**NCTU CN2018 Lab. 1 – Packet Manipulation via Scapy**

**Student name: 黃秉茂 Student ID: 0616098 Department: CS**

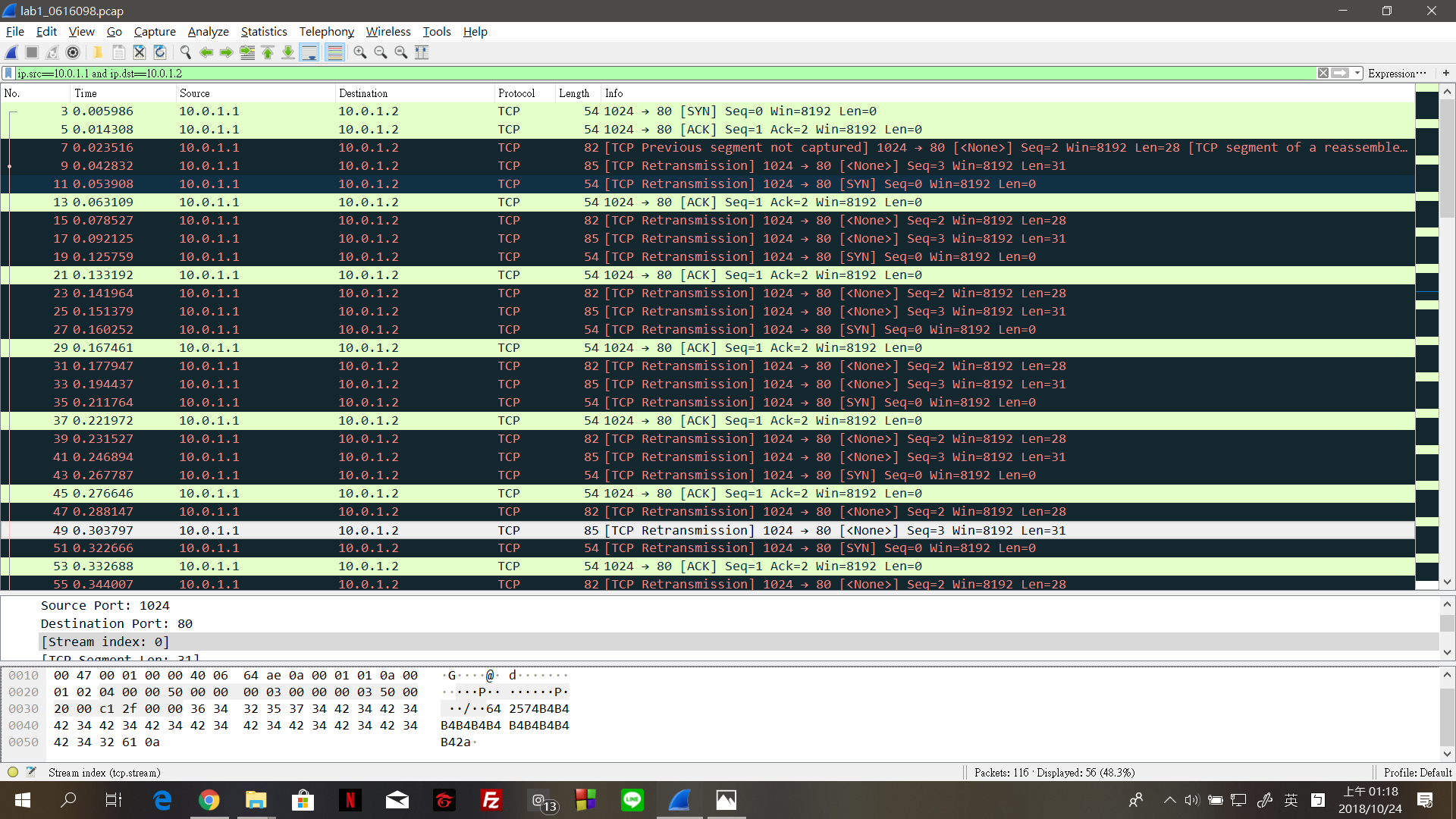
**Part A. Questions**

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

Wireshark?

ip.src==10.0.1.1 and ip.dst==10.0.1.2

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

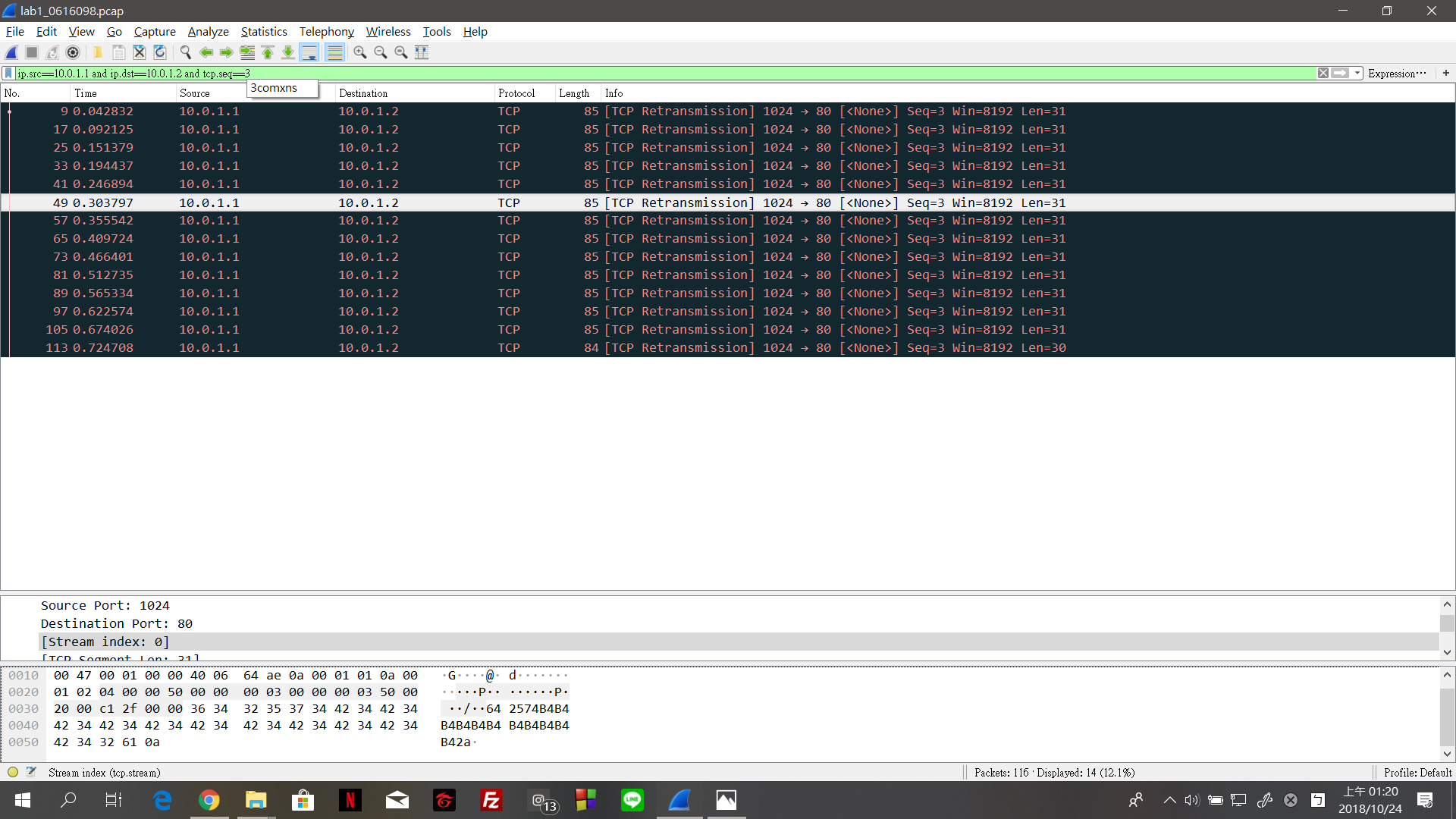


3. What is your command to filter the packet with “secret” payload on

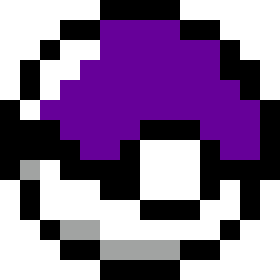
Wireshark?

ip.src==10.0.1.1 and ip.dst==10.0.1.2 and 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**

Task 1 – Environment setup

先建立跟GitHub的關係

$ git clone https://github.com/yungshenglu/Packet\_Manipulation

從GitHub下載所需的檔案

$ git config --global user.name "<NAME>"

$ git config --global user.email "<EMAIL>"

設置上傳的檔案的存放處(GitHub中)

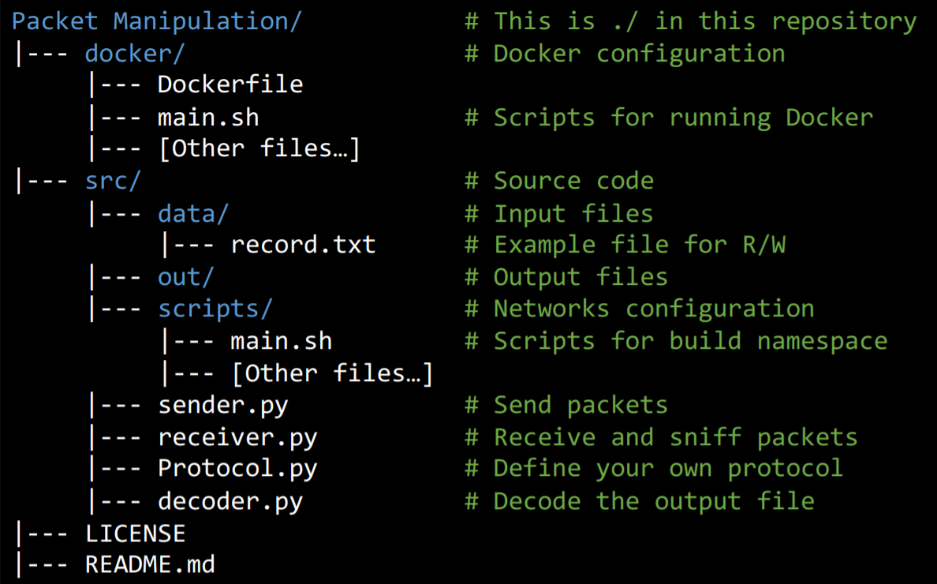
$ git remote set-url origin [https://github.com/nctucn/lab1-<GITHUB\_ID>.git](https://github.com/nctucn/lab1-%3cGITHUB_ID%3e.git)

設置新的到我的存放處的URL

$ git push origin master

Push 我的存放處使他可以使用遠端

packet manipulation project的結構



Configure Dockerfile

設定Dockerfile?

<https://docs.docker.com/engine/reference/builder/>

# Download base image from yungshenglu/ubuntu-env:16.04

FROM yungshenglu/ubuntu-env:16.04

First, download base image from yungshenglu/ubuntu-env:16.04

# Update software respository (Task 1.)

RUN apt-get update

更新所有軟件的存放處

# Install software repository (Task 1.)

RUN apt-get install –y tcpdump

安裝所有軟件的存放處

# Install pip packages (Task 1.)

RUN pip install scapy

Install pip packages

# Set the container listens on the specified ports at runtime (Task 1.) EXPOSE 22

設定container和NO.22 ports用來接收

# Clone the repository from GitHub (Task 1.)

RUN git clone https://github.com/yungshenglu/Packet\_Manipulation.git

把Packet\_Manipulation從助教的GitHub下載

使用 SSH登入Docker container

打開 PieTTY 和連結 Docker

IP address: 192.168.99.100

Port: 9487

Login as root

Login: root

Password: cn2018

進入main.sh

cd /

cd Packet\_Manipulation/src/scripts

前往Packet\_Manipulation/src/scripts

vim main.sh

進入main.sh

輸入i + Enter進入Insert模式

在main.sh修改程式碼模仿h1在 ./src/scripts/main.sh創建給h2的namespace

# Create h2 network namespaces (Task 1.)

ip netns add h2

創建h2的網路namespace

# Delete h2 network namespaces (Task 1.)

ip netns del h2

刪除h2的網路namespace

# Bring up the lookup interface in h2 (Task 1.)

ip netns exec h2 ip link set lo up

Bring up the lookup interface in h2

# Set the interface of h2 to h2-eth0 (Task 1.)

ip link set h2-eth0 netns h2

設置h2到h2-eth0的介面

# Delete the interface of h2-eth0 (Task 1.)

ip link delete h2-eth0

刪除h2到h2-eth0的介面

# Activate h2-eth0 and assign IP address (Task 1.)

ip netns exec h2 ip link set dev h2-eth0 up

ip netns exec h2 ip link set h2-eth0 address 00:0a:00:00:02:02

ip netns exec h2 ip addr add 10.0.1.2/24 dev h2-eth0

啟動h2-eth0和給他IP位置

# Disable all IPv6 on h2-eth0 (Task 1.)

ip netns exec h2 sysctl net.ipv6.conf.h2-eth0.disable\_ipv6=1

刪除所有h2-eth0的IPv6

# Set the gateway of h2 to 10.0.1.254 (Task 1.)

ip netns exec h2 ip route add default via 10.0.1.254

設定h2的gateway為10.0.1.254

輸入Esc + :wq 離開main.sh

執行main.sh以建立namespace

$ chmod +x main.sh

$ ./main.sh net

Task 2 – Define protocol via Scapy

進入Protocol.py

cd ..

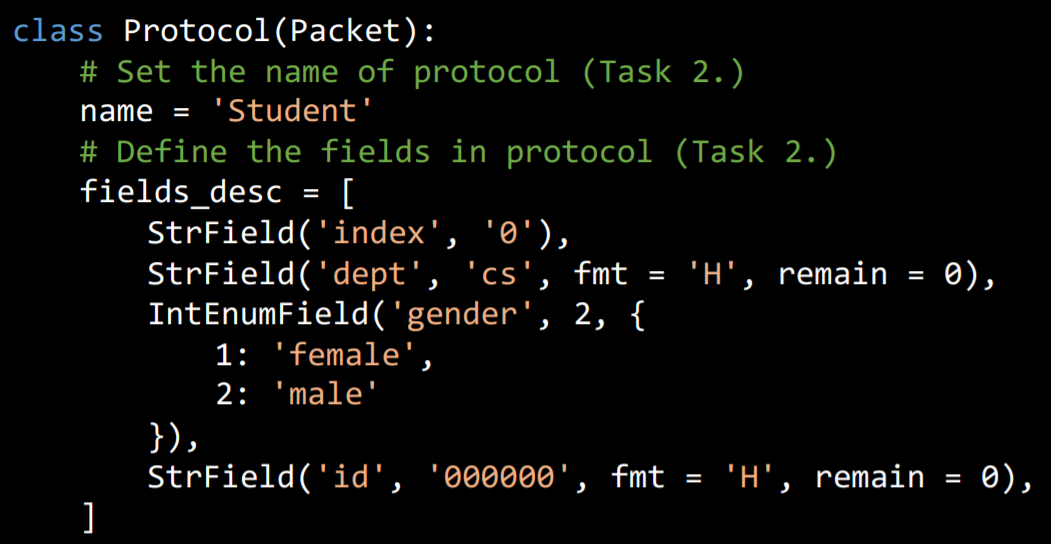
退回Packet\_Manipulation/src

vim Protocal.py

進入Protocol.py

輸入i + Enter 進入Insert模式

複製code到 ./src/Protocol.py 以定義自己的protocol: ID header format



輸入Esc + :wq 離開Protocol.py

Task 3 – Send packets

進入sender.py

cd ..

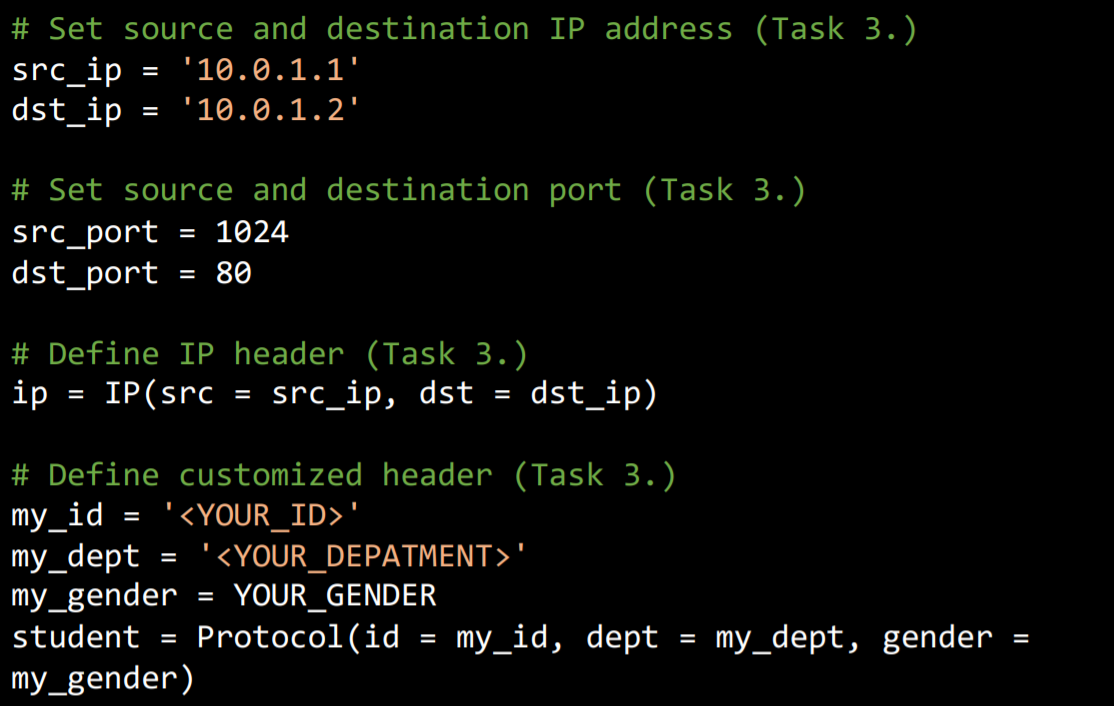
退回Packet\_Manipulation/src

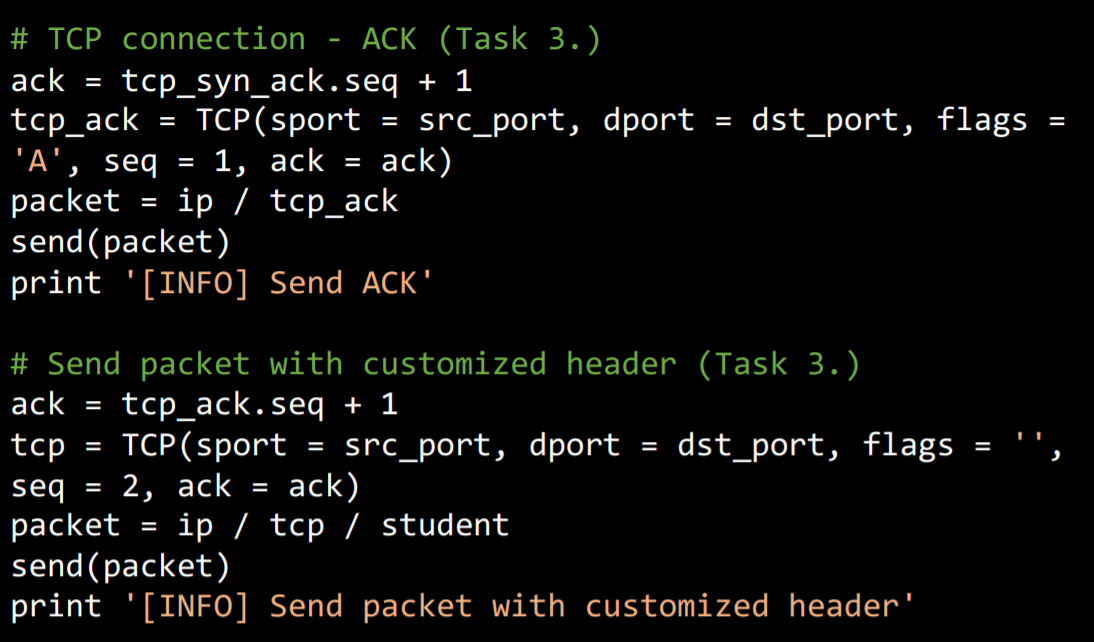
vim sender.py

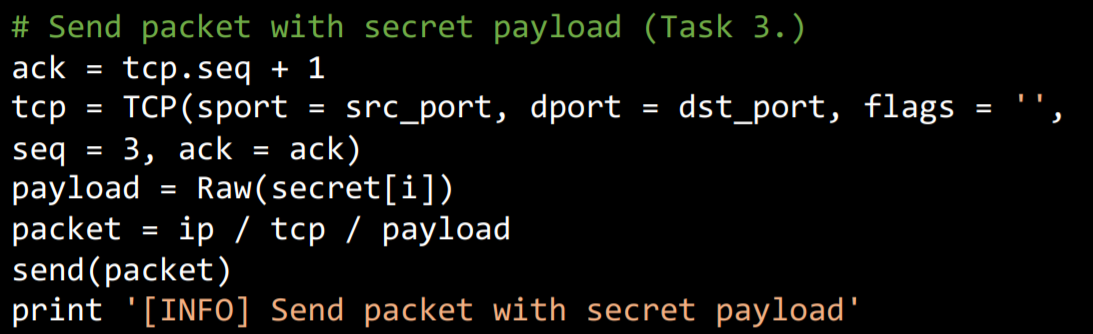
進入sender.py

輸入i + Enter 進入Insert模式

複製code到 ./src/sender.py 以設定我的封包 header







輸入Esc + :wq 離開sender.py

Task 4 – Sniff packets

進入receiver.py

cd ..

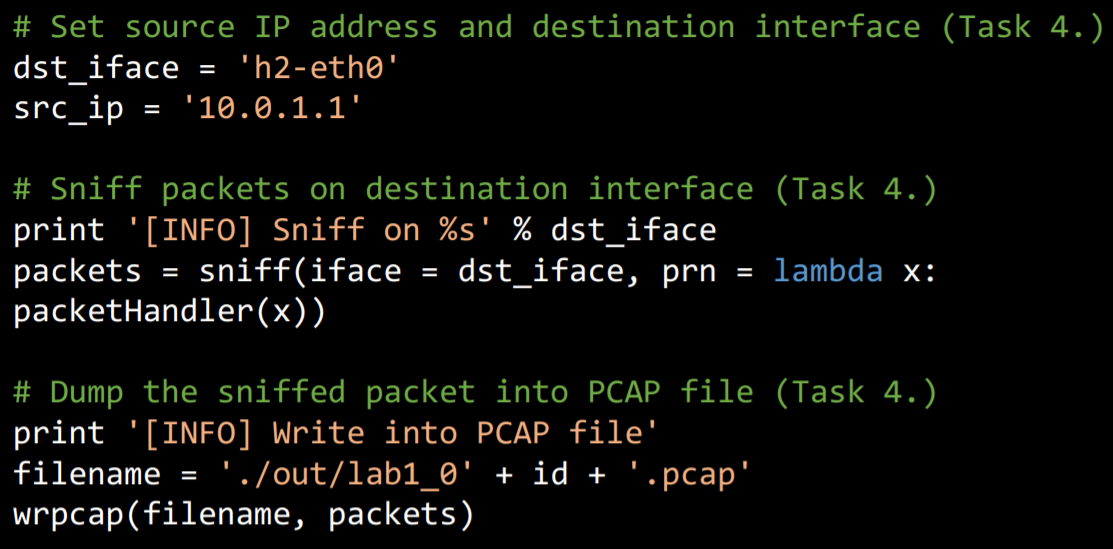
退回Packet\_Manipulation/src

vim receiver.py

進入receiver.py

輸入i + Enter 進入Insert模式

複製code到 ./src/receiver.py 以察覺和接收我的封包



輸入Esc + :wq 離開receiver.py

Task 5 – Run sender and receiver

打開 tmux 使 PieTTY 變成水平的兩個頁面

# Open tmux

$ tmux

打開tmux

# Open new pane in horizontal

Ctrl-b Shift-%

按Ctrl-b + Shift-% 以打開新的水平頁面

# Switch between two panes

Ctrl-b Arrow-right key

Ctrl-b Arrow-left key

按Ctrl-b + Arrow-right key切換到右邊頁面

按Ctrl-b + Arrow-left key切換到左邊頁面

轉換到h1跟h2的namespace

# Run namespace h1 in your left pane

$ ./scripts/main.sh run h1

在左邊的頁面進入namespace h1

Ctrl-b Arrow-right key

切換到右邊頁面

# Run namespace h2 in your right pane

$ ./scripts/main.sh run h2

在右邊的頁面進入namespace h2

先執行receiver.py

# Run receiver.py

h2> python receiver.py

執行receiver.py

接著執行sender.py

Ctrl-b Arrow-left key

切換到左邊頁面

# Run sender.py

h1> python sender.py

執行sender.py

停止接收封包

Ctrl-b Arrow-right key

切換到右邊頁面

Ctrl-c

跳出receiver

使用 tcpdump 展示我的 PCAP 檔案，在接收到所有封包後我會在./src/out/ 得到lab1\_ID.pcap 和recv\_secret.txt

# Dump the PCAP via tcpdump

$ tcpdump -qns 0 -X -r .pcap

Task 6 – Push your files to remote

Push my image to Docker Hub

# Create a new image from a container’s changes

$ docker commit cn2018\_c /cn2018\_lab1

Create a new image from a container’s changes

# Login to your Docker registry

$ docker login

登入docker

之後會要求輸入帳號和密碼

# Push an image to a registry

$ docker push /cn2018\_lab1

Push an image to my registry

把檔案丟進GitHub

# Get and set repository or global options

$ git config --global user.name "<NAME>"

$ git config --global user.email "<EMAIL>”

設置上傳的檔案的存放處(GitHub中)

# Add your files into staging area

$ git add .

Add my files into staging area

# Commit your files

$ git commit –m "Commit lab1 in class”

在資料後面加上註解

# Set the remote URL to your remote repository

$ git remote set-url origin [https://github.com/nctucn/lab1-<YOUR\_ID>.gi](https://github.com/nctucn/lab1-%3cYOUR_ID%3e.gi)t

設置新的到我的存放處的URL

# Push your files to remote repository

$ git push origin master

Push 我的存放處使它可以使用遠端

Task 7 – Load PCAP via Wireshark

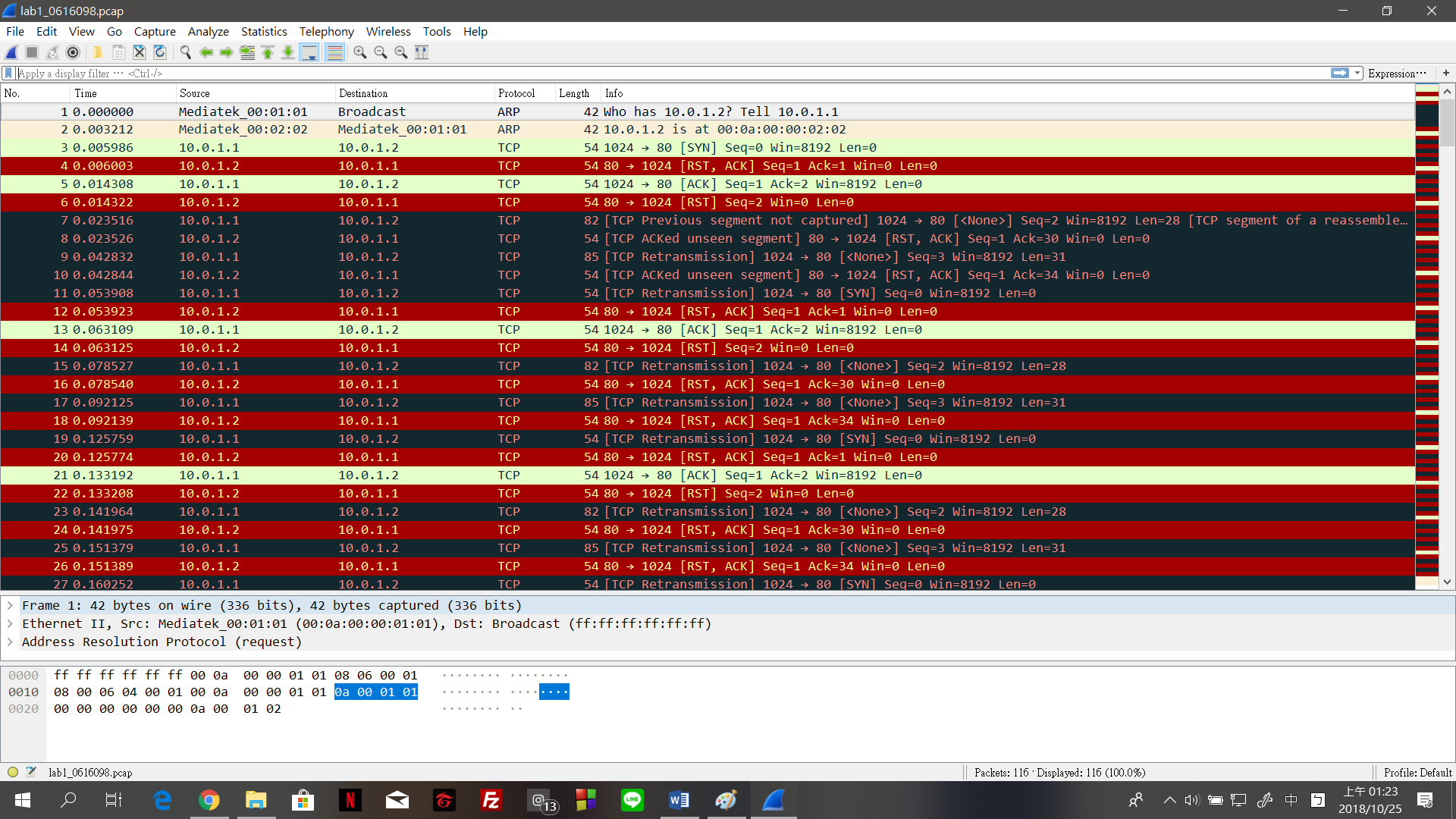
Download the code from GitHub

$ git clone <https://github.com/nctucn/lab1-yungshenglu.git>

Download Wireshark

Open the PCAP file using Wireshark with clicking ” open a capture file ” and find

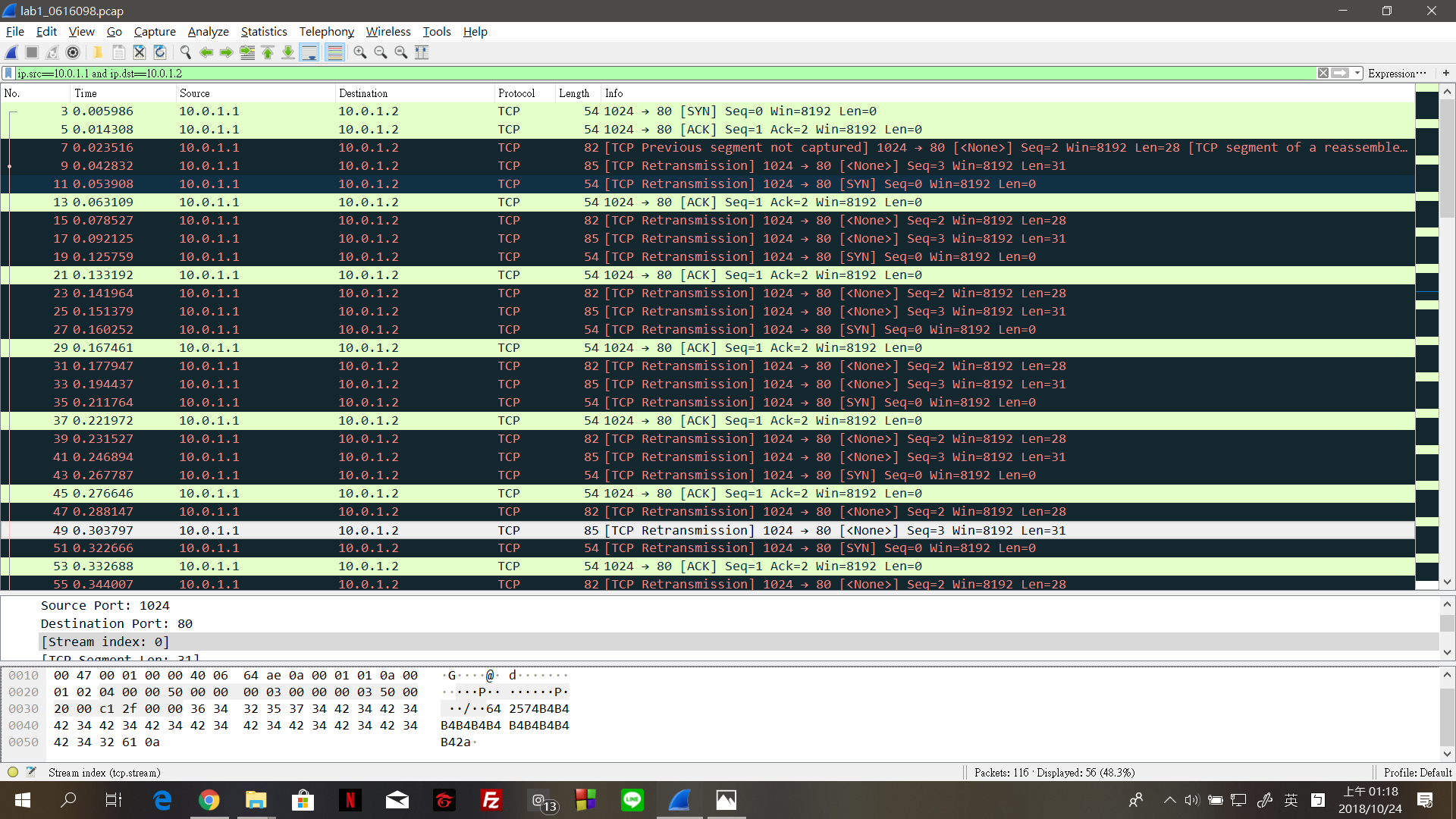
The PCAP file (./src/out/lab1\_yourID.pcap)



Task 8 – Filter the target packet

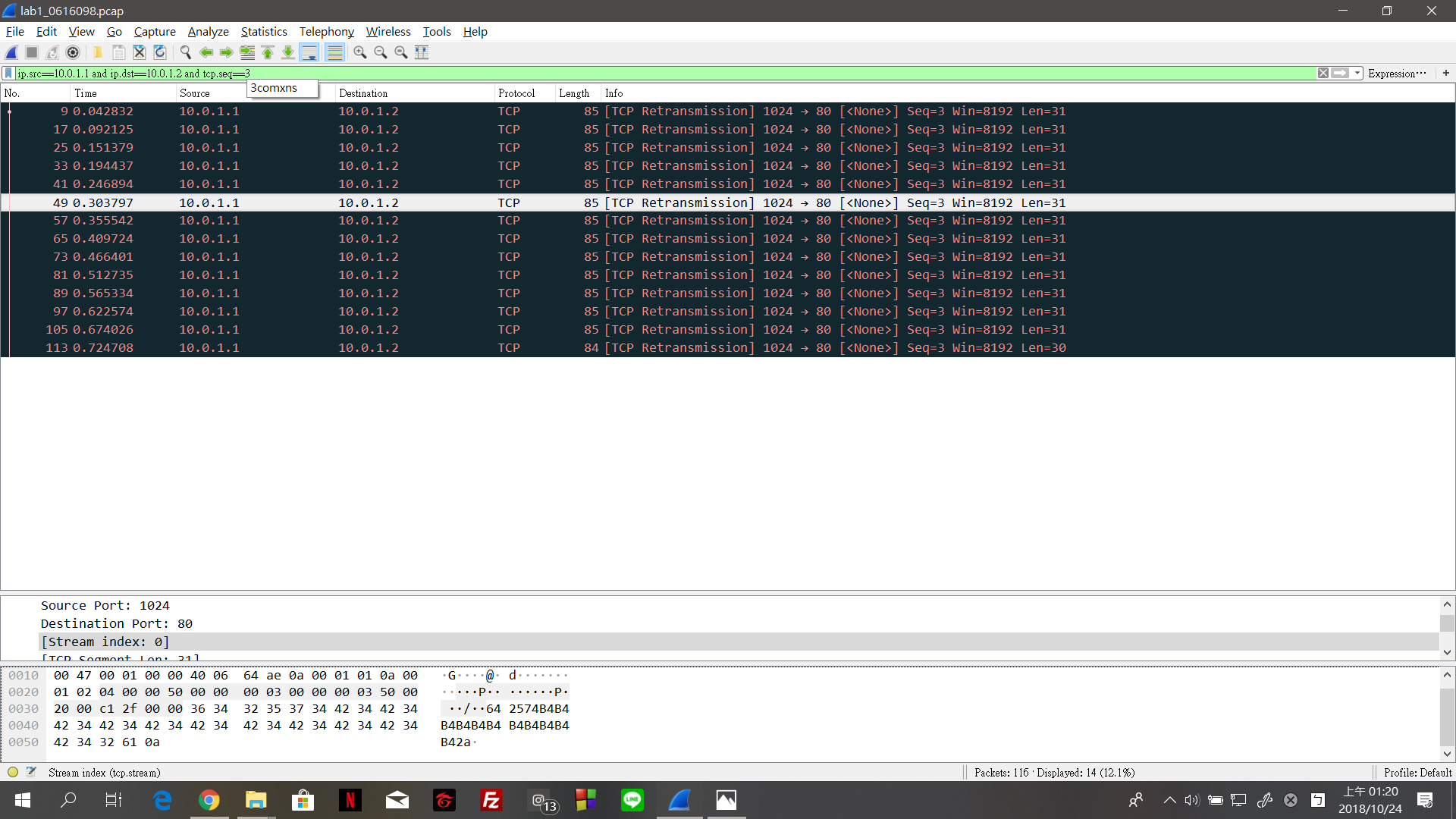
Filter the packets of our defined protocol

Filter rule: ip.src==10.0.1.1 and ip.dst==10.0.1.2



Filter the packets with the “secret” bits

Filter rule: ip.src==10.0.1.1 and ip.dst==10.0.1.2 and tcp.seq==3



What is my secret key?

89061608906160

How to find it?

The first digit in a ”secret” payload

Task 9 – Decode the secret key

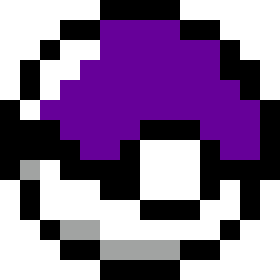
Input the secret key into ./src/decoder.py on local machine

Execute decoder.py

$ python decoder.py <YOUR\_SECRET\_KEY>

My result is?

The output file is in ./src/out/ and I get an image related to Pokemon



**BONUS**

1. What you have learned in this lab?

我學到了一些作業系統的指令，也了解了更多作業系統的事，跟平常在打的

code完全不一樣。也學到了網路封包的一些傳送接收的基本知識，還熟悉了一些程式的使用方法。所幸到最後可以了解每一個步驟的意義跟步驟。

1. What difficulty you have met in this lab?

在電腦教室哩，明明code是用複製的卻有無法執行的情況，直到後來到宿

舍重做才莫名其妙地過了。因為對整個LAB很陌生所以一開始完全沒有頭緒，不懂程式意義真的不知道自己到底在幹嘛，只知道一味地複製貼上，不過後來懂了。到最後也不知道哪個環節有問題，明明做好了檔案去沒出來，更影響了我之後的上傳的部分，幸好後來被助教發現是我的路徑好像有意外被改到。