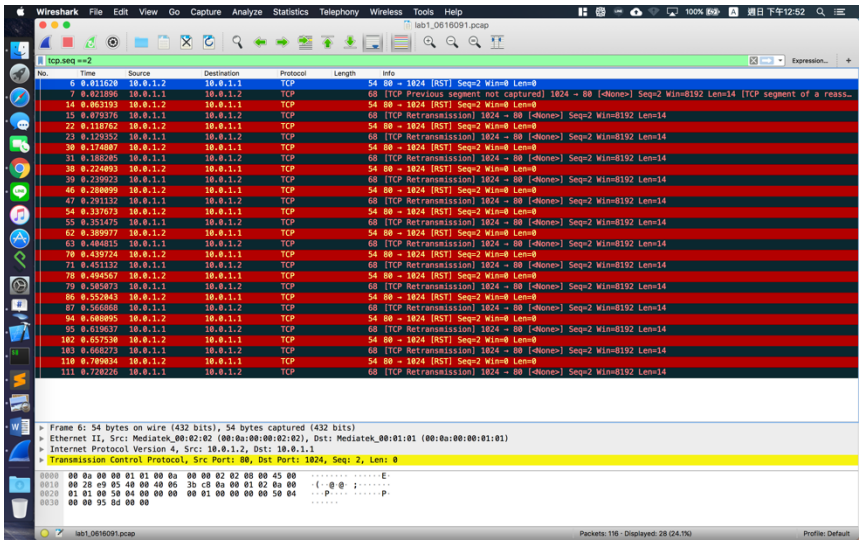
Student name: 陳昱螢 Student ID: 0616091 Department: CS

Part A. Questions

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

Ans: Use “tcp.seq == 2” to filter the packet with customized header.

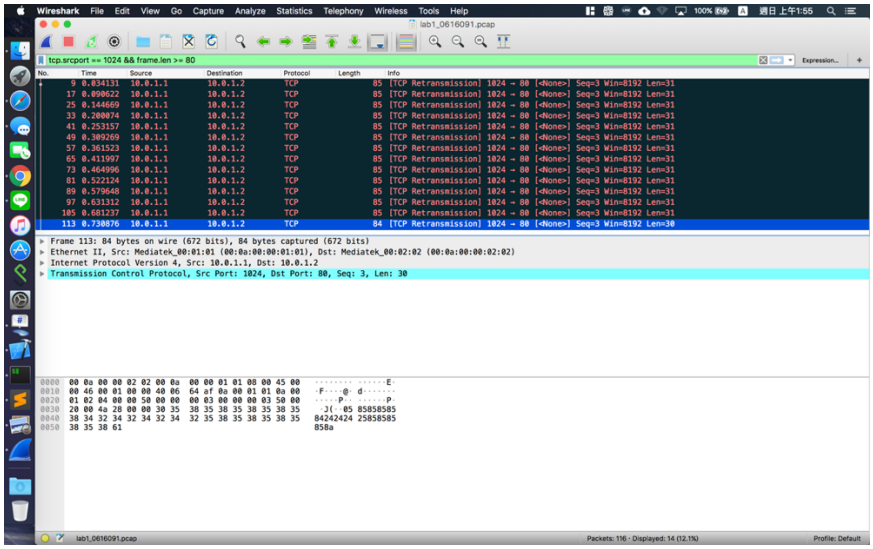
2. Show the screenshot of filtering the packet with customized header.



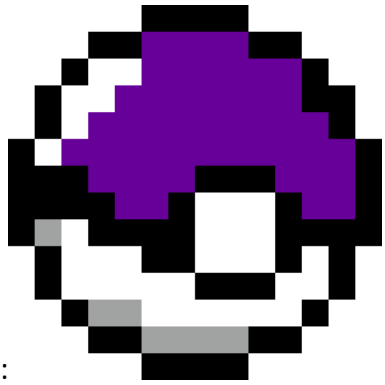
3. What is your command to filter the packet with “secret” payload on Wireshark?

Ans: Use “tcp.srcport == 1024 && frame.len >= 80“ to filter the packet.

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



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

Ans: 

Part B. Description

- Task 1 – Environment setup
  1. Setup github and git
     - 依序在終端機輸入，第一句是從 github 上下載檔案，第二三句則是設定 git 的 username 以及 email。第三句則是設定遠端伺服器的網址，第五句則是將 master 這個分支推向 origin 的位置。

```
$ git clone
https://github.com/yungshenglu/Packet_Manipulation
$ git config --global user.name "<NAME>"
$ git config --global user.email "<EMAIL>"
$ git remote set-url origin
https://github.com/nctucn/lab1-<GITHUB_ID>.git
$ git push origin master
```

  2. Setup Dockerfile
     - 第一句是從 yungshenglu/ubuntu-env:16.04 下載基本的 image，第二句則是更新所有軟體，第三句則是安裝 tcpdump，第四句則是安裝 scapy，第五句則是設定 container 在運作時偵聽的端口，第五句則是從 github 下載檔案。

```
FROM yungshenglu/ubuntu-env:16.04
RUN apt-get update
RUN apt-get install -y tcpdump
RUN pip install scapy
EXPOSE 22
RUN git clone
https://github.com/yungshenglu/Packet_Manipulation.git
```

  3. Build the Docker
     - 第一句是根據 dockerfile 建立 image，第二句是根據

剛才的 image 建立 container，-p 9487:22 是指說把 container 內的 22port 跟外面的 9487port 接通，因此可以用 127.0.0.1:9487 連到 container。Privileged 則是取得權限。第三句則是將 port 對應到 22 去。

```
docker build -t cn2018 .
docker run -d -p 9487:22 --privileged --name cn2018_c
cn2018
docker port cn2018_c 22
```

4. Login to your Docker container using SSH
   - Windows：Use PieTTY to connect
     - IP address: 127.0.0.1
     - Port:9487
     - Login: root
     - Password: cn2018
   - Mac/Ubuntu：Use terminal
     - 輸入 ssh root@0.0.0.0 –p 9487
     - Password: cn2018
5. Create namespace in main.sh
   - 在./src/scripts/main.sh 裡面輸入下列的 code，h1 已經被創建好了，所以下列是創建 h2 的 namespace。第一句跟第二句分別是創建跟刪除 h2 namespace 用的。第三句是設定 h2 的 interface。第四句是將 h2-eth0 放到 h2 這個 namespace 裡面，第五句則是刪除 h2-eth0 的 interface。第六～八句則是設定 h2-eth0 的 ip 位置並且啟用。第九句則是讓 IPv6 不能作用在 h2-eth0。第十句是設定 h2 的 gateway 是 10.0.1.254。

```
# Create h2 network namespaces (Task 1.)
ip netns add h2
# Delete h2 network namespaces (Task 1.)
ip netns del h2
# Bring up the lookup interface in h2 (Task 1.)
ip netns exec h2 ip link set lo up
# Set the interface of h2 to h2-eth0 (Task 1.)
ip link set h2-eth0 netns h2
# Delete the interface of h2-eth0 (Task 1.)
ip link delete h2-eth0
# 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
# Disable all IPv6 on h2-eth0 (Task 1.)
ip netns exec h2 sysctl net.ipv6.conf.h2-eth0.disable_ipv6=1
# 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
```

6. Run main.sh to build the namespace
    - Chmod +x 是添加執行權限，讓 main.sh 可以順利執行。./main.sh net 則是執行 main.sh 這個檔案，並且使用 net 的功能。

```
$ chmod +x main.sh
$ ./main.sh net
```

- Task 2 – Define protocol via Scapy
    1. Define your protocol: ID header format
        - 在 Protocol.py 裡新增下列的 code，code 的內容是在設定基本的資訊及定義。

```
class Protocol(Packet):
    # Set the name of protocol (Task 2.)
    name = 'Student'
    # Define the fields in protocol (Task 2.)
    fields_desc = [
        StrField('index', '0'),
        StrField('dept', 'cs', fmt = 'H', remain = 0),
        IntEnumField('gender', 2, {
            1: 'female',
            2: 'male'
        }),
        StrField('id', '000000', fmt = 'H', remain = 0),
    ]
```

- Task 3 – Send packets
    1. Set your own packet header
        - 在 sender.py 新增下列的 code，第 1 跟第 2 句是在定義來源及目的地的 ip 位置，第 3 跟第 4 句則是定義來源跟目的地的 port。第五句是在定義 IP header 的內容，剩下的五句則是定義自己的 header 的資訊。

```
# Set source and destination IP address (Task 3.)
src_ip = '10.0.1.1'
dst_ip = '10.0.1.2'

# Set source and destination port (Task 3.)
src_port = 1024
dst_port = 80

# Define IP header (Task 3.)
ip = IP(src = src_ip, dst = dst_ip)

# Define customized header (Task 3.)
my_id = '<YOUR_ID>'
my_dept = '<YOUR_DEPATMENT>'
my_gender = YOUR_GENDER
student = Protocol(id = my_id, dept = my_dept, gender =
my_gender)
```

2. Send packets
   - 在 sender.py 新增下列的 code，第一張圖片裡的 code 是在定義 ACK 的內容以及自己 header 的內容，然後傳遞出去。第二張圖片的 code 則是定義傳遞 secret payload 的內容。

```
# TCP connection - ACK (Task 3.)
ack = tcp_syn_ack.seq + 1
tcp_ack = TCP(sport = src_port, dport = dst_port, flags =
'A', seq = 1, ack = ack)
packet = ip / tcp_ack
send(packet)
print '[INFO] Send ACK'

# Send packet with customized header (Task 3.)
ack = tcp_ack.seq + 1
tcp = TCP(sport = src_port, dport = dst_port, flags = '',
seq = 2, ack = ack)
packet = ip / tcp / student
send(packet)
print '[INFO] Send packet with customized header'
```

```
# Send packet with secret payload (Task 3.)
ack = tcp.seq + 1
tcp = TCP(sport = src_port, dport = dst_port, flags = '',
seq = 3, ack = ack)
payload = Raw(secret[i])
packet = ip / tcp / payload
send(packet)
print '[INFO] Send packet with secret payload'
```

- Task 4 – Sniff packets
  1. Receive and sniff packets
     - 在 receiver.py 裡新增下列的 code，前面兩行是先設置來源的 ip 位置跟目的地。再來則是在目的地的介面 print”Sniff on 目的地”，最後先 print 出”write into PCAP file”後，便將所截取得到資訊寫入 pcap 檔案。

```
# Set source IP address and destination interface (Task 4.)
dst_iface = 'h2-eth0'
src_ip = '10.0.1.1'

# Sniff packets on destination interface (Task 4.)
print '[INFO] Sniff on %s' % dst_iface
packets = sniff(iface = dst_iface, prn = lambda x:
packetHandler(x))

# Dump the sniffed packet into PCAP file (Task 4.)
print '[INFO] Write into PCAP file'
filename = './out/lab1_0' + id + '.pcap'
wrpcap(filename, packets)
```

- Task 5 – Run sender and receiver
  1. Open tmux with horizontal two panes
     - 在終端機使用 tmux，再用下列的按鍵，將視窗分割成兩個視窗。

```
# Hint: Keep your path in ./src/
# Open tmux
$ tmux
# Open new pane in horizontal
Ctrl-b
Shift-%
# Switch between two panes
Ctrl-b
Arrow-left/right key
```

2. Switch into two namespaces
   - 將左邊的視窗開啟名為 h1 的 namespace，再右邊的視窗開啟名為 h2 的 namespace。

```
# Run namespace h1 in your left pane
$ ./scripts/main.sh run h1
# Run namespace h2 in your right pane
$ ./scripts/main.sh run h2
```

3. Run receiver.py first
   - 上面是切換左右視窗的指令。再來是先在 h2 開啟 receiver.py，這樣才不會遺漏訊息。

```
# Switch between two panes
Ctrl-b
Arrow-right key
# Run receiver.py
h2> python receiver.py
```

4. Run sender.py
   - 上面是切換左右視窗的指令。再來是在 h1 開啟 sender.py，開始傳遞訊息。

```
# Switch between two panes
Ctrl-b
Arrow-left key
# Run sender.py
h1> python sender.py
```

5. Use tcpdump to show your PCAP file
   - 使用 tcpdump 來解讀 pcap 的檔案，-r 的意思是從指定的文件讀取數據包數據。

```
# Dump the PCAP via tcpdump
$ tcpdump -qns 0 -X -r <FILENAME>.pcap
```

- Task 6 – Push your files to remote
  1. Push your image to Docker Hub
     - docker commit 是根據你 container 的 change 創造新的 image，而 docker login 是在終端機登入你的 docker 帳號，docker push 則是將你的 image 傳到 Docker Hub

```
# Create a new image from a container's changes
$ docker commit cn2018_c <DOCKER_HUB_ID>/cn2018_lab1
# Login to your Docker registry
$ docker login
# Push an image to a registry
$ docker push <DOCKER_HUB_ID>/cn2018_lab1
```

  2. Push your files to GitHub
     - 設定 git 所需要的名稱及信箱。

```
# Get and set repository or global options
$ git config --global user.name "<NAME>"
$ git config --global user.email "<EMAIL>"
```

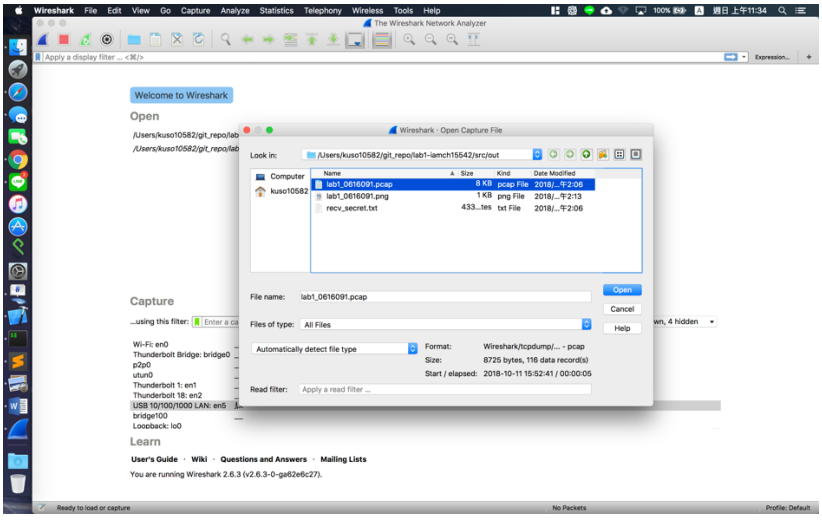
     - add 是將資料夾裡的檔案讓 git 追蹤，commit 則是將暫存區的檔案存檔，後面的-m “Commit lab1 in class”，則是說明你在這次 commit 做了什麼事。

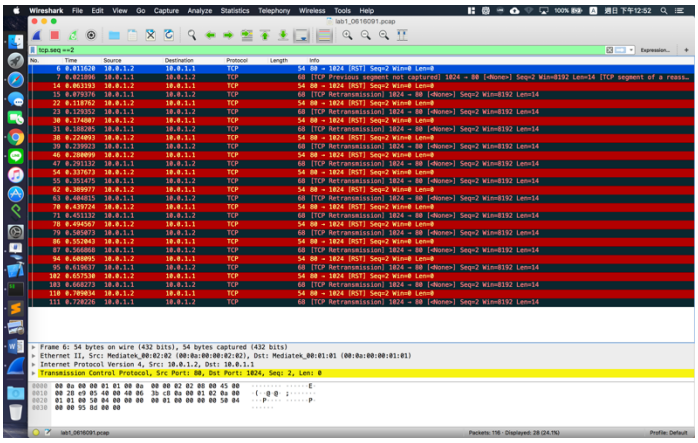
```
# Add your files into staging area
$ git add .
# Commit your files
$ git commit –m "Commit lab1 in class”
```

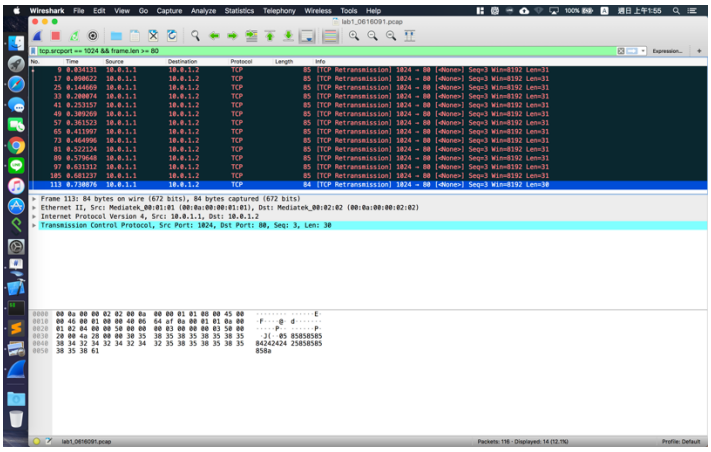
     - git remote set-url origin 是拿來設定遠端伺服器的網址，而 git push origin master 則是把 master 這個分支的內容推向 origin 的位置，若 origin 的遠端 server 並沒有 master 這個分支的話，便會建立一個叫做 master 的分支。

```
# Set the remote URL to your remote repository
$ git remote set-url origin
https://github.com/nctucn/lab1-<YOUR_ID>.git
# Push your files to remote repository
$ git push origin master
```

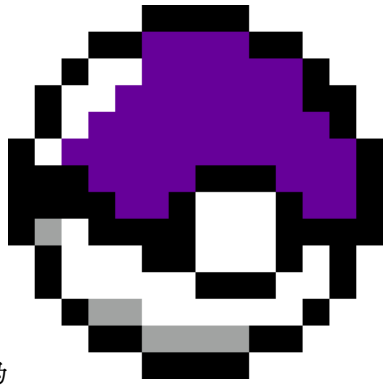
- Task 7 – Load PCAP via Wireshark
  1. Download your code from GitHub
     - 在終端機輸入 git clone https://iamch15542@github.com/nctucn/lab1-iamch15542.git/

2. Install Wireshark 2.6.3
   - 因為我是使用 macos，所以我是到 https://www.wireshark.org/download.html 他的官網下載
3. Open the PCAP file using Wireshark
   - 直接選取資料夾裡面的 pcap 檔案就可以了
   - 

- Task 8 – Filter the target packets
  1. Filter the packets of our defined protocol
     - Use “tcp.seq == 2” to filter the packets
     - 
  2. Filter the packets with the “secret” bits
     - Use “tcp.seq == 3” or “tcp.srcport == 1024 && frame.len >= 80” to filter the packets



- Task 9 – Decode the secret key
  1. Input the secret key into ./src/decoder.py
     - 在終端機執行 decoder.py
     - 輸入 python decoder.py 19061601906160
  2. Will have output in ./src/out/lab1_0616091.png
     - 圖片結果為



Part C. Bonus

1. What you have learned in this lab?

Ans: 學到蠻多東西的，很多終端機裡的指令都是這次 lab 學到的，還有要能自己找尋解決的方法，而不是遇到困難就找助教問。還有學到如何自己創建兩個 namespace 來互相傳遞訊息。

2. What difficulty you have met in this lab?

Ans: 遇到的困難其實蠻多的，一開始是跟我說 h1 的 namespace 已經存在，所以不能創建，去查了才知道可以用 ip netns delete h1 來刪除，刪除後就可以正常創建了。然後創造 namespace 出來後，才知道 sender, receiver, Protocol.py 要在裡面改才有用，所以又花時間改了一下。

回來要做 task7 時，一直無法 clone 成功，會出現” remote: Repository not found.”等等的字樣，查過 google 才知道，因為 repository 是 private 的，因此網

址要加上自己的帳號才行 ex: https://github.com/....要改成 https://帳號@github.com/....這樣才會成功。在 set-url 也是遇到相同的事情，也是要加上帳號才行。