



like this?
there are more
zines at:
<http://jvns.ca/zines>

what's this?

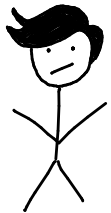
The man page for tcpdump starts like this:

NAME

tcpdump - dump traffic on a network

SYNOPSIS

```
tcpdump [ -AbDefhHIJKLlnNOpqStuUvxX# ] [ -B buffer_size ]  
[ -c count ]  
[ -C file_size ] [ -G rotate_seconds ] [ -F file ]  
[ -i interface ] [ -j tstamp_type ] [ -m module ] [ -M secret ]  
[ --number ] [ -Q in|out|inout ]  
[ -r file ] [ -V file ] [ -s snaplen ] [ -T type ] [ -w file ]  
[ -W filecount ]  
[ -E spi@ipaddr algo:secret,... ]  
[ -y datalinktype ] [ -z postrotate-command ] [ -Z user ]  
[ --time-stamp-precision=tstamp_precision ]  
[ --immediate-mode ] [ --version ]  
[ expression ]
```



that is SO MANY
options omg

it's ok! you
only need to
know like 3!



I'm going to tell you
why I ♥ tcpdump and
how to get started!

JULIA EVANS
@b0rk
<http://jvns.ca>
my blog!

nmap

in ur network
scanning ur ports

telnet

See if a port on
another server
is open

whois

look up a
domain

ssh

can't forget
this one ☺

lsof

what ports
are being used?

sysctl

Configure socket
buffer sizes, and more!

network manager

GUI tool to configure
the network on your
laptop

nethogs / ab / nload
iptraf / netperf / iperf
iftop / netsniff-ng

lots of performance /
benchmarking tools
(they all do different things)

ping

ping, but it
uses TCP

OpenVPN

Set up a
VPN!

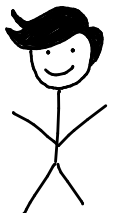
socat

like netcat,
but more
featureful



thanks so much
for reading!

now that I
understand the
basics, the man
page isn't so bad!



network administration tools

Finally, there are a lot more tools than tcpdump! We won't explain them here but here's a list!

ping
"are these computers even connected?"
dig/nslookup
"does that domain exist?"
netstat/ss
"am I using that port?"

ip
configures interfaces, routes, and more. Successor to ifconfig.
arp
see your ARP table!

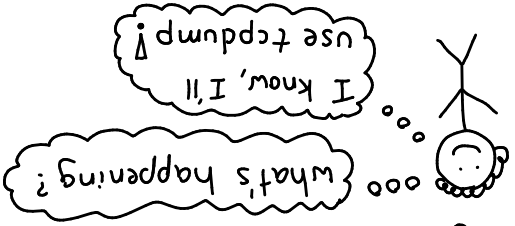
trace route/mtr
what servers are on the way to that server?
nc
Make tcp connections manually!

iptables/nftables
set up firewalls and NAT!
sysctl
Configure socket buffer sizes, and more!
ethtool
understand your ethernet connections

What is tcpdump for?

tcpdump captures network traffic and prints it out for you.

For example! Yesterday DNS lookups on my laptop were slow



\$ sudo tcpdump -n -i any port 53

DNS queries
10:52:03.992138 IP 192.168.1.241.63019 → 192.168.1.1.53: 44000+ A? ask,metafilter.com. (36)
10:52:08.972719 IP 192.168.1.241.63019 → 192.168.1.1.53: 44000+ A? ask,metafilter.com. (36)
10:52:13.919782 IP 192.168.1.241.63019 → 192.168.1.1.53: 44000+ A? ask,metafilter.com. (36)
10:52:13.928894 IP 192.168.1.1.53 > 192.168.1.241.63019: 44000 2/0/0 CNAME metafilter.com., A 54.186.13.33 (80)

DNS response

This means that there were 3 DNS queries (at 10:52:03, 10:52:08, 10:52:13), but only the 3rd one got a response!

I figured my router was probably the problem, I restarted it, and my internet was fast again!

Let's learn how to debug problems with tcpdump!

Questions you can answer with tcpdump

→ what DNS queries is my laptop sending?

"tcpdump -i any port 53"

→ I have a server running on port 1337.
Are any packets arriving at that port at ALL???

"tcpdump -i any port 1337"

→ What packets are coming into my server from IP 1.2.3.4?

"tcpdump port 1337 and host 1.2.3.4"

→ show me all DNS queries that fail

"tcpdump udp[11] & 0xf == 3"

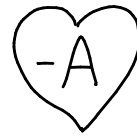
(complicated but it works!)

→ how long are the TCP connections on this box lasting right now?

"tcpdump -w packets.pcap"

and analyze packets.pcap in Wireshark

and here are a few more good ones:




This prints out the packet's contents!
For example, suppose I have a webserver on port 7777.

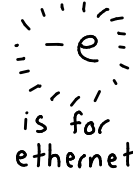
\$ sudo tcpdump -A dest port 7777

will show me all the HTTP requests being sent to that server. Only works for HTTP, not HTTPS.

(I like ngrep more than tcpdump -A for looking at HTTP request bodies though ☺)



By default, tcpdump will translate IP addresses to hostnames.  forces it to just always print out the IP address



Includes Ethernet information! This shows you the MAC address that the packet came from

Example: sudo tcpdump -e -i any port 443



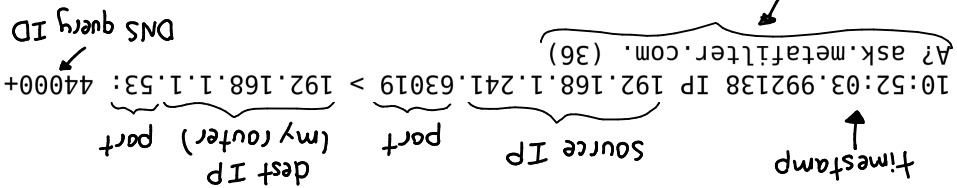
makes sure you only get packets that are to or from your computer

Every line of tcpdump output represents a packet.

- ✖ source + dest IP address and port

- * which TCP flags (good for spotting the beginning of a TCP connection)

* that's it!



TCP packet:

TCP packet:

```
11:36:26.353797 IP 192.168.1.241.45296 > 192.241.182.146.443: Flags [.,],  
ack 2291349910, win 319, options [nop,nop,TS val 10967552 ecr 580196754],  
length 0
```

↑
TCP flags
"." means Ack

Ever seen a "Connection refused" error? Here's what that looks like in tcpdump!

looks like in tcpdump!

```
12:16:38.944390 IP6 localhost.48680 > localhost.8999: Flags [S]
12:16:38.944458 IP6 localhost.8999 > localhost.48680: Flags [R.]
```

We sent a SYN to open the connection but the server replied with a "RST" packet. That gets translated to "connection refused".

command line arguments

I use these 3 arguments the most:

Which network interface to capture packets on. I often use `-i any`. The default interface tcpdump picks isn't always what you want.

Example: `sudo tcpdump -i lo` shows you packets on the local

Instead of printing out packets, write them to a file! This is VERY USEFUL for analyzing the packets later. I use it all the time

Example: `sudo tcpdump host 8.8.8.8 -w my-packets.pcap`

When writing to a file, be careful! You don't want to accidentally fill up your hard drive. [-c 10000] will only

```
Example: sudo tcpdump -c 1000
-w my-packets.pcap
dest port 8080
```



! is far
! interface



!s for write

write



is for
count

count

BPF filters!

tcpdump uses a small language called BPF to let you filter packets.

When you run `$ sudo tcpdump port 53`, "port 53" is a BPF filter. Here's a quick guide!

→ port 53

checks if the source port OR the dest port is 53. Matches TCP port 53 and UDP port 53.

→ host 192.168.3.2

checks if the source or dest IP is 192.168.3.2

→ host 11.22.33.44
and port 80

you can use 'and', 'or', and 'not'

→ src port 80

→ dest port 80

→ tcp port 80

are what they look like ☺
so are
src host 1.2.3.4
dest host 1.2.3.4

→ `udp[11] & 0xf == 3`

you can do bit math like this on packet contents.

This checks for the DNS response code "NXDOMAIN"!

(I googled to find this and it works! ☺)

♥ Wireshark ♥



I want to know more about what's in my packets!

you want Wireshark?

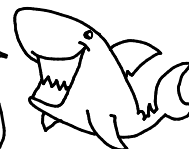


Wireshark is an incredibly powerful packet analysis tool!



what protocols do you understand, Wireshark?

HTTP! TCP!
DNS!
ARP! IP!
MSN! AIM! AOL!
Ethernet! Bluetooth!
A lot, okay?



Things Wireshark has:

- ★ nice graphical interface!
- ★ it can connect TCP packets from the same connection!

- ★ search through your packets easily!

If you want to analyze packets from tcpdump with Wireshark, you can either:

- ① save a .pcap file and open it with Wireshark
- ② use this incantation to pipe tcpdump output into Wireshark!

```
ssh some.remote.host tcpdump -pni any -w - -s0 -U port 8888  
| wireshark -k -i -
```