Packet Capturing - Cheatsheet

- Wireshark
- TCPDump

Wireshark

Filter where the source ip is not 192.168.1.1

```
ip.src != 192.168.1.1
```

Filter where the destination ip is not 192.168.1.1

```
ip.dst != 192.168.1.1
```

Find packets with a string in them

```
frame contains <thing to search>
```

For example:

```
frame contains google
```

Resource: https://www.cellstream.com/reference-reading/tipsandtricks/431-finding-text-strings-in-wireshark-captures

Show hostnames

Go to View -> Name Resolution -> Check the box next to Resolve Network Addresses

Resource: https://unix.stackexchange.com/questions/390852/how-to-filter-by-host-name-in-wireshark

Filter TLS traffic

```
ssl.record.version
```

If you want to only show TLS v1.2 traffic, then you would run:

```
ssl.record.version == 0x0303
```

Versions:

0x0300 SSL 3.0
0x0301 TLS 1.0
0x0302 TLS 1.1
0x0303 TLS 1.2

Resource: https://security.stackexchange.com/questions/190532/filter-tls-in-wireshark-or-other-monitoring-tool

TCPDump

Filter on port 80

```
tcpdump port 80
```

Filter on source port 80

```
tcpdump src port 80
```

Destination port 80

```
tcpdump dest port 80
```

All traffic for 192.168.1.1

```
tcpdump host 192.168.1.1
```

Save output

```
tcpdump tcp -w output.pcap
```

Resource: https://medium.com/swlh/introduction-to-tcpdump-635653f56177

Filter on service

In this case, we are filtering icmp traffic on the eth0 interface where the ICMP type field value is icmp-echo. We finish it with a full protocol decode (-vv) aka verbose output.

```
tcpdump -i eth0 icmp and icmp[icmptype]=icmp-echo -vv
```

Resources: http://alumni.cs.ucr.edu/~marios/ethereal-tcpdump.pdf

http://www.networksorcery.com/enp/protocol/icmp/msg8.htm

Listen for traffic over port 389

```
tcpdump -i eth0 -nn port 389
```

Resource: https://hackertarget.com/tcpdump-examples/