Deepak Konidena

Mandeep Mehta

Wiretap is a Network Protocol Analyzer. The dump that is given to us is in the tcpdump format and mainly contains TCP and UDP traffic. Wiretap makes use of the pcap library for processing the input files "traceroute.pcap" and "wget.pcap"

The interesting function is pcap_loop() which processes packet by packet all packets present in the file. The second argument count, when passed as '-1', ensures pcap_loop() returns only incase of any error. The third argument passed to pcap_loop() is a function pointer which in our case is packet_sniffer which inturn takes three arguments, the last of which u_char* rawPacket is of utmost interest to us. rawPacket points to the first byte of the packet which contains Ethernet, IP, TCP/UDP headers and payload. The headers are stripped individually by the structure pointers (defined in /usr/include/netinet and /usr/include/net directories). At each stage the header length is added to the pointer to point to the next header. This is the tricky part. Then we typecast the u_char to the respective structure pointers to access the individual members of the header.

We observed that most of the packet members above the Ethernet layer, had to be accessed in the host byte order as opposed to the original network byte order. We achieved this by using ntohs() and ntohl() as and when appropriate.

We found checksum calculation a bit challenging. For UDP checksum, a pseudo header is constructed on top of actual UDP header and data and checksum is computed over the entire content. Also, if the payload had odd number of octets, we had to pad an extra zero byte to make it even. Checksum calculation was tough in the sense that it involved accessing individual 16-bit words in the host byte order and adding them, with the carries being added at last.

Assumptions:

a) We assumed for overhead calculation , Overhead = (Ethernet + IP + TCP/UDP) / totalpacketsize. We ignored packets which were not IP at network layer and which were not TCP or UDP at transport layer.
b) For checksum, we assumed omitted checksums as those which contained a zero(0x0000) in their udp checksum field, correct checksums as those which resulted in a 0xffff when added along with the pseudo header. Pseudo header contains Ip source, Ip destination, UDP length and UDP protocol number.
c) For total duration of the capture, we calculated the differences between the tv_sec and tv_usec(present in the timeval ts provided by struct pcap_pkthdr) of the first and last packet and displayed it in seconds.

Usage of Wiretap:

./Wiretap <filename>

References

1) **http://www.netfor2.com/udpsum.htm** **(**We modified it for our use)
2) **http://www.tcpdump.org/**