**🡺 Network Scanning**

Scanning is the process of gathering additional detailed information about the target using highly complex and aggressive reconnaissance techniques. Network scanning refers to a set of procedures used for identifying hosts, ports, and services in a network. Network scanning is also used for discovering active machines in a network and identifying the OS running on the target machine.

In the scanning phase of an attack, the attacker tries to find various ways to intrude into a target system. The attacker also tries to discover more information about the target system to determine the presence of any configuration lapses

**Types of Scanning**

* **Port Scanning:** Lists the open ports and services. Port scanning is the process of checking the services running on the target computer by sending a sequence of messages in an attempt to break in. Port scanning involves connecting to or probing TCP and UDP ports of the target system to determine whether the services are running or are in a listening state. The listening state provides information about the OS and the application currently in use.
* **Network Scanning:** Lists the active hosts and IP addresses. Network scanning is a procedure for identifying active hosts on a network, either to attack or assess the security of the network.
* **Vulnerability Scanning:** Shows the presence of known weaknesses. Vulnerability scanning is a method for checking whether a system is exploitable by identifying its vulnerabilities. A vulnerability scanner consists of a scanning engine and a catalog. The catalog includes a list of common files with known vulnerabilities and common exploits for a range of servers.

**Objective of Network Scanning**

* Discover the network’s live hosts, IP addresses, and open ports of the live hosts.
* Discover the OS and system architecture of the target. This is also known as fingerprinting.
* Discover the services running/listening on the target system.
* Identify specific applications or versions of a particular service.
* Identify vulnerabilities in any of the network systems.

**TCP/IP Communication**

TCP is connection oriented. It prioritizes connection establishment before data transfer between applications. This connection between protocols is possible through the three-way handshake. A TCP session initiates using a three-way handshake mechanism:

* To launch a TCP connection, the source sends a SYN packet to the destination.
* On receiving SYN packet, destination responds by sending a SYN/ACK packet back to source.
* The ACK packet confirms the arrival of the first SYN packet to the source
* Finally, source sends an ACK packet for the ACK/SYN packet transmitted by the destination.
* Triggers an "OPEN" connection, allowing communication between the source and destination.

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

**TCP Communication Flags**

The TCP header contains various flags that control the transmission of data across a TCP connection.

Six TCP control flags manage the connection between hosts and give instructions to the system. Four of these flags (SYN, ACK, FIN, and RST) govern the establishment, maintenance, and termination of a connection. The other two flags (PSH and URG) provide instructions to the system. The size of each flag is 1 bit. As there are six flags in the TCP Flags section, the size of this section is 6 bits.

* **Synchronize (SYN):** It notifies the transmission of a new sequence number. This flag generally represents the establishment of a connection (three-way handshake) between two hosts.
* **Acknowledgement (ACK):** It confirms the receipt of transmission and identifies next expected sequence number. When system successfully receives a packet, it sets value of its flag to “1,”
* **Push (PSH):** When it is set to “1,” it indicates that the sender has raised the push operation to the receiver; this implies that the remote system should inform the receiving application about the buffered data coming from the sender.
* **Urgent (URG):** It instructs the system to process the data contained in packets as soon as possible. When the system sets the flag to “1,” priority is given to processing urgent data first.
* **Finish (FIN):** It is set to “1” to announce that no more transmissions will be sent to the remote system and the connection established by the SYN flag is terminated.
* **Reset (RST):** When there is an error in the current connection, this flag is set to “1” and the connection is aborted in response to the error.

**Nmap Scanning command:** **Syntax: # nmap <options> <Target IP address>**

**Hping Scanning Commands:**

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

**🡺 Host Discovery & Host Discovery Techniques**

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

**ARP Ping Scan**

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

In Zenmap, -PR option is used to perform ARP ping scan. -sn is Nmap command to disable port scan.

**UDP Ping scan**

Nmap or Attackers send UDP packets to the target host, and a UDP response means that the target host is active. If the target host is offline or unreachable, various error messages such as host/network unreachable or TTL exceeded could be returned. The default port number used by Nmap for the UDP ping scan is 40,125. This highly uncommon port is used as the default for sending UDP packets to the target. This default port number can be configured using DEFAULT\_UDP\_PROBE\_PORT\_SPEC during compile time in Nmap.

In Zenmap, the -PU option is used to perform the UDP ping scan.

**ICMP ECHO Ping Scan**

Attackers use the ICMP ping scan to send ICMP packets to the destination system to gather all necessary information about it. This is because ICMP does not include port abstraction, and it is different from port scanning. However, it is useful to determine what hosts in a network are running by pinging them all. ICMP ECHO ping scan involves sending ICMP ECHO requests to a host. If the host is alive, it will return an ICMP ECHO reply. This scan is useful for locating active devices or determining if ICMP is passing through a firewall.

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

Nmap uses the -P option to ICMP scan the target. To increase the number of pings in parallel using the -L option. To tweak the ping timeout value - T option is used. In Zenmap, the -PE option is used to perform the ICMP ECHO ping scan.

**ICMP ECHO Ping Sweep**

A ping sweep (also known as an ICMP sweep) is a basic network scanning technique that is adopted to determine the range of IP addresses that map to live hosts (computers). Although a single ping will tell the user whether a specified host computer exists on the network, ICMP echo scanning pings all the machines in the target network to discover live machines. Attackers send ICMP probes to the broadcast or network address, which relays to all the host addresses in the subnet. The live systems will send the ICMP echo reply message to the source of the ICMP echo probe.When a system pings, it sends a single packet across the network to a specific IP address. This packet contains 64 bytes (56 data bytes and 8 bytes of protocol header information). The sender then waits or listens for a return packet from the target system. If the connections are good and the target computer is “alive,” a good return packet is expected. However, this will not be the case if there is a disruption in communication.

In Zenmap, the -PE option with a list of IP addresses is used to perform ICMP ECHO ping sweep.

**Ping Sweep Countermeasures**

* Configure the firewall to detect and prevent ping sweep attempts instantaneously
* Use IDS & IPS such as Snort to detect and prevent ping sweep attempts
* Carefully evaluate the type of ICMP traffic flowing through the enterprise networks
* Terminate the connection with any host that is performing more than 10 ICMP ECHO requests
* Use DMZ and allow only commands such as **ICMP ECHO\_REPLY, HOST UNREACHABLE**, and **TIME EXCEEDED** in DMZ Zone
* Limit the ICMP traffic with Access Control Lists (ACLs) to your ISP’s specific IP addresses

**ICMP Timestamp Ping Scan**

In ICMP timestamp ping an attackers query a timestamp message to acquire the information related to the current time from the target host machine. The target machine responds with a timestamp reply to each timestamp query that is received. However, the response from the destination host is conditional, and it may or may not respond with the time value depending on its configuration by the administrator at the target’s end. This ICMP timestamp pinging is generally used for time synchronization. Such a ping method is effective in identifying whether the destination host machine is active. This technique is usefull when traditional methods doesn’t work.

In Zenmap, the -PP option is used to perform an ICMP timestamp ping scan.

**ICMP Address Mask Ping Scan**

In ICMP address mask ping an attackers send an ICMP address mask query to the target host to acquire information related to the subnet mask. However, the address mask response from the destination host is conditional, and it may or may not respond with the appropriate subnet value depending on its configuration by the administrator at the target’s end. Such a ping method is effective in identifying whether the destination host machine is active.

In Zenmap, the -PM option is used to perform an ICMP address mask ping scan.

**TCP SYN Ping Scan**

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

In Zenmap, the -PS option is used to perform a TCP SYN ping scan.

**TCP ACK Ping Scan**

TCP ACK ping uses the default port 80. The attackers send an empty TCP ACK packet to the target host directly. Since there is no prior connection between the attacker and the target host, after receiving the ACK packet, the target host responds with an RST flag to terminate the request. The reception of this RST packet at the attacker’s end indicates that the host inactive.

In Zenmap, the -PA option is used to perform a TCP ACK ping scan.

**IP Protocol Ping Scan**

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

Multiple IP packets for ICMP (protocol 1), IGMP (protocol 2), and IP-in-IP (protocol 4) are sent by default when no protocols are specified. For configuring the default protocols, change DEFAULT\_PROTO\_PROBE\_PORT\_SPEC in nmap.h during compile time. For specific protocols such as ICMP, IGMP, TCP (protocol 6), and UDP (protocol 17), the packets are to be sent with proper protocol headers, and for the remaining protocols, only the IP header data is to be sent with the packets. Any response from any probe indicates that a host is online.

In Zenmap, the -PO option is used to perform an IP protocol ping scan.

**🡺 Port and Service Discovery**

Once attackers detect the live systems in the target network, they try to find open ports and services in the discovered live systems.

**Port Scanning Techniques:**

**TCP Connect/Full Open Scan**

In TCP Connect scanning, the OS’s TCP connect() system call tries to open a connection to every port of interest on the target machine. If the port is listening, the connect() call will result in a successful connection with the host on that particular port; otherwise, it will return an error message stating that the port is not reachable. TCP Connect scan completes a three-way handshake with the target machine. Making a separate connect() call for every targeted port in a linear manner would take a long time over a slow connection. The attacker can accelerate the scan using many sockets in parallel. Using non-blocking, I/O allows the attacker to set a short time-out period and watch all the sockets simultaneously. It is easily detectable and filterable.

In Zenmap, the -sT option is used to perform TCP Connect/full open scan

**Stealth Scan (Half-open Scan)**

It involves resetting the TCP connection between the client and the server abruptly before completion of the three-way handshake signals, hence making the connection half-open. A stealth scan sends a single frame to a TCP port without any TCP handshaking or additional packet transfers. This type of scan sends a single frame with the expectation of a single response. The stealth scan is also called a “SYN scan,” because it only sends the SYN packet.

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

In Zenmap, the -sS option is used to perform a stealth scan/TCP half-open scan.

**Inverse TCP Flag Scan**

Attackers send TCP probe packets with a TCP flag (FIN, URG, PSH) set or with no flags. When the port is open, the attacker does not get any response from the host, whereas when the port is closed, he or she receives the RST from the target host. Security mechanisms such as firewalls and IDS detect the SYN packets sent to the sensitive ports of the targeted hosts. Programs such as Synlogger and Courtney are available to log half-open SYN flag scan attempts.

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

Common flag configurations used for a probe packet include:

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

**Xmas Scan**

Xmas scan is a type of inverse TCP scanning technique. This technique can be used to scan large networks and find which host is up and what services it is offering. This technique describes all TCP flag sets. When all flags are set, some systems hang; hence, the flags are often set in the nonsense pattern URG-PSH-FIN. Attackers use the TCP Xmas scan to determine if ports are closed on the target machine via the RST packet. This scan only works when systems are compliant with RFC 793-based TCP/IP implementation. It will not work against any current version of Microsoft Windows.

BSD Networking Code: This method relies on the BSD networking code. Thus, you can use this only for UNIX hosts; it does not support Windows NT.

In Zenmap, -sX perform the Xmas scan, -sF perform the FIN scan, -sN perform the NULL scan.

**TCP Maimon scan**

This scan is very similar to NULL, FIN, and Xmas scan, but the probe used here is FIN/ACK. In most cases, to determine if the port is open or closed, the RST packet should be generated as a response to a probe request. However, in many BSD systems, the port is open if the packet gets dropped in response to a probe. Nmap interprets a port as open|filtered when there is no response from the Maimon scan probe even after many retransmissions. The port is closed if the probe gets a response as an RST packet. The port is filtered when the ICMP unreachable error (type 3, code 1, 2, 3, 9, 10, or 13) is returned from the target host.

In Zenmap, the -sM option is used to perform the TCP Maimon scan.

**ACK Flag Probe Scan**

Attackers send TCP probe packets with the ACK flag set to a remote device and then analyze the header information (TTL and WINDOW field) of the received RST packets to find out if the port is open or closed. The ACK flag probe scan exploits the vulnerabilities within the BSD-derived TCP/IP stack.

Categories of ACK flag probe scanning include:

* **TTL-based ACK Flag Probe scanning:** In this scanning, you will first need to send ACK probe packets to different TCP ports and then analyze the TTL field value of the RST packets received. If the TTL value of the RST packet on a particular port is less than the boundary value of 64, then that port is open.

In Zenmap, the syntax nmap –ttl [time] [target] is used to perform TTL-based scan.

* **Window-based ACK Flag Probe scanning**: In this scanning, you will first need to send ACK probe packets to different TCP ports and then analyze the window field value of the received RST packets. The user can use this scanning when all the ports return the same TTL value. If the window value of RST packet on a particular port is non-zero, then that port is open.

In Zenmap, the -sW option is used to perform a window scan.

* **ACK Flag Probe Scanning using Nmap:** In Zenmap, -sA option perform an ACK flag probe scan.
* **Checking the Filtering Systems of Target Networks:** The attacker sends an ACK probe packet to check the filtering mechanism (firewalls) of packets employed by the target network. Sending an ACK probe packet with a random sequence number and getting no response from the target means that the port is filtered (stateful firewall is present); an RST response from the target means that the port is not filtered (no firewall is present).

**IDLE/IPID Header Scan**

The IDLE/IPID Header scan is a TCP port scan method that you can use to send a spoofed source address to a computer to find out what services are available. It offers complete blind scanning of a remote host. Most network servers listen on TCP ports, such as web servers on port 80 and mail servers on port 25. A port is considered “open” if an application is listening on the port. One way to determine whether a port is open is to send a "SYN" (session establishment) packet to the port. The target machine will send back a "SYN|ACK" (session request acknowledgement) packet if the port is open or an "RST" (Reset) packet if the port is closed.

In Zenmap, the -sI option is used to perform the IDLE scan.

**IDLE Scan**

Every IP packet on the Internet has a fragment Internet protocol identification (IPID) number that uniquely identifies fragments of an original IP datagram.

* **Step-1: Choose a “Zombie” and Probe its Current IP Identification (IPID) Number**

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

* **Step 2: The attacker sends a SYN packet to the target machine on port 80, spoofing the IP address of the zombie.**
* **Idle Scan: Step 2.1 (Open Port)**If the port is open, the target will send the SYN+ACK packet to the zombie (as the IP address was spoofed) to proceed with the three-way handshake. Since every IP packet has a "fragment identification" number, which increases by one for every packet transmission, the zombie will now use its next available IPID
* **Idle Scan: Step 2.2 (Closed Port) Assume that the port on the target is closed.** Subsequently, on receiving the SYN packet from the attacker (you), the target will respond with an RST, and zombie will remain idle without taking any further action.
* **Step 3 Now, follow step 1 again to probe the IPID number.**

Thus, using an idle scan, an attacker can find out the open ports and services on the target machine by spoofing his/her IP address with a zombie’s IP address.

**UDP Scanning**

* **UDP Raw ICMP Port Unreachable Scanning**

There is no three-way handshake for the UDP scan. In UDP scanning you can send a packet but you cannot determine whether the host is alive, dead, or filtered. However, you can use one ICMP that checks for open or closed ports. If you send a UDP packet to a port without an application bound to it, the IP stack will return an ICMP port unreachable packet. If any port returns an ICMP error, it will be closed, leaving the ports that did not answer if they are open or filtered through the firewall. this scanning is slow because it limits the ICMP error message rate as a form of compensation to machines that apply RFC 1812 section 4.3.2.8.

In Zenmap, the -sU option is used to perform a UDP scan.

* **UDP RECVFROM () and WRITE () Scanning**

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

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

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

**SCTP INIT Scanning**

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

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

Attackers send INIT chunk to the target host. If the port is listening or open, it sends an acknowledgement as an INIT+ACK chunk. If the target is inactive and it is not listening, then it sends an acknowledgement as an ABORT chunk. After several retransmissions, if there is no response, then port is indicated as a filtered port. COOKIE ECHO chunk is blocked by non-stateful firewall rule sets.

In Zenmap, the -sY option is used to perform the SCTP INIT scan.

**SCTP COOKIE ECHO Scanning**

In this type of scan, attackers send the COOKIE ECHO chunk to the target, and if the target port is open, it will silently drop the packets onto the port and you will not receive any response from the target. If the target sends back the ABORT chunk response, then the port is considered as a closed port. The COOKIE ECHO chunk is not blocked by non-stateful firewall rule sets.

In Zenmap, the -sZ option is used to perform the SCTP COOKIE ECHO scan.

**SSDP Scanning**

Simple Service Discovery Protocol (SSDP) is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses. The SSDP service controls communication for the Universal Plug and Play (UPnP) feature. It generally works when the machine is not firewalled, sometimes work through a firewall also. The SSDP service will respond to a query sent over IPv4 or IPv6 broadcast addresses. This response includes information about the UPnP feature associated with it. The attacker may use the UPnP SSDP M-SEARCH information discovery tool to check whether the machine is vulnerable to UPnP exploits.

**List Scanning**

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

In Zenmap, the -sL option is used to perform a list scan.

**IPv6 Scanning**

IPv6 increases the size of the IP address space from 32 bits to 128 bits to support higher levels of the addressing hierarchy. A number of scanning tools do not support ping sweeps on IPv6 networks. Attackers need to harvest IPv6 addresses from network traffic, recorded logs, or "Received from" and other header lines in archived email or Usenet news messages to identify IPv6 addresses for subsequent port scanning. However, scanning an IPv6 network provides a large number of hosts in a subnet; if an attacker can compromise one subnet host, he/she can probe the "all hosts" link local multicast address if the hosts numbers are sequential or use any regular scheme. An attacker needs to analyze 264 addresses to verify if a particular open service is running on a host in that subnet. At a conservative rate of one probe per second, such a scan would take about 5 billion years to complete. Attackers can use Nmap to perform IPv6 scanning.

In Zenmap, the -6 option is used to perform the IPv6 scan.

**Service Version Discovery**

Every port is assigned a specific service, and every service has its own version. Service version detection helps attackers to obtain information about the running services and their versions on a target system. By obtaining accurate service version numbers, an attacker can determine which exploits the target system is vulnerable to. The version detection technique is nothing but examination of the TCP and UDP ports. The probes from the Nmap service-probes database are used for querying various services and matching expressions for recognizing and parsing responses.

In Zenmap, the -sV option is used to detect service versions.

**Nmap Scan Time Reduction Techniques**

In Nmap, performance and accuracy take high priority, and this only be achieved only by reducing the long scan time.

* **Omit Non-critical Tests:** While performing the Nmap scan, the time complexity can be reduced by the following methods:
* port scan (-sn) can be skipped if one has to check whether the hosts are online or not.
* Advanced scan types (-sC, -sV, -O, --traceroute, and -A) can be avoided.
* The DNS resolution should be turned on only when it is necessary.
* **Optimize Timing Parameters:** To control the scan activity, Nmap provides the -T option for scanning ranging from high-level to low-level timing aggressiveness.
* **Separate and Optimize UDP Scans:** As many vulnerable services use the UDP protocol, scanning the UDP protocol is vital, and it should be scanned separately, as TCP scans have different performance requirements and timing characteristics.
* **Upgrade Nmap**
* **Execute Concurrent Nmap Instances:** Running Nmap against the whole network usually makes the system slower and less efficient. Nmap supports parallelization and it can also be customized according to specific needs.
* **Scan from a Favourable Network Location:** It is always advisable to run Nmap from the host’s local network to the target while in the internal network, as it offers defense-in-depth security.
* **Increase Available Bandwidth and CPU Time:** By increasing the available bandwidth or CPU power, the Nmap scan time can be reduced. Nmap bandwidth usage can be tested by running it in the verbose mode -v.

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

**OS Discovery/Banner Grabbing**

Banner grabbing, or "OS fingerprinting," is a method used to determine the OS that is running on a remote target system. It is an important scanning method, as the attacker will have a higher probability of success if the OS of the target system is known. There are two methods for banner grabbing: spotting the banner while trying to connect to a service, such as an FTP site, and downloading the binary file/bin/ls to check the system architecture.

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

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

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

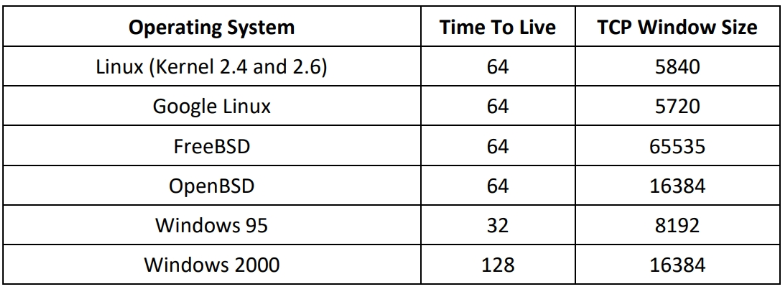
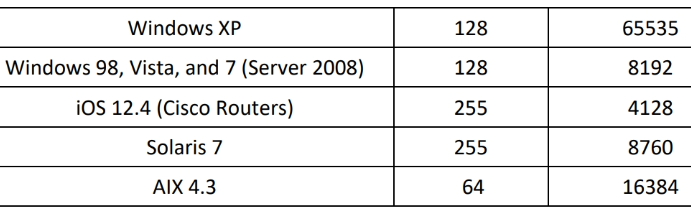
Two types of banner grabbing techniques

* **Active banner grabbing: It** applies the principle that an OS’s IP stack has a unique way of responding to specially crafted TCP packets. In active banner grabbing, the attacker sends a variety of malformed packets to the remote host, and the responses are compared with a database. the scanning utility Nmap uses a series of nine tests to determine an OS fingerprint or banner grabbing.
* **Test 1:** TCP packet with SYN and ECN-Echo flags enabled is sent to an open TCP port.
* **Test 2:** TCP packet with no flags enabled is sent to open TCP port. This is NULL packet.
* **Test 3:** TCP packet with URG, PSH, SYN, & FIN flags enabled is sent to open TCP port.
* **Test 4:** TCP packet with the ACK flag enabled is sent to an open TCP port.
* **Test 5:** TCP packet with the SYN flag enabled is sent to a closed TCP port.
* **Test 6:** TCP packet with the ACK flag enabled is sent to a closed TCP port.
* **Test 7:** TCP packet with the URG, PSH, & FIN flags enabled is sent to closed TCP port.
* **Test 8 PU (Port Unreachable):** A UDP packet is sent to a closed UDP port. The objective is to extract an “ICMP port unreachable” message from the target machine.
* **Test 9 TSeq (TCP Sequence ability test):** This test tries to determine the sequence generation patterns of the TCP initial sequence numbers (TCP ISN sampling), the IP identification numbers (IPID sampling), and the TCP timestamp numbers.
* **Passive Banner Grabbing:** It also depends on the differential implementation of the stack and the various ways in which an OS responds to packets. However, instead of relying on scanning the target host, passive fingerprinting captures packets from the target host via sniffing to study telltale signs that can reveal an OS. Passive banner grabbing includes:
* **Banner grabbing from error messages:** Error messages provide information, such as type of server, type of OS, and SSL tools used by the target remote system.
* **Sniffing the network traffic:** Capturing and analyzing packets from the target enables an attacker to determine the OS used by the remote system.
* **Banner grabbing from page extensions:** Looking for an extension in URL may help in determining application version. Example, .aspx => IIS server & Windows platform.

The four areas that typically determine the OS:

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

**How to Identify Target System**

In a network, various standards are implemented to allow different OSs to communicate with each other. These standards govern the functioning of various protocols such as IP, TCP, UDP, etc. By analyzing certain parameters/fields in these protocols, one can reveal the details of the OS. Parameters such as Time to Live (TTL) and TCP window size in the IP header of the first packet in a TCP session help identify the OS running on the target machine. The TTL field determines the maximum time that a packet can remain in a network, and the TCP window size determines the length of the packet reported. T **OS Discovery using Wireshark**

To identify the target OS, sniff/capture the response generated from the target machine to the request-originated machine using packet-sniffing tools such as Wireshark, etc., and observe the TTL and TCP window size fields in the first captured TCP packet.

**OS Discovery using Nmap**

In Zenmap, the -O option is used to perform OS discovery.

**OS Discovery using Unicornscan**

In Unicornscan, the OS of the target machine can be identified by observing the TTL values in the acquired scan result. To perform Unicornscan, the syntax #unicornscan <target IP address> is used.

**OS Discovery using Nmap Script Engine**

These scripts can be executed parallelly with the same efficiency and speed as Nmap. Attackers can also use various scripts in the Nmap Script Engine for performing OS discovery on the target machine. For example, in Nmap, smb-os-discovery is an inbuilt script used for collecting OS information on the target machine through the SMB protocol. In Zenmap, NSE can be generally activated using the -sC option. If the custom scripts are to be specified, then attackers can use the --script option.

**OS Discovery using IPv6 Fingerprinting**

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

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

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

**🡺 Scanning Beyond IDS and Firewall**

**IDS/Firewall Evasion Techniques:**

**Packet Fragmentation**

Packet fragmentation refers to the splitting of a probe packet into several smaller packets (fragments) while sending it to a network. When these packets reach a host, the IDS and firewalls behind the host generally queue all of them and process them one by one. However, since this method of processing involves greater CPU and network resource consumption, the configuration of most IDS causes them to skip fragmented packets during port scans. Once these fragments reach the destined host, they are reassembled to form a single packet.

**SYN/FIN Scanning Using IP Fragments**

This process of scanning was developed to avoid false positives generated by other scans because of a packet filtering device on the target system. The TCP header splits into several packets to evade the packet filter. For any transmission, every TCP header must have the source and destination port for the initial packet (8-octet, 64-bit). The initialized flags in the next packet allow the remote host to reassemble the packets upon receipt via an Internet protocol module that detects the fragmented data packets using field-equivalent values of the source, destination, protocol, and identification.

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

**Source Routing**

An IP datagram contains various fields, including the IP options field, which stores source routing information and includes a list of IP addresses through which the packet travels to its destination. As the packet travels through the nodes in the network, each router examines the destination IP address and chooses the next hop to direct the packet to the destination. When attackers send malformed packets to a target, these packets hop through various routers and gateways to reach the destination. In some cases, the routers in the path might include configured firewalls and IDS that block such packets. To avoid them, attackers enforce a loose or strict source routing mechanism, in which they manipulate the IP address path in the IP options field.

**Source Port Manipulation**

This technique the actual port numbers are manipulated with common port numbers for evading certain IDS and firewall rules. The main security misconfigurations occur because of blindly trusting the source port number. The administrator mostly configures the firewall by allowing the incoming traffic from well-known ports such as HTTP, DNS, FTP, etc.

In Zenmap, the -g or --source-port option is used to perform source port manipulation.

**IP Address Decoy**

The IP address decoy technique refers to generating or manually specifying IP addresses of the decoys to evade IDS/firewalls. It appears to the target that the decoys as well as the host(s) are scanning the network. This technique makes it difficult for the IDS/firewall to determine which IP address is actually scanning the network and which IP addresses are decoys. The Nmap scanning tool comes with a built-in scan function called a decoy scan, which cloaks a scan with decoys. This technique generates multiple IP addresses to perform a scan, thus making it difficult for the target security mechanisms such as IDS, firewalls, to identify original source from the registered logs. The target IDS might report scanning from 5–0 IP addresses. IP address decoy is a useful technique for hiding your IP address.

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

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

* nmap -D decoy1,decoy2,decoy3,...,ME,... [target]Using this command, you can manually specify the IP addresses of the decoys to scan the victim’s network. Here, you have to separate each decoy IP with a comma (,) and you can optionally use the ME command to position your real IP in the decoy list. If you place ME in the 4th position of the command, your real IP will be positioned at the 4th position accordingly. If you don’t mention ME, Nmap randomly places your IP address.

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

**IP Address Spoofing**

Most firewalls filter packets based on the source IP address. These firewalls examine the source IP address and determine whether the packet is coming from a legitimate source or an illegitimate source. IP address spoofing is a hijacking technique in which an attacker obtains a computer’s IP address, alters the packet headers, and sends request packets to a target machine, pretending to be a legitimate host. The packets appear to be sent from a legitimate machine but are actually sent from the attacker’s machine, while his/her machine's IP address is concealed. When the victim replies to the address, it goes back to the spoofed address and not to the attacker’s real address.

IP spoofing using Hping3: **Hping3 www.certifiedhacker.com -a 7.7.7.7** command helps to send arbitrary TCP/IP packets to network hosts.

**IP Spoofing Detection Techniques**

* **Direct TTL Probes:** In this technique, you initially send a packet (ping request) to the legitimate host and wait for a reply. Check whether the TTL value in the reply matches with that of the packet you are checking. Both will have the same TTL if they are using the same protocol. The initial TTL values vary according to the protocol used. For TCP/UDP: 64 & 128 and ICMP: 128 & 255. If the reply is from a different protocol, then you should check the actual hop count to detect the spoofed packets. Deduct the TTL value in the reply from the initial TTL value to determine the hop count. The packet is a spoofed packet if the reply TTL does not match the TTL of the packet. This technique is successful when the attacker is in a different subnet from that of the victim.
* **IP Identification Number:** Users can identify spoofed packets by monitoring the IP identification (IPID) number in the IP packet headers. The IPID increases incrementally each time a system sends a packet. Every IP packet on the network has a "fragment identification" number, which is increased by one for every packet transmission. To identify whether a packet is spoofed, send a probe packet to the source IP address of the packet and observe the IPID number in the reply. The IPID value in the response packet must be close to but slightly greater than the IPID value of the probe packet otherwise spoofed. This method is effective even when both the attacker and the target are on the same subnet.
* **TCP Flow Control Method:** The TCP can optimize the flow control on both the sender’s and the receiver’s end with its algorithm. The algorithm accomplishes flow control using the sliding window principle. The user can control the flow of IP packets by the window size field in the TCP header. This field represents the maximum amount of data that the recipient can receive and the maximum amount of data that the sender can transmit without acknowledgement. In general flow control, the sender should stop sending data once the initial window size is exhausted. The attacker, who is unaware of the ACK packet containing window size information, might continue to send data to the victim. If the victim receives data packets beyond the window size, they are spoofed packets.

Most spoofing attacks occur during the handshake. In a TCP handshake, the host sending the initial SYN packet waits for SYN-ACK before sending the ACK packet. To check whether you are getting the SYN request from a genuine client or a spoofed one, set SYN-ACK to zero. If the sender sends an ACK with any data, it means that the sender is a spoofed one.

**Creating Custom Packets**

The attacker creates and sends custom packets to scan the intended target beyond the IDS/firewalls.

* **Creating Custom Packets by using Packet Crafting Tools:** Attackers create custom TCP packets to scan the target by bypassing the firewalls. Attackers use various packet crafting tools such as Colasoft packet builder, NetScanTools Pro, to scan the target that is beyond the firewall. Packet crafting tools craft and send packet streams (custom packets) using different protocols at different transfer rates.

There are three views in the Packet Builder

* Packet List displays all constructed packets. When you select one or more packets in Packet List, first highlighted packet is displayed in both Decode & Hex Editor for editing.
* In Hex Editor, the data of the packet are represented as hexadecimal values and ASCII characters; nonprintable characters are represented by a dot (".") in the ASCII section.
* Decode Editor allows the attacker to edit packets without remembering the value length, byte order, and offsets.
* **Creating Custom Packets by Appending Custom Binary Data:** Attackers send binary data (0’s and 1’s) as payloads in the packets sent to the target machine present behind the firewall. The option used by Nmap for appending custom binary data to the sent packets is --data <hex string>. Any <hex string> is specified in the formats 0xAABBCCDDEEFF<...>, AABBCCDDEEFF<...>, or \xAA\xBB\xCC\xDD\xEE\xFF<...>. To perform a byte-order conversion, the specified information should be based on the receiver’s expectations.
* **Creating Custom Packets by Appending Custom String:** Attackers send regular strings as payloads in the packets sent to the target machine for scanning beyond the firewall. The option used by Nmap for appending a custom string to the sent packets is --data-string <string>. The <string> can contain any string and a few characters depend on the system’s location; however, it is not guaranteed whether the same information is retrieved. The string is enclosed with double quotes (“”) and special characters from the shell are not used.
* **Creating Custom Packets by Appending Random Data:** Attackers append a number of random data bytes to most packets sent without using any protocol-specific payloads. The option used by Nmap for appending random data to the sent packets is --data-length <number>. For protocol-specific and no random payloads, --data-length 0 is used. The (-O) OS detection packets are not usually affected, as probe consistency is needed for it to be accurate. By default, a few UDP ports and IP protocols get a custom payload.

**Randomizing Host Order**

The attacker scans the number of hosts in the target network in a random order to scan the intended target that is lying beyond the firewall. The option used by Nmap to scan with a random host order is --randomize-hosts.This technique instructs Nmap to shuffle each group of 16384 hosts before scanning with slow timing options, thus making the scan less notable to network monitoring systems and firewalls. If larger group sizes are randomized, the PING\_GROUP\_SZ should be increased in nmap.h and it should be compiled again. Another method can be followed by generating the target IP list with the list scan command -sL -n -oN <filename> and then randomizing it with a Perl script and providing the whole list to Nmap using the -iL command.

**Sending Bad Checksums**

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

**IP Spoofing Countermeasures**

* **Avoid trust relationships**: Do not rely on IP-based authentication. Attackers may spoof themselves as trusted hosts and send malicious packets to you.
* **Use firewalls and filtering mechanisms**
* **Use random initial sequence numbers:** Most devices choose their ISN based on timed counters. This makes the ISNs predictable.
* **Ingress filtering:** It prevents spoofed traffic from entering the Internet. It is applied to routers because it enhances the functionality of the routers and blocks spoofed traffic. Configuring and using ACLs that drop packets with the source address outside the defined range is one method of implementing ingress filtering.
* **Egress filtering:** It refers to a practice that aims to prevent IP spoofing by blocking outgoing packets with a source address that is not inside.
* **Use encryption:** If you want to attain maximum network security, then use strong encryption for all the traffic placed onto the transmission media without considering its type and location.
* **SYN flooding countermeasures:** Countermeasures against SYN flooding attacks can also help you to avoid IP spoofing attacks.

**Proxy Servers**

A proxy server is an application that can serve as an intermediary for connecting with other computers. Initially, when you use a proxy to request a particular web page on an actual server, the proxy server receives it. The proxy server then sends your request to the actual server on your behalf. It mediates between you and the actual server to transmit and respond to the request. In this process, the proxy receives the communication between the client and the destination application. To take advantage of a proxy server, an attacker must configure client programs so that they can send their requests to the proxy server instead of the final destination. he server logs will record the proxy's source address rather than the attacker’s source address.

A proxy server is used:

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

**Proxy Chaining**

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

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

**Anonymizers**

An anonymizer is an intermediate server placed between you as the end user and the website to access the website on your behalf and make your web surfing activities untraceable. Anonymizers allows you to bypass Internet censors. An anonymizer eliminates all the identifying information (IP address) from your system while you are surfing the Internet, thereby ensuring privacy. Most anonymizers can anonymize the web (HTTP:), file transfer protocol (FTP:), and gopher (gopher:) Internet services. An anonymizer is a service through which one can hide one’s identity when using certain Internet services. It encrypts the data from your computer to the Internet service provider.

Why Use an Anonymizer?

* **Ensuring privacy:** Protect your identity by making your web navigation activities untraceable.
* **Accessing government-restricted content**
* **Protection against online attacks:** An anonymizer can protect you from all instances of online pharming attacks by routing all customer Internet traffic via its protected DNS server.
* **Bypassing IDS and firewall rules:** Firewalls are typically bypassed by employees or students accessing websites that they are not supposed to access.

Types of Anonymizers

* **Networked Anonymizers**: A networked anonymizer first transfers your information through a network of Internet-connected computers before passing it on to the website. Because the information passes through several Internet computers, it becomes cumbersome for anyone trying to track your information to establish the connection between you and the anonymizer.
* **Single-Point Anonymizers**: Single-point anonymizers first transfer your information through a website before sending it to the target website and then pass back the information gathered from the target website to you via the website to protect your identity.

**🡺 Drawing Network Diagrams**

Drawing a network diagram helps an attacker to identify the topology or architecture of a target network. The network diagram also helps to trace the path to the target host in the network and enables the attacker to understand the positions of firewalls, IDS, routers, and other access control devices. The network diagram also helps network administrators to manage their networks.