Great! You're now moving into **Nmap Stage 4** — **Host Discovery & ARP Scanning**, which is essential for identifying *which* systems are actually live before deep scanning them.

**🔍 Nmap Stage 4: Host Discovery & ARP Scanning**

When you're scanning a network, you **first need to find which hosts are up** before you start probing their ports. This is called **host discovery** or **ping scanning**.

**🚨 Why Host Discovery?**

* Avoid wasting time scanning dead IPs.
* Helps enumerate the live systems in a subnet.
* Key part of **recon** phase in penetration testing/red teaming.

**🧪 1. Basic Host Discovery with Nmap**

nmap -sn 192.168.1.0/24

* -sn (ping scan only): Tells Nmap to skip port scan and **only check which hosts are up**.
* Output: Shows a list of live hosts in that subnet.

**🛜 2. ARP Scan (Layer 2 Discovery) – *Local Network Only***

sudo nmap -sn 192.168.1.0/24

* When used **on a local network**, Nmap automatically sends **ARP requests** instead of ICMP.
* This is **very accurate**, as devices *must* reply to ARP on the LAN.

✅ Best way to discover all machines in your subnet (even firewalled ones that block ICMP).

**🛠 Example Output:**

Nmap scan report for 192.168.1.10

Host is up (0.00042s latency).

MAC Address: 08:00:27:12:34:56 (Oracle VirtualBox)

**⚙️ 3. ICMP Echo Request (Ping)**

nmap -PE -sn 192.168.1.0/24

* -PE: Use **ICMP Echo Request** (like classic ping).
* Might be blocked by firewalls, so not always reliable.

**🧯 4. TCP Ping Discovery**

nmap -PS80,443 -sn 192.168.1.0/24

* -PS: TCP SYN Ping (on port 80, 443).
* Useful if ICMP is blocked — some systems respond to TCP pings.

**🔐 5. Disable DNS Resolution (Faster)**

nmap -n -sn 192.168.1.0/24

* -n: Skip DNS resolution — makes the scan **much faster**.

**🛠 Pro Mode: Combining All Techniques**

sudo nmap -PE -PS80,443 -PA21,23 -PU53 -sn 192.168.1.0/24

* -PA: TCP ACK Ping (to bypass some firewalls).
* -PU: UDP Ping (port 53 commonly allowed).
* Multiple ping types = **better coverage**.

**🎯 Bonus: Target Specific Hosts from List**

nmap -sn -iL targets.txt

Where targets.txt contains:

192.168.1.10

192.168.1.20

**📦 Summary**

| **Technique** | **Command Example** | **Use Case** |
| --- | --- | --- |
| ARP Scan | sudo nmap -sn 192.168.1.0/24 | Best for local LAN |
| ICMP Ping | nmap -PE -sn 192.168.1.0/24 | Classic ping, often blocked |
| TCP Ping | nmap -PS80,443 -sn 192.168.1.0/24 | Good for systems blocking ICMP |
| UDP Ping | nmap -PU53 -sn 192.168.1.0/24 | Useful when UDP traffic allowed |
| No DNS | nmap -n -sn 192.168.1.0/24 | Speed up scans by skipping name lookup |

Want a **lab challenge** using ARP scan and host discovery in VirtualBox or real networks? Let me know, and I’ll guide you step-by-step.

**Host Discovery Shortcuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sL | nmap 192.168.1.1-3 -sL | No Scan. List targets only |
| -sn | nmap 192.168.1.1/24 -sn | Disable port scanning. Host discovery only. |
| -Pn | nmap 192.168.1.1-5 -Pn | Disable host discovery. Port scan only. |
| -PS | nmap 192.168.1.1-5 -PS22-25,80 | TCP SYN discovery on port x.  Port 80 by default |
| -PA | nmap 192.168.1.1-5 -PA22-25,80 | TCP ACK discovery on port x.  Port 80 by default |
| -PU | nmap 192.168.1.1-5 -PU53 | UDP discovery on port x.  Port 40125 by default |
| -PR | nmap 192.168.1.1-1/24 -PR | ARP discovery on local network |
| -n | nmap 192.168.1.1 -n | Never do DNS resolution |

**Port Specification Shortcuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -p | nmap 192.168.1.1 -p 21 | Port scan for port x |
| -p | nmap 192.168.1.1 -p 21-100 | Port range |
| -p | nmap 192.168.1.1 -p U:53,T:21-25,80 | Port scan multiple TCP and UDP ports |
| -p- | nmap 192.168.1.1 -p- | Port scan all ports |
| -p | nmap 192.168.1.1 -p http,https | Port scan from service name |
| -F | nmap 192.168.1.1 -F | Fast port scan (100 ports) |
| --top-ports | nmap 192.168.1.1 --top-ports 2000 | Port scan the top x ports |
| -p-65535 | nmap 192.168.1.1 -p-65535 | Leaving off initial port in range makes the scan start at port 1 |
| -p0- | nmap 192.168.1.1 -p0- | Leaving off end port in range  makes the scan go through to port 65535 |

**Service and Version Detection Shortcuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sV | nmap 192.168.1.1 -sV | Attempts to determine the version of the service running on port |
| -sV --version-intensity | nmap 192.168.1.1 -sV --version-intensity 8 | Intensity level 0 to 9. Higher number increases possibility of correctness |
| -sV --version-light | nmap 192.168.1.1 -sV --version-light | Enable light mode. Lower possibility of correctness. Faster |
| -sV --version-all | nmap 192.168.1.1 -sV --version-all | Enable intensity level 9. Higher possibility of correctness. Slower |
| -A | nmap 192.168.1.1 -A | Enables OS detection, version detection, script scanning, and traceroute |

**OS Detection Short cuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -O | nmap 192.168.1.1 -O | Remote OS detection using TCP/IP stack fingerprinting |
| -O --osscan-limit | nmap 192.168.1.1 -O --osscan-limit | If at least one open and one closed TCP port are not found it will not try OS detection against host |
| -O --osscan-guess | nmap 192.168.1.1 -O --osscan-guess | Makes Nmap guess more aggressively |
| -O --max-os-tries | nmap 192.168.1.1 -O --max-os-tries 1 | Set the maximum number x of OS detection tries against a target |
| -A | nmap 192.168.1.1 -A | Enables OS detection, version detection, script scanning, and traceroute |

**Timing and Performance Shortcuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -T0 | nmap 192.168.1.1 -T0 | Paranoid (0) Intrusion Detection System evasion |
| -T1 | nmap 192.168.1.1 -T1 | Sneaky (1) Intrusion Detection System evasion |
| -T2 | nmap 192.168.1.1 -T2 | Polite (2) slows down the scan to use less bandwidth and use less target machine resources |
| -T3 | nmap 192.168.1.1 -T3 | Normal (3) which is default speed |
| -T4 | nmap 192.168.1.1 -T4 | Aggressive (4) speeds scans; assumes you are on a reasonably fast and reliable network |
| -T5 | nmap 192.168.1.1 -T5 | Insane (5) speeds scan; assumes you are on an extraordinarily fast network |
|  |  |  |

| **Switch** | **Example input** | **Description** |
| --- | --- | --- |
| --host-timeout <time> | 1s; 4m; 2h | Give up on target after this long |
| --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time> | 1s; 4m; 2h | Specifies probe round trip time |
| --min-hostgroup/max-hostgroup <size<size> | 50; 1024 | Parallel host scan group sizes |
| --min-parallelism/max-parallelism <numprobes> | 10; 1 | Probe parallelization |
| --scan-delay/--max-scan-delay <time> | 20ms; 2s; 4m; 5h | Adjust delay between probes |
| --max-retries <tries> | 3 | Specify the maximum number of port scan probe retransmissions |
| --min-rate <number> | 100 | Send packets no slower than <numberr> per second |
| --max-rate <number> | 100 | Send packets no faster than <number> per second |

**NSE Scripts Short cuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -sC | nmap 192.168.1.1 -sC | Scan with default NSE scripts. Considered useful for discovery and safe |
| --script default | nmap 192.168.1.1 --script default | Scan with default NSE scripts. Considered useful for discovery and safe |
| --script | nmap 192.168.1.1 --script=banner | Scan with a single script. Example banner |
| --script | nmap 192.168.1.1 --script=http\* | Scan with a wildcard. Example http |
| --script | nmap 192.168.1.1 --script=http,banner | Scan with two scripts. Example http and banner |
| --script | nmap 192.168.1.1 --script "not intrusive" | Scan default, but remove intrusive scripts |
| --script-args | nmap --script snmp-sysdescr --script-args snmpcommunity=admin 192.168.1.1 | NSE script with arguments |

Useful NSE Script Examples

| **Command** | **Description** |
| --- | --- |
| nmap -Pn --script=http-sitemap-generator scanme.nmap.org | http site map generator |
| nmap -n -Pn -p 80 --open -sV -vvv --script banner,http-title -iR 1000 | Fast search for random web servers |
| nmap -Pn --script=dns-brute domain.com | Brute forces DNS hostnames guessing subdomains |
| nmap -n -Pn -vv -O -sV --script smb-enum\*,smb-ls,smb-mbenum,smb-os-discovery,smb-s\*,smb-vuln\*,smbv2\* -vv 192.168.1.1 | Safe SMB scripts to run |
| nmap --script whois\* domain.com | Whois query |
| nmap -p80 --script http-unsafe-output-escaping scanme.nmap.org | Detect cross site scripting vulnerabilities |
| nmap -p80 --script http-sql-injection scanme.nmap.org | Check for SQL injections |

**Firewall / IDS Evasion and Spoofing Short cuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -f | nmap 192.168.1.1 -f | Requested scan (including ping scans) use tiny fragmented IP packets. Harder for packet filters |
| --mtu | nmap 192.168.1.1 --mtu 32 | Set your own offset size |
| -D | nmap -D 192.168.1.101,192.168.1.102, 192.168.1.103,192.168.1.23 192.168.1.1 | Send scans from spoofed IPs |
| -D | nmap -D decoy-ip1,decoy-ip2,your-own-ip,decoy-ip3,decoy-ip4 remote-host-ip | Above example explained |
| -S | nmap -S www.microsoft.com www.facebook.com | Scan Facebook from Microsoft (-e eth0 -Pn may be required) |
| -g | nmap -g 53 192.168.1.1 | Use given source port number |
| --proxies | nmap --proxies http://192.168.1.1:8080, http://192.168.1.2:8080 192.168.1.1 | Relay connections through HTTP/SOCKS4 proxies |
| --data-length | nmap --data-length 200 192.168.1.1 | Appends random data to sent packets |

Example IDS Evasion command

nmap -f -t 0 -n -Pn –data-length 200 -D 192.168.1.101,192.168.1.102,192.168.1.103,192.168.1.23 192.168.1.1

**Output Shortcuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -oN | nmap 192.168.1.1 -oN normal.file | Normal output to the file normal.file |
| -oX | nmap 192.168.1.1 -oX xml.file | XML output to the file xml.file |
| -oG | nmap 192.168.1.1 -oG grep.file | Grepable output to the file grep.file |
| -oA | nmap 192.168.1.1 -oA results | Output in the three major formats at once |
| -oG - | nmap 192.168.1.1 -oG - | Grepable output to screen. -oN -, -oX - also usable |
| --append-output | nmap 192.168.1.1 -oN file.file --append-output | Append a scan to a previous scan file |
| -v | nmap 192.168.1.1 -v | Increase the verbosity level (use -vv or more for greater effect) |
| -d | nmap 192.168.1.1 -d | Increase debugging level (use -dd or more for greater effect) |
| --reason | nmap 192.168.1.1 --reason | Display the reason a port is in a particular state, same output as -vv |
| --open | nmap 192.168.1.1 --open | Only show open (or possibly open) ports |
| --packet-trace | nmap 192.168.1.1 -T4 --packet-trace | Show all packets sent and received |
| --iflist | nmap --iflist | Shows the host interfaces and routes |
| --resume | nmap --resume results.file | Resume a scan |

Helpful Nmap Output examples

| **Command** | **Description** |
| --- | --- |
| nmap -p80 -sV -oG - --open 192.168.1.1/24 | grep open | Scan for web servers and grep to show which IPs are running web servers |
| nmap -iR 10 -n -oX out.xml | grep "Nmap" | cut -d " " -f5 > live-hosts.txt | Generate a list of the IPs of live hosts |
| nmap -iR 10 -n -oX out2.xml | grep "Nmap" | cut -d " " -f5 >> live-hosts.txt | Append IP to the list of live hosts |
| ndiff scanl.xml scan2.xml | Compare output from nmap using the ndif |
| xsltproc nmap.xml -o nmap.html | Convert nmap xml files to html files |
| grep " open " results.nmap | sed -r 's/ +/ /g' | sort | uniq -c | sort -rn | less |  |

**Miscellaneous Options Other Useful Nmap Commands Short cuts**

| **Switch** | **Example** | **Description** |
| --- | --- | --- |
| -6 | nmap -6 2607:f0d0:1002:51::4 | Enable IPv6 scanning |
| -h | nmap -h | nmap help screen |

| **Command** | **Description** |
| --- | --- |
| nmap -iR 10 -PS22-25,80,113,1050,35000 -v -sn | Discovery only on ports x, no port scan |
| nmap 192.168.1.1-1/24 -PR -sn -vv | Arp discovery only on local network, no port scan |
| nmap -iR 10 -sn -traceroute | Traceroute to random targets, no port scan |
| nmap 192.168.1.1-50 -sL --dns-server 192.168.1.1 | Query the Internal DNS for hosts, list targets only |

**Scanning a Target via nmap commands**

Target specification

Nmap provides multiple target specification options with which a user can mention subnets, individual IPs, IP ranges, and IP lists to be scanned. This will allow the user to scan specific hosts identified from the host discovery.

A sample complete syntax of Nmap is as follows:

Nmap -sS -sV -PN -T4 -oA testsmtp -p T:25 -v -r 192.168.1.\*

\* Note: You can type your own IP

**How to specify a target**

The nmap command interprets any content appended without an associated switch as a target. The following is a basic syntax that specifies an IP address or a hostname to scan without any associated switches:

**nmap 127.0.0.1nmap localhost**

The hostname is resolved with the configured DNS server and the IP address is obtained to perform the scan. If multiple IP address are associated with one hostname, the first IP address will be scanned and the result will be displayed. The following syntax allows nmap to perform scans on all the IP addresses resolved with the hostname provided in the command:

**nmap xyz.com\***

Nmap also supports scanning the whole subnet, provided that you append the mask at the end of an IP address or hostname. Then, Nmap will consider all the resolved IP addresses in the range of the mask mentioned. For example, 10.0.0.1/24 would scan the 256 hosts between 10.0.0.1 and 10.0.0.255, including .1, and .255. 10.0.0.21/24 would scan exactly the same targets.

Nmap also allows you to resolve an entire subnet and then exclude certain hosts from scanning. For example, the following syntax allows you to scan all the hosts resolved for 10.0.0.1/24 except any IP addresses whose last network bits are .1 or .255:

**nmap 10.0.0.2-254**

This can be used in any of the four network bits, such as 10.0.1-254.1-254, which will allow you to skip IP addresses  10.0.0.0 , 10.0.0.255,  10.0.255.0, and 10.0.255.255. Nmap also supports fully qualified IPv6 addresses, but not octet range. For an IPv6 address with non-global scope, the zone suffix ID needs to be mentioned.

Nmap supports various input formats for a user to specify the targets. The following are the switches that can be used to mention the hosts on the specified format:

**nmap –iL <inputfilename>**

This will allow the user to create a text file with a list of all the IP addresses/range to be scanned. This is a feasible option when you have many IP addresses to be scanned. For example, if you want to scan all the IP addresses from different subnets for a medium-scale organization with more than 10,000 assets, it is not feasible to enter these IP addresses on the command line. Instead, create a text file with a list of all the IP addresses to be scanned and mention the filename with the absolute path after -iL. Nmap then fetches the list of IP addresses from the file and performs the scan:

nmap -iR <num hosts>

For large organizations and internet-based scans, you may want to scan random targets or identify unknown targets. The –iR switch with the appended number of random hosts to be identified for scans will allow the user to perform these operations. For example, if you are trying to identify eight random hosts with the ftp port open, the following syntax can be used:

**nmap –sS –Pn –p 21 –iR 8 --open**

The following syntax will help you to exclude servers when your input is a range of servers, a subnet, or a pre-existing large list of servers. The hosts mentioned along with this switch are omitted from scanning, thereby preventing the servers from being hit with any unwanted traffic:

**nmap --exclude <host1>[,<host2>[,...]]**

The following command works similarly to the preceding syntax, except that the host exclusion list is fetched from a file instead of manually mentioning the server list. This is feasible when the list of hosts to be excluded from the scan is long:

**nmap --excludefile <exclude\_file>**

**How to perform host discovery**

The following options are provided by Nmap to perform host discovery:

* –sL: This option lists the IP addresses present in the provided subnet. It also tries to resolve the IP addresses to their hostnames. The hostnames can help an attacker or a penetration tester find out a great deal about the network. You will not be able to combine this with any other options, such as OS discovery, because the functionality is to just list the IP addresses.
* -sn: This option tells Nmap not to perform a port scan once the host discovery is performed. Instead it just lists out the live IP addresses found. This uses an ICMP echo to identify the available hosts, which will not work if there is a firewall present in the network.
* -Pn (No ping): Generally, Nmap performs activities such as probing, port detection, service detection, and OS detection options only if the hosts are found live. This option allows Nmap to perform all the operations on the list of hosts provided to scan. For example, if a class CIP address with subnet/28is specified, then Nmap performs probing on all the255hosts instead of checking for live hosts and performing the activity on them. This is an extensive scan option and generates a lot of traffic.
* -PS (port list): This option sends an empty TCP packet with SYN flag set. This is also called a syn ping packet. Generally, for a full TCP connection to happen, an ACK is generated by the host on receiving the SYN packet. Once the ACK packet is received, the Nmap host generates a SYN/ACK packet, which then establishes a connection. Instead, Nmap sends an RST, which is a reset flag packet, to drop the connection and thus declare the port to be open. This will allow you to determine the open ports without actually creating a connection, because any connection made will be logged at the network and system levels. This option also allows attackers to not leave any tracks while performing the detection.

**There is no space between -PS and the port number. You can specify a range of ports to perform the operation on as well.**

* -PA(port list): This is similar to SYN scanning and is also known as the TCP ACK ping scan. Nmap generates TCP packets with ACK set. ACK basically acknowledges any data transferred over the connection, but there will be no existing connection from the Nmap machine to the host, thus it returns an RST-flag-enabled packet. This will allow Nmap to determine that the port is open and has a service functioning.
* -PU (port list): This is also similar to TCP scans, but this UDP ping scan is for UDP ports. For most ports the packet is empty, except for any service-specific ports, such as DNS and NTP. If a DNS ping packet reaches a closed port, the UDP probe should trigger an ICMP unreachable response from the host. If this response is not generated or the connection appears to be idle, it means that the port is functioning and a service is running on the port.
* -PY (port list): This switch generates an SCTP packet containing a part of INIT data. This means that you are trying to establish a connection. If the destination port is closed, an ABORT packet is sent back; otherwise, the connection moves on to the next step of a four-way handshake by replying with an INIT-ACK. Once the INIT-ACK is received, the Nmap machine sends an INIT-ACK and marks the port as open instead of creating a connection.
* -PO (protocol list): This protocol list scan allows Nmap to configure the packet with a couple of protocols enabled in the packet header, such as ICMP and IGMP, to see whether there are any host unreachable responses to determine that the protocols are not supported by the destination port, thereby marking the port as closed.
* -PR (ARP Ping): ARP scan allows Nmap to send ARP requests to the remote host. If there is any response then Nmap marks the host as live without examining any other results. This also supports IPv6.
* --disable-arp-ping: This allows a user to obtain specific results when a network device or proxy responds to the ARP requests, creating a situation where all the hosts appear to be up.
* --traceroute: Traceroute is a post scan module that determines the best port to use to reach the remote host. This works by sending low TTL packets.
* -n: This allows users to skip the DNS resolution process. This can be slow, and thus the scan takes a lot of time.
* -R: This option is the counterpart to -n. It mandates that Nmap performs reverse DNS resolutions for all the live hosts.
* --system-dns: This can be used to specify that the DNS servers used for resolution should be the DNS servers that are configured on the hosts.
* --dns-servers <server1>[,<server2>[,...]]: This option can be used to define specific DNS addresses to be used for reverse DNS resolution.

**How to identify open ports**

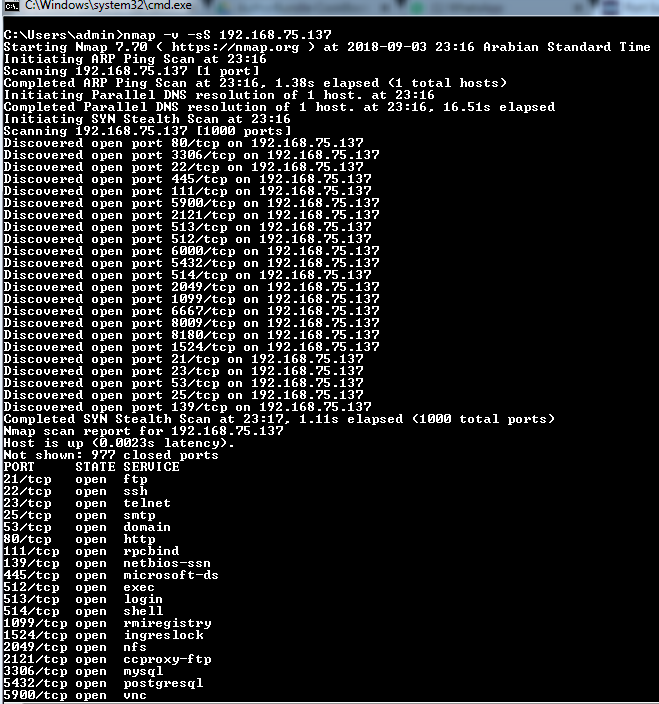
**The following are the six port states that are present in Nmap:**

* **open: This means that the port is functioning and has a service running or accessing it. The service can thus accept any connections made as per the protocol and service in use on this port.**
* **closed: A closed port is not being accessed by any service, there is no service running on it. Thus, no connections made externally will be successful on these ports.**
* **filtered: This status is associated with ports from which no response was received due to the packet filtering mechanism present within the network.  This might be caused by an intermediate network protection device.**
* **unfiltered: This status is associated with the ports that Nmap was not able to determine whether they were open or closed. Mostly ACK scan labels ports to be in unfiltered state; moreover, scans such as SYN and FIN can help resolve such issues.**
* **Open|filtered: Nmap classifies ports with this type when no response is received from them. The UDP, IP protocol, FIN, NULL, and Xmas scans associate this status with the ports.**
* **closed|filtered: This status is associated with ports that Nmap was not able to determine whether they were open or closed. Only idle scans use this status. Nmap provides various scan options for the user to craft a packet to obtain the desired result for Nmap to classify whether the port is open or closed. Most of these scan types are only allowed for administrative users because they have access to creating and sending raw packets.**
* **-sS (TCP SYN Scan): This is also called a half-open scan because TCP requires a three-way handshake to be completed before a connection is established. The Nmap machine generates a TCP SYN packet to which the remote port responds with TCP ACK, and then instead of sending a SYN/ACK packet, Nmap sends an RST flag to destroy the handshake, thereby preventing a connection. The port is considered if the Nmap SYN packet receives an ACK or SYN packet as a response.**
* **-sT (TCP connect scan): If a user does not have the required privileges to send a raw packet, or when a SYN scan is not an option, a TCP connect scan is used. As the name suggests, Nmap performs a complete three-way handshake and creates a connection to consider a port to be open.**
* **-sU (UDP scans): UDP scans send a packet to well-known ports, such as 53 and61, and it can then be performed on all ports. It sends protocol-specific packets to the famous ports and a generic UDP packet to the remaining ports. If the ports scanned return an ICMP unreachable error, then the port is closed. But if there is no response from a port it is marked as open filtered. In order to find out whether the port is actually running a service and is open, we can run a service detection scan.**
* **-sY (SCTP INIT scan): The SCTP INIT scan has already been discussed in the *How to perform host discovery* section. In order to perform this scan, there should be a running SCTP module.**
* **-sN; -sF; -sX (TCP NULL, FIN, and Xmas scans): In order to perform a deeper probe, Nmap provides an option to craft packets with different flags, such as FIN, PSH, and URG. If no flags are set, then it is called a Null scan. If FIN flags are set, then it is called a FIN scan, and if all three flags are set, then it is called an Xmas scan.**
* **-sA (TCP ACK scan): The TCP ACK scan has already been discussed in the *How to perform host discovery* section.**
* **-sW (TCP Window scan): The TCP Window scan works by the value of the TCP Window field of the RST packets received. Most systems have a window of zero for the RST packet of closed ports and a positive value for the open ports. This lists the port as closed instead of unfiltered once the RST packet is received.**
* **--scanflags (Custom TCP scan): The Custom TCP scan allows a user to set various flags in the TCP packet, such as URG, SYN, ACK, FIN, PSH, URG, and RST, thereby allowing the user to create a custom packet for the probe.**
* **-sO (IP protocol scan): This scan allows you to define the protocol for which the scan is being performed, such as TCP, UDP, ICMP, and IGMP, thus a specific packet is created for the probe.**
* **-b <FTP relay host> (FTP bounce scan): This allows the user to connect to one FTP host and then relay the files to another FTP host, which is mentioned in the argument.**

**How do it…**

**These are the steps:**

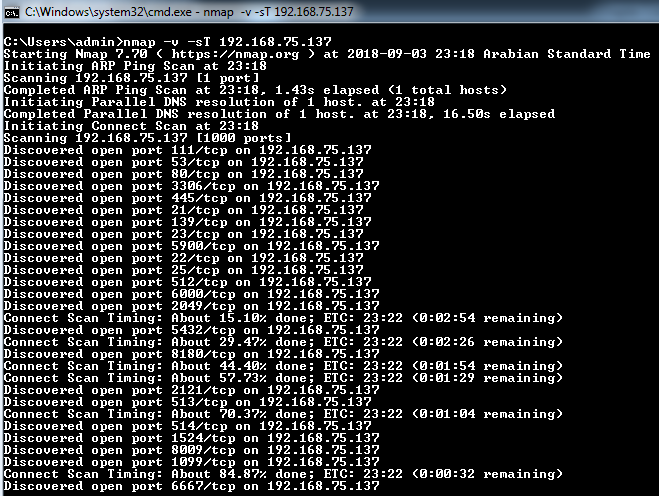
1. **Open nmap in Command Prompt.**
2. **Run the following syntax in the Command Prompt to perform a TCP SYN scan:**

****

**nmap –v –sS 192.168.75.137**

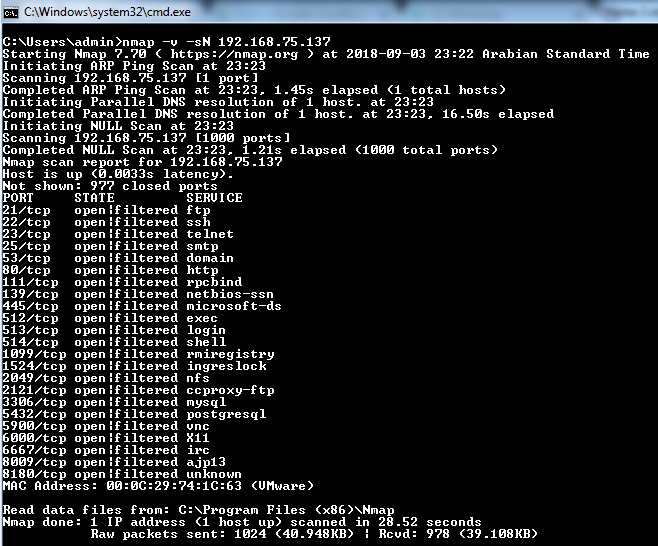
1. **Run the following syntax in the Command Prompt to perform a TCP Connect scan:**

**nmap –v –sT 192.168.75.137**

****

1. **Run the following syntax in the Command Prompt to perform a TCP NULL scan:**

**nmap –v –sN 192.168.75.137**

****

**How to manage specification and scan order**

**Nmap provides various options to specify ports to be scanned in a random or sequential order. All the Nmap scans, without any ports specified or any specific NSE script provided as an argument, by default scan only the top 1,000 ports:**

* **-p <port ranges>: This option can be used to configure the ports to be scanned in multiple formats. It can be a range or a list. General representation of the syntax would be –p1-65535if you want to perform a full port scan or–p1,2,3, or4 as a random list that can be non-serial in nature.**
* **--exclude-ports <port ranges>: It is a tedious task to prepare a list of ports to be scanned when the requirement is a full port with a few exclusions. In such cases, you can use the exclude ports flag to exclude the ports that are not to be scanned.**
* **-F (Fast (limited port) scan): The fast scan further reduces the default number of ports scanned from 1,000 to 100. This will reduce the scan time immensely and thus provide quicker results, as the name suggests.**
* **-r (Don't randomize ports): By default, Nmap randomizes the port order for the scan. This option allows the user to instruct Nmap to follow a strict order for the ports to be scanned.**
* **--port-ratio <ratio>: This scans all ports in the Nmap-services file with a ratio greater than the one given. <ratio>must be between0.0and1.0.**
* **--top-ports <n>: This scans the <n>highest-ratio ports found in the Nmap-services file after excluding all ports specified by --exclude-ports.<n>must be1or greater.**

**How do it…**

**Here are the steps:**

1. **Open nmap in Command Prompt.**
2. **Run the following syntax in the Command Prompt to perform a scan between ports 0-100:**

**nmap 192.168.75.137 –p0-100**