tcpdump Cheatsheet

By Dejan Panovski • Updated on Mar 11, 2026 • Download PDF

Quick reference for capturing and filtering network packets with tcpdump in Linux

The `tcpdump` command captures and filters network packets from the command line. This cheatsheet covers interfaces, capture filters, protocol and host matching, writing to pcap files, and practical troubleshooting patterns.

Basic Syntax

Core `tcpdump` command forms.

`sudo tcpdump`	Start capturing on the default interface
`sudo tcpdump -i eth0`	Capture on a specific interface
`sudo tcpdump -i any`	Capture on all interfaces
`sudo tcpdump -D`	List available interfaces
`sudo tcpdump -h`	Show help and usage

Limit and Format Output

Control how much data is shown and how packets are displayed.

`sudo tcpdump -c 10`	Stop after 10 packets
`sudo tcpdump -n`	Do not resolve hostnames
`sudo tcpdump -nn`	Do not resolve hostnames or service names
`sudo tcpdump -v`	Verbose output
`sudo tcpdump -X`	Show packet contents in hex and ASCII

Protocol Filters

Capture only the protocol traffic you care about.

`sudo tcpdump tcp`	Capture TCP packets only
`sudo tcpdump udp`	Capture UDP packets only
`sudo tcpdump icmp`	Capture ICMP packets only
`sudo tcpdump arp`	Capture ARP traffic
`sudo tcpdump port 53`	Capture DNS traffic on port 53

Host and Port Filters

Match packets by source, destination, host, or port.

`sudo tcpdump host 192.168.1.10`	Capture traffic to or from one host
`sudo tcpdump src host 192.168.1.10`	Capture packets from one source host
`sudo tcpdump dst host 192.168.1.10`	Capture packets to one destination host
`sudo tcpdump port 22`	Capture SSH traffic
`sudo tcpdump src port 443`	Capture packets from source port 443

Combine Filters

Use boolean operators to build precise capture expressions.

`sudo tcpdump 'tcp and port 80'`	Capture HTTP traffic over TCP
`sudo tcpdump 'host 10.0.0.5 and port 22'`	Capture SSH traffic for one host
`sudo tcpdump 'src 10.0.0.5 and dst port 443'`	Match one source and HTTPS destination
`sudo tcpdump 'port 80 or port 443'`	Capture HTTP or HTTPS traffic
`sudo tcpdump 'net 192.168.1.0/24 and not port 22'`	Capture a subnet except SSH

Write and Read Capture Files

Save traffic to a file or inspect an existing pcap capture.

`sudo tcpdump -w capture.pcap`	Write packets to a pcap file
`sudo tcpdump -r capture.pcap`	Read packets from a pcap file
`sudo tcpdump -i eth0 -w web.pcap port 80`	Save filtered traffic to a file
`sudo tcpdump -nn -r capture.pcap`	Read a file without name resolution
`sudo tcpdump -r capture.pcap 'host 10.0.0.5'`	Apply a filter while reading a pcap

Common Use Cases

Practical commands for day-to-day packet inspection.

`sudo tcpdump -i any port 22`	Watch SSH connections
`sudo tcpdump -i any port 53`	Inspect DNS queries and replies
`sudo tcpdump -i eth0 host 8.8.8.8`	Trace traffic to one external host
`sudo tcpdump -i any 'tcp port 80 or tcp port 443'`	Watch web traffic
`sudo tcpdump -i any icmp`	Check ping and ICMP traffic

Troubleshooting

Quick checks for common `tcpdump` issues.

`You do not have permission to capture on that device`	Run with `sudo` or verify packet-capture capabilities
`No packets appear`	Confirm the correct interface with `tcpdump -D` and use `-i any` if needed
`Hostnames make output slow`	Add `-n` or `-nn` to disable name resolution
`Output is too noisy`	Add `-c`, protocol filters, or host/port filters to narrow the capture
`Need to inspect later`	Write to a file with `-w capture.pcap` and review it with `tcpdump -r` or Wireshark

Related Guides

Use these guides for broader networking and packet-capture workflows.

tcpdump Command in Linux	Full `tcpdump` guide with detailed examples
ss Command in Linux	Inspect sockets and listening services
ping cheatsheet	Test reachability and latency
IP command cheatsheet	Check interfaces, addresses, and routes
How to Check Open Ports in Linux	Review listening ports before capturing traffic