NCTUCN LAB1-Packet Manipulation via Scapy

Student name: Chen Liu Student ID: 0410119 Department: EE

Part A. Questions

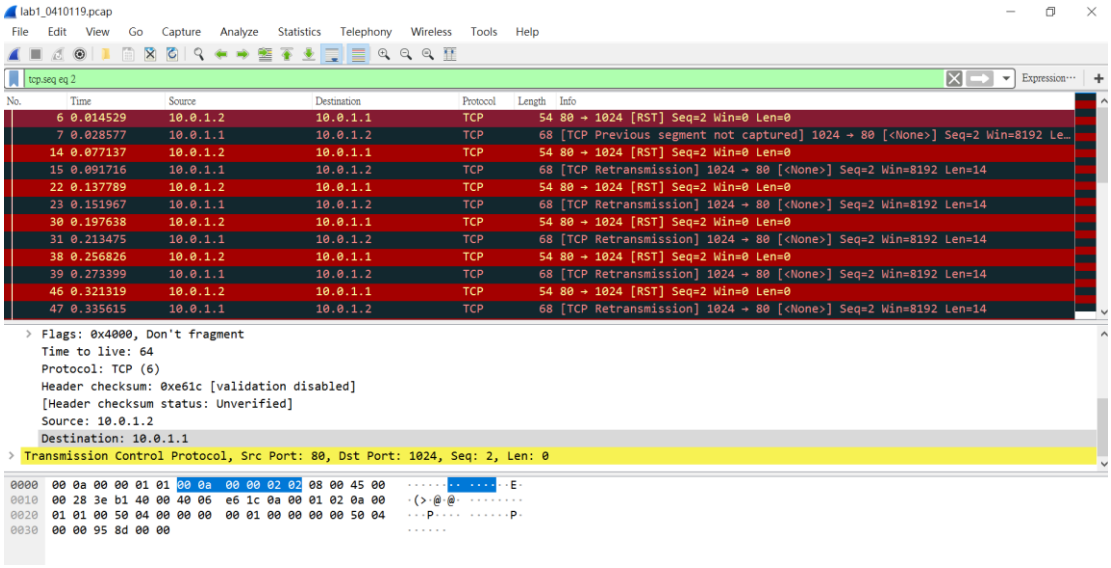
1. What is your command to filter the packet with customized header on Wireshark?

   tcp.seq eq 2

   In sender.py, when send the custimized header, tcp.seq is set 2. There, we filter the packets with tcp.seq==2

```
tcp = TCP(sport = src_port, dport = dst_port, flags = '',
seq = 2, ack = ack)
packet = ip / tcp / student
```

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



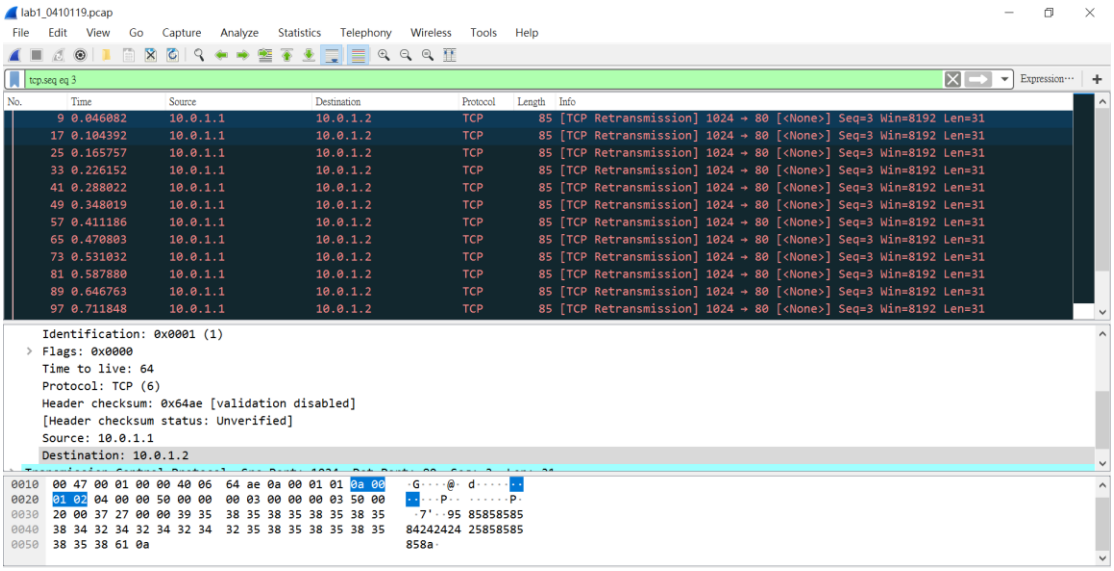
3. What is your command to filter the packet with “secret” payload on Wireshark?

   tcp.seq eq 3

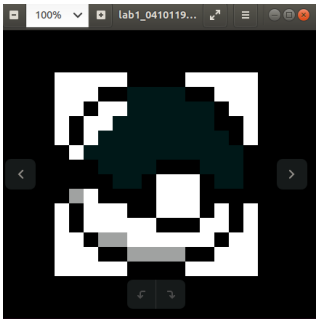
   In sender.py, when send the custimized header, tcp.seq is set 2. There, we filter the packets with tcp.seq==2

```
tcp = TCP(sport = src_port, dport = dst_port, flags = '',
seq = 3, ack = ack)
payload = Raw(secret[i])
packet = ip / tcp / payload
```

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 Environment setup

A. Download and upload repository

Reference: https://git-scm.com/docs

✧ Clone the repository of the URL into a newly directory

```
$ git clone
https://github.com/yungshenglu/Packet_Manipulation
```

✧ Setup your name and email

```
$ git config --global user.name "<NAME>"
$ git config --global user.email "<EMAIL>"
```

✧ changes an existing remote repository URL

```
$ git remote set-url origin
https://github.com/nctucn/lab1-<GITHUB_ID>.git
```

✧ upload the repository to the URL we have set

```
$ git push origin master
```

B. Configure Dockerfile

Reference: https://docs.docker.com/engine/reference/builder/

- ✧ Download base image from yungshenglu/ubuntu-env:16.04

```
FROM yungshenglu/Ubuntu-env:16.0
```

- ✧ Update software repository

```
RUN apt-get update
```

- ✧ Install tcpdump which is a packet sniffer

```
RUN apt-get install tcpdump
```

- ✧ Install Scapy which is an interactive packet manipulation

```
RUN pip install scapy
```

- ✧ informs Docker that the container listens on the specified network ports at runtime

```
EXPOSE 22
```

- ✧ clone the repository from what we have uploaded in Task 1, A

```
RUN git clone https://github.com/nctucn/lab1-xxx.git
```

C. Build container and ssh the container

- ✧ Change main.sh to be executable than build the container

```
$ sudo chmod +x main.sh
$ ./main.sh build cn2018 9487
```

- ✧ Login the container using SSH

```
$ ssh root@0.0.0.0 -p 9487
Password: cn2018
```

D. Create network namespace

Reference: https://blog.scottlowe.org/2013/09/04/introducing-linux-network-namespaces/

- ✧ Filled in the command for h2 into the corresponding function to complete the main.sh

```
# 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
```

- ✧ Change main.sh to be executable than run function called net in main.sh to build namespace

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

Task 2 Define protocol via Scapy

A. Define ID header format

- ✧ Set the name of protocol

```
name = 'Student'
```

- ✧ Define the fields in protocol

```
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 - Modify sender.py

A. Set source and destination

```
# 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)
```

B. Setup customized packet header

```
# Define customized header (Task 3.)
# Hint: Remember to replace the information with yours
student = Protocol(id = 'YOUR_ID', dept = 'YOUR_DEPT',
gender = YOUR_GENDER)
```

C. add the codes of first packet to make sure the connection

Reference: https://baike.baidu.com/item/ACK

```
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'
```

D. add the codes of second packet with customized header

```
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'
```

E. add the codes of third packet with secret payload

```
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 – Modify receiver.py

A. Set source IP address and destination interface

```
dst_iface = 'h2-eth0'
src_ip = '10.0.1.1'
```

B. Sniff packets on destination interface

```
print '[INFO] Sniff on %s' % dst_iface
packets = sniff(iface = dst_iface, prn = lambda x:
packetHandler(x))
```

C. Dump the sniffed packet into PCAP file

```
print '[INFO] Write into PCAP file'
filename = './out/lab1_0' + id + '.pcap'
wrpcap(filename, packets)
```

Task 5 Run sender and receiver

A. Split the terminal into two panes using tmux

Reference: https://blog.chh.tw/posts/tmux-terminal-multiplexer/

```
# Open tmux
$ tmux
# Open new pane in horizontal
Ctrl-b
Shift-%
# Switch between two panes
Ctrl-b
Arrow-left/right key
```

B. Run namespaces h1 and h2 in different pane

```
# Run namespace h1 in your left pane
$ ./scripts/main.sh run h1
# Run namespace h2 in your right pane
$ ./scripts/main.sh run h2
```

C. Run receiver.py **first**, and then run sender.py(Make sure that receiver is ready before sender sends packets)

```
# Switch between two panes
Ctrl-b
Arrow-right key
# Run receiver.py
h2> python receiver.py
```

```
# Switch between two panes
Ctrl-b
Arrow-left key
# Run sender.py
h1> python sender.py
```

after receiving all packets, you receive your PCAP file and recv_secret.txt

Task 6 Push your files to remote

A. Push your image to Docker Hub

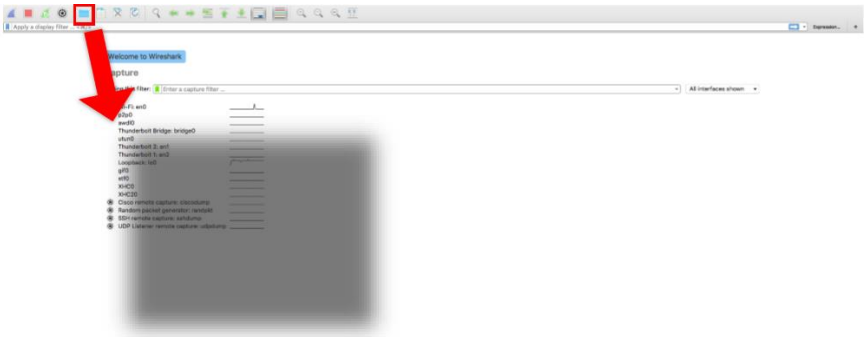
```
# Create a new image from a container's changes
$ docker commit cn2018_c <DOCKER_HUB_ID>/cn2018_lab1
# Login to your Docker registry
$ docker login
# Push an image to a registry
$ docker push <DOCKER_HUB_ID>/cn2018_lab1
```

B. Push your files to Github and leave message saying your progress

```
# 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

A. Open the PCAP file using Wireshark



Task 8 Filter the target packet

A. Filter the packets

reference: https://wiki.wireshark.org/DisplayFilters

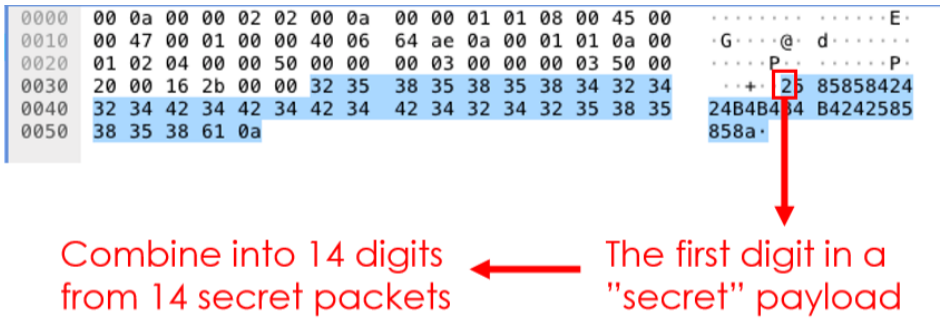
- ✧ Enter command to filter the customized protocol

  >> tcp.seq eq 2

- ✧ Enter command to filter the packets with secret bits

  >> tcp.seq eq 3

B. Getting the first digit in a secret packet



Task 9 Decode the secret key

A. Decode secret key with decoder.py

```
$ python decoder.py <YOUR_SECRET_KEY>
```

B. Open the output file lab1_<student ID>.png, and you will get an image of Pokemon ball if succeed

Task 10 upload the repository

A. Enter the command like Task 6, B to submit those update files

Part C Bonus

1. What you have learned in this lab?

   I learned some basic commands of Git and how to build the container. Also, I knew how the packets sent and received using Scapy, and how the Wireshark filter the packets we want.

2. What difficulty you have met in this lab?

   The main difficulty is that I don't know in which software(Git Bash, Pietty…) I should enter those command. This is because I do not really understand the functionalities of each step. For example, when you know the space in the container is an independent and virtual space, you won't enter git command in local machine and upload the files which aren't modified.