tcpdump cheat sheet

Tcpdump is a commandline tool that is used to dump traffic on a network. This tool comes in hand when you want to analyse network captures within the command line. Basically it can do most of the wireshark job.

Additional Note & Reference

To be fair: This gist is itself a fork I created some time ago, but the original gist or author seems to not exist anymore, and it looks like that I'm now in the lead ;-) Please see the revision history for details.

Furthermore some more and basic advanced examples may be of interest (thanks to twitter://@howtouselinux1):

- Tcpdump Cheat Sheet (Basic Advanced Examples)

Options

The following are some of options that I prefer when using `tcpdump` for my daily use.

tcpdump [OPTIONS]

```
-i any : Listen to all the interfaces
-i virbr0: Listen to a specific interface virbr0
-D: Show the list of available interface
-n: Don't resolve the hostnames
-nn: Don't resolve hostnames or port names.
-q: quite output
-t: Don't print a timestamp on each dump line.
-tttt: Give maximally human-readbale timestamp output
-X: Show the packet's contents in both HEX ad ASCII
-XX: Same as -X but shows the ethernet header.
-v, -vv, -vvv: Being more verbose(increase number of packet information)
-c: Only capture number of packets and stop
-s: Define the snaplength(size) of the capture in bytes. Use -s0 to get
everything.
-S: Print absolute sequence numbers.
-e: Get the ethernet header as well
-E: Decrypt IPSEC traffic by providing an encryption key.
```

Expressions

`tcpdump` allow us to use expression so we can narrow down our solution to get exactly what we're looking for.

There are 3 types of expression: `type`, `dir` and `proto`

- Type options are: `host`, `net`, and `port`
- Direction are: `src` and `dst`
- Protocol : `tcp`, `udp`, `icmp`, `ah` etc

Examples

1. Basic communication to see what happens on the network

```
$ tcpdump -i any
```

2. Monitor specific interface

```
$ tcpdump -i virbr0
```

3. Raw output view with verbose output,no host/port resolution,absolute sequence number and human-readable timestamps.

```
$ tcpdump -ttttnnvvS
```

4. Find traffic by IP

```
$ tcpdump host 192.168.122.131
```

5. Seeing packets with HEX output

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

6. Filtering by Source and Destination

```
$ tcpdump src 192.168.122.131
$ tcpdump dst 192.168.122.14
```

7. Finding packets by network

```
$ tcpdump net 192.168.122.0/24
```

8. Show traffic related to a specific port

```
$ tcpdump port 3389
```

9. Show traffic of one protocol

```
$ tcpdump icmp
```

10. Show only IPv6 Traffic

```
$ tcpdump ip6
```

11. Find traffic using Port ranges

```
$ tcpdump portrange 21-25
```

12. Find traffic base on packet size

```
$ tcpdump less 32


$ tcpdump greater 32


$ tcpdump <= 102
```

13. Writing captures to a file

```
$ tcpdump port 80 -w output
```

14. Reading from pcap files

```
$ tcpdump -r output.pcap
```

More Examples

1. Options Combination
   - AND : `and` or `&&`
   - OR : `or` or `||`
   - EXCEPT : `not` or `!`

```
$ tcpdump -nnvvS src 192.168.122.1 and dst port 4444
```

2. Complex grouping and special characters For complex grouping we use `()` to specify our options

```
$ tcpdump 'src 192.168.122.84 and (dst port 4444 or 22)'
```

3. Isolating Specific TCP Flags. The filter `tcp[13]` look at offset 13 in `TCP HEADER`,hence the number represent the location within the byte, while the `!=0` means that the flag is set to 1.
   Show all URGENT (URG) packets\

```
$ tcpdump 'tcp[13] & 32!=0'
```

   Show all ACKNOWLEDGE(ACK) packets\

```
$ tcpdump 'tcp[13] & 16!=0'
```

Show all PUSH (PSH) packets\

```
$ tcpdump 'tcp[13] & 8!=0'
```

Show all RESET (RST packets\

```
$ tcpdump 'tcp[13] & 4!=0'
```

Show all SYNCHRONIZE (SYN) packets\

```
$ tcpdump 'tcp[13] & 2!=0'
```

Show all FINISH (FIN) packets\

```
$ tcpdump 'tcp[13] & 1!=0'
```

Show all SYNCRONIZE/ACKNOWLEDGE (SYNACK) packets\

```
$ tcpdump 'tcp[13]=18'
```

Alternative we could also use `tcpflags` syntax

```
$ tcpdump 'tcp[tcpflags] == tcp-syn'
$ tcpdump 'tcp[tcpflags] == tcp-rst'
$ tcpdump 'tcp[tcpflags] == tcp-fin'
```

4. Identifying malformed/malicious packets.
   - Packets with both rst and syn flags shouldn't be the case.

     ```
     $ tcpdump 'tcp=[13] = 6'
     ```

   - Find cleartext http get requests

     ```
     $ tcpdump 'tcp[32:4] = 0x47455420'
     ```

   - Find ssh connection on any port via (banner text)

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

Revision #2
Created 2025-09-21 10:39:10 UTC by magenbrot
Updated 2025-09-21 13:30:25 UTC by magenbrot