

**SI430: Lab 05 --- Packet Sniffing**

Names: **­ Brendan Lewis, Ian Coffey**

Alphas: 261194, 263708

External source(s)

Honor

We wrote the code on our own except the help from the external source(s) listed above. Moreover, we didn’t copy any part of the code from other midshipmen.

Initials: BAL & IKC

Challenges

It was difficult setting up the virtual machines in order to get each one to communicate with each other at first, but from there the main challenge became sending and deciphering the packets

What we learned and what was interesting to us

We learned a lot about the structure of packets and how to take them apart using code. We also learned that hexdump -C is very useful when debugging

# **Part 1:**

# 

# 

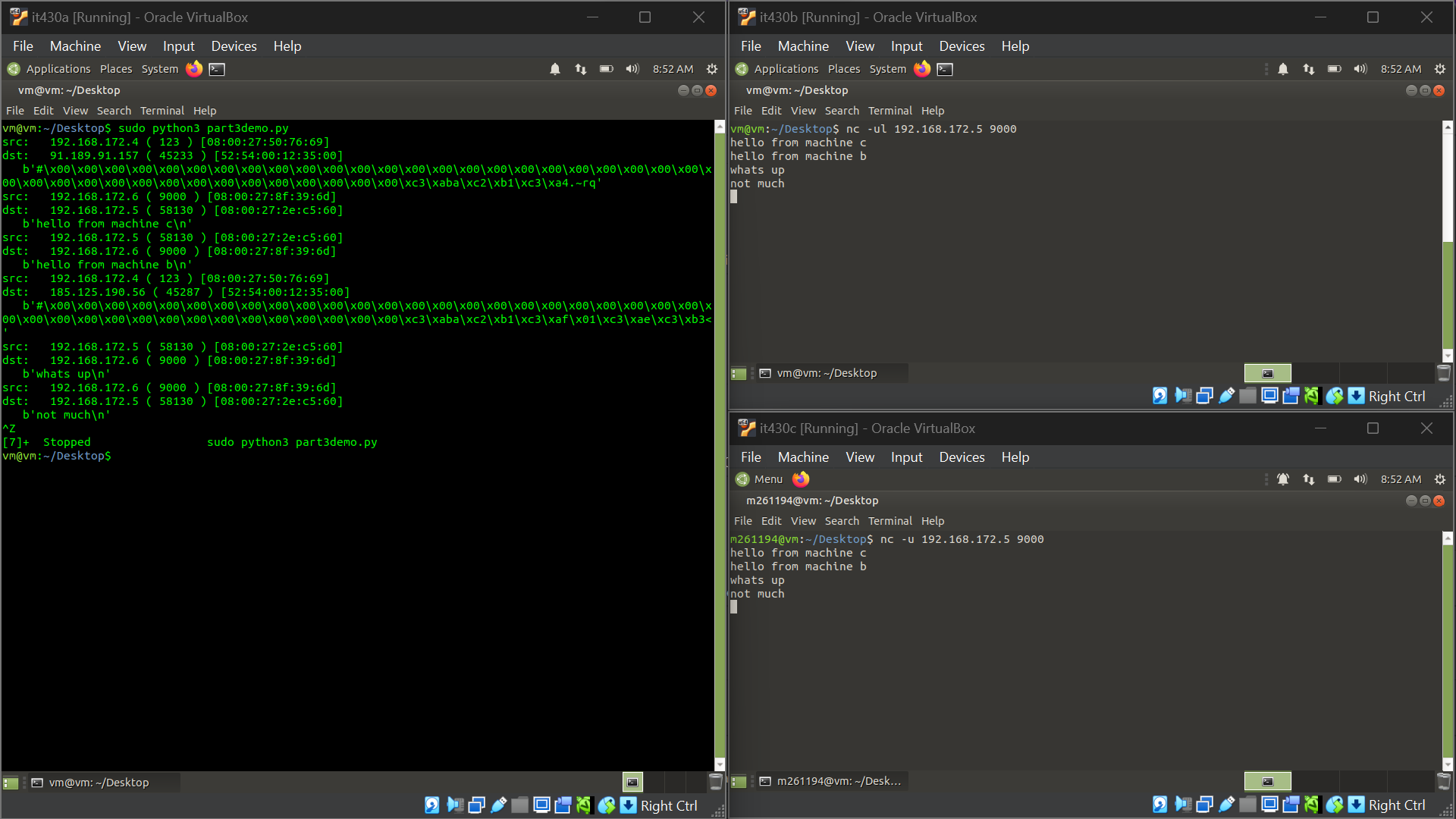
After getting the virtual machines configured, this part was relatively simple. Using ifconfig and port 9000, the net cat connections between each virtual machine was fairly easy; only needing a few commands per terminal. For the pings between the virtual machines, I used the same address I found in ifconfig and as seen in the images, the pings between each machine was successful.

# **Part 2:**

# 

Part 2 of this lab ended up being the most work intensive portion. While we were able to repurpose most of our code from lab 4, it still took us a while to decipher and break down each packet that we received. The indexing of each portion of information proved to be challenging, and the ports gave us some trouble. But all this program does it receive a packet, make sure it is a UDP packet, and then use specific index numbers to gather the needed information and print it to the screen. The pictures above show our program printing out a “hello world” and “goodbye world” message that was sent from machine b.

# **Part 3:**

 Part 3 of the lab required little to no extra work, as we were simply just re-purposing part 2 to sniff packets from other machines. After turning on promiscuous mode and setting up a net cat connection between machine b and c, machine a was successfully able to intercept and read UDP packets sent between b and c. While there were other extraneous packets caught by our program, it also caught the messages between b and c.