Nmap Options

Nmap's versatile command-line options are essential for tailoring network discovery and security audits. They allow users to gather specific information and gain deeper insights into network infrastructures.

by Sean Sanders

Specifying Scan Targets in Nmap

Nmap offers diverse methods to define your scan targets, allowing for precise control over which systems are probed.

Direct Input

Specify targets by hostname, IP address, or network range, such as `scanme.nmap.org` or `192.168.1.0/24`.

Input List (-iL)

Load targets from a file containing a list of hostnames, IPs, or CIDR ranges, ideal for large target sets.

Random Targets (-iR)

Scan a specified number of random hosts on the internet, useful for general reconnaissance or research.

Exclude Hosts (--exclude)

Prevent specific hosts or networks from being scanned by explicitly listing them with this option.

Exclude File (--excludefile)

Reference a file containing a list of targets that Nmap should completely avoid during the scan operation.

Nmap Host Discovery Techniques

Host discovery is the initial phase of any Nmap scan, determining which hosts on a network are active and responsive. Nmap provides various options to tailor this process.

Scan Behavior Basics

-sL: List Scan – Simply lists targets without sending packets. Useful for verifying target ranges.

-sn: Ping Scan – Disables port scanning, only performing host discovery. Quickly identifies online hosts.

-Pn: No Ping – Treats all specified hosts as online, skipping discovery. Essential for targets behind firewalls.

Advanced Probing Methods

-PS/PA/PU/PY[portlist]: Uses TCP SYN, TCP ACK, UDP, or SCTP probes to specific ports for discovery.

-PE/PP/PM: Leverages ICMP echo, timestamp, and netmask requests to identify live hosts.

-PO[protocol list]: Sends IP Protocol Pings to determine host availability based on protocol responses.

DNS Resolution Control

-n: No DNS Resolution – Prevents Nmap from performing reverse DNS lookups, speeding up scans.

-R: Always Resolve – Forces Nmap to always perform reverse DNS resolution on target IPs.

--dns-servers: Specify custom DNS servers to use instead of the system's defaults.

--system-dns: Explicitly instructs Nmap to use the operating system's configured DNS resolver.

Path Tracing

--traceroute: Maps the network path (hops) to each host using Time-to-Live (TTL) expiration, similar to the standard traceroute command.

Nmap Port Specification & Scan Order

Precisely control which ports Nmap scans and in what order, optimizing efficiency and targeting specific services.

-p: Specify Ports

Define exact ports or ranges to scan. Supports TCP (T), UDP (U), and SCTP (S). Examples: `-p22`, `-p U:53,T:80-90`.

--exclude-ports: Exclude Ports

Skip specific ports or ranges to avoid unwanted interactions or focus on relevant services. E.g., `--exclude-ports 135,445`.

-F: Fast Scan Mode

Accelerate scans by targeting a reduced list of commonly open ports, ideal for quick assessments.

-r: Sequential Scan

Disable Nmap's default port randomization, scanning ports in numerical order for predictable results or specific testing scenarios.

--top-ports: Most Common Ports

Scan the `<number>` most frequently open ports from Nmap's database, useful for rapid service identification.

--port-ratio: Popularity Threshold

Scan ports more common than a specified ratio (0.0 to 1.0), allowing fine-grained control over port selection based on popularity.

Nmap Scripting Engine (NSE) Scan Options

The Nmap Scripting Engine (NSE) extends Nmap's capabilities, allowing users to automate a wide range of networking tasks, from vulnerability detection to advanced discovery. These options fine-tune how NSE scripts are executed.

-sC: Default Script Scan

Activates Nmap's default set of safe and efficient scripts, equivalent to `--script=default`.

--script=<Lua scripts>

Specifies individual scripts, categories, or directories of scripts to execute, using comma-separated values.

--script-args

Provides arguments to NSE scripts using a `key=value` format, allowing customization of script behavior.

--script-args-file

Loads script arguments from a specified file, useful for complex or numerous parameters.

--script-trace

Shows all incoming and outgoing data sent and received by NSE scripts, aiding in debugging.

--script-updatedb

Refreshes the Nmap Scripting Engine database, ensuring it has the latest available scripts.

--script-help

Displays detailed help information for specified scripts or script categories.

Dissecting Nmap Scan Techniques

Nmap's service and version detection capabilities allow you to precisely identify what's running on open ports, beyond just knowing they are open.

-sV: Service/Version Info

Initiate a robust service and version detection scan. This option probes open ports to determine the exact service name, application name, version number, and sometimes even the operating system of the running service.

--version-intensity <level>

Control the aggressiveness of the version scan. A higher intensity level (up to 9) means Nmap will try more probes, increasing accuracy but also scan time. A lower intensity (0) performs a quicker, less comprehensive check.

--version-light

A shortcut for `--version-intensity 2`. This option balances speed and accuracy by only trying the most common and likely probes, ideal for quick assessments without being overly verbose.

--version-all

Equivalent to `--version-intensity 9`. This ensures Nmap attempts every available probe against each open port, maximizing the chances of accurate service and version identification, albeit at a slower pace.

--version-trace

Display detailed version scan activity. This is primarily for debugging purposes, showing you exactly which probes Nmap is sending and the responses it receives, aiding in understanding the detection process.

Dissecting Nmap Scan Techniques

Nmap employs a wide array of scanning techniques, each designed for a specific purpose, from stealthy port identification to detailed protocol analysis.

Common TCP Scans

- **-sS (SYN scan):** The default, fast, and stealthy half-open scan that doesn't complete the TCP handshake.
- **-sT (Connect() scan):** A full TCP handshake scan, less stealthy but reliable when SYN scans are blocked.
- **-sA (ACK scan):** Used to map firewall rules, determining if ports are filtered or unfiltered by analyzing ACK responses.
- **-sW (Window scan):** Examines TCP window sizes to potentially distinguish open ports from closed ones.
- **-sM (Maimon scan):** Differentiates between open and closed ports based on RST packet responses, similar to FIN/Xmas scans.
- **--scanflags <flags>:** Customize TCP scan flags (e.g., SYN, ACK, FIN, RST) for advanced analysis or evasion.

UDP and Stealthy TCP Scans

- **-sU (UDP scan):** Probes UDP ports to identify open, closed, or filtered states, often by sending empty UDP packets.
- **-sN (Null scan):** Sends TCP packets with no flags set, exploiting RFC 793 behavior to detect open ports stealthily.
- **-sF (FIN scan):** Sends TCP packets with only the FIN flag set; open ports should send no response, closed ports respond with RST.
- **-sX (Xmas scan):** Sends TCP packets with FIN, URG, and PSH flags set (like a "Christmas tree" packet). Behaves similarly to Null/FIN scans for port state detection.

Advanced Protocol Scans

- **-sI <zombie host[:probeport]> (Idle scan):** A highly stealthy scan that uses a "zombie" host to bounce scan packets, hiding the attacker's IP.
- **-sY (SCTP INIT scan):** Scans Stream Control Transmission Protocol (SCTP) ports for open services.
- **-sZ (SCTP COOKIE-ECHO scan):** Another SCTP scanning method, checking for SCTP COOKIE-ECHO chunks.
- **-sO (IP protocol scan):** Determines which IP protocols (e.g., TCP, UDP, ICMP) are supported by target hosts.
- **-b <FTP relay host> (FTP bounce scan):** Uses a vulnerable FTP server as a proxy to conduct scans, often bypassing firewalls.

Nmap OS Detection Options

Nmap's Operating System (OS) detection capabilities identify the underlying OS of target hosts, providing crucial intelligence for penetration testing and network security. These options fine-tune the OS fingerprinting process.

-O: Enable OS Detection

Activates Nmap's OS fingerprinting engine to identify the operating system, its version, and device type of a target host.

--osscan-limit: Limit OS Detection

Instructs Nmap to only attempt OS detection against hosts that have at least one open TCP port, making the process more efficient for large scans.

--osscan-guess: Aggressive OS Guessing

Enables Nmap to guess the OS more aggressively when a perfect match isn't found, increasing the likelihood of identifying the OS, albeit with potentially lower confidence.

Nmap Timing and Performance Options

Nmap provides a granular set of controls to fine-tune scan timing and performance, allowing users to balance speed, stealth, and accuracy based on network conditions and objectives. These options are crucial for efficient and effective network exploration.

-T<0-5>: Timing Template

Set a predefined timing template (0-5), where 5 is aggressive and 0 is paranoid, dictating the overall speed and stealth of the scan.

--min/max-hostgroup

Control the number of hosts scanned in parallel within a host group, optimizing for network load or scan speed.

--min/max-parallelism

Define the number of probes Nmap can send in parallel to a single host, improving efficiency for responsive targets.

--min/max/initial-rtt-timeout

Adjust the probe round trip time (RTT) values to adapt to network latency, preventing premature timeouts or excessive delays.

--max-retries

Limit the number of times Nmap retransmits port scan probes, preventing scans from lingering on unresponsive hosts.

--host-timeout

Specify a maximum duration for Nmap to spend scanning a single target host before giving up on it.

--scan-delay/--max-scan-delay

Introduce a delay between probes, useful for avoiding IDS/IPS detection or reducing network congestion.

--min/max-rate

Set minimum and maximum packet sending rates per second to control scan intensity and manage network bandwidth usage.

Nmap Evasion and Spoofing Techniques

Nmap offers powerful options to bypass firewalls and Intrusion Detection Systems (IDS) by manipulating scan traffic, or by disguising the scanner's identity, allowing for stealthier and more advanced network reconnaissance.

Packet Fragmentation & Customization

- **-f; --mtu <val>:** Fragments scan packets into smaller pieces to evade detection by simple packet filters.
- **--data-length <num>:** Appends random data to packets, making them appear less suspicious.
- **--ip-options <options>:** Allows specifying custom IP header options for advanced evasion.
- **--badsum:** Sends packets with invalid TCP/UDP/SCTP checksums, testing how firewalls handle malformed packets.

Identity Masking & Redirection

- **-D <decoy1,decoy2[,ME],...>:** Mixes the scanner's real IP with fake "decoy" IP addresses, making it harder to trace.
- **-S <IP_Address>:** Spoofs the source IP address, potentially to impersonate another host.
- **--spoof-mac <mac address>:** Changes the MAC address used for scanning, hiding the physical device identity.
- **-e <iface>:** Specifies the network interface to use for sending and receiving packets.

Traffic Relay & Port Control

- **-g/--source-port <portnum>:** Uses a specific source port number for scan packets, useful for bypassing port-based filters.
- **--proxies <url1,[url2],...>:** Routes scan traffic through HTTP or SOCKS4 proxies, masking the source.
- **--ttl <val>:** Sets the IP Time-To-Live field to a specific value, which can help in evading some hop-count based detection.

Nmap Output and Miscellaneous Options

Nmap provides extensive options for controlling how scan results are presented, their verbosity, and advanced configurations for specific use cases. These settings help users tailor Nmap's behavior to their needs, from stealthy reconnaissance to detailed reporting.

Output Formatting & Files

- **-oN/-oX/-oS/-oG <file>:** Direct scan output to a specified file in Normal, XML, Script Kiddie, or Grepable format.
- **-oA <basename>:** Simultaneously save scan results in Normal, XML, and Grepable formats using a common base filename.
- **--append-output:** Append scan results to existing output files rather than overwriting them.
- **--resume <filename>:** Continue an interrupted scan from where it left off, using a previously saved output file.

Verbosity & Scan Insights

- **-v:** Increase the verbosity level, providing more detailed information about the scan's progress and findings.
- **-d:** Increase the debugging level, useful for troubleshooting Nmap's internal operations.
- **--reason:** Display the underlying reason Nmap determined a port to be in a particular state (e.g., "syn-ack" for open).
- **--open:** Filter output to only show ports that Nmap identified as open or potentially open.
- **--packet-trace:** Show all packets sent and received by Nmap during the scan, invaluable for deep analysis.
- **--iflist:** Print a list of the host's network interfaces and routes, useful for network configuration debugging.

XML Output Customization

--stylesheet <path/URL>: Specify a custom XSL stylesheet to transform XML output into an easily readable format, like HTML.

Web XML Reference

--webxml: Automatically references an XSL stylesheet from Nmap.Org for more portable and web-friendly XML output.

Disable Stylesheet

--no-stylesheet: Prevent Nmap from associating any XSL stylesheet with the XML output, resulting in raw XML.

Non-Interactive Mode

--noninteractive: Disable any runtime interactions via the keyboard, ensuring Nmap runs autonomously without requiring user input.

Advanced Nmap Options Explained

Beyond the core scanning techniques, Nmap offers a suite of advanced and miscellaneous options that provide fine-grained control over scan behavior, data handling, and environmental assumptions. These options cater to specialized scenarios, from targeting IPv6 networks to managing Nmap's operational privileges.

-6: IPv6 Scanning

Enable Nmap to scan IPv6 addresses, expanding its reach to modern network infrastructures that utilize the latest Internet Protocol.

-A: Aggressive Scan

Activate an aggressive scan mode that encompasses OS detection, version detection, script scanning, and traceroute for a comprehensive target profile.

--datadir <dirname>

Specify a custom location for Nmap's data files (like service probes and OS signatures), allowing for tailored configurations.

--send-eth/--send-ip

Control whether Nmap sends packets using raw Ethernet frames or IP packets, impacting how the scan interacts with the network stack.

--privileged

Assume the user has full root/administrator privileges, enabling Nmap to perform advanced operations like raw socket access.

--unprivileged

Instruct Nmap to operate without raw socket privileges, altering its behavior to use higher-level system calls for port scanning.

-V: Print Version

Display the Nmap version number, useful for verifying the installed release and checking for updates.

-h: Help Summary

Print a concise summary of Nmap's command-line options and usage, a quick reference for all available features.

Nmap Examples

Now that we've explored the various options and commands Nmap has to offer, let's dive into practical examples. This section will demonstrate how to apply Nmap's powerful features to real-world scenarios, providing clear illustrations of its versatility in network discovery, security auditing, and system reconnaissance.

Dissecting an Nmap Scan Command

Nmap is a critical tool for network exploration. Let's analyze a common command to scan all ports and process the output efficiently.

```
ports=$(nmap -p- --min-rate=1000 -Pn -T4 10.10.10.245 | grep '^[0-9]' | cut -d '/' -f 1 | tr '\n' ',' | sed s/,$//)
```

Nmap Command Breakdown

`-p-`

Scans all 65535 TCP/UDP ports, ensuring comprehensive coverage of potential services.

`--min-rate=1000`

Speeds up the scan by maintaining a minimum sending rate of 1000 packets per second.

`-Pn`

Skips host discovery, treating all targets as online to bypass potential firewall blocks.

`-T4`

Applies an "aggressive" timing template, optimizing for faster execution at the risk of detection.

Output Processing Pipeline

- `grep '^'`: Filters for lines starting with a digit, identifying port numbers.
- `cut -d '/' -f 1`: Extracts the port number from other output details.
- `tr '\n' ','`: Converts newline-separated ports into a single, comma-separated string.
- `sed s/,$//`: Removes any trailing comma for clean formatting.

The entire expression assigns the resulting comma-separated port list to the `ports` variable.

Advanced Nmap Options Explained

```
nmap -p$ports -Pn -sC -sV 10.10.10.245
```

Building on our previous scan, let's dissect further Nmap options to refine our network exploration and service identification.

`-p$ports`

This option targets only specified ports. The `$ports` variable, from a prior scan, feeds a comma-separated list (e.g., "21,22,80") to Nmap, ensuring focused scanning.

`-Pn`

Skips host discovery, treating the target as online. This is crucial for bypassing firewall blocks that might prevent Nmap from sending initial ping probes, ensuring the scan proceeds.

`-sC`

Activates Nmap's default script scan (`--script=default`). It deploys NSE scripts for advanced service detection, vulnerability identification, and detailed enumeration on open ports.

`-sV`

Enables service/version detection. Nmap actively probes open ports to accurately identify the specific application and its version running behind each service, offering deeper insights.

`10.10.10.245`

This is the target specification, pinpointing the exact IP address of the host that Nmap will scan. It's the destination for all the defined scan options.