tcpdump Packet capture

START HERE

sudo tcpdump -D
List interfaces

sudo tcpdump -c 20
Stop after 20 packets

sudo tcpdump -i *<iface>* -nn
Live capture, no lookups

sudo tcpdump -A
Show payload as text

sudo tcpdump -i any port 53
Filter by port

sudo tcpdump -w *<file>*.pcap
Save a capture

sudo tcpdump host *<ip>*
Traffic to or from a host

tcpdump -r *<file>*.pcap
Read it back

Interface -i, -Q → `filter` **BPF filter** expression → `-s, -c` **Capture** kernel buffer → `-w` / `print` **Output** terminal or pcap

CONCEPTS

Options go first, then the filter. Quote the filter so the shell leaves parentheses alone.

The filter runs in the kernel, so a precise filter drops unwanted packets early and loses fewer of the ones you want.

Capturing needs root or the capture capability. Use -Z <user> to drop privileges once the device is open.

CAPTURE

tcpdump -D
List capture interfaces

tcpdump -i *<iface>*
Capture on one interface

tcpdump -i any
Capture on all interfaces
The any pseudo-interface is not promiscuous and uses a cooked link header, so -e shows no real MAC addresses.

tcpdump -c *<n>*
Stop after n packets

tcpdump -s 0
Capture full packets
Modern versions default to 262144 bytes, which is enough for normal frames.

tcpdump -s *<bytes>*
Limit bytes kept per packet

tcpdump -Q in
Only inbound packets (in, out, inout)

tcpdump -p
No promiscuous mode

tcpdump -B *<kib>*
Raise the capture buffer

tcpdump --immediate-mode
Deliver packets without buffering

tcpdump -I
Wi-Fi monitor mode

OUTPUT

tcpdump -n
No host name lookups

tcpdump -nn
No host or port name lookups

tcpdump -v
More protocol detail (-vv, -vvv more)

OUTPUT (cont.)

tcpdump -q
Short, quiet lines

tcpdump -A
Payload as ASCII

tcpdump -X
Payload as hex and ASCII

tcpdump -xx
Hex dump with link header

tcpdump -e
Show link-level (MAC) header

tcpdump -tttt
Readable date and time

tcpdump -ttt
Time since previous packet

tcpdump -t
No timestamps

tcpdump -S
Absolute TCP sequence numbers

tcpdump -#
Number each packet line

tcpdump -l
Line-buffer stdout for pipes
Use with tee or grep, for example: tcpdump -l | tee out.txt.

tcpdump --lengths
Show captured and original length

SAVE AND READ

tcpdump -w *<file>*.pcap
Write raw packets to a file

tcpdump -r *<file>*.pcap
Read a saved capture

tcpdump -w *<file>*.pcap --print
Save and also print

tcpdump -U -w *<file>*.pcap
Flush each packet to the file

tcpdump -C *<size>* -w *<file>*
Rotate by size (millions of bytes)

tcpdump -W *<n>*
Keep at most n files (ring buffer)

tcpdump -G *<sec>* -w *<file>*-%FT%T.pcap
Rotate by time

tcpdump -z gzip
Run a command on each closed file

tcpdump -Z *<user>*
Drop root after opening device

tcpdump --count -r *<file>*
Only print packet count

SAVE AND READ (cont.)

tcpdump -F *<file>*
Read the filter from a file

tcpdump -w - | *<cmd>*
Write the pcap to stdout

FILTERS: HOSTS AND PORTS

host *<ip>*
Traffic to or from a host

src host *<ip>*
Only from this host

dst host *<ip>*
Only to this host

net *<cidr>*
A whole network, e.g. 10.0.0.0/8

port *<n>*
Source or destination port

src port *<n>*
Source port only

dst port *<n>*
Destination port only

portrange *<a>*-*<b>*
A range of ports

ether host *<mac>*
A MAC address

vlan *<id>*
Frames tagged with a VLAN

gateway *<host>*
Packets routed via a gateway

FILTERS: PROTOCOLS AND LOGIC

tcp
TCP only (also udp, icmp)

icmp
ICMP, e.g. ping

udp
UDP only

arp
ARP requests and replies

ip6
IPv6 only

ip proto *<name|n>*
By IP protocol number

A and B
Both must match

A or B
Either matches

not A
Negate (also !)

'(A or B) and C'
Group with quoted parentheses

FILTERS: PROTOCOLS AND LOGIC (cont.)

less *<n>*
Packets of n bytes or fewer

greater *<n>*
Packets of n bytes or more

ether broadcast
Link-layer broadcasts

ip multicast
IPv4 multicast

TCP DETAILS

'tcp[tcpflags] & tcp-syn != 0'
Packets with SYN set

'tcp[tcpflags] & tcp-rst != 0'
Resets

'tcp[tcpflags] & (tcp-syn|tcp-fin) != 0'
Connection start or end

'tcp[tcpflags] == tcp-syn'
SYN only, no ACK

'tcp[tcpflags] & tcp-push != 0'
Packets with PSH set

Reading a line

Flags	Meaning
[S]	SYN, connection start
[S.]	SYN-ACK
[.]	ACK only
[P.]	data pushed
[F.]	FIN, closing
[R]	RST, refused or reset

SIGNALS AND DEBUG

tcpdump -K
Skip checksum verification

RECIPES

Watch DNS

```
sudo tcpdump -i any -nn port 53
```

HTTP requests as text

```
sudo tcpdump -i <iface> -nn -A -s 0 'tcp dst port 80'
```

Capture to a ring of files

```
sudo tcpdump -i <iface> -nn -w cap -C 100 -W 10 -Z $USER
# keeps ten files of about 100 MB each
```

Only new connections to a host

```
sudo tcpdump -nn 'dst host <ip> and tcp[tcpflags] == tcp-syn'
# add -w syn.pcap to save it for Wireshark
```

Quiet ping debugging

```
sudo tcpdump -i <iface> -nn -tttt icmp and host <ip>
```