Project 1

Steps to compile and run the program

1. As we’re using command line input there is no need to run and debug the program.
2. To simply run the program without any filters –
   1. python3 pktsniffer.py filename.pcap
3. To check the whether the pcap file has packets with host of with a certain IP address
   1. python3 pktsniffer.py filename.pcap host 192.168.0.0
4. To check whether the packets either have a specific source or destination port number–
   1. python3 pktsniffer.py filename.pcap port 433
5. To check whether the packets either have a specific source or destination IP Address-
   1. python3 pktsniffer.py filename.pcap ip 192.168.0.1
6. To check whether the packets belong to a certain protocol (example – TCP, UDP, ICMP)-
   1. python3 pktsniffer.py tcp
   2. python3 pktsniffer.py udp
   3. python3 pktsniffer.py icmp
7. To check whether belong to a certain subnet –
   1. python3 pktsniffer.py filename.pcap net 192.168.0.0/24
8. No more than 1 filter can be applied at a single time.



Libraries

As the program use of libraries if not present in your system it will give an error.

List of libraries required –

1. scapy
2. ipaddress

Note – These libraries were not used to parse the pcap file but to filter and read it.

Filters

Each filter when applied processes the pcap file and removes all the packets that don’t pass the filter. After this the new set of packets is sent to the parser to display the contents.

Once a filter is applied, it does not stay on for the next iteration. For example, if host filter is applied and after the result is displayed, a protocol filter is applied, the program wont filter from the first iteration of packets. It’ll again read the entire pcap file and only filter the protocol mentioned.

Restrictions

The program will not parse any IPv6 packets. The program will not be able to parse any protocol other than TCP, UDP and ICMP.

Output

The output will be in the following format - Text

Description automatically generated