WIRETAP

By Shivaraman Janakiraman
Magesh Khanna Vadivelu

INTRODUCTION:

Our project involves sniffing and analyzing packets being transmitted or received over a LAN. The information obtained can be used in troubleshooting network configuration and reachability.

MODULES:

1) *Start date and time, total duration, and total number of packets in the packet capture:*
   This involves finding out the date and time of the packet capture, duration in which packet is analyzed and the total number of packets which are analyzed in the offline mode.
2) *Unique Ethernet addresses along with total number of packets containing each address:*
   This involves finding out Ethernet address by reading the Ethernet header of the packet where the source and destination addresses are stored. For each source and destination address, the number of packets using it is determined by making use of structures where we store the Ethernet addresses.
3) *Unique Network layer protocols seen, and how many packets use them:*
   This involves finding out the network layer protocols that packet uses which is done by reading the *ether_type* from the Ethernet header and also finding out the number of packets that uses any of the network protocols.
4) *Unique IP addresses along with the total number of packets containing each address:*
   This involves finding out the IP address by reading the IP header of the packet where the source and destination addresses are stored. For each source and destination address, the number of packets using it is determined by making use of structures where we store IP addresses.
5) *Unique Transport layer protocols seen, and how many packets use them:*
   This involves finding out the transport layer protocols that packet uses which is done by reading *ip_proto* from the IP header and also finding out the number of packets that uses any of the transport protocols.
6) *Unique TCP and UDP ports along with total number of packets using them:*
   This involves finding out the TCP and UDP ports that the packets use which is identified by reading the IP header in the packet. The structure that we used for storing the ports contain *th_sport* and *th_dport* and *uh_sport and uh_dport* for TCP and UDP source and destination ports respectively.

7) *Determining the number of UDP packets with a correct checksum,* incorrect checksum, and those that omit checksum calculations:
   This involves finding out the packets that has correct or incorrect or that omits checksum based on the one's complement checksum calculation
8) *Reporting the number of packets containing each flag for TCP:*
   This involves finding out the number of packets that contains each flag for TCP i.e., FIN, SYN, RST, PUSH, ACK, URG, ECE, CWR. These flags are contained in TCP header. This header is analyzed for finding out the number of packets containing each flag.
9) *Finding out Average, minimum, and maximum packet sizes:*
   Here the size of each packet is identified and each of the sizes is compared to get the minimum and maximum size. In order to find the average of the sizes, we sum the sizes of all the packets and divide it by the total count of packets.

Packet Flow Summary:

Flow of packets from source to destination is identifiable irrespective of port number it flows to and fro.

LIMITATIONS:

For the extra credit problem, we were not able to number the packets flow from destination to source and also the implementation is grouped by IP addresses i.e., Port numbers were not taken into consideration.

REFERENCES:

1) http://www.ethereal.com/
2) http://en.wikipedia.org/wiki/Tcphdr
3) http://en.wikipedia.org/wiki/Iphdr
4) Packet Capture With libpcap and other Low Level Network Tricks
5) http://www.tcpdump.org/pcap.htm