Port unreachable." Most malicious programs, Trojans, and spyware use UDP ports to access the target.

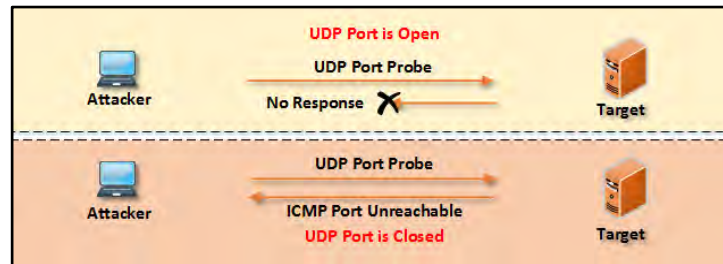


Figure 3-34: UDP Scanning Response

You can perform this type of scan in Nmap by using the following command syntax:

```
nmap -sU -v <ip address or range>
```

Observe the result in the following figure:

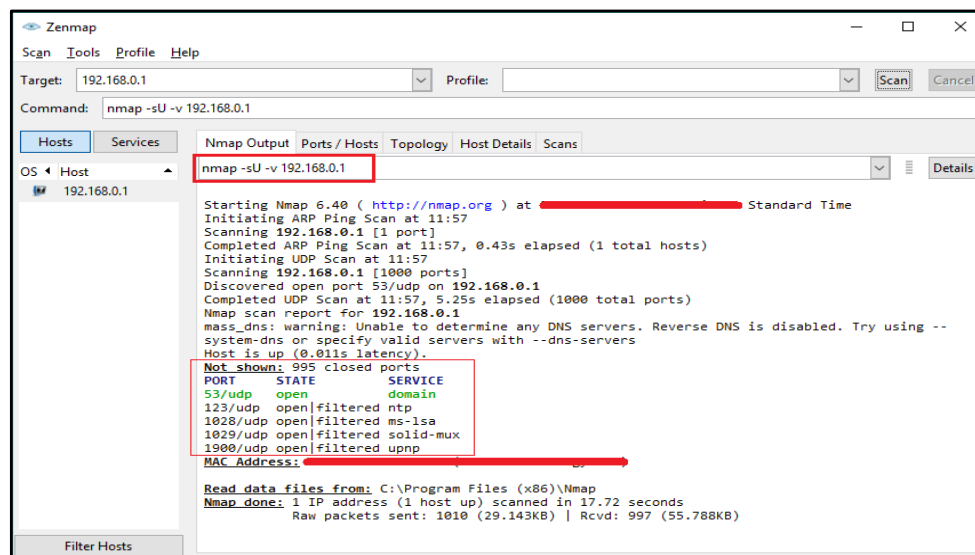


Figure 3-35: UDP Port Scanning

Advantage:

A UDP scan is less formal when it comes to open ports because it bypasses the overhead of a TCP handshake. Nonetheless, in the event that ICMP responses are generated for inaccessible ports, the total quantity of frames may surpass that of a typical TCP scan. Since Microsoft-based operating systems rarely implement ICMP rate limiting, UDP scans can be performed very efficiently on Windows-based devices.

Disadvantage:

UDP scans provide limited information and only offer port availability. To obtain additional details, such as version information, a version detection scan (-sV) or OS fingerprinting (-O) must be conducted alongside the UDP scan. Furthermore, privileged access is required, making this option only available to users with the appropriate permissions. Due to the higher volume of TCP traffic on most networks, UDP scans typically exhibit lower efficiency than TCP scans.

SCTP INIT Scan

Stream Control Transmission Protocol (SCTP) is a reliable, message-oriented transport layer protocol that shares some similarities with both TCP and UDP. It is frequently used in tasks involving multiple connections and multiple streams and is commonly applied in Voice over Internet Protocol (VoIP), internet phone services, and Signaling System 7 (SS7) services. SCTP creates connections through a four-step handshake process.

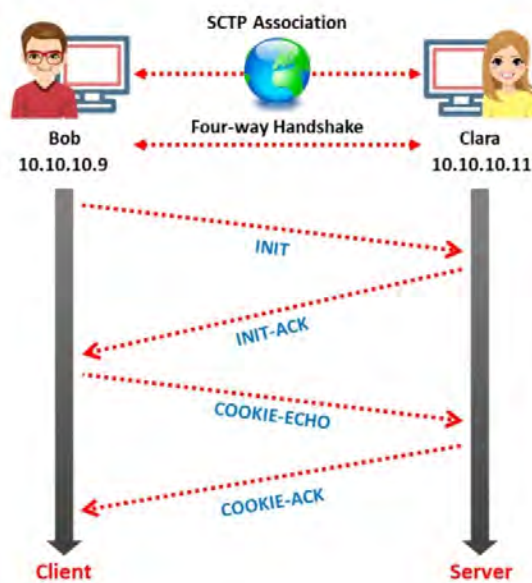


Figure 3-36: SCTP Association Four-Way Handshake

The SCTP INIT scan is a fast technique that allows for scanning thousands of ports per second on a network with minimal interference from firewalls, making it ideal for networks with stronger security measures. This scan is similar to the TCP SYN scan but is stealthier and less intrusive because it does not complete the full SCTP association, leaving the connection half-open.

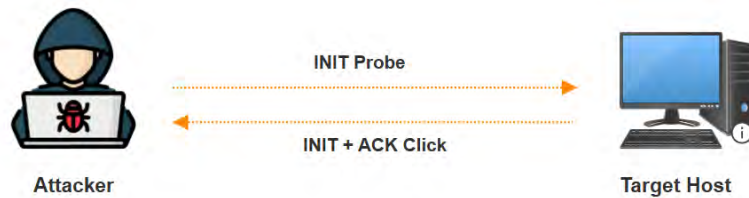


Figure 3-37: Sctp INIT Scan When a Port Is Listening (Open)

In this process, attackers initiate the connection by sending an INIT chunk to the target system. If the target port is open, the system responds with an INIT+ACK bit. This acknowledges the request.

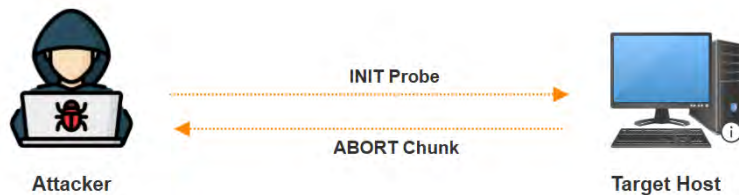


Figure 3-38: Sctp INIT Scan When a Port Is Not Listening (Closed)

In the event that the port is either closed or inactive, the response will consist of an ABORT chunk. If there is no reply following several retransmissions, the port is considered to be filtered. Additionally, filtered ports may also result from receiving an ICMP unreachable error from the target server.

SCTP COOKIE ECHO Scan

The SCTP COOKIE ECHO scan is a more advanced scanning method that uses the COOKIE ECHO chunk to probe target ports. When an attacker sends this chunk, if the port is open, the target silently drops the packets without sending a response. A reply with an ABORT chunk from the target indicates that the port is closed. This scan is less detectable by non-stateful firewalls compared to the INIT scan, and only advanced IDS systems can identify it.



Figure 3-39: Sctp COOKIE ECHO scan result when a port is open

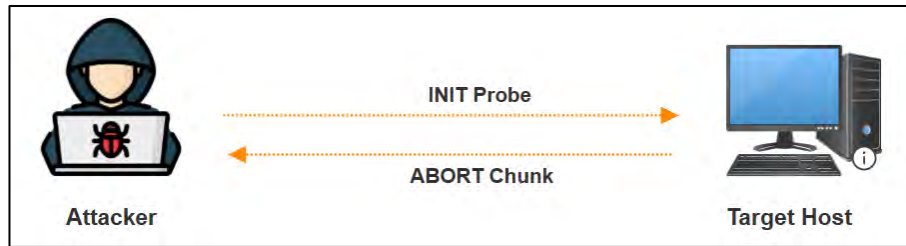


Figure 3-40: Sctp COOKIE ECHO Scan Result for a Closed Port

The Sctp COOKIE ECHO scan is less noticeable than the INIT scan, making it more stealthy and harder to detect. The system is unable to effectively differentiate between open and filtered ports, frequently presenting both categories as "open|filtered" in the scanning results.

SSDP and List Scan

Simple Service Discovery Protocol (SSDP) is a protocol used for discovering network services without the assistance of server-based configurations like Dynamic Host Configuration Protocol (DHCP), Domain Name System (DNS), and static network host configuration. SSDP facilitates the discovery of Plug and Play devices using Universal Plug and Play (UPnP) protocols. This protocol is designed to be compatible with both IPv4 and IPv6.

A list scan is a technique used to discover active network hosts indirectly. It produces and displays a compilation of IP addresses or hostnames without performing any pinging or scanning of the respective hosts. As a result, it shows all IP addresses as "not scanned" (o hosts up), but Nmap performs a reverse DNS resolution to carry out each host by Nmap to learn their names. In Zenmap, the -sL option is used to execute a list scan. One advantage of a list scan is that it can be a good sanity check and help detect incorrectly defined IP addresses in the command line or option files. It mainly corrects these errors before running any active scan.

IPv6 Scan

IPv6 scanning is more challenging and complex compared to IPv4 due to the expanded address space, which increases from 32 bits to 128 bits, providing a larger search space of 2^{64} addresses within a subnet. Traditional scanning methods are less feasible for IPv6 networks because of this vast address space, and many scanning tools do not support ping sweeps on IPv6. Attackers often collect IPv6 addresses from network activity, logs, or email headers for later port scanning. Once an attacker compromises one host within an IPv6 subnet, they can probe all hosts using link-local multicast addresses or by utilizing address schemes if the host numbers are sequential. However, performing a full scan of IPv6 addresses is computationally intensive, and scanning a subnet could take years at a conservative rate. Nmap supports IPv6 scanning, with the -6 option used in Zenmap to perform these scans.

Port Scanning with AI

Port scanning with AI allows attackers to automate and streamline the process of identifying open ports on a target network. Using AI-powered tools, such as ChatGPT, attackers can quickly and efficiently issue commands to Nmap for scanning tasks.

Example # 1

Attackers can ask AI models like ChatGPT to help them use Nmap by giving instructions like:

"Use Nmap to find open ports on target IP 10.10.1.11"

```
[attacker@parrot]~$ sgpt --chat sn --shell " Use Nmap to find open ports on target IP 10.10.1.11"
nmap 10.10.1.11
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-14 08:16 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00076s latency).
Not shown: 994 closed tcp ports (conn-refused)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
```

Figure 3-41: Using Nmap to Discover Open Ports on a Target IP

Example # 2

In another example, an attacker could use ChatGPT to initiate a stealth scan by typing: **"Perform stealth scan on target IP 10.10.1.11 and display the results."**

This command would enable the attacker to execute a covert scan that attempts to avoid detection, gathering information on open ports without triggering security alerts.

```
[root@parrot]~/home/attacker$ #sgpt --shell "Perform stealth scan on target IP 10.10.1.11 and display the results"
nmap -sS -Pn 10.10.1.11 && echo "Stealth scan completed on 10.10.1.11"
[E]xecute, [D]escribe, [A]bort: E
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-04-01 06:12 EDT
Nmap scan report for 10.10.1.11
Host is up (0.00066s latency).
Not shown: 994 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
80/tcp    open  http
135/tcp   open  msrpc
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds
3389/tcp  open  ms-wbt-server
MAC Address: 00:15:5D:01:80:00 (Microsoft)

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

Figure 3-42: Conducting a Stealth Scan on Target IP

Service Version Discovery

It is a technique used to identify the specific service and its version running on a particular port. Since some versions of services have known vulnerabilities, attackers can exploit these weaknesses to compromise a target system. By detecting the service versions, attackers can gather information about the running services and determine potential vulnerabilities based on known exploits for those versions. This process involves scanning TCP and UDP ports and using probes from Nmap's service-probes database to query various services. Zenmap uses the -sV option to detect the service version.

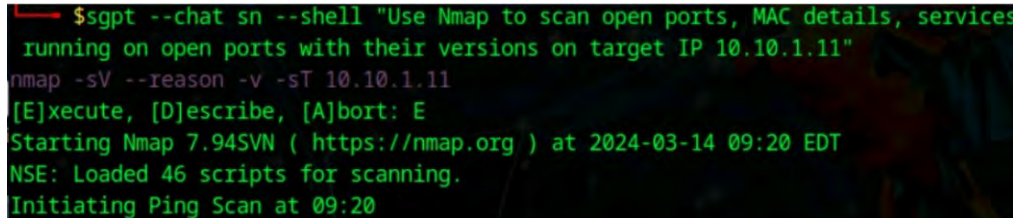
Service Version Discovery with AI

Attackers can utilize AI-powered technologies to streamline and automate service discovery. With the assistance of AI, attackers can quickly identify live systems, open ports, and the services running on those ports, including their versions, on target networks.

Example:

Attackers can prompt AI models like ChatGPT to guide them through using Nmap by providing commands such as:

"Use Nmap to scan open ports, MAC details, and services running on open ports with their versions on target IP 10.10.1.11."



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

Figure 3-43: AI Prompt for Nmap Service and MAC Details

```
nmap -sV --reason -v -sT 10.10.1.11
```

The following Nmap command is designed to perform port scanning, service enumeration, and version detection on a specific target IP address. The command identifies open ports, determines the services running on those ports, and retrieves the versions of those services.

Command	Description
nmap	Launches the Nmap tool.
-sV	Enables service version detection, allowing identification of service versions.
--reason	Displays the reason for each port state (open, closed, filtered).
-v	Increases verbosity for more detailed output.
sT	Performs a TCP connect scan, establishing full connections to each port being scanned.
10.10.1.11	Specifies the target IP address.

Table 3-3: Description of the Nmap Command for Service Version Discovery

Nmap Scan Time Reduction Techniques

In Nmap, optimizing performance and accuracy involves several strategies to reduce scan times and ensure efficiency. Here are key techniques that can be applied:

- Omit Non-critical Tests:** To reduce scan times, it is important to avoid running unnecessary or intensive scans when only minimal information is required. Limiting the number of ports scanned and skipping the port scan (-sn) when only host availability is needed can significantly improve speed. Advanced scan types such as—SC,—sV, -O,—tracroute, and -A should be avoided unless necessary. Additionally, enabling DNS resolution only when required will save processing time.
- Optimize Timing Parameters:** Nmap provides the -T option, which allows users to adjust the scan's timing aggressiveness. By setting the timing level, users can control the scan's speed and accuracy, making it particularly useful for highly filtered networks. Selecting an optimal timing parameter can help reduce scan times while still obtaining reliable results.
- Separate and Enhance UDP Scans:** Conducting UDP scans is essential, as numerous vulnerable services operate over this protocol. However, UDP scans should be conducted separately from TCP scans due to differences in their performance characteristics and timing behavior. UDP scans are more susceptible to ICMP error rate-limiting, making them slower and more prone to congestion. By isolating UDP scans, you can improve the overall scanning performance.
- Upgrade Nmap:** Using the latest version of Nmap is important for maintaining high performance. Each new release often includes bug fixes, algorithm improvements, and features designed to optimize scan speed and accuracy. New capabilities, such as local ARP scanning, can also improve network discovery and reduce scanning time.
- Execute Concurrent Nmap Instances:** Running multiple Nmap instances concurrently can be highly effective for speeding up the scanning process across a large network. By dividing the scan into smaller groups and scanning them simultaneously, you can achieve faster results. This parallelization approach leverages Nmap's ability to handle multiple tasks at once, improving overall efficiency.

- **Scan from a Favorable Network Location:** Performing Nmap scans from the host's local network typically provides better results and a higher degree of security, as it allows for internal network defense-in-depth. External scanning, however, is necessary when performing firewall tests or when assessing the network from the perspective of an external attacker. Choosing the right scanning location can impact scan speed and accuracy.
- **Increase Available Bandwidth and CPU Power:** To further reduce scan times, increasing available bandwidth or CPU power can make a significant difference. This can be achieved by upgrading network lines or halting other bandwidth-intensive processes during the scan. Additionally, Nmap's congestion control algorithms prevent network flooding, ensuring that the scan remains accurate and efficient while making the best use of available resources.

OS Discovery (Banner Grabbing/OS Fingerprinting)

An attacker uses OS discovery and banner grabbing techniques to identify network hosts running specific applications and OS versions that may have known vulnerabilities. Banner grabbing involves capturing and analyzing service messages that contain information about the software and its version. This can be done either passively by sniffing traffic or actively by connecting to the target system. It is a key method for attackers to gather information without launching a full attack. This section covers the types of banner grabbing and the tools used for it.

OS Discovery/Banner Grabbing

Banner grabbing, often referred to as OS fingerprinting, is a method employed to identify the Operating System (OS) that is operational on a remote system. It is a valuable method for attackers as knowing the OS increases the chances of successful exploitation since many vulnerabilities are OS-specific. Two methods for banner grabbing are spotting a banner when connecting to services like FTP and downloading files (e.g., /bin/lis) to check the system architecture. More advanced methods, like stack querying, involve sending packets to the network host and analyzing the responses to determine the OS.

Active Banner Grabbing

In active banner grabbing, an attacker sends specially crafted TCP packets to a target and analyzes the response, which varies based on the OS's implementation of the TCP/IP stack. For example, tools like Nmap perform several tests to identify an OS by sending specific packets with different flags. Tests may include sending TCP packets with varying flag combinations, sending UDP packets to closed ports, and analyzing TCP sequence numbers. The patterns found in the responses help determine the OS type.

Nmap's OS Fingerprinting Tests

Nmap employs nine specific tests to determine an OS fingerprint actively:

1. **SYN and ECN-Echo Test:** Sends a TCP packet with the SYN and ECN-Echo flags enabled to an open TCP port.

2. **NULL Packet Test:** Sends a TCP packet with no flags enabled (a NULL packet) to an open TCP port.
3. **Combination Flag Test:** Sends a TCP packet with the URG, PSH, SYN, and FIN flags enabled to an open TCP port.
4. **ACK Test (Open Port):** This test sends a TCP packet with the ACK flag enabled to an open TCP port.
5. **SYN Test (Closed Port):** This test sends a TCP packet with the SYN flag enabled to a closed TCP port.
6. **ACK Test (Closed Port):** This test sends a TCP packet with the ACK flag enabled to a closed TCP port.
7. **URG, PSH, and FIN Test:** Sends a TCP packet with the URG, PSH, and FIN flags enabled to a closed TCP port.
8. **Port Unreachable (PU) Test:** This test sends a UDP packet to a closed UDP port, aiming to extract an "ICMP port unreachable" message from the target system.
9. **TCP Sequence Ability (TSeq) Test:** Sends six TCP packets with the SYN flag enabled to an open TCP port to analyze the patterns in the initial sequence numbers (ISN) generated by the TCP implementation.

These tests are designed to identify distinct patterns in the responses and compare them to a signature database. This allows attackers or security professionals to identify the operating system and its version with high accuracy. The active banner-grabbing technique provides critical insights into the target system's characteristics, which can be used to identify potential vulnerabilities.

Passive Banner Grabbing

Unlike active banner grabbing, passive banner grabbing involves sniffing network traffic to capture packets from the target system. By analyzing these packets, attackers can identify characteristics that reveal the OS without directly interacting with the system, making it less detectable. This method relies on the differences in how different OSes implement their TCP/IP stack.

There are some methods of passive grabbing, including:

- **Error Messages:** Error messages provide valuable details such as the type of server, operating system, or SSL tools used by the target system.
- **Network Traffic Sniffing:** Capturing and analyzing packets transmitted by the target can reveal information about the OS and network behavior.
- **Page Extensions:** Observing URL extensions can help identify the application or server version.

Why Banner Grabbing?

The primary purpose of banner grabbing is to identify the target's OS and applications, enabling attackers to pinpoint vulnerabilities and exploit them. Passive methods offer a stealthier approach

to gathering this information without triggering alarms, making them a preferred technique in environments with active IDS/IPS systems.

How to identify the Target System OS

Identifying the target OS is crucial for attackers seeking to compromise a network or machine. Networks use various protocols such as IP, TCP, and UDP, and these protocols have specific parameters that reveal details about the OS. Parameters like Time to Live (TTL) and TCP window size found in the IP header of the first packet in a TCP session can help identify the OS. The TTL field defines how long a packet can remain in the network, and the TCP window size defines the length of the packet. These values vary by OS, providing clues to the attacker.

Attackers can use various tools for OS discovery, including Wireshark, Nmap, Unicornscan, and Nmap Script Engine. Additionally, IPv6 fingerprinting can be used to identify the OS.

OS Discovery using Wireshark

To identify the target OS using Wireshark, attackers can sniff and capture the response from the target machine in a network session. By examining the TTL and TCP window size fields in the first captured TCP packet, attackers can compare these values with known OS values to identify the target system's OS. This packet-sniffing technique allows for indirect OS discovery without actively scanning the target.

OS Discovery using Nmap and Unicornscan

OS Discovery using Nmap

Identifying the Operating System (OS) is crucial for exploiting a target machine. Nmap is an effective tool for discovering the OS running on a target system. By using the `-O` option in Nmap, attackers can perform OS discovery, which reveals detailed information about the OS on the target machine. This method helps attackers tailor their strategies based on the specific vulnerabilities of the target OS.

OS Discovery using Unicornscan

Unicornscan is another tool that can be used to find out the operating system of a target computer by looking at the Time to Live (TTL) values in the scan results. To carry out an OS discovery scan using Unicornscan, the command `#unicornscan <target IP address>` is used. The resulting TTL value can help determine the OS; such as a TTL value of 128 typically indicates that the target is running Microsoft Windows.

OS Discovery using Nmap Script Engine

The Nmap Scripting Engine (NSE) allows attackers to automate networking tasks using scripts that can be executed alongside Nmap scans. For OS discovery, attackers can use specific scripts like `smb-os-discovery`, which collects OS information via the SMB protocol. In Nmap, NSE is activated with the `-sC` option for default scripts, or custom scripts can be specified using the `--script` option.

The results from these scripts are displayed in both the Nmap normal and XML output formats, providing detailed OS information about the target machine.

OS Discovery using IPv6 Fingerprinting

IPv6 fingerprinting is a technique used to identify the OS of a target machine by sending probes and analyzing the responses, similar to IPv4 fingerprinting. In contrast, IPv6 incorporates more sophisticated probes along with a dedicated OS detection engine specifically designed for IPv6 environments. Nmap transmits approximately 18 probes in a designated sequence to facilitate IPv6 operating system discovery. These probes include sequence generation (S1–S6), ICMPv6 echo requests (IE1 and IE2), Node Information Queries (NI), Neighbor Solicitation (NS), UDP (U1), TCP explicit congestion notification (TECN), and various TCP probes (T2–T7). This allows attackers to identify the OS running on a target system using the distinct behaviors of IPv6. In Zenmap, the -6 option, along with the -O option, is used to perform OS discovery via IPv6 fingerprinting.

```
nmap -6 -O <target>
```

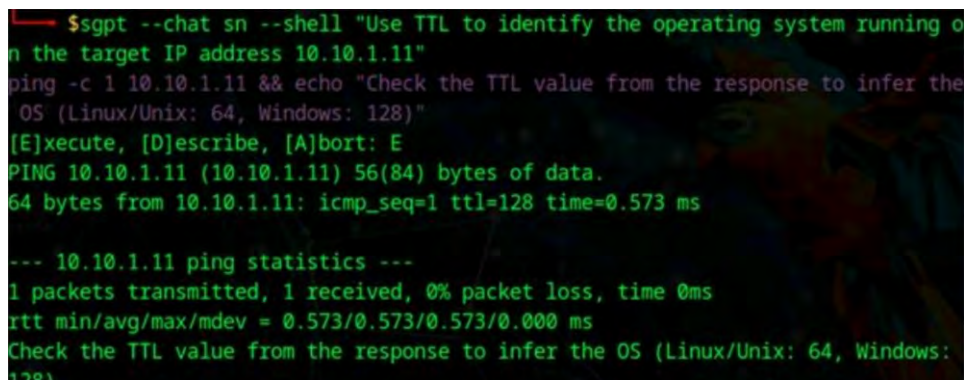
OS Discovery with AI

Attackers can leverage AI-powered technologies, such as ChatGPT, to automate and streamline OS discovery tasks. By utilizing AI, attackers can quickly identify the operating systems running on target IP addresses in a network.

Example:

The attacker uses ChatGPT to use TTL ping commands to find the target OS, such as the following prompt:

"Use TTL to identify the operating system running on the target IP address 10.10.1.11."



```
$sgpt --chat sn --shell "Use TTL to identify the operating system running on the target IP address 10.10.1.11"
ping -c 1 10.10.1.11 && echo "Check the TTL value from the response to infer the OS (Linux/Unix: 64, Windows: 128)"
[E]xecute, [D]escribe, [A]bort: E
PING 10.10.1.11 (10.10.1.11) 56(84) bytes of data:
64 bytes from 10.10.1.11: icmp_seq=1 ttl=128 time=0.573 ms

--- 10.10.1.11 ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 0.573/0.573/0.573/0.000 ms
Check the TTL value from the response to infer the OS (Linux/Unix: 64, Windows: 128)
```

Figure 3-44: Identify an OS Using TTL Values

```
ping -c 1 10.10.1.11 && echo "Check the TTL value from the response to infer the OS (Linux/Unix: 64, Windows: 128)"
```

The command sequence begins with `ping -c 1 10.10.1.11`, which sends a single ICMP echo request to the target IP address (10.10.1.11) to verify if the host is reachable. If the ping is successful, indicated by the use of `&&`, the subsequent command is executed. The next command, `echo "Check the TTL`

value from the response to infer the OS (Linux/Unix: 64, Windows: 128)", displays a message instructing the user to check the Time to Live (TTL) value in the response. The TTL value can help determine the operating system: a TTL of 64 is commonly associated with Linux/Unix systems, while a TTL of 128 typically corresponds to Windows systems. This process allows attackers to infer the target system's OS based on the TTL value from the ICMP response.

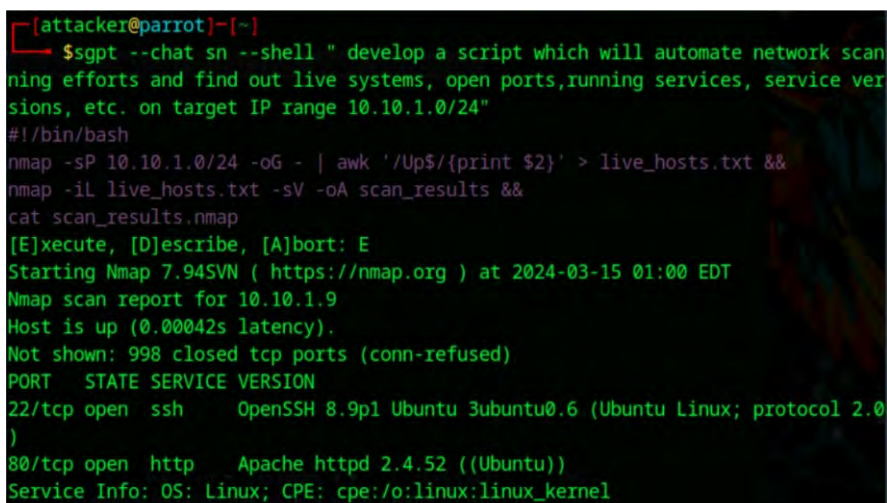
Develop and Execute a Custom Script to Automate Network Scanning Activities with AI

Attackers can leverage the power of AI to automate network scanning tasks, enhancing efficiency and enabling the execution of custom scripts that gather detailed information about target systems. By leveraging AI technologies like ChatGPT, attackers can craft tailored prompts to develop scripts that streamline the process of discovering live hosts, identifying open ports, enumerating running services, and detecting service versions within specific IP ranges.

Example:

An attacker could use a prompt such as:

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



```
[attacker@parrot]~$  
$sgpt --chat sn --shell " develop a script which will automate network scanning efforts and find out live systems, open ports, running services, service versions, etc. on target IP range 10.10.1.0/24"  
#!/bin/bash  
nmap -sP 10.10.1.0/24 -oG - | awk '/Up$/ {print $2}' > live_hosts.txt &&  
nmap -iL live_hosts.txt -sV -oA scan_results &&  
cat scan_results.nmap  
[E]xecute, [D]escribe, [A]bort: E  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-03-15 01:00 EDT  
Nmap scan report for 10.10.1.9  
Host is up (0.00042s latency).  
Not shown: 998 closed tcp ports (conn-refused)  
PORT      STATE SERVICE  
22/tcp    open  ssh      OpenSSH 8.9p1 Ubuntu 3ubuntu0.6 (Ubuntu Linux; protocol 2.0)  
80/tcp    open  http     Apache httpd 2.4.52 ((Ubuntu))  
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Figure 3-45: AI-Generated Prompt for Nmap Host Discovery Scanning

The bash script provided demonstrates how to automate the scanning process for the target IP range 10.10.1.0/24. The script first performs a ping sweep (nmap -sP 10.10.1.0/24) to identify live hosts within the specified subnet, outputting the results in a grepable format (-oG). It then extracts the IPs of live hosts using awk and saves them to a file named live_hosts.txt. Next, the script runs a service version scan (nmap -iL live_hosts.txt -sV) on these live hosts and stores the results in a series of output files. Finally, it shows the outcomes from the scan_results.nmap file by using cat. This process provides a comprehensive overview of the target network, including active systems and their services.

Scanning Beyond IDS and Firewall

Attackers use fragmentation to evade security devices such as Firewalls, IDS, and IPS. The most common and widely used technique involves breaking the payload into smaller packets. An IDS must reconstruct this stream of incoming packets to analyze and detect potential attacks. These fragmented packets are deliberately modified to complicate reassembly and hinder detection efforts. Another approach to using fragmentation involves transmitting the fragmented packets in a non-sequential order. Fragmented packets are sent out of order with intentional pauses, causing delays in reassembly.

Packet Fragmentation

Packet fragmentation is when a probe packet is split into smaller pieces while being sent over a network. These fragments, when sent to a target host, must be reassembled to form the original packet. Intrusion Detection Systems (IDS) and firewalls generally queue and process such fragments individually, which demands high CPU and network resources. To optimize performance, many IDS configurations skip inspecting fragmented packets during scans, making them vulnerable to evasion techniques. Attackers exploit this by using tools like Nmap to fragment probe packets, allowing them to bypass IDS-based port-scanning defenses. When the fragmented packets arrive at the target host, they are reassembled for further processing.

SYN/FIN Scanning Using IP Fragments

SYN/FIN scanning with IP fragments is a sophisticated variation of traditional scanning methods designed to bypass packet-filtering devices. In this technique, the TCP header is divided into multiple fragments and transmitted across the network. The initial fragment contains essential information like source and destination ports, while subsequent fragments carry flag data needed for communication. Upon reaching the target system, the Internet Protocol (IP) module reassembles these fragments using parameters such as source, destination, protocol, and identification fields. This allows attackers to avoid detection by many firewalls and Intrusion Detection Systems (IDS), making it a highly effective stealthy scanning method.

While effective, this technique can sometimes lead to unintended consequences. Improper reassembly of fragmented packets on the server side may result in unpredictable behavior, such as fragmentation of the IP header data. Some hosts may struggle to process these fragments, leading to system crashes, reboots, or even monitoring dumps in network devices.

Certain firewalls attempt to block fragmented packets using kernel-level options like `CONFIG_IP_ALWAYS_DEFRAG` in Linux. However, such measures are rarely implemented due to their performance drawbacks. Since many IDS rely on signature-based methods to detect scanning attempts, fragmentation can effectively evade detection mechanisms, increasing the risk of significant disruptions to the target network. Attackers often leverage SYN/FIN scanning with IP fragmentation to exploit this vulnerability, ensuring stealthy scans with minimal detection probability.

Zenmap, a graphical interface for Nmap, provides support for SYN/FIN scans using IP fragmentation, as demonstrated in the accompanying screenshot.

Source Routing

An IP datagram consists of multiple fields, including the IP options field, which stores source routing information. This field specifies a list of IP addresses that determine the route the packet takes to its destination. Typically, as an IP packet traverses a network, routers examine its destination IP address to determine the next hop for forwarding.

Attackers exploit the source routing mechanism by embedding malformed packets with a predefined route in the IP options field. This technique allows attackers to bypass intermediate routers or gateways configured with firewalls and Intrusion Detection Systems (IDS) that might otherwise block their packets. By enforcing a loose source routing mechanism, attackers specify a partial path for the packet, allowing some routing flexibility. In contrast, a strict source routing mechanism mandates that the packet follow an exact path defined by the attacker.

By manipulating the IP address path, attackers ensure that packets take an attacker-controlled route, evading firewall-/IDS-configured routers and effectively reaching the target system undetected. This strategy allows attackers to bypass network security controls and gain access to sensitive systems. The accompanying figure 3-37 illustrates source routing, where the originator specifies the packet's journey through the network.

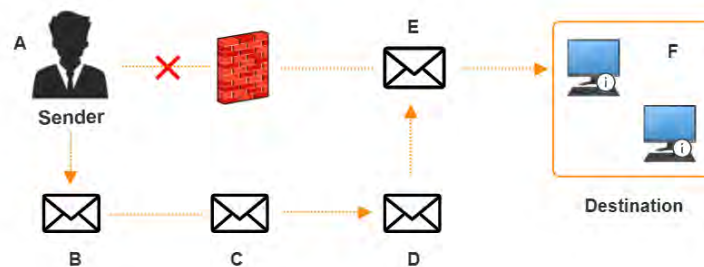


Figure 3-46: Source Routing

Source Port Manipulation

Source port manipulation is a technique attackers use to bypass firewalls and IDS. It involves altering the source port of a packet to match common, trusted port numbers such as HTTP (80), DNS (53), or FTP (21). Many firewalls are configured to allow traffic from these well-known ports, assuming they originate from legitimate services. However, attackers exploit this trust by sending malicious traffic disguised as coming from these ports.

For example, if a firewall allows all traffic from port 80 (HTTP), attackers can manipulate the source port of their packets to appear as if they are coming from port 80. This often results in the firewall permitting malicious packets to reach the target system, bypassing security controls.

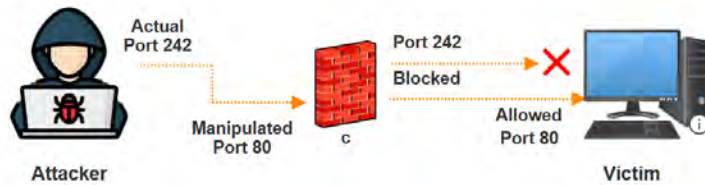


Figure 3-47: Firewall Allowing Spoofed Port 80 Traffic

While modern firewalls with application-level proxies and protocol-parsing capabilities can mitigate this threat, source port manipulation remains effective against misconfigured or outdated systems. In Zenmap, the `-g` or `--source-port` option allows users to specify a source port for this purpose, facilitating the evasion of firewall rules. This makes it a powerful tool for attackers to exploit vulnerabilities in network defenses.

IP Address Decoy

The IP address decoy technique involves generating or manually specifying fake IP addresses (decoys) to hide the true source of a network scan. Mixing decoy IPs with the attacker's real IP creates the illusion that multiple IPs are scanning the target simultaneously. This approach makes it challenging for IDS or firewalls to determine the actual scanning source, complicating their analysis and response.

Nmap, a popular network scanning tool, supports decoy scans with built-in options. Two primary methods are:

- Using the `-D RND:<number>` option, Nmap automatically generates a specified number of random decoy IPs. For example:
`nmap -D RND:10 [target]`
In this command, Nmap generates 10 decoy IPs and intermixes them with the real scanning IP.
- The `-D` option allows manual specification of decoy IP addresses. Optionally, the `ME` keyword can be used to position the attacker's real IP at a chosen spot in the list. For example:
`nmap -D decoy1,decoy2,decoy3,...,ME,... [target]`
Here, the attacker's real IP is inserted in the position marked as `ME`. If `ME` is omitted, Nmap places the real IP in a random position.

IP Address Spoofing

IP Address Spoofing is a method of gaining unauthorized access to computers by pretending to have a different IP address. An attacker illegally pretends to be any user's computer by sending altered IP packets with a fake IP address. The spoofing method includes changing a header with a fake source IP address, a checksum, and arrangement values. Packet-switched networking causes an out-of-order series of incoming packets. When these out-of-order packets are received at the destination, they are reassembled to extract the message.



EXAM TIP: Be aware of IP address spoofing techniques where attackers alter packet headers to masquerade as another system. Detection methods include checking TTL values and IP Identification (IPID).

IP spoofing can be detected by different techniques, including the direct TTL probing technique and through IP Identification Number (IPID). In sending direct TTL probes, packets are sent to the host suspected of sending spoofed packets, and responses are observed. IP spoofing can be detected by comparing TTL values from the suspected host's reply. If the TTL value is different from the one in the fake packet, it will be a spoofed packet. However, TTL values can vary even in normal traffic, and this technique identifies spoofing when the attacker is on a different subnet.

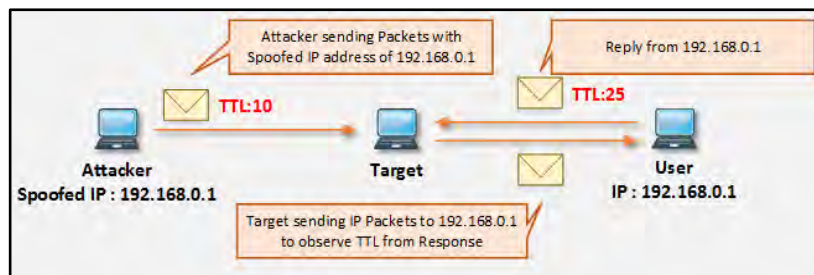


Figure 3-48: Direct TTL Probing

Similarly, additional probes are sent to verify the host's IPID. If the IPID value is not close to the recent values, the suspected traffic is spoofed. This method can be applied if the attacker is inside a subnet.

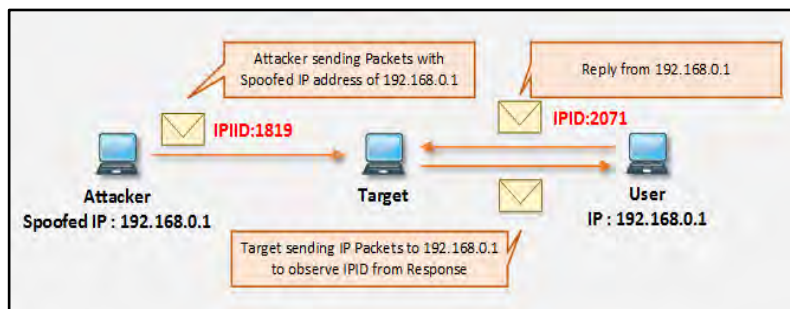


Figure 3-49: Verifying IPID Number

MAC Address Spoofing

MAC address spoofing is a technique used by attackers to bypass network firewalls that rely on source MAC address filtering. Firewalls inspect the MAC address in packet headers to verify if the packets originate from authorized devices. By spoofing MAC addresses, attackers masquerade as legitimate users to circumvent these restrictions, enabling them to scan systems behind the firewall undetected.

Attackers commonly use tools like Nmap to perform MAC address spoofing during network scans. The following options in Nmap facilitate this technique:

- **nmap -sT -Pn --spoof-mac o [Target IP]**
This command generates a random MAC address, replacing the attacker's real address in outgoing packets. The --spoof-mac o option ensures the randomization of the MAC address, helping to avoid detection and logging of the actual address.
- **nmap -sT -Pn --spoof-mac [Vendor] [Target IP]**
In this command, attackers specify a vendor, and Nmap assigns a MAC address associated with that vendor to the packets. For example, specifying a vendor like Cisco or Intel makes the scan appear as if it originates from a device manufactured by that vendor. This approach exploits trust often placed on devices from reputable vendors.
- **nmap -sT -Pn --spoof-mac [new MAC] [Target IP]**
Attackers can explicitly define a specific MAC address to use during the scan. The --spoof-mac [new MAC] option allows for complete control over the address attached to outgoing packets, ensuring it aligns with the intended deception strategy.

Creating Custom Packets

Attackers utilize custom packets to bypass IDS and firewalls, enabling them to scan and target protected networks. This process involves crafting and sending packets tailored to evade detection and overcome network defenses. Various tools, such as Colasoft Packet Builder and NetScanTools Pro, are used to create custom TCP packets for scanning purposes. These tools allow attackers to generate and send packet streams using different protocols at varying transfer rates.

Colasoft Packet Builder

Colasoft Packet Builder is a powerful tool that facilitates the creation, editing, and transmission of custom network packets. It offers three primary views for packet customization including packet list, hex editor, and decode editor. The Packet List displays all constructed packets and allows users to select packets for editing. The Hex Editor represents packet data in hexadecimal and ASCII formats, enabling direct modification. The Decode Editor simplifies editing by abstracting technical details like value length, byte order, and offsets. Attackers can create new packets using the add or insert command and control parameters such as intervals, loop counts, and delays during packet transmission.

This tool is particularly versatile, enabling users to audit network security, craft fragmented packets to bypass firewalls and IDS or flood a target with excessive packets, potentially leading to a Denial of Service (DoS) attack. While Colasoft Packet Builder is a legitimate network assessment tool, attackers can exploit its capabilities to evade detection and exploit network vulnerabilities, emphasizing the need for robust security measures.

Randomizing Host Order and Sending Bad Checksums

Randomizing Host Order

An attacker may scan the host count within a target network in a non-sequential manner to pinpoint specific targets located behind a firewall. Nmap provides the --randomize-hosts option to enable randomized scanning of hosts. This approach randomizes groups of 16,384 hosts before

scanning, particularly when slower timing parameters are used, minimizing the risk of detection by network monitoring systems and firewalls.

To randomize larger group sizes, the PING_GROUP_SZ parameter in the nmap.h file can be modified, followed by recompiling the program. Alternatively, you can create a target IP list using the list scan command (-sL -n -oN), randomize it with a Perl script, and then provide the randomized list to Nmap using the -iL option.

Sending Bad Checksums

Attackers may send packets with incorrect or false TCP/UDP checksums to bypass specific firewall rules. Both TCP and UDP use checksums to ensure data integrity, but improperly configured systems can reveal information when handling packets with invalid checksums. If a reply is received, it likely originates from an IDS or firewall that has not verified the checksum. Conversely, if there is no response or the packets are dropped, it indicates the system may be properly configured. Nmap enables this technique with the --badsum option, which sends packets containing invalid TCP, UDP, or SCTP checksums to the target host.

Proxy Servers

Proxy servers anonymize web traffic to ensure user privacy. When an individual seeks to utilize a resource on a publicly accessible server, the proxy server functions as an intermediary. It facilitates the interaction between the client and the server on behalf of the user. The request is initially routed through the proxy server, which processes it, whether it is a web page request, file download, or connection request to another server. Proxy servers are primarily used to provide access to the World Wide Web (WWW) by bypassing IP address blocking.

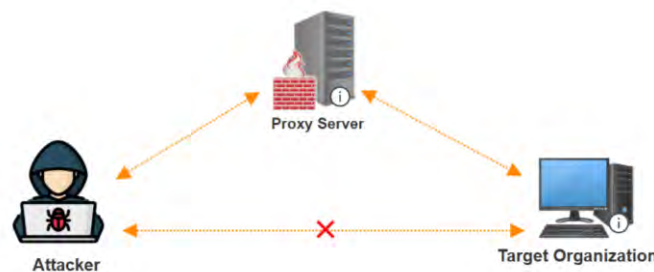


Figure 3-50: Attacker Using a Proxy Server to Connect to the Target

The uses of a proxy server, in a nutshell, can be summarized as:

- Hiding the source IP address to bypass restrictions based on IP blocking.
- Impersonating
- Remote Access to Intranet
- Redirecting all requests to the proxy server to hide the identity
- Proxy Chaining to avoid detection

Why Attackers Use Proxy Servers

Attackers often use proxy servers to mask their true identity and avoid detection. The primary objective is to hide their IP address to prevent the origin of the attack from being traced back to them. By routing their requests through a proxy, the attacker ensures that the logs on the destination server record the proxy's IP address instead of their own. This helps attackers carry out their activities without being identified. Additionally, proxies allow attackers to bypass firewalls, IDS, and other security measures that might otherwise detect their presence. Since the traffic appears to come from the proxy server and not the attacker, it becomes harder for security systems to identify malicious behavior.

Moreover, proxies provide anonymity, which is critical for attackers looking to cover their tracks while performing malicious actions. They can also help attackers access restricted websites and networks, bypassing network restrictions or geographical blocks. By using proxies, attackers can hide their actions behind layers of anonymity, making it more difficult for security teams or law enforcement to trace the attack's origin.



EXAM TIP: Attackers use proxy servers to hide their true IP and bypass network defenses. Proxy chaining involves using multiple proxies to further hide the attacker's identity.

Free Proxy Servers

Some free proxy servers are readily available on the internet, allowing users to browse the web anonymously without revealing their true IP address. Attackers often use these proxies to evade detection while conducting their attacks. In some cases, attackers even chain multiple proxies together, further complicating efforts to trace their activities. By employing these techniques, attackers can execute their malicious actions with relative impunity, relying on proxies to conceal their identity and avoid being caught.

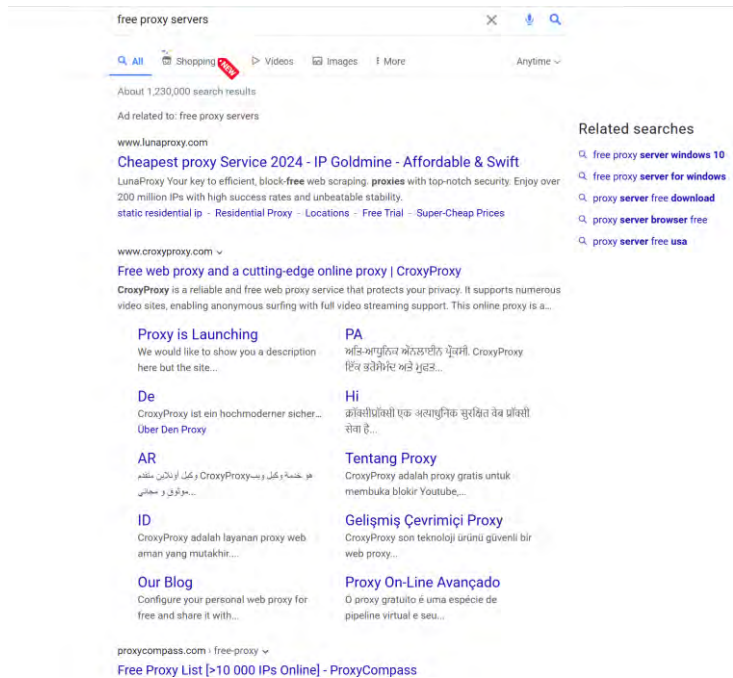


Figure 3-51: Free Proxy Servers

Proxy Chaining

Proxy Chaining is basically a technique for using multiple proxy servers. In proxy chaining, one proxy server forwards the traffic to the next proxy server in the chain. This process is not recommended for production environments, nor is it a long-term solution. However, this technique leverages your existing proxy.

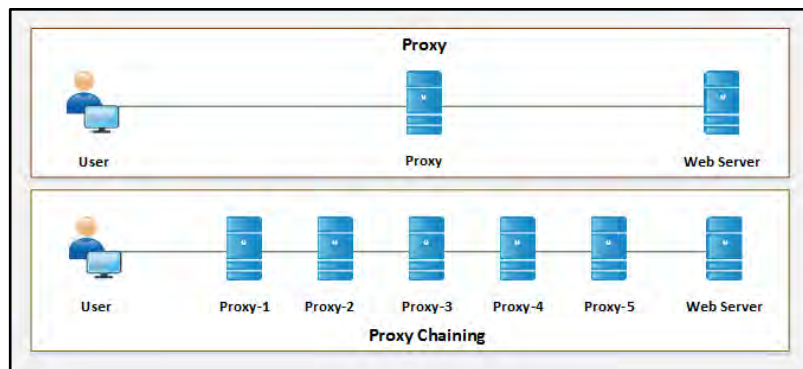


Figure 3-52: Proxy Chaining

Proxy Tool

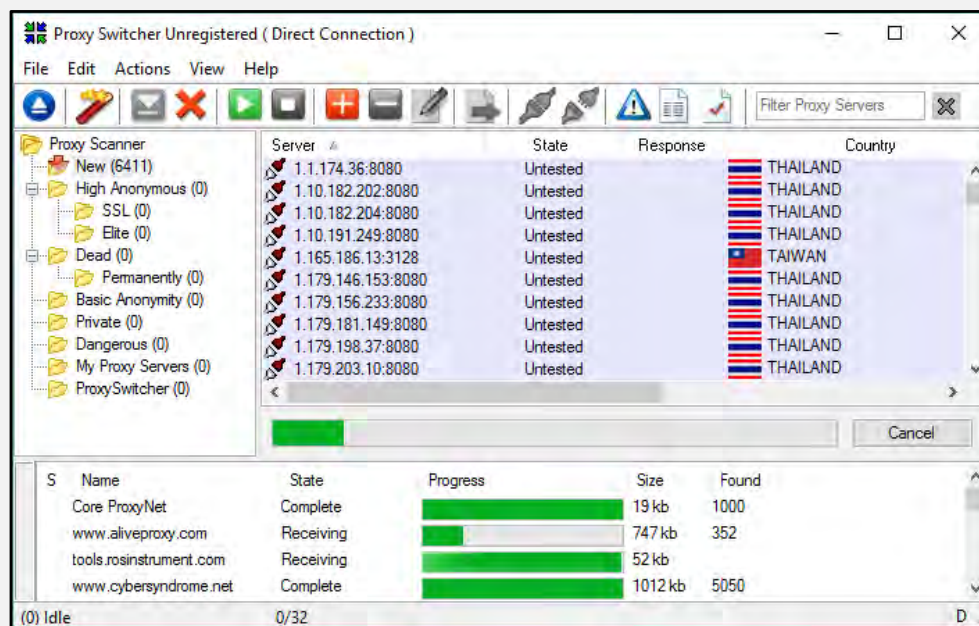
There are several proxy tools you can find, and you can also look online for a proxy server and set it up yourself on your web browser. Available proxy tools include:

1. Proxy Switcher
2. Proxy Workbench
3. TOR

4. CyberGhost VPN

Proxy Switcher

A Proxy Switcher tool scans for the available proxy servers. Any proxy server can be used to mask your IP address. The figure below shows the proxy server search process performed by the Proxy Switcher tool.



Proxy Tools for Mobile

Several proxy applications are available on Google Play Store and App Store for Android and iOS devices.

Application	Download URL
Proxy Droid	https://play.google.com
Net Shade	https://itunes.apple.com

Table 3-04: Proxy Tools for Mobile

Anonymizers

Anonymizer is a tool that fully hides or gets clear of identity-related details to make actions impossible to track. The main reasons for using anonymizers are to reduce risk, spot and stop information theft, get around limits and censorship, and perform activities online that cannot be tracked.

Why Use an Anonymizer?

- **Ensuring Privacy:** Anonymizers make browsing activities untraceable, protecting user identity until personal information is voluntarily disclosed.

- **Accessing Restricted Content:** Governments often restrict access to sensitive content or websites. An anonymizer located outside the target country can circumvent these restrictions.
- **Protection Against Online Attacks:** Routing internet traffic through protected DNS servers and anonymizers can shield users from online phishing attacks.
- **Bypassing IDS and Firewalls:** By connecting to websites through the anonymizer's address, users can evade organizational firewall rules, appearing as if they are only connected to the anonymizer.

Types of Anonymizers

There are two types of anonymizers include:

1. Networked Anonymizers

It functions by routing user information through a series of interconnected computers before it reaches the target website. This multi-node process significantly complicates traffic analysis, making it difficult for anyone to trace the connection back to the original user. For example, if a user requests to visit a web page, the request is first sent through intermediary computers, such as A, B, and C, before finally reaching the destination website.

- **Advantage:** The layered routing structure makes traffic analysis highly complex, enhancing user anonymity.
- **Disadvantage:** Each intermediary node in the communication path introduces a potential vulnerability, as confidentiality may be compromised at any of these points.

2. Single-Point Anonymizers

It works by routing user information through a single intermediary website before forwarding it to the target website. The target website's response is then passed back to the user through the same intermediary. This process ensures the user's identity, including their IP address, remains hidden from the target website.

- **Advantage:** The direct routing method ensures arms-length communication, effectively masking the user's IP address and other identifying information.
- **Disadvantage:** Compared to networked anonymizers, single-point anonymizers provide less resistance to advanced traffic analysis techniques, which could potentially expose the user's activity.

Anonymizer Tools

Anonymizers utilize technologies like SSH, VPNs, and HTTP proxies to allow access to blocked or censored internet content while often omitting advertisements.

Whonix:

Whonix is a specialized desktop operating system designed for high-level security and privacy. It operates on a reconfigured Debian base within virtual machines and uses the Tor network for online

anonymity. This setup mitigates malware threats and IP leaks while maintaining usability, making it a robust tool for anonymous browsing and security.



Figure 3-53: Screenshot of Whonix

Censorship Circumvention Tool

Tails

The Amnesic Incognito Live System (Tails) is a well-known tool for bypassing censorship built on Debian GNU/Linux. It is basically a live Operating System (OS) that can run on almost every computer via a USB or DVD. Tails is specially designed to help you use the internet anonymously, leaving no trace behind. It preserves privacy and anonymity.

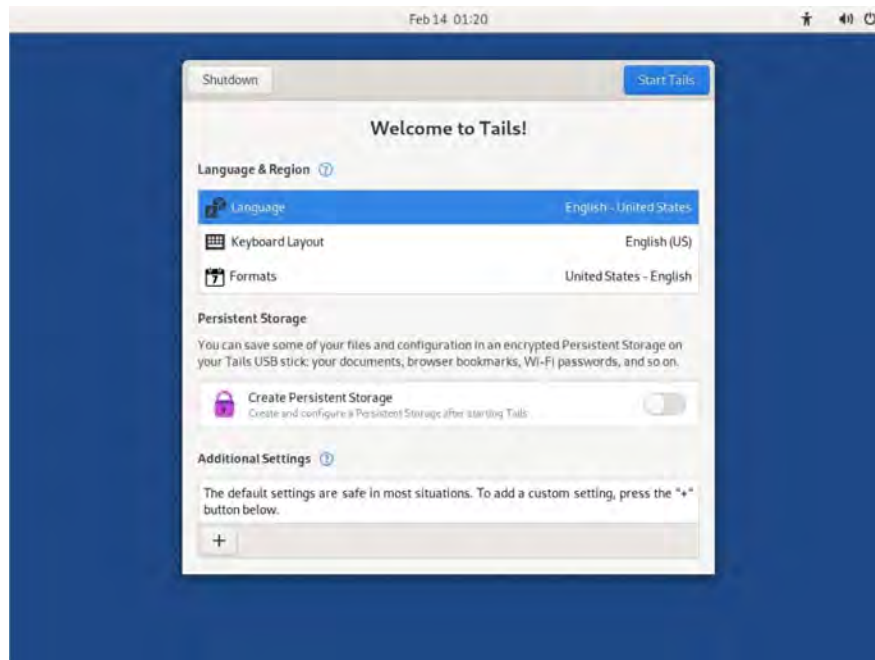


Figure 3-54: Screenshot of Tails

AstrillVPN

Another tool for circumventing censorship is AstrillVPN. It is Virtual Private Network (VPN) software that facilitates bypassing Internet censorship and accessing geo-blocked websites, applications, and services by concealing the user's IP address and location. It utilizes advanced data encryption and secure transmission methods to safeguard user information. Additionally, AstrillVPN does not log traffic data or DNS queries, ensuring that browsing activity and metadata remain untraceable, thus offering robust privacy and security features.

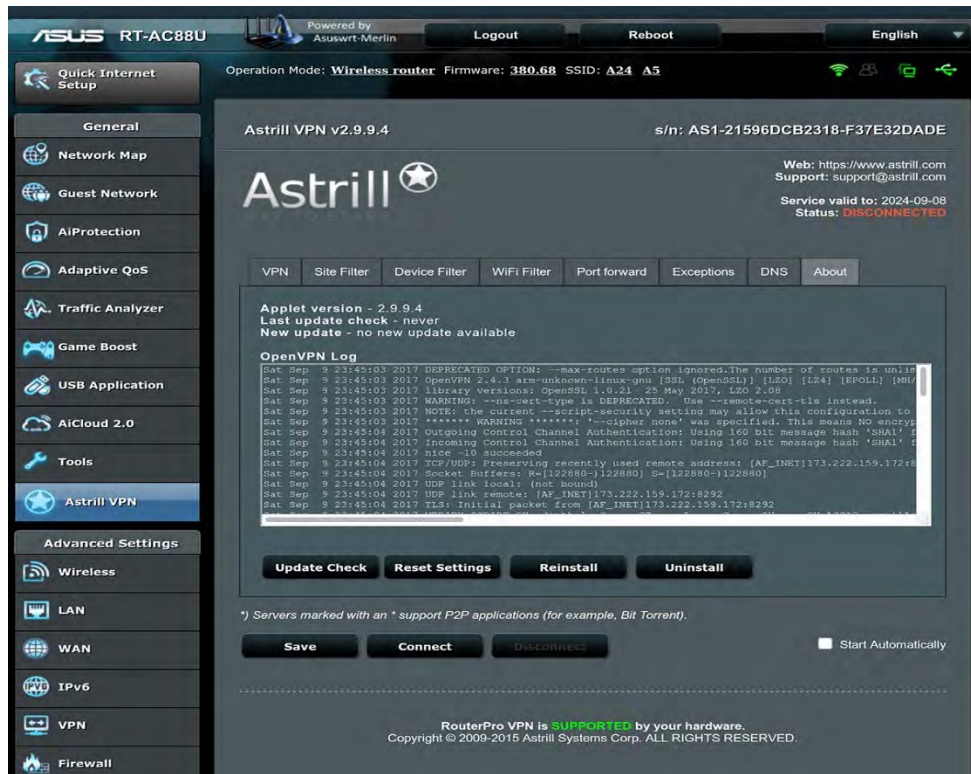


Figure 3-55: Screenshot of AstrillVPN

Anonymizers for Mobile

- Orbot
- Psiphon
- Open Door

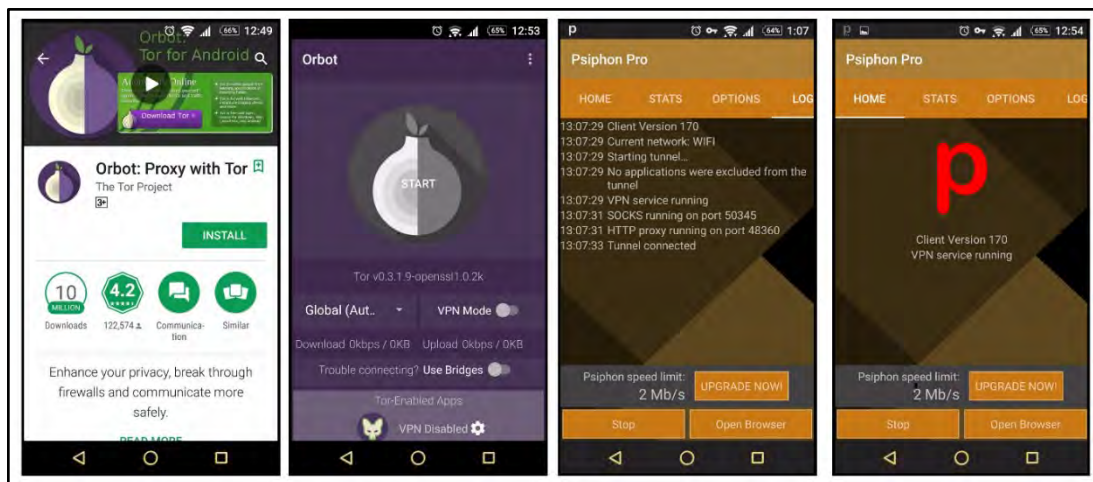


Figure 3-56: Anonymizers for Mobile

Network Scanning Countermeasures

In ethical hacking, the ethical hacker, often referred to as the "pen tester," is responsible for an additional critical task beyond merely identifying vulnerabilities—implementing countermeasures to address the security weaknesses discovered. This step is essential because identifying security loopholes in a network serves little purpose unless proactive measures are taken to fortify it against real-world attackers. Adopting countermeasures ensures that the network remains secure, resilient, and better protected from potential scanning attacks and intrusions. The following section outlines various strategies and practices to defend against network scanning attacks effectively.

Ping Sweep Countermeasures

It is essential to defend against reconnaissance activities that attempt to map out a network by sending ICMP echo requests (ping). To mitigate the risks associated with ping sweeps, organizations can adopt the following countermeasures:

- Configure firewalls to block incoming ICMP echo requests from unknown or untrusted sources to prevent unauthorized scanning attempts.
- Implement IDS/IPS solutions like Snort to detect and block ping sweep activities by identifying suspicious scanning patterns.
- Monitor and evaluate the types of ICMP traffic passing through enterprise networks to ensure only legitimate traffic is allowed.
- Automatically terminate connections with any host that sends more than 10 ICMP echo requests in a short time span to prevent flooding.
- Set up a DMZ to allow only specific ICMP commands such as ECHO_REPLY, HOST_UNREACHABLE, and TIME_EXCEEDED, restricting attackers' ability to map network resources.
- Use ACLs to limit ICMP traffic to specific IP addresses from the ISP, controlling unnecessary ICMP communication.
- Implement rate limiting on ICMP packets to slow down or stop ping sweeps, making it harder for attackers to gather information.
- Divide the network into smaller, isolated segments to limit what attackers can discover and reduce potential damage from a compromised network.
- Deploy private IP addresses for internal devices and use NAT at the network perimeter to prevent external identification of internal IP addresses.

Port Scanning Countermeasures

Port scanning can provide attackers with critical information, such as IP addresses, host names, open ports, and services running on those ports. Open ports are particularly vulnerable, as they can serve as entry points for attackers. However, implementing effective countermeasures can secure a system or network against port scanning attempts. These countermeasures include:

- Establish firewall and Intrusion Detection System (IDS) regulations to identify and prevent port scanning attempts.
- Ensure the firewall inspects the data within each packet, not just the TCP header, to detect port scanning attempts.
- Test port scanning tools against hosts on the network to ensure firewalls can detect port scanning activity.
- Keep router, IDS, and firewall firmware updated with the latest releases.
- Configure firewalls to protect against fast port scans and SYN floods.
- Employ an IDS to detect OS detection attempts by hackers using tools like Nmap.
- Limit the number of open ports and filter the rest to reduce the risk of exploitation.
- Use custom firewall rules to block unwanted ports, including 135-159, 256-258, 389, 445, 1080, 1745, and 3268.
- Block unwanted services on open ports and update service versions to non-vulnerable versions.
- Configure border routers to block inbound ICMP messages and outbound ICMP Type 3 (Destination Unreachable) messages.
- Ensure firewalls and routers can block source-routing attempts by attackers.
- Prevent the bypassing of routing and filtering mechanisms through the use of source routing or specific source ports.
- Test the network configuration with TCP and UDP port scans and ICMP probes to identify accessible ports.
- Configure anti-scanning and anti-spoofing rules to protect against scanning attempts.
- Ensure commercial firewalls are patched, have defined anti-spoofing rules, and disable fast-mode services.
- Use TCP wrappers to limit access based on domain names or IP addresses.
- Implement proxy servers to block fragmented or malformed packets.
- Conduct open port scans on empty hosts or honeypots to confuse or mislead port scanning efforts.
- Use an IPS to detect port scans and blacklist malicious IP addresses.
- Implement port knocking to conceal open ports from unauthorized access.
- Implement egress filtering to control outbound traffic and detect internal hosts scanning external targets.
- Use VLANs to isolate and restrict access between different types of traffic.

IP Spoofing Detection Techniques

There are some common techniques for detecting IP spoofing include:

Direct TTL Probes

In the Direct TTL Probes technique, a packet (ping request) is initially sent to a legitimate host, and the Time-to-Live (TTL) value of the reply is compared to the TTL value of the original packet. Both values should match if the same protocol is being used. Typical initial TTL values for different protocols are generally 64 and 128 for TCP/UDP, and 128 and 255 for ICMP.



Figure 3-57: IP Spoofing Detection Technique: Direct TTL Probes

To identify spoofed packets, the hop count is determined by taking the difference between the TTL (Time to Live) value in the response and the original TTL value in the outgoing packet. If the reply TTL does not match the original packet's TTL, it suggests that the packet is spoofed. However, if the attacker is aware of the hop count between the source and the target, they could manipulate the TTL to avoid detection, leading to a potential false negative. This technique is more effective when the attacker is in a different subnet from the victim, as the difference in hop count is more detectable.

IP Identification Number

Users can spoof fake packets by watching the IP Identification (IPID) number in the IP packet headers. The IPID increases by one with each new packet transmitted by the system. Every IP packet on a network is assigned a unique IP identification number, which increases incrementally with each transmission. To find spoofed packets, a test packet is sent to the source IP address of the packet being checked, and the IPID number in the response is examined. The IPID value in the response should be slightly greater than, but close to, the IPID value of the probe packet. If the response packet's IPID is not close to the probe's IPID, it suggests that the source address is spoofed. This technique is particularly effective even when the attacker and target are on the same subnet.

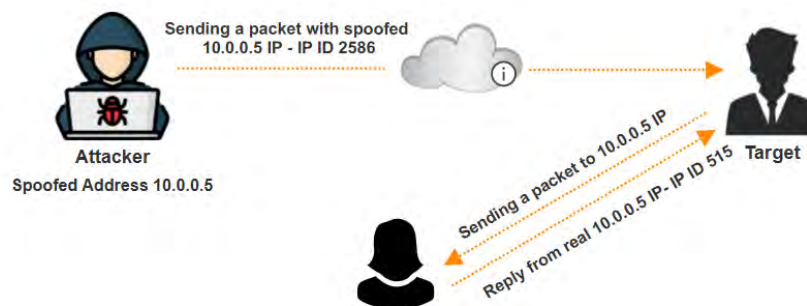


Figure 3-58: IP Spoofing Detection Technique: IP Identification Number

TCP Flow Control Method

TCP improves flow control for both the sender and the receiver by using the sliding window technique. This method allows control over the flow of IP packets through the window size field in the TCP header. The window size represents the maximum amount of data the receiver can accept, and the sender can transmit without waiting for an acknowledgment (ACK).

Under normal flow control, the sender halts transmission when the initial window size is fully utilized. However, an attacker unaware of the updated ACK packet containing window size information might continue sending data. Any data packets received beyond the allowed window size can be identified as spoofed. To enhance flow control and detect spoofed packets early, it is advisable to set the initial window size to a small value.

Spoofing attacks are most common during the TCP handshake phase, as crafting multiple spoofed responses with correct sequence numbers is challenging. To mitigate such attacks, a flow control-based spoofed packet detection method can be applied during the handshake process.

In a typical TCP handshake, the process begins when the host sends an initial SYN packet to initiate the connection with the target server. The server replies with a SYN-ACK packet. This indicates its readiness to initiate the connection. Finally, the host completes the handshake by replying with an ACK packet, confirming the connection establishment. To detect spoofed SYN requests, configure the SYN-ACK packet with a zero window size. If the sender replies with an ACK containing additional data, the sender is likely spoofed. Legitimate clients should respond to a zero-window SYN-ACK with an ACK containing no data.

Furthermore, attackers generating spoofed packets cannot adapt to changes in the congestion window size because they do not receive the SYN-ACK response from the target. If traffic persists after the congestion window size is depleted, those packets are likely spoofed. Implementing these methods during the handshake and monitoring the window size can significantly reduce the risk of spoofing attacks.

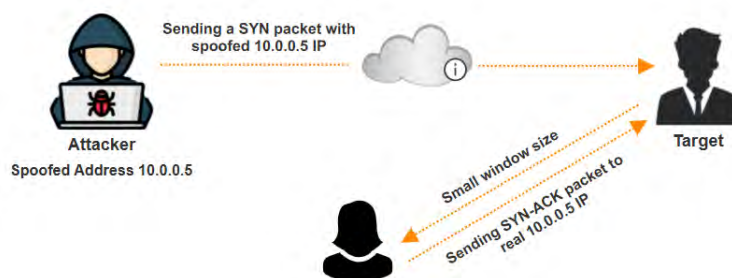


Figure 3-59: IP Spoofing Detection Technique: TCP Flow Control Method

IP Spoofing Countermeasures

IP spoofing is a hacking technique where attackers impersonate trusted devices to infiltrate target networks. To safeguard against such threats, robust countermeasures should be integrated into network security protocols. Some key measures include:

1. Avoid Trust Relationships:

Relying solely on IP-based authentication can expose networks to spoofing attacks. Attackers may pose as trusted hosts, sending malicious packets that are mistakenly considered safe. To mitigate this, all packets should be thoroughly verified, even those from trusted hosts. Strengthening trust-based authentication with password protection and multi-factor authentication significantly enhances security.

2. Use Firewalls and Filtering Mechanisms:

Firewalls play a critical role in filtering both incoming and outgoing packets, preventing unauthorized access, and minimizing data breaches. Access Control Lists (ACLs) can further restrict network access by blocking malicious packets. However, insider threats remain a concern, as internal actors may leak sensitive information to competitors. Outgoing packet monitoring is equally important, as attackers may deploy hidden malicious programs, such as sniffers, to exfiltrate data. Properly scanning and filtering outgoing packets can prevent these threats, ensuring comprehensive network protection.

3. Use Random Initial Sequence Numbers (ISNs):

Many devices use predictable ISNs based on timed counters, making them vulnerable to attackers who can analyze and predict ISNs for future connections. To mitigate this, randomizing ISNs ensures unpredictability, preventing attackers from hijacking TCP connections or sniffing network traffic.

4. Ingress Filtering:

Ingress filtering blocks spoofed traffic at its entry point, particularly at routers. By configuring Access Control Lists (ACLs) to drop packets with invalid source addresses, ingress filtering enhances router functionality and prevents unauthorized traffic from entering the network.

5. Egress Filtering:

Egress filtering targets outgoing packets, ensuring that packets leaving the network carry valid source addresses. This practice prevents spoofed packets from being sent and reduces the risk of compromised internal devices participating in attacks.

6. Use Encryption:

Encrypting all transmitted data is one of the most effective methods to prevent IP spoofing. Strong encryption protocols, such as IPSec, ensure data authentication, integrity, and confidentiality. Encryption sessions on routers secure communication between trusted hosts and local networks. Attackers face significant challenges decrypting encrypted data, often forcing them to abandon their attempts. Regularly updating to the latest encryption algorithms further strengthens security.

7. SYN Flooding Countermeasures:

Since IP spoofing is often used in SYN flooding attacks, implementing SYN flood countermeasures (such as SYN cookies) also helps mitigate spoofing risks by managing connection requests more effectively.

8. Other Measures:

- Strengthen website integrity and confidentiality by transitioning from IPv4 to IPv6 during the development phase, leveraging IPv6's advanced security features.
- Implement digital certificate authentication methods, including domain-based and two-way certificate verification, to ensure secure communication and user/device authentication.
- Utilize a secure VPN when accessing public Internet services, such as free Wi-Fi and hotspots, to safeguard data and prevent unauthorized interception.
- Deploy specialized mitigation devices, such as Behemoth scrubbers, for high-speed, deep-packet inspection capable of analyzing up to 100 million packets per second to detect and block malicious traffic effectively.
- Implement dynamic IPv6 address variation by using a random address generator to minimize the duration of active vulnerabilities.
- Configure routers to encode information about fragmented packets entering the network for enhanced monitoring and analysis.
- Enable routers to verify incoming data packets by comparing their signatures with stored packet digests.
- Use Network Address Translation (NAT) modifications to conceal intranet hosts from the external network, improving privacy and security.
- Configure internal switches to maintain DHCP static address tables, effectively filtering out malicious spoofed traffic.
- Employ secure versions of communication protocols, such as HTTPS, SFTP, and SSH, which provide encryption and authentication to protect data in transit.

Scanning Detection and Prevention Tools

Security professionals utilize advanced tools to detect network and port scanning attempts initiated by attackers. These tools provide comprehensive visibility, real-time detection, and effective prevention mechanisms to secure networks against malicious activities.

1. ExtraHop

ExtraHop offers complete visibility into network activities, enabling real-time detection and intelligent responses to malicious scanning attempts. It can automatically discover and classify all devices, including unmanaged IoT devices while identifying vulnerabilities. ExtraHop analyzes all network interactions in real-time, including cloud transactions and encrypted SSL/TLS traffic, to ensure comprehensive visibility. Its auto-discovery and device classification capabilities allow security teams to monitor and evaluate all network communications effectively.



Figure 3-60: ExtraHop

There are many other tools available for detecting and preventing network scanning, which help enhance network security by identifying scanning attempts and providing real-time responses. These tools offer advanced analytics, threat detection, and automated prevention to safeguard against malicious activities. Some notable tools include:

- Splunk Enterprise Security
- Scanlogd
- Vectra Detect
- IBM Security QRadar XDR
- Cynet 360 AutoXDR

Summary

Network scanning helps identify active hosts, open ports, and vulnerabilities using tools like Nmap and Nessus.

Host discovery techniques, such as ICMP and ARP scanning, identify active devices, whereas port and service discovery uncover potential vulnerabilities.

OS discovery techniques, such as banner grabbing, can help identify the operating system of devices on the network.

Scanning beyond IDS and firewalls involves employing techniques such as decoy scans and IP fragmentation to circumvent detection.

Countermeasures such as firewalls, IDS/IPS, ACLs, encryption, and network segmentation can prevent unauthorized scans and protect against scanning attempts.

Rate-limiting, NAT, and regular updates to network defenses enhance protection against malicious scans.

MindMap



Figure 3-61: Mind Map of Scanning Networks

Practice Questions

1. What is the primary purpose of OS discovery (banner grabbing)?
 - A. Determine OS and services on a target system.
 - B. Install malicious programs on the system.
 - C. Bypass firewalls and network security.
 - D. Overload the network with excess traffic.

2. What is the main difference between active and passive banner grabbing?

- A. Active sends probes; passive monitors traffic.
 - B. Active works on open ports; passive on closed ports.
 - C. Active is stealthy; passive is easily detected.
 - D. Active uses UDP; passive uses TCP protocols.
3. What is the purpose of SSDP in network scanning?
- A. Scan UPnP networks for active devices.
 - B. Detect devices without DHCP or DNS.
 - C. Map open ports on IPv4 and IPv6 networks.
 - D. Find filtering systems in firewalls.
4. What is a limitation of the SCTP COOKIE ECHO scan?
- A. Cannot differentiate between open and filtered ports.
 - B. Is easily detected by security systems.
 - C. Needs root privileges for execution.
 - D. Does not work on firewalled networks.
5. What indication suggests that a port is not open during a UDP scan?
- A. No reply from the target system.
 - B. SYN+ACK response from the port.
 - C. RST packet sent by the system.
 - D. ICMP Port Unreachable message.
6. During the scanning phase, fingerprinting refers to:
- A. Mapping devices on a network.
 - B. Detecting OS and architecture.
 - C. Finding services on the system.
 - D. Closing unnecessary open ports.
7. Which of the following best describes the primary objective of the network scanning phase?
- A. Exploit vulnerabilities on the system.
 - B. Find open ports and running services.
 - C. Patch vulnerabilities on devices.
 - D. Document the network topology.
8. Which TCP flag is used to terminate a connection properly?
- A. RST
 - B. FIN
 - C. URG
 - D. PSH

9. What information can Nmap NOT provide?
 - A. Open ports
 - B. Network topology
 - C. Hardware vulnerabilities
 - D. OS version

10. Why is UDP considered a "connectionless" protocol?
 - A. It does not ensure delivery or reliability.
 - B. It establishes a handshake before communication.
 - C. It provides error recovery functions.
 - D. It requires less header information than TCP.

11. What is the significance of Hping3's IP spoofing feature?
 - A. It identifies the target's OS.
 - B. It bypasses firewalls for scanning.
 - C. It anonymizes the source of probes.
 - D. It facilitates packet fragmentation.

12. Which type of scan involves sending ICMP ECHO requests to various IP addresses within a subnet to detect active hosts?
 - A. ICMP Timestamp Ping Scan
 - B. ICMP Address Mask Ping Scan
 - C. ICMP ECHO Ping Sweep
 - D. TCP ACK Ping Scan

13. What does the Nmap command `nmap -sX -v <ip address or range>` perform?
 - A. Xmas Scan
 - B. Null Scan
 - C. FIN Scan
 - D. Stealth Scan

14. What happens when an Xmas Scan packet is sent to an open port on a modern system?
 - A. The port responds with a SYN-ACK packet.
 - B. The port sends an RST packet.
 - C. The request is ignored or dropped.
 - D. The port sends an ICMP echo reply.

15. Which scanning technique is most effective for stealthy port scanning?
 - A. UDP Scan

- B. TCP Connect Scan
- C. SCTP INIT Scan
- D. ACK Flag Probe Scan

16. Which scanning option in Zenmap is used to generate a list of hosts without scanning them?

- A. -sL
- B. -sU
- C. -sS
- D. -sY

17. What type of response does an open port provide during an SCTP COOKIE ECHO scan?

- A. ABORT chunk
- B. No response
- C. SYN+ACK chunk
- D. ICMP Unreachable error

18. Which of the following fields is commonly analyzed for OS discovery?

- A. TTL and TCP Window Size
- B. Source IP address and MAC address
- C. Packet fragmentation offset and checksum
- D. ARP and DNS query fields

19. What does a TTL value of 128 typically indicate during OS discovery?

- A. Linux
- B. macOS
- C. Unix
- D. Windows

20. Which Nmap feature automates OS discovery using pre-written scripts?

- A. Zenmap
- B. Nmap Scripting Engine (NSE)
- C. IPv6 Probes
- D. Packet Fragmentation

21. Which method exploits the IP options field to bypass network security controls?

- A. Source routing
- B. SYN/FIN scanning
- C. Fragmentation offset manipulation
- D. Proxy chaining

22. What is the main advantage of IP address spoofing?
- A. Increasing packet delivery speed
 - B. Concealing the attacker's identity
 - C. Reducing network traffic
 - D. Encrypting the packet data
23. What is the function of proxy chaining?
- A. Improving packet encryption
 - B. Using multiple proxies to hide the attacker's origin
 - C. Increasing data throughput
 - D. Reducing latency in packet delivery
24. What is the primary benefit of using anonymizers?
- A. To increase the packet transmission rate
 - B. To improve connection stability
 - C. To enhance the packet integrity
 - D. To hide the user's identity and bypass restrictions
25. What is packet fragmentation used for in network security?
- A. To increase network speed
 - B. To bypass Intrusion Detection Systems (IDS) and Firewalls
 - C. To enhance packet reassembly
 - D. To improve encryption techniques

Answers

1. Answer: A

Explanation: OS discovery focuses on identifying the operating system and active services on a target system. This information enables attackers to identify vulnerabilities specific to the detected OS.

2. Answer: A

Explanation: Active banner grabbing sends crafted packets to the target system, eliciting responses to identify OS or services. In contrast, passive banner grabbing relies on analyzing traffic without engaging the target, making it harder to detect but potentially less informative.

3. Answer: B

Explanation: SSDP enables the discovery of devices using Plug and Play capabilities without requiring DHCP or DNS configurations. It simplifies network scanning by identifying devices through multicast queries, especially in local environments with multiple networked devices.

4. Answer: A

Explanation: The SCTP COOKIE ECHO scan is unable to differentiate between open and filtered ports reliably. This limitation occurs because both open and filtered ports often respond in a way that appears similar, reducing the scan's effectiveness in detailed port analysis.

5. Answer: D

Explanation: In a UDP scan, closed ports typically send an ICMP Port Unreachable message. Conversely, open ports usually remain silent, making it challenging to differentiate open ports from those filtered by firewalls.

6. Answer: B

Explanation: Fingerprinting involves identifying the operating system and system architecture of a target machine. Tools like Nmap perform this analysis by examining response patterns, such as TTL values or TCP window sizes, to deduce the OS type and version.

7. Answer: B

Explanation: The network scanning phase identifies open ports and services to gather critical reconnaissance information. This allows attackers to map potential attack vectors, paving the way for targeted exploitation of discovered vulnerabilities.

8. Answer: B

Explanation: The FIN (Finish) flag is sent when a connection needs to terminate gracefully, ensuring both sides agree on ending the session. This controlled closure minimizes disruption and ensures data integrity.

9. Answer: C

Explanation: Nmap specializes in identifying open ports, network layout, and OS details but cannot analyze hardware vulnerabilities. Such tasks require specialized tools like Nessus or vulnerability scanners focused on hardware.

10. Answer: A

Explanation: UDP is termed "connectionless" because it does not establish connections or verify data delivery. It lacks features like acknowledgment, retransmission, and error correction, making it faster but less reliable than TCP.

11. Answer: C

Explanation: Hping3's IP spoofing feature allows users to mask the source IP address of network probes, making it more difficult for security systems to trace the origin of the scan. This technique enhances anonymity during reconnaissance or attack activities.

12. Answer: C

Explanation: The ICMP ECHO Ping Sweep sends ICMP Echo requests (ping) to all IP addresses in a subnet. Hosts that respond are considered live, providing valuable information about which systems are active in a network.

13. Answer: A

Explanation: The -sX option in Nmap triggers an Xmas Scan, which sends TCP packets with the FIN, URG, and PSH flags set. The scan's name comes from the "Xmas tree" of flags, which can confuse certain firewalls and intrusion detection systems.

14. Answer: C

Explanation: Modern systems typically ignore or drop Xmas Scan packets because the combination of flags is invalid and not a standard request. As a result, the scan often goes undetected or fails to elicit useful responses.

15. Answer: C

Explanation: The SCTP INIT Scan is a stealthy scanning method that only sends an INIT chunk (connection request) and does not complete the handshake, making it less detectable by IDS or firewalls.

16. Answer: A

Explanation: The—SL option in Zenmap performs a list scan that identifies hosts within a target range but does not actively scan or probe them. This is useful for gathering a list of hosts without triggering network alerts.

17. Answer: B

Explanation: In an SCTP COOKIE ECHO Scan, an open port generally does not respond, dropping the packet without any reply. This lack of response helps attackers infer that the port may be open or filtered.

18. Answer: A

Explanation: The Time to Live (TTL) and TCP Window Size are key values used in OS discovery because they are specific to the operating system's networking behavior. By analyzing these, tools like Nmap can determine the operating system running on the target system.

19. Answer: D

Explanation: A TTL value of 128 is most commonly associated with Windows operating systems. This value is part of how Windows handles packet routing, and it's often used as a fingerprint for OS detection.

20. Answer: B

Explanation: The Nmap Scripting Engine (NSE) uses pre-written scripts to automate tasks like OS discovery. Scripts like smb-os-discovery help quickly identify the operating system and services without manual intervention, improving scanning efficiency.

21. Answer: A

Explanation: Source routing involves manipulating the IP options field to specify a path that the packet should take across the network. This technique allows packets to bypass intermediate security devices, like firewalls, making it useful for stealth scanning.

22. Answer: B

Explanation: IP address spoofing enables an attacker to alter the source IP address in a packet header, making the traffic seem as though it originates from a trusted device. This helps the attacker remain anonymous and avoid detection by firewalls or IDS, as the true origin of the attack is hidden.

23. Answer: B

Explanation: Proxy chaining is a technique where an attacker routes their internet traffic through multiple proxies (intermediate servers). This makes it much more difficult for anyone monitoring the traffic to trace it back to the source, as each proxy hides the true origin of the request, thus adding an additional layer of anonymity.

24. Answer: D

Explanation: Anonymizers, such as Tor or VPNs, help mask a user's real IP address and location by routing traffic through intermediate servers or networks. This allows the user to remain anonymous online and bypass geographic or network-based restrictions, such as access to blocked content or websites.

25. Answer: B

Explanation: Packet fragmentation involves breaking a large packet into smaller pieces to evade detection by IDS or firewalls. These security devices typically scan packets for malicious content, but fragmented packets may not be fully reassembled, allowing malicious payloads to slip through undetected. This method is often used in network attacks to avoid security measures.