CYBRARY

TCPDump Cheat Sheet

Contents

TCPDump

TCPDump Overview

Name: tcpdump – dump traffic on a network.

Here a few options you can use when using tcpdump.

Using these options, we will try to build some simple use cases.

Options

-i any : Listen on all interfaces just to see if you're seeing any traffic.

-i eth0 : Listen on the eth0 interface.

-D : Show the list of available interfaces

-n : Don't resolve hostnames.

-nn : Don't resolve hostnames or port names.

-q : Be less verbose (more quiet) with your output.

-t : Give human-readable timestamp output.

-tttt : Give maximally human-readable timestamp output.

-X : Show the packet’s contents in both hex and ASCII.

-XX : Same as -X, but also shows the ethernet header.

-v, -vv, -vvv : Increase the amount of packet information you get back.

-c : Only get x number of packets and then stop.

-s : Define the size of the capture in bytes. Use -s0 to get everything, unless you are intentionally capturing less.

-S : Print absolute sequence numbers.

-e : Get the ethernet header as well.

-q : Show less protocol information.

-E : Decrypt IPSEC traffic by providing an encryption key.

Now, a brief excerpt about expressions, that allows you to trim out various types of traffic and find exactly what you're looking for.

There are three main types of expression: ***type, dir, and proto*** .

Type options are: host, net, and port.

Direction lets you do src, dst, and combinations thereof.

Proto (col) lets you designate: tcp, udp, icmp, ah, and many more.

The Use Cases

Now, let's try using this information in real use cases:

tcpdump -D

Listing possible network interfaces on the system

```
$ tcpdump -D
```

```
1.eth0
```

2.eth1

3.eth2

tcpdump -i interface-name

Capture packets from a particular interface

tcpdump -i eth1

tcpdump -c N

Capture only N number of packets

tcpdump -i eth1 -c 10

tcpdump -w file.pcap

Capture the packets and write into a file

tcpdump -i eth1 -w tmp.pcap

tcpdump -s 0

Capture and store network frames full-length

```
tcpdump -i eth1 -w tmp.pcap -s 0
```

tcpdump -r file.pcap

Reading the packets from a saved file

```
tcpdump -tttt -r tmp.pcap
```

tcpdump -tttt

Capture packets with proper readable timestamp

```
tcpdump -i eth1 -tttt
```

tcpdump greater N

Read packets longer than N bytes

```
tcpdump -i eth1 -w tmp.pcap greater 1024
```

Specify protocol type

To receive only the packets of a specific protocol type – fddi, tr, wlan, ip, ip6, arp,

rarp, decnet, tcp and udp

```
tcpdump -i eth1 arp
```

tcpdump host IP

Will show you traffic from 1.2.3.4, whether it's the source or the destination.

```
tcpdump host 1.2.3.4
```

tcpdump src/dst

Filtering by source and sestination: it's quite easy to isolate traffic based on either source or destination using src and dst.

```
tcpdump src 2.3.4.5
```

```
tcpdump dst 3.4.5.6
```

tcpdump net x.x.x.x/xx

Filter packets by network: you can combine this with the src or dst options as well.

tcpdump net 1.2.3.0/24

tcpdump port PORT_NO

Receive packets flows on a particular port

tcpdump -i eth1 port 22

tcpdump -i eth1 src port 1026

tcpdump less/greater

Filter traffic based on Packet Size: you can use less, greater, or their associated symbols that you would expect from mathematics.

tcpdump -i eth1 less 32

tcpdump -i eth1 greater 64

tcpdump -i eth1 <= 128

tcpdump dst IPADDRESS and port PORT-NO

Capture packets for particular destination IP and Port

```
tcpdump -i eth1 dst 10.181.140.216 and port 22
```

tcpdump -vvv

Display more packet information

E.g. `tcpdump -i eth1 -vvv`

tcpdump -e

Display link level header of every packet: -e

```
tcpdump -i eth1 -e -t
```

```
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes
```

```
52:54:00:e1:1c:10 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3, length 60: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl 0x03 : STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003, length 43
```

```
52:54:00:e1:1c:10 (oui Unknown) > 01:80:c2:00:00:00 (oui Unknown), 802.3,
length 60: LLC, dsap STP (0x42) Individual, ssap STP (0x42) Command, ctrl
0x03 : STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003,
length 43
```

tcpdump -t

Don’t print a timestamp on each dump lin: without using **-t** option we can see the below output timestamp is dumped.

```
tcpdump -i eth2
```

```
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes

08:44:51.295229 STP 802.1d, Config, Flags [none], bridge-id
8000.52:54:00:e1:1c:10.8003, length 43

08:44:53.296795 STP 802.1d, Config, Flags [none], bridge-id
8000.52:54:00:e1:1c:10.8003, length 43
```

and with **-t** option:

```
tcpdump -i eth2 -t
```

```
listening on eth2, link-type EN10MB (Ethernet), capture size 65535 bytes

STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003, length 43

STP 802.1d, Config, Flags [none], bridge-id 8000.52:54:00:e1:1c:10.8003, length 43
```

tcpdump -n

Display packets with IP address instead of DNS names: -nBasically tcpdump converts the plain address to DNS names. Using n option we can make tcpdump to display ip address.

```
tcpdump -i eth1 -n
```

tcpdump -A

Display Captured Packets in **ASCII**

```
tcpdump -i eth1 -A
```

tcpdump -XX

Display Captured Packets in **HEX** and **ASCII**

```
tcpdump -i eth1 -XX
```

```
tcpdump -nnvXSs 0 -c1 icmp
```

Hex output : useful when you want to see the content of the packets in question, and it’s often best used when you’re isolating a few candidates for closer scrutiny.

Some everyday examples

tcpdump can output content in **ASCII** , so you can use it to search for cleartext content using other command-line tools like grep.

The **-l** switch lets you see the traffic as you’re capturing it and helps when sending to commands like grep.

Find HTTP User Agents

```
tcpdump -vvAls0 | grep 'User-Agent:'
```

Cleartext GET Requests

```
tcpdump -vvAls0 | grep 'GET'
```

Find HTTP Host Headers

```
tcpdump -vvAls0 | grep 'Host:'
```

Find HTTP Cookies

```
tcpdump -vvAls0 | grep 'Set-Cookie|Host:|Cookie:'
```

Find SSH Connections

This one works regardless of what port the connection comes in on, because it’s getting the banner response.

```
tcpdump 'tcp[(tcp[12]>>2):4] = 0x5353482D'
```

Find DNS Traffic

```
tcpdump -vvAs0 port 53
```

Find FTP Traffic

```
tcpdump -vvAs0 port ftp or ftp-data
```

Find NTP Traffic

```
tcpdump -vvAs0 port 123
```

Find Cleartext Passwords

```
tcpdump port http or port ftp or port smtp or port imap or port pop3 or port telnet -lA
| egrep -i -B5
'pass=|pwd=|log=|login=|user=|username=|pw=|passw=|passwd=|password=|pass:|
user:|username:|password:|login:|pass |user '
```