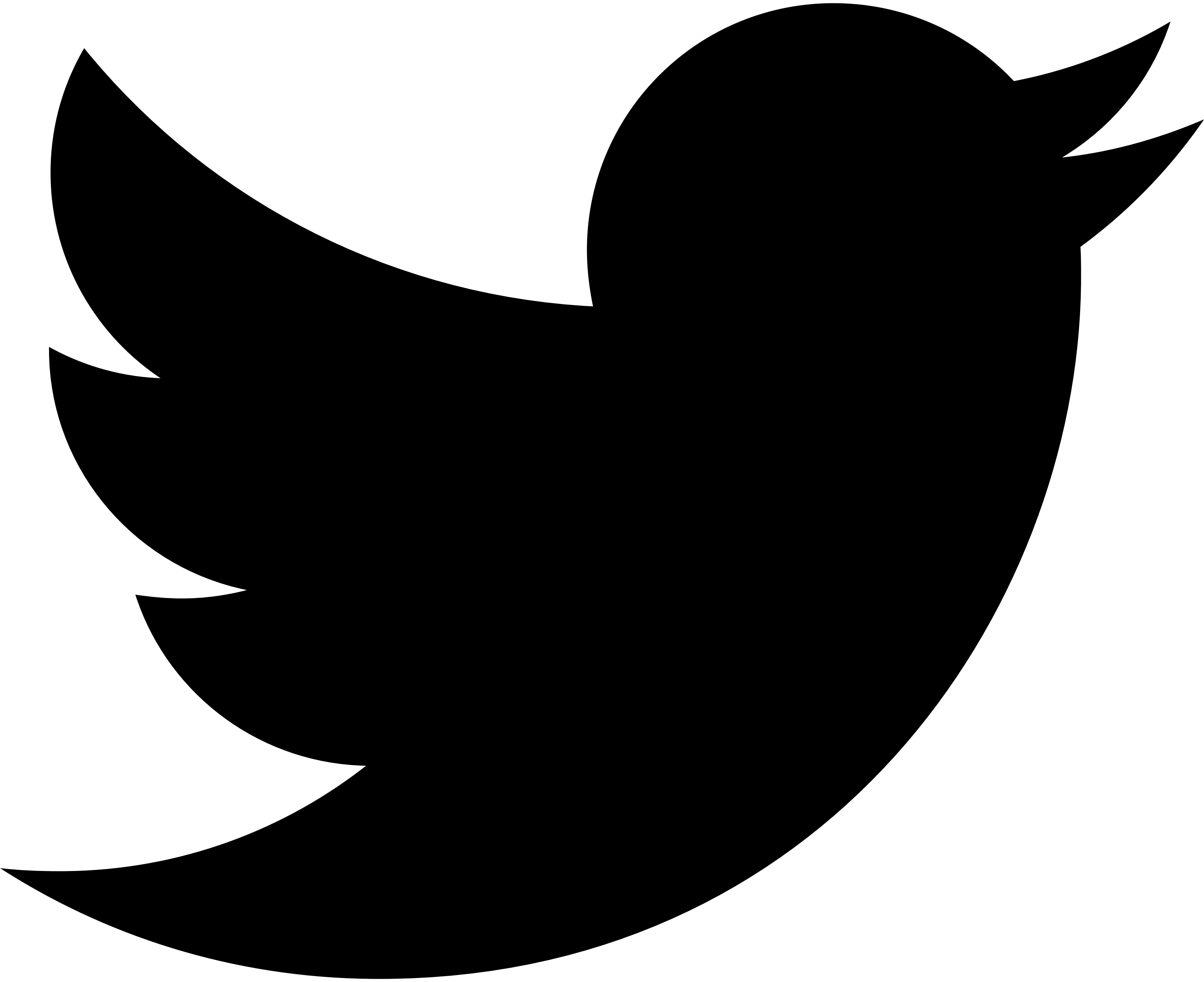
The Ultimate tcpdump Cheat Sheet: Packet Capture Made Easy

February 15, 2023 / By Cassandra Lee



Listen to the article

Have you ever wondered how to monitor network traffic using the command line? Do you want to capture packets automatically when away from your workstation? Or maybe you might want the computer to analyze network data as a scheduled task? Learning to use tcpdump will prove valuable if these sound like what’s on your mind.

This tcpdump cheat sheet will help you get familiar with the appropriate tcpdump filters and commands to use in various situations. We first present the tcpdump switches and commands available, followed by usage examples of each tcpdump command. Getting the syntax right is important if you want to use tcpdump effectively.

Download a PDF copy of this tcpdump cheat sheet for your records **here**, and scroll below to find a list of the common commands in tcpdump.

Search cheats here

What Is TCPDump?

tcpdump is a command-line tool used to capture traffic on the network and analyze captured packets of data passing through your machine.

Its functionality is similar to Wireshark, but it's especially helpful when you can't access a graphical user interface and when automation is essential. Therefore, you can run tcpdump on remote servers or devices on demand or as a scheduled background job as part of an executable script.

Several Linux distributions come pre-loaded with tcpdump; if not, use the distribution's **package manager** to install tcpdump. You can find the location of tcpdump on your operating system with the command `which tcpdump`.

Capture Commands

Use the following commands to capture data packets.

COMMAND	EXAMPLE USAGE	EXPLANATION
`-i any`	`tcpdump -i any`	Capture from all interfaces; may require superuser (`sudo`/`su`)
`-i eth0`	`tcpdump -i eth0`	Capture from the interface `eth0`
`-c count`	`tcpdump -i eth0 -c 5`	Exit after receiving `count (5)` packets
`-r`	`tcpdump -i eth0 -r`	Read and analyze saved capture file

COMMAND	EXAMPLE USAGE	EXPLANATION
`captures.pcap`	`captures.pcap`	`captures.pcap`
`tcp`	`tcpdump -i eth0 tcp`	Show TCP packets only
`udp`	`tcpdump -i eth0 udp`	Show UDP packets only
`icmp`	`tcpdump -i eth0 icmp`	Show ICMP packets only
`ip`	`tcpdump -i eth0 ip`	Show IPv4 packets only
`ip6`	`tcpdump -i eth0 ip6`	Show IPv6 packets only
`arp`	`tcpdump -i eth0 arp`	Show ARP packets only
`rarp`	`tcpdump -i eth0 rarp`	Show RARP packets only
`slip`	`tcpdump -i eth0 slip`	Show SLIP packets only
`-I`	`tcpdump -i eth0 -I`	Set interface as monitor mode
`-K`	`tcpdump -i eth0 -K`	Don't verify checksum
`-p`	`tcpdump -i eth0 -p`	Don't capture in promiscuous mode

Filter Commands

You can add special **filter expressions** to the tcpdump keyword to pick out specific packets. They're especially helpful when you want to analyze saved packet capture files. Each filter expression is a single- or multi-word parameter and its argument, separated by spaces. You may also apply **logical operators** to combine two filter expressions.

In the following examples, we're using `127.0.0.1` as a placeholder for IPv4/IPv6 addresses.

FILTER EXPRESSION	EXPLANATION
`src host 127.0.0.1`	Filter by source IP/hostname `127.0.0.1`
`dst host 127.0.0.1`	Filter by destination IP/hostname `127.0.0.1`
`host 127.0.0.1`	Filter by source or destination = `127.0.0.1`

FILTER EXPRESSION	EXPLANATION
`ether src 01:23:45:AB:CD:EF`	Filter by source MAC `01:23:45:AB:CD:EF`
`ether dst 01:23:45:AB:CD:EF`	Filter by destination MAC `01:23:45:AB:CD:EF`
`ether host 01:23:45:AB:CD:EF`	Filter by source or destination MAC `01:23:45:AB:CD:EF`
`src net 127.0.0.1`	Filter by source network location `127.0.0.1`
`dst net 127.0.0.1`	Filter by destination network location `127.0.0.1`
`net 127.0.0.1`	Filter by source or destination network location `127.0.0.1`
`net 127.0.0.1/24`	Filter by source or destination network location `127.0.0.1` with the tcpdump subnet mask of length `24`
`src port 80`	Filter by source port = 80
`dst port 80`	Filter by destination port = 80
`port 80`	Filter by source or destination port = 80
`src portrange 80-400`	Filter by source port value between 80 and 400
`dst portrange 80-400`	Filter by destination port value between 80 and 400
`portrange 80-400`	Filter by source or destination port value between 80 and 400
`ether broadcast`	Filter for Ethernet broadcasts
`ip broadcast`	Filter for IPv4 broadcasts
`ether multicast`	Filter for Ethernet multicasts
`ip multicast`	Filter for IPv4 multicasts
`ip6 multicast`	Filter for IPv6 multicasts
`ip src host mydevice`	Filter by IPv4 source hostname `mydevice`

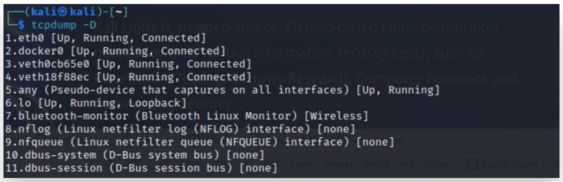
FILTER EXPRESSION	EXPLANATION
`arp dst host mycar`	Filter by ARP destination hostname `mycar`
`rarp src host 127.0.0.1`	Filter by RARP source `127.0.0.1`
`ip6 dst host mywatch`	Filter by IPv6 destination hostname `mywatch`
`tcp dst port 8000`	Filter by destination TCP port = 8000
`udp src portrange 1000-2000`	Filter by source TCP ports in 1000–2000
`sctp port 22`	Filter by source or destination port = 22

For details on how filter expressions work, go to **the tcpdump website**.

Display Commands

These tcpdump switches tell the terminal how to display the output.

COMMAND	EXAMPLE	EXPLANATION
`-A`	`tcpdump -i eth0 -A`	Print each packet (minus its link level header) in ASCII. Handy for capturing web pages. Without `-A` With `-A`
`-D`	`tcpdump -D`	Print the list of the network interfaces available on the system and on which tcpdump can capture

COMMAND	EXAMPLE	EXPLANATION
		packets. 
`-e`	`tcpdump -i eth0 -e`	Print the link-level header on each output line, such as MAC layer addresses for protocols such as Ethernet and IEEE 802.11.
`-F params.conf`	`tcpdump -i eth0 -F /path/to/params.conf`	Use the file `params.conf` as input for the **filter expression**. (Ignore other expressions on the command line.)
`-n`	`tcpdump -i eth0 -n`	Don't convert addresses (i.e., host addresses, port numbers, etc.) to names.
`-S`	`tcpdump -i eth0 -S`	Print absolute, rather than relative, TCP sequence numbers. (Absolute TCP sequence numbers are longer.)
`--time-stamp-precision=tsp`	`tcpdump -i eth0 --time-stamp-precision=nano`	When capturing, set the timestamp precision for the capture to `tsp`: • `micro` for microsecond (default) • `nano` for nanosecond.
`-t`	`tcpdump -i eth0 -t`	Omit the timestamp on each output line.
`-tt`	`tcpdump -i eth0 -tt`	Print the timestamp, as seconds since January 1, 1970, 00:00:00, UTC, and fractions of a second since that time, on each dump line.
`-ttt`	`tcpdump -i eth0 -ttt`	Print a delta (microsecond or nanosecond resolution depending on the `--time-stamp-precision` option) between the current and previous line on each output line. The default is microsecond resolution.

```
┌──(kali㉿kali)-[~]
└─$ tcpdump -D
1.eth0 [Up, Running, Connected]
2.docker0 [Up, Running, Connected]
3.veth0cb65e0 [Up, Running, Connected]
4.veth18f88ec [Up, Running, Connected]
5.any (Pseudo-device that captures on all interfaces) [Up, Running]
6.lo [Up, Running, Loopback]
7.bluetooth-monitor (Bluetooth Linux Monitor) [Wireless]
8.nflog (Linux netfilter log (NFLOG) interface) [none]
9.nfqueue (Linux netfilter queue (NFQUEUE) interface) [none]
10.dbus-system (D-Bus system bus) [none]
11.dbus-session (D-Bus session bus) [none]
```

COMMAND	EXAMPLE	EXPLANATION
`-tttt`	`tcpdump -i eth0 -tttt`	Print a timestamp as hours, minutes, seconds, and fractions of a second since midnight, preceded by the date, on each dump line.
`-ttttt`	`tcpdump -i eth0 -ttttt`	Print a delta (microsecond or nanosecond resolution depending on the `--time-stamp-precision` option) between the current and first line on each dump line. The default is microsecond resolution.
`-u`	`tcpdump -i eth0 -u`	Print undecoded network file system (NFS) handles.
`-v`	`tcpdump -i eth0 -v`	Produce verbose output. When writing to a file (`-w` option) and at the same time not reading from a file (`-r` option), report to standard error, once per second, the number of packets captured.
`-vv`	`tcpdump -i eth0 -vv`	Additional verbose output than `-v`
`-vvv`	`tcpdump -i eth0 -vvv`	Additional verbose output than `-vv`
`-x`	`tcpdump -i eth0 -x`	Print the headers and data of each packet (minus its link level header) in hex.
`-xx`	`tcpdump -i eth0 -xx`	Print the headers and data of each packet, including its link level header, in hex.
`-X`	`tcpdump -i eth0 -X`	Print the headers and data of each packet (minus its link level header) in hex and ASCII.
`-XX`	`tcpdump -i eth0 -XX`	Print the headers and data of each packet, including its link level header, in hex and ASCII.

Output Commands

Customize your tcpdump output with the following commands.

COMMAND	EXAMPLE	EXPLANATION
`-w captures.pcap`	`tcpdump -i eth0 -w captures.pcap`	Output capture to a file `captures.pcap`
`-d`	`tcpdump -i eth0 -d`	Display human-readable form in standard output
`-L`	`tcpdump -i eth0 -L`	Display data link types for the interface
`-q`	`tcpdump -i eth0 -q`	Quick/quiet output. Print less protocol information, so output lines are shorter.
`-U`	`tcpdump -i eth0 -U -w out.pcap`	**Without -w option** Print a description of each packet's contents. **With -w option** Write each packet to the output file `out.pcap` in real time rather than only when the output buffer fills.

Miscellaneous Commands

The following commands don't fall into the categories above.

Here are logical operators that tcpdump uses, with `127.0.0.1` as a placeholder for IPv4/IPv6 addresses:

OPERATOR	SYNTAX	EXAMPLE	DESCRIPTION
AND	`and, &&`	`tcpdump -n src 127.0.0.1 and dst port 21`	Combine filtering options joined by “and”
		`src port 22`	“or”
EXCEPT	`not, !`	`tcpdump dst 127.0.0.1 and not icmp`	Negate the condition prefixed by “not”
LESS	`less, <, (<=)`	`tcpdump dst host 127.0.0.1 and less 128`	Shows packets shorter than (or equal to) 128 bytes in length. < only applies to length 32, i.e., `<32`.

OPERATOR	SYNTAX	EXAMPLE	DESCRIPTION
GREATER	greater, >, (>=)	`tcpdump dst host 127.0.0.1 and greater 64`	Shows packets longer than (or equal to) 64 bytes in length. > only applies to length 32, i.e., `>32`.
EQUAL	=, ==	`tcpdump host 127.0.0.1 = 0`	Show packets with zero length

Example Usage

In the examples below, we craft specific commands by combining tcpdump switches and tcpdump filters.

EXAMPLE	EXPLANATION
`tcpdump -r outfile.pcap src host 10.0.2.15`	Print all packets in the file `outfile.pcap` coming from the host with IP address 10.0.2.15
`tcpdump -i any ip and not tcp port 80`	Listen for non-HTTP packets (which have TCP port number 80) on any network interface
`tcpdump -i eth0 -n >32 -w pv01.pcap -c 30`	Save 30 packets of length exceeding 32 bytes to `captures.pcap` without DNS resolution on the `eth0` network interface
	Capture ICMP traffic and print ICMP packets in hex and ASCII and the following features: With: • headers Without: • link level headers • timestamps.
`tcpdump 'tcp port 80 and (((ip[2:2] - ((ip[0]&0xf)<<2)) - ((tcp[12]&0xf0)>>2)) != 0)'`	Print all IPv4 HTTP packets to and from port 80, i.e. print only packets that contain data, not, for example, SYN and FIN packets and ACK-only packets.

00:00 00:00 1

Conclusion

We hope this tcpdump cheat sheet has been a handy guide in your studies and work. Remember to check out our **networking courses** and **articles on networking**.

The Complete Cyber Security Course! Volume 2 : Network Security

4.8 ★★★★★

00:00 00:00 1

Linux Network Administration

4.9 ★★★★★

Frequently Asked Questions

⊖ How to read the tcpdump output?

“Read” can mean reading from a file and interpreting the on-screen output. To read tcpdump output from a file **captures.pcap**, use **tcpdump -r /path/to/captures.pcap**.

Interpreting output is different: each tcpdump line begins with a timestamp by default, and you need to know the **protocol** associated with the packet to understand the remainder.

⊕ How do I capture only five packets using tcpdump?

00:00 00:00 1

⊕ How do I decode a packet capture?

⊕ How do you analyze tcpdump output in Wireshark?

⊕ What is the difference between tcpdump and Wireshark?

FIND OUT MORE

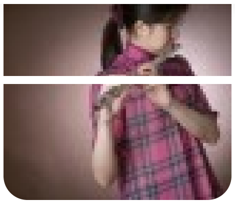
CATEGORIES

CHEAT SHEETS

NETWORKING

00:00

00:00 1



Cassandra Lee

I make connections across disciplines: cyber security, writing/journalism, art/design, music, mathematics, technology, education, psychology, and more. I've been advocating for girls and women in STEM since the 2010s, having written for Huffington Post, International Mathematical Olympiad 2016, and Ada Lovelace Day, and I'm honored to join StationX. You can find me on **LinkedIn** and **Linktree**.

Related Articles

Nmap Cheat Sheet 2023: All the Commands, Flags & Switches

Read More »

Linux Command Line Cheat Sheet: All the Commands You Need

Read More »

Wireshark Cheat Sheet: All the Commands, Filters & Syntax

Read More »

Common Ports Cheat Sheet: The Ultimate Ports & Protocols List

Read More »

00:00 00:00 1

00:00 00:00 1