NMAP CHEAT-SHEET

GOVERDHAN KUMAR

Nmap (Network Mapper) is a powerful open-source network scanning and security auditing tool. It is used to discover hosts and services on a computer network, thus creating a "map" of the network. Nmap was first developed by Gordon Lyon (also known as Fyodor Vaskovich) and has been actively maintained and enhanced by a dedicated community of developers.

Key features of Nmap include:

Host Discovery: Nmap can identify live hosts on a network by sending packets and analyzing the responses it receives. This helps network administrators to know what devices are active on their network.

Port Scanning: Nmap can perform various types of port scans to identify which ports are open on a target system. This information is crucial for understanding potential vulnerabilities and services running on the system.

Version Detection: Nmap can attempt to identify the version of services running on open ports. This helps in assessing the security posture of those services and determining if any known vulnerabilities exist.

OS Fingerprinting: Nmap is capable of performing OS detection, where it tries to identify the operating system running on a target machine based on its responses to Nmap probes.

Scriptable Interaction: Nmap allows users to write custom scripts using the Nmap Scripting Engine (NSE). These scripts can be used for advanced tasks like vulnerability scanning, service discovery, and more.

Flexible Output Formats: Nmap offers various output formats, making it easy to process and analyze the results. The outputs can be in plain text, XML, or even interactive formats.

Used for Security Auditing: Nmap is widely used by network administrators and security professionals to assess the security of their networks, identify potential vulnerabilities, and improve overall network defenses.

Command-Line Interface (CLI): Nmap is primarily used through the command line, allowing users to specify various options and parameters for customized scans.

However, it's essential to note that Nmap is a potent tool, and its misuse can have serious consequences. Unauthorized scanning of networks and systems may be illegal and can lead to severe legal repercussions. It is crucial to use Nmap responsibly and with permission from the network owners.

Nmap is available for various platforms, including Windows, macOS, and Linux, making it a versatile tool for network scanning and security auditing purposes.

nmap [Scan Type] [Options] {target}

Scan Type: This can be a specific scan type like a SYN scan (-sS), a TCP connect scan (-sT), an UDP scan (-sU), or more. We'll explore some common scan types later.

Options: These are additional parameters that modify the behavior of the scan. For example, you can use -p to specify specific ports to scan, -F for a fast scan, or -A for OS detection and version scanning.

target: This can be an IP address, a range of IP addresses, a hostname, or a CIDR notation specifying a subnet.

1. Basic Port Scanning:

nmap -p 80,443 <target>: Scan for open ports commonly used for web services, such as HTTP (port 80) and HTTPS (port 443).

2. Version Detection:

nmap -sV -p 80,443 <target>: Probe open ports to determine the service and version information running on ports 80 and 443.

3. Script Scanning:

nmap -p 80,443 --script http-enum <target>: Use Nmap's built-in script http-enum to enumerate directories and files on web servers.

4. Detecting Web Technologies:

nmap -p 80,443 --script http-headers <target>: Retrieve HTTP headers to identify the web server and web technologies in use.

5. Vulnerability Scanning:

nmap -p 80,443 --script http-vuln-* <target>: Run various NSE scripts to check for common web vulnerabilities such as SQL injection, XSS, etc.

6. HTTP Methods:

nmap -p 80,443 --script http-methods <target>: Identify which HTTP methods are supported by the web server (e.g., GET, POST, PUT, DELETE).

7. Directory Brute-Forcing:

nmap -p 80,443 --script http-enum --script-args http-enum.basepath='/webapp/' <target>: Brute force directories under a specific base path.

8. SSL Certificate Information:

nmap -p 443 --script ssl-cert <target>: Retrieve SSL certificate information for the HTTPS service on port 443.

9. OS Detection and Traceroute:

nmap -O -p 80,443 <target>: Conduct OS detection on the target, and include traceroute to identify the path taken to reach the target.

Host Discovery:

-sn: Ping Scan - Discover live hosts on the network using ICMP echo requests (ping).

-Pn: Treat all hosts as online, skip host discovery (assume all targets are alive).

Port Scanning:

-sS: TCP SYN Scan - Stealthy and fast scan, often used for default scanning.

-sT: TCP Connect Scan - Completes the full TCP handshake, making it more detectable.

-sU: UDP Scan - Identifies open UDP ports and services.

-sF, -sN, -sX: TCP FIN, Null, and Xmas scans, respectively.

Version Detection:

-sV: Probe open ports to determine service and version information.

OS Detection:

-O: Enable OS detection to identify the operating system of the target.

Scripting Engine (NSE):

--script <script-name>: Run an NSE script to perform additional tasks like vulnerability scanning or service discovery.

--script-args <arguments>: Pass arguments to an NSE script.

Timing and Performance:

-T<0-5>: Set the timing template (0=paranoid, 5=insane) to control scan speed and aggressiveness.

--max-retries <value>: Set the number of probe retransmissions.

--max-scan-delay <time>: Set the maximum delay between probes.

Output Control:

-**oN <file>:** Save scan results in normal text format.

-**oX <file>:** Save scan results in XML format.

-**oG <file>:** Save scan results in grepable format.

Port Specification:

-p <port ranges>: Specify the port ranges to scan (e.g., -p 1-100, -p 80,443, etc.).

-F: Fast Mode - Scan only the most common ports.

Miscellaneous:

-A: Aggressive Scan - Enables OS detection, version detection, script scanning, and traceroute.

-v: Increase verbosity level.

--traceroute: Trace the route to the target hosts during the scan.

--reason: Display the reason for the scan's status.

Basic Scanning Examples:

a. Simple Scan: To perform a basic TCP SYN scan on a target, use:

nmap -sS {target}

b. Scan a Specific Port Range: To scan a specific range of ports, use the -p option followed by the port numbers:

nmap -p 1-100 {target}

c. Detecting OS and Services: To perform a more comprehensive scan that includes OS detection, version detection, and script scanning, use the -A option:

nmap -A {target}

Output Formats:

By default, Nmap outputs the scan results to the terminal. However, you can save the results in various formats using the -o option. For example:

nmap -oN scan_results.txt {target} # Save results in normal text format

nmap -oX scan_results.xml {target} # Save results in XML format

Nmap Scripting Engine (NSE):

Nmap provides a powerful scripting engine that allows you to extend its functionality. NSE scripts can perform additional tasks like vulnerability scanning, brute forcing, and service discovery. To use an NSE script, use the --script option:

nmap --script <script-name> {target}

Comprehensive Scan Types:

TCP SYN Scan (-sS): Fast and stealthy, ideal for general port scanning.

TCP Connect Scan (-sT): Similar to SYN scan, but it connects to the target ports, making it more detectable.

UDP Scan (-sU): Scans for open UDP ports, useful for services not covered by TCP.

Comprehensive Scan (-A): OS detection, version detection, script scanning, and traceroute.

Please remember that while Nmap can be a useful tool for web pentesting, it should not be the sole tool used for comprehensive testing. Manual analysis, other web vulnerability scanners (like Burp Suite or OWASP ZAP), and in-depth code reviews are equally crucial for a thorough assessment of web application security. Additionally, ensure you have proper authorization before performing any web pentesting activities.

Follow :

https://www.linkedin.com/in/goverdhankumar
https://github.com/wh04m1i
https://linktr.ee/g0v3rdh4n