nmap Cheatsheet

By Dejan Panovski • Updated on Aug 14, 2026 •

Quick reference for host discovery, port scanning, and service detection with nmap in Linux

The `nmap` command scans networks to find live hosts, open ports, and the services behind them. This cheatsheet covers target specification, host discovery, scan types, version and OS detection, timing, output formats, and NSE scripts. Only scan hosts and networks you own or have written permission to test.

Basic Syntax

Core `nmap` command forms.

`nmap 192.168.1.10`	Scan the 1000 most common TCP ports on one host
`sudo nmap 192.168.1.10`	Same scan as root, which enables the faster SYN scan
`nmap -v 192.168.1.10`	Increase verbosity, repeat as `-vv` for more detail
`nmap --reason 192.168.1.10`	Show why each port is in its reported state
`nmap --open 192.168.1.10`	Report only ports that are open

Target Specification

Point a scan at one host, a range, a subnet, or a list.

`nmap 192.168.1.10 10.0.0.5`	Scan several hosts in one run
`nmap 192.168.1.0/24`	Scan a whole subnet in CIDR notation
`nmap 192.168.1.1-50`	Scan an address range
`nmap -iL targets.txt`	Read targets from a file, one per line
`nmap --exclude 192.168.1.1 192.168.1.0/24`	Skip specific hosts in a larger scan
`nmap -6 2001:db8::1`	Scan an IPv6 target

Host Discovery

Find which hosts are up before spending time on port scans.

`sudo nmap -sn 192.168.1.0/24`	Ping scan: list live hosts without scanning ports
`nmap -Pn 192.168.1.10`	Skip discovery and treat the host as online
`nmap -PS22,80,443 192.168.1.0/24`	Discover hosts with TCP SYN probes to those ports
`nmap -PA80 192.168.1.0/24`	Discover hosts with TCP ACK probes
`sudo nmap -PE 192.168.1.0/24`	Discover hosts with ICMP echo requests
`nmap -n 192.168.1.0/24`	Skip reverse DNS lookups to speed up the scan

Port Selection

Control which ports the scan covers.

`nmap -p 22 192.168.1.10`	Scan a single port
`nmap -p 22,80,443 192.168.1.10`	Scan a list of ports
`nmap -p 1-1024 192.168.1.10`	Scan a port range
`nmap -p- 192.168.1.10`	Scan all 65535 TCP ports
`nmap -F 192.168.1.10`	Fast scan of the top 100 ports
`nmap --top-ports 20 192.168.1.10`	Scan the 20 most common ports
`sudo nmap -sU -sS -p U:53,T:80,443 192.168.1.10`	Mix UDP and TCP ports, which needs both scan types

Scan Types

Pick how nmap probes each port. The raw-packet scans need root.

`sudo nmap -sS 192.168.1.10`	TCP SYN scan: fast, the default when running as root
`nmap -sT 192.168.1.10`	TCP connect scan: the fallback for an unprivileged user
`sudo nmap -sU 192.168.1.10`	UDP scan: slow, so pair it with `--top-ports`
`sudo nmap -sA 192.168.1.10`	ACK scan: map which ports a firewall filters
`sudo nmap -sn 192.168.1.0/24`	No port scan at all, discovery only
`nmap -sL 192.168.1.0/24`	List scan: show the targets without sending probes

Service and OS Detection

Identify what is listening and what the host is running.

`nmap -sV 192.168.1.10`	Detect service names and version numbers
`nmap -sV --version-intensity 9 192.168.1.10`	Probe harder for versions, from 0 (light) to 9
`sudo nmap -O 192.168.1.10`	Guess the operating system from TCP/IP fingerprints
`sudo nmap -A 192.168.1.10`	Aggressive scan: `-sV`, `-O`, default scripts, and traceroute
`sudo nmap --traceroute 192.168.1.10`	Trace the route to each target

Timing and Performance

Trade speed against accuracy and network load.

`nmap -T4 192.168.1.10`	Faster timing, a good default on a local network
`nmap -T2 192.168.1.10`	Slower and gentler on fragile or busy networks
`nmap --min-rate 1000 192.168.1.0/24`	Send at least 1000 packets per second
`nmap --max-retries 1 192.168.1.0/24`	Give up sooner on unanswered probes
`nmap --host-timeout 5m 192.168.1.0/24`	Abandon any host still scanning after 5 minutes

Output Formats

Save results for review, diffing, or scripting.

`nmap -oN scan.txt 192.168.1.10`	Write human readable output to a file
`nmap -oX scan.xml 192.168.1.10`	Write XML for other tools to parse
`nmap -oG scan.gnmap 192.168.1.10`	Write grepable output for `grep` and `awk`
`nmap -oA scan 192.168.1.10`	Write all three formats using one base name
`nmap --append-output -oN scan.txt 192.168.1.11`	Append to an existing output file

NSE Scripts

Run the Nmap Scripting Engine for deeper checks.

`nmap -sC 192.168.1.10`	Run the default script set
`nmap --script http-title 192.168.1.10`	Run one named script
`nmap --script "http-*" 192.168.1.10`	Run every script matching a pattern
`nmap --script vuln 192.168.1.10`	Run the vulnerability script category
`nmap --script ssl-enum-ciphers -p 443 example.com`	List the TLS ciphers a server accepts
`nmap --script-help ssh-auth-methods`	Read what a script does before running it

Common Use Cases

Practical scans for day-to-day network work.

`sudo nmap -sn 192.168.1.0/24`	Inventory the hosts on a local network
`sudo nmap -p- -T4 192.168.1.10`	Find every open TCP port on one server
`nmap -sV --open 192.168.1.10`	List running services and their versions
`nmap -p 80,443 --open 192.168.1.0/24`	Locate the web servers on a subnet
`sudo nmap -sU --top-ports 20 192.168.1.10`	Check the most common UDP services
`sudo nmap -A -oA audit 192.168.1.10`	Full audit of one host, saved in every format

Troubleshooting

Quick checks for common `nmap` problems.

`You requested a scan type which requires root privileges`	Run the command with `sudo`, or use `-sT` instead of `-sS`
`Note: Host seems down`	Add `-Pn` when ICMP is blocked but the host is reachable
`Scan takes far too long`	Add `-T4`, narrow the ports with `-F` or `--top-ports`, and add `-n`
`Every UDP port shows open|filtered`	UDP has no handshake, so add `-sV` or scan fewer ports for a clearer answer
`Results differ from a local port check`	The firewall may filter the port; compare with `ss -tulpn` on the host itself

Related Guides

Use these guides for broader port and network troubleshooting.

nmap Command in Linux	Full `nmap` guide with detailed examples
How to Check Open Ports in Linux	Check listening ports from the host itself
ss Command in Linux	Inspect sockets and listening services
netcat cheatsheet	Test single ports and move data between hosts
tcpdump cheatsheet	Capture the packets behind a scan result