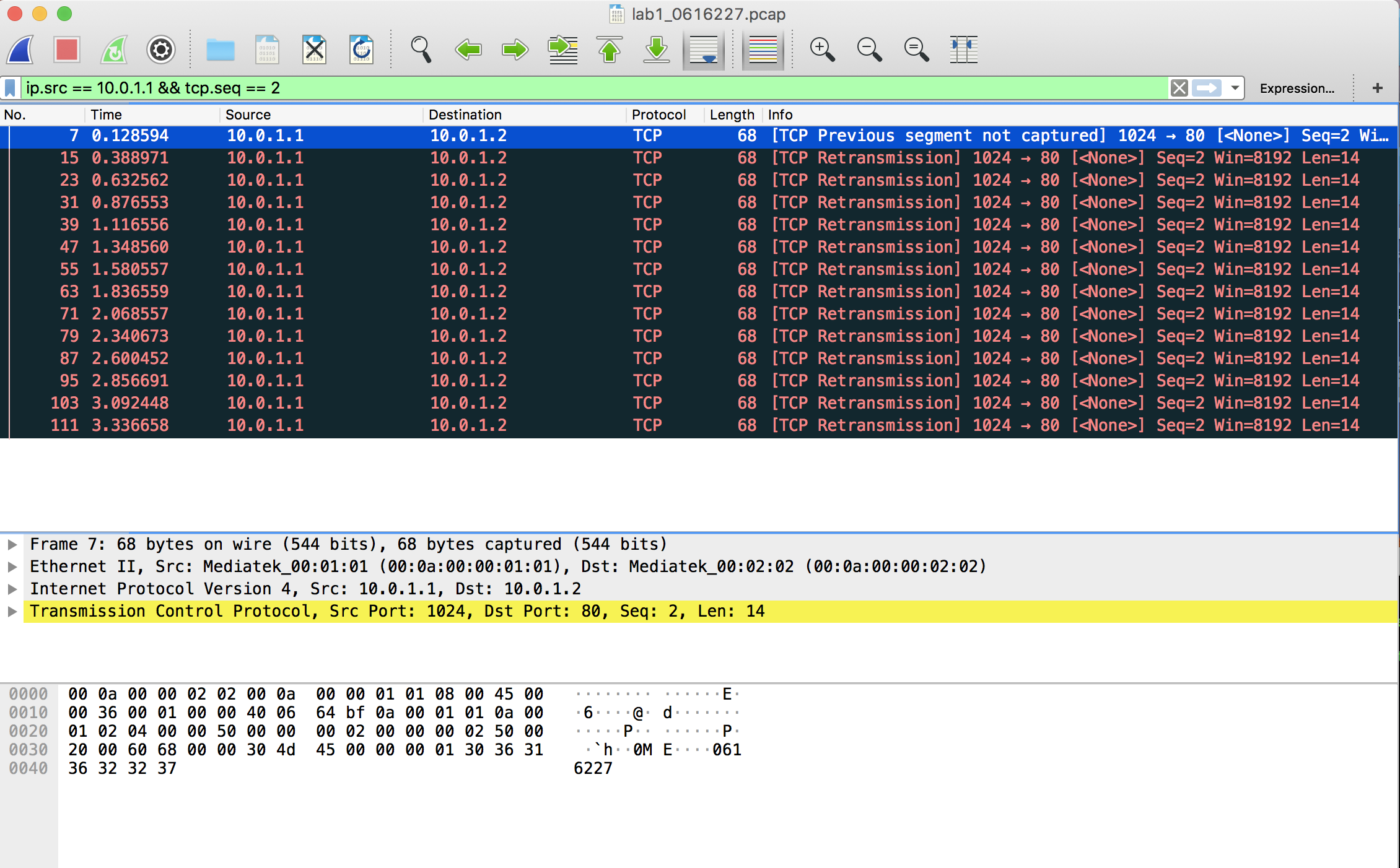
**0616227 陳亭妤 HW1\_Task10.Report**

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

ip.src == 10.0.1.1 && tcp.seq == 2 (從Task3&4可知)

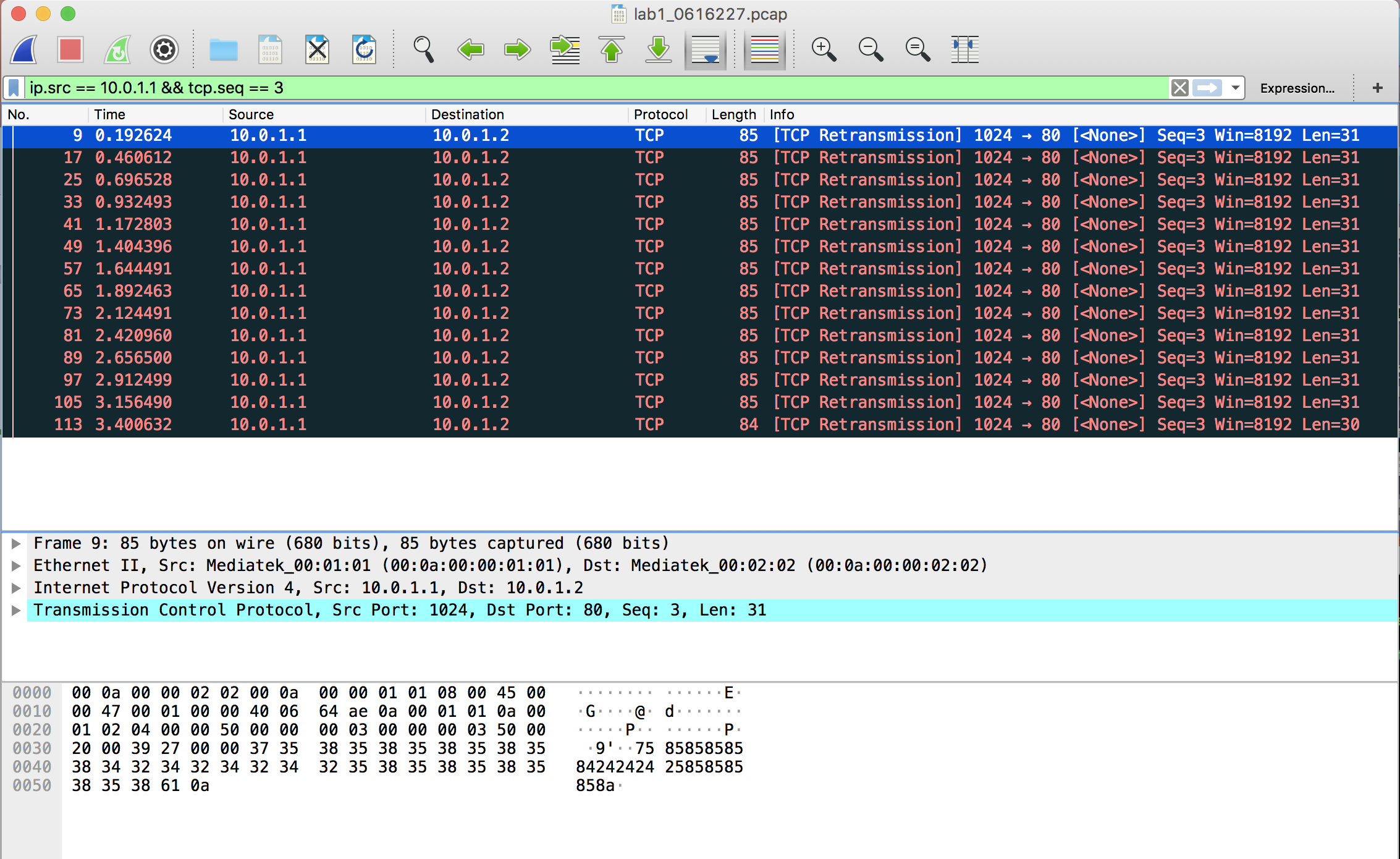
1. Show the screenshot of filtering the packet with customized header.



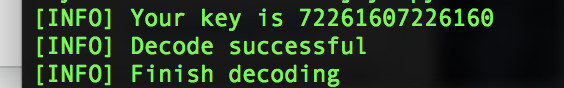
1. What is your command to filter the packet with “secret” payload on Wireshark?

ip.src == 10.0.1.1 && tcp.seq == 3 (從Task3&4可知)

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



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



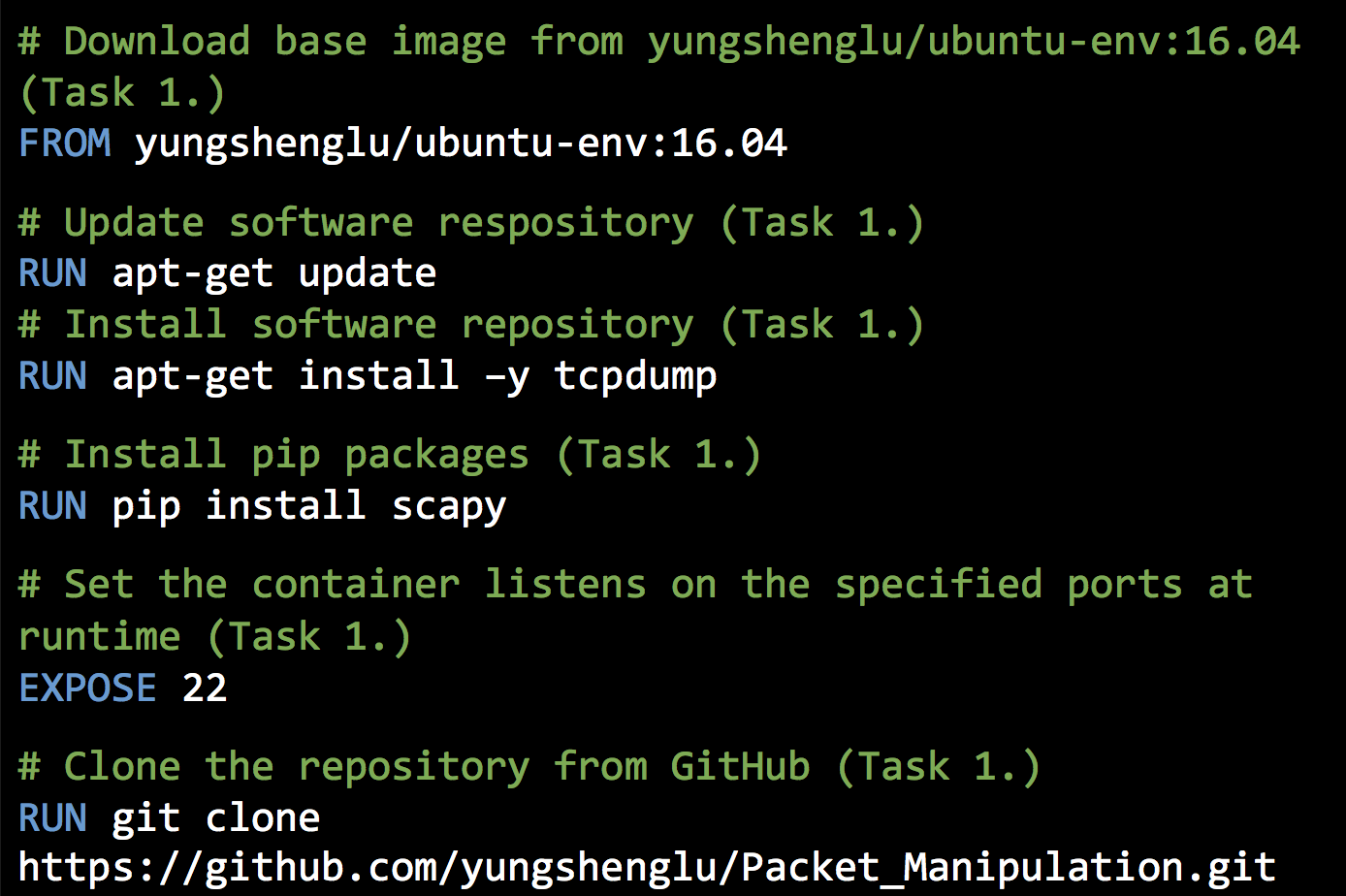
**＊Describe each step in this lab in detail**

Task1 :

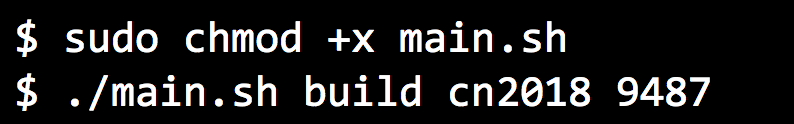
將所需的資料從GitHub上載下來



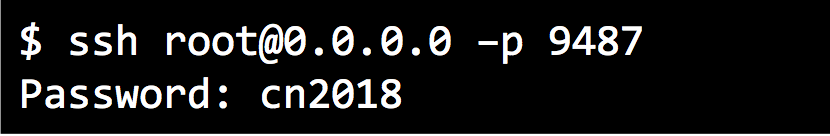
建立Docker的環境



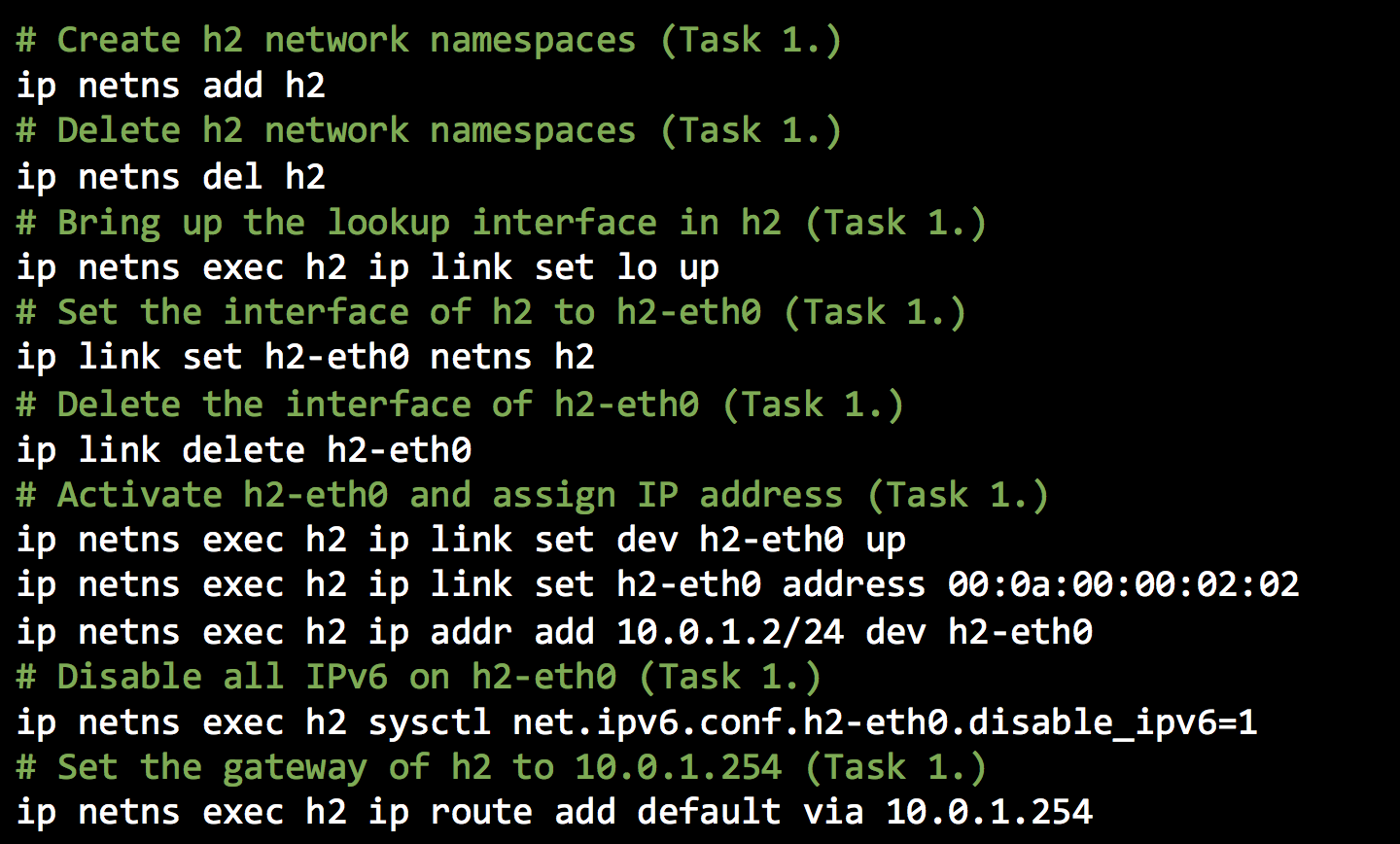
執行上面的程式檔 (chmod +x file => 將檔案變為可執行檔)



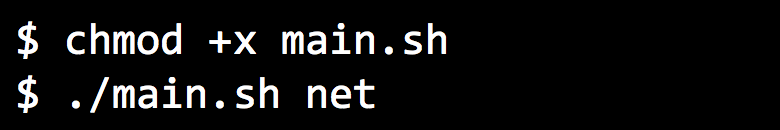
將終端機連線至Docker



建立h2的namespace

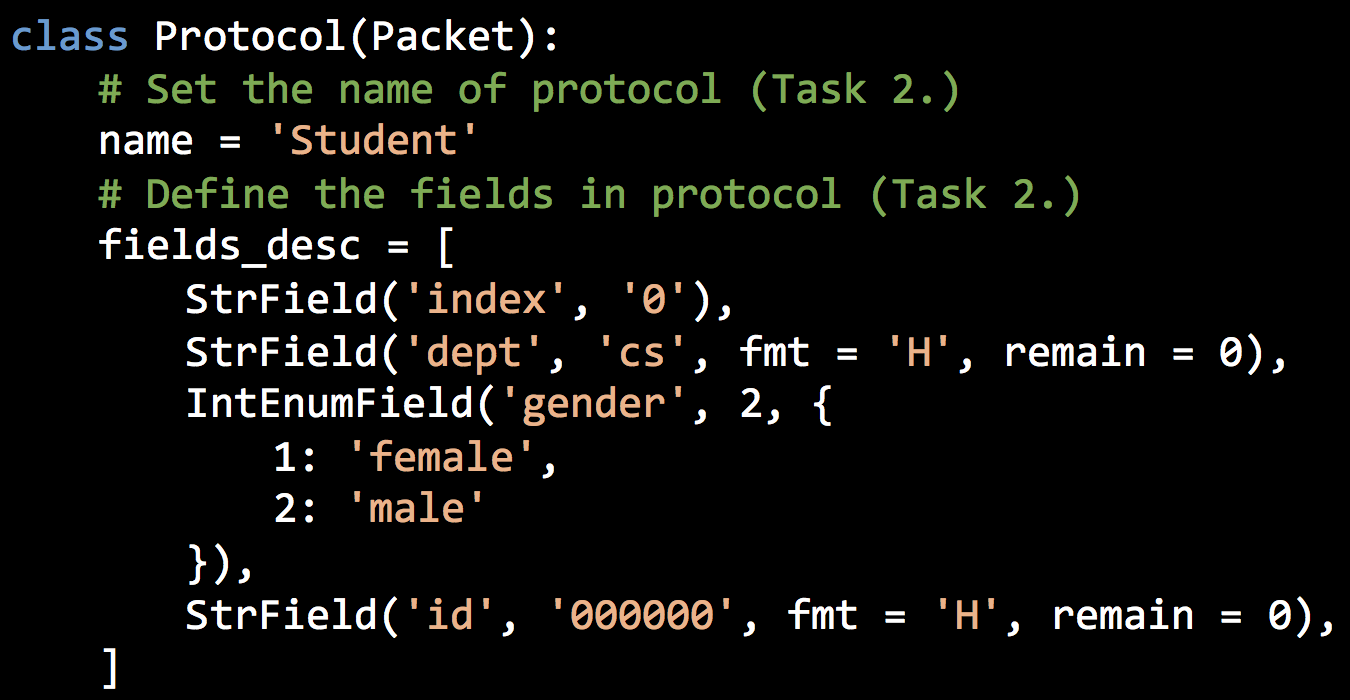


執行上述的程式來建立h2的namespace



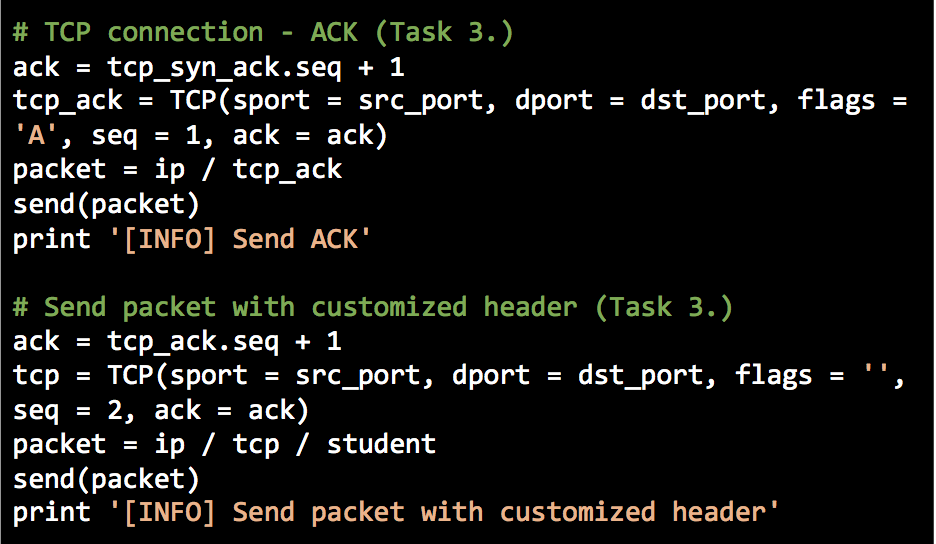
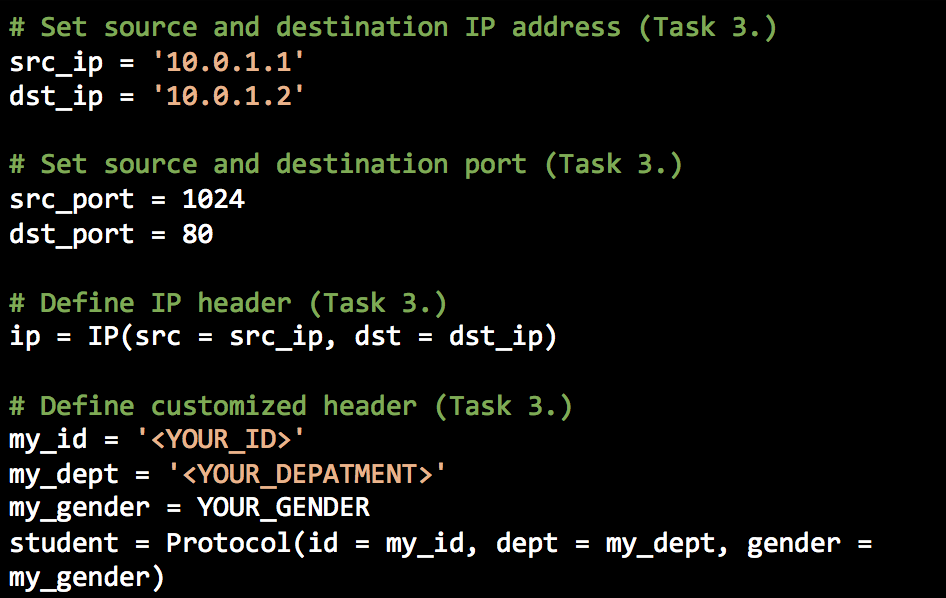
Task2 :

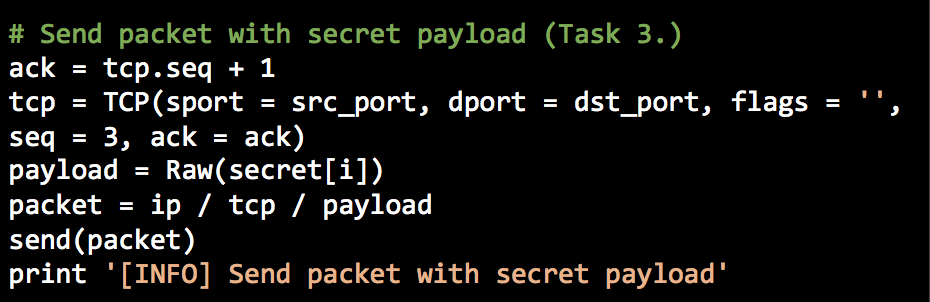
設定自己的protocol以及ID header 的格式



Task3 :

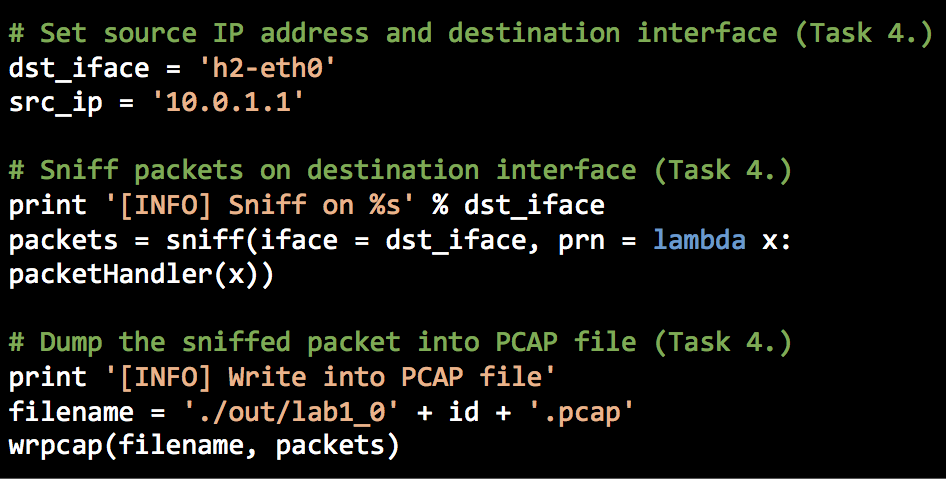
設定整個要傳送出去的封包





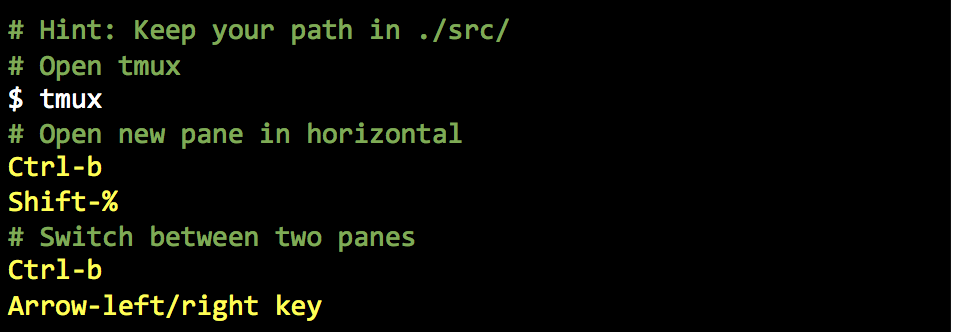
Task4 :

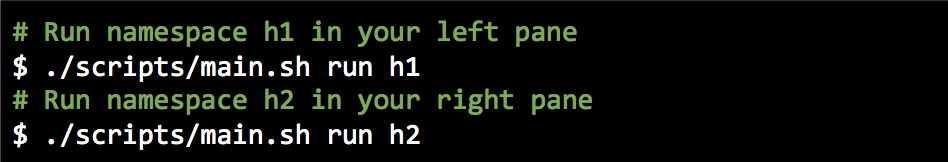
設定擷取封包的條件，並將擷取下來的封包讀進一個.pcap檔



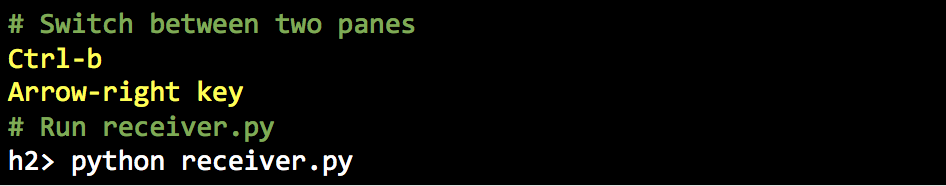
Task5 :

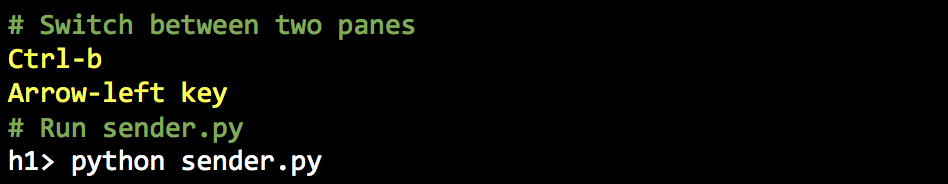
用tmux將一個視窗切割成左右兩邊(左邊是傳送端，右邊是接收端)



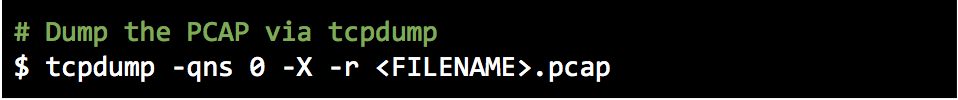


先執行receiver.py，再執行sender.py，不然先跑sender.py會收不到東西





用tcpdump來生成一個.pcap檔



Task6 :

將圖片上傳至Docker Hub



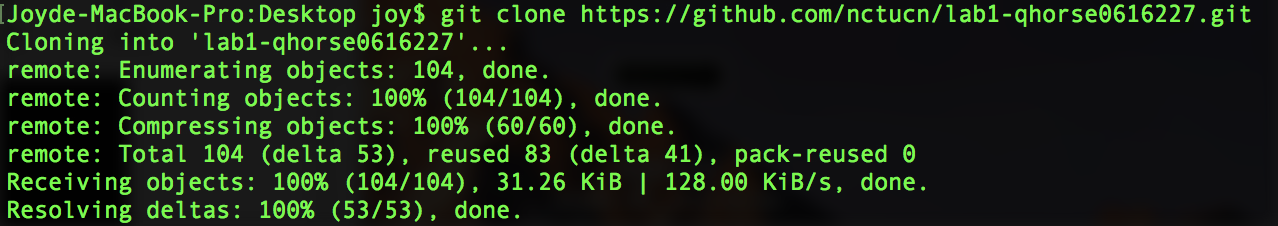
將檔案上傳至Git Hub



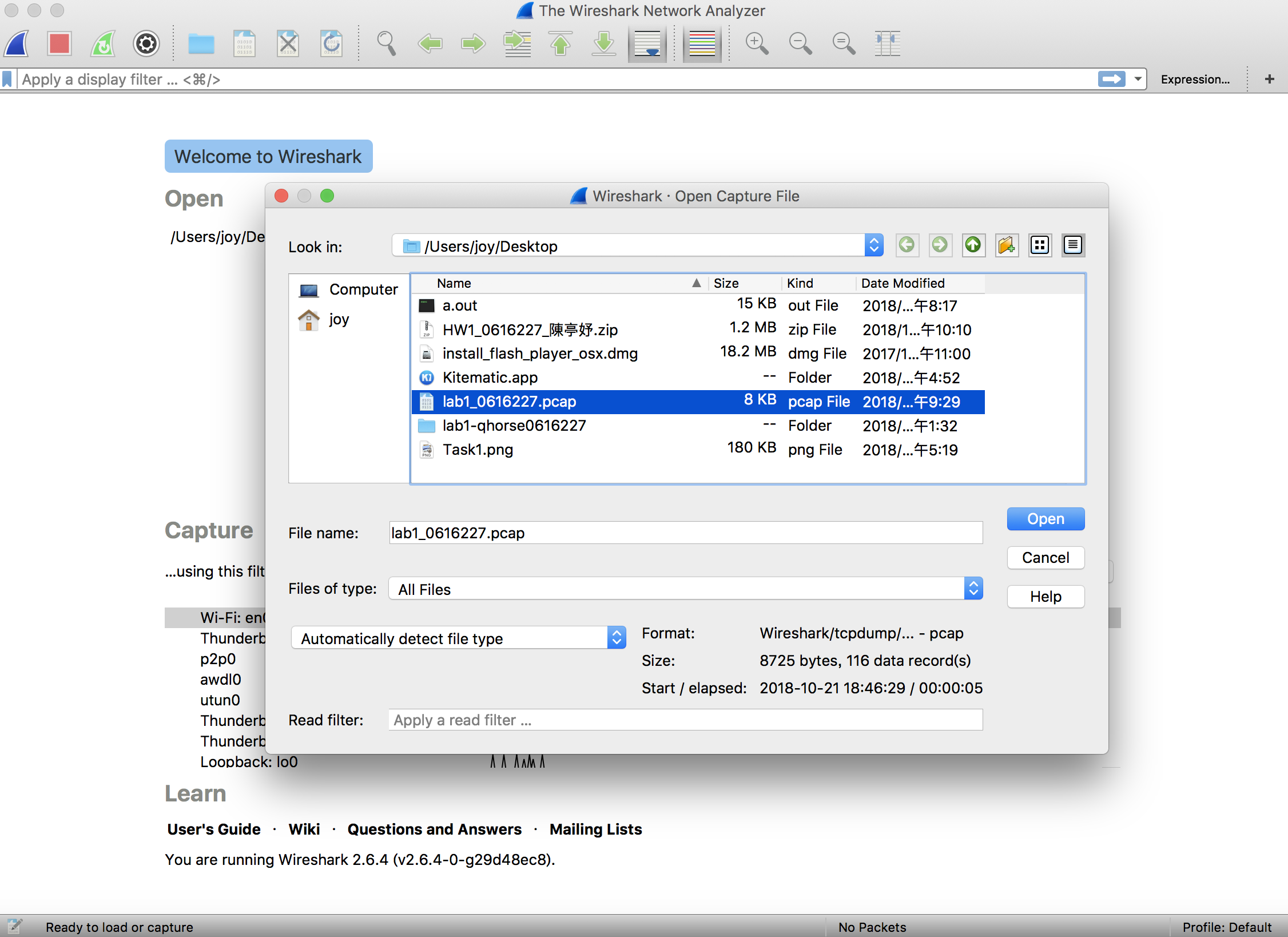
============================== HW ================================

Task7 :

將code從自己的Git Hub上下載下來



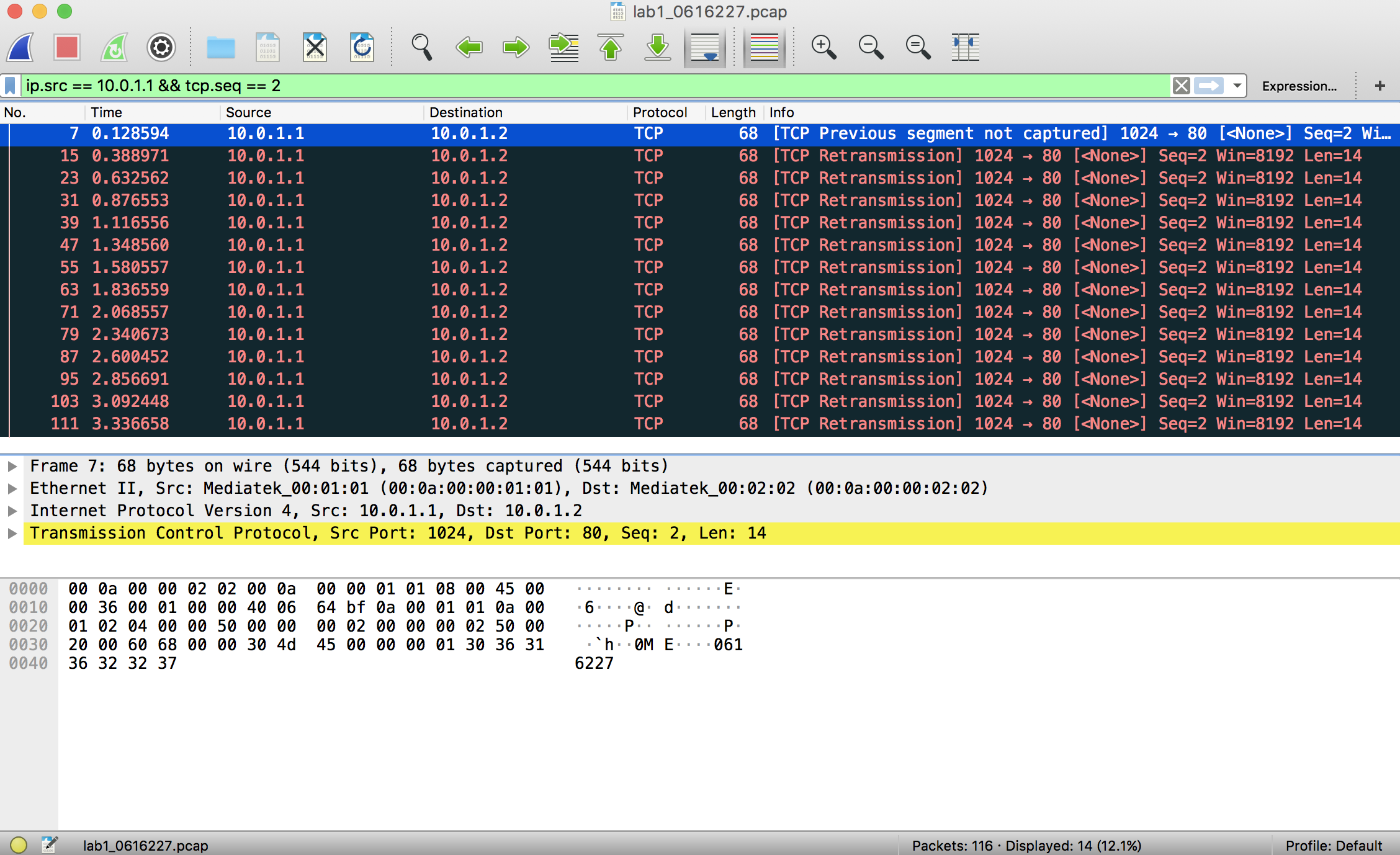
用wireshark打開PCAP file



Task8 :

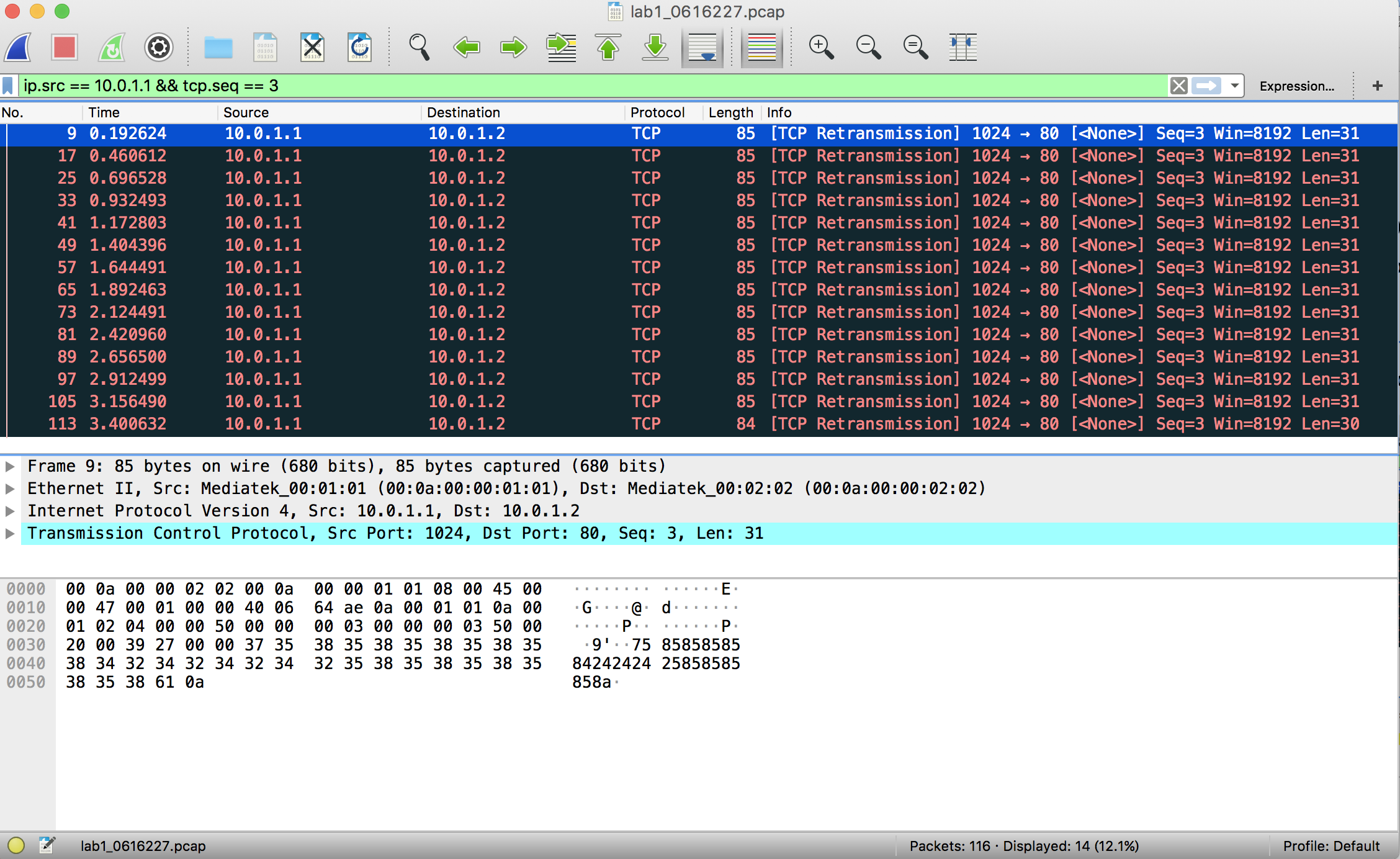
依照自己訂的protocol過濾目標的封包

Filter rule : ip.src == 10.0.1.1 && tcp.seq == 2



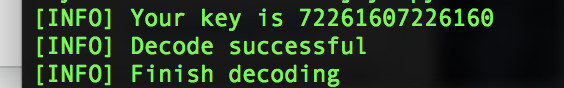
Filter the packets with the “secret” bits

Filter rule : ip.src == 10.0.1.1 && tcp.seq == 3

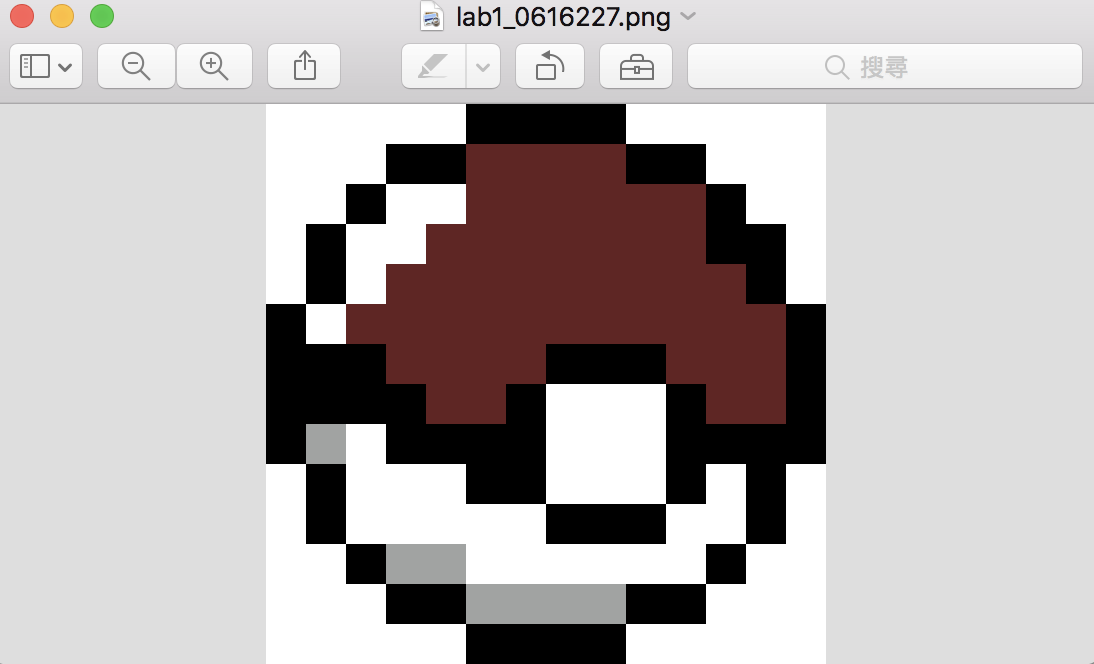


Task9 :

Decode the secret key



在./src/out/獲得一張圖片如下



**Bonus**

1. What you have learned in this lab?

我學會了一些Git Hub的基本指令，也稍微了解tmux的用法，透過實作也更加懂得傳送封包時的原理。

2. What difficulty you have met in this lab?

因為原本就不太熟悉終端機，更沒有用過Git Hub，所以一開始在lab花了一段時間才比較知道要怎麼操作，原本在lab用學校電腦(Windows)做會有三個視窗可以打指令，會有點搞混不知道哪個指令要打在哪裡，不過回去之後用自己的筆電(MacOS)就比較沒有這個困擾．