

## Discovery Scans

<b>-PR</b>	<p>Send an ARP (address resolution protocol) request to a target for a response</p> <p>ARPs are not usually blocked by firewalls</p> <p>Default discovery method for any nmap scan on an ethernet network</p>
<b>-sn</b>	<p>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</p>
<b>-PS&lt;portlist&gt;</b>	<p>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</p> <p>Syntax indicates no space between <b>-PS</b> and the port list</p> <p>Will be followed by a port scan unless the <b>-sn</b> option is used</p>

## Nmap Scan Types

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

<b>-sS</b>	<b>nmap -sS 192.168.1.50</b>	Send TCP SYN to target for response to check Check for TCP 3-way handshake <ul style="list-style-type: none"><li>■ If port is open, will respond with SYN ACK</li><li>■ RST if port is closed</li></ul>
<b>-sU</b>	<b>nmap -sU 192.168.1.50</b>	Do a UDP scan <ul style="list-style-type: none"><li>■ Ports that respond are open</li><li>■ Ports that do not respond are displayed as open   filtered (unknown)</li><li>■ Ports that send an ICMP unreachable error (type 3 code 3) are closed</li></ul>
<b>-sL</b>	<b>nmap -sL 192.168.1.50</b>	List the targets that will be scanned
<b>-sA</b>	<b>nmap -sA www.example.com</b>	Find out if a host/network is protected by a firewall. <ul style="list-style-type: none"><li>■ "Filtered" results indicate firewall is on.</li><li>■ "Unfiltered" results indicate port is accessible, but might be open or closed.</li><li>■ Run with -A option to determine if accessible ports are actually open or closed (nmap -sA -A www.example.com)</li></ul>

## Stealth Scans

<b>-sS</b>	<b>nmap -sS 192.168.1.50</b>	<p>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</p> <p>This is less likely to be logged by the target</p>
<b>-sA</b>	<b>nmap -sA 192.168.1.0/24</b>	Send a TCP ACK; used to map out firewall rule sets, determine which ports are filtered, and if a firewall is stateful or not
<b>-sN</b>	<b>nmap -sN 192.168.1.2-10</b>	<p>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</p> <p>Used to sneak through a non-stateful firewall</p>
<b>-sF</b>	<b>nmap -sF www.example.com</b>	Send a TCP FIN; used to sneak through a non-stateful firewall
<b>-sX</b>	<b>nmap -sX 192.168.1.0/24</b>	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	<p>Skip discovery; assume all hosts are online for port scan</p> <p>Useful if targets have their firewall up and only offer services on unusual ports</p>
-sl <zombie> <target>	nmap -sl -Pn -p-zombie.middle.tld www.example.com	<p>Conduct a blind TCP port scan (idle scan); no packets are sent directly from your attacker machine to the target</p> <p>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</p>
-b <FTP relay> <FTP target>	<ul style="list-style-type: none"> <li>■ nmap -v -b name:password@old-ftp-server.example.com</li> <li>■ ftp-target-server.example.com -Pn</li> </ul>	<p>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</p> <p>Because of widespread abuse, the FTP relay feature has been disabled by most vendors</p>

## Stealth Scans - *pt 3*

<b>-T &lt;0-5&gt;</b>	<b>nmap 192.168.1.0/24 -T 2</b>	<p>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</p> <p>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.</p>
<b>-f</b>	<b>nmap -f 192.168.1.50</b>	<p>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</p> <p>MTU is the maximum fragment size</p>
<b>-D [decoy1, decoy2, decoy3, etc.] &lt;target&gt;</b>	<b>nmap -D 192.168.1.10 192.168.1.15 192.168.1.30 192.138.1.50</b>	<p>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</p>
<b>-e &lt;interface&gt;</b>	<b>nmap -e eth0 192.168.1.50</b>	Specify the interface Nmap should use
<b>-S &lt;spoofed source address&gt;</b>	<b>nmap -e eth0 -S www.google.com 192.168.1.50</b>	<p>Spoof the source address; will not return useful reports to you, but can be used to confuse an IDS or the target administrator</p>

## Stealth Scans - *pt 4*

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

## Nmap Options

<b>-p &lt;port range&gt;</b>	<pre>nmap -p 80 192.168.1.50</pre> <pre>nmap -p 80,443 www.example.com</pre> <pre>nmap -p1024-3000 192.168.1.0/24</pre> <pre>nmap -p U:53,111,137,T:21-25,80,139,443 192.168.1.0/24</pre> <pre>nmap -p- 192.168.1.50</pre>	<p>Scan only specified port/s</p> <p>Port status can be OPEN, CLOSED (no service on port), or FILTERED (perhaps a firewall)</p> <p>UDP ports: U; TCP ports: T; ALL TCP ports: -p-</p>
<b>-r</b>	<pre>nmap --top-ports 200</pre>	Scan top <indicated number> ports
<b>--top-ports &lt;number&gt;</b>	<pre>nmap --top-ports 200</pre>	Scan top <indicated number> ports
<b>-6</b>	<pre>nmap -6 2001:f0d0:1003:51::4</pre> <pre>nmap -6 www.example.com</pre> <pre>nmap -6 fe80::8d50:86ce:55ad:bc5c</pre>	Scan IPv6 addresses
<b>-iL &lt;input file name&gt;</b>	<pre>nmap -iL /tmp/test.txt</pre>	Scan hosts listed in file
<b>--exclude</b>	<pre>nmap 192.168.1.0/24 --exclude 192.168.1.5</pre>	Exclude certain hosts from scan
<b>-n</b>	<pre>nmap -n 192.168.1.0/24</pre>	Do not resolve names (time saver)
<b>-R</b>	<pre>nmap -R 192.168.1.0/24</pre>	Try to resolve all names with reserved DNS
<b>-F (fast mode)</b>	<pre>nmap -F 192.168.1.50</pre>	Scan fewer ports than default

## Nmap Options - *pt 2*

<b>-iL &lt;input file name&gt;</b>	<b>nmap -iL /tmp/test.txt</b>	Scan hosts listed in file
<b>--exclude</b>	<b>map 192.168.1.0/24 --exclude 192.168.1.5</b>	Exclude certain hosts from scan
<b>-n</b>	<b>nmap -n 192.168.1.0/24</b>	Do not resolve names (time saver)
<b>-R</b>	<b>nmap -R 192.168.1.0/24</b>	Try to resolve all names with reserved DNS
<b>-F</b>	<b>nmap -F 192.168.1.50</b>	Scan fewer ports than default
<b>-O</b>	<b>nmap -O 192.168.1.50</b>	Enable OS detection, not always accurate
<b>-A</b>	<b>nmap -A 192.168.1.50</b>	Enable OS detection, service version detection, script scanning, and traceroute
<b>--version-intensity &lt;level&gt;</b>	<b>nmap -sV --version-intensity 9 192.168.1.50</b>	Use with -sV Specified level of interrogation from 0 (light) to 9 (attempt all probes)
<b>--script=&lt;script name&gt;</b>	<b>nmap --script=banner.nse 192.168.1.50</b>	Use NSE script
<b>-sC</b>	<b>nmap -sC 192.168.1.50</b>	Scan using all default scripts
<b>-v -v&lt;-4 : 4&gt;</b>	<b>nmap -vv 192.168.1.50  nmap -v-1 192.168.1.50</b>	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]
<b>-oN/-oX/-oS/-oG /-oA &lt;filename&gt;</b>	<b>nmap 192.168.1.50 -oA results.txt</b>	Save output in normal, XML, script kiddie, Grepable, or all



