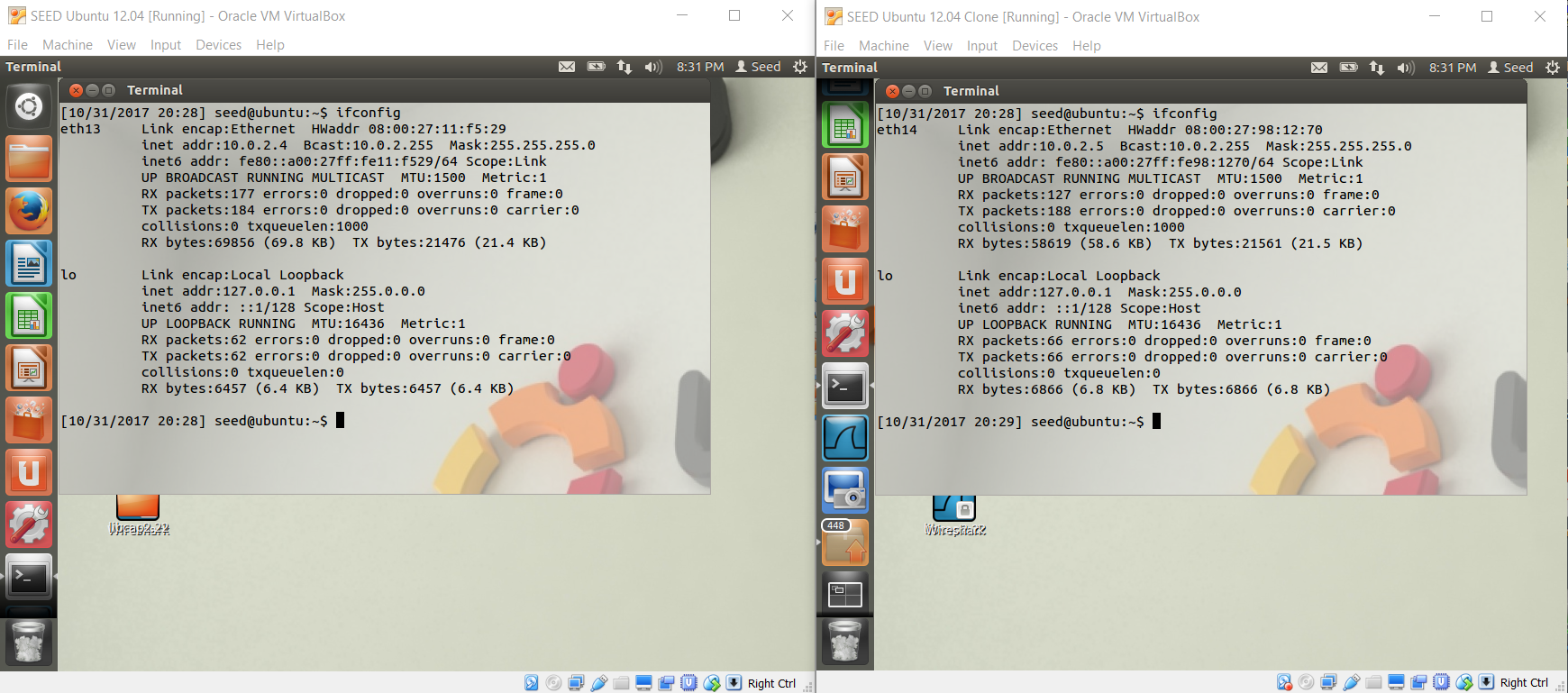
CS380 Exercise 5

Ethan Smith

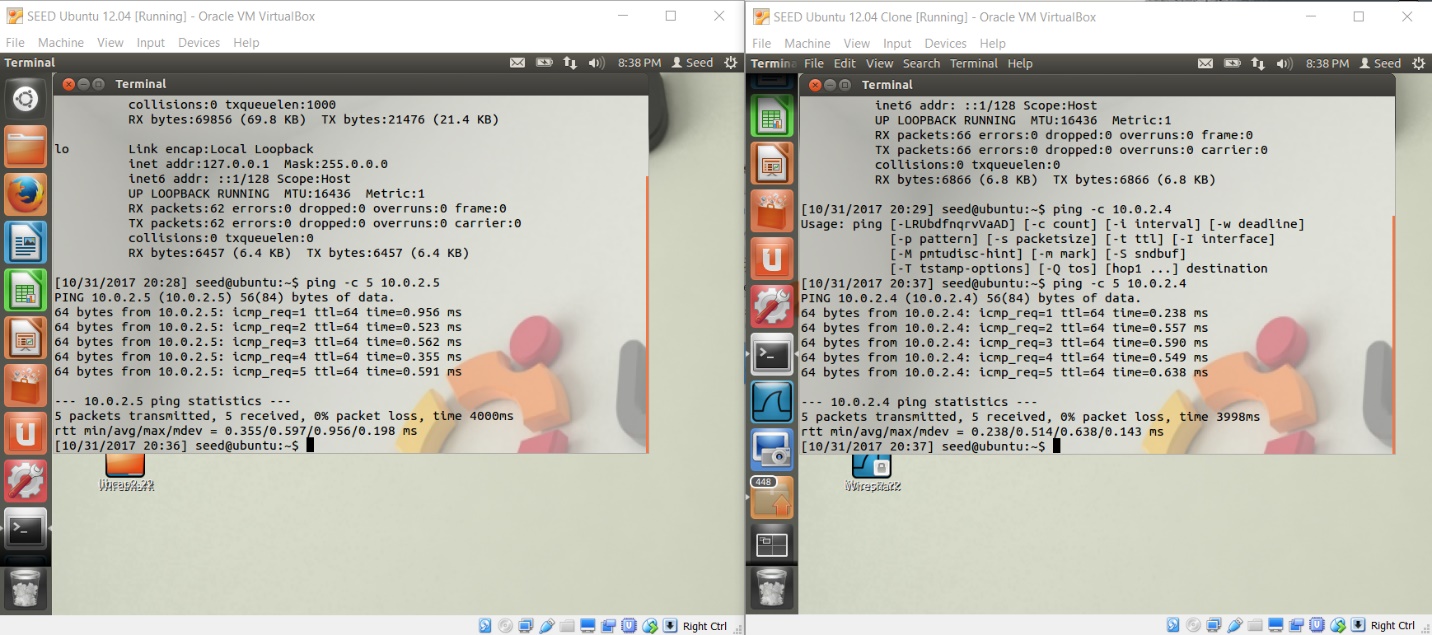
David Shin

GitHub URL: https://github.com/davidws22/CS380-EX5

ifconfig

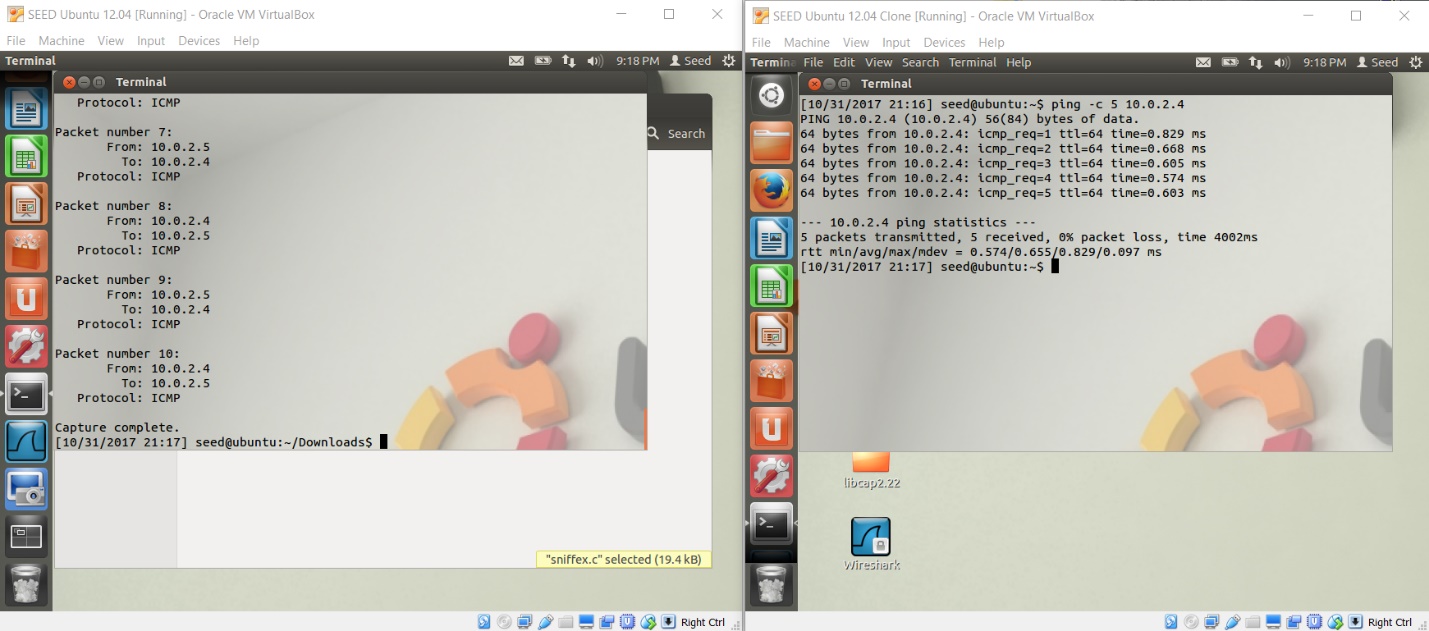


Ping 5

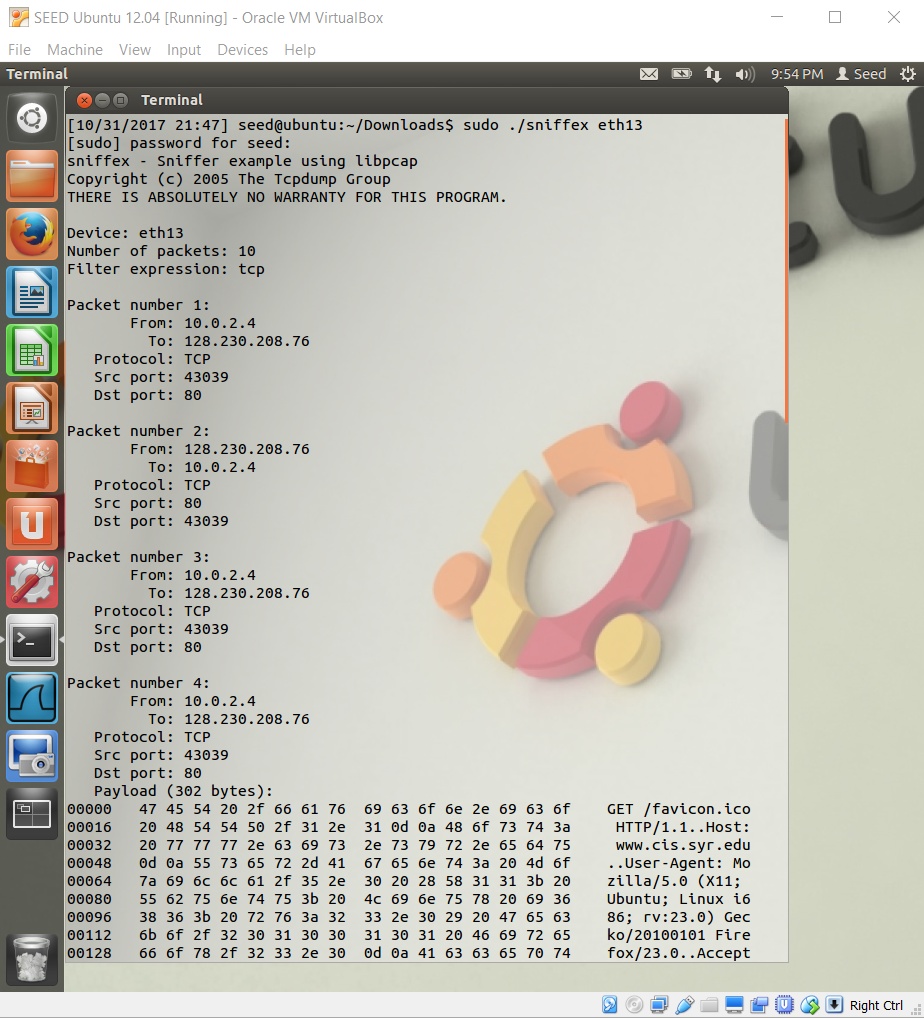


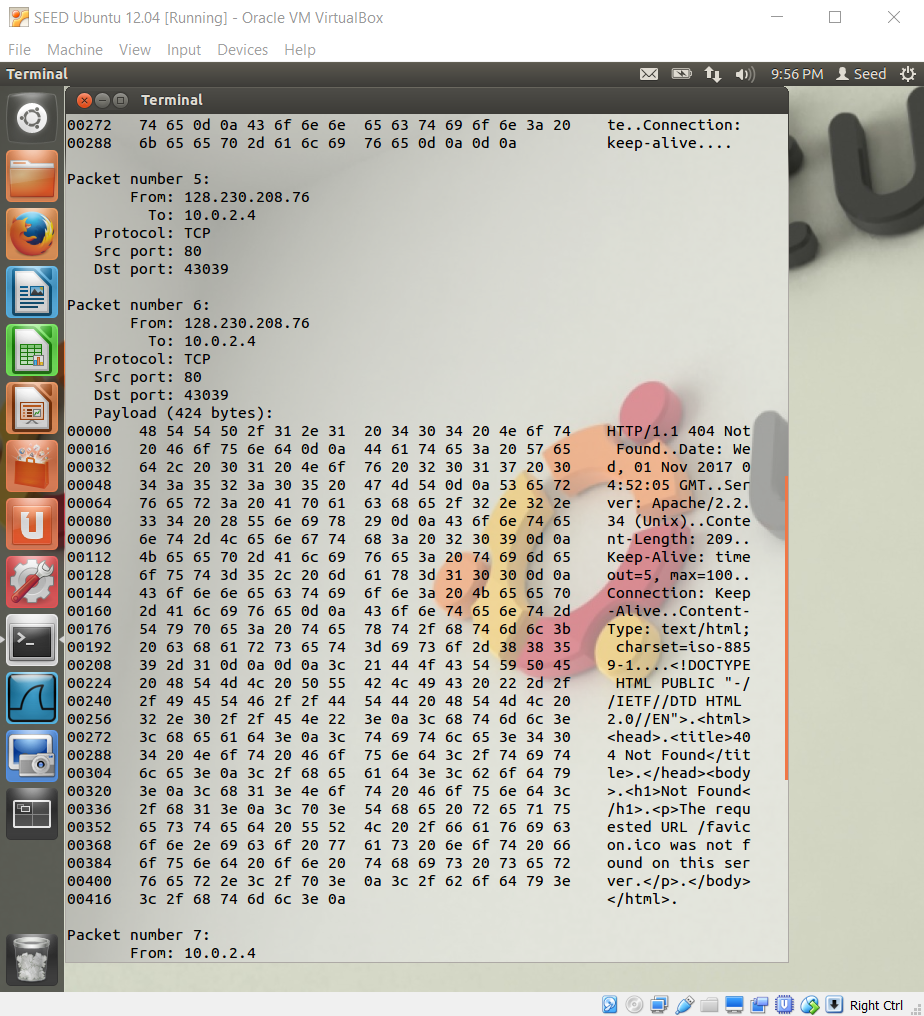
Use of the PCAP library is generally comprised of five steps. First we need to specify which interface we need to sniff on. In C, we specify which interface we would like to sniff on by providing a command-line argument. In our case it was eth14. Second, we need to set up a valid file handle by calling the pcap\_open\_live() function, which returns a session handler. Third, we specify which type of packet to sniff for whether it may be TCP or IP. Fourth, we will loop until n number of packets has been saved by using the function pcap\_next(), which returns a pointer to an unsigned char. Finally, pcap closes its session and ends the program.

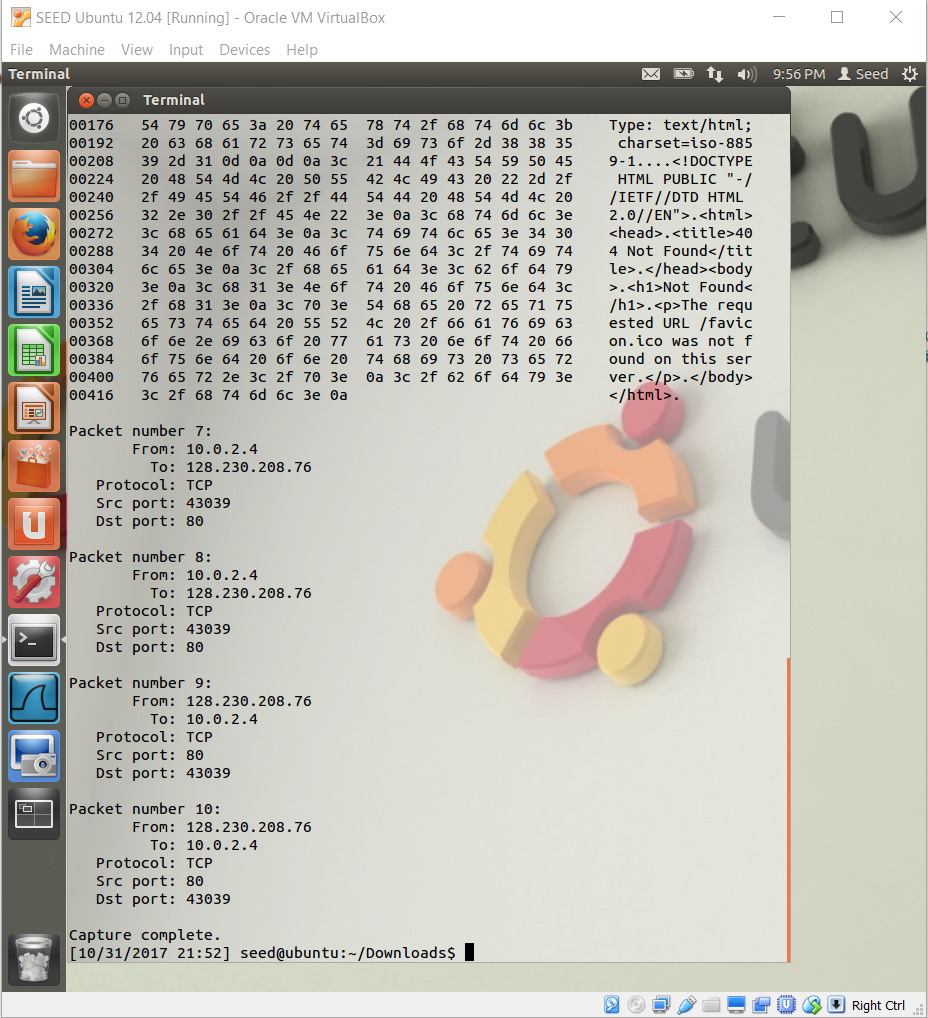
Sniffex1



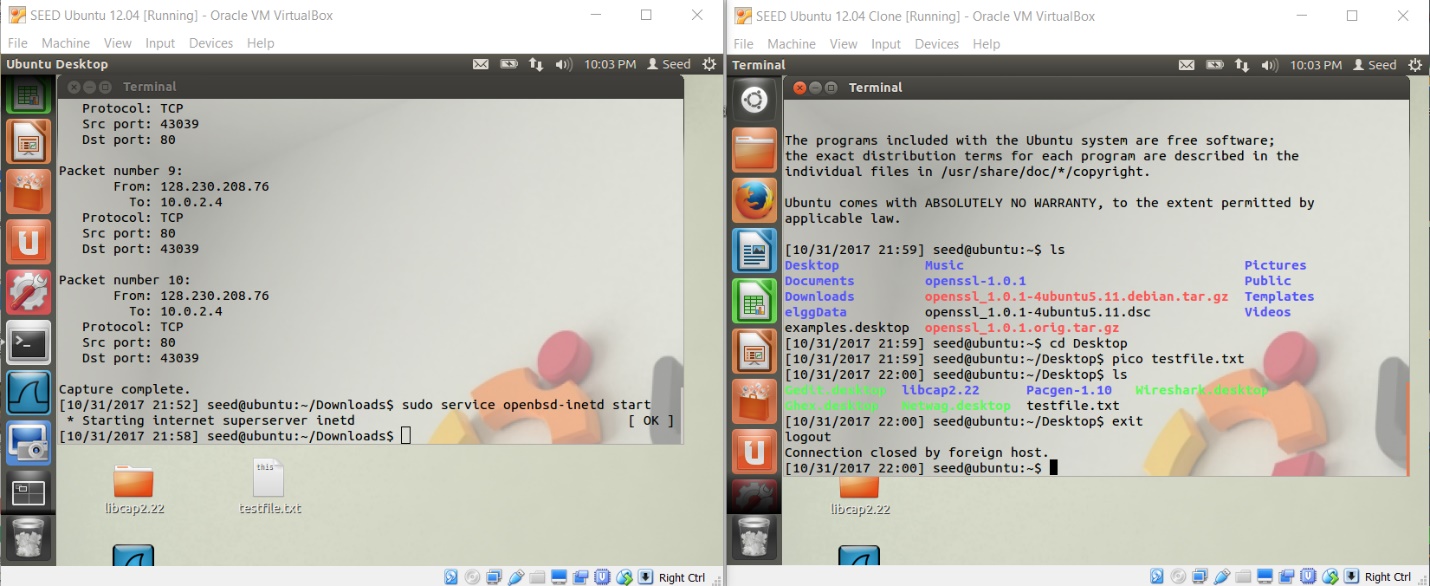
TCP filtering



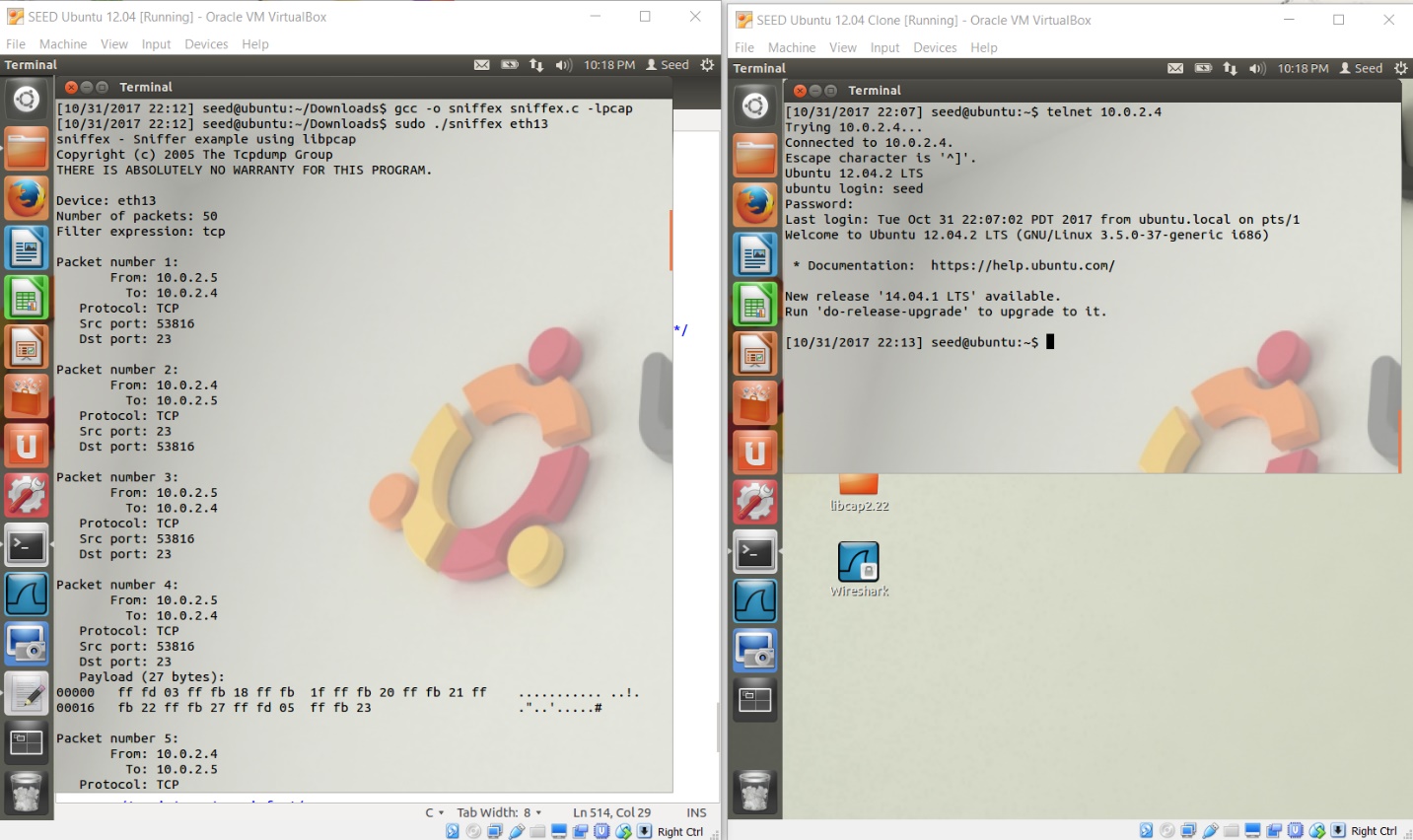


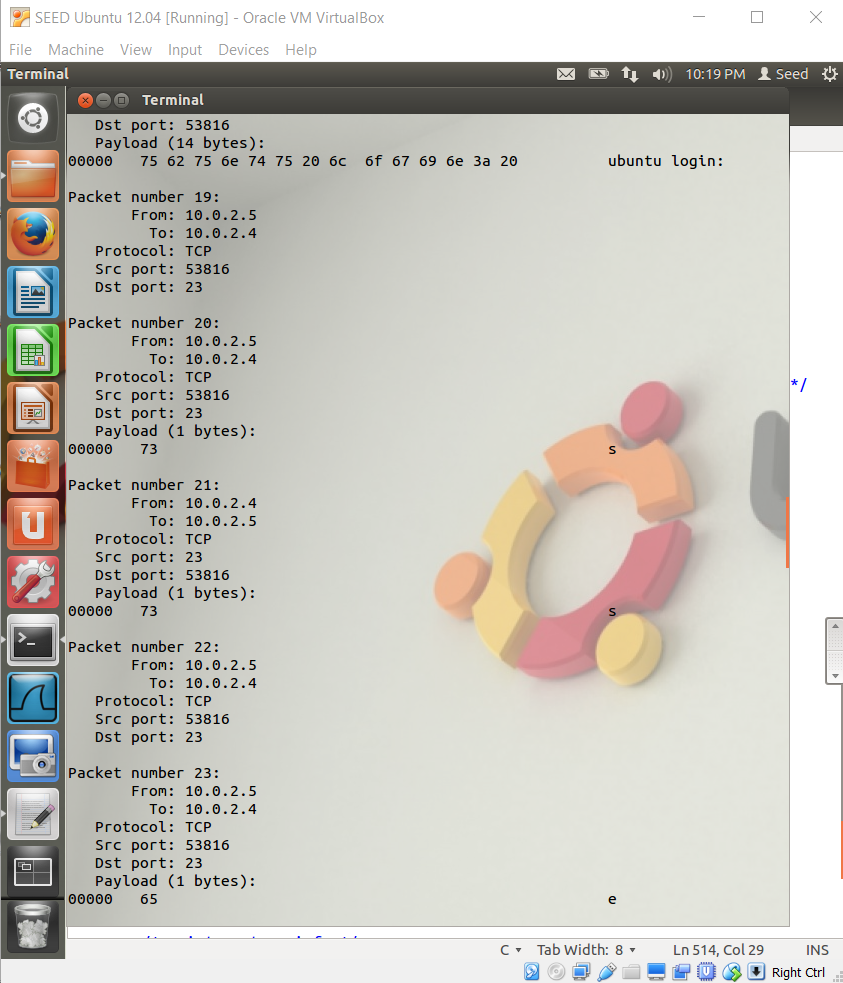


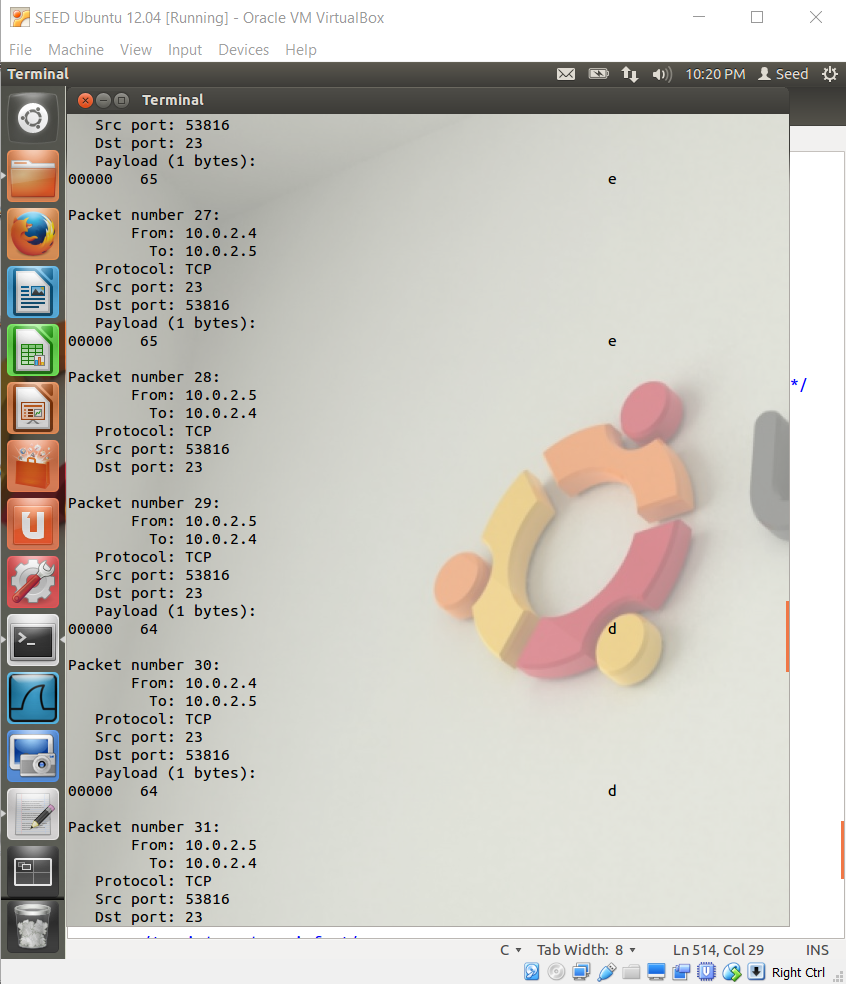
telnet remote file 1

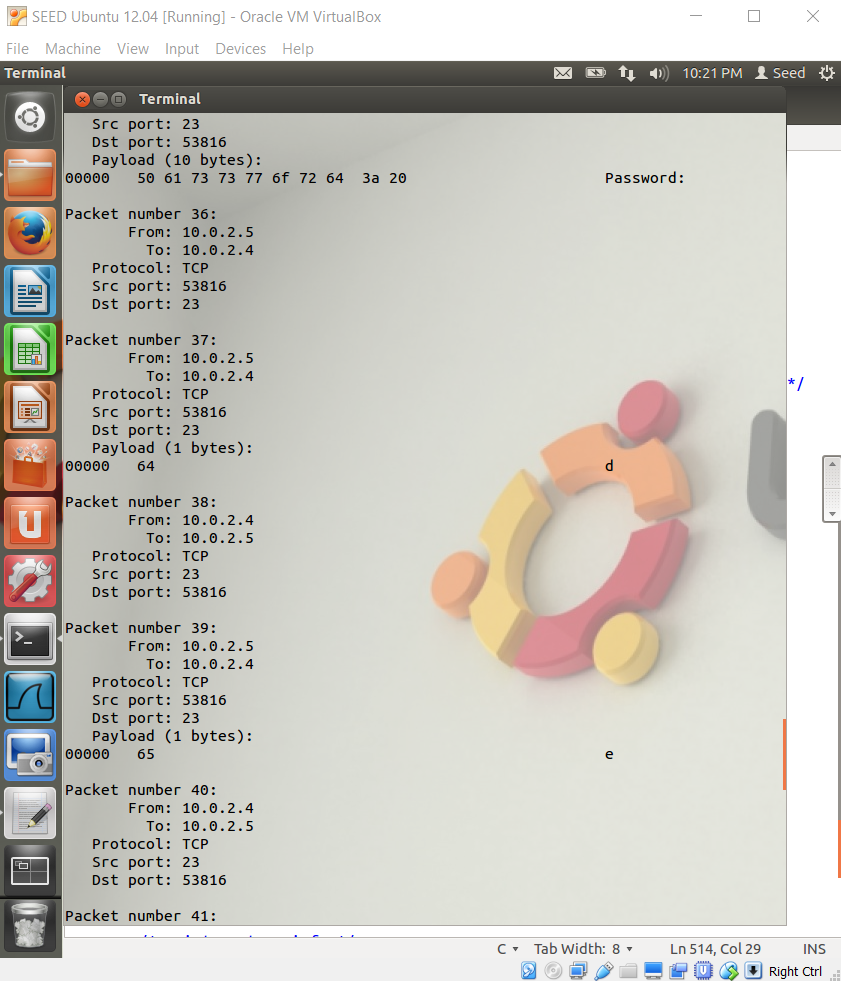


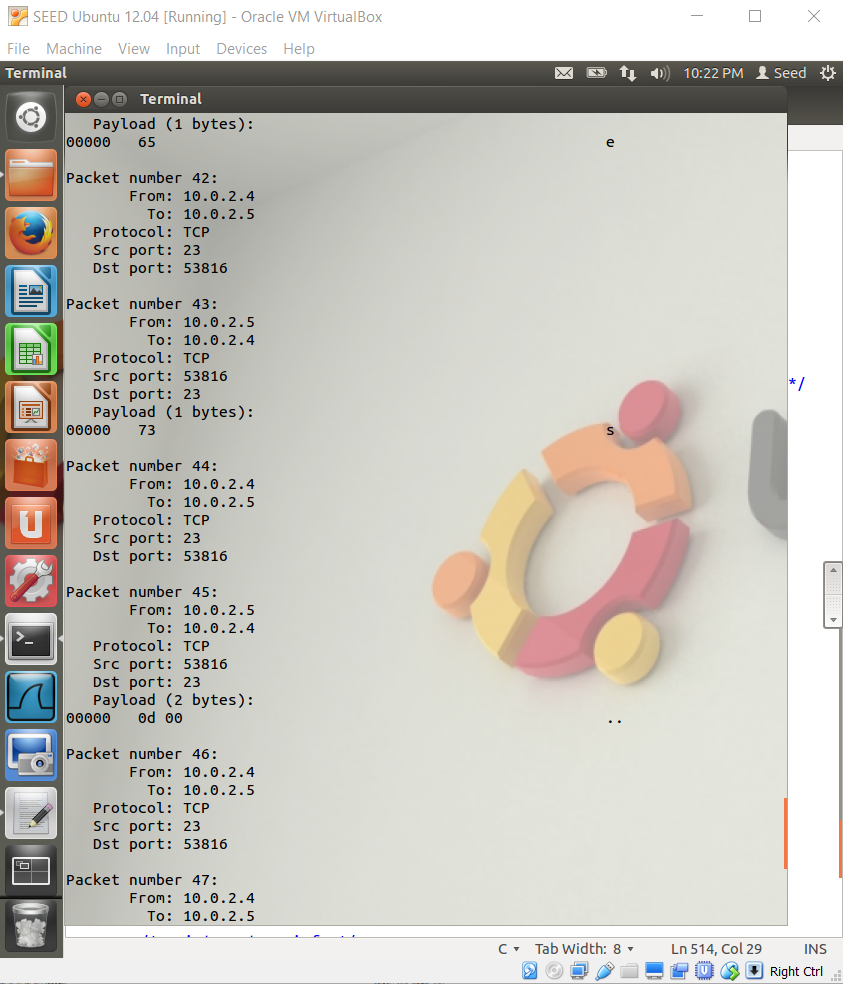
Finding the password with sniffix



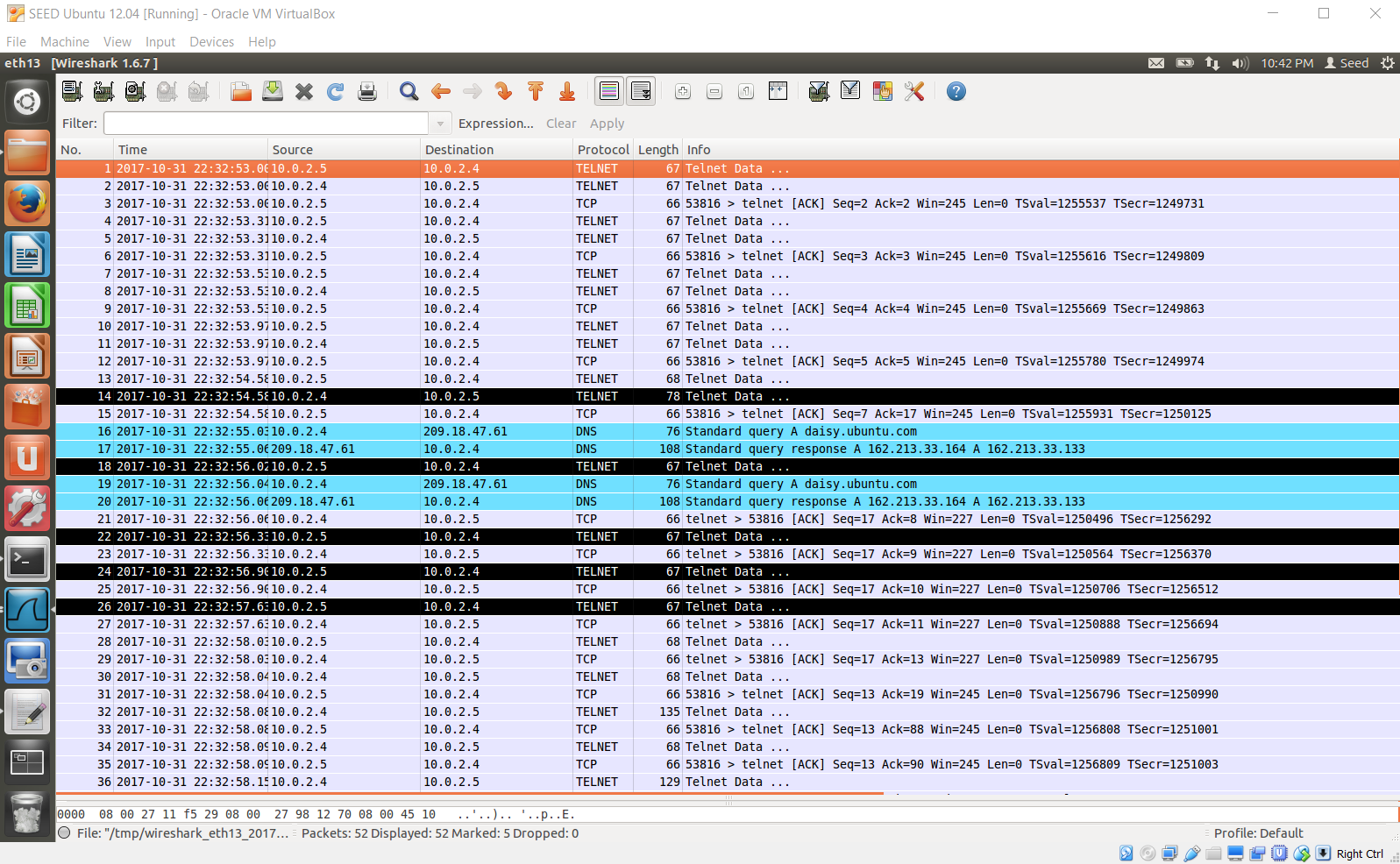








Wireshark (password found in marked packets)



Thoughts on Telnet:  
Telnet is not secure because when using a program such as sniffex.c to sniff for TCP packets, data is presented in clear-text, meaning there is no encryption to hide what is being sent over the network, which raises a security issue.

Wireshark vs ssh (password not identifiable)

