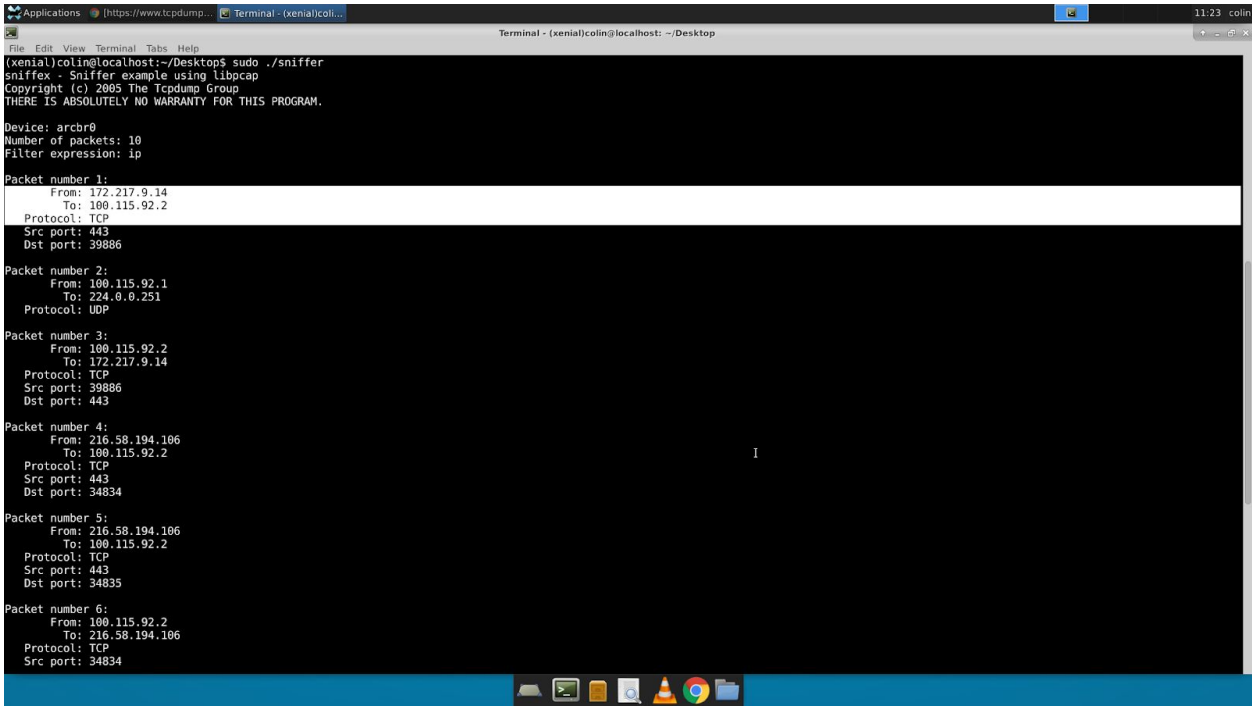
Upon running sniffex.c and the program it created, I was able to obtain the following packet capture.



The library calls needed to be able run a sniffer program are pcap_lookupdev(), which finds a valid network interface to sniff, pcap_open_live(), which creates a sniffing session with the selected device, pcap_datalink() if you want to use link-layer headers, pcap_compile() to compile the sniffing program, pcap_setfilter(), which allows for the filtering out of unspecified packets by type, pcap_next(), which actually does the packet capturing, pcap_loop(), which is a callback function to allow for multiple packet captures, and pcap_close(), which ends the connection with the network interface and stops packet collection.

Running sniffex requires root access because it interfaces with a hardware device which requires root access to do. The code that causes it to fail is when pcap_lookupdev() is called. This function is what establishes the connection with the network interface and since no root access is had, the function returns NULL and the program fails.

```
Terminal - (xenial)colin@localhost: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
(xenial)colin@localhost:~/Desktop$ ./sniffer
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: arcbr0
Number of packets: 10
Filter expression: ip
Couldn't open device arcbr0: arcbr0: You don't have permission to capture on tha
t device (socket: Operation not permitted)
(xenial)colin@localhost:~/Desktop$
```

When promiscuous mode is on, the sniffer intercepts all traffic that the network interface finds and not just the traffic that was intended to be sent to the network interface. By default, sniffex sniffs in promiscuous mode. Below are screen captures of runs with and without promiscuous mode on.

```
Terminal - (xenial)colin@localhost: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
(xenial)colin@localhost:~/Desktop$ sudo ./promiscuous_on
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: arcbr0
Number of packets: 3
Filter expression: ip

Packet number 1:
       From: 216.58.193.138
         To: 100.115.92.2
   Protocol: TCP
   Src port: 443
   Dst port: 56564

Packet number 2:
       From: 100.115.92.2
         To: 216.58.193.138
   Protocol: TCP
   Src port: 56564
   Dst port: 443

Packet number 3:
       From: 216.58.193.142
         To: 100.115.92.2
   Protocol: TCP
   Src port: 443
   Dst port: 40200

Capture complete.
(xenial)colin@localhost:~/Desktop$
```

```
Terminal - (xenial)colin@localhost: ~/Desktop
File  Edit  View  Terminal  Tabs  Help

Device: arcbr0
Number of packets: 3
Filter expression: ip

Packet number 1:
       From: 100.115.92.1
         To: 224.0.0.251
   Protocol: UDP

Packet number 2:
       From: 100.115.92.1
         To: 224.0.0.251
   Protocol: UDP

Packet number 3:
       From: 100.115.92.1
         To: 224.0.0.251
   Protocol: UDP

Capture complete.
(xenial)colin@localhost:~/Desktop$
```

In order to filter the packets, I modified the filer_exp[] char array to specify the kind of traffic that I want to intercept. Below is the output of my ICMP filter and my FTP filter. For my

ICMP filter, I simply pinged my host device with my VM, and for my FTP filter, I just navigated to an HTTP webpage since HTTPS is port 443 and not port 80.

Terminal - (xenial)colin@localhost: ~/Desktop

File Edit View Terminal Tabs Help

```
(xenial)colin@localhost:~/Desktop$ sudo ./tcp
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: arcbr0
Number of packets: 3
Filter expression: tcp dst portrange 10-100

Packet number 1:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Packet number 2:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Packet number 3:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Capture complete.
(xenial)colin@localhost:~/Desktop$
```

```
Terminal - (xenial)colin@localhost: ~/Desktop
File  Edit  View  Terminal  Tabs  Help
(xenial)colin@localhost:~/Desktop$ sudo ./tcp
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: arcbr0
Number of packets: 3
Filter expression: tcp dst portrange 10-100

Packet number 1:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Packet number 2:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Packet number 3:
       From: 192.168.1.2
         To: 192.139.46.66
   Protocol: TCP
   Src port: 50582
   Dst port: 80

Capture complete.
(xenial)colin@localhost:~/Desktop$
```

I set up telnet between my VM and my host device and entered my root password, of which the first two characters are ‘3’ and ‘l’. Below is the output of my sniffing program when performing a sniff on the telnet connection.

Terminal - (xenial)colin@localhost: ~/Desktop

File Edit View Terminal Tabs Help

```
[09/24/2015 18:41] seed@ubuntu:~/Downloads$ sudo ./telnet
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: eth0
Number of packets: 10
Filter expression: tcp port 23

Packet number 1:
       From: 192.168.56.1
         To: 192.168.1.2
   Protocol: TCP
   Src port: 58572
   Dst port: 23
   Payload (1 bytes):
00000   33                                               3

Packet number 2:
       From: 192.168.1.2
         To: 192.168.56.1
   Protocol: TCP
   Src port: 23
   Dst port: 58572

Packet number 3:
       From: 192.168.56.1
         To: 192.168.1.2
   Protocol: TCP
   Src port: 58572
   Dst port: 23
   Payload (1 bytes):
00000   6c                                               l

Packet number 4:
       From: 192.168.1.2
         To: 192.168.56.1
   Protocol: TCP
   Src port: 23
   Dst port: 58572

Capture complete.
(xenial)colin@localhost:~/Desktop$
```