**Khanh Nguyen**

**CSCE465-500**

**UIN 525000335**

**Homework 1**

**Problem 1: Please use your own words to describe the sequence of the library calls that are essential for sniffer programs. This is meant to be a summary, not a detailed explanation like the one in the tutorial.**

The sequence of library calls is as of below:

1. pcap\_lookupdev: finds the capture device.
2. pcap\_lookupnet: returns network number of the capture device.
3. pcap\_open\_live: starts sniffing.
4. pcap\_datalink: returns the type of link-layer headers.
5. pcap\_compile: compiles filter expression.
6. pcap\_setfilter: sets the filter from pcap\_compile.
7. pcap\_next: captures a single packet at a time.
8. pcap\_loop: continues sniffing by looping.
9. pcap\_freecode: deallocates memory.
10. pcap\_close: end sniffing.

**Problem 2: Why do you need the root privilege to run sniffex? Where does the program fail if executed without the root privilege?**

The root privilege is required because the code needs access to network device and performs sniffing through network socket.

Without the root privilege, it will fail at pcap\_open\_live (open access to capture device).

**Problem 3: Please turn on and turn off the promiscuous mode in the sniffer program. Can you demonstrate the difference when this mode is on and off? (Note: even if the promiscuous mode does not work in TCR, you can still demonstrate your trial with the mode on and off in the sniffer program) Please describe the use of promiscuous mode (explained in the class) and explain your observations.**

The promiscuous mode can be turn on and off by setting the third argument of pcap\_open\_live(dev, SNAP\_LEN, 1, 1000, errbuf). 1 is on and 0 is off.

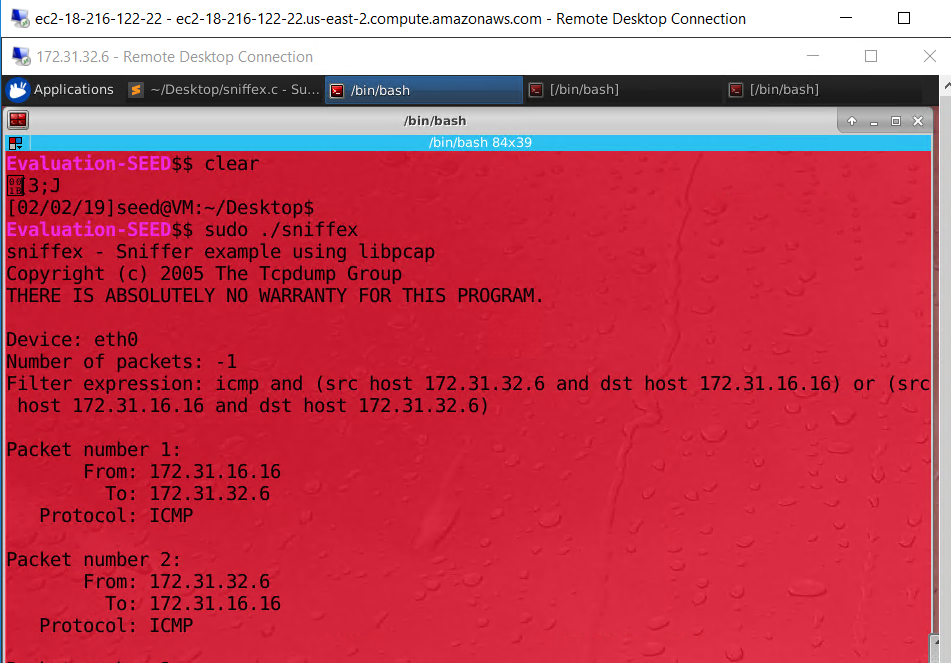
When the promiscuous mode is on, it shows all the contents of the package captured. Therefore the output is much more in promiscuous mode.

**Problem 4: Please write filter expressions to capture each of the following. In your lab reports, you need to include screendumps to show the results of applying each of these filters.**

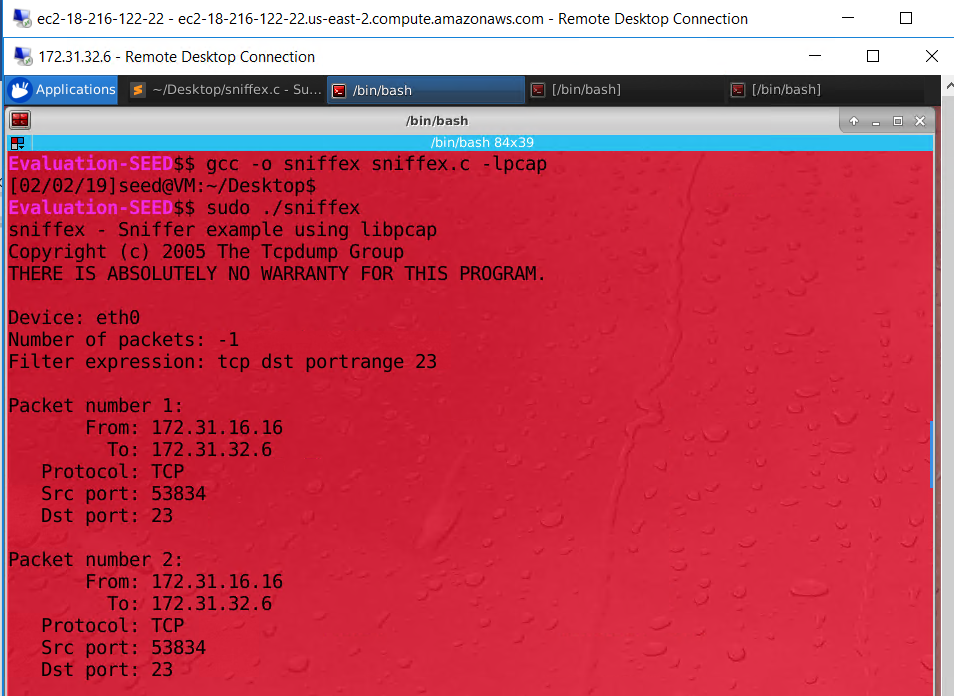
**• Capture the ICMP packets between two specific hosts.**

**• Capture the TCP packets that have a destination port range from to port 10 - 100.**

To capture ICMP packets between two specific hosts, filter expression can be changed to “**icmp and (src host 172.31.32.6 and dst host 172.31.16.16) or (src host 172.31.16.16 and dst host 172.31.32.6)**”. Screenshot is below:

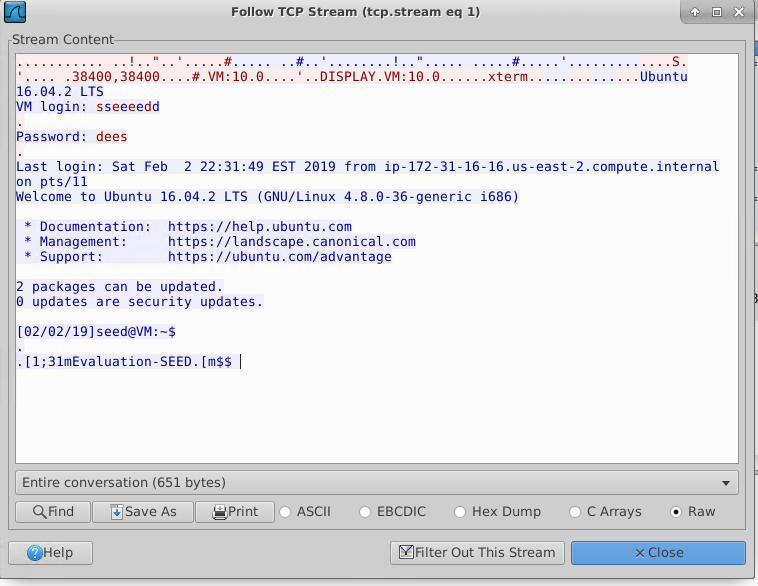


To capture TCP packets that have a destination port range from to port 10 – 100, filter expression can be changed to “**tcp dst portrange 10-100**”. The Screenshot below:

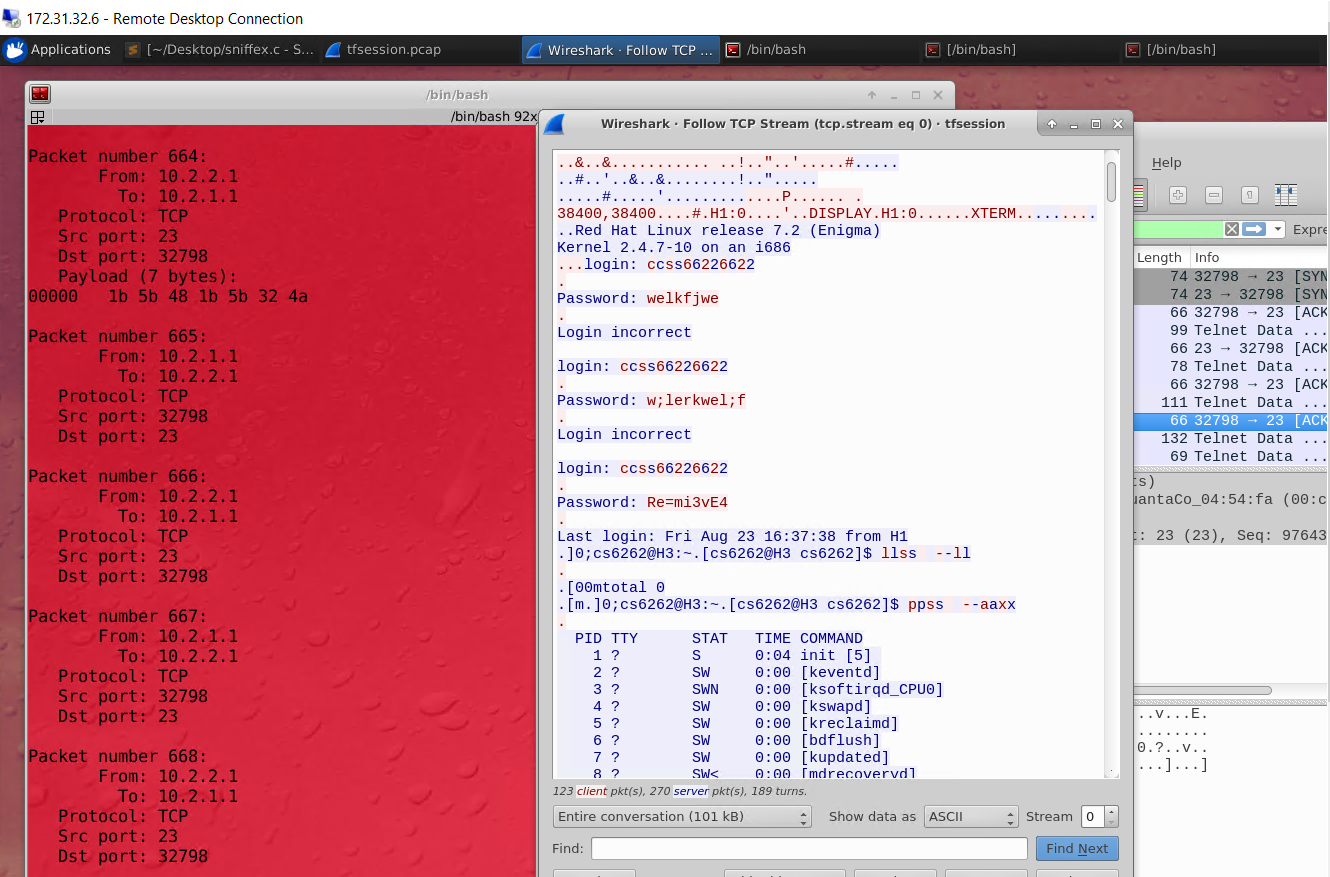


**Problem 5: Please show how you can use sniffex to capture the password(s) when somebody is using telnet on the network that you are monitoring. You can start from modifying sniffex.c to implement the function. You also need to start the telnetd server on your VM. If you are using our pre-built VM, the telnetd server is already installed; just type the following command to start it.**

Password can be capture by changing the filter expression to “tcp port 23” and telnet from another VM.



To capture password offline we can change the pcap\_open\_live(..) to pcap\_open\_offline(("tfsession.pcap",errbuf). The password captured from wireshark is below:



**Problem 6: Please use your own words to describe the sequence of the library calls that are essential for packet spoofing. This is meant to be a succinct summary.**

1. Socket(): creating a raw socket that help the program inject packets into the network.
2. Socketsetopt(): Set the socket option (we the ICMP protocol to send the package)
3. Construct the packet (swap the ip addresses of source and destination when duplicating the reply package)
4. Sendto():Send the packet through the raw socket made.

**Problem 7: Why do you need the root privilege to run the programs that use raw sockets? Where does the program fail if executed without the root privilege?**

This is because raw sockets can expose all information being transmitted over the network. The program will fail when a raw socket is attempted to be created.

**Problem 8: Please combine your sniffing and the spoofing programs to implement a sniff-and-thenspoof program. This program monitors its local network; whenever it sees an ICMP echo request packet, it spoofs an ICMP echo reply packet with the information of another machine (other than the sender and receiver) in the network (in your AWS enclave).**

This is the wireshark screenshot of the sniff-and-then-spoof program created. (The package with ip address of 172.31.1.1 is the spoofed package. Notice that there are two reply packages: one is real (172.31.32.6) and one is fake. ). See attachment for the code.

