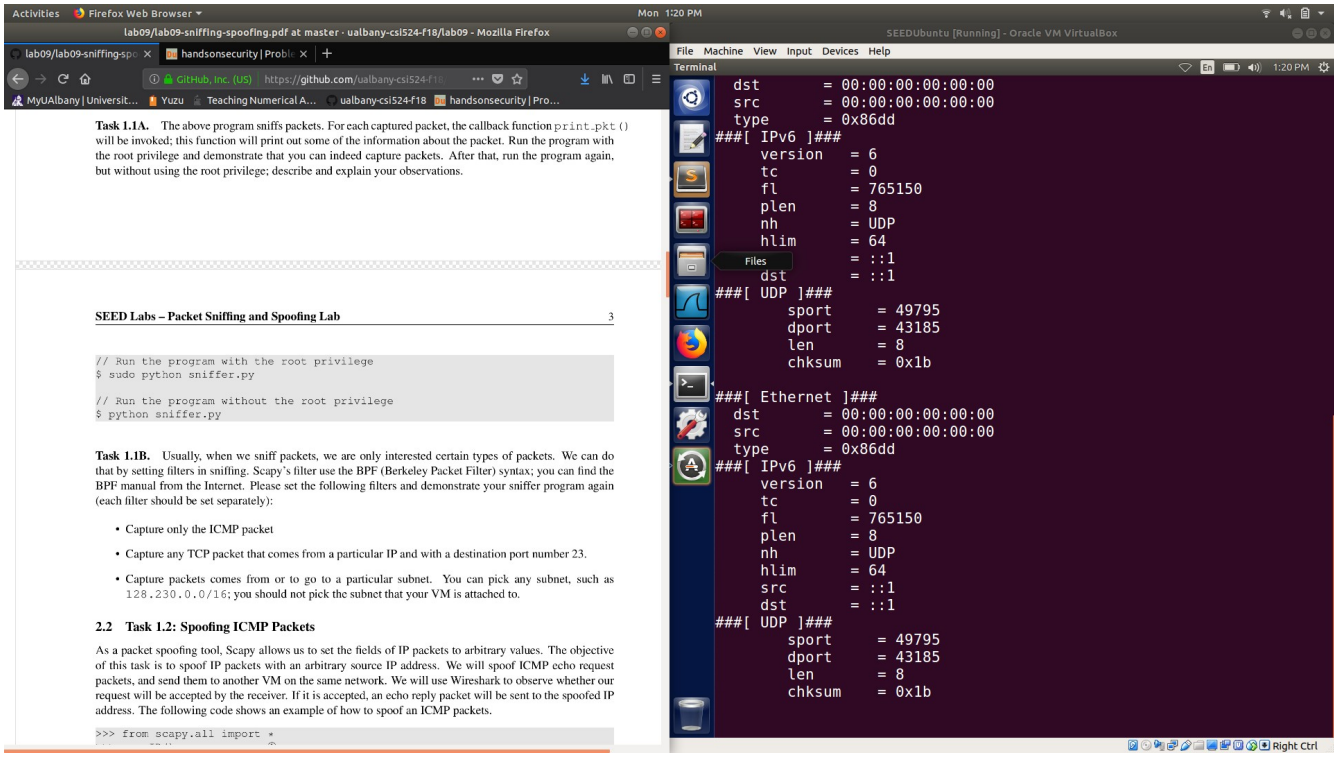
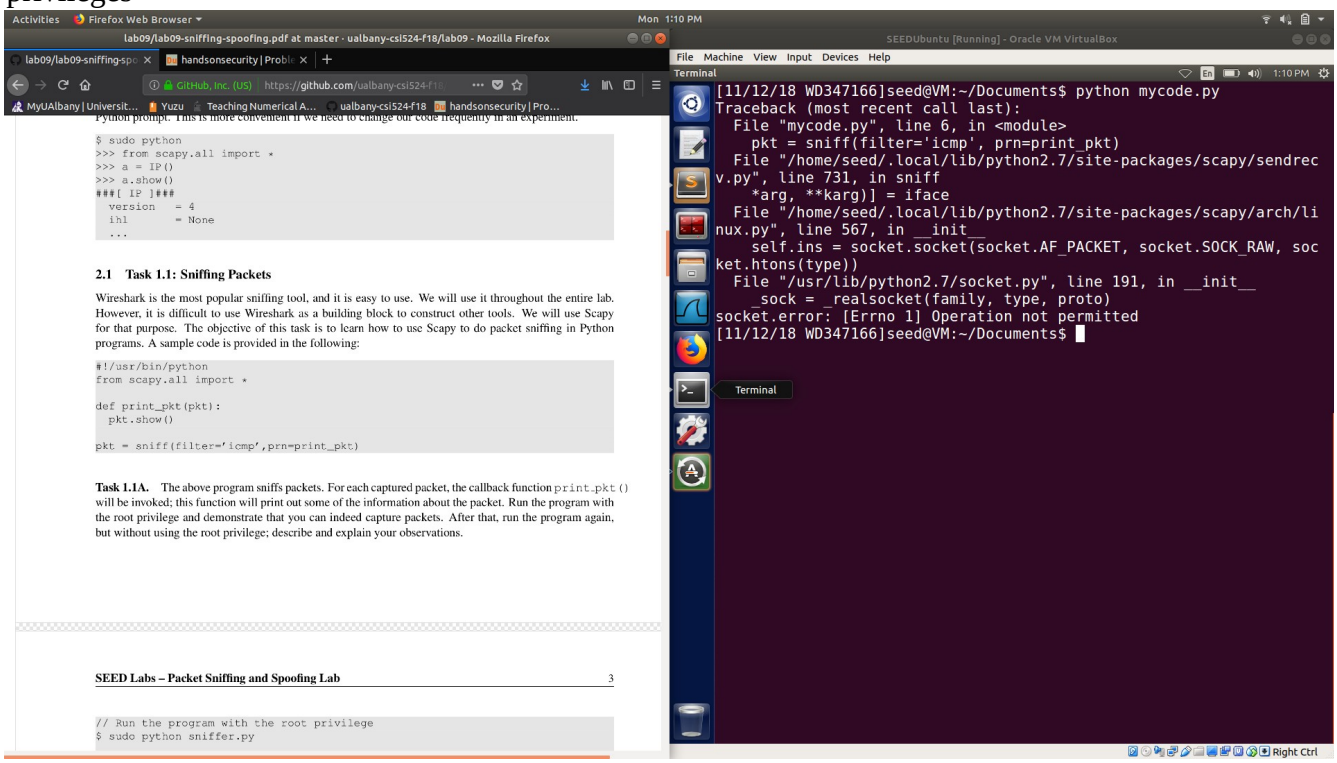
William Dahl
ICSI 424 Information security
Lab 9
November 16th, 2018

Task 1.1 A: In this screen shot I show that I am able to capture packets using the python program when using the superuser

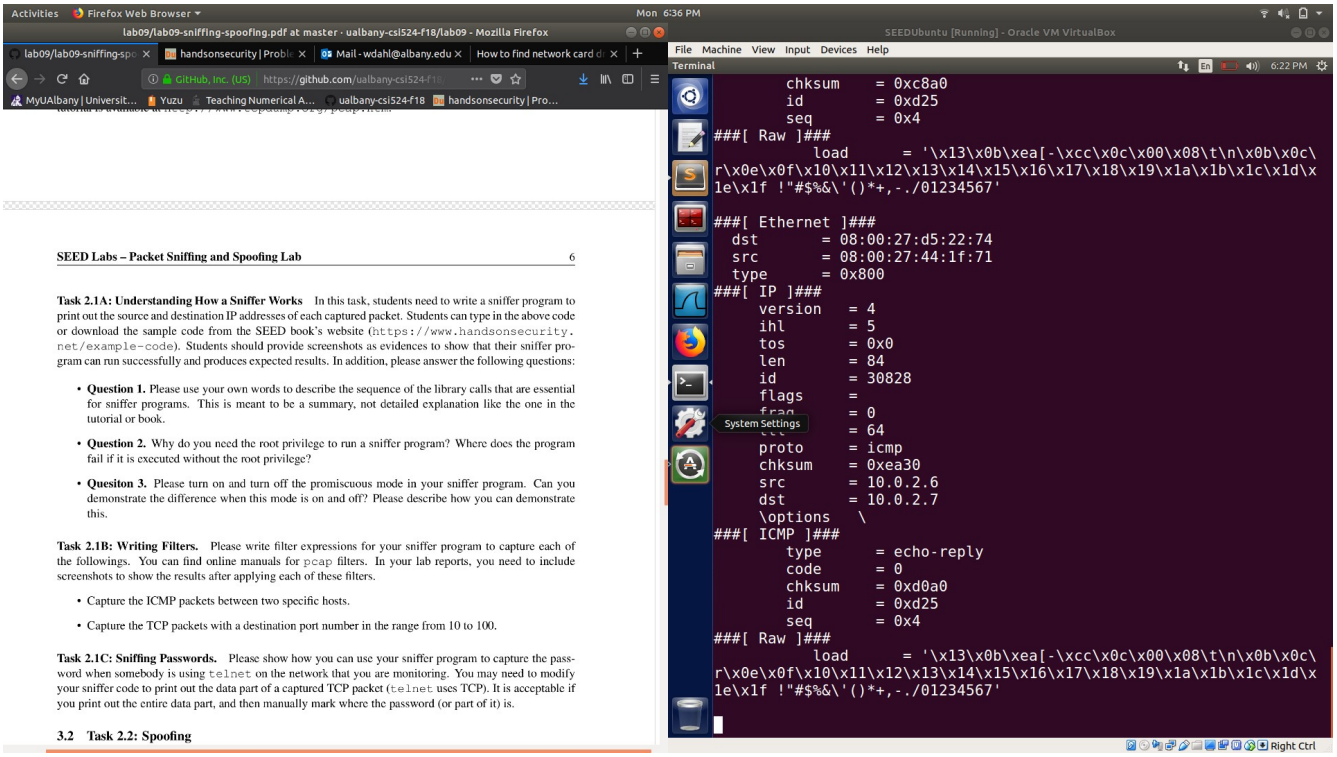


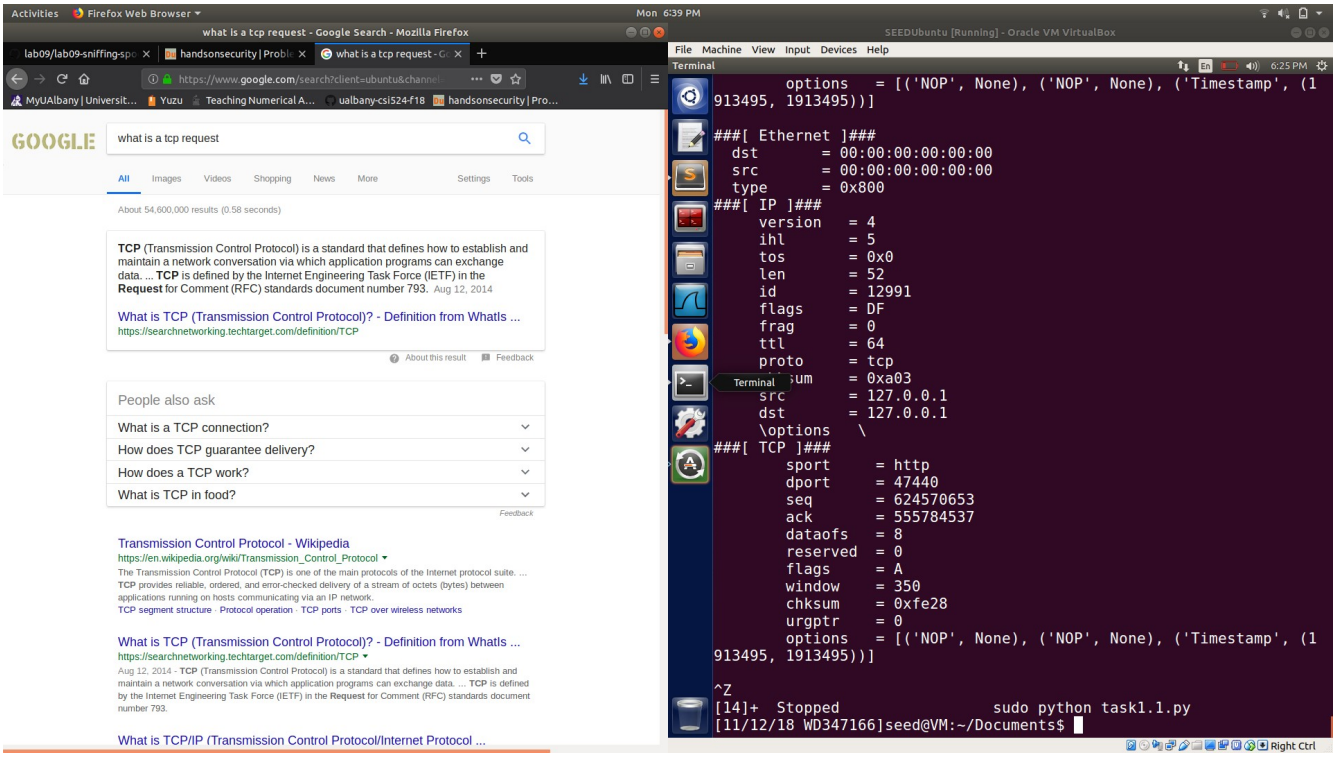
In this screen shot I show the error that I received when I attempted to run the program without the root privileges

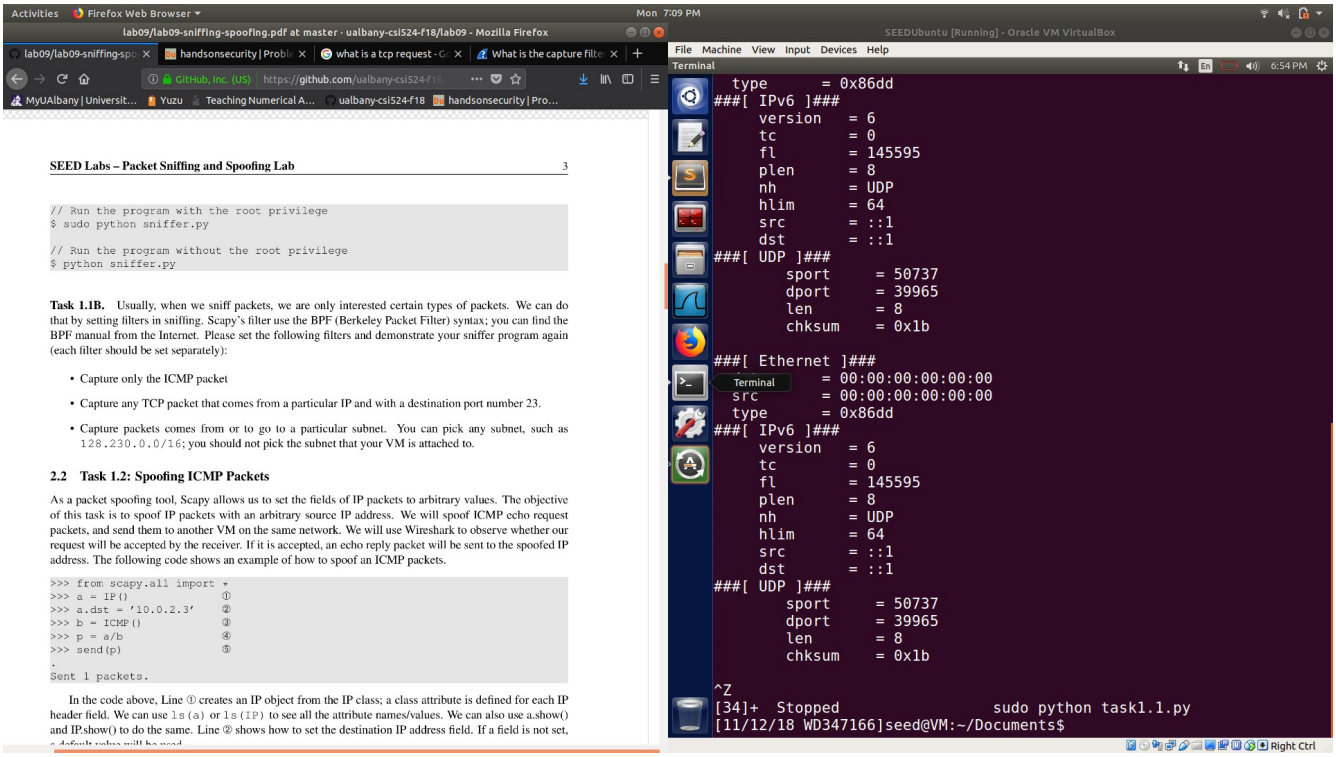


I got this error because a normal user does not have the privilege to run the realsocket function which is invoked by scapy when sniff is called.

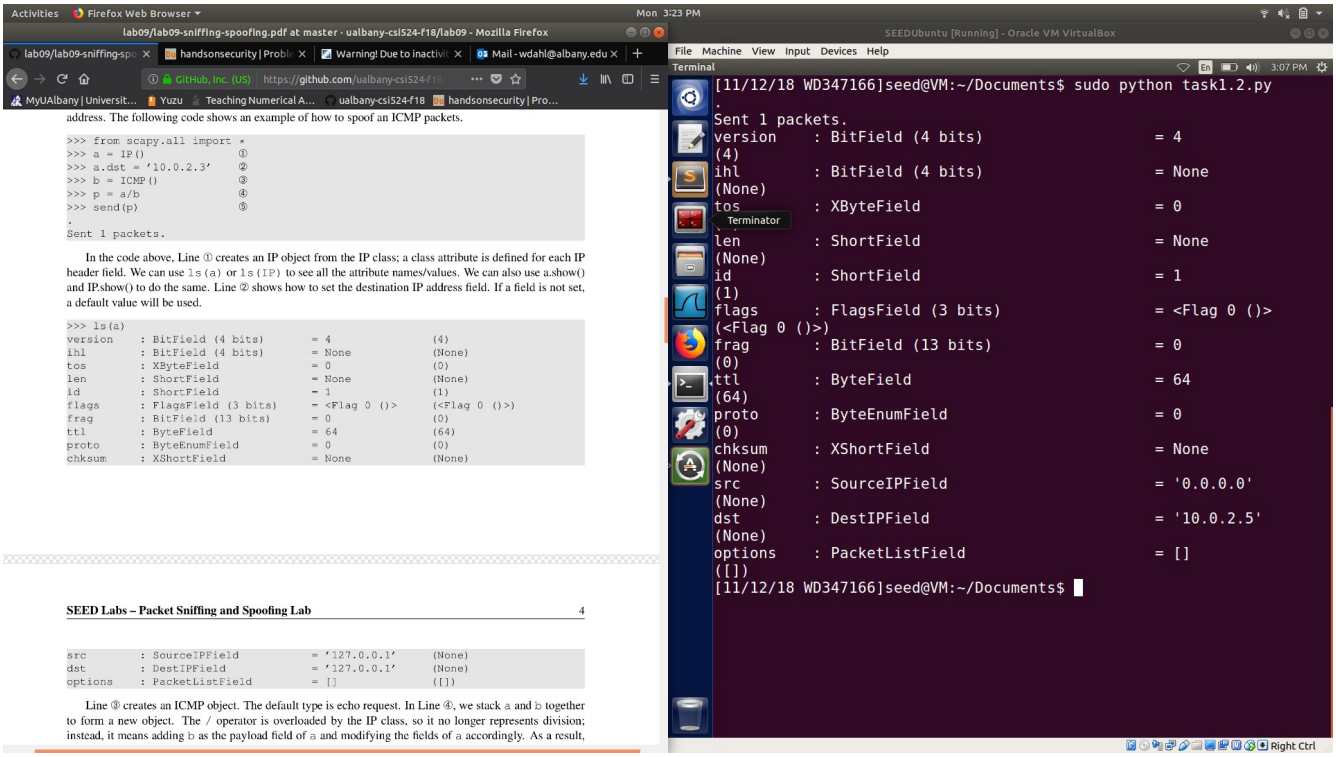
Task 1.1 B: In these screen shots I show the output from employing the filters “icmp” “tcp host 10.0.2.7 port 23” and “net 128.230.0.0/16” respectively.



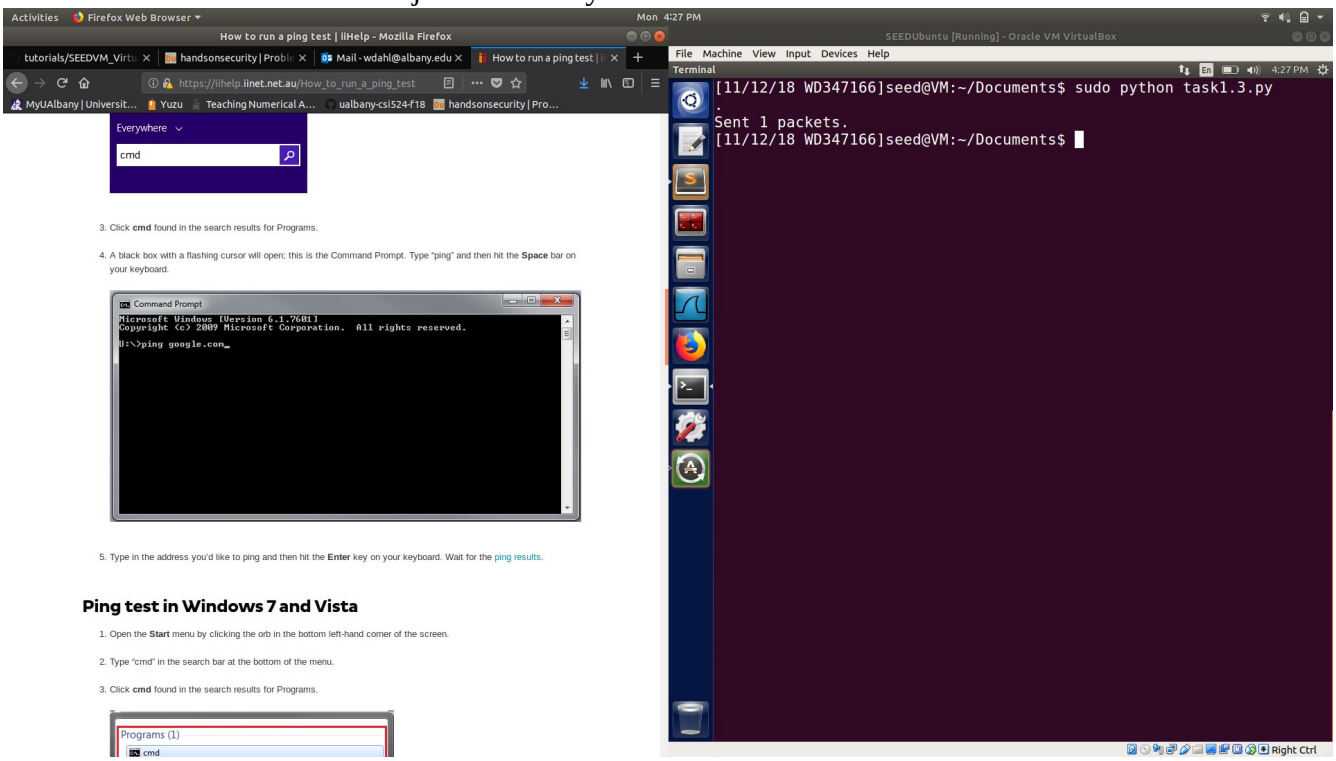




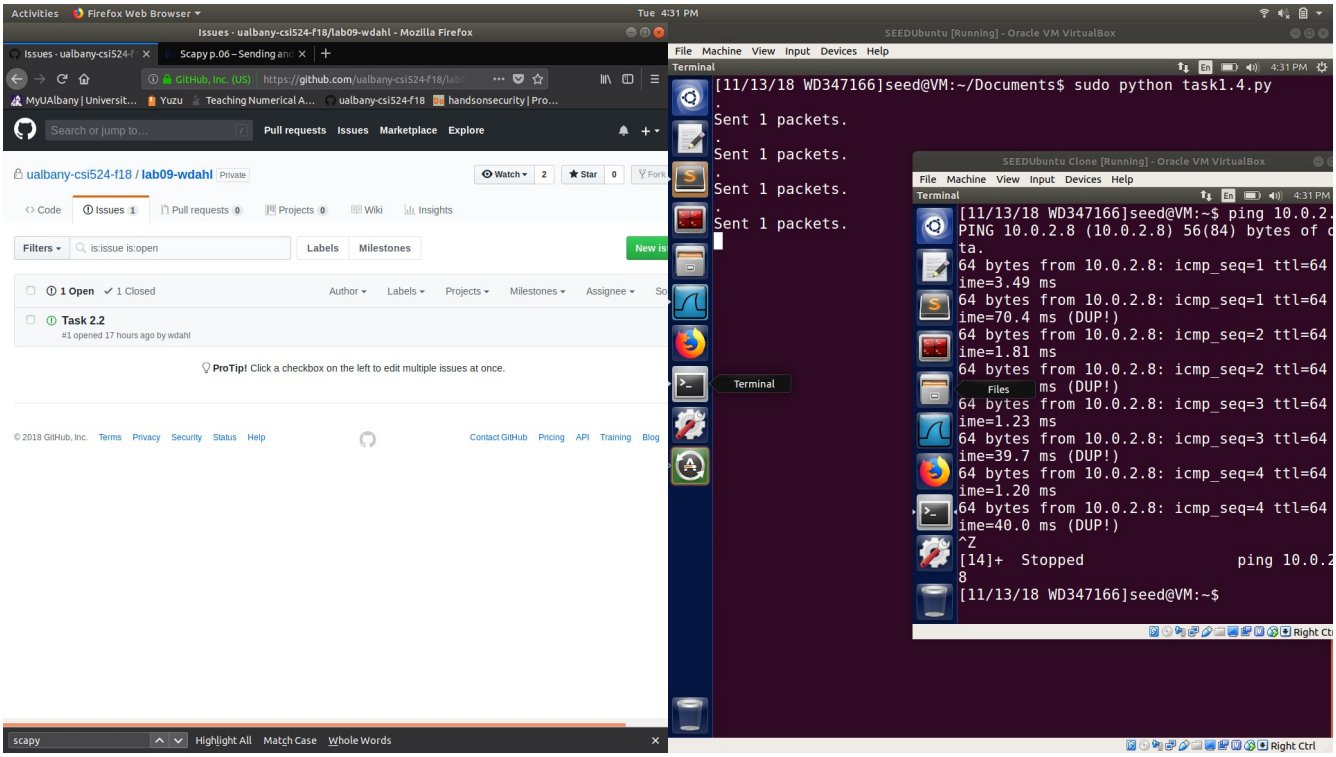
Task 1.2: In this screen shot I show that I was able to spoof an ICMP echo request packet with some arbitrary source IP address – 10.0.2.5 – of another VM.



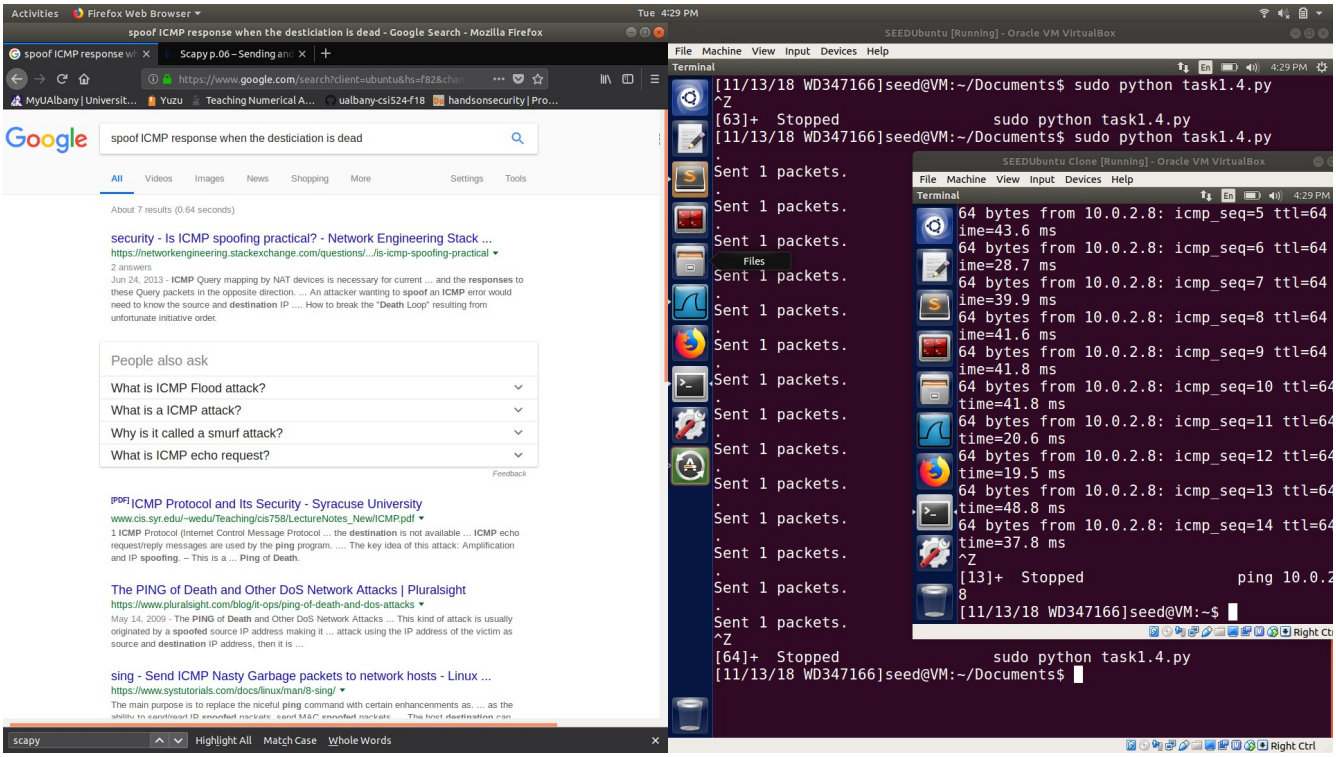
Task 1.3: In this screen shot I show that I was able to send a packet to the other VM I have with an IP address of 10.0.2.7 with a ttl of just 1 from my vm with an IP address of 10.0.2.6



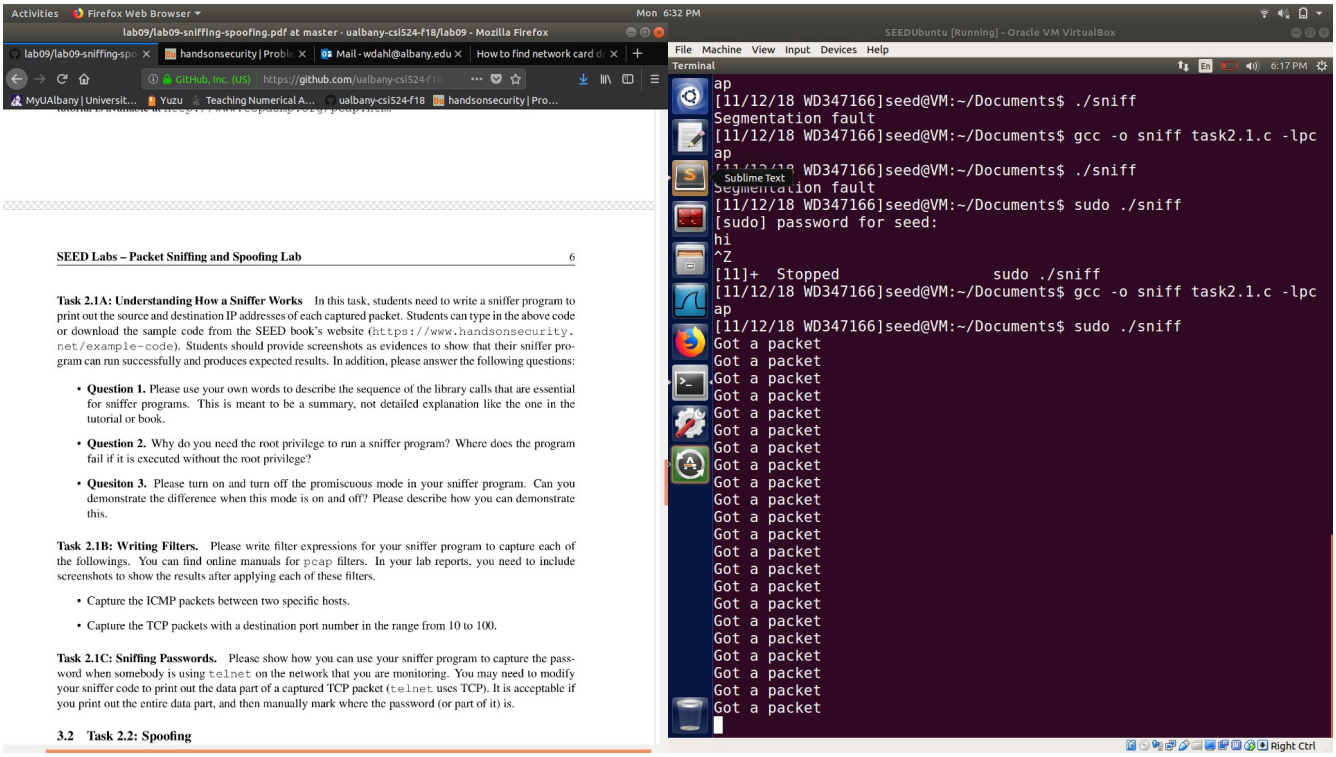
Task 1.4: In this screen shots I show that I am able to ping 10.0.2.8 from 10.0.2.7 and sniff those packets and send a packet back from 10.0.2.6 while 10.0.2.8 is alive



In these next two screen shots I show that I can send spoofed packets back to 10.0.2.7 when it pings 10.0.2.8 from 10.0.2.6 when 10.0.2.8 is down.



Task 2.1 A: In this screen shot I show that my sniffer program was able to sniff out packets



Questions:

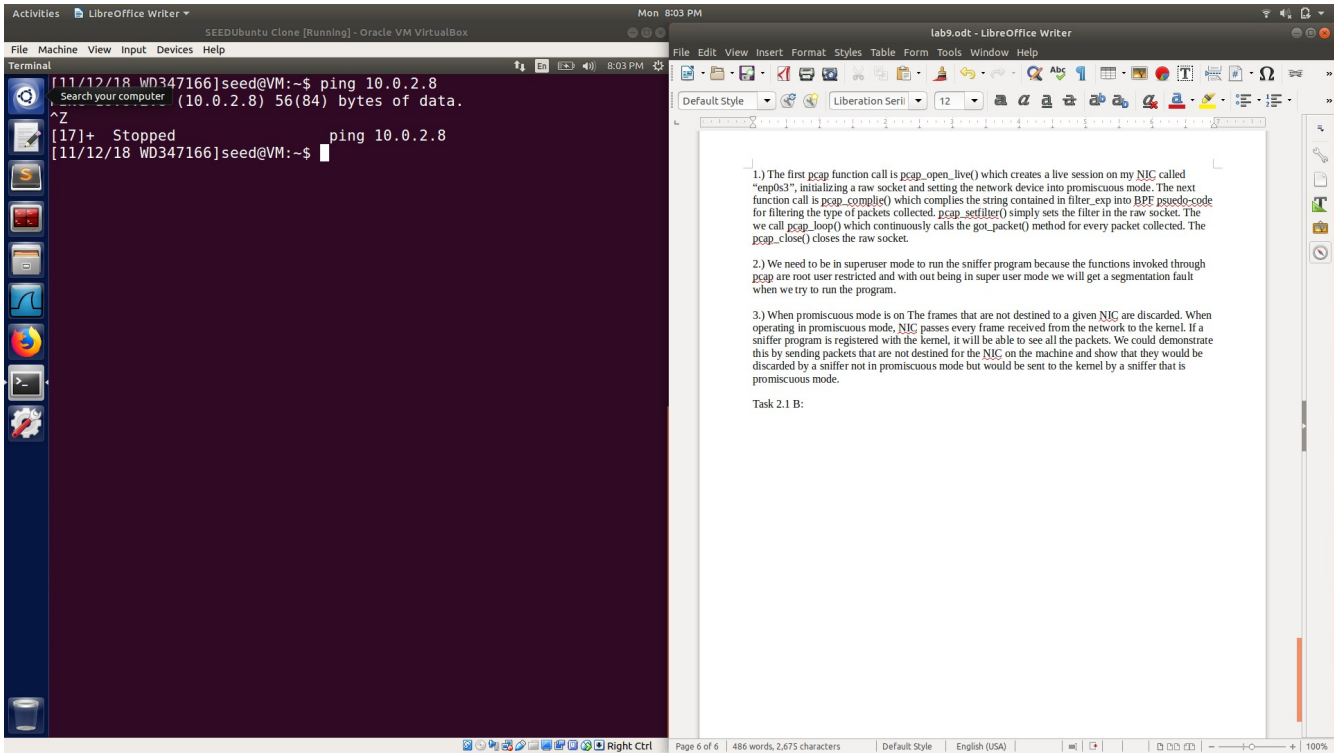
1.) The first pcap function call is pcap_open_live() which creates a live session on my NIC called “enp0s3”, initializing a raw socket and setting the network device into promiscuous mode. The next function call is pcap_complie() which complies the string contained in filter_exp into BPF psuedo-code for filtering the type of packets collected. pcap_setfilter() simply sets the filter in the raw socket. The we call pcap_loop() which continuously calls the got_packet() method for every packet collected. The pcap_close() closes the raw socket.

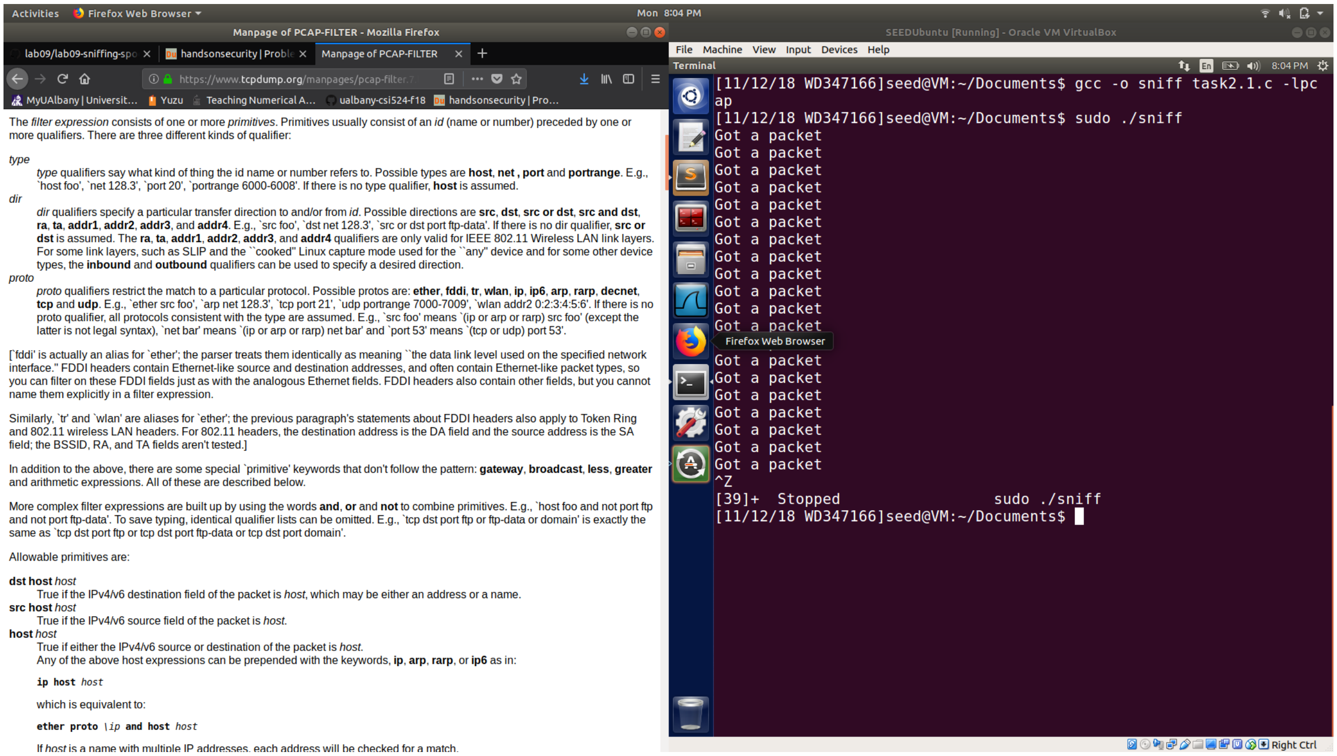
2.) We need to be in superuser mode to run the sniffer program because the functions invoked through pcap are root user restricted and with out being in super user mode we will get a segmentation fault when we try to run the program.

3.) When promiscuous mode is on The frames that are not destined to a given NIC are discarded. When operating in promiscuous mode, NIC passes every frame received from the network to the kernel. If a sniffer program is registered with the kernel, it will be able to see all the packets. We could demonstrate this by sending packets that are not destined for the NIC on the machine and show that they would be discarded by a sniffer not in promiscuous mode but would be sent to the kernel by a sniffer that is promiscuous mode.

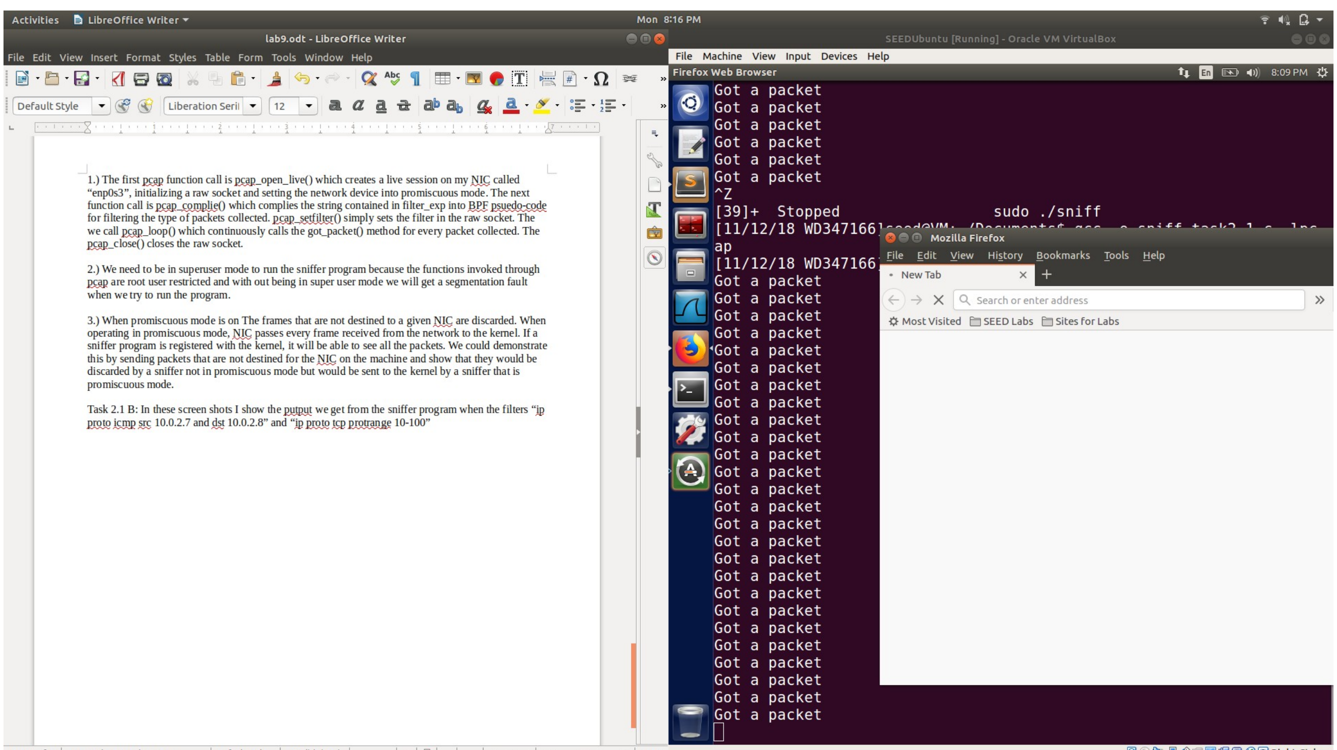
Task 2.1 B: In these screen shots I show the putput we get from the sniffer program when the filters “ip proto icmp src 10.0.2.7 and dst 10.0.2.8” and “ip proto tcp protrange 10-100” respectively.

These two screen shots show that packets are captured on 10.0.2.6 when 10.0.2.7 pings 10.0.2.8

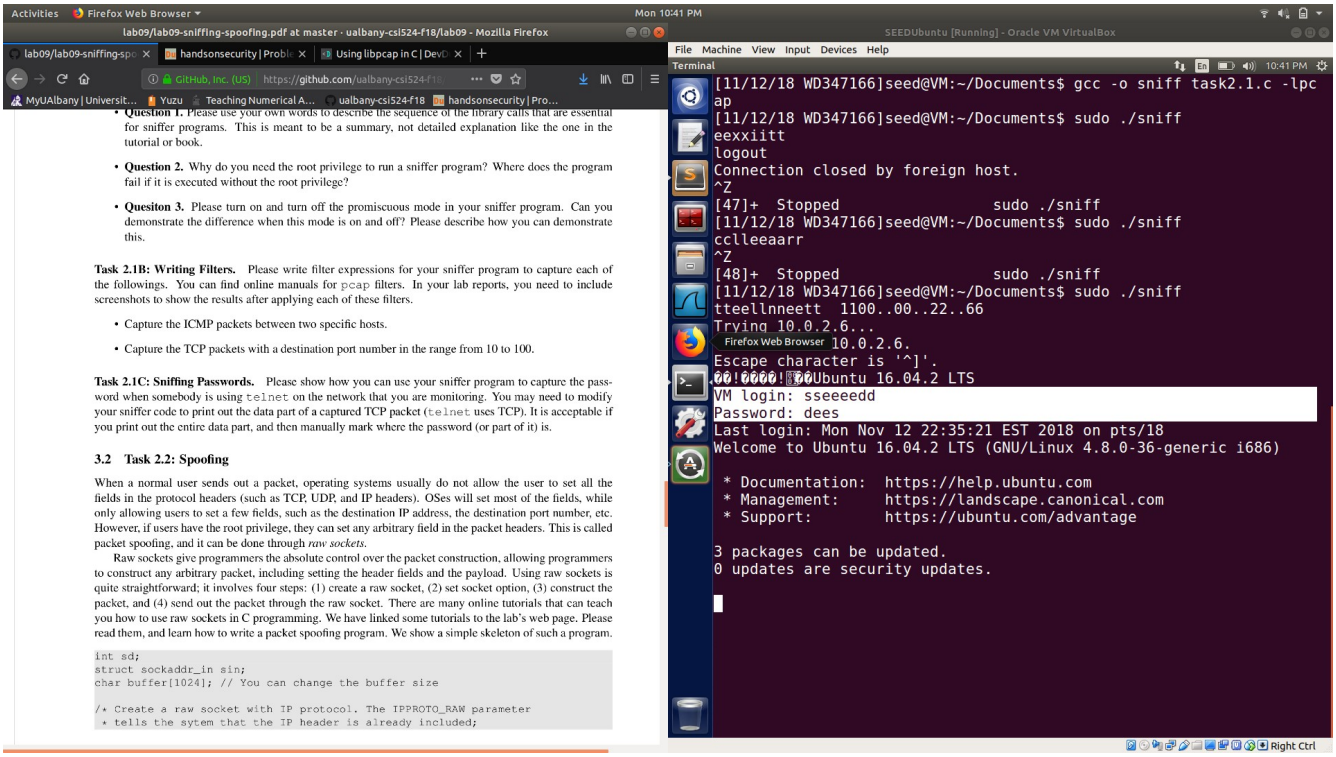




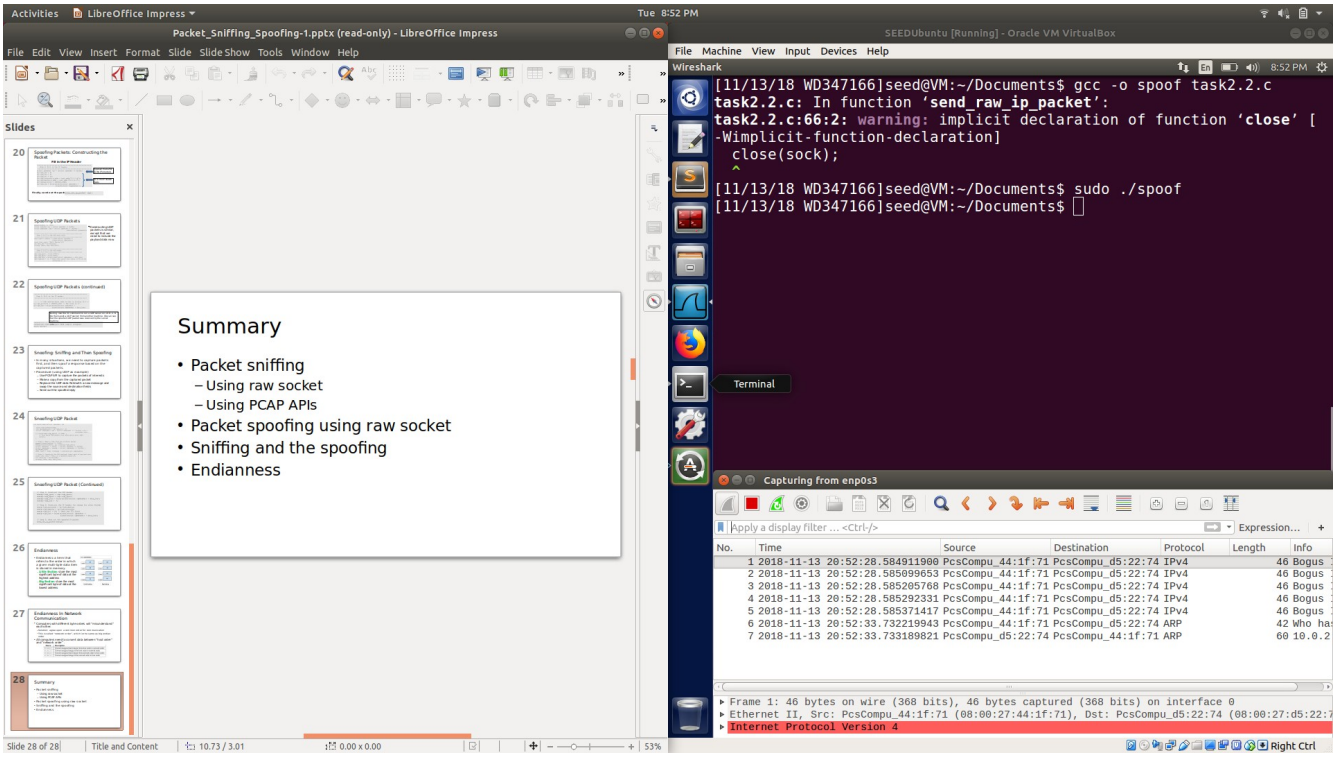
this screen shot shows that packets are received from opening the internet browser and getting tcp packets



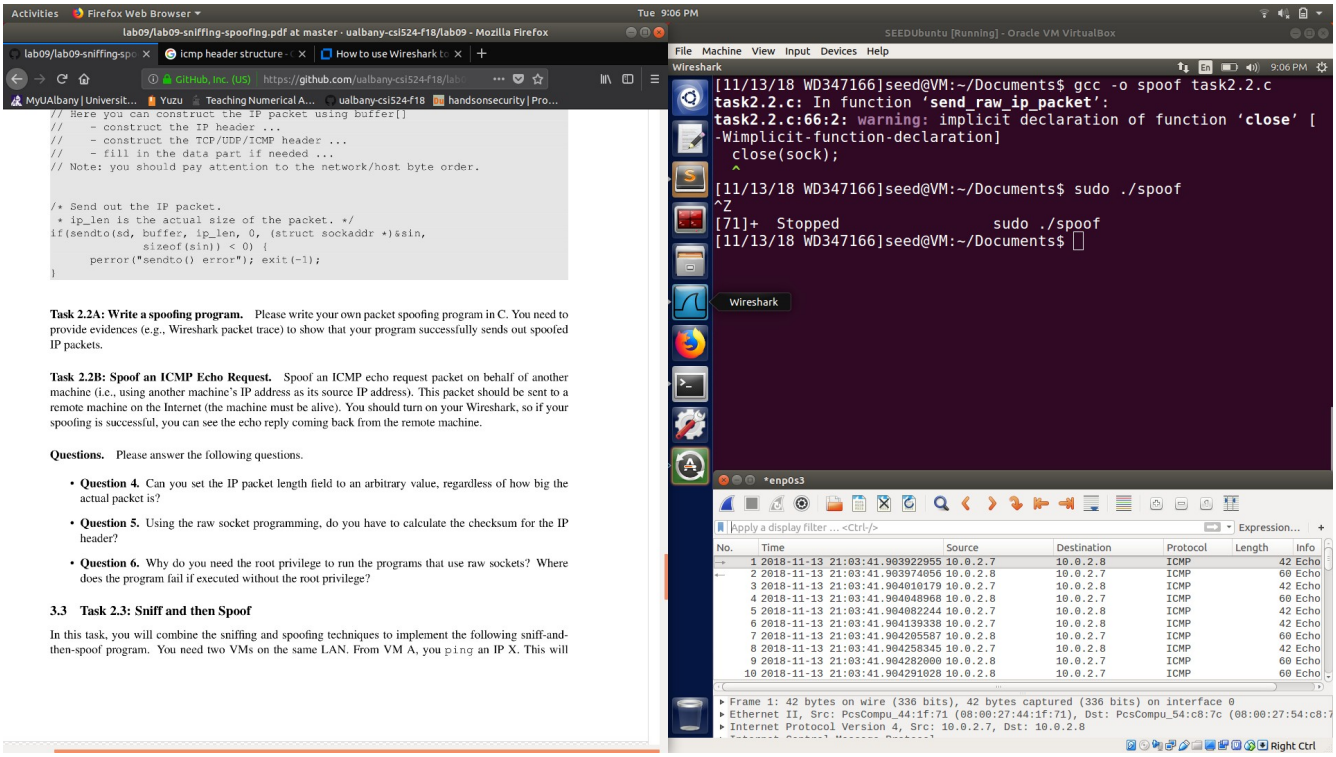
Task 2.1 C: In this screen shot I show how I was able to get the user name and password entered by someone on a network that I was monitoring,



Task 2.2 A: Th this screen shot I show that wire shark was able to pick up the IP packets that I was spoofing



Task 2.2 B: In this screen shot I show that my spoofer program was able to send ICMP packets from 10.0.2.8 to 10.0.2.7. You can also see the response sent back.



Questions:

4.) No you cannot, because we are using raw sockets so the length must be exact because it will not be set by the operating system like in regular sockets.

5.) You do not need to calculate the checksum for the IP header because the system will do that automatically when we send the packet, however you do need to for the protocol type header.

6.) You need root privilege to run the program because the certain library function calls require the euid to be root for security purposes. With out the root privileges the program will fail when an attempt at making a raw socket occurs.

Task 2.3 In these next two screen sots I show that my spoofer program was able to send ICMP packets to 10.0.2.7 when it pinged 10.0.2.8 when the vm was alive. It was also able to send back responses when the vm being pinged (10.0.2.8) was dead which caused the ping to not return a message that the host was unreachable thus showing that my spoofer was able to successfully spoof the ICMP response