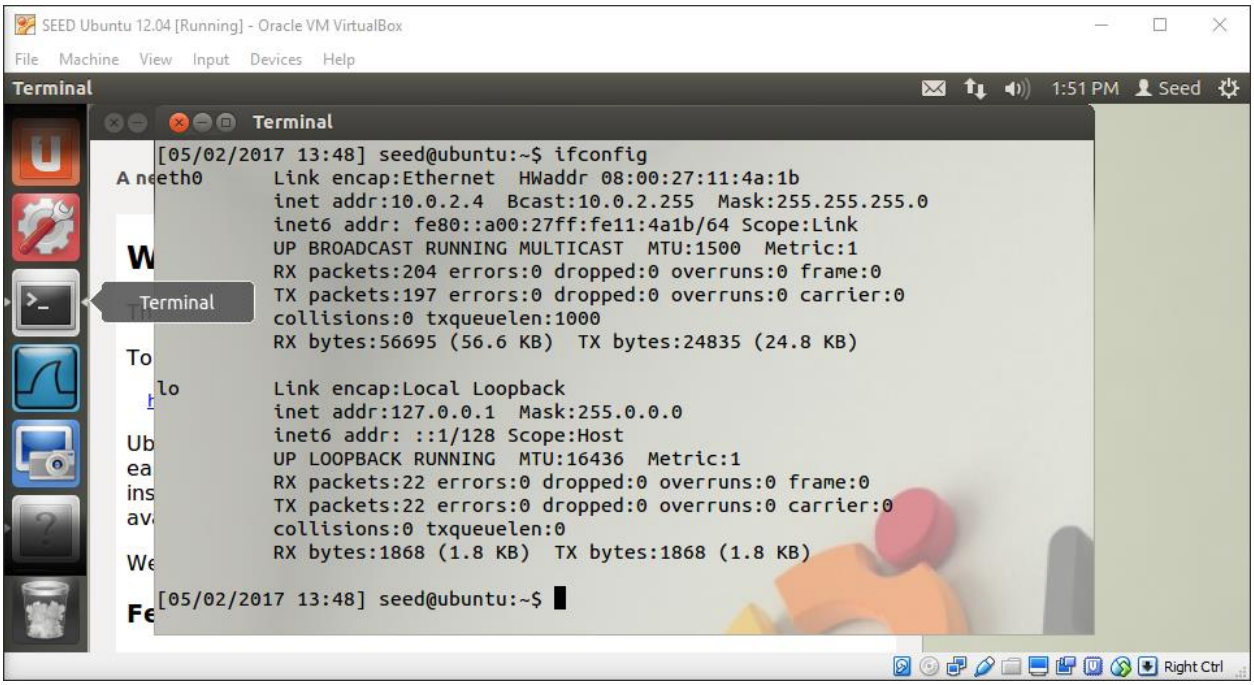
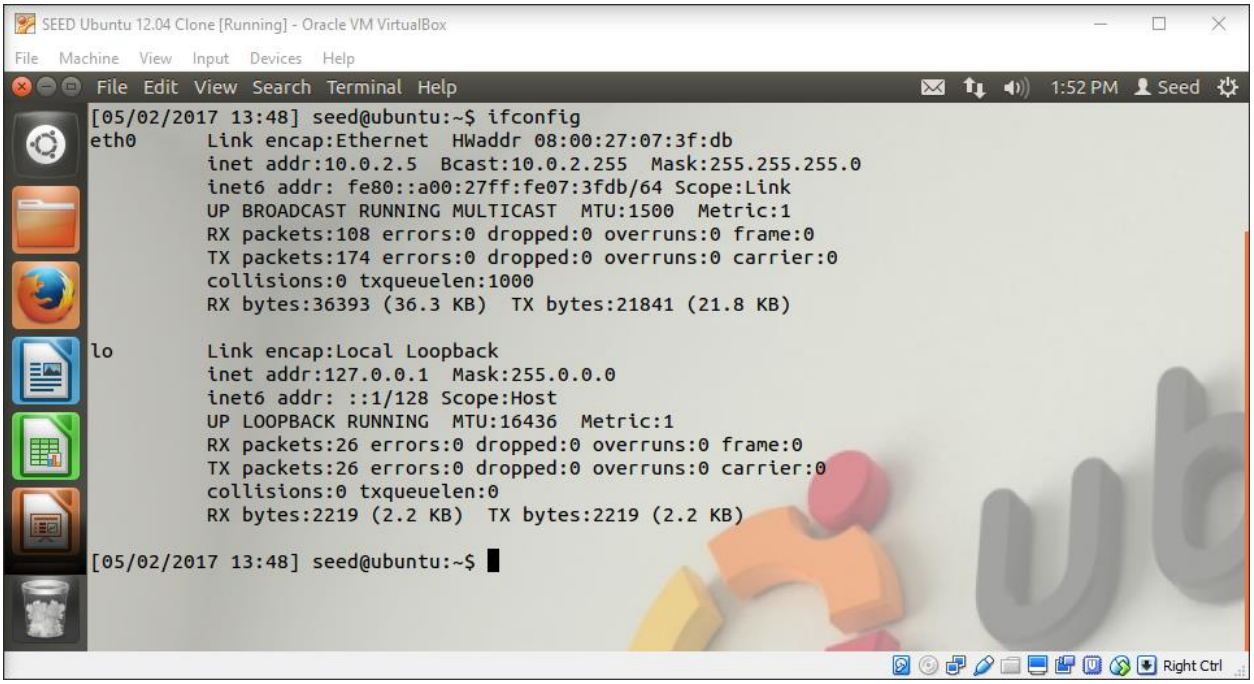
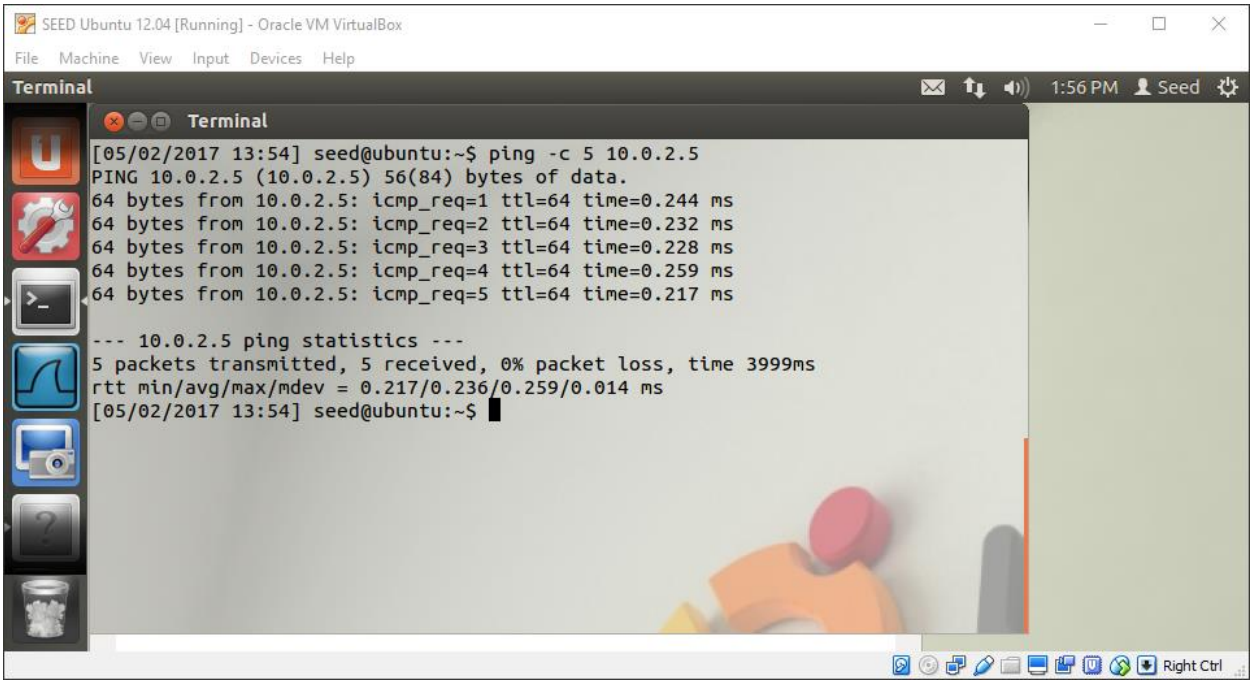
Michael Acosta

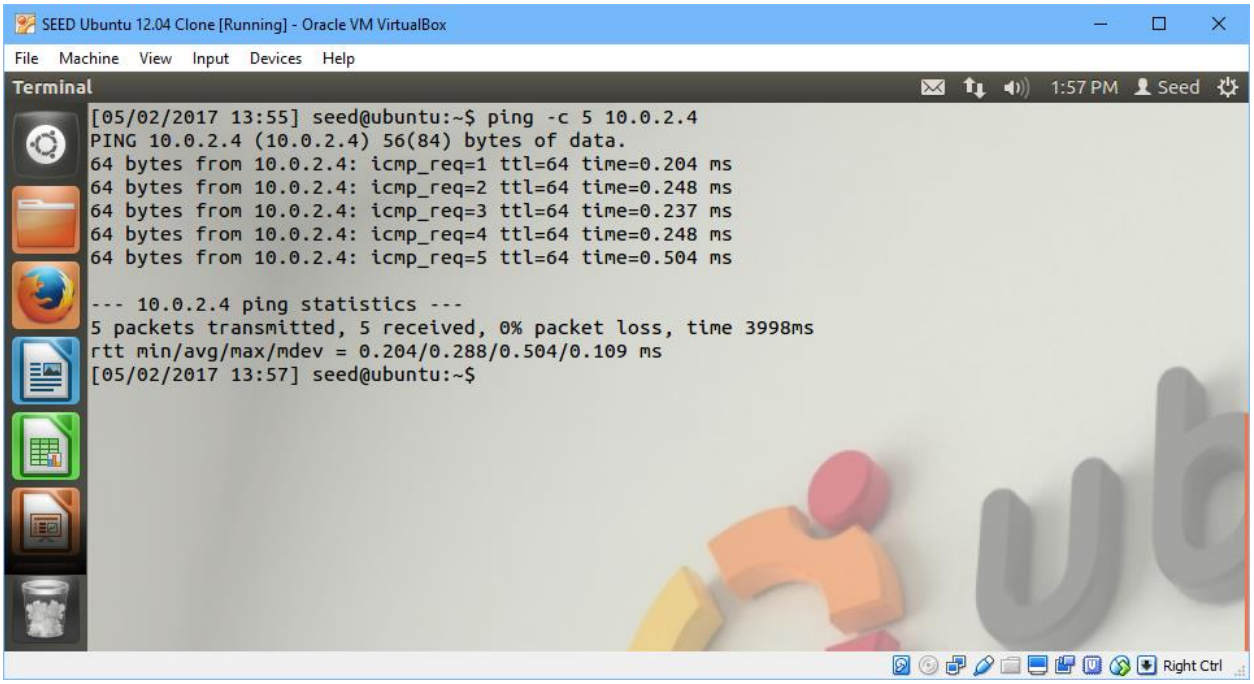
https://github.com/m-acosta/cs380-exercise5

Problem 1







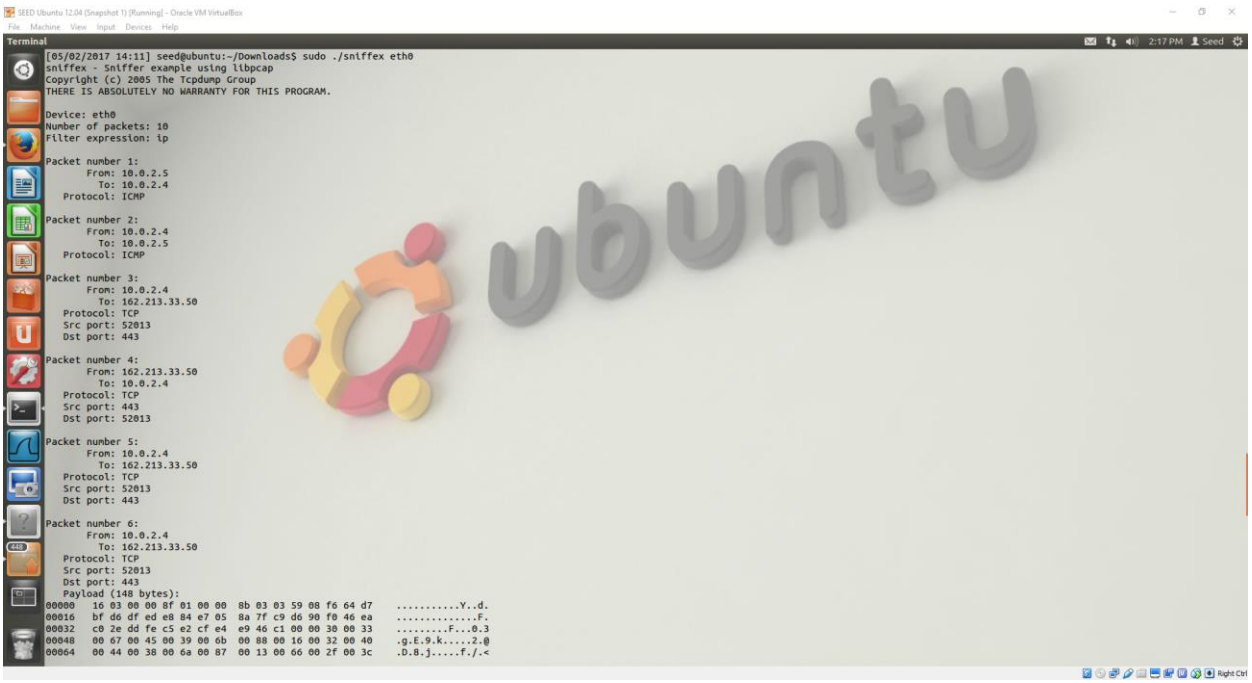


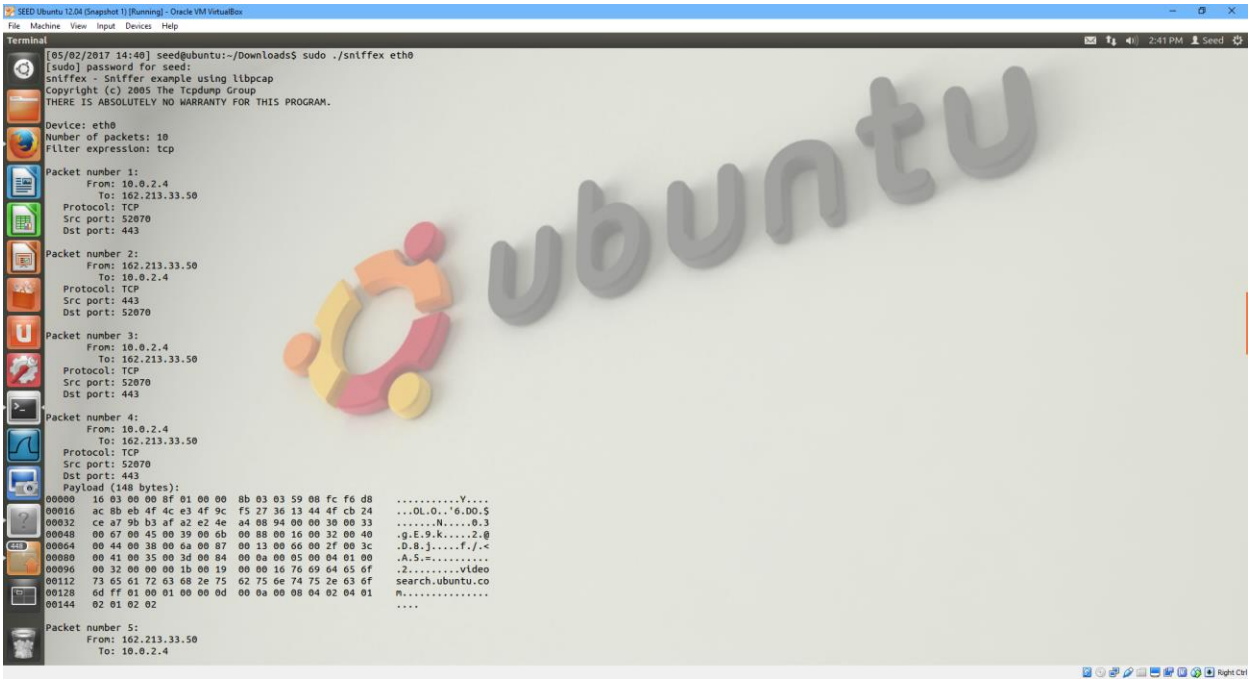
Problem 2

The interface on which to sniff is specified by the string in the command line, and that is initialized with pcap. Each time the sniffer is run, it has a different file handle to differentiate between devices and runs. The type of traffic can be set with a filter, and then handled with the

pcap compiler. In the main execution loop, pcap will wait for packets (specified by the filter) and that packet is printed to the user in a manner they can see.

When running the ./sniffex ethX command, it fails because it does not have permission to capture on that socket. The operation is not permitted.

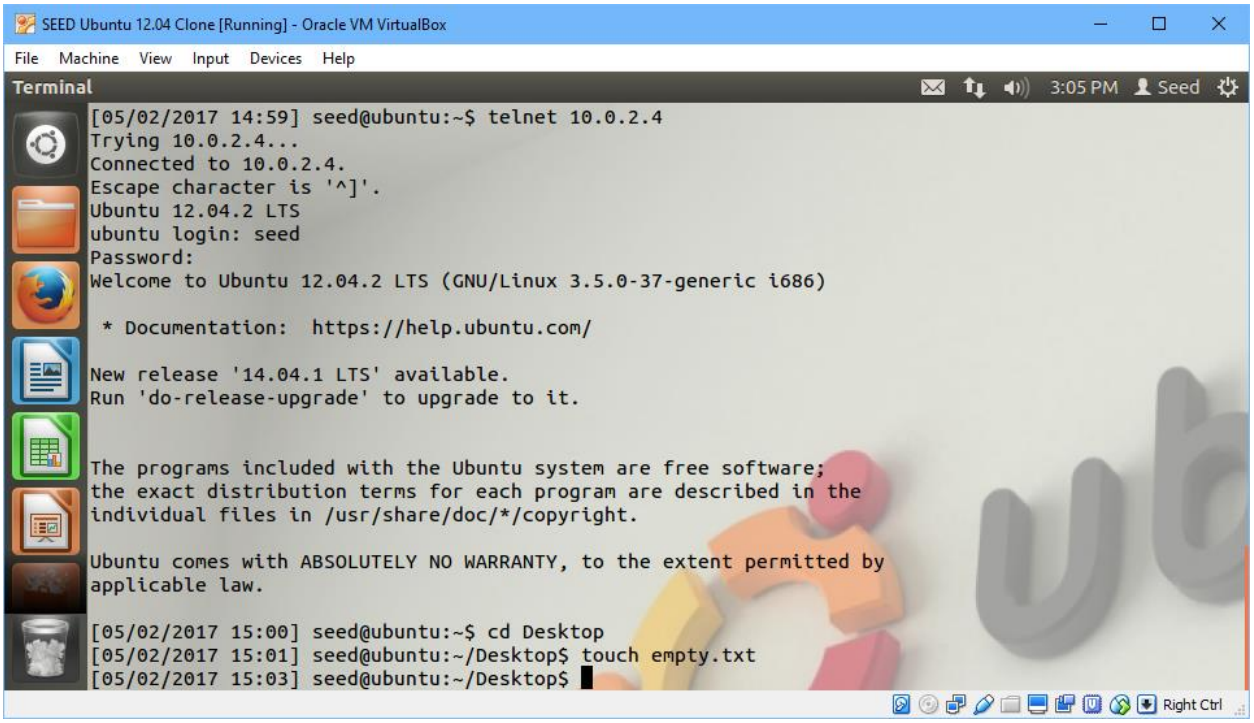


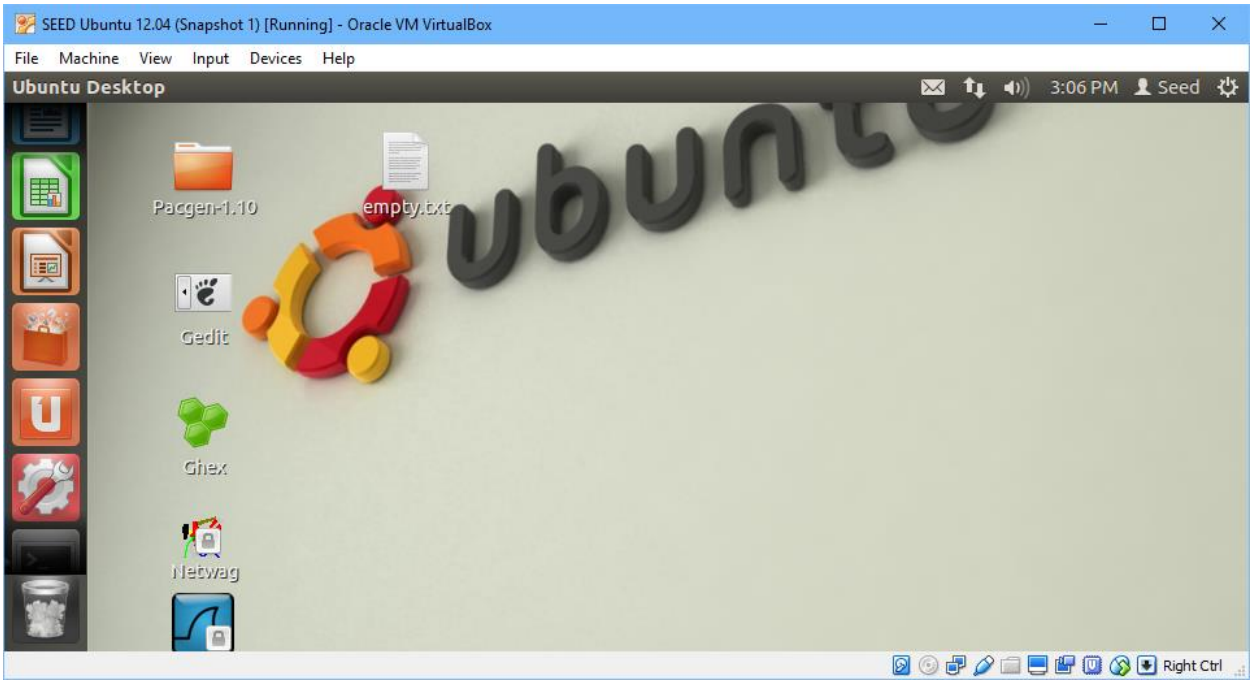


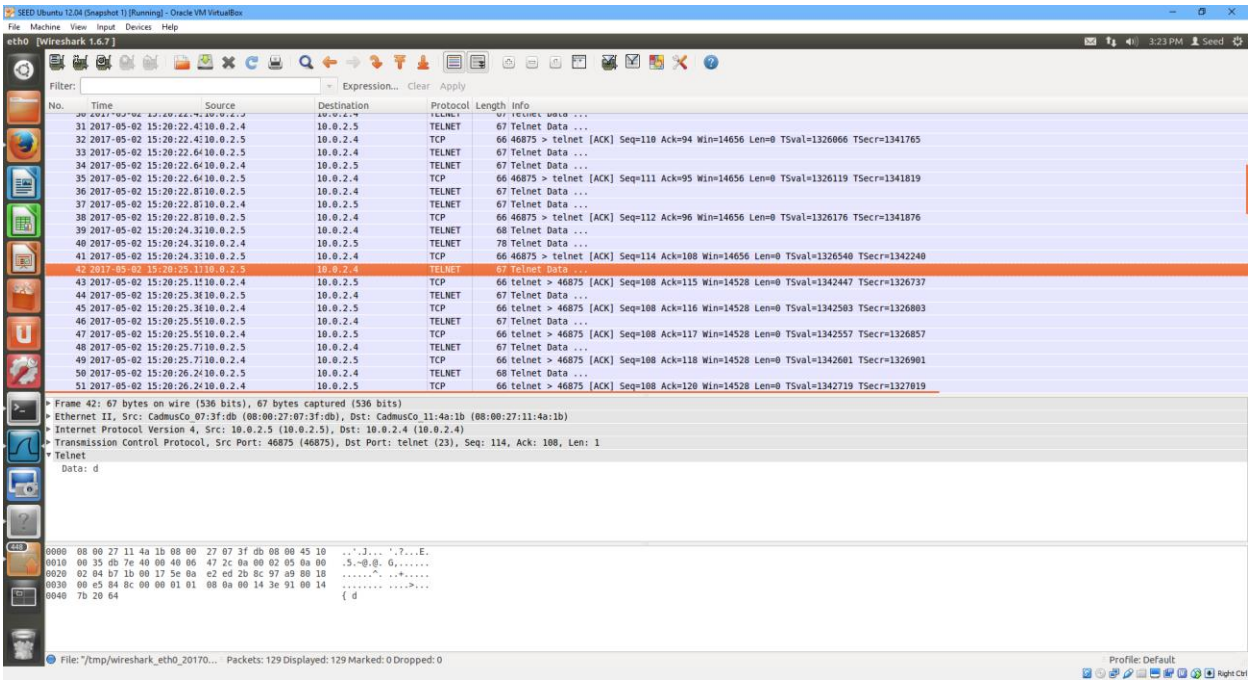
With the ip filter expression, a packet is received of type ICMP from 10.0.2.5 which is the clone machine. A packet is then sent back to that address as a confirmation. Then there are more packets (TCP) to and from the address 162.213.33.50.

With the tcp filter, the packets from the clone machine don't get reported by the sniffer. Only the packets from the 162.213.33.50 are shown. This is because the packets aren't sent by the TCP protocol from the other machine.

Problem 3

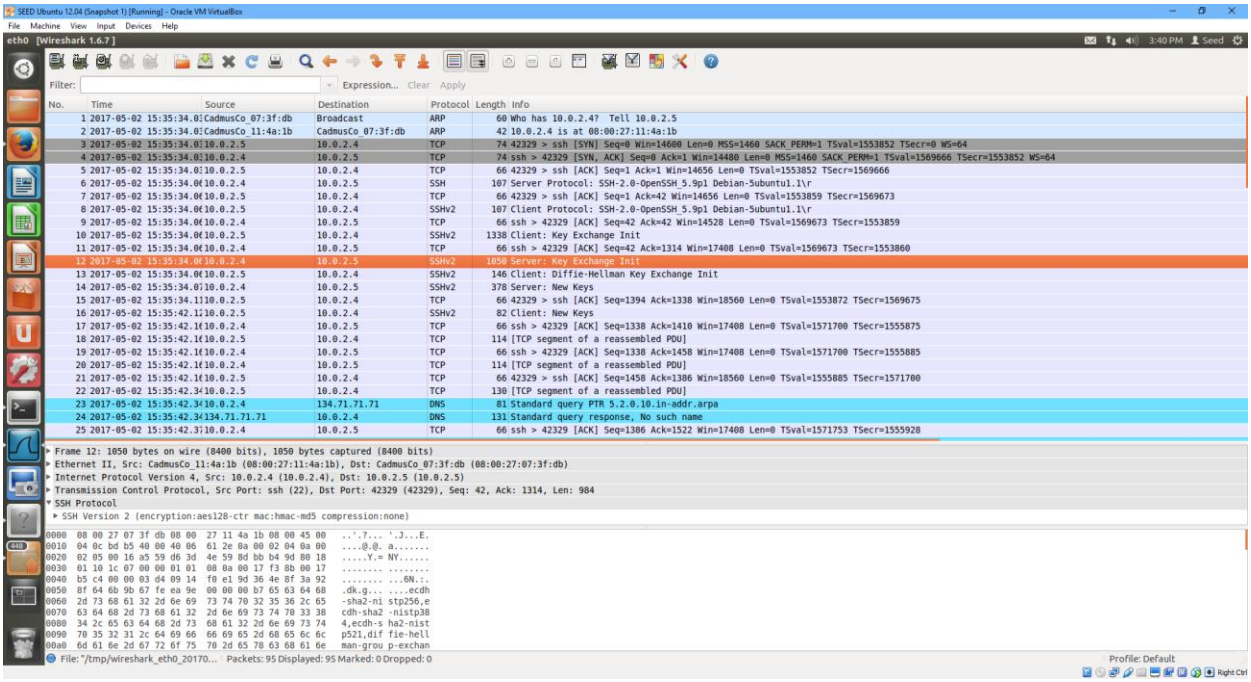



SEED Ubuntu 12.04 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Ubuntu Desktop
3:06 PM Seed
Pacgen-1.10
empty.txt
Gedit
Ghex
Netwag
Right Ctrl



The password, dees, shows up with both packet sniffers. In the Wireshark window, the three telnet packets below the selected one are the remaining: ees. Telnet does not encrypt this password data at all, the packets are sent as is. So a sniffer set up on a telnet server can capture every user's password which is very unsafe.

Problem 4



Wireshark shows the SSH and SSHv2 traffic, and then the following TCP packets but none of them show the password information at all. The packets are all encrypted, even past the header.