CS5313: Computer Networks **Instructor: Dr. Deepak Tosh**
Fall 2020 **Due Date:** 10/25/2020

Assignment # 3

GOAL:

Packet sniffing is an important concept in networks and mostly applied when it comes to network engineering and security applications. Being able to understand how packets are captured is essential for understanding deeper security measures in networking. There are many packet sniffing tools, such as Wireshark, Tcpdump, Netwox, Scapy, etc. Some of these tools are widely used by security experts, as well as by attackers. Being able to use these tools is important for students, but what is more important for students in a networks course is to understand how some of these tools work, i.e., how packet sniffing is implemented in software. The objective of this lab is two-fold: (a) learning to use a well-known sniffing tool; (b) write simple sniffer and filtering programs to gain an in-depth understanding of the technical aspects of packet sniffing tools.

Disclaimer: This assignment must be done alone by the students. You can always take help from the Web and other peers, but the code must be written solely by the student.

Environment Setup:

You can download any Ubuntu VM (preferably Ubuntu 16.04 32-bit version from here) and install it in your VirtualBox environment. Then, you can clone this VM to create another VM so that a network of two VMs can easily be set up. Please ensure, you have Python 3 and telnet installed in the first VM before cloning.

Task-1: Sniffing packets using Scapy (30 pts)

Many tools including Wireshark can be used to do sniffing, but most of them only provide fixed functionalities. **Scapy** is different: it can be used not only as a tool, but also as a building block to construct other sniffing and spoofing tools, i.e., we can integrate the Scapy functionalities into our own program. Here, you will be exploring Scapy to conduct packet sniffing.

If the VM does not have Scapy installed,

```
$ sudo pip3 install scapy
```

(a) A sample scapy code is provided below. Run the program with the root privilege and demonstrate that you can indeed capture packets. After that, run the program again, but without using the root privilege; describe and explain your observations

```
#!/usr/bin/python3
from scapy.all import *
def print_pkt(pkt):
      pkt.show()

pkt = sniff(prn=print_pkt)
```

(b) Usually, when we sniff packets, we are only interested certain types of packets. We can do that by setting filters in sniffing. Scapy's filter use the BPF (Berkeley Packet Filter) syntax;

you can find the BPF manual from the Internet. Please set the following filters and demonstrate your sniffer program again (each filter should be set separately):

i. Capture only the ICMP packet
ii. Capture any TCP packet that comes from a particular IP and with a destination port number. (Feel free to choose your own IP and port #)

Task-2: Writing a C-Program to Sniff Packets (70 pts)

Sniffer programs can be easily written using the *pcap* library. With *pcap*, the task of sniffers becomes invoking a simple sequence of procedures in the *pcap* library. At the end of the sequence, packets will be put in buffer for further processing as soon as they are captured. All the details of packet capturing are handled by the *pcap* library. A reference tutorial for *pcap*, can be found at http://www.tcpdump.org/pcap.htm.

Note: A starter code "mySniffer.c" is provided for you.

(a) First, you need to write a sniffer program (by modifying the starter code) that prints out the **source and destination IP addresses** of each captured packet. You should provide screenshots as evidences to show that your sniffer program can run successfully and produces expected results. In addition, answer the following questions:
   i. In your own words, describe the sequence of the library calls that are essential for sniffer programs. (Just summary only)
   ii. Why do you need the root privilege to run a sniffer program? Where does the program fail if it is executed without the root privilege?

(b) Please write **filter expressions** for your sniffer program to capture each of the followings. You can find online manuals for pcap filters. In your lab reports, you need to include screenshots to show the results after applying each of these filters.
   i. Capture the ICMP packets between two specific hosts.
   ii. Capture the TCP packets with a destination port number in the range from X to Y (user inputs)

(c) **Sniffing Passwords.** Please show how you can use your sniffer program to capture the password when somebody is using *telnet* on the network that you are monitoring/sniffing. You may need to **modify your sniffer code to print out the data part** of a captured TCP packet (as *telnet* uses TCP). It is acceptable if you print out the entire data part, and then manually mark where the password (or part of it) is.
   Note: You need to search/read on how to use *telnet*, if you have never used it.

What to be submitted:

1. A **PDF formatted report** with all required answers, evidences, codes, execution samples, instructions for running the submitted programs, and references used. The filename should be formatted as "*Lastname_firstname_Assign3.pdf*".
2. Well-documented source codes and *README.txt* in the form of **one single Zip** file.

Submission:

Submit your **Python/C codes** for all the implemented algorithms, *README.txt* and the report on **Blackboard only.**