



NMAP USER GUIDE

Designed and Authored by Kershia Mukoro

TABLE OF CONTENTS

1	INTRODUCTION.....	3
1.1	Purpose	3
1.2	What is Nmap?	3
1.3	Benefits of Using Nmap.....	4
2	Installation.....	6
2.1	Windows.....	6
2.2	Linux.....	6
2.3	MacOS.....	6
3	Using NMAP.....	8
3.1	PREREQUISITES.....	8
3.2	Nmap Help.....	8
3.3	Scanning Techniques.....	11
3.4	Basic Scanning Techniques	11
3.5	Basic Scan Examples	12
3.6	Advanced Scanning Techniques.....	13
3.7	Advanced Scan Examples:.....	14
4	Expanding Your Nmap Toolkit.....	15
4.1	Verbosity and Exporting Scan Results.....	15
4.2	Export Methods	15
4.3	Nmap Scripting Engine (NSE)	16
4.4	Zenmap.....	17
5	Practical Examples	18
	Identifying Unknown Devices on Your Network.....	18
	Checking for Open Ports on a Server	18
	Detecting the Operating System of a Remote Host	19
	Finding Vulnerabilities with Nmap Scripting Engine	19
	Monitoring Network Traffic Flow.....	20
	Scanning a Range of IPs for a Specific Service	20
6	Troubleshooting.....	21
6.1	Installation Issues	21
6.2	Scan Not Starting	21

6.3	Scans Taking Too Long.....	21
6.4	Failing to Detect Services or OS.....	22
6.5	Encountering Errors with Scripts.....	22
6.6	Network or Firewall Blocking Scans.....	22
6.7	General Tips.....	23
7	Frequently Asked Questions (FAQs)	24
	Is Nmap legal?	24
	Can Nmap be detected?	24
	What is Nmap primarily used for?.....	24
	How do I update Nmap?	24
	Can Nmap scan through VPN?.....	24
	Do I need root access to use all the features of Nmap?	25
	What operating systems can run Nmap?.....	25
	How can Nmap help improve network security?	25
	Is Nmap difficult to learn for beginners?	25
	Can Nmap perform vulnerability scans?.....	25
8	Accessibility	26
8.1	Screen Readers	26
8.2	Keyboard Shortcuts	26
	8.2.1 Basic Navigation.....	27
	8.2.2 Editing	27
	8.2.3 History.....	27
	8.2.4 Process Control.....	28
	8.2.5 Window Management.....	28
	8.2.6 Miscellaneous.....	28
8.3	Text Size and Color Contrast	28
8.4	Command Line Accessibility Tips	29
8.5	Accessible Documentation	29
9	Glossary.....	30

1 INTRODUCTION

1.1 PURPOSE

This guide offers a concise overview of Nmap, the essential network scanning tool for IT security and inventory assessments. It includes key features, simple installation steps for multiple platforms, troubleshooting advice, and practical examples to help both new and experienced users efficiently leverage Nmap for network security and management. Use this guide to quickly understand and apply Nmap's capabilities in your IT environment.

1.2 WHAT IS NMAP?

Nmap is a free, open-source network discovery tool capable of swiftly scanning local and remote networks. Nmap can scan through thousands of connected devices and interrogate ports on specific targets. IP packets are used to identify and uncover information about the network and network devices. Network administrators can use Nmap to inventory, monitor, and discover real-time information on:

- Network devices
- Remote host statuses
- Hosts' services
- Operating systems (OS) and versions
- Firewall types
- Security vulnerabilities

1.3 BENEFITS OF USING NMAP

Nmap offers a range of benefits that make it an invaluable tool for network administration, security analysis, and penetration testing. Here's a list of the key advantages of using Nmap:

- **Network Mapping Efficiency:** Nmap simplifies the process of mapping a network, allowing users to quickly identify all connected devices such as servers, routers, switches, and mobile devices across single or multiple networks without the need for complex commands or configurations.
- **Service Identification:** Detects and identifies services running on systems, including web servers, DNS servers, and other common applications. This helps understand and manage what is happening on a network effectively.
- **Version Detection and Vulnerability Assessment:** Nmap not only identifies applications but also determines their versions with reasonable accuracy. This capability is essential for detecting known vulnerabilities specific to application versions, aiding in proactive security management.
- **Operating System Detection:** Nmap can determine the operating system and version on networked devices, helping users choose appropriate tools and methods for penetration testing based on the identified OS.
- **Scriptable Interaction:** The Nmap Scripting Engine allows users to automate a wide variety of networking tasks using scripts. This feature is particularly useful for conducting advanced network scans, security auditing, and vulnerability scanning more efficiently.

- **Graphical User Interface (Zenmap):** Zenmap, the GUI for Nmap, enhances user experience by providing visual mappings of networks. This makes it easier to analyze and report network structures, which is beneficial for both experienced and novice users.
- **Simplicity and Versatility:** Nmap supports simple commands for basic tasks like checking if a host is up, which makes it accessible for beginners, as well as complex scripting for experienced users, making it a versatile tool suitable for various use cases.
- **Security Auditing Tool:** Use Nmap for security auditing. The ability to simulate attacks using scripts from the Nmap Scripting Engine helps security professionals identify and mitigate potential vulnerabilities before they can be exploited.

2 INSTALLATION

Nmap is compatible with Windows, Linux, and Mac OS X. Download and install Nmap from the [official download page](#).

2.1 WINDOWS

- Go to the [Nmap download page](#).
- Click on the Windows installer link to download the executable file.
- Run the downloaded file and follow the installation prompts.

2.2 LINUX

- For Debian-based distributions, use:

```
sudo apt-get install nmap
```

- For Red Hat-based distributions, use:

```
sudo yum install nmap
```

- For Arch Linux, use:

```
sudo pacman -S nmap
```

2.3 MacOS

- Using Homebrew:

```
brew install nmap
```

- Alternatively, download the macOS version from the [Nmap download page](#) and follow the installation instructions provided.

3 USING NMAP

3.1 PREREQUISITES

Use of Nmap requires familiarity with command-line interfaces (CLI) for basic network monitoring.

3.2 NMAP HELP

- Run the command `nmap` with no arguments to use Nmap's built-in help command to display a list summarizing all flags and options (see **Figure 1** and **Figure 2**).¹
- The command `nmap -h` returns the same list of flags and options.
- The latest version of the man page is available [here](#).

¹ ATTENTION

Certain options will only work with specific types of scans and Nmap will warn users of unsupported option combinations.

```

Nmap 7.94 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags <flags>: Customize TCP scan flags
  -sI <zombie host[:probeport]>: Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b <FTP relay host>: FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p <port ranges>: Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  --exclude-ports <port ranges>: Exclude the specified ports from scanning
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports sequentially - don't randomize
  --top-ports <number>: Scan <number> most common ports
  --port-ratio <ratio>: Scan ports more common than <ratio>
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity <level>: Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)

```

Figure 1 Options Summary Page 1

```

SCRIPT SCAN:
-sC: equivalent to --script=default
--script=<Lua scripts>: <Lua scripts> is a comma separated list of
    directories, script-files or script-categories
--script-args=<n1=v1,[n2=v2,...]>: provide arguments to scripts
--script-args-file=filename: provide NSE script args in a file
--script-trace: Show all data sent and received
--script-updatedb: Update the script database.
--script-help=<Lua scripts>: Show help about scripts.
    <Lua scripts> is a comma-separated list of script-files or
    script-categories.

OS DETECTION:
-O: Enable OS detection
--osscan-limit: Limit OS detection to promising targets
--osscan-guess: Guess OS more aggressively

TIMING AND PERFORMANCE:
Options which take <time> are in seconds, or append 'ms' (milliseconds),
's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
-T<0-5>: Set timing template (higher is faster)
--min-hostgroup/max-hostgroup <size>: Parallel host scan group sizes
--min-parallelism/max-parallelism <numprobes>: Probe parallelization
--min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout <time>: Specifies
    probe round trip time.
--max-retries <tries>: Caps number of port scan probe retransmissions.
--host-timeout <time>: Give up on target after this long
--scan-delay/--max-scan-delay <time>: Adjust delay between probes
--min-rate <number>: Send packets no slower than <number> per second
--max-rate <number>: Send packets no faster than <number> per second

FIREWALL/IDS EVASION AND SPOOFING:
-f; --mtu <val>: fragment packets (optionally w/given MTU)
-D <decoy1,decoy2[,ME],...>: Cloak a scan with decoys
-S <IP_Address>: Spoof source address
-e <iface>: Use specified interface
-g/--source-port <portnum>: Use given port number
--proxies <url1,[url2],...>: Relay connections through HTTP/SOCKS4 proxies
--data <hex string>: Append a custom payload to sent packets
--data-string <string>: Append a custom ASCII string to sent packets
--data-length <num>: Append random data to sent packets
--ip-options <options>: Send packets with specified ip options
--ttl <val>: Set IP time-to-live field
--spoof-mac <mac address/prefix/vendor name>: Spoof your MAC address
--badsum: Send packets with a bogus TCP/UDP/SCTP checksum

OUTPUT:
-oN/-oX/-oS/-oG <file>: Output scan in normal, XML, s|<rIpt kIddi3,
    and Grepable format, respectively, to the given filename.
-oA <basename>: Output in the three major formats at once

```

Figure 2 Options Summary Page 2

3.3 SCANNING TECHNIQUES

Nmap's command-line interface gives users the ability to perform different kinds of scans. Run basic scans with simple command syntax or use a combination of command flags and parameters to execute more complex, finely tuned scans.

3.4 BASIC SCANNING TECHNIQUES

- **Ping Scan:** `nmap -sn (target)`

Use this command to quickly scan the target to check if it's online without performing a full port scan. A ping scan identifies all the IP addresses online without sending packets to the hosts.

- **Port Scan:** `nmap (target)`

The basic command to perform a port scan on the target to identify open ports.

- **Version Detection:** `nmap -sV (target)`

This command enables version detection and gives more information about the services running on the open ports.

3.5 BASIC SCAN EXAMPLES

- To scan a single IP:

```
nmap 192.168.1.1
```

- To scan a range:

```
nmap 192.168.1.1-255
```

- To scan a subnet:

```
nmap 192.168.1.0/24
```

- To scan from a file:

```
nmap -iL /input_ips.txt
```

3.6 ADVANCED SCANNING TECHNIQUES

- **Stealth Scan:** `nmap -sS (target)`

This command performs a SYN scan. This type of scan is less likely to be logged by the target's system, making it a stealthier option for scanning.

- **OS Detection:** `nmap -O (target)`

Enables operating system detection, which tries to identify the operating system running on the target.

- **Aggressive Scan:** `nmap -A (target)`

This combines version detection, OS detection, script scanning, and traceroute in one command, and provides a comprehensive overview of the target.

- **Script Scan:** `nmap --script=(script category) (target)`

Allows the user to specify a category of NSE (Nmap Scripting Engine) scripts to run against the target. Offers more in-depth probing based on the scripts' purposes.

3.7 ADVANCED SCAN EXAMPLES:

- For stealth scanning a single IP:

```
nmap -sS 192.168.1.1
```

- For aggressive scanning of a subnet:

```
nmap -A 192.168.1.0/24
```

- For OS detection:

```
nmap -O 192.168.1.1
```

- For running default scripts against a target:

```
nmap --script=default 192.168.1.1
```

Nmap is a powerful tool and advanced scanning techniques should be used with caution. Advanced scans can be intrusive and are more likely to be detected by intrusion detection systems. Use Nmap ethically, and always ensure you have explicit permission to scan the network or device. For more detailed information and advanced scanning techniques, refer to the [Nmap Official Documentation](#).

4 EXPANDING YOUR NMAP TOOLKIT

4.1 VERBOSITY AND EXPORTING SCAN RESULTS

Penetration tests can extend over days or weeks. Exporting Nmap results prevents redundant work and aids in report creation.

4.2 EXPORT METHODS

- **Verbose Output:** Provides detailed information about the scan and is useful for monitoring the actions taken by Nmap on a network:

```
nmap -v scanme.nmap.org
```

- **Normal Output:** Exports the scan results to a text file:

```
nmap -oN output.txt scanme.nmap.org
```

- **XML Output:** Preferred by many pen-testing tools for its parsability:

```
nmap -oX output.xml scanme.nmap.org
```

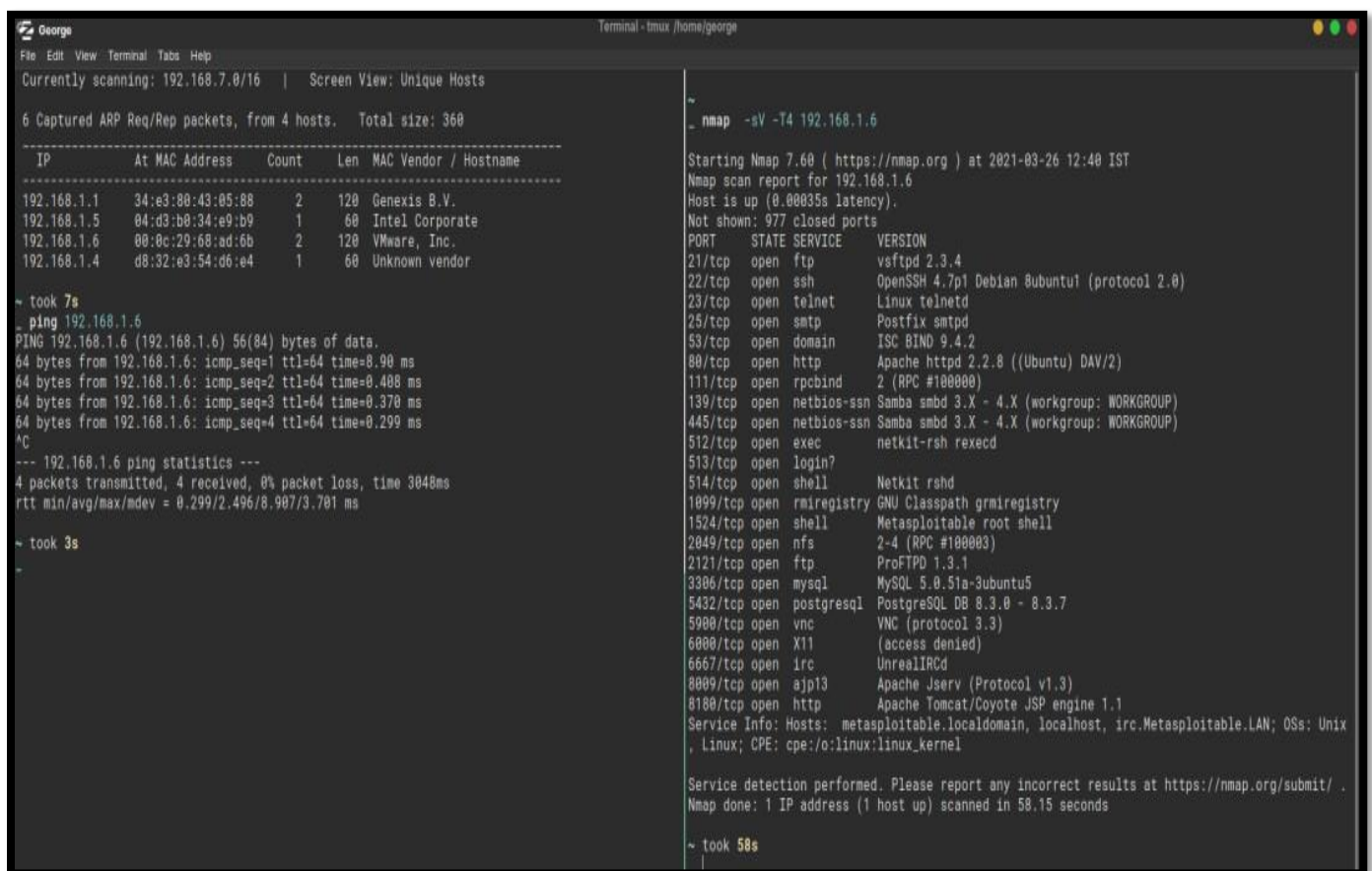
- **Multiple Formats:** Export scan results in all available formats simultaneously:

```
nmap -oA output scanme.nmap.org
```

4.3 NMAP SCRIPTING ENGINE (NSE)

NSE offers a powerful toolset for automating networking tasks through scripting.

Users can access numerous scripts, modify them with Lua, or create custom scripts tailored to specific needs. NSE also includes scripts for network attacks. More in-depth information on Nmap's scripting engine can be found [here](#).



```
George
File Edit View Terminal Tabs Help
Currently scanning: 192.168.7.0/16 | Screen View: Unique Hosts

6 Captured ARP Req/Rep packets, from 4 hosts. Total size: 360

-----
IP           At MAC Address  Count  Len  MAC Vendor / Hostname
-----
192.168.1.1   34:e3:80:43:05:88  2    120  Genexis B.V.
192.168.1.5   04:d3:b0:34:e9:b9  1     60  Intel Corporate
192.168.1.6   00:0c:29:68:ad:6b  2    120  VMware, Inc.
192.168.1.4   d8:32:e3:54:d6:e4  1     60  Unknown vendor

~ took 7s
~ ping 192.168.1.6
PING 192.168.1.6 (192.168.1.6) 56(84) bytes of data:
64 bytes from 192.168.1.6: icmp_seq=1 ttl=64 time=0.90 ms
64 bytes from 192.168.1.6: icmp_seq=2 ttl=64 time=0.408 ms
64 bytes from 192.168.1.6: icmp_seq=3 ttl=64 time=0.370 ms
64 bytes from 192.168.1.6: icmp_seq=4 ttl=64 time=0.299 ms
^C
--- 192.168.1.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3048ms
rtt min/avg/max/mdev = 0.299/2.496/8.907/3.701 ms

~ took 3s
-

~ nmap -sV -T4 192.168.1.6

Starting Nmap 7.60 ( https://nmap.org ) at 2021-03-26 12:40 IST
Nmap scan report for 192.168.1.6
Host is up (0.00035s latency).
Not shown: 977 closed ports
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          vsftpd 2.3.4
22/tcp    open  ssh          OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp    open  telnet       Linux telnetd
25/tcp    open  smtp         Postfix smtpd
53/tcp    open  domain       ISC BIND 9.4.2
80/tcp    open  http         Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp   open  rpcbind      2 (RPC #100000)
139/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp   open  exec         netkit-rsh rshd
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
1099/tcp  open  rmiregistry  GNU Classpath grmiregistry
1524/tcp  open  shell        Metasploitable root shell
2049/tcp  open  nfs          2-4 (RPC #100003)
2121/tcp  open  ftp          ProFTPD 1.3.1
3306/tcp  open  mysql        MySQL 5.0.51a-3ubuntu5
5432/tcp  open  postgresql   PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp  open  vnc          VNC (protocol 3.3)
6000/tcp  open  X11          (access denied)
6667/tcp  open  irc          UnrealIRCd
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
8180/tcp  open  http         Apache Tomcat/Coyote JSP engine 1.1
Service Info: Hosts: metasploitable.localdomain, localhost, irc.Metasploitable.LAN; OSs: Unix
, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 58.15 seconds

~ took 58s
|
```

Figure 3 Nmap Scripting Engine (NSE) for Network Automation and Security

4.4 ZENMAP

Zenmap is the graphical user interface for Nmap, offering an easier start for beginners with visual network mappings. Users can save and search scans without needing to navigate through command-line interfaces. Zenmap is free, open source, and usually included in the Nmap download. Users can download Zenmap from the official [Nmap download page](#) and more information on using Zenmap can be found [here](#).

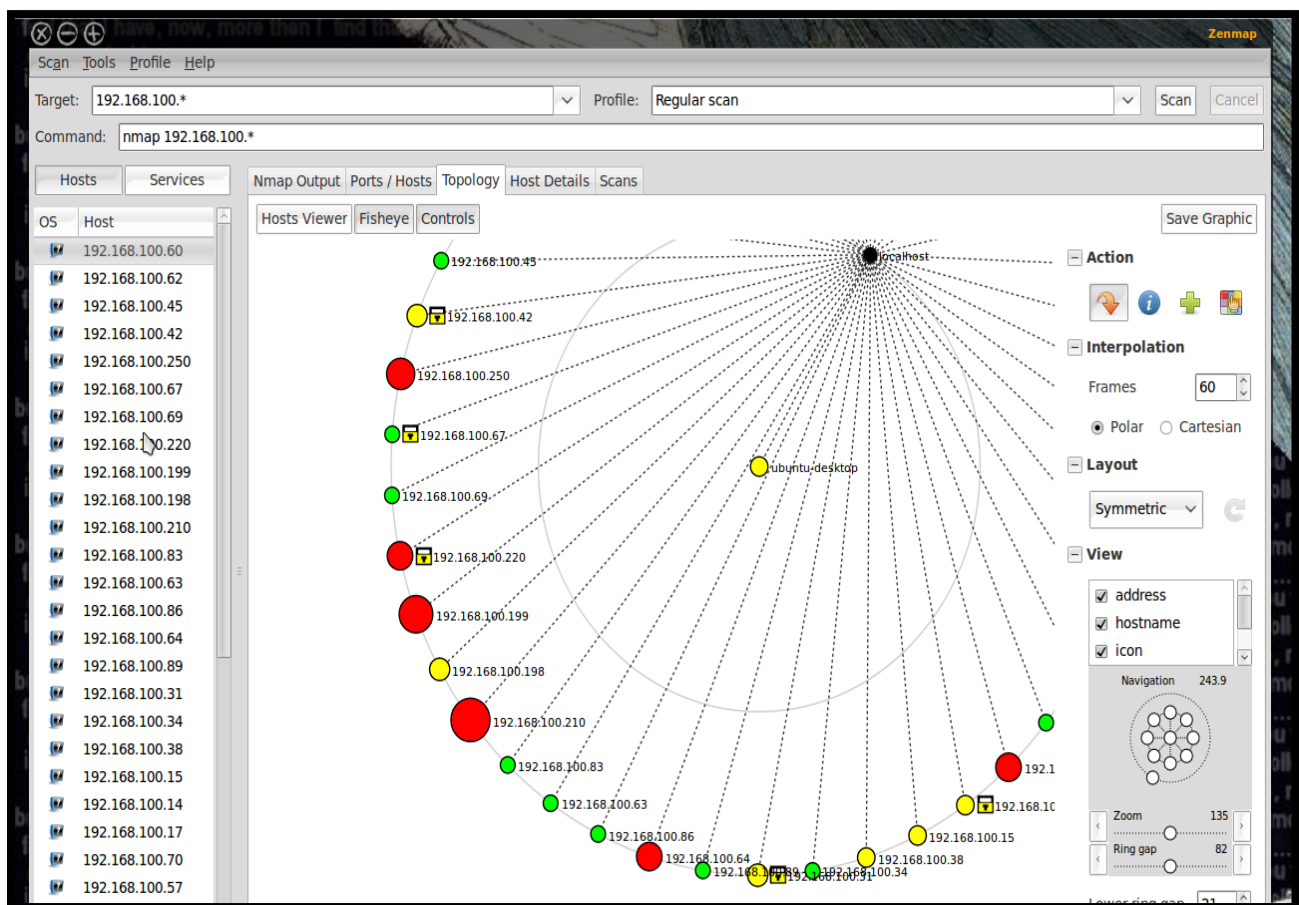


Figure 4 Zenmap's Graphical User Interface for Nmap Network Scanning

5 PRACTICAL EXAMPLES

5.1 IDENTIFYING UNKNOWN DEVICES ON YOUR NETWORK

Scenario: You have noticed unusual network activity and want to identify all devices connected to your network.

Command:

```
nmap -sn 192.168.1.0/24
```

Explanation: This command performs a ping sweep over your entire subnet to list devices that are online. It helps with quickly identifying all active IPs without performing a deep scan, which is useful for an initial investigation.

5.2 CHECKING FOR OPEN PORTS ON A SERVER

Scenario: Before deploying a new application, you want to ensure no unnecessary ports are open on the server.

Command:

```
nmap -T4 -F 192.168.1.100
```

Explanation: This command scans the most common ports quickly (-T4 for faster execution and -F for fast mode). It's ideal for routine checks to ensure that only essential ports are open, reducing potential vulnerabilities.

5.3 DETECTING THE OPERATING SYSTEM OF A REMOTE HOST

Scenario: You need to verify the operating system of a remote machine to ensure compliance with your IT infrastructure policies.

Command:

```
nmap -O --osscan-guess 192.168.1.101
```

Explanation: This command performs an OS detection. The --osscan-guess option tells Nmap to take its best guess if uncertain. OS detection is useful for compliance or troubleshooting.

5.4 FINDING VULNERABILITIES WITH NMAP SCRIPTING ENGINE

Scenario: You suspect a device might be vulnerable to a specific exploit and want to check it using Nmap's scripting capabilities.

Command:

```
nmap --script=vuln 192.168.1.102
```

Explanation: This command uses Nmap's built-in vulnerability scanning scripts to check for common vulnerabilities. It is a powerful way to proactively find and mitigate potential security issues.

5.5 MONITORING NETWORK TRAFFIC FLOW

Scenario: You need to track how traffic flows through your network to identify potential bottlenecks or unauthorized activity.

Command:

```
nmap --traceroute 192.168.1.103
```

Explanation: This command traces the path that packets take to reach the host. It is useful for diagnosing network routing issues and optimizing network paths.

5.6 SCANNING A RANGE OF IPS FOR A SPECIFIC SERVICE

Scenario: You need to find all printers (usually on port 9100) in a large subnet.

Command:

```
nmap -p 9100 192.168.1.0/24
```

Explanation: This command scans for a specific port across an entire subnet, helping to quickly locate all devices offering that service, such as printers

6 TROUBLESHOOTING

Users may encounter various challenges or errors using Nmap. Here are some quick tips to solving some of the most common issues:

6.1 INSTALLATION ISSUES

- Ensure your system meets Nmap's requirements:
- **Linux:** Use your package manager, e.g., for Debian-based systems, use:

```
sudo apt-get install nmap
```

- **Windows:** Download the installer from the [official Nmap site](#). If you encounter issues, try running the installer as an administrator.

6.2 SCAN NOT STARTING

- Verify target IP address or domain is correct and reachable.
- Ensure you have network connectivity.
- If running Nmap with **sudo** or as an administrator, double-check your permissions.

6.3 SCANS TAKING TOO LONG

- Use more specific IP ranges or fewer ports.
- Consider splitting the scan into smaller segments for large networks,
- Adjust timing options, e.g., **nmap -T4** for a faster scan, noting that increasing speed can reduce accuracy.

6.4 FAILING TO DETECT SERVICES OR OS

- Check the options selected are correct, e.g., **-sV** for service version detection and **-O** for OS detection.
- Increase verbosity with **-v** for more scan details, which might give insights into what is happening.
- For OS detection, make sure enough ports are open and responsive. Ensure you have at least one open and one closed port for Nmap to function correctly.

6.5 ENCOUNTERING ERRORS WITH SCRIPTS

- Verify the script syntax and ensure Nmap's script database includes the script.
- Use **--script-help** to get information about a script's usage and options.
- If a script requires arguments, ensure they are correctly formatted, e.g., **--script script-name --script-args arg1, arg2**.

6.6 NETWORK OR FIREWALL BLOCKING SCANS

- Some networks or firewalls may block scanning activities. Ensure you have the necessary permissions and that your IP is not blacklisted.
- If possible, configure the firewall to allow your scanning activity, or consider using less aggressive scan techniques.

6.7 GENERAL TIPS

- Always consult the Nmap documentation for detailed information and troubleshooting.
- Join the [Nmap community](#) forums or [mailing lists](#) for support and to share your experiences.
- Remember, unauthorized scanning can be illegal or unethical.
- Always have explicit permission before scanning networks.
- For more detailed guides and advanced usage, refer to the [official Nmap documentation](#).

7 FREQUENTLY ASKED QUESTIONS (FAQs)

Is Nmap legal?

Yes, Nmap is legal software. However, using Nmap to scan networks without permission is illegal in many jurisdictions.

Can Nmap be detected?

Yes, some systems and firewalls are configured to detect scanning activities, including those from Nmap.

What is Nmap primarily used for?

Nmap is primarily used for network discovery and security auditing.

How do I update Nmap?

Depending on your system, you can update Nmap using your package manager or by downloading the latest version from the official Nmap website.

Can Nmap scan through VPN?

Yes, Nmap can scan through a VPN, although the VPN's configuration might influence the results.

Do I need root access to use all the features of Nmap?

Yes, to use all the features of Nmap, especially those that require raw packet capabilities (like OS fingerprinting and certain types of scans), users need root access or its equivalent on Windows systems.

What operating systems can run Nmap?

Nmap is versatile and can be run on various operating systems, including Windows, Linux, and macOS.

How can Nmap help improve network security?

Nmap can identify open ports, services running, and their respective versions on network devices, helping pinpoint potential vulnerabilities that need to be addressed.

Is Nmap difficult to learn for beginners?

While Nmap has a multitude of features and capabilities, there are many resources available that make it accessible for beginners. The basic commands and functions can be learned relatively quickly.

Can Nmap perform vulnerability scans?

Yes, when used with the NSE (Nmap Scripting Engine), Nmap can perform various vulnerability scans, checking for specific weaknesses in systems and applications.

8 ACCESSIBILITY

Ensuring Nmap is accessible to all users, including those with disabilities, is important. Here are some tips and tools to help make using Nmap more accessible:

8.1 SCREEN READERS

Users who are visually impaired can use screen readers to interact with the command line:

- **Windows:** Use built-in screen readers like Narrator, or third-party applications like NVDA (NonVisual Desktop Access).
- **Linux:** Tools like Orca can provide spoken feedback of text that appears on the screen.
- **macOS:** VoiceOver is integrated into the macOS system and provides comprehensive screen reading capabilities.

8.2 KEYBOARD SHORTCUTS

Navigating with keyboard shortcuts can enhance usability for users with limited mobility:

- Most terminals allow you to use keyboard shortcuts to navigate the command line. Learning these shortcuts can reduce the need for mouse input and improve efficiency.
- Customizable shortcuts in terminal applications can help users set up their environments according to their needs.

Here is a list of commonly used shortcuts in terminal applications that enhance navigation and efficiency. These can generally be used in most terminal environments like Bash on Linux or macOS, and Command Prompt or PowerShell on Windows.

8.2.1 Basic Navigation

- **Ctrl + A:** Move cursor to the beginning of the line.
- **Ctrl + E:** Move cursor to the end of the line.
- **Ctrl + Left Arrow/Right Arrow:** Move cursor one word left/right (this may vary depending on the terminal).
- **Ctrl + U:** Clears the line from the cursor to the beginning.
- **Ctrl + K:** Clears the line from the cursor to the end.

8.2.2 Editing

- **Ctrl + W:** Cut the word before the cursor, adding it to the clipboard.
- **Ctrl + Y:** Paste the last thing to be cut from the clipboard.
- **Alt + Backspace:** Delete the word before the cursor (this shortcut might behave differently on some terminals).

8.2.3 History

- **Ctrl + P:** Move one command up in the command history.
- **Ctrl + N:** Move one command down in the command history.
- **Ctrl + R:** Search the command history as you type.

8.2.4 Process Control

- **Ctrl + C:** Terminate the current command.
- **Ctrl + Z:** Suspend the current command by sending it to the background.

8.2.5 Window Management

- **Ctrl + L:** Clear the screen (similar effect to the clear command).
- **Ctrl + S:** Pause the output on the screen (useful for long outputs).
- **Ctrl + Q:** Resume output to the screen if paused.

8.2.6 Miscellaneous

- **Ctrl + D:** Close the terminal if the line is empty. In the shell, this logs you out or exits the terminal.

8.3 TEXT SIZE AND COLOR CONTRAST

Adjusting text size and color contrast can help users with visual impairments:

- Increase the text size in your terminal settings to make the output of Nmap scans easier to read.
- Adjust color schemes for higher contrast in terminal settings. High contrast themes can be particularly helpful.

8.4 COMMAND LINE ACCESSIBILITY TIPS

- Use tab completion to reduce the amount of typing needed, making command line use easier for those with motor disabilities.
- Regularly clear the screen to reduce clutter and make the current command more visible (`clear` on Linux/ macOS and `cls` on Windows).

8.5 ACCESSIBLE DOCUMENTATION

Ensure that all Nmap documentation is accessible:

- Provide text-based versions of graphical content, such as diagrams or screenshots.
- Use clear, concise language and structure content with headings for easier navigation by screen readers.

9 GLOSSARY

Accessibility: The practice of making your websites usable by as many people as possible, often focusing on individuals with disabilities, but also benefiting a broader user base.

Aggressive Scan: An intensive scan that combines multiple features like version detection, OS detection, script scanning, and traceroute to provide a comprehensive view of the target's security profile.

Command-line Interface (CLI): A text-based interface used for entering commands directly to a software application.

Datagrams: Independent packets of data sent over a network without requiring an established connection, which allows for fast but unreliable data transfer. Each datagram contains its own addressing information.

Domain: A domain refers to a group of computers and devices that are managed as a unit with common rules and procedures. Within the Internet, domains are defined by their domain name and have a tree-like structure under the Domain Name System (DNS).

Domain Name: An identification string that defines a realm of administrative autonomy, authority, or control within the Internet.

Domain Name System (DNS): The system by which Internet domain names and addresses are tracked and regulated. Memorizable domain names are converted to the numerical IP addresses needed for locating computer services and devices.

Firewall: A network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules.

Graphical User Interface (GUI): A type of user interface that allows users to interact with electronic devices using graphical icons and visual indicators, instead of just text-based interfaces, typed command labels, or text navigation.

Hops: The number of intermediate devices, like routers, that a data packet passes through from its source to its destination. Each hop represents one step in the journey of the packet across the network.

Host: Any device that connects to a computer network, capable of sending or receiving data, performing services, and communicating with other network devices.

IP Address: A unique string of numbers separated by periods or colons that identifies each computer using the Internet Protocol to communicate over a network.

IP Address (Blacklisted): An IP address that has been denied access or flagged on a network due to suspicious or malicious activity. Blacklisting helps in protecting networks by blocking potentially harmful traffic from known bad sources.

IP (Internet Protocol): The principal communications protocol in the Internet protocol suite for relaying datagrams across network boundaries, essential for connectivity and the functioning of the Internet.

Internet Protocol Suite: A set of communications protocols used for the Internet and similar networks, often referred to as TCP/IP, which includes protocols for

routing network traffic, addressing hosts, and ensuring data can be transmitted reliably.

IP Layer: The part of the network that is responsible for sending data packets from one device to another across different networks, using IP addresses to identify devices.

IPv4/IPv6 (Internet Protocol version 4/version 6): Versions of the internet protocol, with IPv4 using a 32-bit address scheme and IPv6 using a 128-bit address scheme to support more devices.

Intrusion Detection System (IDS): Software or devices that monitor networks or systems for malicious activities or policy violations, typically reporting to an administrator or a Security Information and Event Management (SIEM) system.

IT Infrastructure: The combination of hardware, software, network resources, and services required for the operation, management, and monitoring of an enterprise IT environment. This includes physical devices and facilities, data storage solutions, network systems, and software applications.

Latency: The delay before a transfer of data begins following an instruction for its transfer, critical in activities requiring real-time responses.

Lua: A lightweight, high-level programming language used within NSE (Nmap Scripting Engine) to modify existing scripts or create custom scripts.

NSE (Nmap Scripting Engine): An advanced feature of Nmap that allows for the automation of a wide range of networking tasks, including vulnerability scans.

Network Traffic: The amount of data moving across a network at a given time. Monitored for performance measurement, network management, and security monitoring.

OS Detection: The process of determining the operating system of a host, utilized by Nmap to provide detailed security insights.

Packet: A small segment of data sent over a computer network.

Parsability: Refers to how easily a computer can understand, and process text, code, or data based on set rules. If something is highly parsable, it means it is easy for a computer to read and use.

Penetration Testing (Pen Testing): A security exercise where a cybersecurity expert actively tries to exploit vulnerabilities in a computer system, network, or web application. The purpose is to identify security weaknesses and test the effectiveness of existing security measures. Penetration testers use various methods and tools to simulate real-world attacks under controlled conditions, helping organizations strengthen their defenses against potential cyber threats.

Ping Scan: A method used by Nmap to determine which hosts are online on a network without performing a full port scan.

Port: A virtual data connection used by programs and services to exchange information, identified by specific numbers.

Ports (Open): Network ports that are open and accepting connections, potentially exploitable if not properly secured.

Ports (Closed): Network ports that reject incoming packets. Closed ports respond to inquiries but do not allow communication through, indicating no active service is listening at that port.

Port Scan: A network scan that identifies which ports on a host are open and potentially vulnerable to security breaches.

Raw Packet Capabilities: The ability to craft and manipulate packets at the IP layer, enabling detailed network traffic analysis.

Routing Issues: Problems that occur in the network layer responsible for directing data packets from a source to a destination. Routing issues can lead to data loss, delays, or data sent over less optimal paths.

Script Scan: Uses predefined scripts to perform advanced network discovery and security auditing through Nmap's Scripting Engine (NSE).

Security Information and Event Management (SIEM): A set of tools and services offering a holistic view of an organization's information security.

Services: Programs or processes that run on devices within a network and communicate over it.

Stealth Scan: A type of scan that attempts to interact with a host in a way that minimizes the chances of discovery by intrusion detection systems.

Sudo: A command in UNIX and Linux operating systems that allows a permitted user to execute a command as the superuser or another user, as specified by the security policy.

Subnet: A logically visible section of a larger network, created by dividing it into smaller, manageable pieces. Subnetting is the practice of dividing a network into two or more networks. Subnets help organize network traffic and improve security by reducing the area over which data is broadcast.

Subnet Mask: A 32-bit number that masks an IP address and divides the IP address into network and host address segments.

Traceroute: A network diagnostic tool integrated into Nmap that traces the path packets take from the host to the target, identifying all the hops along the way.

Transmission Control Protocol (TCP): A core protocol of the Internet Protocol Suite that enables reliable, ordered, and error-checked delivery of data between applications running on hosts communicating via an IP network.

Target: In the context of Nmap, a target refers to a specific device, system, or network selected for scanning and evaluation purposes.

Unauthorized Activity: Any action on a network or system that does not have official approval or violates security policies. This may include accessing restricted files, transmitting sensitive data, or exploiting system vulnerabilities.

User Datagram Protocol (UDP): A communications protocol used for establishing low-latency and loss-tolerating connections between applications on the internet.

Verbosity: A setting or option in many programs, like Nmap, that allows the user to increase the amount of detail provided in the output of a command or process. Higher verbosity levels provide more detailed diagnostic information.

Version Detection: A feature in Nmap that allows the user to determine the version of the service running on an open port.

Vulnerability Scan: A process that uses scripts to identify security weaknesses in network devices or applications.