SCANNING NETWORKS

PRO GUIDE

HACKLIDO

Any information presented in this book is solely intended to provide information, guidance, and advice to its readers. The author of this book makes no representations or warranties of any kind, either expressed or implied, regarding the accuracy, completeness, or usefulness of the information contained within. The author shall not be held liable for any losses or damages caused directly or indirectly by the information contained in this book.

The author(s) does not advocate or condone the illegal or unethical actions described in this book. All information contained in this book should be used for educational and research purposes only. The user shall be solely responsible for any and all activities undertaken by them. By purchasing this book, the reader assumes all risks and liabilities and agrees to use the information contained within with full responsibility.

Author:

Arjun S. - Founder at Hacklido

Unit 1: Understanding Networks

OSI Model

It's a 7-Layered representation of how a network should work.Remember easily - **P**lease **D**o **N**ot **T**hrow **S**ausage **P**izza **A**way.

Physical Layer (L1)

- Make sure 0s & 1s get between different hosts.
- Deals with specifications like how thick of copper wire do we use for cabling.
- If wireless, what frequency of radio waves are going to be used.
- Strips the preamble at the start of the frame - which contains a bunch of 1s & 0s that alternates and denotes the incoming frame.
- Strips the frame check sequence at the end - this ensures that the data that was sent is the same as the data that was received.
- Now you have the Frame.

Data Link Layer (L2)

- Allows individual systems to be addressed in such a way that ethernet frames get to the right spot.
- Inspects incoming frames to see if they are addressed for me based on my MAC address.
- MAC addresses are (OEM - Original Equipment Manufacturer) Numbers, unique 48-Bit (8x6) 38-00-25-94-F5-93 each section is of 8 Bytes meaning each digit/character is of 4 Bytes. So it's also a (4x12) composition value that's burned into every Network Card wired or wireless. They have two halves:
    - The first 24 (8x3) bits form the Organizationally Unique Identifier (OUI)
    - The last 24 (8x3) bits form a serial number (formally called an extension identifier).
- Checks the Destination MAC Address field and validates if it's for it.
- It strips off the Source & Destinatination MAC Address fields, but will not discard them, because if there is a return packet, it needs to be sent to the source MAC Address.
- Now you have the IP Packet.

Network Layer (L3)

- MACs are good to go with if the transfer is within the LAN network. But what if we have an internet where the computers are distributed in large areas - MAC addressing becomes insufficient.

- So we use a logical IP Address. So look up the "Destination & Source Address" strip it away and keep it, same as L2.
- Send to L4

Transport Layer (L4)

- Assemble and disassemble different pieces of data as they come in.
- Each individual ethernet frame could only hold 1500 bytes of data, maximum.
- So at this layer, data that is larger than that is broken into chunks and sent, as well as assembled all together at the receiver end.
- Sequencing all the pieces and sending them to the L5.

Session Layer (L5)

- Part of the host that establishes connection, say, to a remote host, a web server, email client, etc... You can use a browser or any similar kinda software.
- Once the connection is made, then the data can be moved between them.
- Look for, Strip off and keep aside the "Source & Destination ports", how IP address identifies to which system the packet is for, same way the Port identifies to which application in the host the data is addressed to.
- Imagine having 5 Chrome tabs open, each listens to each port and receives the data. Here chrome is the application that probably resides in the L7, but is the application capable of reading this data directly?

Presentation Layer (L6)

- Gets the data in a format that a destined application can read and respond to.
- So if I've got a webpage coming in from a web server, my web browser may not know how to read it?
- Well, there was a time where that could be true. The thing is today, all of our applications are so good they can read just about anything.
- So it used to be that I would have to convert this data. Let's say if Microsoft Word was going on the network to get a Word document, but I had a WordPerfect document or some other competitor, I would have had to go through the conversion so that it could read it.
- But today, everybody reads everybody. If you were to ask me, if there was one of the seven layers we could get rid of, it's the presentation layer. He's just not that important anymore.

Application Layer (L7)

- When we think of an application layer, it's not the actual application that we need to care for, instead the "smarts" in them.
- This can actually read and deal with the data, because that's the whole reason for doing the networking right.
- Think of "smarts" like APIs, like the MS Word can access Network File if you use the Open>File. In the back, it uses specific Network APIs to achieve this.

Sending Frame

- When the NIC constructs the frame to be sent, it adds " Source & Destination" MAC Addresses along with a CRC - Cyclic Redundancy Check field for error detection.
- Traditionally, while using the hub, it just repeats the constructed data to all the connections (NICs) it has and the NICs will decide by checking the MAC address like if it's for it, it accepts - else rejects.

Broadcast vs Unicast

- A unicast transmission is addressed to a single device on a network.
- A broadcast transmission is sent to every device on a broadcast domain (Group of computers that can hear each other's broadcast).
- A broadcast address looks like all Fs - FF-FF-FF-FF-FF-FF when a NIC sees this kinda address, it just passes it over to the next layer to process it.
- This is done in an analogy like, Hey, if anyone up here has this "name" kindly respond back. Same way, this broadcast packet will be responded back with the MAC addresses of each NICs.

IP Addressing

- The Destination & Source IP is encapsulated along with the MAC to make the packet traverse across networks. The flow would look something similar to this.
- The lookup is done on the current router and if the destination is not a part of that network, the packet is sent to the default gateway
- This is the connection to your router itself, so your computer puts the destination MAC address as the Router's one and sends it to the router to lookup on to other networks connected to it.
- Router decapsulates the packet, strips the Source and Destination MAC, performs lookup using the routing table and sends it to the next router.

- Remember that packets cannot travel by themselves, they are always encapsulated within frames.
- Also it's worth noting that the frame information keeps changing as the hop between routers and networks happens but the IP information remains the same.

Packets and Ports

- Now there is 2 major problems, as the size of data we can receive is only 1500 Bytes, we might receive it in pieces
- At the same time I may have many applications or even one chrome browser with multiple tabs, how will the data be delivered exactly to that app?
- TCP solves the first problem with Sequencing Number and Acknowledgements.
- and the second problem is dealt by the ports - makes sure that the data gets delivered to the right application.
- **Well-known ports (0–1023): Commons ports**
    - These ports are related to the common protocols that are at the core of the TCP/IP model, DNS, SMTP, etc.
    - Can never be used as a return port number.
- **Registered ports (1024–49151): Vendor specific ports.**
    - These ports are often associated with proprietary applications from vendors and developers.

    - While they are officially approved by the Internet Assigned Numbers Authority (IANA), in practice many vendors simply implement a port of their choosing.
    - Examples include Remote Authentication Dial-In User Service (RADIUS) authentication (1812), Microsoft SQL Server (1433/1434) and the Docker REST API (2375/2376).
- **Dynamic or private ports (49152–65535): Service specific random ports.**
    - Whenever a service is requested that is associated with well-known or registered ports,
    - those services will respond with a dynamic port that is used for that session and then released.

Unit 2: Scanning Networks, countermeasures & tools.

ARP Ping Scan

- It's `-PR` in Zenmap. In Nmap, `-sn` to disable the port scan. Since Nmap uses ARP ping scan by default
- To disable it and perform other desired ping scans, we can use `--disable-arp-ping`
- **Advantages:**
    - More efficient and accurate.
    - Automatically handles requests, retransmission, and timeout.
    - Useful when scanning large address space.
    - Can display response time or latency.

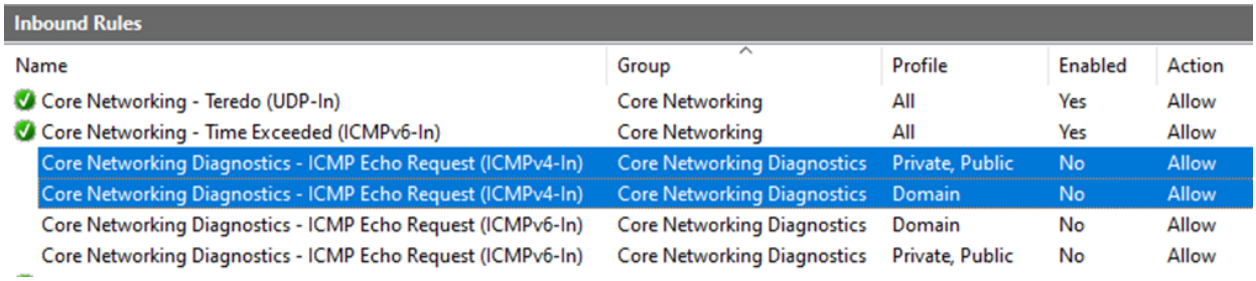
UDP Ping Scan

- Default port number used by Nmap for 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.
- **Advantage:** Detects systems behind firewalls with strict TCP filtering.

ICMP Ping Scan

ICMP ECHO Ping Scan

- ICMP does not include port abstraction (No specific port is allotted to it because it's not a service like FTP, SSH or etc...), and it is different from port scanning.
- This involves sending ICMP ECHO requests to a host. If the host is alive, it returns an ICMP ECHO Reply.
- UNIX/Linux and BSD-based machines use ICMP echo scanning; the TCP/IP stack implementations in these OSs respond to the ICMP echo requests to the broadcast addresses.



Inbound Rules

Name	Group	Profile	Enabled	Action
Core Networking - Teredo (UDP-In)	Core Networking	All	Yes	Allow
Core Networking - Time Exceeded (ICMPv6-In)	Core Networking	All	Yes	Allow
Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)	Core Networking Diagnostics	Private, Public	No	Allow
Core Networking Diagnostics - ICMP Echo Request (ICMPv4-In)	Core Networking Diagnostics	Domain	No	Allow
Core Networking Diagnostics - ICMP Echo Request (ICMPv6-In)	Core Networking Diagnostics	Domain	No	Allow
Core Networking Diagnostics - ICMP Echo Request (ICMPv6-In)	Core Networking Diagnostics	Private, Public	No	Allow

Figure 1: Windows Firewall ICMP Echo Reply Rule

- Whereas this 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 used `-P` to ICMP scan the target. We can increase the parallel pings with `-L` and timeout using `-T`.
- `nmap -PE 208.109.192.1`

ICMP Echo Ping Sweep

- Also Known As: ICMP Sweep adopted to determine the range of IPs.
- Pings calculates round-trip time and resolves hostnames.
- `nmap -sP 208.109.192.1-255`

ICMP Timestamp Ping

- It's generally used for time sync and response from the destination is purely conditional based on the admin's design.
- Also, when the admin blocks the traditional ICMP ECHO ping requests, we use this.
- `nmap -PP 192.168.1.104 --disable-arp-ping`

ICMP Address Mask Ping

- Sent to the target host to acquire information related to the subnet mask.
- The response is again conditional from the destination host and may not respond with appropriate subnet value either depending on the admin's configuration.
- Good alternative for the Timestamp Ping with the same effect.
- `nmap -PM 10.10.1.1` for netmask request discovery probes.

TCP Ping Scan

TCP SYN Ping

- Initiates the three-way handshake by sending an empty TCP SYN flag.
- After receiving the SYN, the target host acknowledges with an ACK flag.
- With this ACK, the attacker confirms that the host is up. and terminates the connection with the RST flag.
- Default destination port is `80`
- `-PS22` and `-PS22-25,80,113,1050,35000`. Note that there can be no space between `-PS` and the port list. If multiple probes are specified they will be sent in parallel.
- **Advantages:**
    - No connection is established, so no trace.
    - Can run pings on multiple ports in parallel so less chances of timeout.

TCP ACK Ping

- Similar to SYN Ping, here we send an empty TCP ACK packet to the target host directly.
- Since there is no prior connection, the target host will respond with a RST flag to terminate the request.

- **Advantages:**
    - Although both SYN and ACK can be used to maximize the chances of bypassing the firewall.
    - SYN pings are most common, so if they are blocked, it's a good alternative.

IP Protocol Scan

- Latest host discovery option that sends IP ping packets with the IP header of any specified protocol number.
- Multiple IP packets for ICMP (Protocol 1), IGMP (Protocol 2), and IP-in-IP (Protocol 4) are sent by default.
- To change the default protocols, look into `DEFAULT_PROTO_PROBE_PORT_SPEC` in `nmap.h`
- Looks for either responses using the same protocol as a probe, or ICMP protocol unreachable messages which signify that the given protocol isn't supported on the destination host.
- Either type of response signifies that the target host is alive.

Ping Sweep Tools

- Angry IP Scanner
- SolarWinds Engineer's Toolset
- NetScanTools Pro
- OpUtils

Script your own Ping Sweep in Python3

```
import os

IP = input("[+] Enter the Host IP Address:\t")

print("[+] Starting Ping Sweeper on " + IP)

dot = IP.rfind(".")
IP = IP[0:dot + 1]

for i in range(1, 255):
    host = IP + str(i)
    response = os.system("ping -c 1 -w 1 " + host + " >/dev/null")
```



```
    if response == 0:
        print(host + " is up")
    else:
        print(host + " is down")

# Save it as sweeper.py
# Run from the CLI - python3 sweeper.py
```

Port Scanning

- A port scanner is an application which is made to probe a host or server to identify open ports.
- Bad actors can use port scanners to exploit vulnerabilities by finding network services running on a host.
- They can also be used by security analysts to confirm network security policies.

How does Port Scan work?

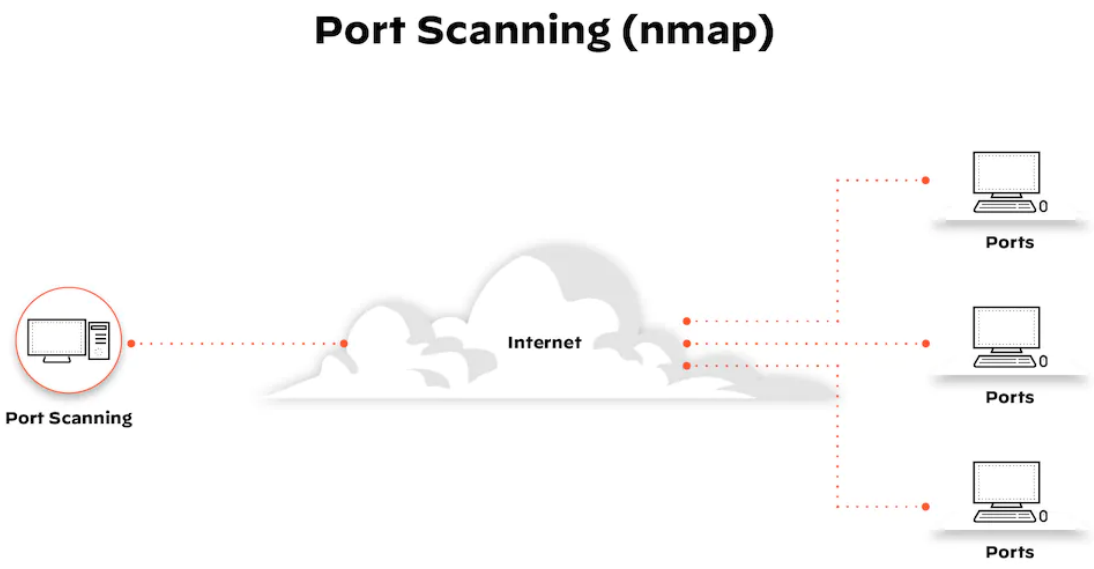


Figure 2: Source: Palo Alto Networks - About Port Scanning

- Running a port scan on a network or server reveals which ports are open and listening (receiving information) as well as revealing the presence of security devices, such as firewalls, that are present between the sender and the target.
- This technique is known as fingerprinting.
- It is also valuable for testing network security and the strength of the system's firewall.
- Due to this functionality, it is also a popular reconnaissance tool for attackers seeking a weak point of access to break into a computer.

About Ports

- Ports vary in their services offered. They are numbered from 0 to 65535, but certain ranges are more frequently used.
- Ports 0 to 1023 are identified as the "well-known ports" or standard ports and have been assigned services by the Internet Assigned Numbers Authority (IANA).
- Some of the most prominent ports and their assigned services include:
    - **Port 20 (UDP)** — File Transfer Protocol (FTP) for data transfer
    - **Port 22 (TCP)** — Secure Shell (SSH) protocol for secure logins, FTP, and port forwarding

- **Port 23 (TCP)** — Telnet protocol for unencrypted text commutations
- **Port 53 (UDP)** — Domain Name System (DNS) translates names of all computers on internet-to-IP addresses
- **Port 80 (TCP)** — World Wide Web HTTP

There are standard services offered on ports after 1023 as well and ports that, if open, indicate an infected system due to its popularity with some far-reaching Trojans and viruses.

Common Ports

packetlife.net

TCP/UDP Port Numbers

7	Echo	**554**	RTSP	**2745**	Bagle.H	**6891-6901**	Windows Live
19	Chargen	**546-547**	DHCPv6	**2967**	Symantec AV	**6970**	Quicktime
20-21	FTP	**560**	rmonitor	**3050**	Interbase DB	**7212**	GhostSurf
22	SSH/SCP	**563**	NNTP over SSL	**3074**	XBOX Live	**7648-7649**	CU-SeeMe
23	Telnet	**587**	SMTP	**3124**	HTTP Proxy	**8000**	Internet Radio
25	SMTP	**591**	FileMaker	**3127**	MyDoom	**8080**	HTTP Proxy
42	WINS Replication	**593**	Microsoft DCOM	**3128**	HTTP Proxy	**8086-8087**	Kaspersky AV
43	WHOIS	**631**	Internet Printing	**3222**	GLBP	**8118**	Privoxy
49	TACACS	**636**	LDAP over SSL	**3260**	iSCSI Target	**8200**	VMware Server
53	DNS	**639**	MSDP (PIM)	**3306**	MySQL	**8500**	Adobe ColdFusion
67-68	DHCP/BOOTP	**646**	LDP (MPLS)	**3389**	Terminal Server	**8767**	TeamSpeak
69	TFTP	**691**	MS Exchange	**3689**	iTunes	**8866**	Bagle.B
70	Gopher	**860**	iSCSI	**3690**	Subversion	**9100**	HP JetDirect
79	Finger	**873**	rsync	**3724**	World of Warcraft	**9101-9103**	Bacula
80	HTTP	**902**	VMware Server	**3784-3785**	Ventrilo	**9119**	MXit
88	Kerberos	**989-990**	FTP over SSL	**4333**	mSQL	**9800**	WebDAV
102	MS Exchange	**993**	IMAP4 over SSL	**4444**	Blaster	**9898**	Dabber
110	POP3	**995**	POP3 over SSL	**4664**	Google Desktop	**9988**	Rbot/Spybot
113	Ident	**1025**	Microsoft RPC	**4672**	eMule	**9999**	Urchin
119	NNTP (Usenet)	**1026-1029**	Windows Messenger	**4899**	Radmin	**10000**	Webmin
123	NTP	**1080**	SOCKS Proxy	**5000**	UPnP	**10000**	BackupExec
135	Microsoft RPC	**1080**	MyDoom	**5001**	Slingbox	**10113-10116**	NetIQ
137-139	NetBIOS	**1194**	OpenVPN	**5001**	iperf	**11371**	OpenPGP
143	IMAP4	**1214**	Kazaa	**5004-5005**	RTP	**12035-12036**	Second Life
161-162	SNMP	**1241**	Nessus	**5050**	Yahoo! Messenger	**12345**	NetBus
177	XDMCP	**1311**	Dell OpenManage	**5060**	SIP	**13720-13721**	NetBackup
179	BGP	**1337**	WASTE	**5190**	AIM/ICQ	**14567**	Battlefield
201	AppleTalk	**1433-1434**	Microsoft SQL	**5222-5223**	XMPP/Jabber	**15118**	Dipnet/Oddbob
264	BGMP	**1512**	WINS	**5432**	PostgreSQL	**19226**	AdminSecure
318	TSP	**1589**	Cisco VQP	**5500**	VNC Server	**19638**	Ensim
381-383	HP Openview	**1701**	L2TP	**5554**	Sasser	**20000**	Usermin
389	LDAP	**1723**	MS PPTP	**5631-5632**	pcAnywhere	**24800**	Synergy
411-412	Direct Connect	**1725**	Steam	**5800**	VNC over HTTP	**25999**	Xfire
443	HTTP over SSL	**1741**	CiscoWorks 2000	**5900+**	VNC Server	**27015**	Half-Life
445	Microsoft DS	**1755**	MS Media Server	**6000-6001**	X11	**27374**	Sub7
464	Kerberos	**1812-1813**	RADIUS	**6112**	Battle.net	**28960**	Call of Duty
465	SMTP over SSL	**1863**	MSN	**6129**	DameWare	**31337**	Back Orifice
497	Retrospect	**1985**	Cisco HSRP	**6257**	WinMX	**33434+**	traceroute
500	ISAKMP	**2000**	Cisco SCCP	**6346-6347**	Gnutella		
512	rexec	**2002**	Cisco ACS	**6500**	GameSpy Arcade		
513	rlogin	**2049**	NFS	**6566**	SANE		
514	syslog	**2082-2083**	cPanel	**6588**	AnalogX		
515	LPD/LPR	**2100**	Oracle XDB	**6665-6669**	IRC		
520	RIP	**2222**	DirectAdmin	**6679/6697**	IRC over SSL		
521	RIPng (IPv6)	**2302**	Halo	**6699**	Napster		
540	UUCP	**2483-2484**	Oracle DB	**6881-6999**	BitTorrent		

Legend

- Chat
- Encrypted
- Gaming
- Malicious
- Peer to Peer
- Streaming

IANA port assignments published at **http://www.iana.org/assignments/port-numbers**

by Jeremy Stretch

TCP Connect/Full-Open Scan

- One of the most reliable, the OS's TCP `connect()` system call tries to open a connection to every port of interest on the target machine.
- Making separate `connect()` call is linear and would take a long time over a slow connection.
- We can accelerate the scan using many sockets in parallel. Using non-blocking, I/O allows us to set a short time-out period and watch all the sockets simultaneously.
- The drawback is this type of scan is easily detectable and filterable.
- `nmap -sT 10.10.10.1`

Stealth Scan (Half-Open Scan)

- Resetting the TCP connection b/w the client and server abruptly before completion of the three-way handshake, hence making it half open.
- It sends a single frame to a TCP port without any TCP Handshaking.
- It's also called "SYN scan" as it sends only the SYN packet.
- `nmap -sS 10.10.1.1`

Inverse TCP Flag Scan

- Sends TCP probe with TCP flags (FIN, URG, PSH) set or with no flags (NULL).
- When the port is open, we do not get any response from the host, whereas when the port is closed, we get the RST from the target host.
- Firewalls, IDS, etc can detect the SYN packets sent to the sensitive ports. Programs such as Syslog are available to log half-open SYN flag scan attempts.
- And in contrast sometimes a probe packet enabled with TCP flags can pass through filters undetected, depending on the design.
- According to RFC 793, an RST/ACK packet is sent for connection reset when the host closes a port. So we can take advantage of this feature to send TCP probe packets to each port with various TCP Flags set.
    - With FIN flag set.
    - Xmas probe with FIN, URG and PUSH flags set.
    - NULL probe with no flags set.
    - SYN/ACK probe.
- All closed ports on the targeted host will send RST/ACK responses. Since OS like windows completely ignore the RFC 793, you cannot see the RST/ACK response. But it works with a UNIX based OS.

TCP Maimon Scan

- Uses FIN/ACK Probe and expects the response to be dropped in BSD systems if the port is open.
- The port is closed if the response is RST. It's filtered if ICMP unreachable error type 3, code 1,2,3,9,10 or 13 is returned.
- `nmap -sM 10.1.1.1`

ACK Flag probe scan

- To analyze the header information (TTL and WINDOW field) of the received RST packets.
- ACK flag probe scan exploits the vulnerability within the BSD-derived TCP/IP stack.
    - **TTL Based ACK flag probe scanning:**
        - Will send ACL probe packets (several thousands) to different TCP ports and then analyze the TTL field value of the RST packets received.
        - `nmap -ttl [time] [target]`
        - If the TTL value of RST packet is less than boundary value 64, then that port is open.
    - **Window-based ACK flag probe scanning:**
        - Similar to the above one, and we will analyze the window field instead.
        - This can be used when all ports return the same TTL.
        - If the window value of the RST packet on a particular port is non-zero, then that port is open.
        - `nmap -sW 10.1.1.1`

IDLE/IPID Header Scan

- Sends a spoofed source address (A zombie device) to a computer to find what services are available.
- Every IP packet on the internet has an "IP identifier" IPID that uniquely identifies fragments of an original IP datagram.
- The OS increases the IPID for each packet sent. Thus proving an IPID reveals to an attacker the number of packets sent since the last probe.
- `nmap -sI 10.10.2.2`
- Scan Steps:
    - 1. Identify the zombie. A zombie that incrementally assigns IPID packets on a global basis is an appropriate or idle zombie for performing idle scans. Probe a zombie with SYN+ACK packet and obtain its IPID from the RST packet that comes as a response because naturally the Zombie Machine won't expect a SYN+ACK packet.
    - 2. Send a SYN packet to the target machine by spoofing the IP to the zombie machine.

- 1. If the port is open the target sends SYN+ACK packet to the zombie to proceed with a three way handshake. Zombie machine responds with the RST packet with an incremented IPID.
- 2. If the port is closed, the target responds with an RST packet and the zombie remains idle.

3. Follow step 1 again to probe the IPID Number.
   - Send a SYN+ACK packet to the zombie and it responds with an RST packet containing the IPID.
   - Assuming that the port on the target was open and that the zombie has already sent an RST packet to the target, the IPID number is increased by 1.
   - Now, the zombie responds with an RST packet to the attacker using its next IPID. Consequently, the IPID is increased by 2 which implies that the port on the target machine was open.

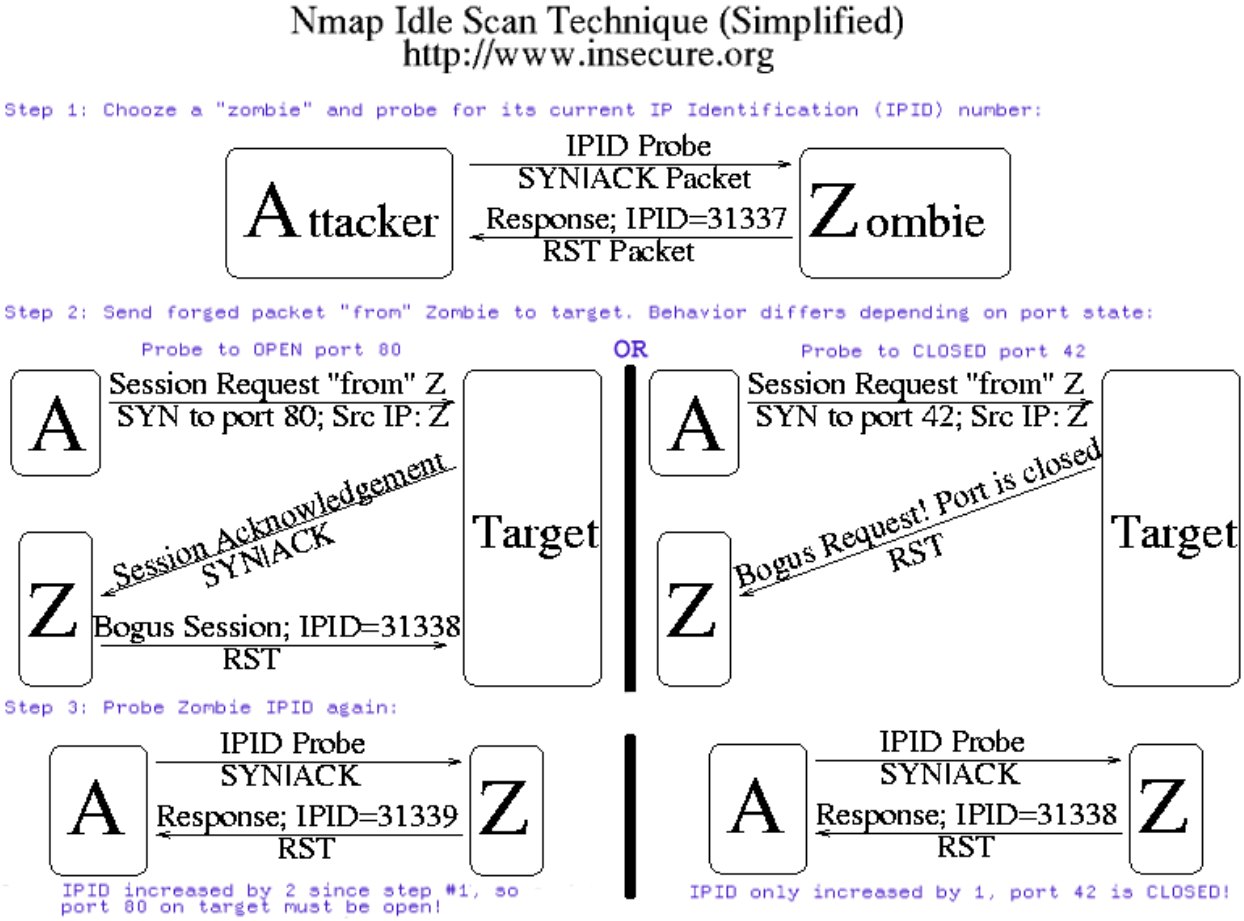


Figure 3: Nmap Idle Scan Technique - Source: insure.org

UDP Raw ICMP Port Unreachable Scan

- 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.
- `nmap -sU 10.10.1.1`
- This is slow because it limits the ICMP error message rate as a form of compensation to machines that apply RFC 1812 (Requirements for IP Version 4 Routers)

UDP RECVFROM() and WRITE() Scanning

- Although non-root users cannot read unreachable port errors directly, linux informs you indirectly when it receives messages.
- Eg: 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," errorno 111)` otherwise.
- This technique is used for determining open ports when non-root users use `-u`. Root users can also use the `-l` (lamer UDP scan) option to force this process.

SCTP INIT Scan

- Stream Control Transport Protocol, reliable message-oriented transport layer protocol. Alternative to TCP and UDP.
- Used to perform multihoming and multi-streaming activities. Some applications include discovering VoIP, IP telephony and signaling system related services.
- Performs a four-way handshake

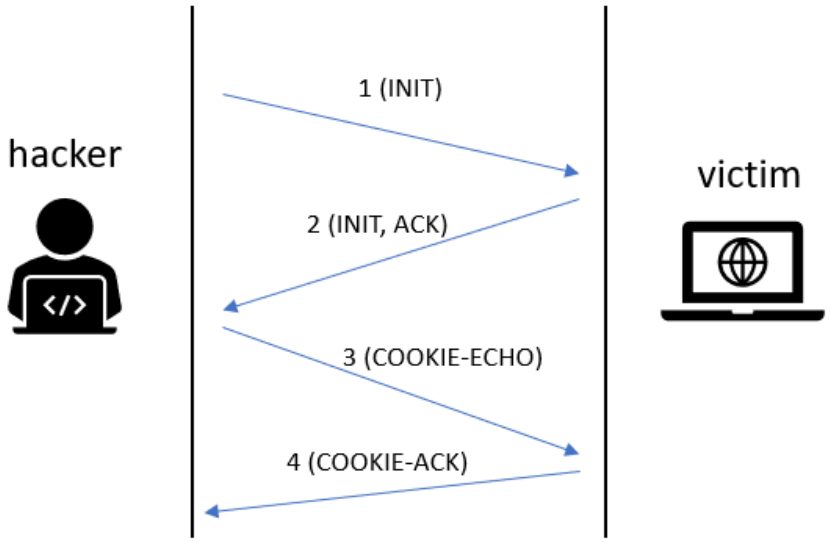


Figure 4: SCTP Init Scan

- 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.
- Can make it half-open, stealthy by
  1. sending INIT chunk to target and if the port is open, it sends INIT+ACK.
  2. If the target is inactive or closed, it sends an ack as an ABORT chunk.
  3. After several retransmissions if there is no response, then filtered.
- `nmap -sY 10.10.1.1`

SCTP COOKIE ECHO Scan

- Attacker sends the COOKIE ECHO chunk to target, and if the target port is open, it will silently drop the packets creating the possibility of being open.
- If ABORT is returned it means it's closed.
- `nmap -sZ 10.1.1.1`

SSDP Scan

- Simple Service Discovery Protocol is a network protocol that generally communicates with machines when querying them with routable IPv4 or IPv6 multicast addresses.
- Controls communication for the Universal Plug and Play (UPnP) feature. It generally works when the machine is not firewalled; however it can sometimes work through one too.
- SSDP service will respond to a query sent over IPv4 or IPv6 broadcast addresses.

- The response includes info about the UPnP feature associated with it, the attacker uses SSDP scanning to detect UPnP vulns that may lead to buffer overflow or DoS attacks.

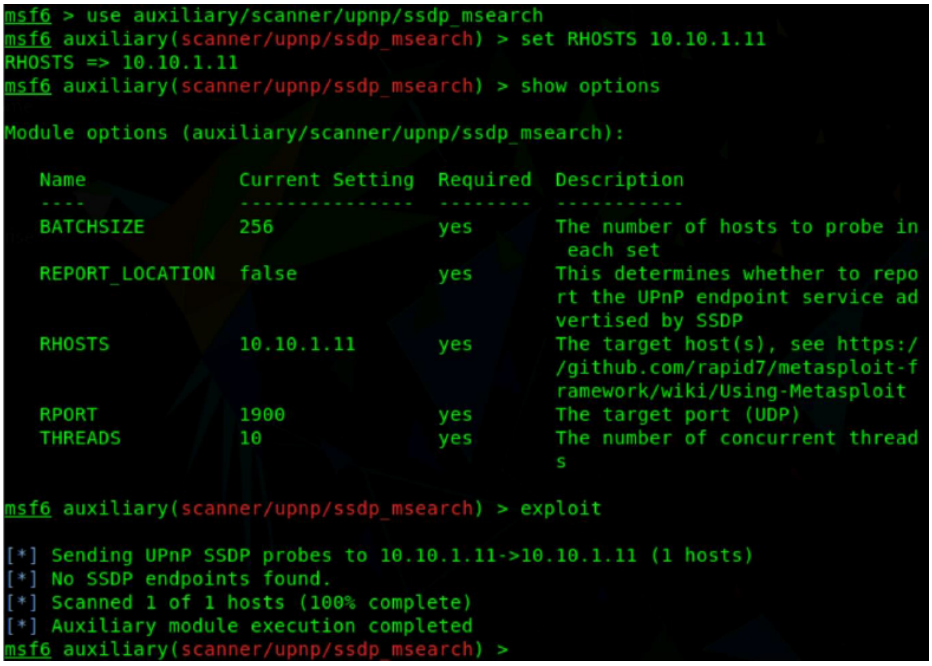


Figure 5: UPnP Scanning

- UPnP SSDP M-SEARCH information discovery tool to check whether the machine is vulnerable to UPnP exploits.

List Scan

- Discovery of the active network hosts is indirect. without actually pinging em.
- By default, a reverse DNS resolution is still carried out on each host by Nmap to learn their names.
- `nmap -sL 10.1.1.1`

IPv6 Scan

- `namp -6 hostname.org`

Service Version Discovery

- Each port is assigned a specific service, and every service has its own version. Some versions of the protocols are insecure and they can be exploited.
- We gotta find the running services and detect their versions to see if they are exploitable.

- Version can be detected by examining 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.
- `nmap -sV 10.1.1.1` to detect versions.

Nmap Scan Time Reduction Techniques

- Omit Non-critical tests
    - Avoiding an intense scan if only a minimal is required.
    - Port scan can be skipped with -sn if the scan is to check host availability.
    - DNS resolution should be turned on only when necessary.
- Optimize timing parameters with -T
- 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.
- Upgrade Nmap
- Execute Concurrent Nmap Instances
- Scan from a favorable Network Location

OS Discovery/Banner Grabbing

- a.k.a OS Fingerprinting is a method used to determine the OS that is running on a remote target system. To exploit OS specific vulnerabilities.
- 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/bins/ls to check the system architecture. A more advanced one depends on stack querying, which transfers the packets to the network host and evaluates them by the reply.
    - The next method, a.k.a Initial Sequence Number (ISN) analysis, identifies the differences in random number generators found in the TCP stack. ICMP response analysis is another method that evaluates the reply.
- Active Banner Grabbing:
    - Attacker sends a variety of malformed packets to the remote host, and the responses are compared with a DB. Responses from different OS vary because of differences in TCP/IP stack implementation.
    - For instance, the scanning utility Nmap uses a series of nine tests to determine an OS fingerprint or banner grabbing.
        - **Test 1:** A TCP packet with the SYN and ECN-Echo flags set is sent to an open TCP port.

        - **Test 2:** A TCP packet with no flags set is sent to open TCP port. Null packet.
        - **Test 3:** A TCP packet with URG, PSH, SYN and FIN set is sent to open TCP port.
        - **Test 4:** A TCP packet with ACK flag set is sent to open TCP port.
        - **Test 5:** A TCP packet with SYN flag set is sent to closed TCP port.
        - **Test 6:** A TCP packet with the ACK flag enabled is sent to a closed TCP port.
        - **Test 7:** A TCP packet with the URG, PSH and FIN flag set is sent to closed TCP port.
        - **Test 8:** PU (Port Unreachable) - A UDP packet is sent to an closed UDP port. Objective is to extract an "ICMP Port Unreachable" message from the target machine.
        - **Test 9:** TSeq (TCP Sequence ability test) - Determines the sequence generation patterns of the TCP initial sequence numbers (aka TCP ISN sampling), the IP identification numbers (aka IPID sampling), and the TCP timestamp numbers. It sends six TCP packets with the SYN flag enabled to an open TCP port.
- The objective of these tests is to find patterns in the initial sequence of numbers that the TCP implementations chose while responding to a connection request. They can be categorized into groups, such as
    - Traditional 64k (many old UNIX boxes),
    - Random increments (newer versions of Solaris, IRIX, FreeBSD, Digital UNIX, Cray, and many others),
    - Or true random (Linux 2.0.*, OpenVMS, newer AIX, etc).
    - Windows boxes use a "time-dependent" model in which ISN is incremented by a fixed amount for each occurrence.
- Passive Banner Grabbing:
    - 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.
    - Banner grabbing from error messages.
    - Sniffing the network traffic.
    - Banner grabbing from page extensions - For example: .aspx indicates IIS server and Windows platform.
    - The four areas that typically determine the OS
        - TTL
        - Window size
        - Dont' Fragment (DF) bit is set or not
        - Type of Service (TOS)
- The following is an analysis of a sniffed packet described by Lance Spitzner in his paper on passive fingerprinting

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

- According to the four criteria,
    - TTL: 45 - Might have gone through 19 hops from 64 -> 45 indicating Linux or FreeBSD box.
    - Window Size: 0x7D78 or 32120 in decimal - It's the default window size used by Linux. In addition, FreeBSG and Solaris tend to maintain the same window sizes throughout a session, however Cisco router and Microsoft Windows NT changes constantly. Window Size is more accurate when measured after the initial 3-way handshake due to TCP slow start.
    - DF: is set - Most systems use the DF bit but it's easy to eliminate the ones that don't, like - SCO or OpenBSD.
    - TOS: 0x0

Identifying Target System OS

Operating System	Time To Live	TCP Window Size
Linux	64	5840
FreeBSD	64	65535
OpenBSD	255	16384
Windows	128	65,535 bytes to 1 Gigabyte
Cisco Routers	255	4128
Solaris	255	8760
AIX	255	16384

Figure 6: TTL and TCP Window Size for OS

- `nmap -O 10.1.1.1`
- `unicornscan 10.1.1.1 -Iv`
- `nmap --script smb-os-discovery.nse 10.1.1.1` uses SMB to collect Os information.
- `nmap -6 -O 10.1.1.1` uses IPv4 fingerprinting with 18 probes in the following order:

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

Unit 3: IDS and Firewall Evasion

Packet Fragmentation

- Splitting of a probe packet into several smaller packets while sending it to a network. generally, these packets are sent to a queue from where they are checked one by one.
- But due to CPU and network resource consumption, the configuration of most ODS causes them to skip fragmented packets during port scans.
- So can use nmap or fragroute to split the probe packet.
- Since many IDS use signature-based methods to indicate scanning attempts on IP and/or TCP headers, the use of fragmentation will often evade this type of packet filtering and detection.
- SYN/FIN Scanning Using IP Fragments - `nmap -f 10.1.1.1`

Source Routing Manipulation

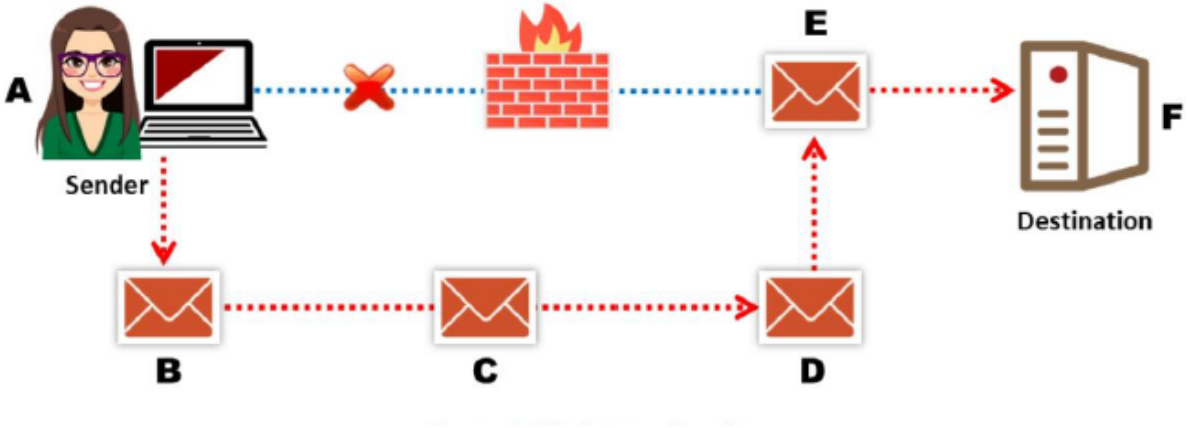


Figure 7: Source Routing Manipulation

- IP options field stores source routing information and includes a list of IP addresses through which the packet travels to its destination.
- Attacker sends 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
- They manipulate the IP address path in the IP options field so that the packet takes the attacker-defined path (without firewalls or IDS) to reach the destination.

Source Port Manipulation

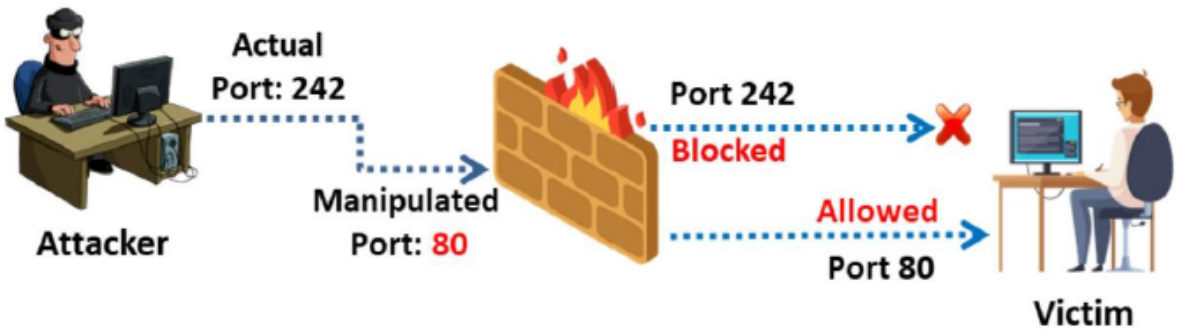


Figure 8: Source Port Manipulation

- Occurs when the firewall blindly trusts the source port number. The admins mostly configure the firewall by allowing the incoming traffic from well-known ports such as HTTP, DNS, FTP, etc.
- Attacker manipulates the original port number with a common port number which easily bypasses the IDS/firewall.
- `nmap -g 80 10.1.1.1` or `nmap --source-port 80 10.1.1.1`

IP Address Decoy

- 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 hosts 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.
- Nmap comes with a built-in scan function called decoy scan which cloaks a scan with decoys. It generates multiple IP addresses to perform a scan, thus making it difficult for the target security mechanism to identify the original source from the registered logs.
- `nmap -D RND: 10 128.54.6.3` or in more granular way we can specify IPs like `nmap -D 192.168.0.1,192.168.0.2,192.168.0.3,192.168.0.4 126.45.54.1`

IP Address Spoofing

- Hijacking technique in which an attacker obtains a computer's IP address, alters the packet headers, and sends request packets to a target machine.
- `hping3 www.hacker.com -a 8.8.8.8`

MAC Address Spoofing

- Similar to IP spoofing, we can spoof the MAC too. `--spoof-mac` option is available in nmap. `nmap --spoof-mac 0 215.2.21.3` randomize a MAC address for you.
- You can also mention vendor specific MAC addresses like, `nmap --spoof-mac Dell 156.21.51.1`
- Or we can specify our own MAC of choice with `nmap --spoof-mac 00:01:02:25:56:AE 126.52.15.65`

Creating Custom Packets

- Colasoft packet builder
- NetScanTools

Randomizing Host Order

- Can use --randomize-hosts option in Nmap like nmap --randomize-hosts 12.1.2.1
- Sending Bad Checksums
- nmap --badsum 121.125.12.21 sends packets with invalid TCP, UDP or SCTP checksums to the target host.
- Proxy Servers
- Can chain multiple proxy servers to increase anonymity.
- Proxy Switcher (proxyswitcher.com)
- CyberGhost VPN
- For Mobile:
    - Shadowsocks
    - ProxyDroid

Anonymizers

- Removes all identity information from the user's computer while the user surfs the internet.
- Whonix is a desktop OS designed for advanced security and privacy.
- Phiphon
- TunnelBear
- Invisible Internet Project (I2P)
- For Mobile:
    - Orbot
    - Phiphon pro
- Censorship circumvention tools
    - Alkasir - Cross platform open source website censorship circumvention tool.
    - Tails - live OS for advanced privacy and security.

Network Scanning Countermeasures

Being a pentest it is also important to adopt countermeasures against the respective vulnerabilities determined through hacking.

Port Scanning Countermeasures

- Configure firewall and IDS rules to detect and block probes.
- Ensure the router, IDS and firewall firmware are updated with their latest releases.
- Employ network IDS and IPS tools like Snort.
- Keep as few ports open as possible and filter the rest.
- Block inbound ICMP message types and all outbound ICMP type-3 unreachable messages at border routers arranged in front of the company's main firewall.

Banner Grabbing Countermeasures

- Display false banners to mislead attackers.
- turn off unnecessary services on the network.
- Use server masking tools to disable or change banner information
- Remove unnecessary HTTP headers and response data and camouflage the server by providing false signatures.
- For Apache 2.x with the `mod_headers` module, use a directive in the httpd.conf file to change the banner information header and set the server as New server name.

- Alternatively, change the ServerSignature line to `ServerSignatureOff` in the `httpd.conf` file.
- Modify the value of Server Tokens from Full to Prod in apache's httpd.conf file to prevent disclosure of the server version.
- Modify the value of `RemoveServerHeader` from 0 to 1 in the UrlScan.ini config found at c: `WindowsSystem32inetservUrlscan` This method prevents disclosure of the server version.
- Trick attackers by modifying the value of AlternateServerName to values such as xyz or myserver.
- Disable HTTP methods such as Connect, Put, Delete, and Options from web application servers.
- Remove `X-Powered-By` header only with the customHeaders option in the `<system.webServer>` section of the web.config file.
- Hiding Gile Extensions from Web Pages
    - Apache users can use `mod_negotiation` directives.
    - IIS users can use tools such as PageXchanger.

IP Spoofing Detection Techniques

- Send a packet to the host of a suspected spoofed packet that triggers a reply and compare the TTL / IPID / Window Size of the suspected packet.
- If those are not the same as the packet being checked, this implies it is a spoofed packet.

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 algo 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 says the max amount of data that the recipient can receive and the max amount of data that can be sent without ack.
- So, the sender should stop sending data whenever the window size is set to zero. The attacker, who is unaware of the ACK packet containing window size might continue to send data to the victim.
- If data packets received are beyond the window size, they are spoofed packets. For effective flow control and early detection of spoofing, the initial window size must be very small.
- Most spoofing attacks occur during handshake, as it is challenging to build multiple spoofing replies with the correct sequence number.
- 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.

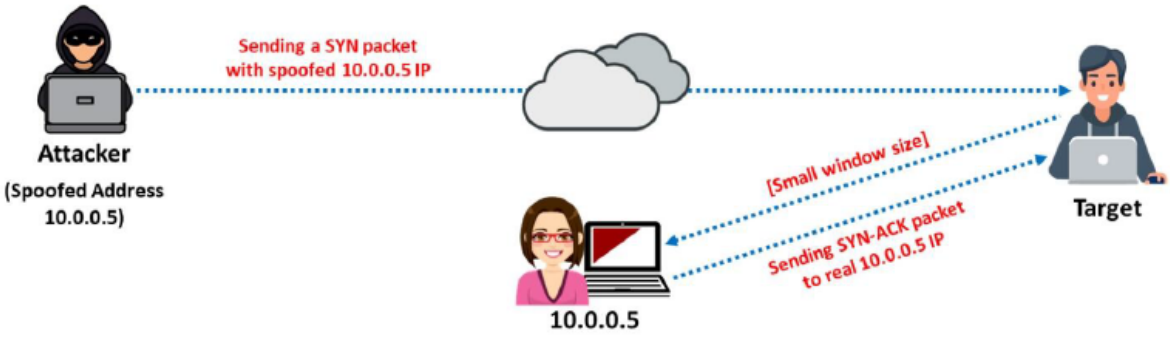


Figure 9: TCP Flow Control

- If the sender sends an ACK with any data, it means that the sender is a spoofed one. This is because for the 0 SYN-ACK, the reply should be an ACK packet without additional data.
- Also since the attacker won't receive the SYN-ACK packets, the attacker cannot respond to changes in the congestion window size. When the received traffic continues after a window size is exhausted, the packets are most likely spoofed.

IP Spoofing Countermeasures

- **Avoid trust relationships:** It is advisable to test all packets, even when they originate from a trusted host as attackers might have masqueraded as trusted hosts.
- **Use Firewalls and Filtering Mechanisms:** All incoming and outgoing packets should be filtered to avoid attacks and loss of sensitive information. ACLs can be used to block access. Concentration on filtering outgoing traffic is also important in terms of insider threat and data exfiltration by the attacker.
- **Use random Initial Sequence Numbers:** Make it less predictable, the attacker can determine the ISN of the next TCP connection by analyzing an ISN of the current session.
- **Ingress & Egress Filtering:** Prevents spoofed traffic from entering the internet and blocking outgoing packets with a source address from outside.
- **Use Encryption:** Use strong encryption for all traffic placed on transmission media without considering its type and location. IPSec can be used to provide data authentication, integrity, and confidentiality.
- SYN flooding countermeasures will help against IP spoofing attacks.

- **Other countermeasures includes:**
    - Enhancing integrity and confidentiality of websites by migrating from IPv4 to IPv6 during development.
    - Implement digital cert auth mechanisms such as domain and two-way auth certificate verification.
    - Using a secure VPN
    - Employing application-specific mitigation devices such as Behemoth Scrubbers for deep level packet investigation at a high speed of nearly 100 million packets / second.
    - Hide intranet hosts from external networks by implementing modifications to the network address translation - NAT.
    - Configure internal switches to table the DHCP static addresses to filter malicious spoofed traffic.

Scanning Detection and Prevention Tools

- ExtraHop.
- Splunk.
- Scanlogd.
- IBM Security QRadar XDR.
- Cynet 360.
- Vectra Cognito Detect.

Thank you for reading all along. We hope this book added some value to you and improved your skillset. Sign Up at hacklido.com and follow us for more content to upskill your cyber-skills!