

# LAB PROJECT 1: Packet Sniffing and Spoofing Lab

Kushal Hebbar | C13031425 | khebbar@g.clemson.edu

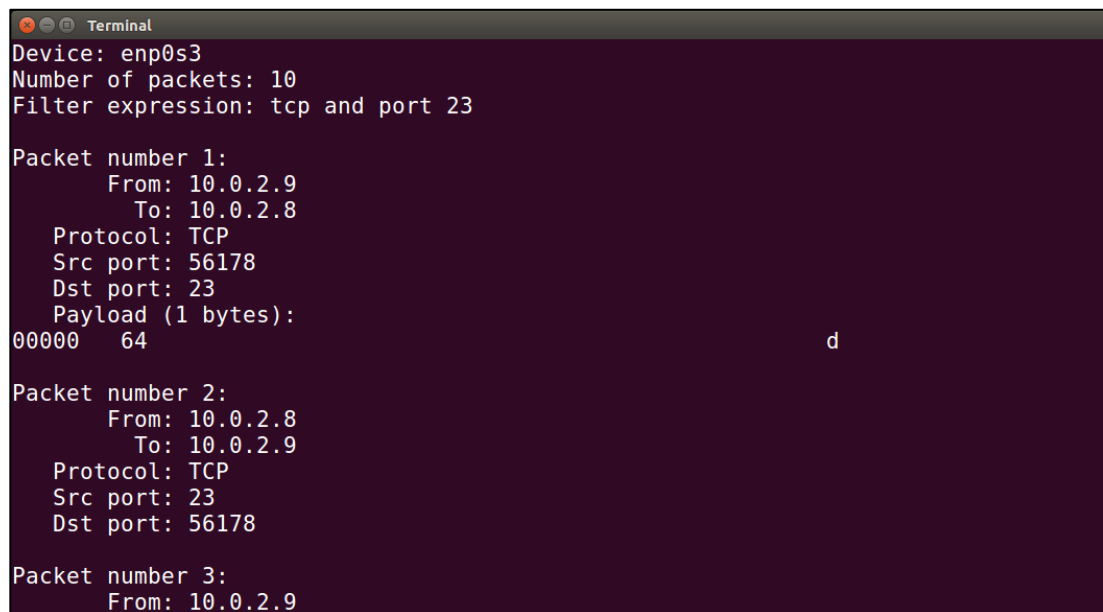
- **Problem 5:** Please show how you can use sniffex to capture the password when somebody is using telnet on the network that you are monitoring. You may need to modify the sniffex.c a little bit if needed. You also need to start the telnetd server on your VM. If you are using our pre-built VM, the telnetd server is already installed; just type the following command to start it.

% sudo service openbsd-inetd start

Considering we are sniffing telnet passwords, we use the port 23 which is the default telnet port. The filter expression in the sniffer program is changed as follows:

```
char filter_exp[] = "tcp and port 23";
```

The Sniffer program is executed in VM1 with IP: 10.0.2.8 and telnet server is started in VM2 with IP: 10.0.2.9. The password for the system is then entered in VM2 which is sniffed and displayed in VM1 in the payload section as shown in the figures below,



```
Terminal
Device: enp0s3
Number of packets: 10
Filter expression: tcp and port 23

Packet number 1:
  From: 10.0.2.9
  To: 10.0.2.8
  Protocol: TCP
  Src port: 56178
  Dst port: 23
  Payload (1 bytes):
00000  64                                     d

Packet number 2:
  From: 10.0.2.8
  To: 10.0.2.9
  Protocol: TCP
  Src port: 23
  Dst port: 56178

Packet number 3:
  From: 10.0.2.9
```

```
Terminal
  To: 10.0.2.8
  Protocol: TCP
  Src port: 56178
  Dst port: 23
  Payload (1 bytes):
000000  65                                     e

Packet number 4:
  From: 10.0.2.8
  To: 10.0.2.9
  Protocol: TCP
  Src port: 23
  Dst port: 56178

Packet number 5:
  From: 10.0.2.9
  To: 10.0.2.8
  Protocol: TCP
  Src port: 56178
  Dst port: 23
  Payload (1 bytes):
000000  65                                     e
```

```
[09/23/18]seed@VM:~$ telnet 10.0.2.8
Trying 10.0.2.8...
Connected to 10.0.2.8.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Sun Sep 23 17:59:12 EDT 2018 from 10.0.2.8 on pts/19
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i686)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

3 packages can be updated.
0 updates are security updates.
```

## ❖ Task 2: Spoofing

- **Problem 6:** Please use your own words to describe the sequence of the library calls that are essential for packet spoofing. This is meant to be a summary.

The four library calls essential for a packet spoofing process are as follows:

1. Creating a raw socket
2. Setting socket option
3. Constructing the packet
4. Sending out the packet through raw socket

- A raw socket is a socket that allows applications to directly transport packets bypassing all other applications in the network of operating system. Creating a raw socket is the first step of spoofing which helps the program to transport packets in the network.
  - To create custom packets, we need to have knowledge about the structures of the various protocols such as IP, ICMP, TCP and UDP along with information on the data type and sizes of packets.
  - With the knowledge of the packet header we can construct datagrams and insert them into the network.
  - If the datagram created has no errors, then it results in a zero value that denotes that the creation process was successful else it returns a message denoting the error.
- **Problem 7:** Why do you need the root privilege to run the programs that use raw sockets? Where does the program fail if executed without the root privilege?

Raw sockets allows a user to insert packets into a network and transfer packets from one application to the other which denotes that having access to a raw socket means having access to other applications in the network which in turn raises security concerns. These security concerns are the reason why programs that use raw sockets need root permission. A user with root permission can alter the packet header without any interference from the OS which is essentially packet spoofing.

If a spoofing program is executed without root permission the program will throw an exception and fails to create raw socket which denotes, 'Raw Socket: Operation not permitted'.