

**IT430: Lab 06 --- Packet Spoofing**



Names: Alex Traynor, Owen Pitchford

Alphas: 256402, 255076

External source   
(E.g., I referred to this site for xxx. My partner helped me debug my code…)

We both referenced the course notes while working on this lab, including the [linked article](https://inc0x0.com/tcp-ip-packets-introduction/tcp-ip-packets-3-manually-create-and-send-raw-tcp-ip-packets/) on the lab instructions.

Honor

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

Initials: \_AMT\_ and \_\_TOP\_\_

Challenges

(E.g., it was difficult to figure out how to xxx)

It was difficult to figure out how to store all of the data in the correct format in bytes for the TCP headers.

The version number that requires bit shifting took some thinking.

What we learned and what was interesting to us

(Specify what you learned and what was interesting to you)

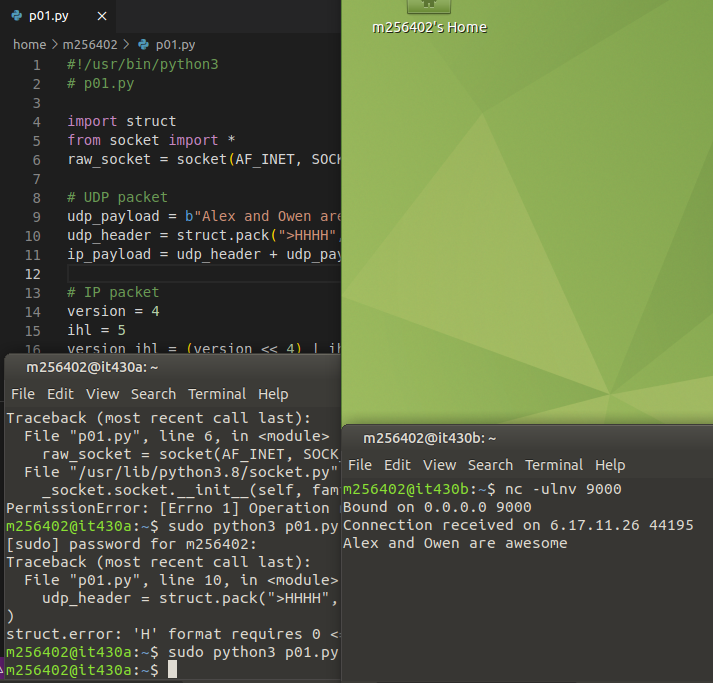
By actually making the packets we both have a deeper understanding of the TCP protocol and encapsulation as a whole. Additionally, it was interesting to see how simple we could spoof the source of a packet and how it can be dangerous. In this scenario, it would be quite easy to perform a SYN Flood attack.

How much work each of did (0: nothing, 10: everything)

* Alex Traynor: 5
* Owen Pitchford: 5

# **Part 1**

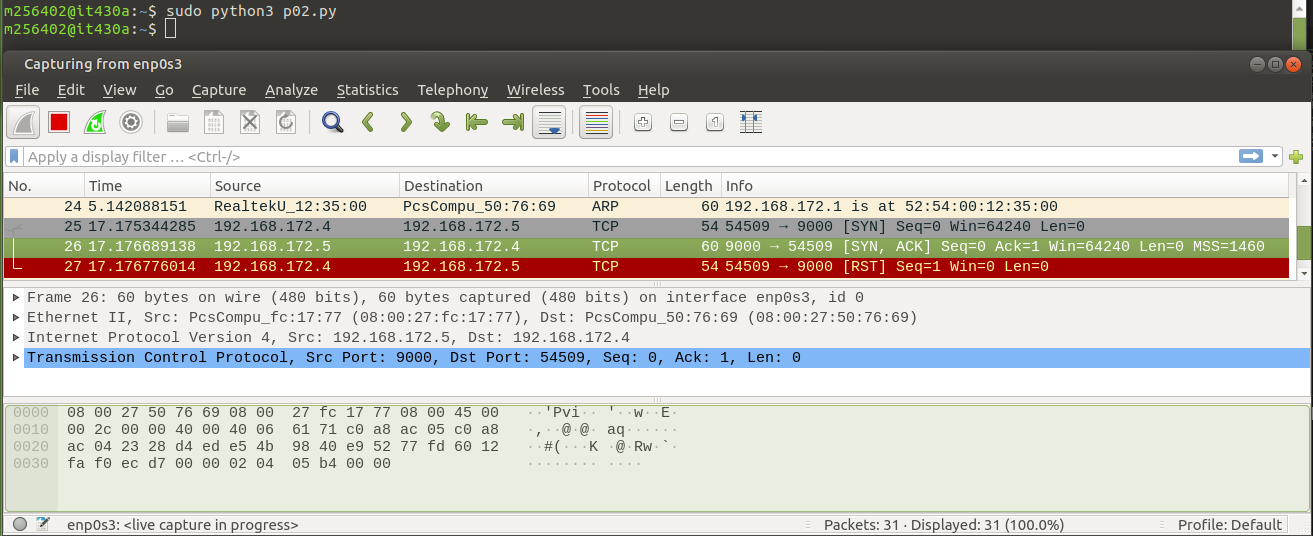
Below is a screenshot of the successful spoofing attack by it430a. Both the source address and source port were modified. The source port is Owen’s school lunch code and the source address is a combination of our birthdays. The expanded code for p01.py can be found in our submission.



# 

# **Part 2**

Below is a screen capture of our Wireshark window after the p02.py program was run. Our code works by constructing a spoofed packet in the same manner as in part 1. This task only seems more complicated because the TCP header is a bit tougher to construct, but using the methods provided by Dr. Choi this became straightforward.

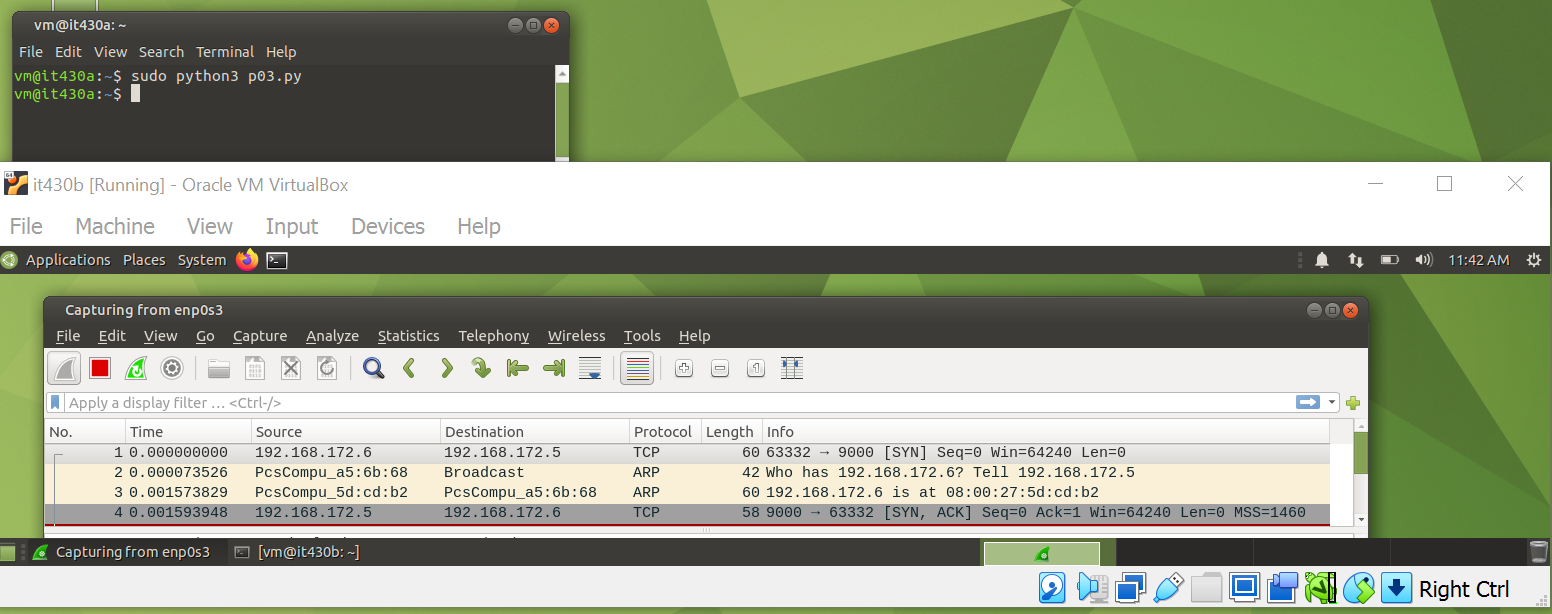


# 

# **Part 3**

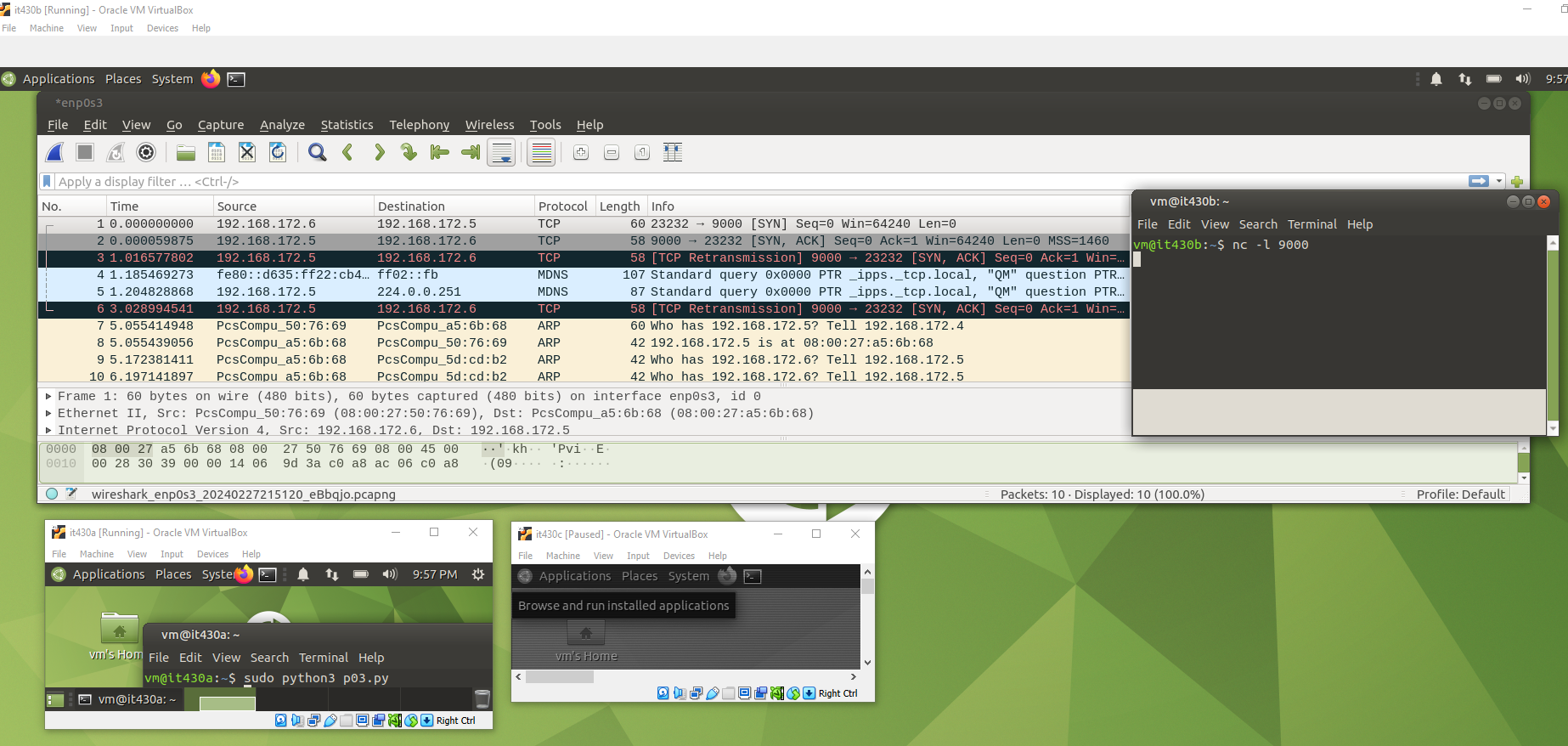
Activity 1

Below is a screen capture of our Wireshark window after Activity 1. You can see that p03.py was run on it430a, the SYN packet comes from it430b (192.168.172.5) and that the SYN, ACK gets sent to it430c (192.168.172.6).



Activity 2

Below is a screen capture of our Wireshark window after Activity 2. You can see that p03.py was run on it430a, the SYN packet comes from it430b (192.168.172.5) and the the SYN, ACK get sent to it430c (192.168.172.6). You can see that no RST packet is sent because it430c is paused so it430b attempts to retransmit more SYN ACK packets. The connection on port 9000 is half-open in a state called SYN\_RECV.



# **Part 4**

The author opens up section 2.4 discussing the final group of “actors” in security engineering–that is the independent security researchers. Whether hobbyists, professors, or actual hackers, these people search for exploits for a variety of reasons. Professors often seek these exploits for personal gain, such as money while hobbyists hack because they enjoy it and will often report their findings to system administrators/companies. This group also includes Black Hat hackers whose motivation is malice.

The author expands this last form of actor into a few categories of abusers. From hate campaigns to relationship/sex abuse, these individuals motivated by malice have caused governments, policymakers, and private companies a large headache. Government agencies have to worry about controlling propaganda to support the freedom of elections while large companies such as Apple and Google face backlash for allowing apps that support relationship and or sex abuse to be published on their app stores. For me, this section reinforces the importance of security online because it highlights the types of ramifications that are possible from “data” being stolen. It also makes me fortunate to have had a somewhat decent education on this in the past where a lot of victims have not.

In summary, the past few reading assignment chapters have reinforced the idea that security measures must be taken with the whole picture in mind. As a computer scientist, my security measures are tested by more than just stereotypical “hackers”. Use and abuse are both possibilities by unexpected parties. This threat scales based on the type and purpose of the system you are working on.