NCTU CN2018 Lab. 1 – Packet Manipulation via Scapy

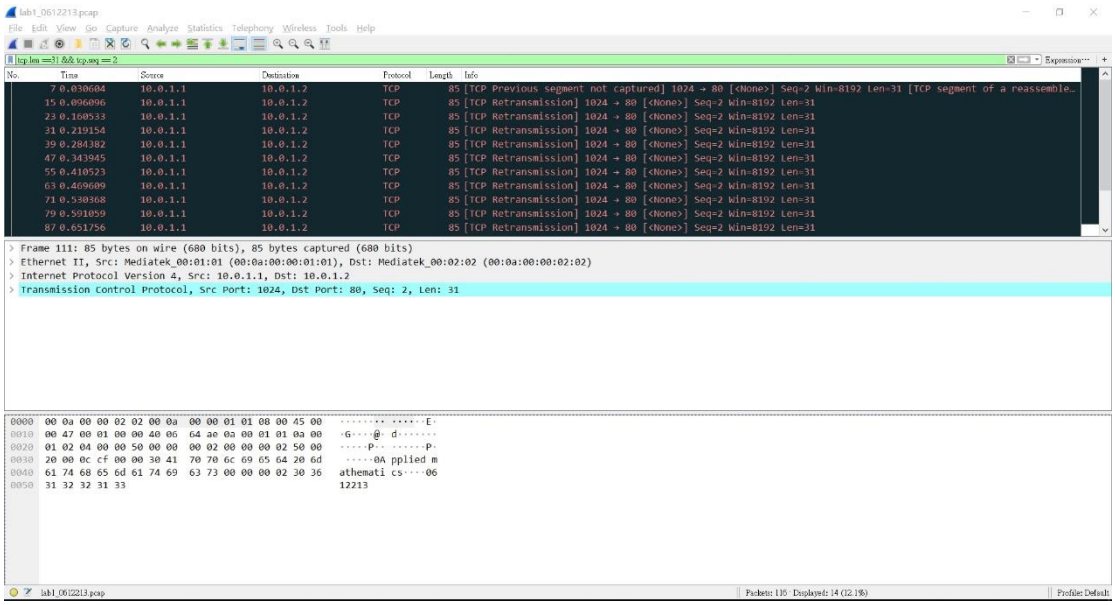
Student name:曾振豪 Student ID: 0612213 Department: Applied Mathematics

Part A. Questions

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

   tcp.len ==31 && tcp.seq == 2

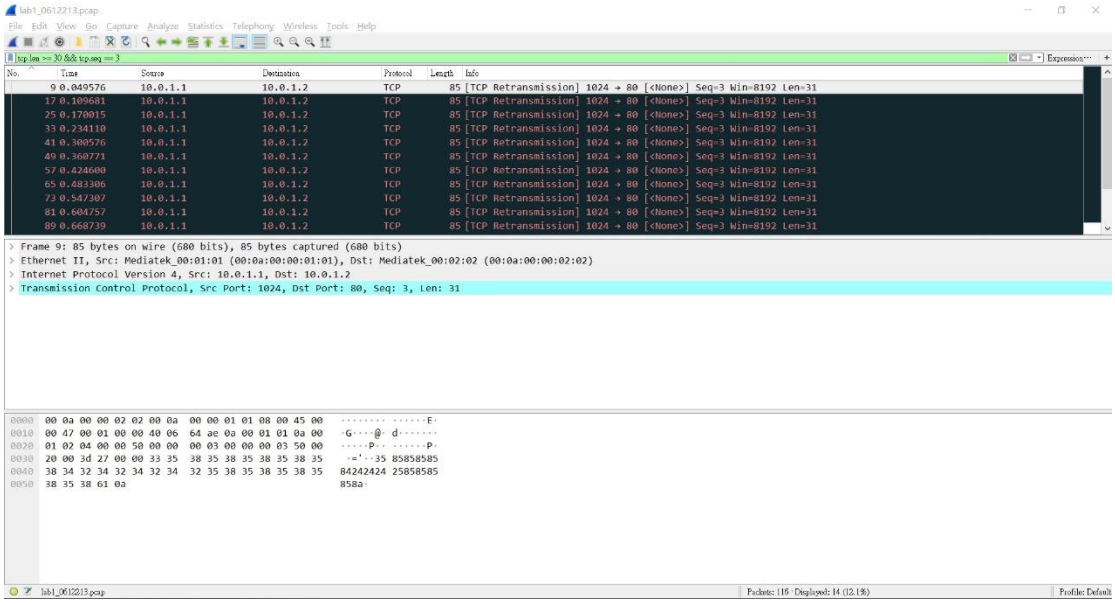
2. Show the screenshot of filtering the packet with customized header.



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

   tcp.len >=30 && tcp.seq == 3

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



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

Part B. Description

Task 1. Enviroment setup

- Download required files from GitHub:

  Open “Git Bash” and type “git clone https://github.com/yungshenglu/Packet_Manipulation”

- Get and set repository or global options:

  Type “git config --global user.name "howhowcan"

  git config --global user.email "gktt58@gmail.com"”

- Set a new remote URL to your repository:

  Type “git remote set-url origin https://github.com/nctucn/lab1-howohwcan.git”

- Configure Dockerfile:

  Copy the following configuration to the Dockerfile (./docker/Dockerfile):

  ```
  # Download base image from yungshenglu/ubuntu-env:16.04
  FROM yungshenglu/Ubuntu-env:16.0
  # Update software repository
  RUN apt-get update
  # Install software repository
  RUN apt-get install tcpdump
  # Install packages
  RUN pip install scapy
  # Set the container listens on the specified ports at runtime
  EXPOSE 22
  # Clone the repository from GitHub
  RUN git clone https://github.com/nctucn/lab1-howhowcan.git
  ```

• Open the CMD and change the path to ./docker/ and build the environment as follows:

```
# Build the image from Dockerfile
docker build -t cn2018 .
# Build a container named cn2018_c from cn2018
docker run -d -p 9487:22 --privileged --name cn2018_c cn2018
# List port 22 mapping on cn2018_c
docker port cn2018_c 22
```

• Login to your Docker container using SSH:

Open the PieTTY and connect to the Docker

IP address: The remote IP Port: 9487

Login as root

Login: root Password: cn2018

• Create the namespace in ./src/scripts/main.sh for h2:

```
# 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.
```

- Run main.sh to build the namespace:

  Type the following command

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

Task 2. Define protocol via Scapy

- Define the protocol: Define ID header format in ./src/Protocol.py:

```
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

- Setup the packet header in ./src/sender.py:

```
    # 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.)
    # Hint: Remember to replace the information with yours
```

```
student = Protocol(id = 'YOUR_ID', dept = 'YOUR_DEPT', gender =
YOUR_GENDER)
```

- Send packets:

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

```
# 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

- Receive and sniff packets:

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

```
# 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

- Open tmux with horizontal two panes:

  Type the command “tmux” and use “Shift-%” to open two panes and press “Ctrl-b”,so that we can use “Arrow-left/right key” to switch between two panes

- Switch into two namespaces:

  # Run namespace h1 in your left pane

  ./scripts/main.sh run h1

  # Run namespace h2 in your right pane

  ./scripts/main.sh run h2

- Run receiver.py:

  Press “Ctrl-b” “Arrow-right key” to switch into right pane

  Run receiver.py

  h2> python receiver.py

- Run sender.py:

  Press “Ctrl-b” “Arrow-left key” to switch into left pane

  Run sender.py

  h1> python sender.py

- Get lab1_0612213.pcap and recv_secret.txt in ./src/out/

Task 6. Push your files to remote

- Push your image to Docker Hub:

  Type the following command:

  # Create a new image from a container’s changes

  docker commit cn2018_c howhow8765/cn2018_lab1

  # Login to your Docker registry

  docker login

  # Push an image to a registry

docker push howhow8765/cn2018_lab1

- Push your files to GitHub:

  Type the following command:

  # Add your files into staging area

  git add .

  # Commit your files

  git commit –m "Commit lab1 in class"

  # Push your files to remote repository

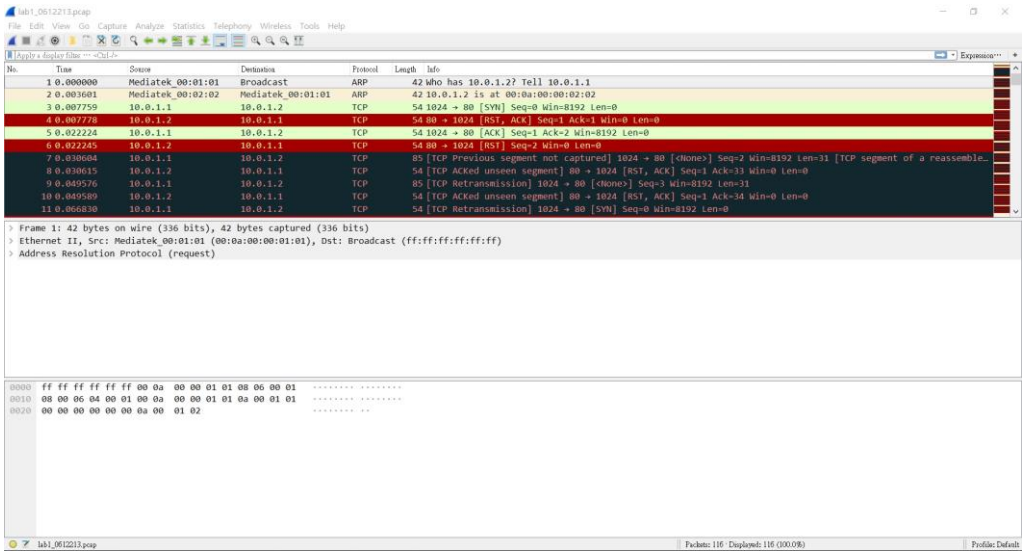
  git push origin master

Task 7. Load PCAP via Wireshark

- Download the code from GitHub:

  Open “Git Bash” and type “git clone https://github.com/nctucn/lab1-howhowcan.git”

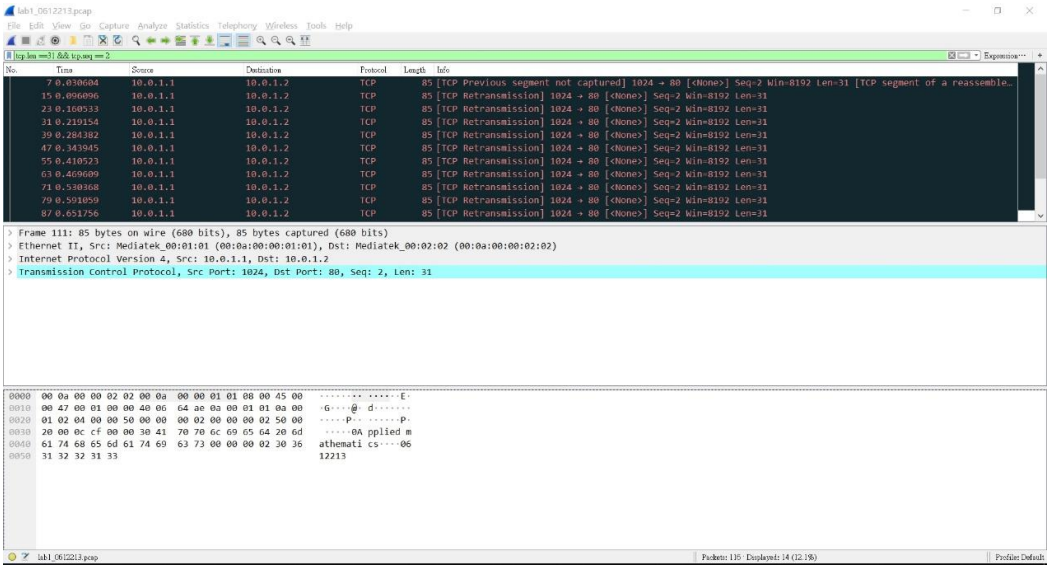
- Open the PCAP file using Wireshark



Task 8. Filter the target packet

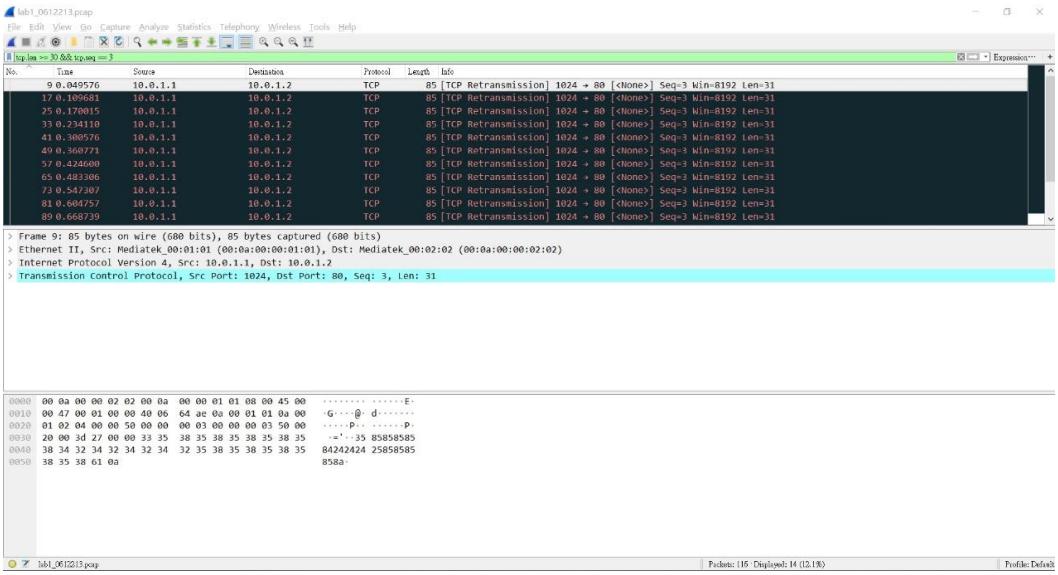
- Filter the packets of our defined protocol

  Filter Rule: tcp.len ==31 && tcp.seq == 2



- Filter the packets with the “secret” bits

  Filter Rule: tcp.len >=30 && tcp.seq == 3



- Combine first digits in every secret payload to get the secret key

  My secret key is 31221603122160

Task 9. Decode the secret key

- Input the secret key into ./src/decoder.py on local machine:

  Open CMD and change the path to ./src/ and type

  python decoder.py 31221603122160

• Check lab1_0612213.png in ./src/out/ :

Part C. Bonus

• What you have learned in this lab?

The concepts of Scapy, docker, git, CMD and Wireshark and how to use them

• What difficulty you have met in this lab?

1. I don’t know the concepts of the tools used in the lab so that I don’t know some command lines should be entered in git bash, CMD, or the remote.

2. I am not familiar with git so I took a long time to deal with the problem of pushing my local files to my github.

Thanks for TAs’ hard work :)