CHEAT SHEETS HERO

Nmap Cheat Sheet

A comprehensive cheat sheet for Nmap, covering essential scanning techniques, options, and usage examples for network discovery and security auditing.

Basic Scan Types

Scan Types Overview

Nmap offers a variety of scan types to discover hosts and services on a network. These techniques use different TCP, UDP, and ICMP protocols to gather information.

Common Scan Flags

`nmap -sT <target>`	TCP Connect Scan: Establishes a full TCP connection (three-way handshake) to detect open ports. Requires no special privileges.
`nmap -sS <target>`	TCP SYN Scan (Stealth Scan): Sends SYN packets to the target. If a SYN-ACK is received, the port is open. If a RST is received, the port is closed. Requires root privileges.
`nmap -sU <target>`	UDP Scan: Sends UDP packets to the target. Requires root privileges and can be slow but detects open UDP ports.
`nmap -sP <target>`	Ping Scan: Discovers active hosts on a network by sending ICMP echo requests. Deprecated, use `-sn` instead.
`nmap -sn <target>`	Host Discovery: Discovers active hosts on a network, similar to ping scan, but more reliable.
`nmap -sV <target>`	Version Detection: Determines the service and version running on open ports.

Example Usage

```
nmap -sS 192.168.1.100
```

Performs a SYN scan on the target IP address.

```
nmap -sU 192.168.1.100
```

Performs a UDP scan on the target IP address.

```
nmap -sV 192.168.1.100
```

Attempts to determine service versions on the target IP address.

Advanced Scanning Techniques

Stealth Scan Options

`nmap -sF <target>`	TCP FIN Scan: Sends a FIN packet. Open ports are expected to ignore the packet, while closed ports respond with an RST.
`nmap -sX <target>`	TCP Xmas Scan: Sends a packet with FIN, URG, and PSH flags set. Closed ports respond with an RST.
`nmap -sN <target>`	TCP Null Scan: Sends a packet with no flags set. Closed ports respond with an RST.

Bypassing Firewalls/IDS

`nmap -f <target>`	Fragment Packets: Helps bypass simple firewalls by fragmenting the packets.
`nmap --mtu <value> <target>`	Specify MTU: Sets a specific Maximum Transmission Unit (MTU) to avoid triggering certain IDS rules.
`nmap --data-length <number> <target>`	Append Random Data: Adds random data to the end of packets to avoid signature-based detection.
`nmap --spoof-mac <MAC address/prefix/vendor> <target>`	Spoof MAC Address: Spoofs the MAC address of your network interface to hide your identity.
`nmap -g <portnumber> <target>`	Source Port Manipulation: Use a specific port number

Timing and Performance

`nmap -T<0-5> <target>`	Timing Templates: Sets the timing template. 0 is the slowest (paranoid), 5 is the fastest (insane).
`nmap --min-rtt-timeout <time> --max-rtt-timeout <time> --initial-rtt-timeout <time> <target>`	Adjust RTT Timeout: Fine-tunes the round-trip time (RTT) timeout values.

Port Specification and Service Detection

Port Specification

`nmap -p <port(s)> <target>`	Specify Ports: Scans only the specified ports. Example: `-p 22,80,443` or `-p 1-1000`
`nmap -F <target>`	Fast Scan: Scans only the ports listed in the nmap-services file.
`nmap --top-ports <number> <target>`	Top Ports: Scans the specified number of most common ports.
`nmap -p- <target>`	Scan all 65535 ports.

Service and Version Detection

`nmap -sV <target>`	Version Detection: Enables version detection to determine the service and version information.
`nmap --version-intensity <0-9> <target>`	Version Intensity: Sets the intensity of version scanning. Higher values increase accuracy but take longer.
`nmap --version-light <target>`	Version Light: Uses light version scanning.
`nmap --version-all <target>`	Version All: Tries every single probe.

OS Detection

`nmap -O <target>`	OS Detection: Attempts to determine the operating system of the target.
`nmap --osscan-limit <target>`	OS Scan Limit: Limits OS detection to promising targets.
`nmap --osscan-guess <target>`	OS Scan Guess: Guesses the OS more aggressively.

Nmap Scripting Engine (NSE)

NSE Basics

The Nmap Scripting Engine (NSE) allows you to run powerful scripts to automate a wide variety of networking tasks. These scripts can discover vulnerabilities, perform version detection, and more.

Common NSE Categories

- **auth**: Scripts related to authentication bypass and checking.
- **broadcast**: Scripts that discover services by broadcasting requests on the local network.
- **default**: Commonly used scripts providing basic information.
- **discovery**: Scripts that try to discover more information about the network.
- **dos**: Scripts that test for denial-of-service vulnerabilities.
- **exploit**: Scripts that attempt to exploit known vulnerabilities.
- **fuzzer**: Scripts that send random data to services in an attempt to crash them.
- **intrusive**: Scripts that are considered intrusive and may cause damage.
- **malware**: Scripts that check for malware and backdoors.
- **safe**: Scripts that are considered safe to run.
- **vuln**: Scripts that check for vulnerabilities.

Script Selection and Execution

`nmap --script=<script(s)> <target>`	Run Scripts: Executes the specified NSE scripts. Example: `--script smb-vuln-ms17-010` or `--script vuln`
`nmap --script-args <args> <target>`	Script Arguments: Provides arguments to the NSE scripts.
`nmap --script-help <script(s)>`	Script Help: Displays help information about the specified script(s).
`nmap --script-updatedb`	Update Script Database: Updates the NSE script database.