2 lab tasks

2.1 Task 1: Writing Packet Sniffing Program

Task1.a:

Problem 1:

The first call is pcap_lookupdev(), which finds the device name that we want to sniff on.
The 2nd is pcap_open_live, which opens a sniffing session. We provide it with the device name that we found in the first step.
After that, we need to call pcap_datalink() to determine what type of ethernet header we are woeking with.
If we want to filter certain types of traffic, we call pcap_compile()
Next, we call pcap_loop() to capture n number of packets.

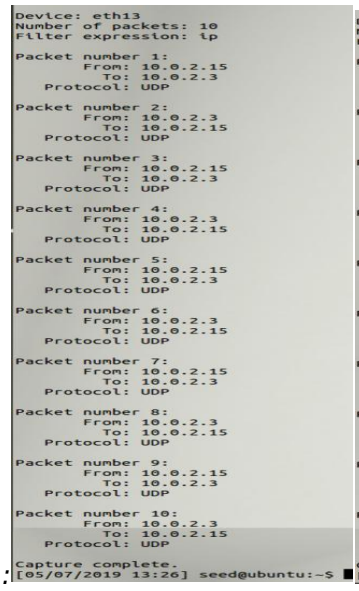
Problem 2:

We need to run sniffex with root privilege because we cannot enter promiscuous mode which is needed to capture all traffic on a given network.

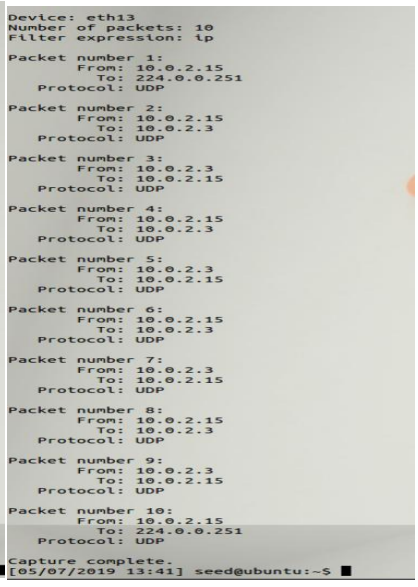
Problem 3:

In theory, promiscuous mode should affect the packets captured. However, in this test, I could not really see a difference.
To demonstrate this, here are some screen caps:

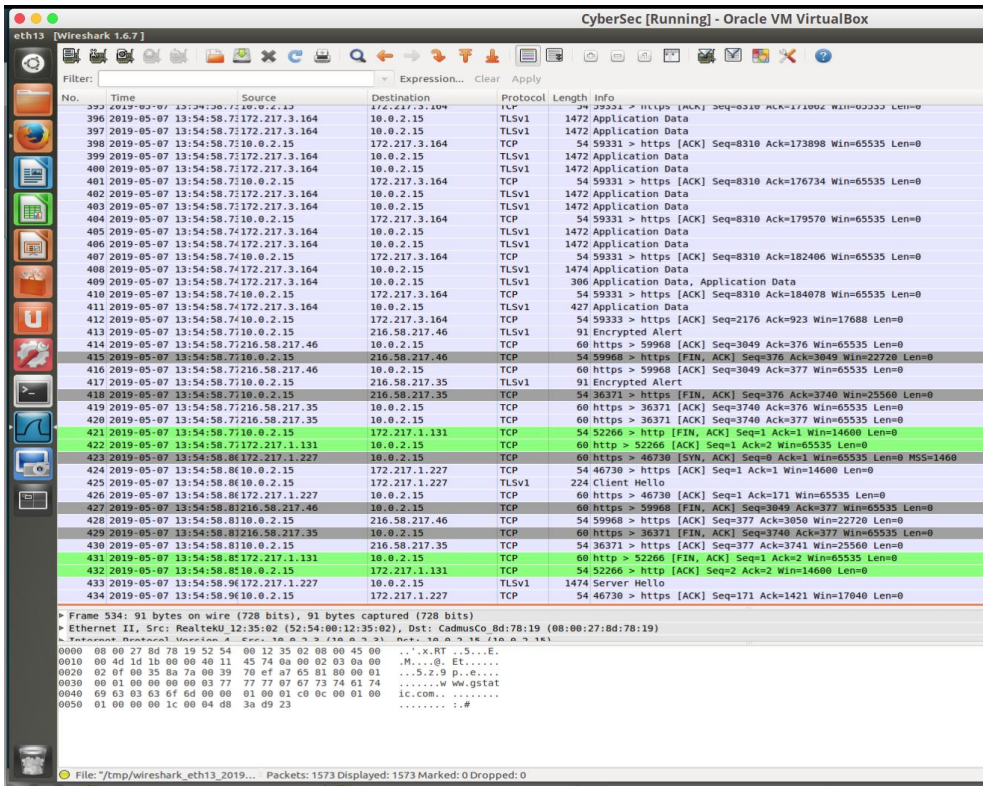
Promiscuous Mode off:



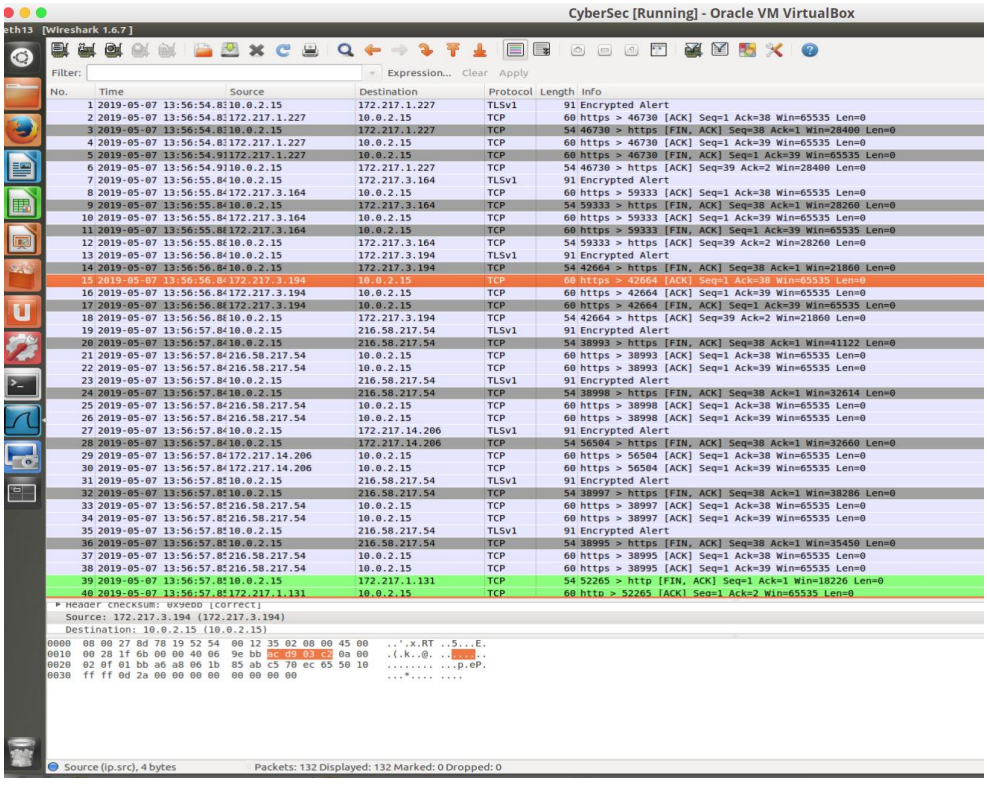
Promiscuous mode on:



Promiscuous Mode on:



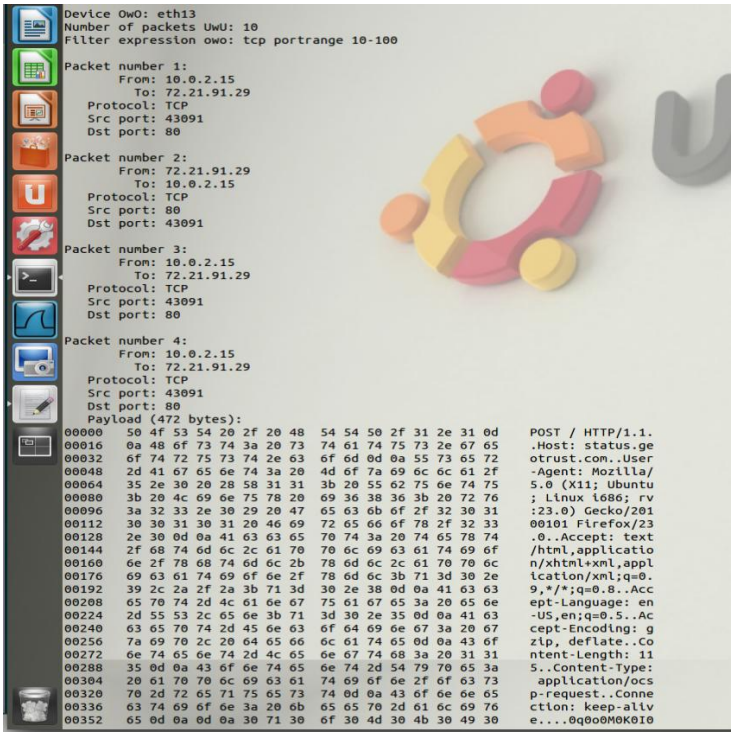
Promiscuous Mode off:



Task 1.b

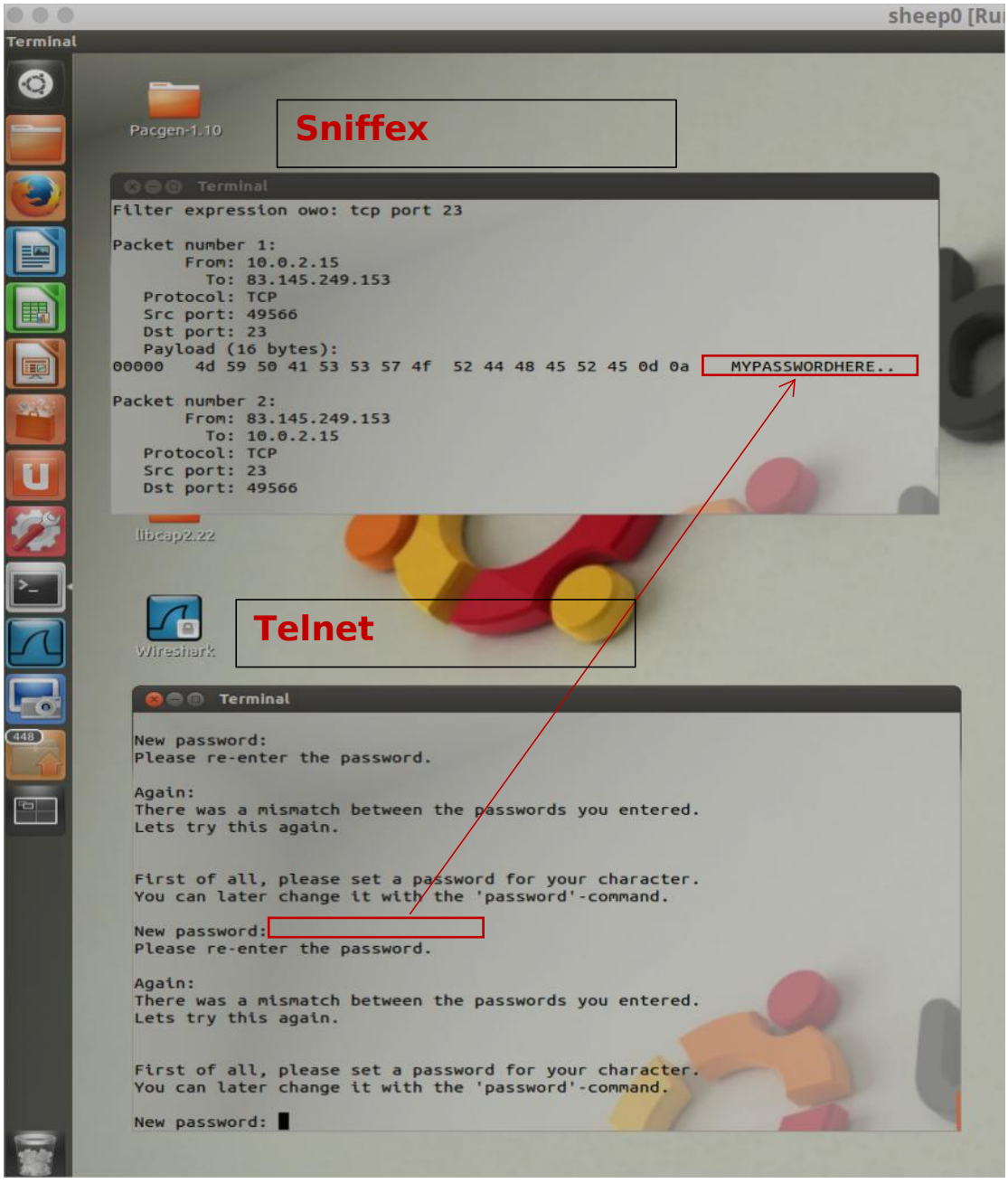
• Capture the ICMP packets between two specific hosts

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



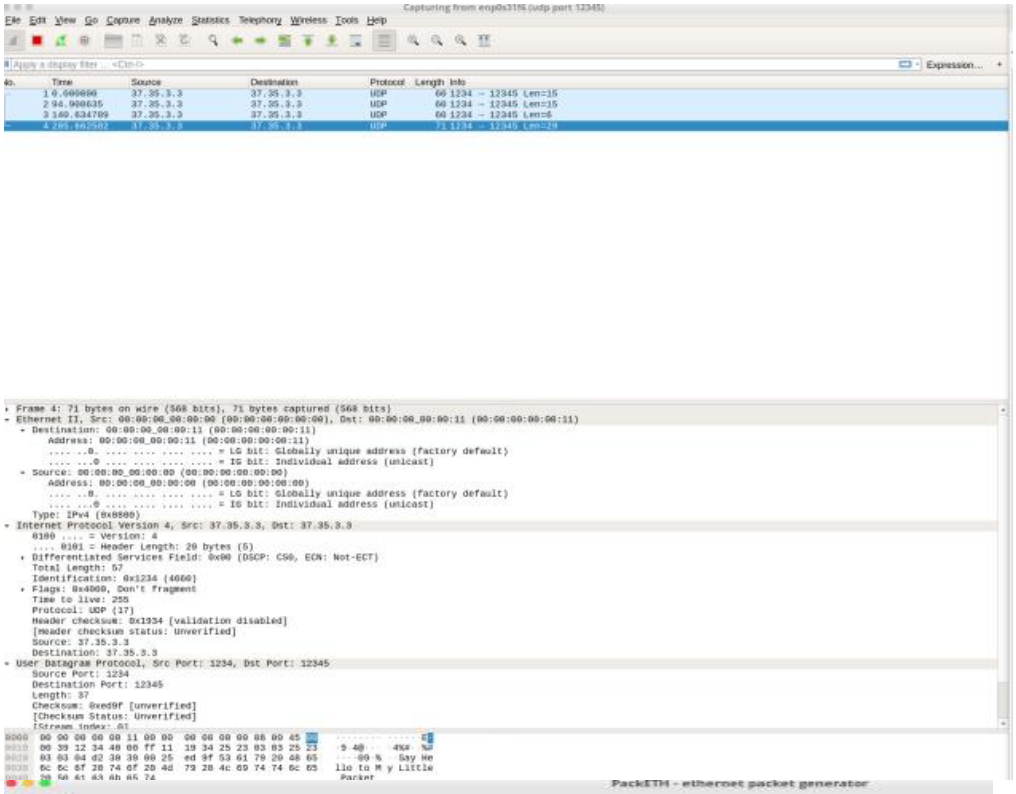
Task 1.c

To sniff passwords over telnet, we need to change the filter expression to only capture TCP packets on port 23 (the default port for telnet). Then, we can sniff the packets. What we will see is a request, and then a response containing the password as plain text.
Here is a screen cap to show this:



Task 2.a

In this part, I gave a UDP packet a custom payload that says “Say Hello to my Little Packet”. The actual hex values for this string can be seen at the bottom of the 2nd screen cap, while the translated text can be seen at the bottom of the 1st.



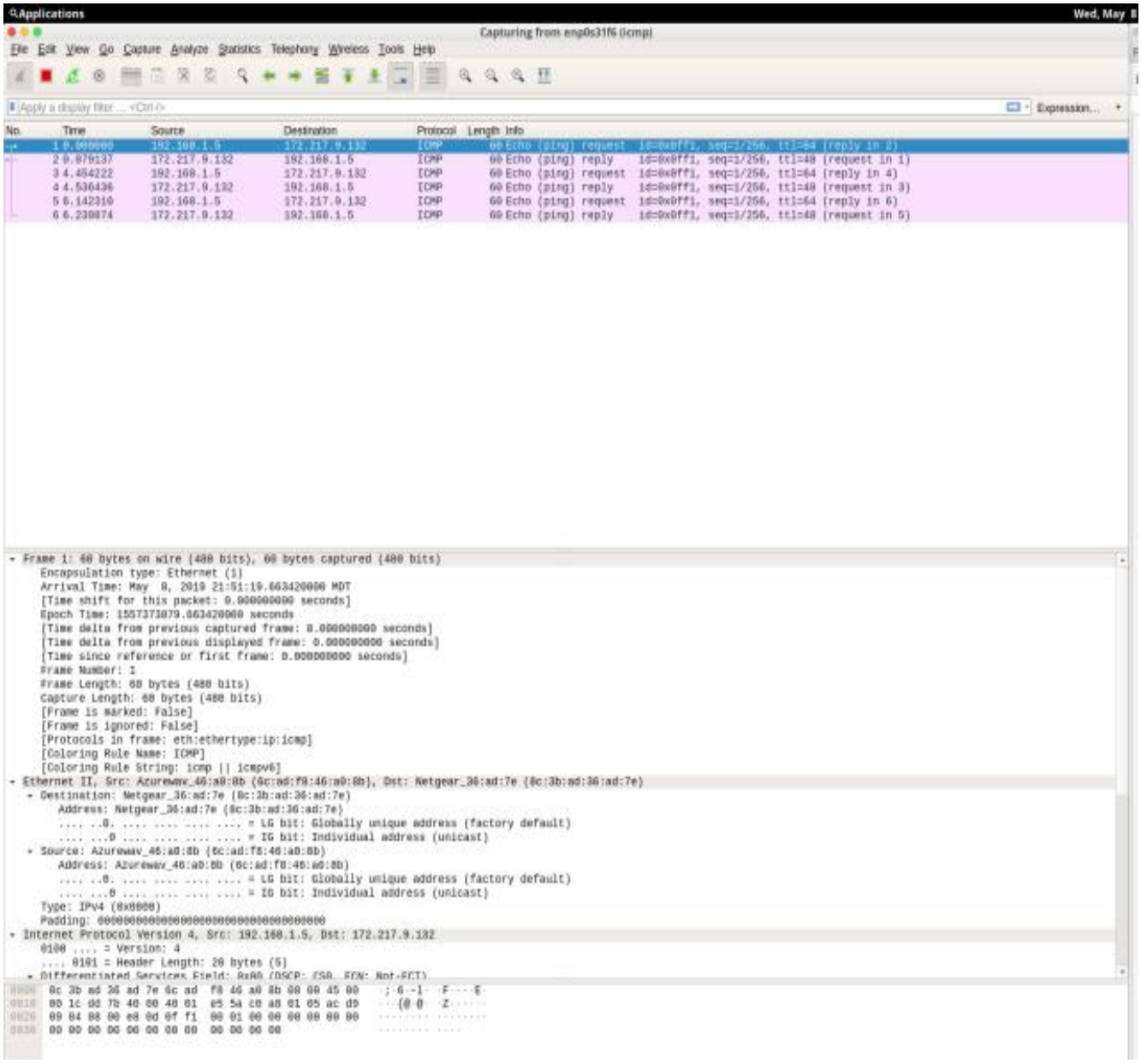
Task 2.b

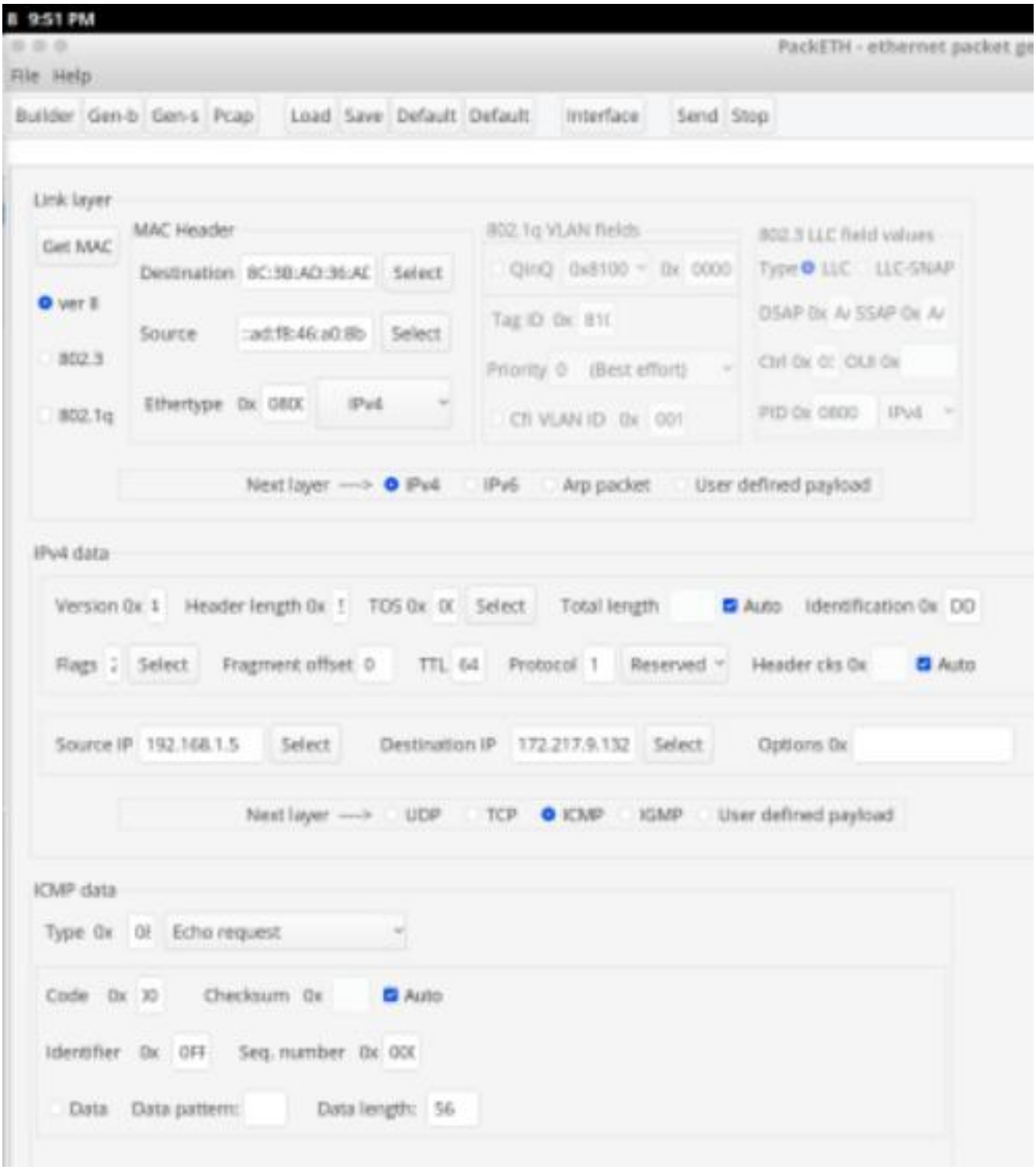
I decided to spoof an ICMP request on behalf of my roommate's computer.

First, i started a packet capture with wireshark, and looked at a packet that was sent between my computer and his computer, so i could get the MAC address and IP for his computer.

Then, I started a new packet capture with ICMP as a filter, and executed the command ~$ ping www.google.com. Again, i was able to get a suitable MAC address and IP from this.

Using my roommate's PC as a source, and google as a destination, i sent out an echo request and successfully got a response:



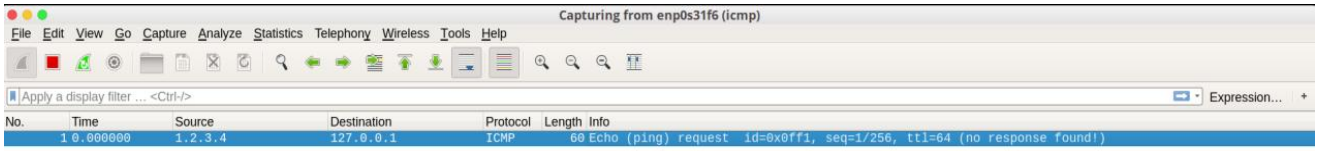


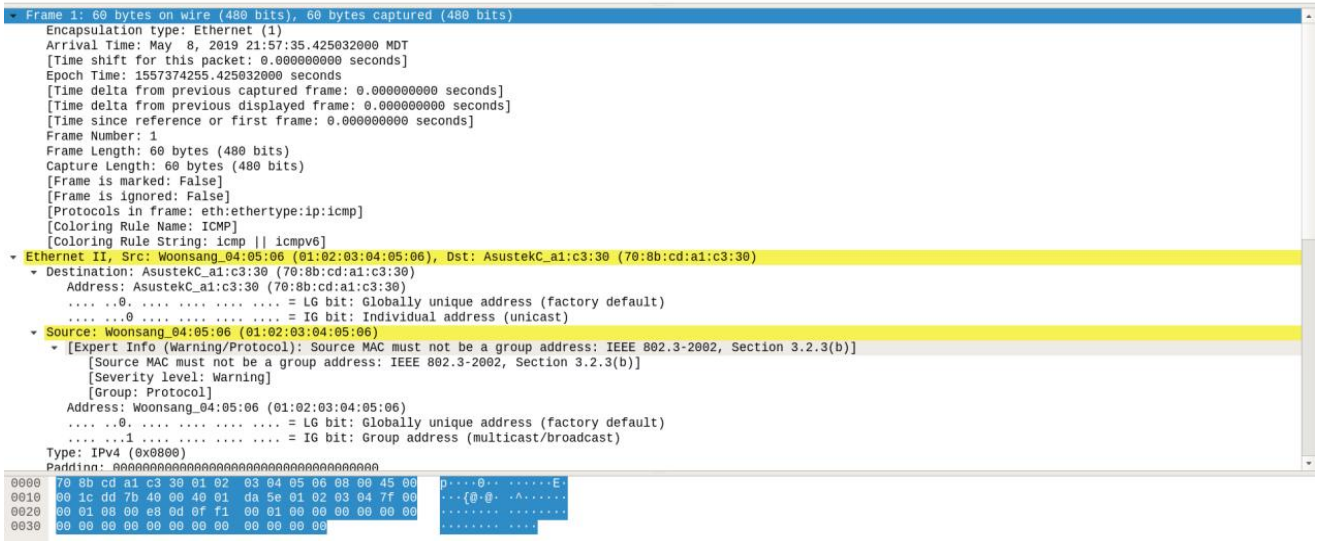
(Wow this is blurry, avert your eyes)

Note: I didn't get permission from my roommate to spoof packets on his behalf. I'm a hacker extraordinaire, I don't need permission.

Task 2.c

The source has been spoofed, and is the second entry highlighted in yellow.





Question 4:
Yes, you can set the packet length to any value, so long as it does not exceed the maximum packet size. There are ways to send packets that are bigger than this, but it is not the norm.

Question 5:
You can, but you can also set it to an arbitrary value and it will still work.

Question 6:
It cannot access the network interfaces if you are not root, which means it can't send packets.