

**Part A. Questions**

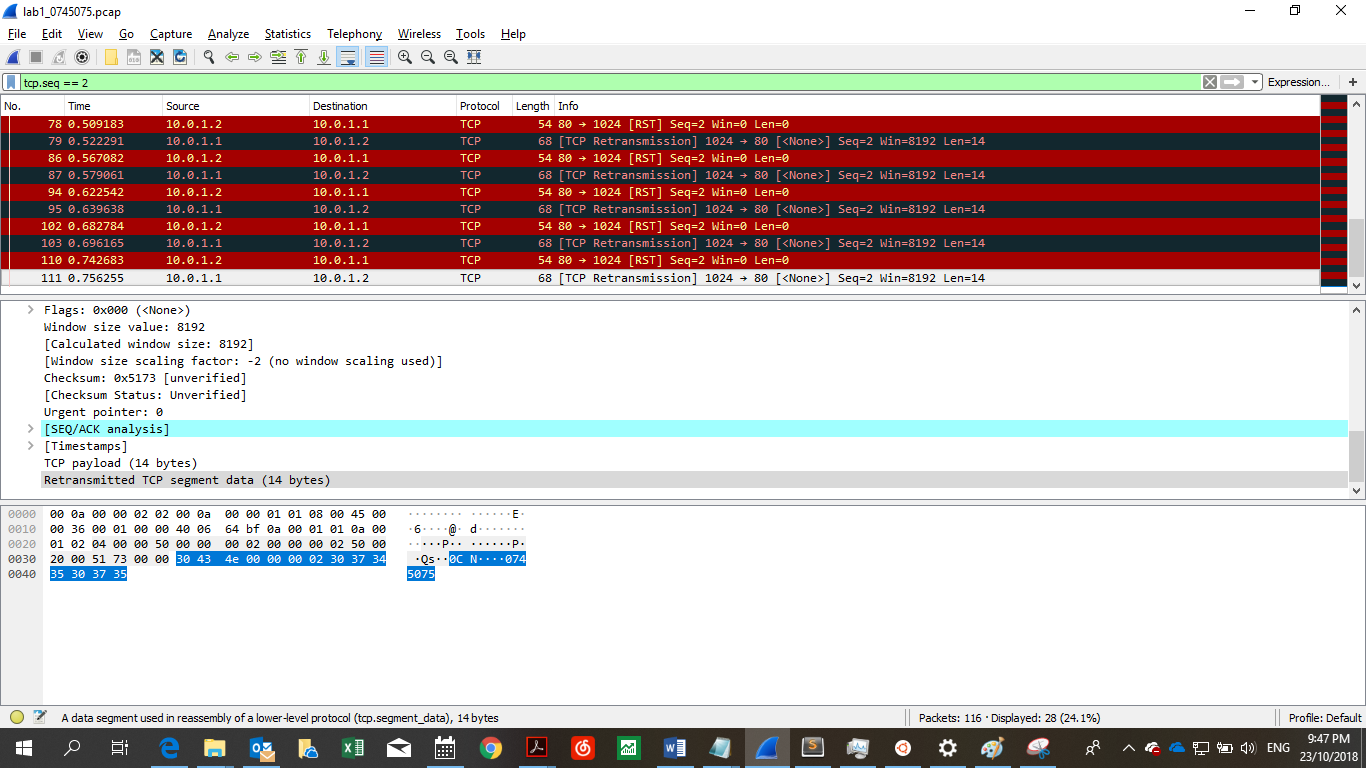
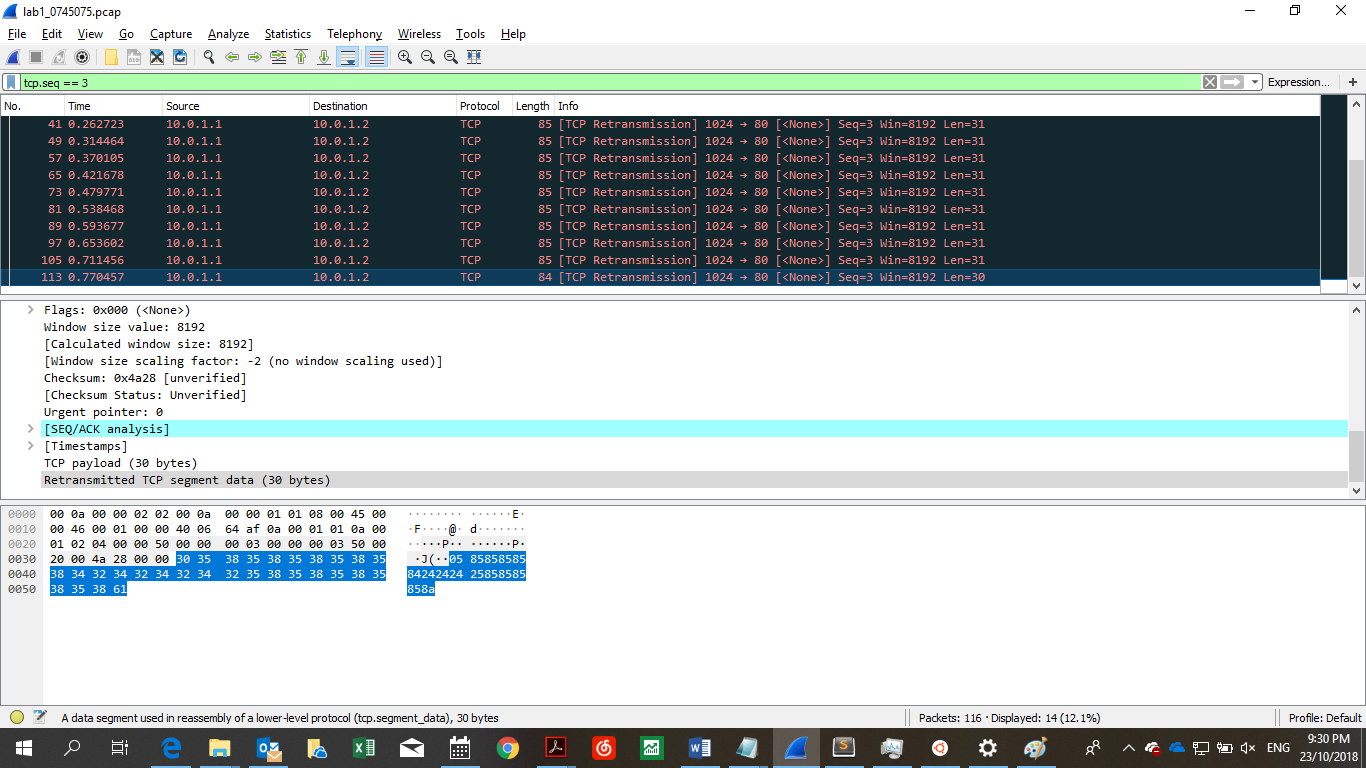
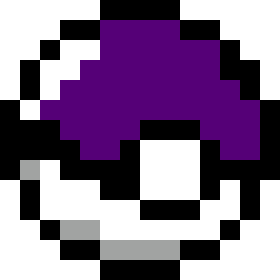
1. What is your command to filter the packet with customized header on Wireshark?

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?

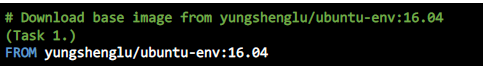
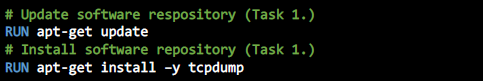
4. Show the screenshot of filtering the packet with “secret” payload.

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

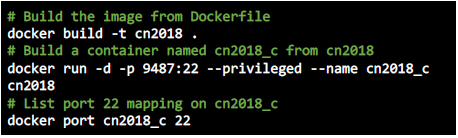
1. tcp.seq == 2
2. 
3. tcp.seq == 3
4. 
5. 

**Part B. Description**

Task 1 – Environment setup

* 1. • Configure Dockerfile
  2. o How to configure Dockerfile? https://docs.docker.com/engine/reference/builder/
  3. o First, download base image from yungshenglu/ubuntu-env:16.04 
  4. o Second, update software repositories. 
  5. O Third, install pip packages



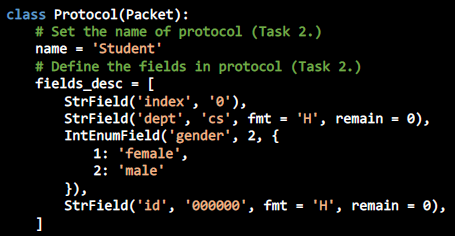
* 1. o Fourth, set the container listens on the specified ports at runtime. 
  2. O Fifth, use argument to assign passwd to create image.
  3. o Sixth, SSH login fix. Otherwise user is kicked off after login.
  4. o Seventh, set the envionment variables.
  5. o Eighth, set the container listens on the specified ports at runtime (Task 1.)
  6. o Ninth, set the entry point.
  7. o Twentieth, clone the repository from GitHub. 
  8. o Open the CMD (administration) and change the path to ./docker/ and build the environment as follows:
  9. 
  10. • Build Environment
  11. o Build the image from Dockerfile
  12. o Build a container named cn2018\_c from cn2018
  13. o List port 22 mapping on cn2018\_c
  14. • Login to Docker container using SSH
  15. o Open the PieTTY and connect to the Docker.
  16. o Create the namespace in ./src/scripts/main.sh for h2
  17. o Run main.sh to build the namespace

Task 2 - Define protocol via Scapy

• Define your protocol: ID header format

o Set the name of protocol

o Define the fields in protocol



Task 3 – Send packets

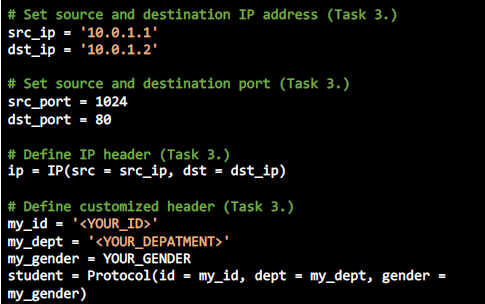
• Set your own packet header in ./src/sender.py

o Set cource and destination IP address.

o Set source and destination port.

o Define IP header.

o Define customised header.

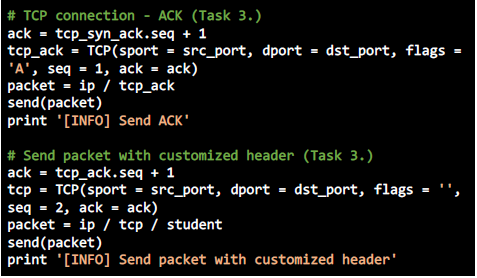
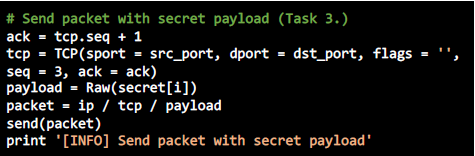


• Send packets:

o TCP connection – ACK.

o Send packer with customised header.

o Send packer with payload.

Add the codes below in ./src/sender.py:  

Task 4 – Sniff packets

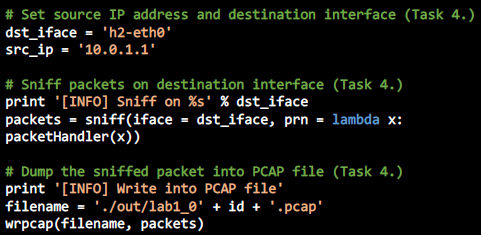
• Receive and sniff packets:

o Set source IP address and destination interface.

o Sniff packets on destination interface.

o Dump the sniffed packet into PCAP file.

Add the codes below in ./src/receiver.py



Task 5. Run sender and receiver

• Open tmux with horizontal two panes

• Switch into two namespaces

• Run receiver.py first

• Run sender.py

• Use tcpdump to show PCAP file

Task 6. Push your files to remote

• Push your files to GitHub

Task 7. Load the packet trace via Wireshark

• Open the PCAP file using Wireshark

Task 8. Filter the target packets

• Filter the packets of our defined protocol

• Enter your filter command on DisplayFilters

• Filter the packets with the “secret” bits

• Enter your filter command on DisplayFilters

• Find out the first digit of the “secret” payload in these packets and combine them as a 14-digit “secret” key

Task 9. Decode the secret key

• Input the secret key into ./src/decoder.py

• Execute decoder.py

Task 10. Report

• The output file is in ./src/out/

• You will get an image related to Pokemon if succeed

• Bonus

1. What you have learned in this lab?

2. What difficulty you have met in this lab?

1. I have learnt about how packet sniffing works. It is interesting how we are using TCP connection and our very own protocol to send packets of data. I also tried my hand at using wireshark to filter the received data into message.
2. I am unfamiliar with GitHub, therefore, the files seem unable to commit.