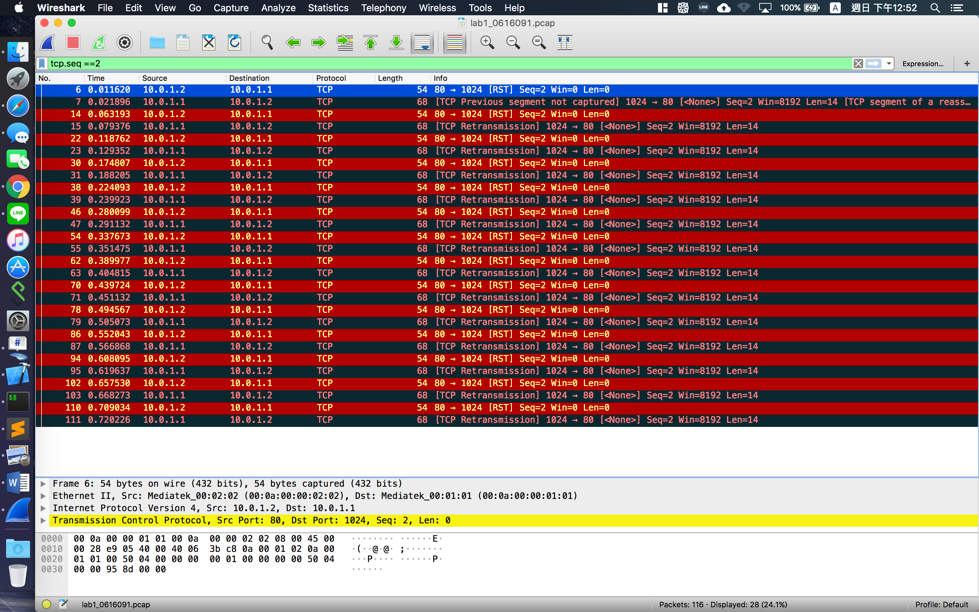
Student name: 陳昱螢Student ID: 0616091 Department: CS

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.

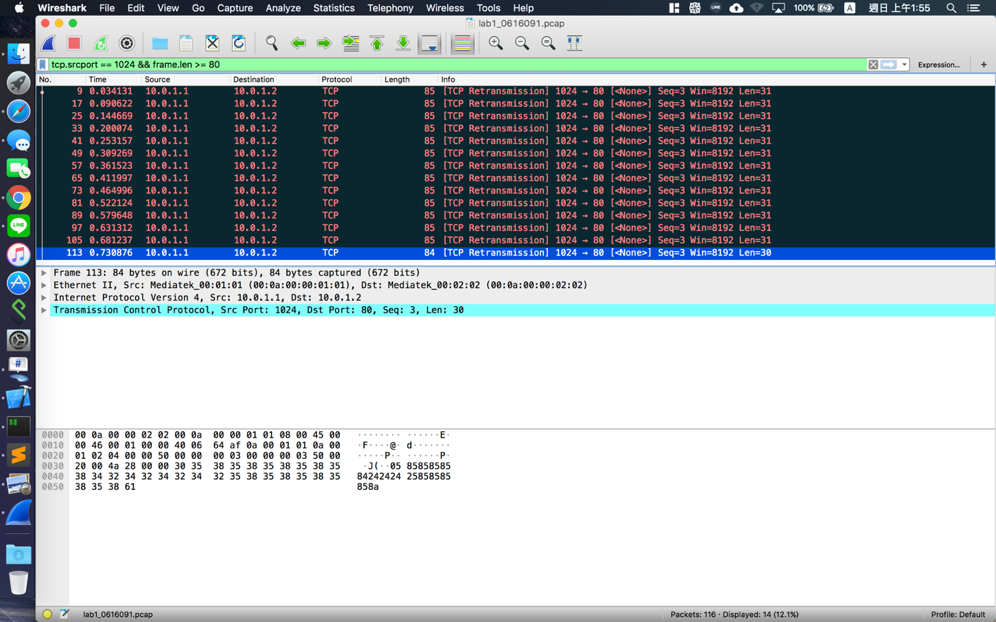
2. Show the screenshot of filtering the packet with customized header.



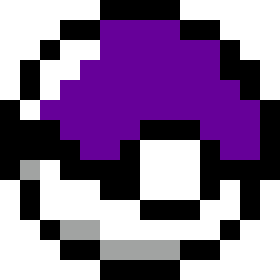
3. What is your command to filter the packet with “secret” payload on Wireshark?

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

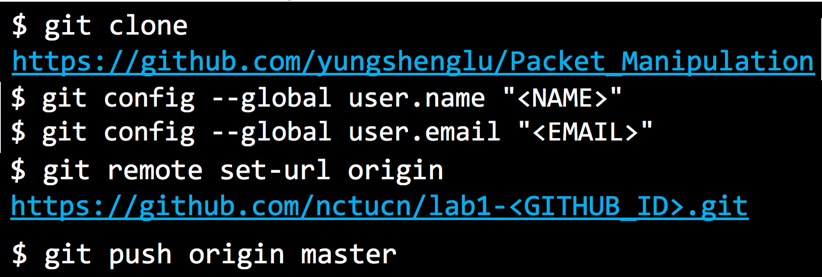
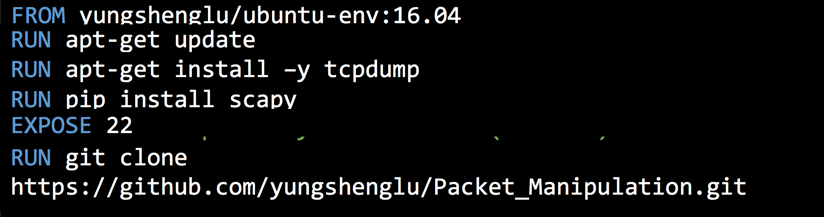
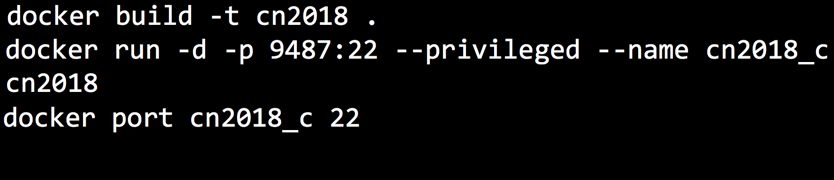
4. Show the screenshot of filtering the packet with “secret” payload .

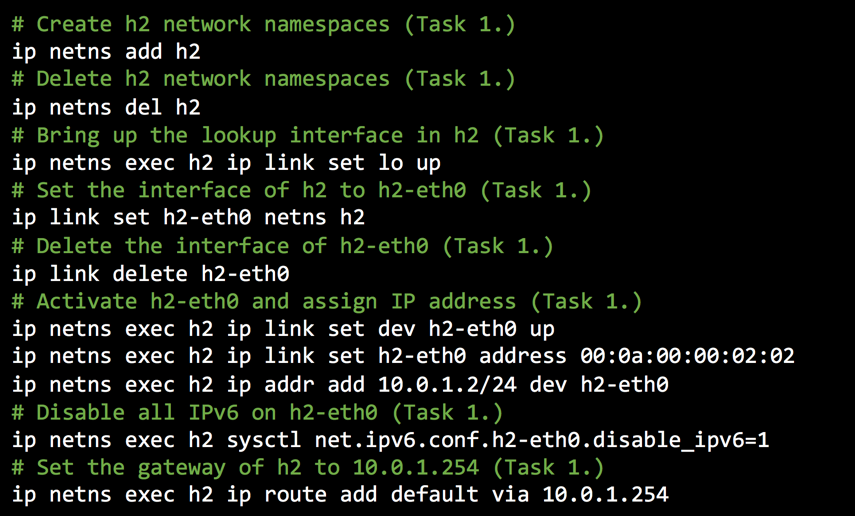


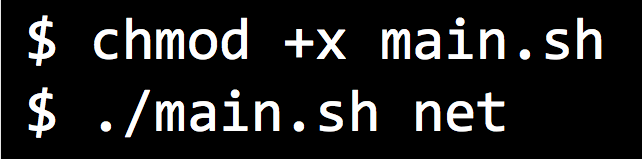
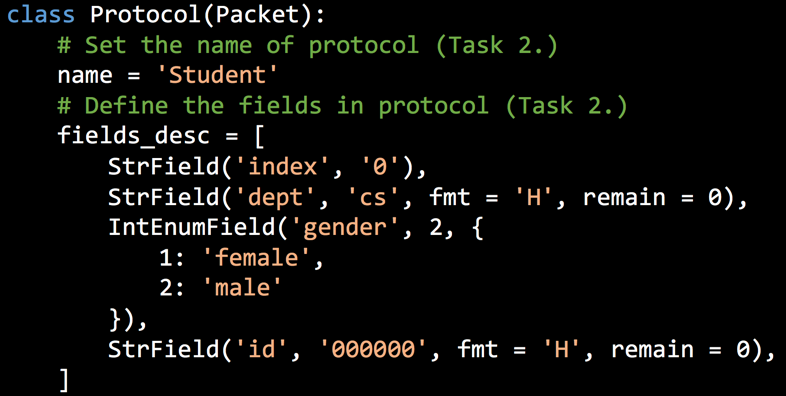
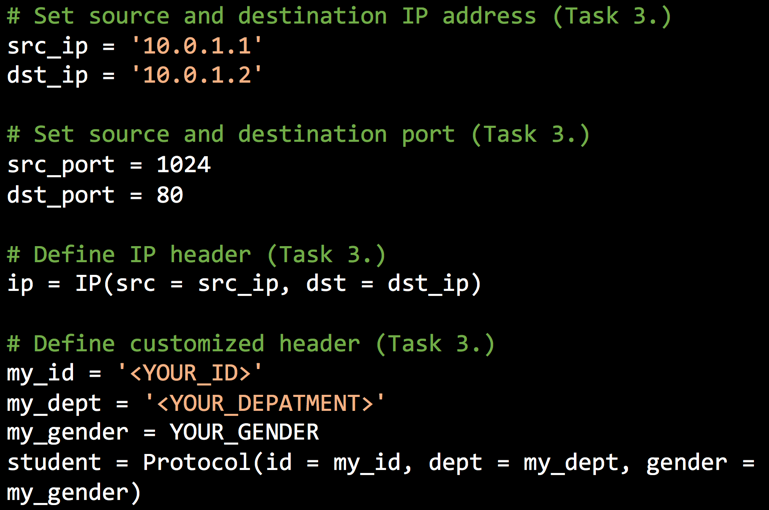
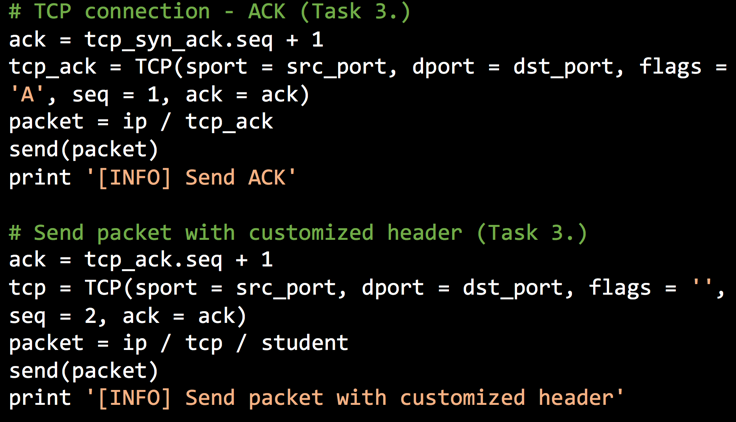
5. Show the result after decoding the “secret” payload.

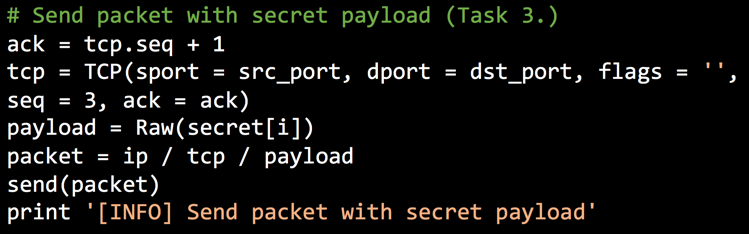
Ans:

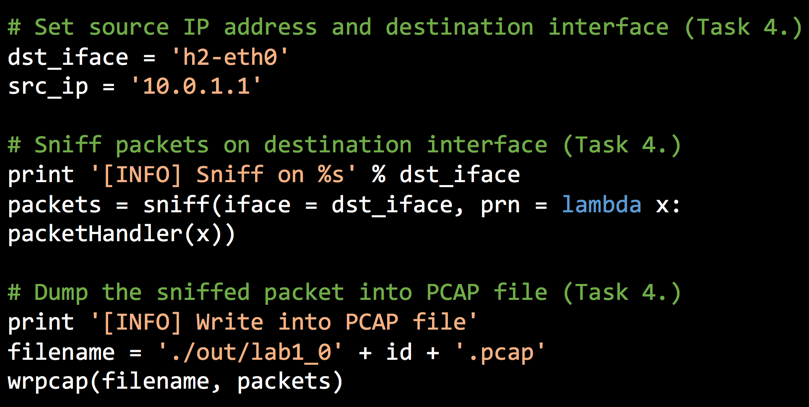
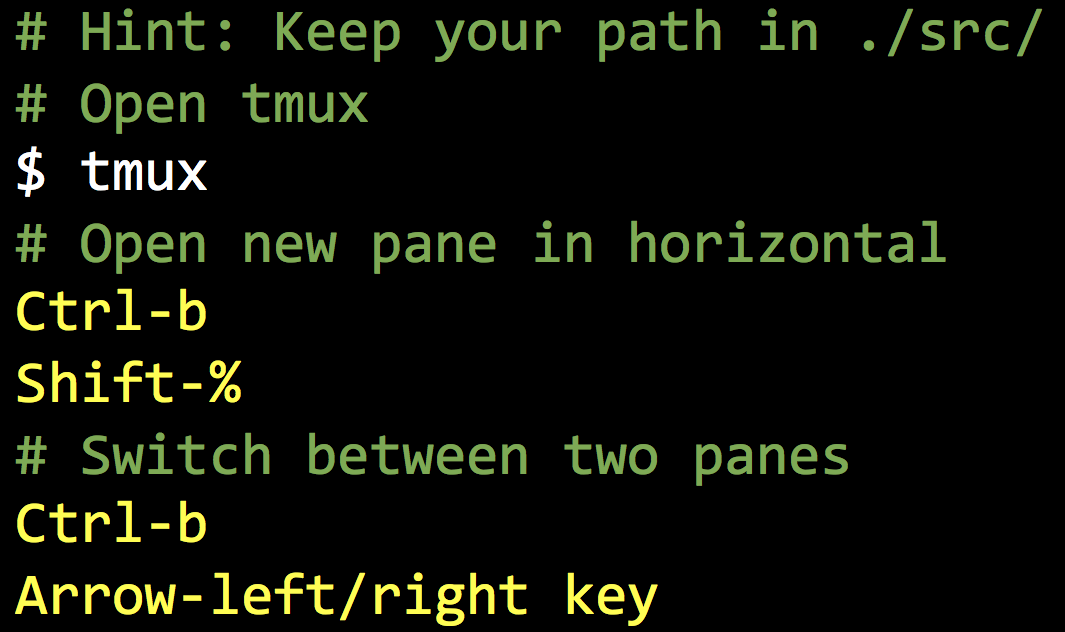
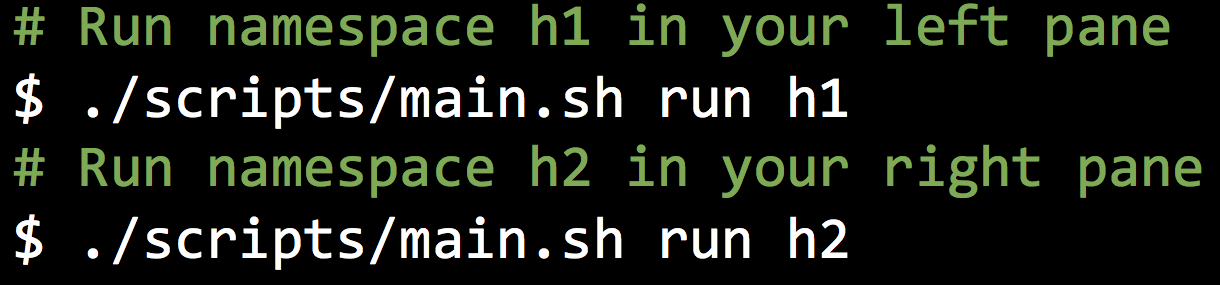
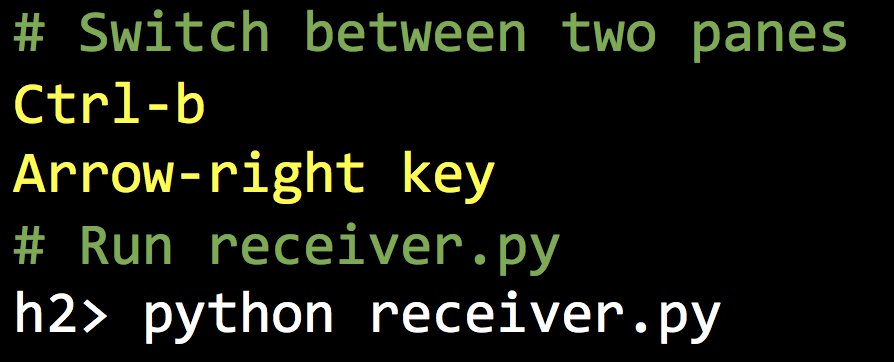
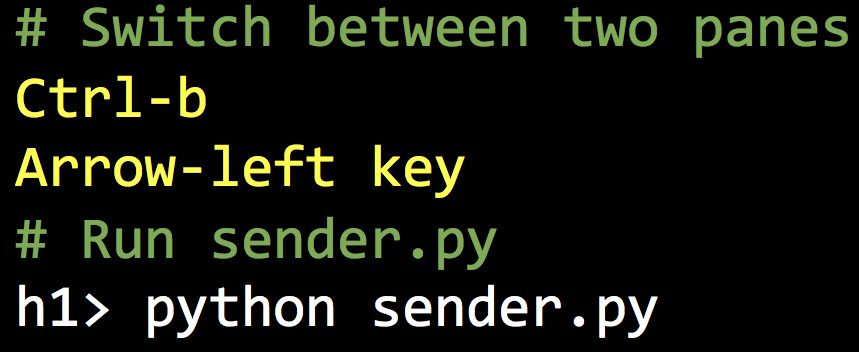
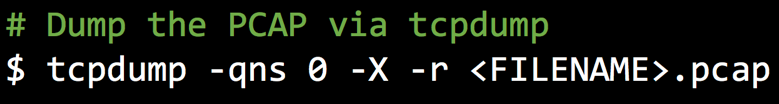
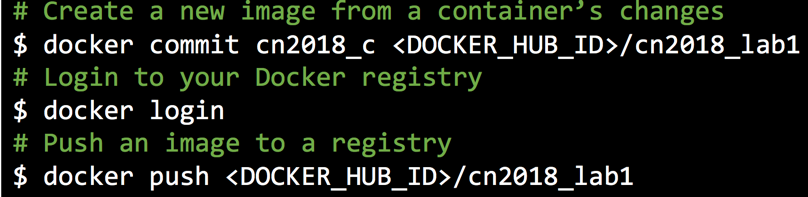
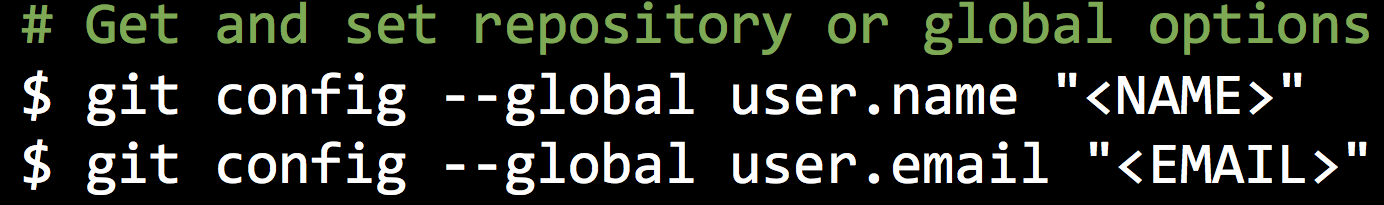
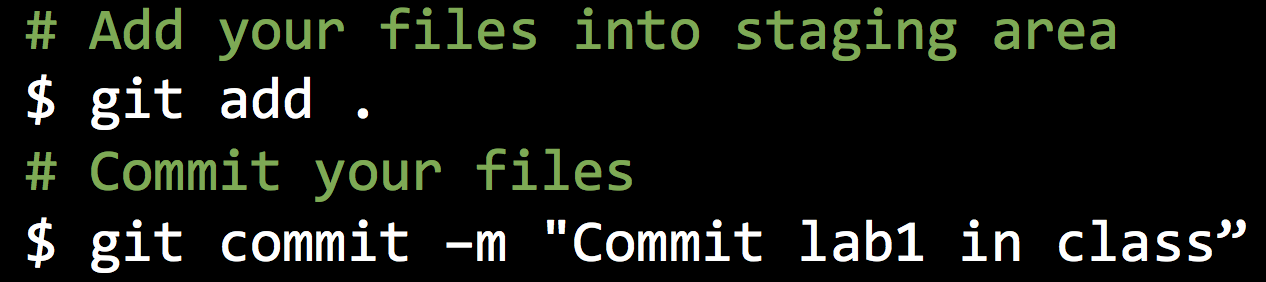
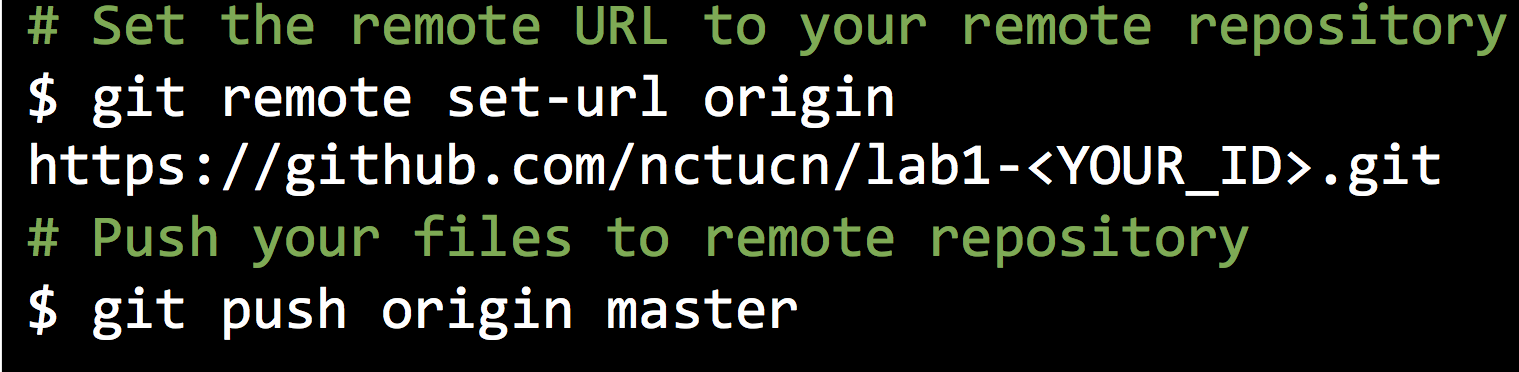
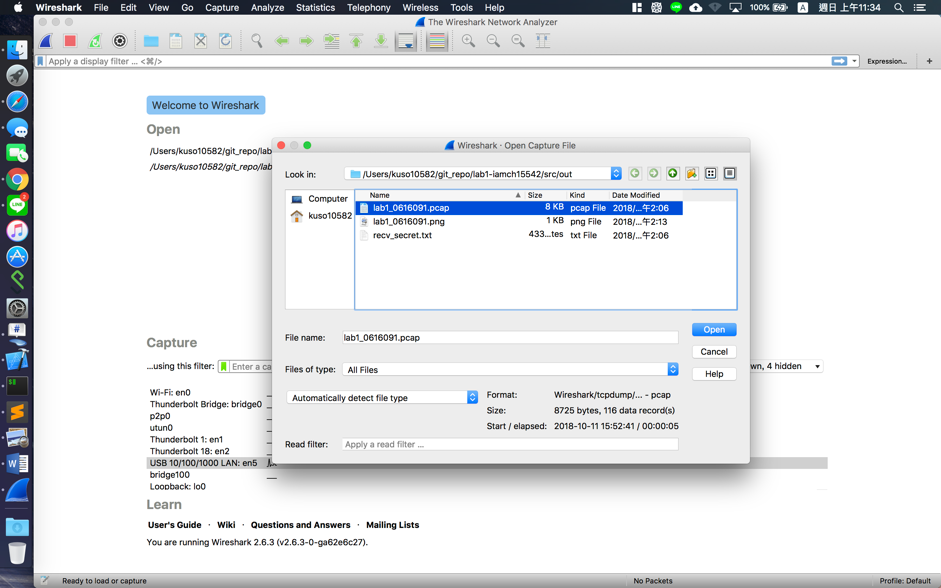
Part B. Description

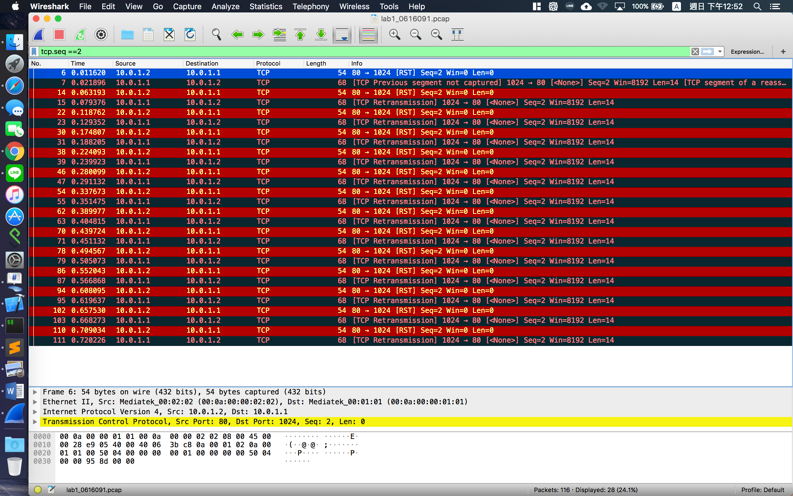
* Task 1 – Environment setup
  1. Setup github and git
     + 依序在終端機輸入，第一句是從github上下載檔案，第二三句則是設定git的username以及email。第三句則是設定遠端伺服器的網址，第五句則是將master這個分支推向origin的位置。
  2. Setup Dockerfile
     + 第一句是從yungshenglu/ubuntu-env:16.04下載基本的image，第二句則是更新所有軟體，第三句則是安裝tcpdump，第四句則是安裝scapy，第五句則是設定container在運作時偵聽的端口，第五句則是從github下載檔案。
  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。
  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。



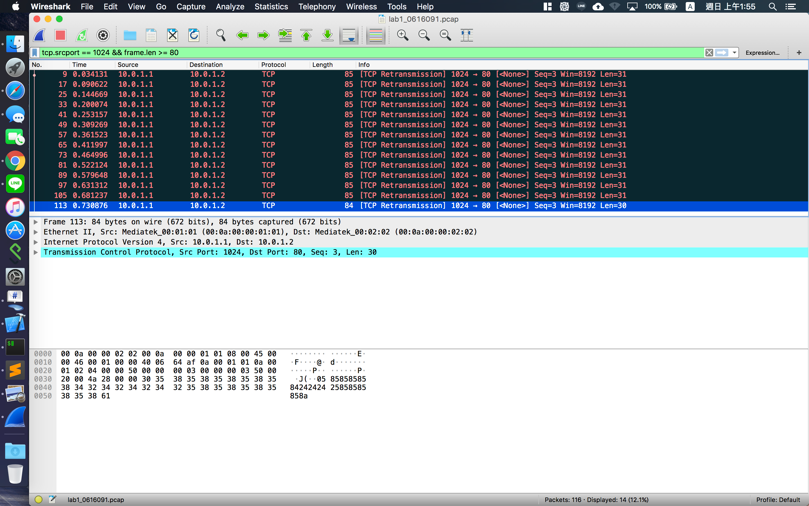
* 1. Run main.sh to build the namespace
     + Chmod +x 是添加執行權限，讓main.sh可以順利執行。./main.sh net則是執行main.sh這個檔案，並且使用net的功能。
* Task 2 – Define protocol via Scapy
  1. Define your protocol: ID header format
* 在Protocol.py裡新增下列的code，code的內容是在設定基本的資訊及定義。
* Task 3 – Send packets
  1. Set your own packet header
     + 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。
  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。

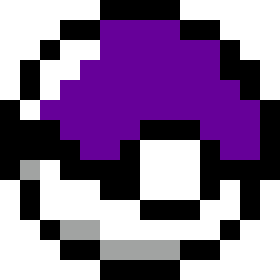


* Task 4 – Sniff packets
  1. Receive and sniff packets
* 在receiver.py裡新增下列的code，前面兩行是先設置來源的ip位置跟目的地。再來則是在目的地的介面print”Sniff on 目的地”，最後先print出”write into PCAP file”後，便將所截取得到資訊寫入pcap檔案。
* Task 5 – Run sender and receiver
  1. Open tmux with horizontal two panes
* 在終端機使用tmux，再用下列的按鍵，將視窗分割成兩個視窗。
  1. Switch 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。
  1. 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。
  1. 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。
  1. 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。
* 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
  1. Push your files to GitHub
* 設定git所需要的名稱及信箱。
* 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。
* 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。
* 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/>
  1. 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
  1. 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



* 1. 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
  1. 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。