Assignment 5

Network traffic monitoring using the Packet Capture library

In this assignment, you will get familiar with the Packet Capture library (libpcap). This tutorial assumes background knowledge in networks and C programming language. For more information about the packet capture library, visit the following websites: https://linux.die.net/man/3/pcap and https://www.tcpdump.org.

Using libpcap, we will capture packets right as they come off of the network card.

1. You are expected to:
    a. Monitor the traffic live from a network interface (pcap_open_live)
    b. Read a pcap file (pcap_open_offline).
2. For this assignment, you will capture network traffic and you will process the incoming TCP and UDP packets. Do not use pcap_compile or pcap_setfilter.

More specifically, you are expected to do the following:

1. Select one interface that you wish to monitor or select the pcap file name.
2. Start capturing/reading packets.
3. Apply any provided filter.
4. Decode each received packet (i.e., is it a TCP or UDP packet?)[1].
5. Skip any packet that is not TCP or UDP.
6. Print the packet's source and destination IP addresses.
7. Print the packet's source and destination port numbers.
8. Print the packet's protocol.
9. Print the packet's TCP/UDP header length and TCP/UDP payload length in bytes.
10. Find where is the payload in memory.
11. Can you tell if an incoming TCP packet is a retransmission? If yes, how? If not, why?
12. Can you tell if an incoming UDP packet is a retransmission? If yes, how? If not, why?
13. In your program (when possible), mark each retransmitted packet as "Retransmitted".
14. On exit, your program must print the following statistics:
    a. Total number of network flows captured[2].
    b. Number of TCP network flows captured.
    c. Number of UDP network flows captured.

[1] Support both IPv4 and IPv6 packets.

[2] A network flow is defined by the 5-tuple {source IP address, source port, destination IP address, destination port, protocol}.

d. Total number of packets received (include the packets you skipped, that weren’t TCP or UDP packets.).
e. Total number of TCP packets received.
f. Total number of UDP packets received.
g. Total bytes of TCP packets received.
h. Total bytes of UDP packets received.

Tool Specification

Your tool will receive the following arguments from the command line upon execution.

Options:

-i	Network interface name (e.g., eth0)
-r	Packet capture file name (e.g., test.pcap)
-f	Filter expression (e.g., port 8080)
-h	Help message

Examples for execution:

- ./pcap_ex -i eth0 (save the packets in log.txt)
- ./pcap_ex -r test_pcap_5mins.pcap (print the outputs in terminal)
- ./pcap_ex -i eth0 -f “port 8080”

Notes

1. The options defined in the “Tool specification” section must remain as-is.
2. If no appropriate option was given, your program has to print the appropriate error message.
3. You need to create a Makefile to compile your library and programs (you must submit it with your source code).
4. You are provided with a sample packet capture to test your program. Its duration is 5 minutes.
5. You need to create a README with your name, your AM and a short description of your implementation.
6. You must submit the following files: README, Makefile, pcap_ex.c.
7. You should write the outputs of the execution (with **-i**) in a **log.txt** file and the outputs of the execution (with **-r**) appear in **terminal**.

8. You should place all these files in a folder named <AM>_assign5 and then compress it as a .zip file. For example, if your login is 2020123456 the folder should be named 2020123456_assign5 you should commit 2020123456_assign5.zip.