Discovery Scans

-PR	Send an ARP (address resolution protocol) request to a target for a response ARPs are not usually blocked by firewalls Default discovery method for any nmap scan on an ethernet network
-sn	No port scan; discovery only; use combination of ICMP, ECHO, REQUEST, TCP SYN to port 443; TCP ACK to port 80; and an ICMP timestamp request
-PS<portlist>	Discover hosts by sending a TCP SYN to specified port/s; Default is port 80; Any response (SYN, ACK, RST) demonstrates the target is up Syntax indicates no space between **–PS** and the port list Will be followed by a port scan unless the **–sn** option is used

Nmap Scan Types

Option	*Example*	*Description*
-h	**nmap -h**	Help on Nmap
-V	**nmap -V**	Nmap version
-d	**nmap -d 192.168.1.50**	Enable debugging to view all steps of output
-sT	**nmap -sT 192.168.1.50**	Complete a TCP 3-way handshake for non-root users
-sV	**nmap -sV 192.168.1.50**	Probe open ports for service version

-sS	**nmap -sS 192.168.1.50**	Send TCP SYN to target for response to check Check for TCP 3-way handshake ▪ If port is open, will respond with SYN ACK ▪ RST if port is closed
-sU	**nmap -sU 192.168.1.50**	Do a UDP scan ▪ Ports that respond are open ▪ Ports that do not respond are displayed as open|filtered (unknown) ▪ Ports that send an ICMP unreachable error (type 3 code 3) are closed
-sL	**nmap -sL 192.168.1.50**	List the targets that will be scanned
-sA	**nmap -sA www.example.com**	Find out if a host/network is protected by a firewall. ▪ "Filtered" results indicate firewall is on. ▪ "Unfiltered" results indicate port is accessible, but might be open or closed. ▪ Run with -A option to determine if accessible ports are actually open or closed (nmap -sA -A www.example.com)

Stealth Scans

-sS	**nmap -sS 192.168.1.50**	The original "stealth" scan; send a TCP SYN and if the target responds with a SYN ACK, do not complete the handshake, but instead send a RST This is less likely to be logged by the target
-sA	**nmap -sA 192.168.1.0/24**	Send a TCP ACK; used to map out firewall rule sets, determine which ports are filtered, and if a firewall is stateful or not
-sN	**nmap -sN 192.168.1.2-10**	Send a TCP segment with no flags raised; this is not the normal state for TCP, which always has at least one flag (usually ACK) raised Used to sneak through a non-stateful firewall
-sF	**nmap -sF www.example.com**	Send a TCP FIN; used to sneak through a non-stateful firewall
-sX	**nmap -sX 192.168.1.0/24**	Send a TCP segment with FIN, PSH, and URG flags raised, thus lighting up the packet; This is an illogical combination and is used to quietly get through a non-stateful firewall

Stealth Scans - *pt2*

-Pn	**nmap -Pn -p- 192.168.1.0/24**	Skip discovery; assume all hosts are online for port scan Useful if targets have their firewall up and only offer services on unusual ports
-sI <zombie> <target>	**nmap -sI -Pn -p- zombie.middle.tld www.example.com**	Conduct a blind TCP port scan (idle scan); no packets are sent directly from your attacker machine to the target Uses a "zombie" (middle man) host to obtain information about open ports on the target; After locating a machine that can be used as a zombie, it can be reused for more scans
-b <FTP relay> <FTP target>	▪ **nmap -v -b name:password @old-ftp-server.example.com** ▪ **ftp-target-server.example.com -Pn**	Conduct an FTP bounce scan; exploit FTP proxy connections in which a user asks a "middle man" FTP server to send files to another FTP server Because of widespread abuse, the FTP relay feature has been disabled by most vendors

Stealth Scans - *pt 3*

-T <0-5>	**nmap 192.168.1.0/24 -T 2**	Use different timing templates to throttle the speed of your queries to make the scan less noticeable; T0 is the slowest, and T5 is the fastest Nmap denotes these speeds as paranoid, sneaky, polite, normal, aggressive, and insane, respectively; T4 is the recommended choice for a fast scan that is still stable. T3 is the default.
-f	**nmap -f 192.168.1.50**	Split packets (include pings) into 8-byte fragments to make it more difficult for packet filtering firewalls and intrusion detection to detect the purpose of packets MTU is the maximum fragment size
-D [decoy1, decoy2, decoy3, etc.] <target>	**nmap -D 192.168.1.10 192.168.1.15 192.168.1.30 192.138.1.50**	Used to mask a port scan by using decoys; creates bogus packets from the decoys so the actual attacker blends in with the crowd; It appears that both the decoys and the actual attackers are performing attacks
-e <interface>	**nmap -e eth0 192.168.1.50**	Specify the interface Nmap should use
-S <spoofed source address>	**nmap -e eth0 -S www.google.com 192.168.1.50**	Spoof the source address; will not return useful reports to you, but can be used to confuse an IDS or the target administrator

Stealth Scans - *pt 4*

--spoof-mac [vendor type | MAC address]	**nmap -sT -PN --spoof-mac apple 192.168.1.50** **nmap -sT -PN --spoof-mac B7:B1:F9:BC:D4:56 192.168.1.50**	Use a bogus source hardware address; you can specify a random MAC based on vendor, or explicitly specify the MAC address
--source-port <port number>	**nmap --source-port 53 192.168.1.36**	Use a specific source port number (spoof source port) to dupe packet filters configured to trust that port; same as -g <port number> option
--source-port <port number>	**nmap --source-port 53 192.168.1.36**	Use a specific source port number (spoof source port) to dupe packet filters configured to trust that port; same as -g <port number> option
--randomize-hosts	**nmap --randomize-hosts 192.168.1.1-100**	Randomize the order of the hosts being scanned.
--proxies <proxy:port, proxy:port…>	**nmap --proxies http://192.168.1.30:8080, http://192.168.1.90:8008**	Relay TCP connections through a chain of HTTP or SOCKS4 proxies; especially useful on the Internet.

Nmap Options

-p <port range>	**nmap -p 80 192.168.1.50** **nmap -p 80,443 www.example.com** **nmap -p1024-3000 192.168.1.0/24** **nmap -p U:53,111,137,T:21-25,80, 139,443 192.168.1.0/24** **nmap -p- 192.168.1.50**	Scan only specified port/s Port status can be OPEN, CLOSED (no service on port), or FILTERED (perhaps a firewall) UDP ports: U; TCP ports: T; ALL TCP ports: -p-
-r	**nmap --top-ports 200**	Scan top <indicated number> ports
--top-ports <number>	**nmap --top-ports 200**	Scan top <indicated number> ports
-6	**nmap -6 2001:f0d0:1003:51::4** **nmap -6 www.example.com** **nmap -6 fe80::8d50:86ce:55ad:bc 5c**	Scan IPv6 addresses
-iL <input file name>	**nmap -iL /tmp/test.txt**	Scan hosts listed in file
--exclude	**map 192.168.1.0/24 --exclude 192.168.1.5**	Exclude certain hosts from scan
-n	**nmap -n 192.168.1.0/24**	Do not resolve names (time saver)
-R	**nmap -R 192.168.1.0/24**	Try to resolve all names with reserved DNS
-F (fast mode)	**nmap -F 192.168.1.50**	Scan fewer ports than default

Nmap Options - *pt 2*

-iL <input file name>	**nmap -iL /tmp/test.txt**	Scan hosts listed in file
--exclude	**map 192.168.1.0/24 --exclude 192.168.1.5**	Exclude certain hosts from scan
-n	**nmap -n 192.168.1.0/24**	Do not resolve names (time saver)
-R	**nmap -R 192.168.1.0/24**	Try to resolve all names with reserved DNS
-F	**nmap -F 192.168.1.50**	Scan fewer ports than default
-O	**nmap -O 192.168.1.50**	Enable OS detection, not always accurate
-A	**nmap -A 192.168.1.50**	Enable OS detection, service version detection, script scanning, and traceroute
--version-intensity <level>	**nmap -sV --version-intensity 9 192.168.1.50**	Use with –sV Specified level of interrogation from 0 (light) to 9 (attempt all probes)
--script=<script name>	**nmap --script=banner.nse 192.168.1.50**	Use NSE script
-sC	**nmap -sC 192.168.1.50**	Scan using all default scripts
-v **-vv** **-v1** **-v-1**	**nmap -v 192.168.1.50** **nmap -v-1 192.168.1.50**	Increase verbosity of output The more ‘v’s the more verbose Alternatively you can specify the exact level number after the **-v** command There are 9 levels [-4 : 4]
-oN/-oX/-oS/-oG /-oA <filename>	**nmap 192.168.1.50 -oA results.txt**	Save output in normal, XML, script kiddie, Grepable, or all