

2.1.a Writing a Packet Sniffing Program

```
Terminal
14560  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
14576  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....
14592  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 .....

Packet number 9:
  From: 172.217.7.142
  To: 10.0.2.15
  Protocol: TCP
  Src port: 80
  Dst port: 46652

Packet number 10:
  From: 172.217.7.142
  To: 10.0.2.15
  Protocol: TCP
  Src port: 80
  Dst port: 46652

Capture complete.
[12/05/2017 17:06] seed@ubuntu:~/Documents$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: eth13
Number of packets: 10
Filter expression: ip

Packet number 1:
  From: 10.0.2.15
  To: 152.19.240.8
  Protocol: UDP

Packet number 2:
  From: 152.19.240.8
  To: 10.0.2.15
  Protocol: UDP

Packet number 3:
  From: 10.0.2.15
  To: 152.19.240.8
  Protocol: UDP

Packet number 4:
  From: 152.19.240.8
  To: 10.0.2.15
  Protocol: UDP
```

```
Terminal
Protocol: UDP

Packet number 3:
  From: 10.0.2.15
  To: 152.19.240.8
  Protocol: UDP

Packet number 4:
  From: 152.19.240.8
  To: 10.0.2.15
  Protocol: UDP

Packet number 5:
  From: 10.0.2.15
  To: 152.19.240.8
  Protocol: UDP

Packet number 6:
  From: 152.19.240.8
  To: 10.0.2.15
  Protocol: UDP

Packet number 7:
  From: 10.0.2.15
  To: 152.19.240.8
  Protocol: UDP

Packet number 8:
  From: 152.19.240.8
  To: 10.0.2.15
  Protocol: UDP

Packet number 9:
  From: 10.0.2.15
  To: 172.217.7.142
  Protocol: TCP
  Src port: 46652
  Dst port: 80

Packet number 10:
  From: 172.217.7.142
  To: 10.0.2.15
  Protocol: TCP
  Src port: 80
  Dst port: 46652

Capture complete.
[12/05/2017 17:08] seed@ubuntu:~/Documents$
```

Problem 1

`pcap_lookupdev()` looks up the interface that is being used on the device.

`pcap_open_live()` used for creating a sniffing session

`pcap_datalink()` returns a value indicating the type of link-layer header the device uses

`pcap_compile()` compiles the filter that we will use for sniffing

`pcap_setfilter()` sets the filter for the sniffing session

`pcap_next()` returns a `u_char` pointer to the packet described by the filter

`pcap_loop()` returns a sniffed packet until it reaches the count of packets wanted by the attacker

Problem 2

You need root privileges to enable promiscuous mode so the host won't know that the attacker is sniffing all traffic on the wire. Having root privileges will allow the attacker to hide that they enabled promiscuous mode.

2.1.b Sniffing Passwords



```
Terminal
sniffex.c:497:3: warning: pointer targets in passing argument 1 of 'print_payload' differ in signedness [-Wpointer-sign]
sniffex.c:373:1: note: expected 'const u_char *' but argument is of type 'const char *'
sniffex.c:424:31: warning: variable 'ethernet' set but not used [-Wunused-but-set-variable]
[12/05/2017 16:01] seed@ubuntu:~/Downloads$ sudo ./sniffex lo
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: lo
Number of packets: 10
Filter expression: port 23

Packet number 1:
  From: 127.0.0.1
  To: 127.0.0.1
  Protocol: TCP
  Src port: 20
  Dst port: 23
  Payload (4 bytes):
0000  64 65 65 73                                dees
```

The sniffer program sniffed out the payload information which was the password. I had to adjust the filter expression and I set the device on the command line.

2.2 Spoofing

Question 4

If the packet length is less than the IP header length, the packet will not send and instead an error is broadcasted that the packet was a bogus length. Basically, as long as the packet length is as large as the IP header length, you can send the packet no matter how big the payload.

Question 5

You do have to calculate the checksum when using C raw socket programming. Scapy calculates the checksum for you.

Question 6

Raw sockets can spoof custom packets that can interfere with incoming traffic which can be bad. Furthermore, root privilege let you break networking rules set in place. The program fails when it tries to create a raw socket for custom packets.