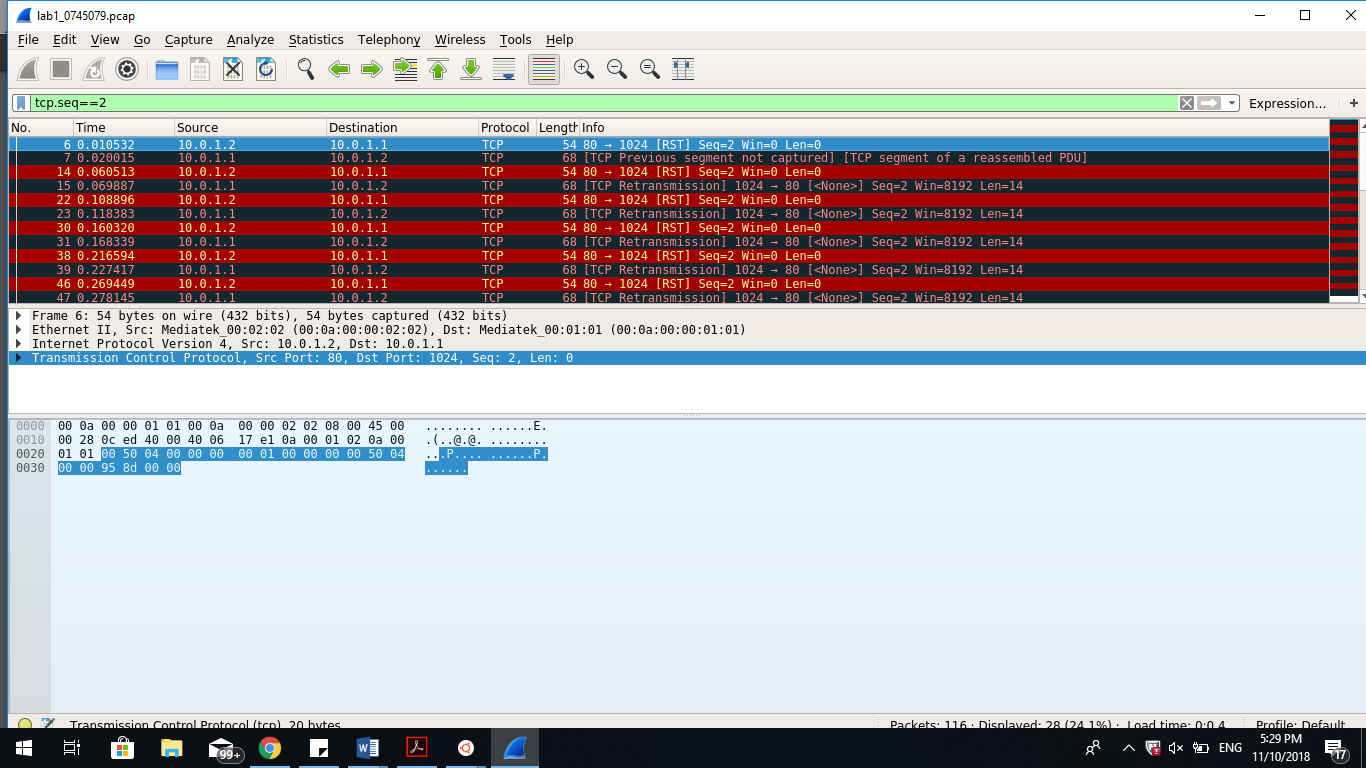
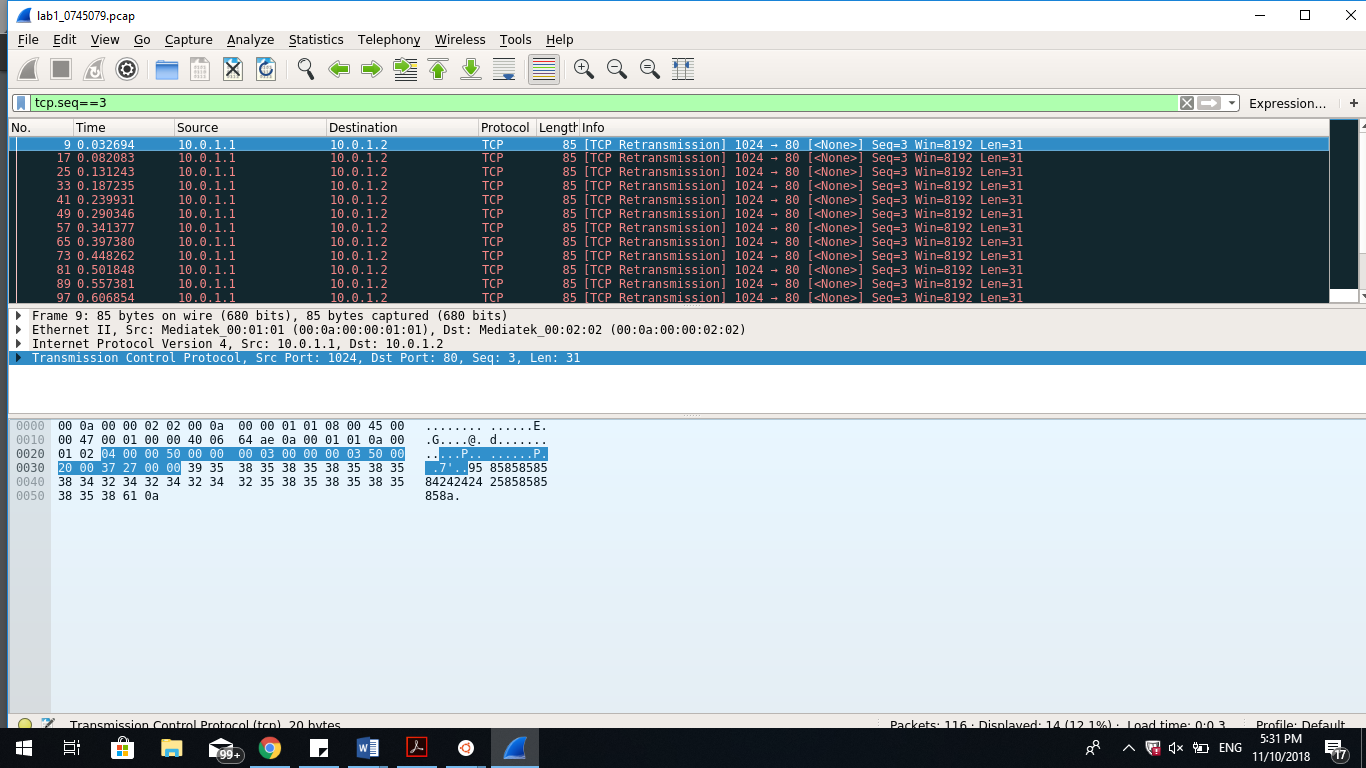
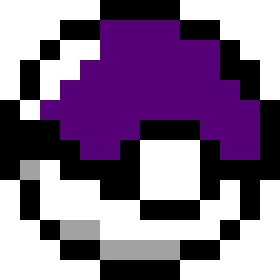
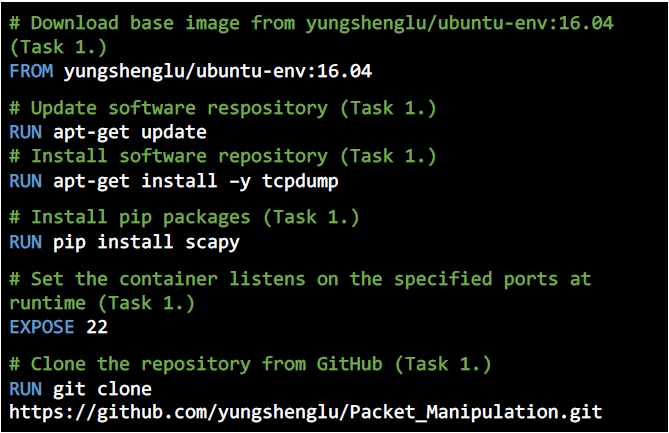
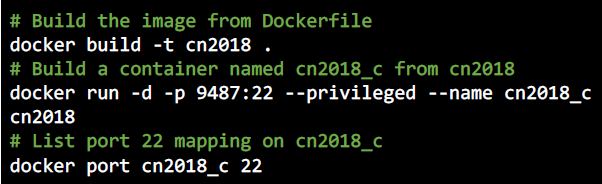
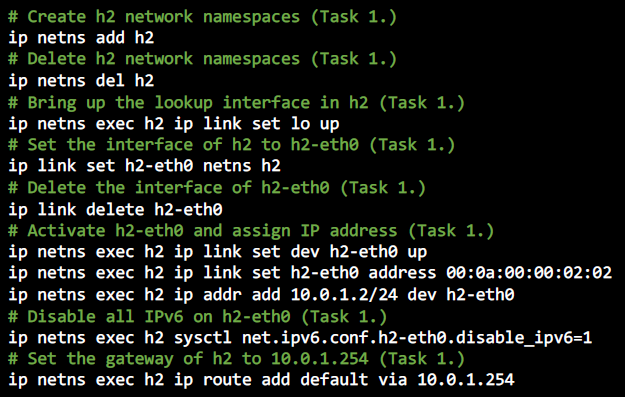
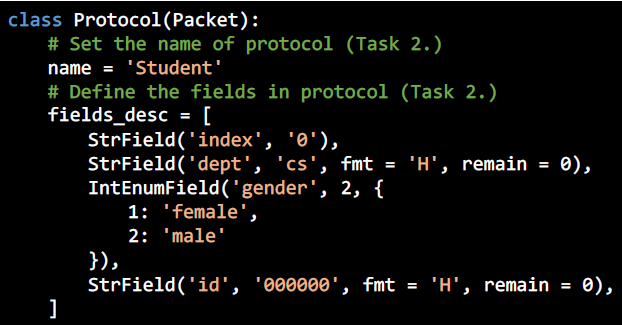
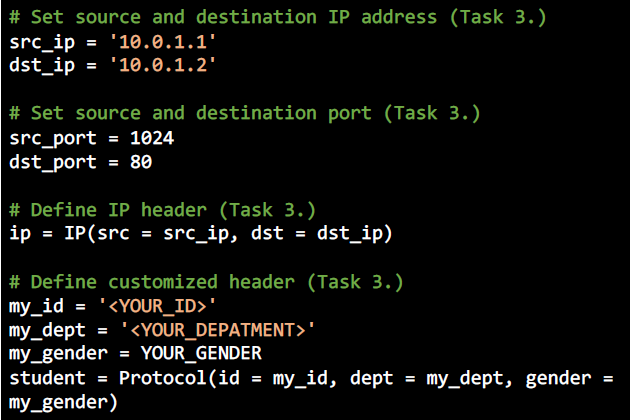
1. tcp.seq == 2
2. 
3. tcp.seq == 3
4. 
5. 

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

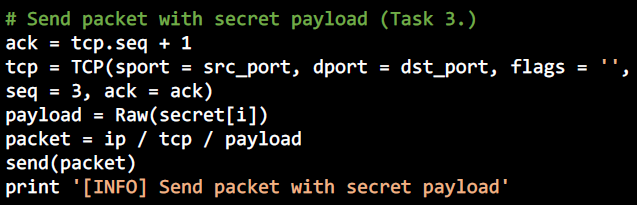
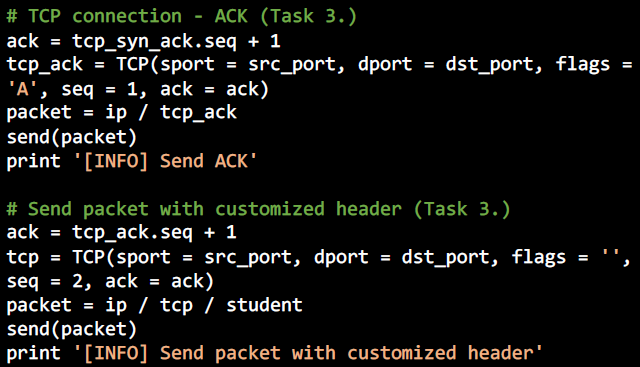
1. Environment setup
   1. Launch any preferred terminal (cmd/powershell/ubuntu/bash). For the sake of this report I will be assuming PowerShell as choice of terminal.
   2. Use “git clone <https://github.com/yungshenglu/Packet_Manipulation>” to clone the repository to local storage.
   3. Use “git config --global user.name ‘<NAME>’ ” and “git config --global user.email ‘<EMAIL>’ ” to set the configuration of your git commits. Fill in preferred name to display in commit and github email.
   4. Set remote URL to your personal repository. In my case, it would be git remote set-url origin <https://github.com/nctucn/lab1-SamsonChoo.git>
   5. Push the repository: git push origin master
   6. Use cd docker to access the docker folder.
   7. Use vim Dockerfile to add the following lines in the Dockerfile through vim (the comments explain quite clearly the function of each line):
   8. Save and exit the Dockerfile to go back to the docker folder. Key in the following commands to build the environment now:

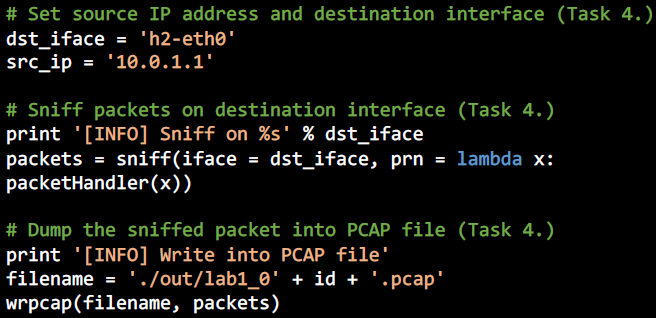
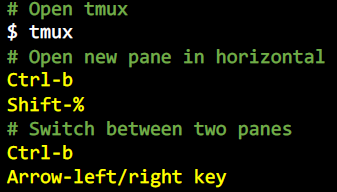
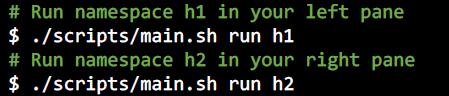
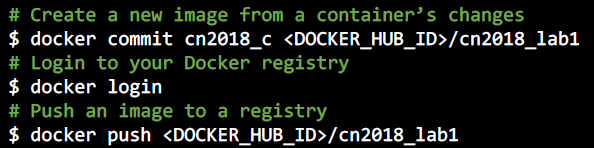


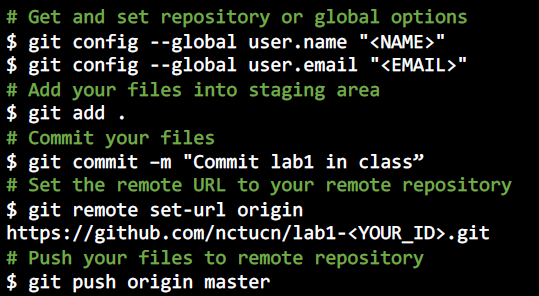
* 1. Open the Kitematic and click [create] for ubuntu-upstart. Get the docker IP address from the screen and connect to the docker through PieTTY to port 9487.
  2. From the container terminal, use “cd”(change directory) to access ./src/scripts and use “vim main.sh” to add the following lines in the file:
  3. Run main.sh to build the namespaces h1 and h2 through the following commands:

1. Define your own protocol
   1. Change directory to src folder and “vim Protocol.py” to define the protocol with the following lines:
2. Send the packets
   1. Add the following lines to src/sender.py through vim again:

In my case, my\_id = “0745079”, my\_dept = “cs”, and my\_gender = 2



1. Receive and sniff the packets
   1. Add the following lines to src/receiver.py:
2. Run sender and receiver
   1. Open tmux in src folder and create a second pane:
   2. Run h1 and h2 in the two panes respectively:
   3. Run “python receiver.py” in one pane to start the receiver and await for the packet, followed by “python sender.py” in the other pane to initiate the sending of packets.
   4. Use “tcpdump -qns 0 -X -r <FILENAME>.pcap” to show the pcap file, in this case, the file is “lab1\_0745079.pcap”.
3. Push the completed files
   1. Push to dockerhub:
   2. Push to github:



1. Load PCAP from Wireshark
   1. Pull the project from github to your own machine by keying “git clone <https://github.com/nctucn/lab1-SamsonChoo.git>” into the terminal
   2. Run VcXsrv, a X server used as a display system for the Wireshark that is already installed in my Ubuntu.
   3. Run “Wireshark” in Ubuntu to launch Wireshark.
   4. Select lab1\_0745079.pcap to open in Wireshark.
2. Filter the target package
   1. From the code given in sender.py, we know that we use “tcp.seq == 2” to get packets with customised header and “tcp.seq == 3” to get packets with the secret payload. Enter the commands above in the DisplayFilters to filter for the wanted packets.
   2. From the 14 secret packets, get the first digit of each packets to form a 14-digit number.
3. Decode the secret key
   1. Open the terminal, go to src folder, and run python3 decoder.py <SECRET KEY>. Use “pip install --user” to install any missing packages.
   2. Check the output image in src/out folder. It should be a purple pokeball.

**Bonus**

* What you have learned in this lab?
  + I have learned how to use Github and Dockerhub, the basic operations such as committing, pulling and pushing. I have learnt to practise using vim as an editor. I have learnt that the ip address 127.0.0.1 refers to the local host. I have learnt how to create and use a container, and what it is. I learnt how to define my own protocols, and how to retrieve the packets I want. I have learnt how to encode and decode a package. I also learnt how to use tmux to create several panels, which is very useful, so that I do not need to open several terminals.
* What difficulty you have met in this lab?
  + The instructions were unclear at first, hence I was stuck many times. For example, the instructions asked us to connect to PieTTY using 127.0.0.1, but that is the local IP instead of the container’s address. Another problem I faced was that I edited all my files on my local machine, but I forgot that it is separate from the container and that I should have edited the files on the container terminal instead. I also took a long time to find out how to filter the packets that I want from, but I managed to find out after reading through the sender code carefully. I also was not able to decode my file initially as the code would not run properly. This issue was resolved by installing the relevant packages and updating my python version, as some of the modules were only available in the latest python version, such as the clear attribute from list.