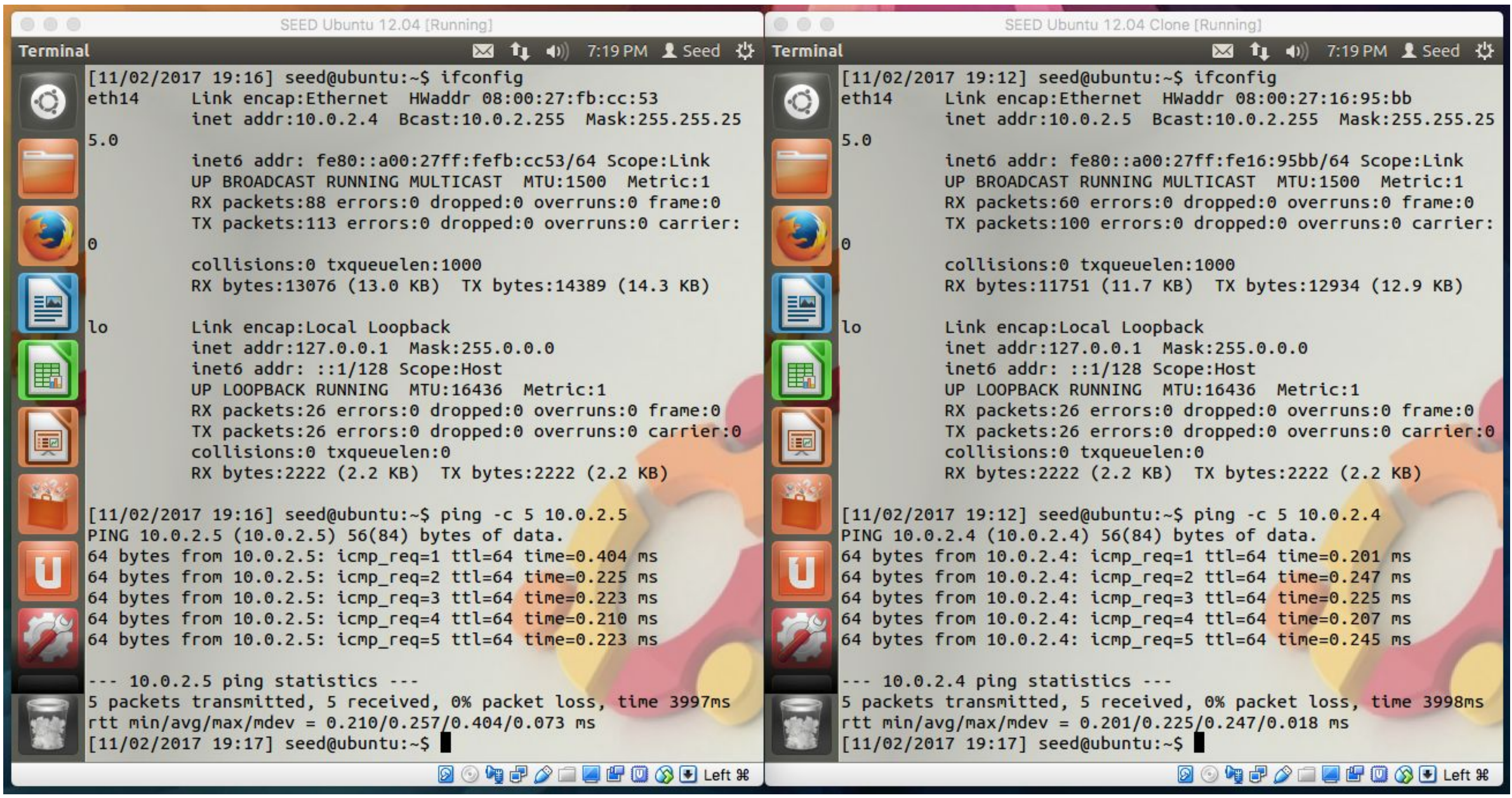
Problem 1

VM 1:

```
eth14 inet 10.0.2.4
ping: 5 transmitted, 5 recieved, 0% packet loss, time 3997ms
```

VM 2:

```
eth14 inet 10.0.2.5
ping: 5 transmitted, 5 recieved, 0% packet loss, time 3998ms
```



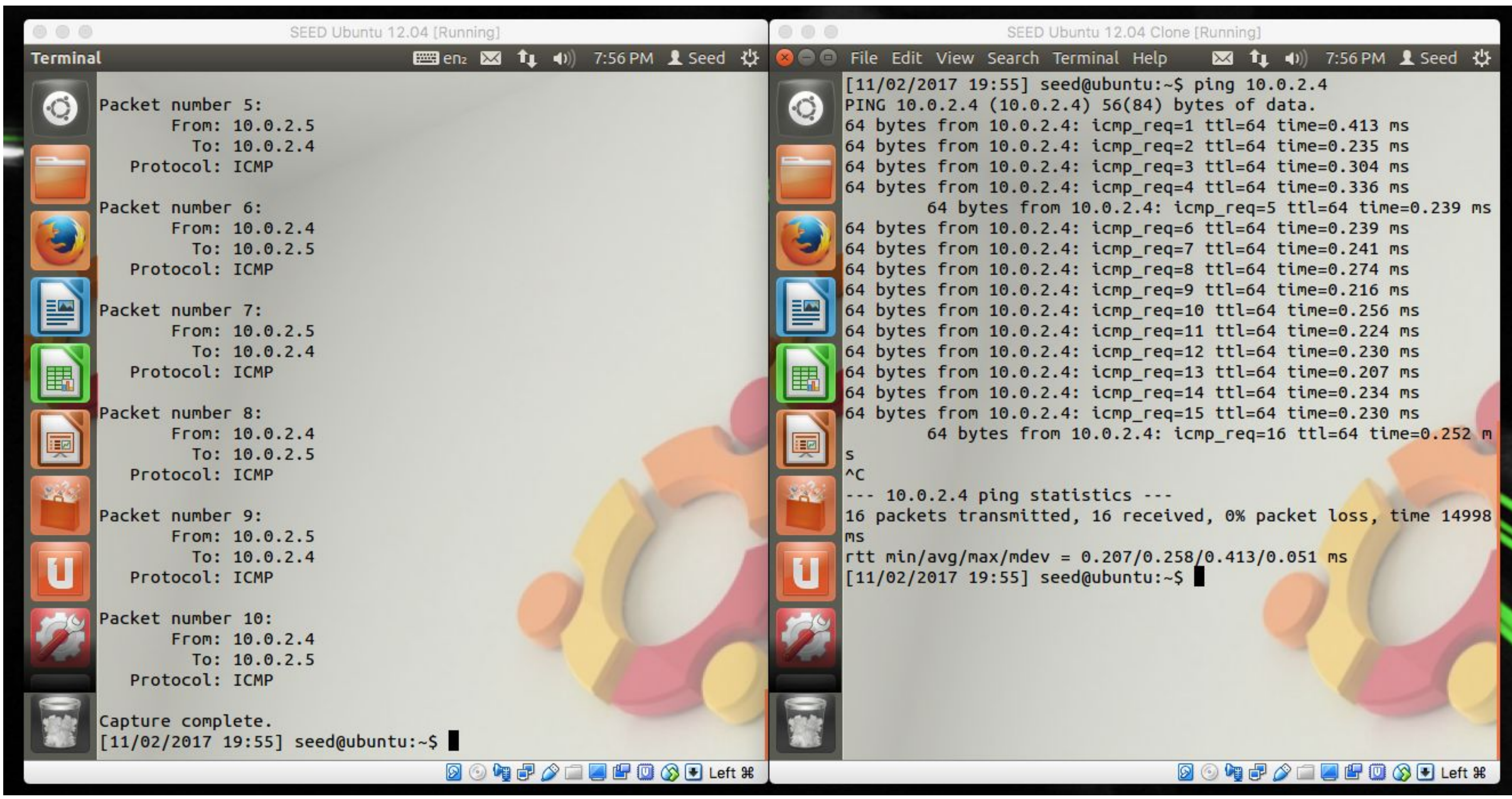
Problem 2

To begin using the pcap library, you open a sniffing device by calling the `pcap_open_live()` method. We can use `pcap_compile()` and `pcap_setfilter()` to filter out packets we don't want. We can use this to listen to only traffic on port 22 (SSH) for example. One of the ways we can sniff for packets is by using `pcap_next()` to iteratively capture packets. Alternatively, we can use `pcap_loop()` to continuously sniff packets.

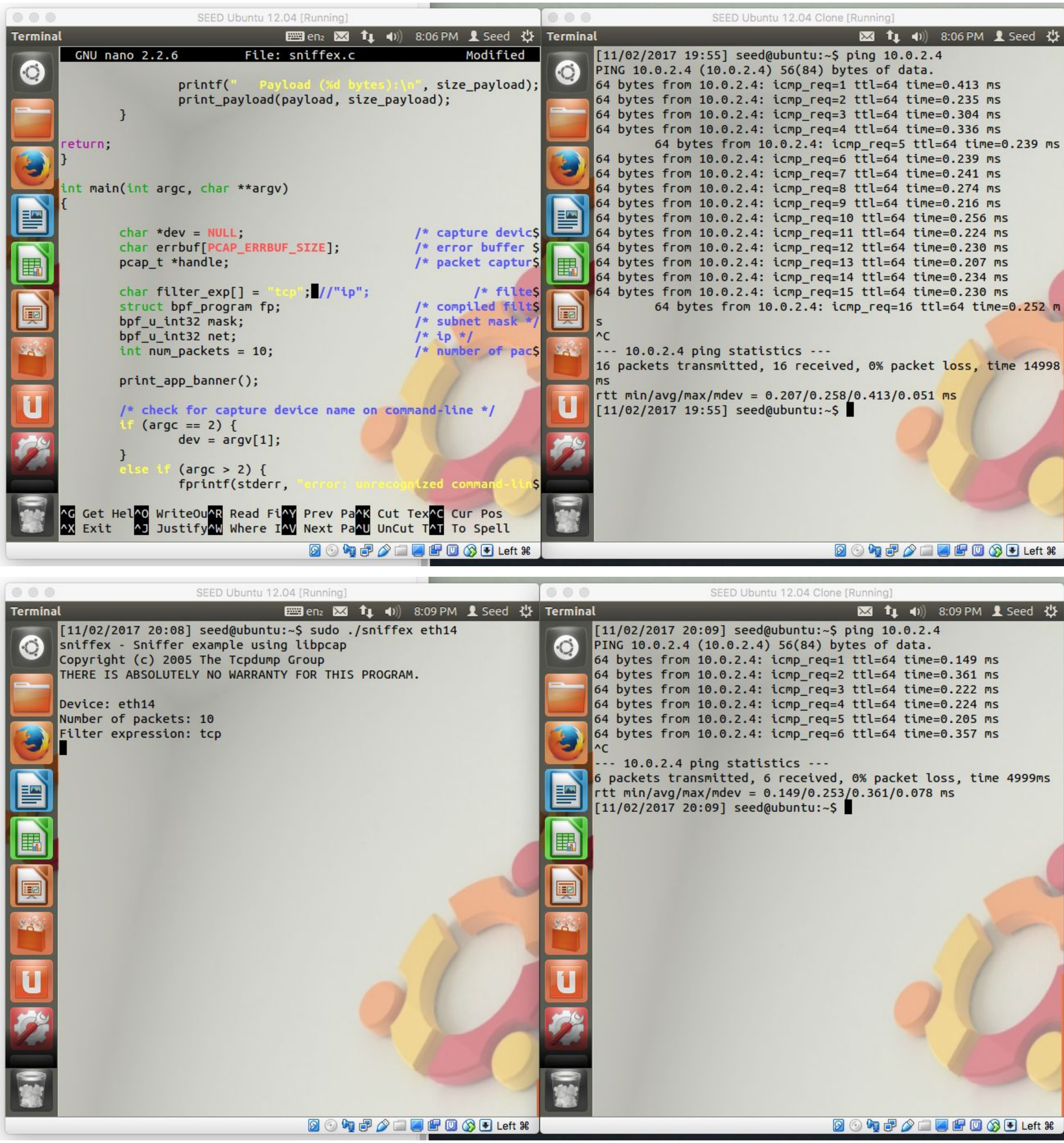
When starting sniffex, we need to run sniffex with sudo to grant it root access (to access lower level networking stuff).

Sniffing on VM 1 while pinging VM 1 from VM 2 results in the following:

```
Packet number 1:
       From: 10.0.2.4
         To: 10.0.2.5
   Protocol: ICMP
```

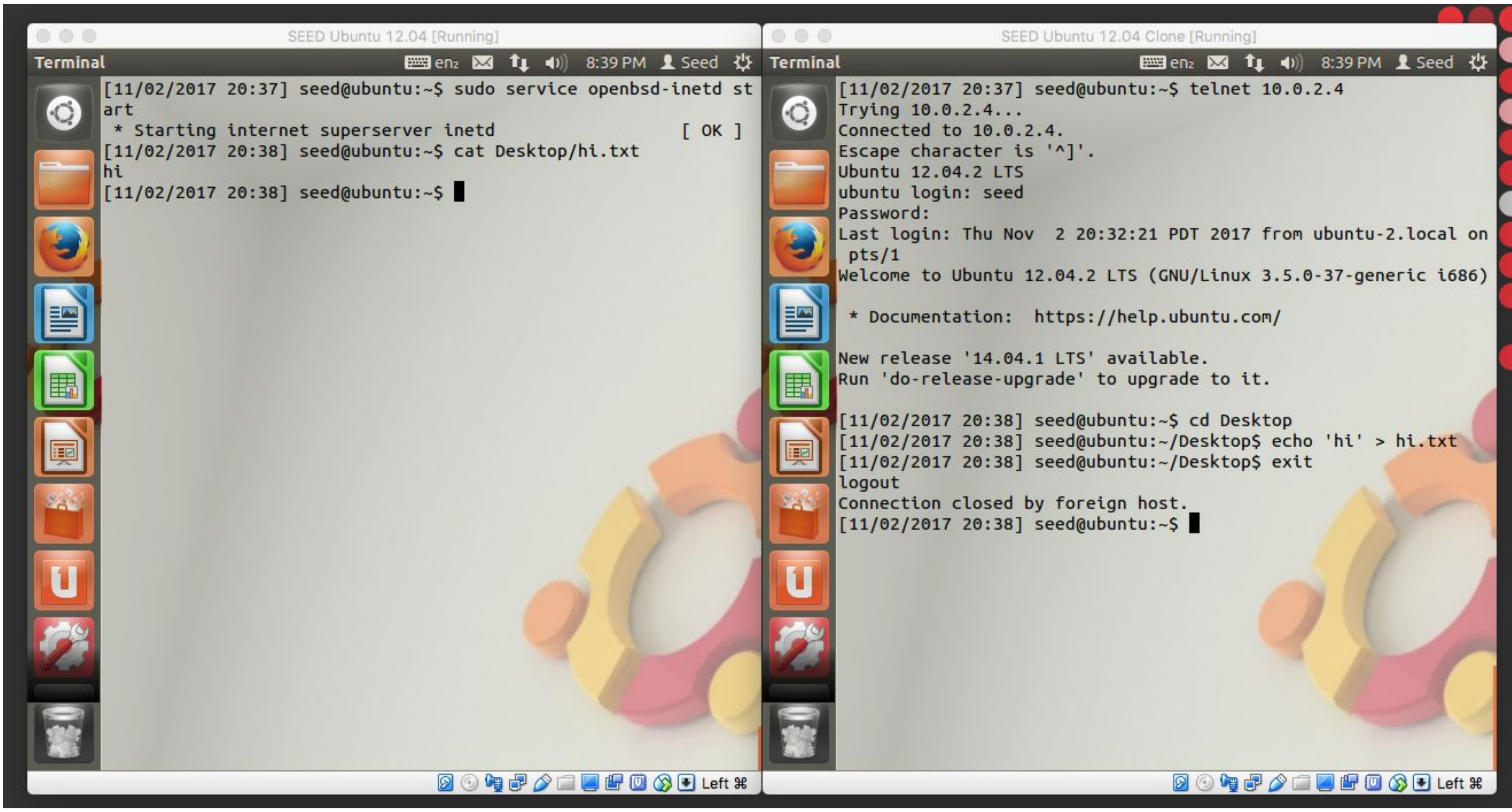


After changing the filter from "ip" to "tcp", we no longer capture the ICMP packets.



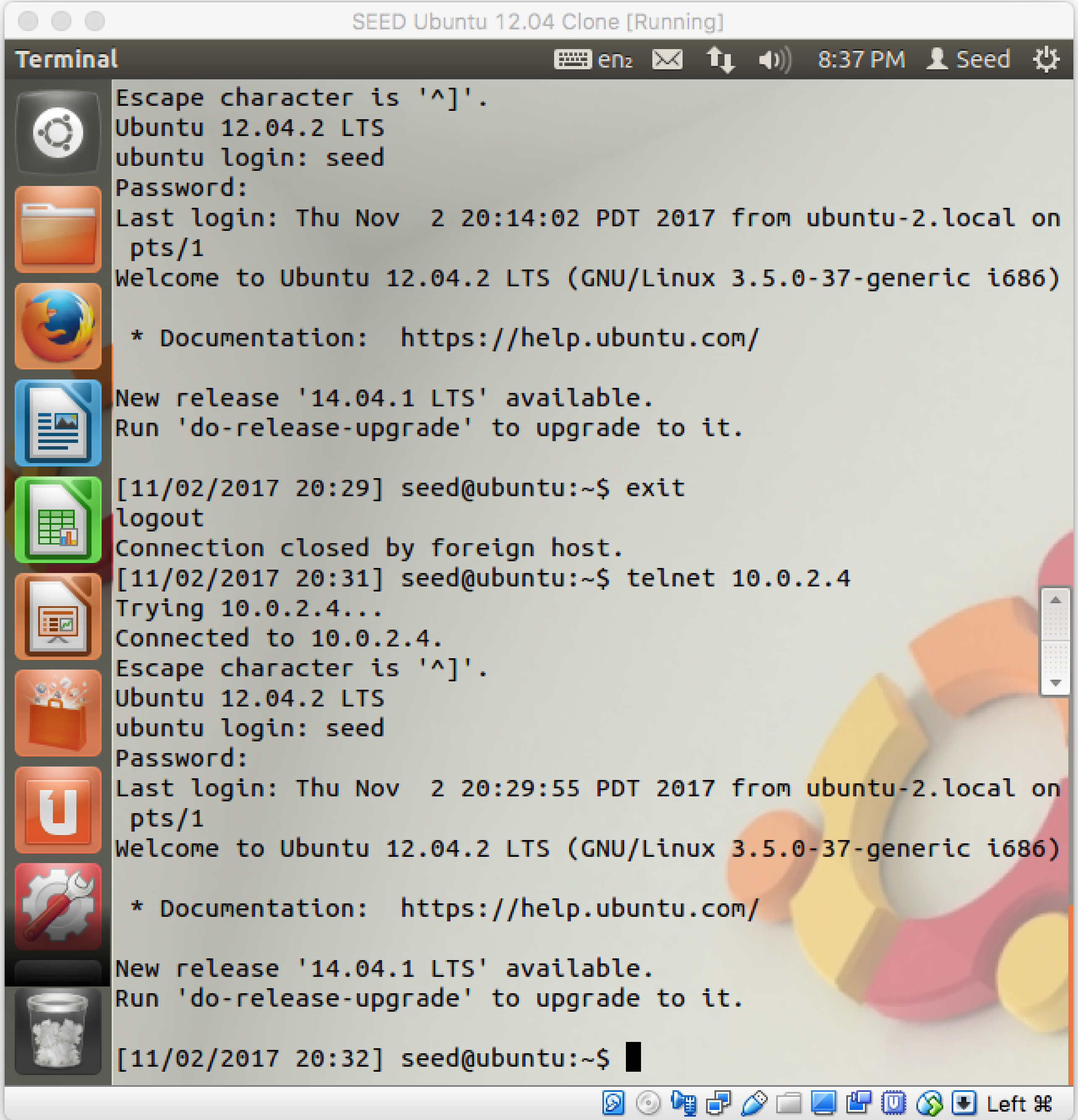
Problem 3

Below is the output for logging in with telnet, creating a file, and verifying the file's existence.

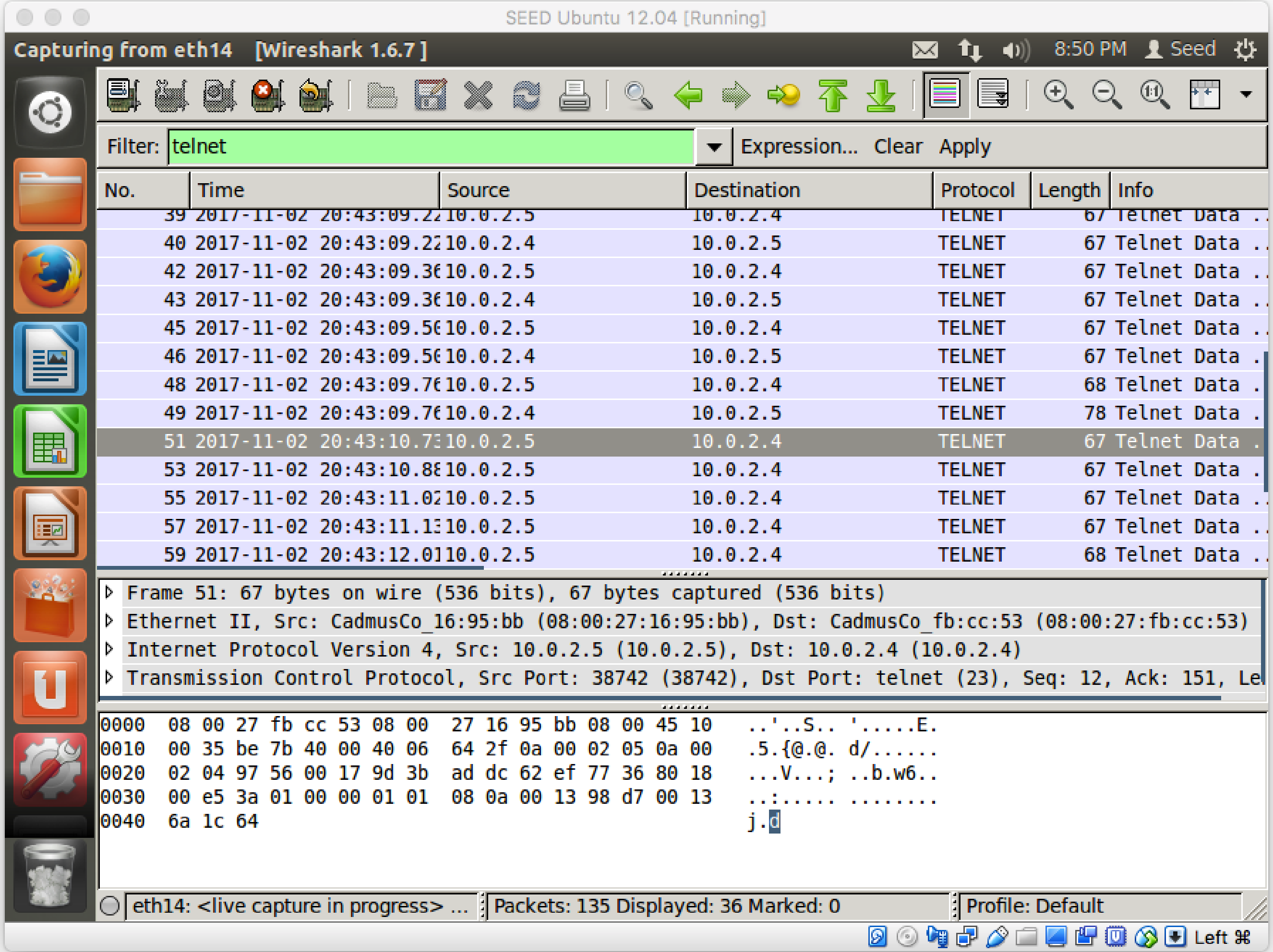


Now we do the same, but capture the packets during login with sniffex. As we can see in the below screenshots, you can see in plain text the password (sent in packets number 33-39)

```
Packet number 33:
       From: 10.0.2.5
         To: 10.0.2.4
   Protocol: TCP
   Src port: 38742
   Dst port: 23
   Payload (1 bytes):
00000   64                                                d

Packet number 34:
       From: 10.0.2.4
         To: 10.0.2.5
   Protocol: TCP
   Src port: 23
   Dst port: 38742

Packet number 35:
       From: 10.0.2.5
         To: 10.0.2.4
   Protocol: TCP
   Src port: 38742
   Dst port: 23
   Payload (1 bytes):
00000   65                                                e

Packet number 36:
       From: 10.0.2.4
         To: 10.0.2.5
   Protocol: TCP
   Src port: 23
   Dst port: 38742

Packet number 37:
       From: 10.0.2.5
         To: 10.0.2.4
   Protocol: TCP
   Src port: 38742
   Dst port: 23
   Payload (1 bytes):
00000   65                                                e

Packet number 38:
       From: 10.0.2.4
         To: 10.0.2.5
   Protocol: TCP
   Src port: 23
   Dst port: 38742

Packet number 39:
       From: 10.0.2.5
         To: 10.0.2.4
   Protocol: TCP
   Src port: 38742
   Dst port: 23
   Payload (1 bytes):
00000   73                                                s
```



Using Wireshark, we find the password characters in the last byte of the TELNET packets of length 67 as shown below.



This plaintext password communication is an obvious concern, as using telnet means potentially giving an attacker remote root access to your machine.

Problem 4

If we log in with SSH, we can see SSH traffic, however the password is clearly encrypted. Instead of plaintext bytes being sent, we see several key exchange packets along with other encrypted traffic. This is obviously a much more secure method of communication.