

WIRESHARK EXERCISE

1. Open Wireshark and start capturing Ethernet traffic.

There are various filters possible in Wireshark

2. Show the first N packets captured: As an example filter `frame.number <= 2` shows the first two.

3. By double-clicking on a frame, you can observe the content as ASCII.

4. By double-clicking on a frame, you can observe the packets also in HEX format.

5. After you have stopped capturing packets, you will be able to save the capture. Use 'File > Save As' for saving all capture data and 'File > Export Specified Packets' to save packets with filters applied.

6. You can later load a saved file to Wireshark interface.

7. Find packets with certain IP address using `ip.addr == x.x.x.x`. To match only the destination addresses, use `ip.dst == x.x.x.x` and to match the source addresses, `ip.src == x.x.x.x`.

8. Through 'View' > 'Time Display Format' you can designate the time format you want to see. The default is set to 'Seconds since Beginning of Capture'.

9. Find packets longer than N bytes. Example filter is `frame.len > 2048`

10. Similarly, you can find packets less than N bytes. `frame.len < 1024`

11. To find packets using only the specific protocols use protocol name in filter box. For example, for ARP simply type `arp` to filter box.

12. To examine all the packets going through a specific port use `tcp.port == X` for TCP ports and similarly `udp.port == X` for UDP.

13. An important thing is to be able to use multiple filters at the same time. You can use any Boolean logic to accomplish this with combinations of `&&` and `||` and `!`. For example `ip.dst == 192.168.0.15 && tcp.port == 22` filter finds packets that flow to IP address 192.168.0.15 through port number 22.

14. You can filter out packets you don't want. A filter is given as: `!arp && !tcp`
What does this filter accomplish?

15. Now filter out all UDP packets going out from your IP address. You need to find your own IP address.