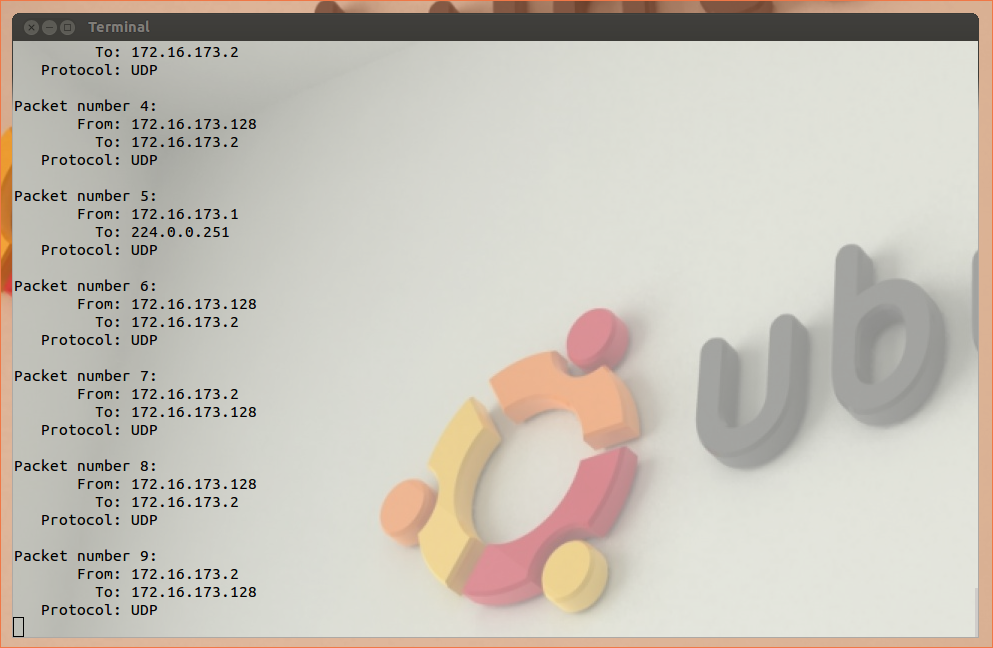
Brian Bauman

**CSC 435 Assignment 6**

*Note: All code used has been submitted with this report in D2L.*

2.1.a

Problem 1

The referenced libraries are *pcap.h* (packet capturing), some standard C libraries for input/output, printing, and error handling, and some sockets/networking libraries.

Problem 2

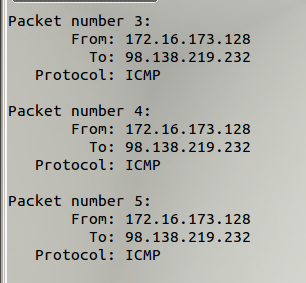
Only the root user is authenticated to monitor traffic across the network interface.

Problem 3

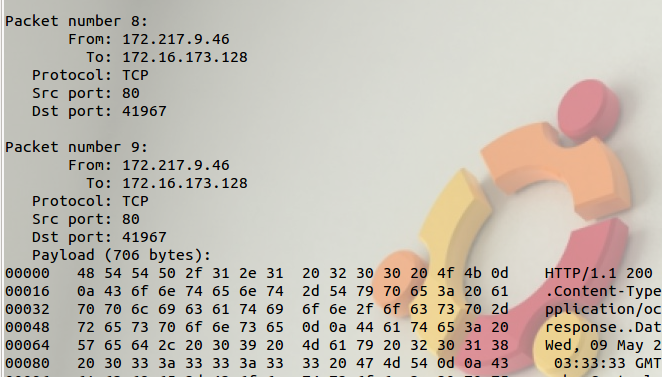
I can’t tell any noticeable difference when changing the source code of sniffer to disable promiscuous mode when calling the pcap library. This is likely because, the way I’m networked, there are no other hosts in my subnet, so no other traffic can be sniffed. Unfortunately, it seems as though my network inferface does not support bridging with my VM, which would allow this. Were this to work, I would see packets sent from my host OS to the internet when sniffing from my VM (promiscuous) or only my own packets (non-promiscuous).

2.1.b

filter expression: “icmp”

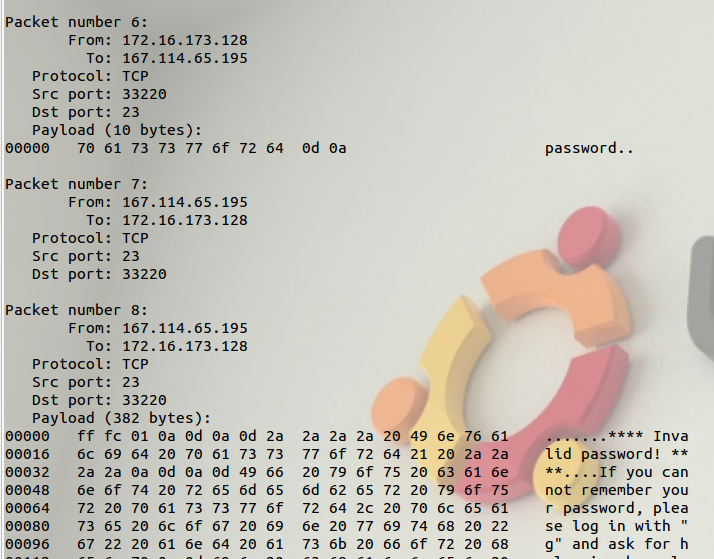


filter expression “tcp portrange 10-100”:



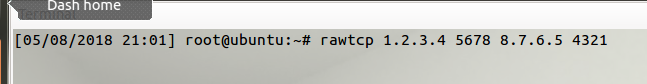
2.1.c

Below is an example of my attempted login to [freechess.org](http://freechess.org)'s telnet server. i entered a username of “username” and a password of “password”, which it rejected:

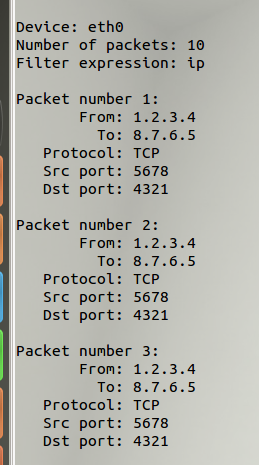


2.2.a

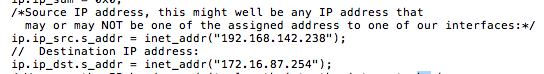
Running this command from an online packet spoofer (arguments are source IP, source port, destination IP, destination port):

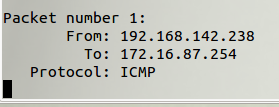


generates these packets:



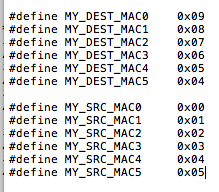
2.2.b

Using a tool downloaded from Github, which lets you choose the source and destination IP addresses in the source code:

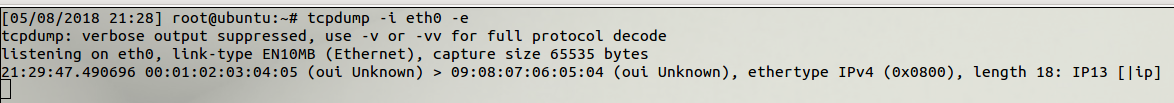
we are able to spoof the ICMP packet and see this from the sniffer:

2.2.c

Running a program (downloaded from the internet, but modified) to send a spoofed ethernet header (snippet shown here):



returns the following when running a tcpdump and inspecting the MAC address information:



You can see that it respects the input source/destination MAC addresses.

Question 4

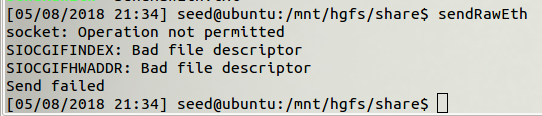
Yes, this property can be modified to be any value.

Question 5

No, you can send an incorrect checksum. However, the receiving process will likely discard the packet in the case that the checksum does not match.

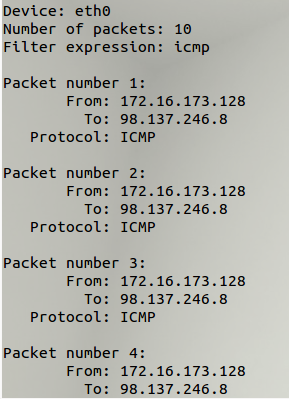
Question 6

Only the root user has the permission required to interact with the network interfaces. The following error arises when trying to use raw sockets as another user:

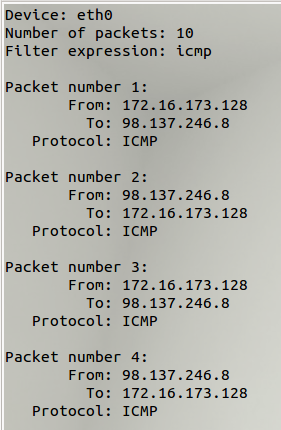


3

Before implementing the sniff and spoof program, a capture of ICMP packets (generated by pinging [yahoo.com](http://yahoo.com)), looked like:



As you can see, there were no replies. With a sniff-and-spoof program from the internet running on another VM, we now see this when sniffing for ICMP packets after beginning to ping [yahoo.com](http://yahoo.com):



Now there are responses for each ping (being spoofed by our program on VM B). Here is the output of that program:

