**NCTU CN 2018 Lab-1 Packet Manipulation via Scapy**

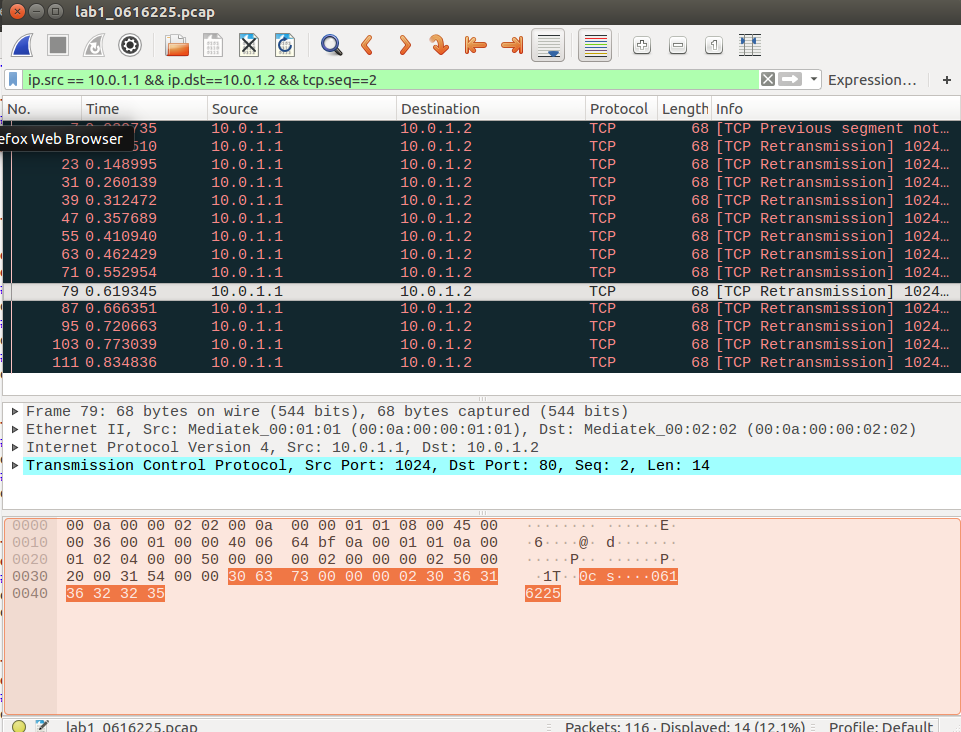
**0616225 張承遠**

**Part A-Question:**

1. What is your command to filter the packet with customized header on Wireshark?

A: ip.src == 10.0.1.1 && ip.dst == 10.0.1.2 && tcp.port ==2

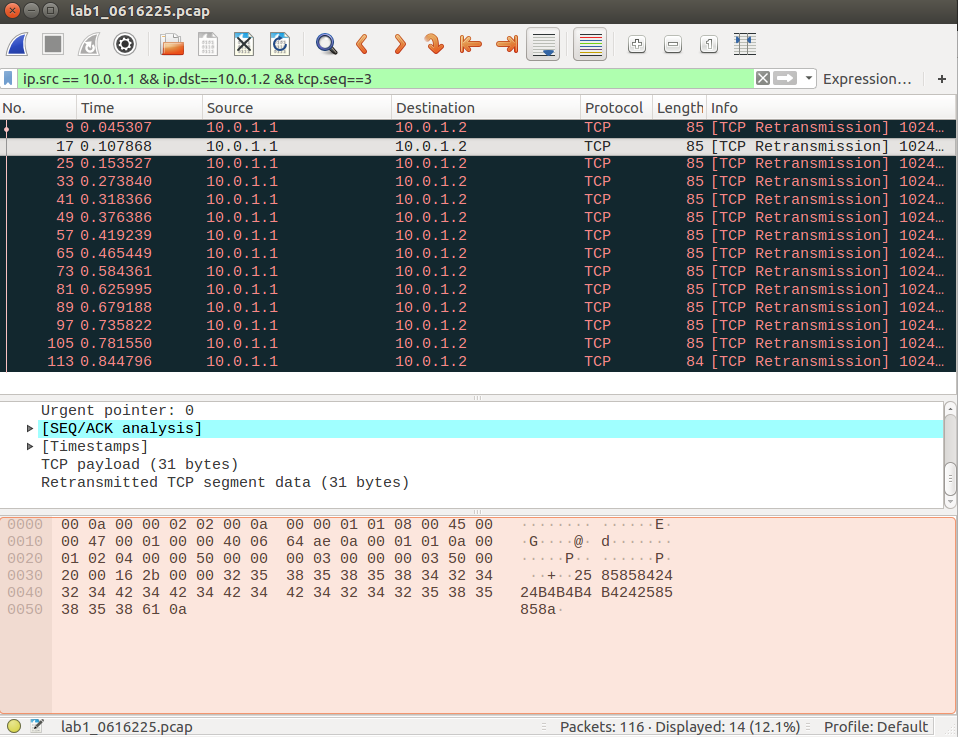
2. Show the screenshot of filtering the packet with customized header.



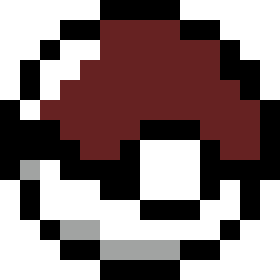
3. What is your command to filter the packet with “secret” payload on Wireshark?

A: ip.src == 10.0.1.1 && ip.dst == 10.0.1.2 && tcp.seq == 3

4. Show the screenshot of filtering the packet with “secret” payload .



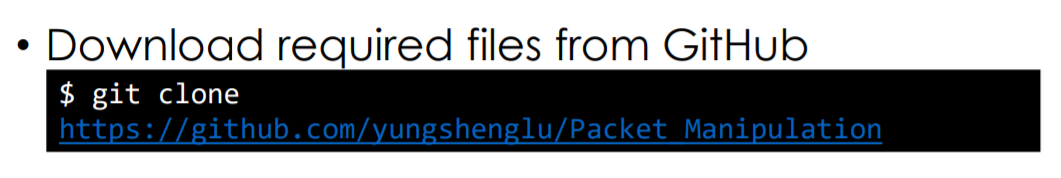
5. Show the result after decoding the “secret” payload.



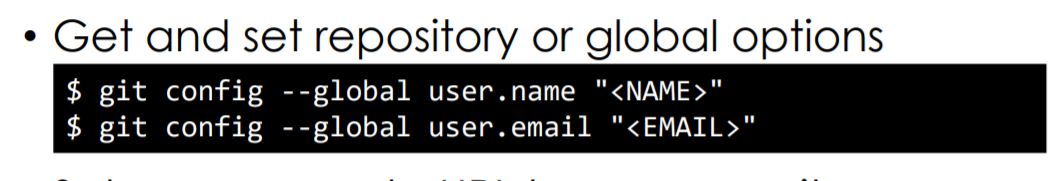
**Part B-Description:**

**Task1-Environment Setup**

**1.do github setting**



To begin with, download the required repository from TA’s github.



Then, set my github name and email in order to push data in the future.

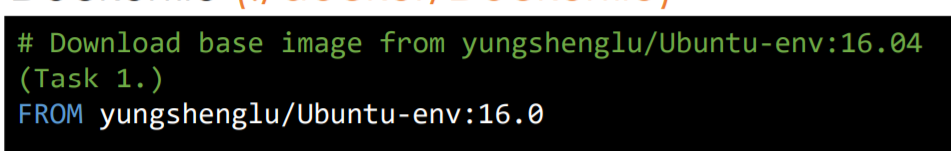


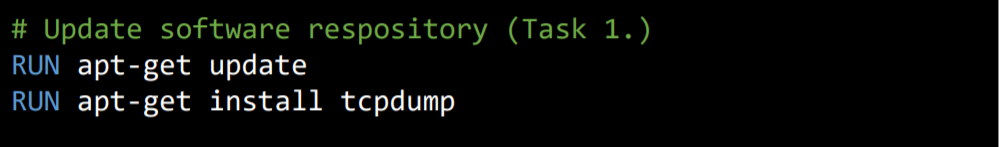
Set a remote URL to let your data can upload correctly.



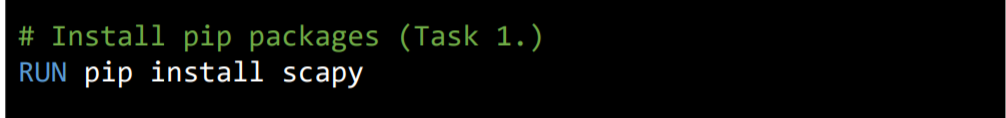
Last, push the origin data to my github, just like a base.

**2.DockerFile setting**

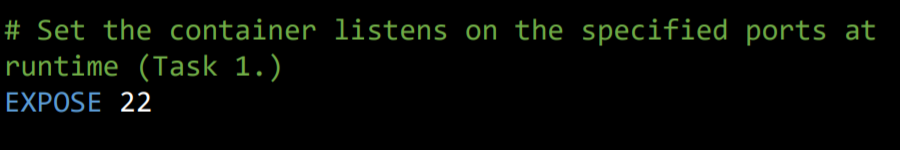
First, download the base data from TA’s github.



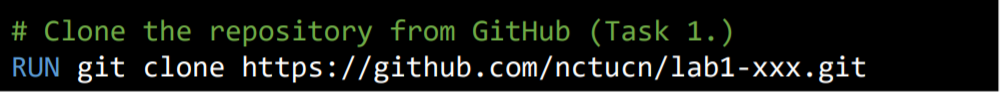
Second, update all the software repository. Then, install tcpdump which can track the packa’s direction.



Install Scapy that can help me analyze the packet. It’s useful.

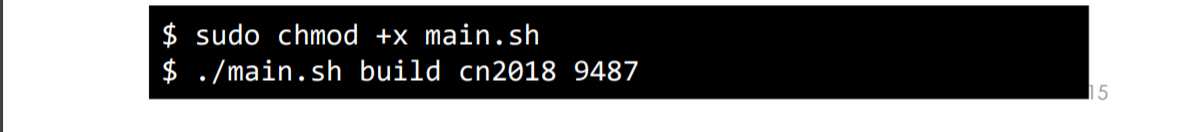


The command is to assign the container to a specified port.



Last, clone the repository I just stored before. However, when I execute the command, I got error and I didn’t know how to fix it, so I skipped the command. (Furthermore, I found that the sample in the lab PDF didn’t do this command as well)

**3.Build Environment**

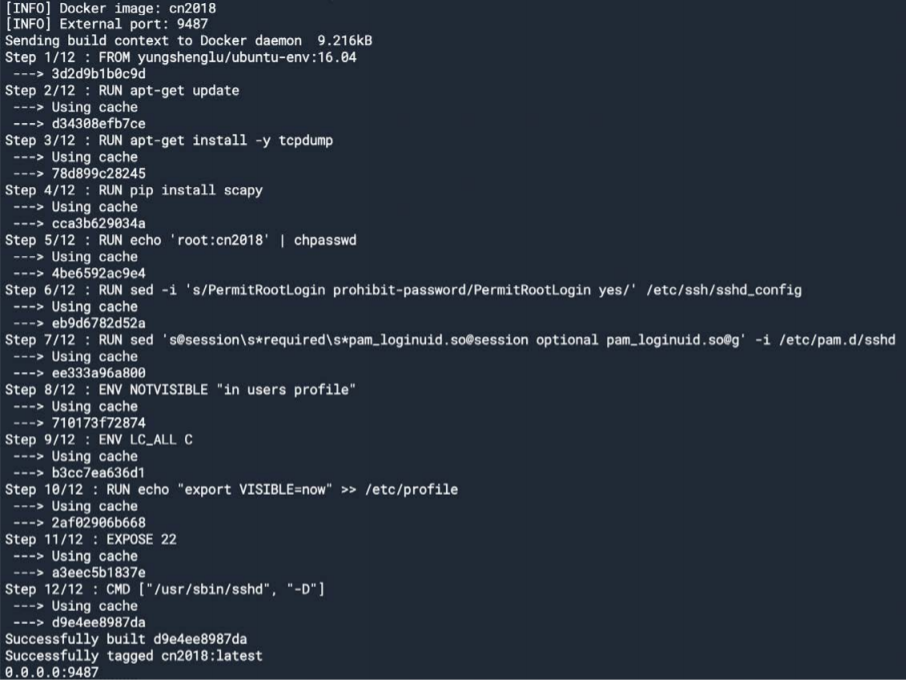


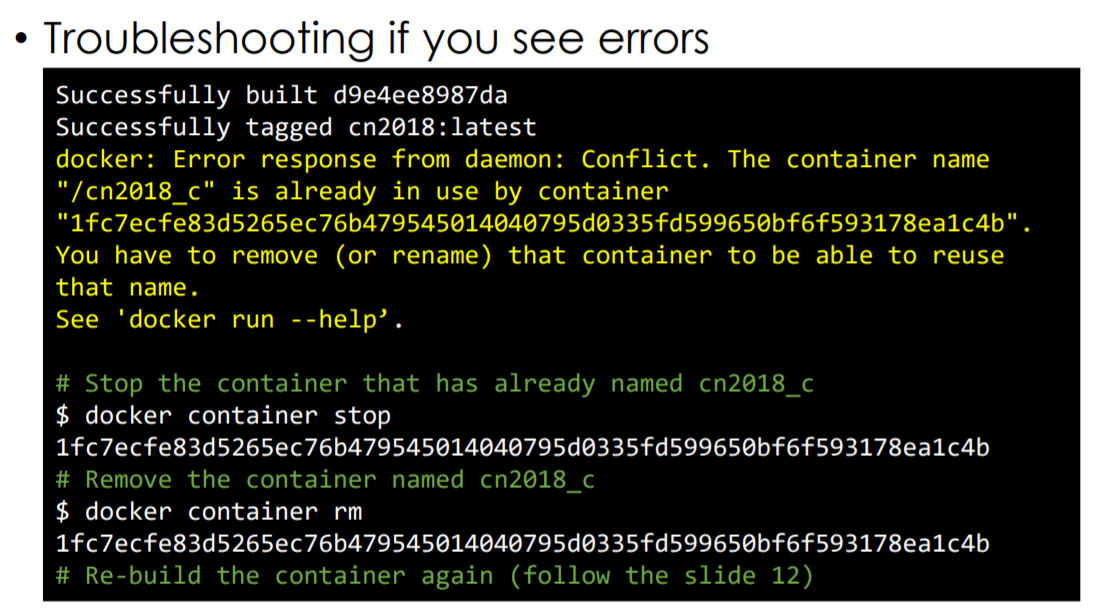
Because I use Ubuntu as my system, I didn’t do the command for windows.

First, run main.sh.

“chmod +x ‘Filename’ ” the command can give the filename have the permission to execute.

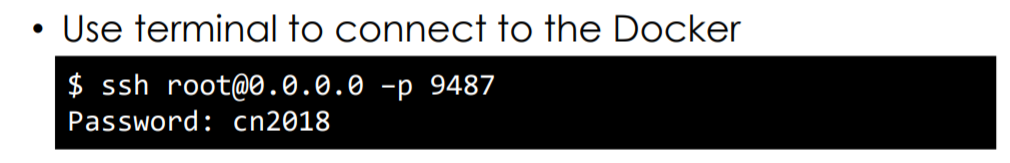
Second, build the environment.

If succeed, I will got the following result. 



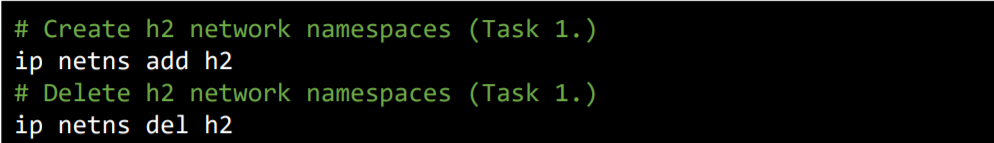
If the error said that container name has already in use occurred when building environment, stop the container and remove it. Then rebuild again.

**4.Connect to docker container**

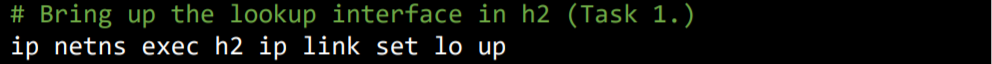


Connect to ssh server.

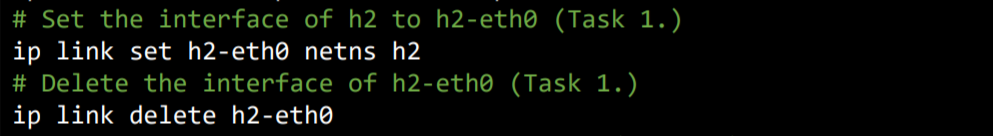
**5.Create namespace and set**



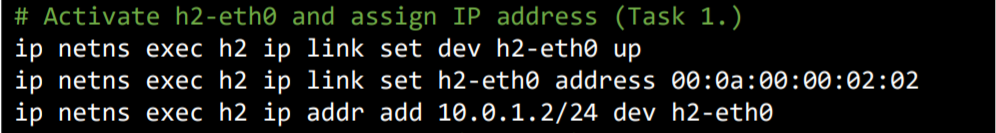
Create and delete h2 network namespace.



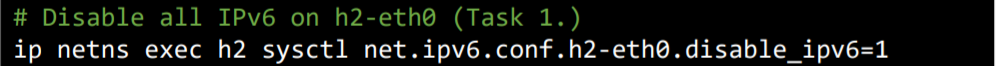
**Create lo(lookup) interface.**



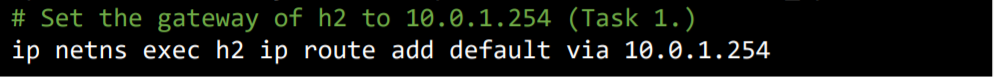
Set and delete the interface of h2-eth0



Activate h2-eth0 and assign IP address.



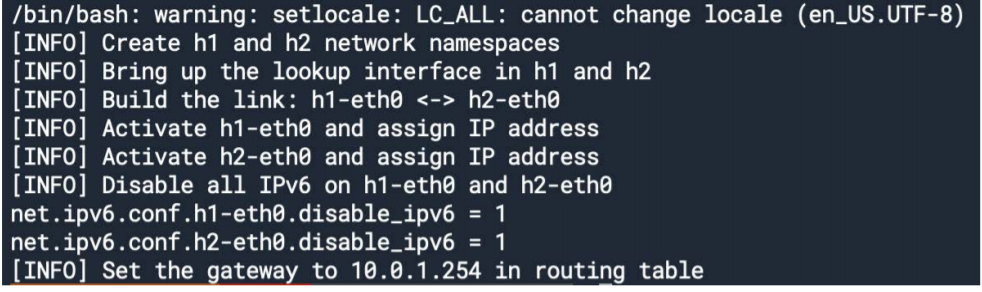
Disable all IPv6 on h2-eth0

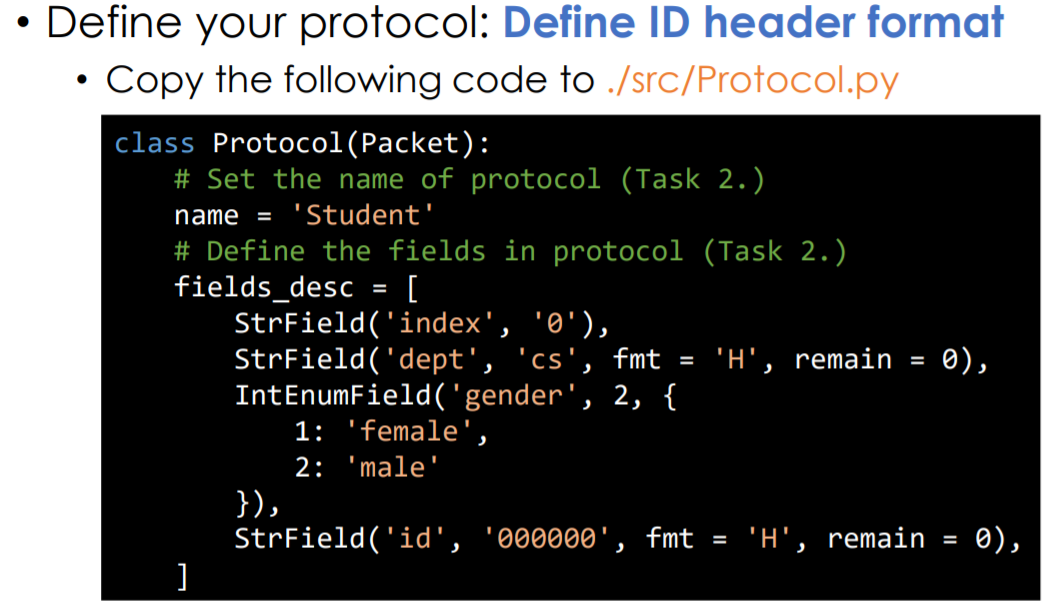


Set the gateway of h2 to 10.0.1.254



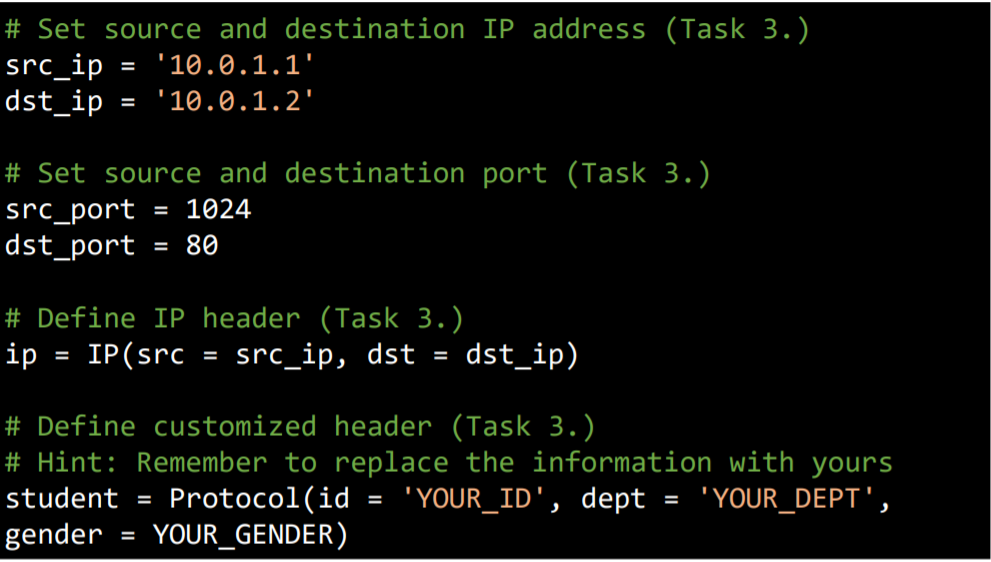
Execute the main.sh(just like above)

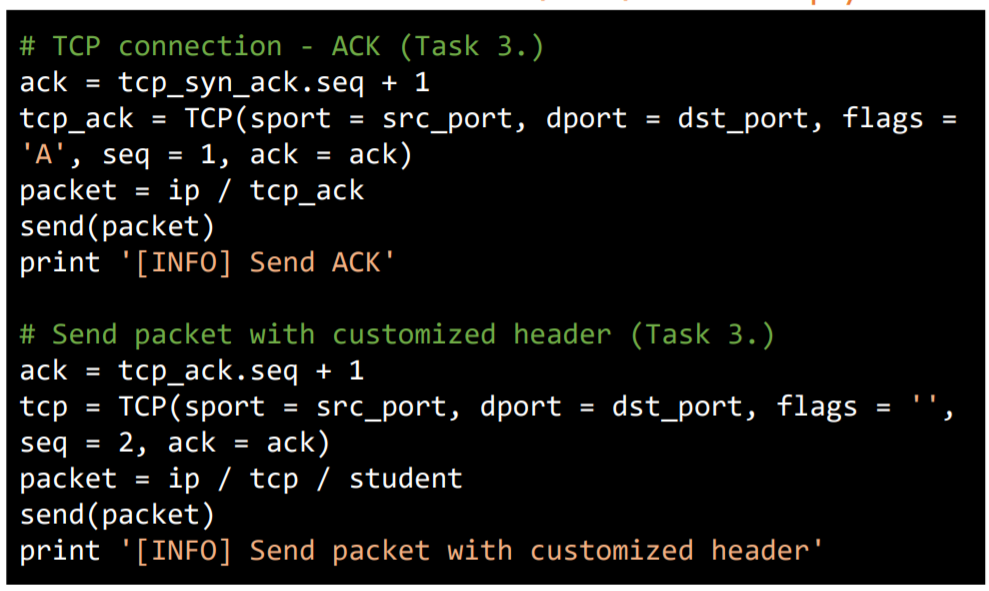


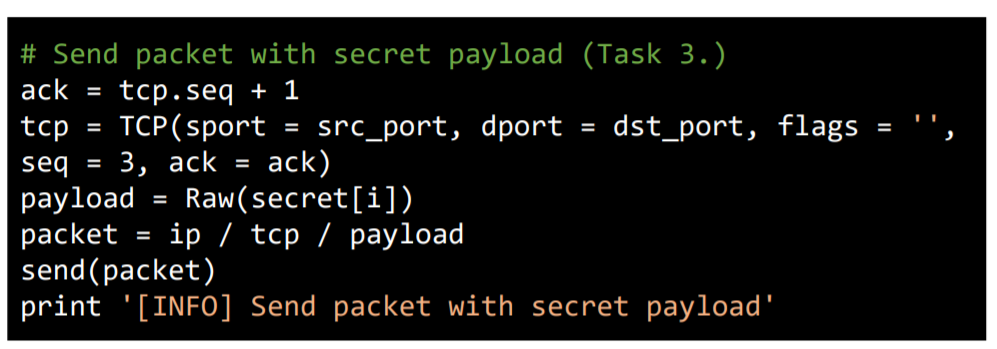
**Task 2 Define protocal via Scapy**

Define my own protocal by adding gender, department, and student id.

**Task 3 Send packets**

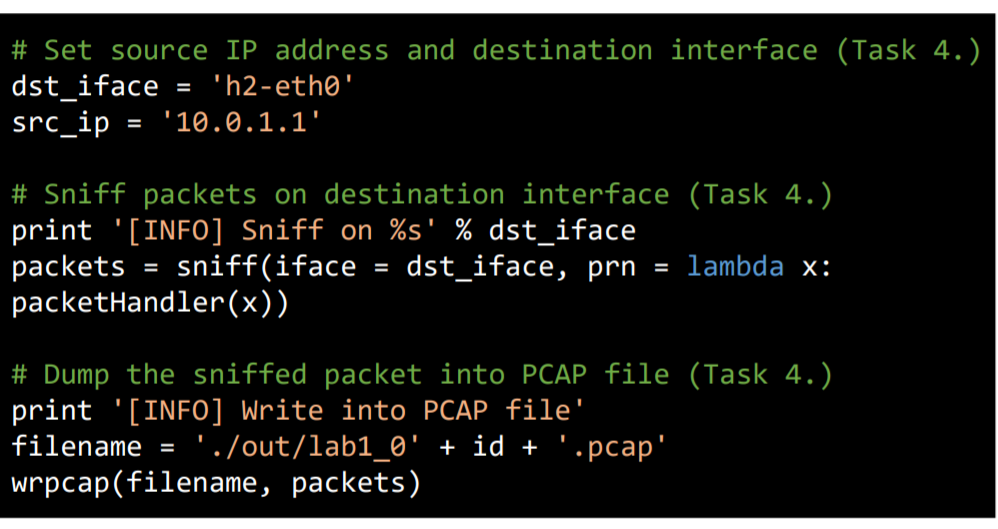






Type these code into sender.py. First, we set IP addtess and Port. Then define our customized header. (id, dept. gender). Then set TCP connection and the data we want to send.

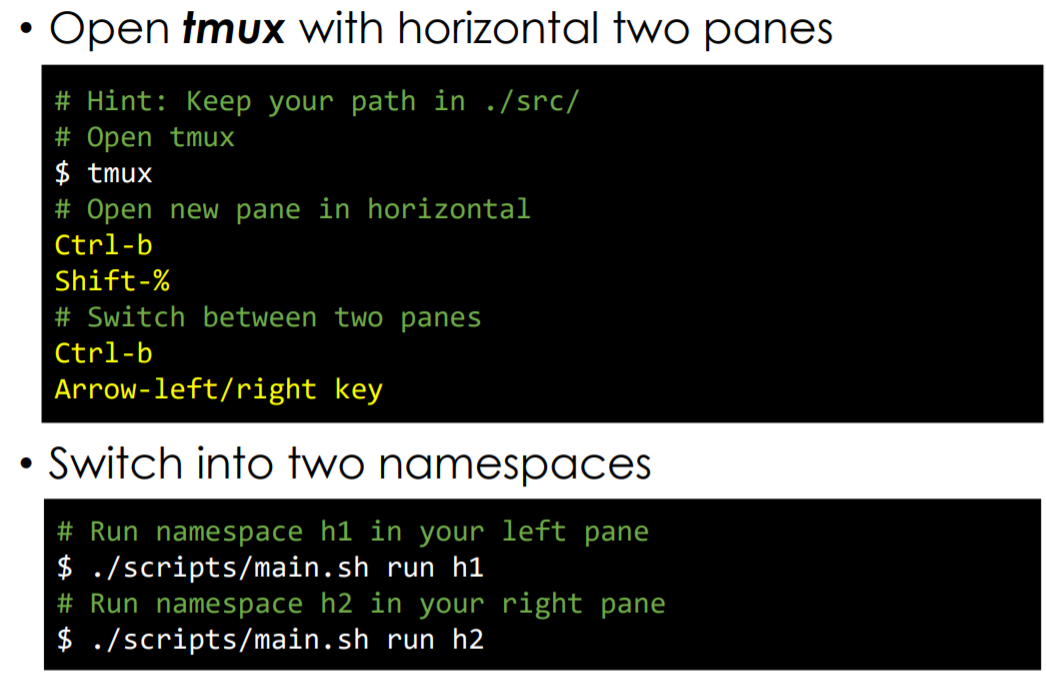
**Task 4 Sniff Packet**



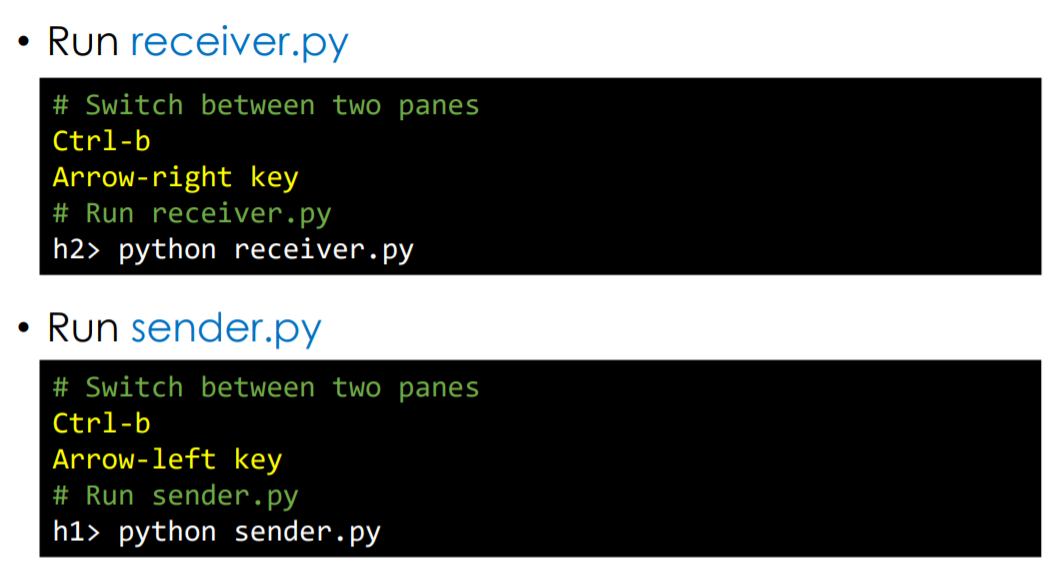
Type these code into receiver.py. First, set source IP and destination interface. Then, sniff packets and dump the sniffed packet into PCAP file.

**Task 5 Run sender and receiver**

1. **Install and use tmux**



Use tmux to cut the terminal into two pages, then run h1 and h2 separately.



Run receiver.py in namespace h2, and run sender.py in h1. After enter the command, h1 will start to send packets to h2.

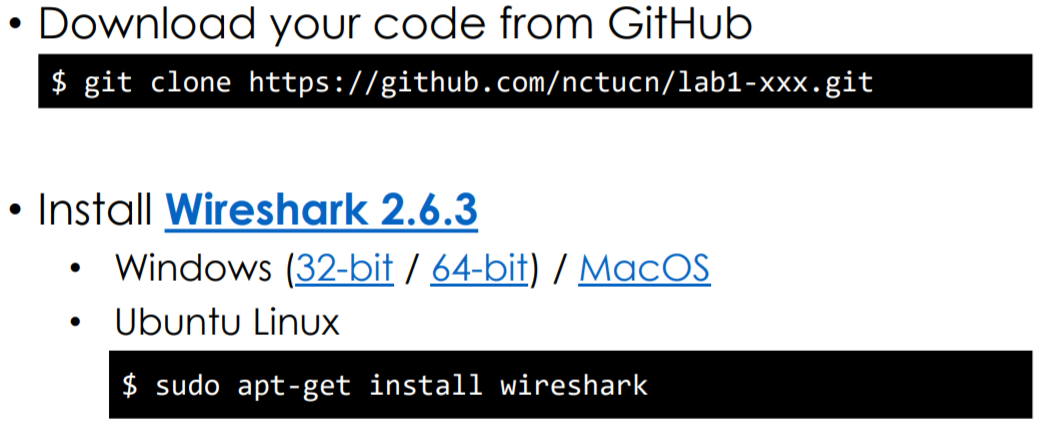
**Task 6 Push your files to remote**





After doing so many tasks, It’s time to push my document into DockerHub and Github. Remember to add commit when pushing document to Github.

**Task 7 Load PCAP via Wireshark**



Download the repository from my github, then open the PCAP files using Wireshark.

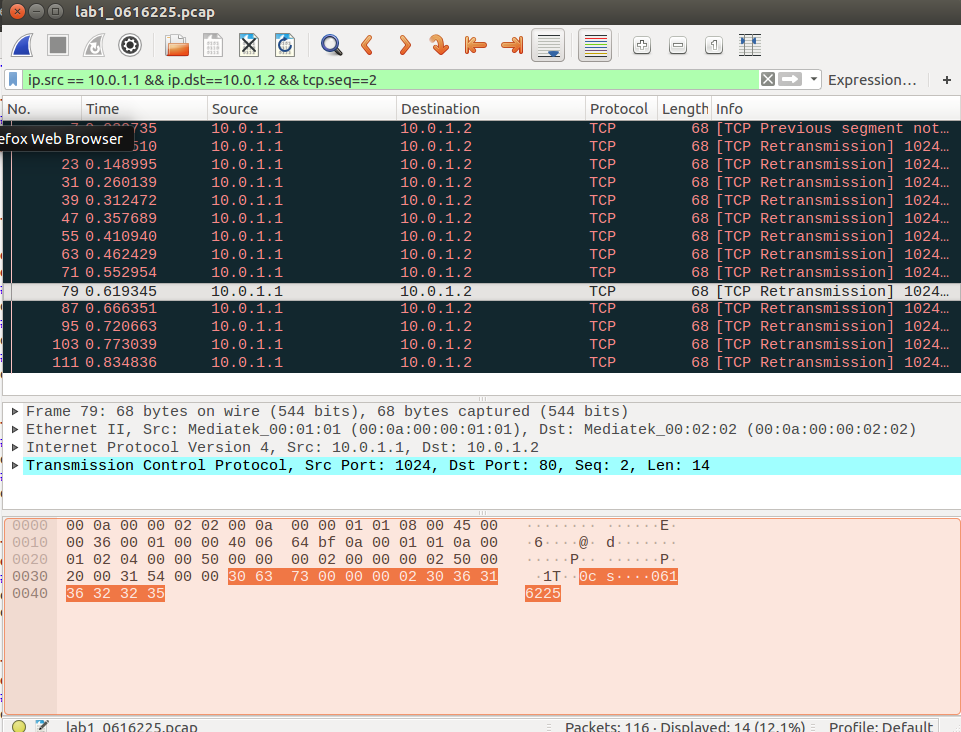
**Task 8 Filter the target packet**



Filter Rule: I found that the packet which have the defined protocol have the same TCP sequence and ip source and destination. So my filter rule is :

Ip.src == 10.0.1.1 && ip.dst == 10.0.1.2 && tcp.seq == 2

After typing it, I got the packet below and they all have the defined protocol.

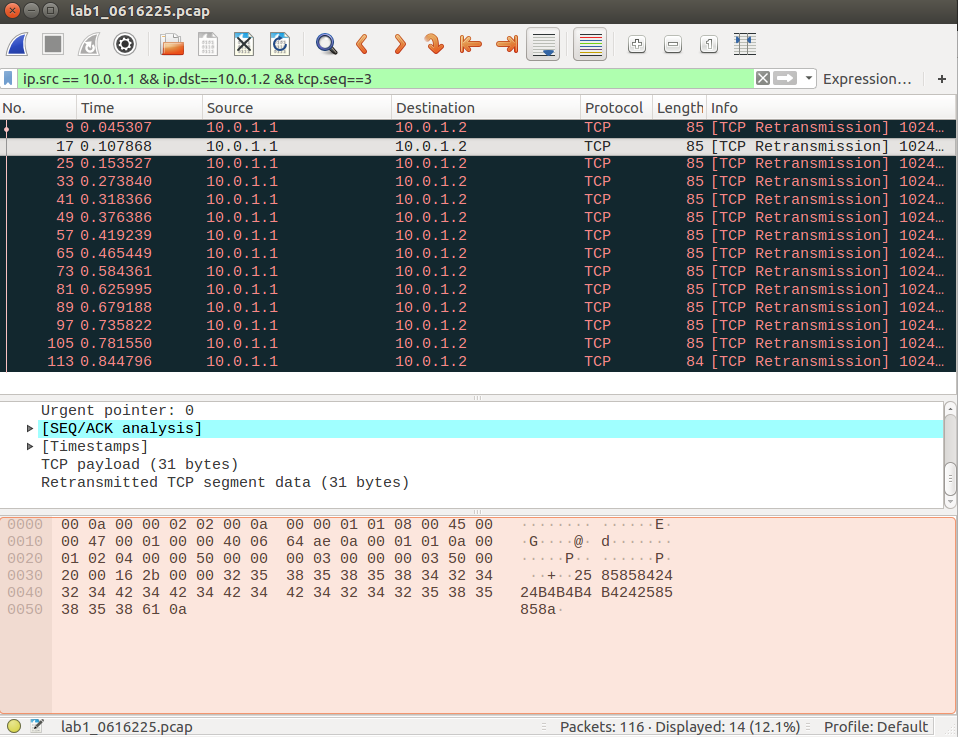




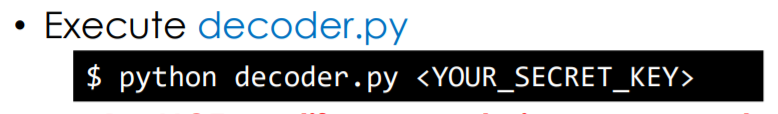
Filter Rule: I found that the packet which have the secret bits have the same TCP sequence, so are ip source and ip destination. So my filter rule is:

Ip.src == 10.0.1.1 && ip.dst == 10.0.1.2 && tcp.seq == 3

After typing it, I got fourteen packets and every packets have one secret bit in the first alphabet. After combining these secret bit, I got my secret bit “52261605226160”.



**Task 9 Decode the secret key**



Finally, the last step is to decode with my secret bit, so execute decoder.py and type my secret bit into decoder.py. After doing that, I successfully got the picture, which is a dark Pokemon ball. I think I succeed.

**Bonus:**

**Q1.What you have learned in this lab?**

A: Actually, I think I learned a lot of things in this lab and in the assignment. First, I learned how to install virtual machine and install Ubuntu. This help me understand the Linux environment more. Second, I learned some basic command for Linux terminal such as “cd”, “ls”, “rm”…… When I had some trouble, I would surf the net to ask for answer. Furthermore, I learned how to use docker container and how to connect to docker. I also learned how to basically use Wireshark and the rule of filter packets. Last but not least, after doing all the work in the lab, I understand how packets transmit in the Net concretely. It is helpful for the lecture because without this lab, I could only learn these knowledge from book and didn’t know how these packets do practically in the Net. To sum up, I think this lab really helpful for me, and I learned lots of knowledge. I will expect the next lab in the future.

**Q2. What difficulty you have met in this lab?**

A: The first question I have met was I couldn’t install Docker in windows environment because it said that I have to update my windows. Hence, I decided to use Ubuntu instead of windows. I know I would have to learn a lot of new things from then on. Then, at first, I did all the lab in my Ubuntu instead of Docker because I didn’t fully understand what Docker is. After finding this problem, I decided to remake. Furthermore, many errors occurred when I use terminal in Ubuntu. For example, I got some error because I didn’t do the command with roots or got error because I didn’t install the required document. To sum up, I spent so much time debugging, but I think it is worthy because I learned lots of new things.