Hengyi Shi

cs528

Lab1 Report

Task 1a:

Problem 1

This is the sequence of pcap library function calls that are essential for the sniffex program:

1. pcap_lookupdev(errbuf):
   To find a device to capture. It will return a list which if it is not empty, it will use the first device in the list as target.
2. pcap_lookupnet(dev, &net, &mask, errbuf):
   It is used to find the IPv4 network number, it will also find the network mask regarding to the network device that's been captured.
3. pcap_open_live(dev, SNAP_LEN, 1, 1000, errbuf):
   It is used to look at packets on the network by obtaining a packet capture handle.
4. pcap_datalink(handle):
   for all the packets that's benne captured, this function finds the link-layer headers for each packet.
5. pcap_compile(handle, &fp, filter_exp, 0, net):
   To determine whether a filter expression is valid or not.
6. pcap_setfilter(handle, &fp):
   Set the compiled filter expression to the device handler.
7. pcap_loop(handle, num_packets, got_packet, NULL):
   It allows to process packet filtering process until the count reaches to the "num_packets".
8. pcap_freecode(&fp):
   It is used to free up the memory allocated by "pcap_compile()"
9. pcap_close(handle):
   it is used to close a capture device or savefile.

Problem 2

The function pcap_lookupdev() and pcap_open_live() need the root privilege to execute. Since the program directly access the low-level network interface. Root

Problem 3

For function "pcap_t *pcap_open_live(char *device, int snaplen, int promisc, int to_ms, char *ebuf)" to turn on the promiscuous mode, we need to set "promisc" value to 1. And we can turn off the promiscuous mode by set "promisc" value to 0.

With the promisc mode on, sniffing host can capture all packets from the net.

With the promisc mode off, sniffing host can only capture packets that were to/from that host.

Tack 1b:

- “(icmp) and ((src net 192.168.15.5) or (dst net 192.168.15.5))”

mc09.cs.purdue.edu - PuTTY

```
Device: eth14
Number of packets: 10
Filter expression: (icmp) and ((src net 192.168.15.5) or (dst net 192.168.15.5))

Packet number 1:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: ICMP

Packet number 2:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: ICMP

Packet number 3:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: ICMP

Packet number 4:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: ICMP
```

mc09.cs.purdue.edu - PuTTY

```
--- 192.168.15.7 ping statistics ---
8 packets transmitted, 0 received, +6 errors, 100% packet loss, time 7048ms
pipe 3
[02/14/2019 12:09] cs528user@cs528vm:~$ ping 192.168.15.6
PING 192.168.15.6 (192.168.15.6) 56(84) bytes of data.
64 bytes from 192.168.15.6: icmp_req=1 ttl=64 time=0.667 ms
64 bytes from 192.168.15.6: icmp_req=2 ttl=64 time=0.569 ms
64 bytes from 192.168.15.6: icmp_req=3 ttl=64 time=0.968 ms
64 bytes from 192.168.15.6: icmp_req=4 ttl=64 time=0.470 ms
64 bytes from 192.168.15.6: icmp_req=5 ttl=64 time=0.929 ms
64 bytes from 192.168.15.6: icmp_req=6 ttl=64 time=0.772 ms
64 bytes from 192.168.15.6: icmp_req=7 ttl=64 time=0.506 ms
64 bytes from 192.168.15.6: icmp_req=8 ttl=64 time=0.496 ms
^C
--- 192.168.15.6 ping statistics ---
8 packets transmitted, 8 received, 0% packet loss, time 7005ms
```

- “(tcp) and (dst portrange 50-100)”

mc09.cs.purdue.edu - PuTTY

```
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: eth14
Number of packets: 10
Filter expression: (tcp) and (dst portrange 50-100)

Packet number 1:
       From: 192.168.15.5
         To: 128.10.25.13
   Protocol: TCP
   Src port: 52410
   Dst port: 50

Packet number 2:
       From: 192.168.15.5
         To: 128.10.2.13
   Protocol: TCP
   Src port: 35387
   Dst port: 50

Packet number 3:
       From: 192.168.15.5
         To: 128.10.12.13
```

mc09.cs.purdue.edu - PuTTY

```
[02/14/2019 12:10] cs528user@cs528vm:~$ ssh user@data.cs.purdue.edu -p 50
ssh: connect to host data.cs.purdue.edu port 50: Connection refused
[02/14/2019 12:22] cs528user@cs528vm:~$
[02/14/2019 12:23] cs528user@cs528vm:~$
[02/14/2019 12:23] cs528user@cs528vm:~$
```

•”port 23”

mc09.cs.purdue.edu - PuTTY

```
Trying 192.186.15.6...
telnet: Unable to connect to remote host: Connection timed out
[02/14/2019 12:31] cs528user@cs528vm:~$ telnet 192.168.15.6
Trying 192.168.15.6...
Connected to 192.168.15.6.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
cs528vm login: cs528user
Password:
Last login: Thu Feb 14 12:28:35 PST 2019 from 192.168.15.2 on pts/3
Welcome to Ubuntu 12.04.2 LTS (GNU/Linux 3.5.0-37-generic i686)

 * Documentation:  https://help.ubuntu.com/

New release '14.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.
```

mc09.cs.purdue.edu - PuTTY

```
00000   50 61 73 73 77 6f 72 64  3a 20                    Password:

Packet number 53:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23

Packet number 54:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   63                                                c

Packet number 55:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 56:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   73                                                s

Packet number 57:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 58:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   35                                                5
```

mc09.cs.purdue.edu - PuTTY

```
   Payload (1 bytes):
00000   35                                                  5

Packet number 59:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 60:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   32                                                  2

Packet number 61:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 62:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   38                                                  8

Packet number 63:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 64:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
```

mc09.cs.purdue.edu - PuTTY

```
00000   70                                                      p

Packet number 65:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 66:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   61                                                      a

Packet number 67:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 68:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   73                                                      s

Packet number 69:
       From: 192.168.15.6
         To: 192.168.15.5
   Protocol: TCP
   Src port: 23
   Dst port: 50186

Packet number 70:
       From: 192.168.15.5
         To: 192.168.15.6
   Protocol: TCP
   Src port: 50186
   Dst port: 23
   Payload (1 bytes):
00000   73                                                      s
```

Task 2

a. For icmp spoofing use “gcc -o icmp_spoofer icmp_spoofer.c” to compile and “sudo ./icmp_spoofer” to run the program.

```
cs528user@cs528vm:~$ sudo ./icmp_spoofer
Spoofing successed!.
```

mc09.cs.purdue.edu - PuTTY

```
^Ccs528user@cs528vm:~$ sudo ./sniffex
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Device: eth14
Number of packets: 10
Filter expression: icmp

Packet number 1:
       From: 192.168.15.7
         To: 192.168.15.5
   Protocol: ICMP
```

For ethernet spoofing use “gcc -o ethernet_spoofer ethernet_spoofer.c” to compile and “sudo ./ethernet_spoofer” to run the program.

```
cs528user@cs528vm:~$ gcc -o ethernet_spoofer ethernet_spoofer.c
cs528user@cs528vm:~$ sudo ./ethernet_spoofer
Spoofing successed!
cs528user@cs528vm:~$
```

mc09.cs.purdue.edu - PuTTY

```
Packet number 4:
       From: 192.168.15.9
         To: 192.168.15.5
   Protocol: ICMP
```

Question 4:

Yes, the IP packet length field can be set to any arbitrary value, regardless of the actual packet size. Since there are no limit test for IP packet length.

Question 5:

No, IP checksum value will not affect packet from transport. It seems like raw socket library don't have any handler for wrong IP checksum value.

Question 6:

Program fail when it tries to create a raw socket because it needs root privilege to run. As we might need to set value to specify some fields, the root privilege ensured that we are authorized to do so.

Question 7:

1. socket(): allow us to create raw socket with ip protocol.
2. sendto(): allow us to send out buffer out the socket.
3. inet_addr(): allow us to convert IP address from string format to internet address format for iphdr struct.

Task 3:

The snip shows the IP (192.168.15.6) reply after received ping.

mc09.cs.purdue.edu - PuTTY

```
Packet number 1:
sniffed a package:
       From: 192.168.15.5
         To: 192.168.15.6
spoof a reply package....
Spoofing complete! Sending spoofed package to 192.168.15.5.

Packet number 2:
sniffed a package:
       From: 192.168.15.6
         To: 192.168.15.5
spoof a reply package....
Spoofing complete! Sending spoofed package to 192.168.15.6.

Packet number 3:
sniffed a package:
       From: 192.168.15.5
         To: 192.168.15.6
spoof a reply package....
Spoofing complete! Sending spoofed package to 192.168.15.5.

Packet number 4:
sniffed a package:
```

mc09.cs.purdue.edu - PuTTY

```
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$
cs528user@cs528vm:~$ ping 192.168.15.6
PING 192.168.15.6 (192.168.15.6) 56(84) bytes of data.
64 bytes from 192.168.15.6: icmp_req=1 ttl=64 time=1.14 ms
64 bytes from 192.168.15.6: icmp_req=2 ttl=64 time=0.775 ms
64 bytes from 192.168.15.6: icmp_req=3 ttl=64 time=0.672 ms
64 bytes from 192.168.15.6: icmp_req=4 ttl=64 time=0.598 ms
^C
--- 192.168.15.6 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3001ms
```