Packet Sniffing and Spoofing Lab

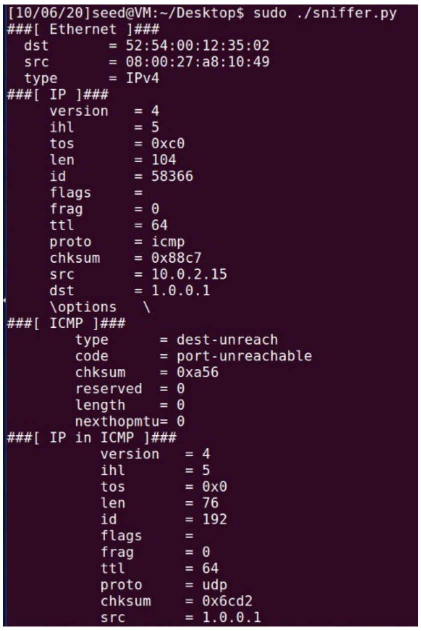
18307130281 庄颖秋

1 Task Set 1

1.1 Task 1.1

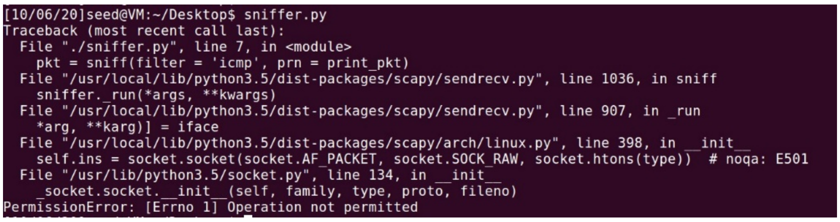
1.1.1 Task 1.1 A

- Run with root privilege



```
[10/06/20]seed@VM:~/Desktop$ sudo ./sniffer.py
###[ Ethernet ]###
  dst       = 52:54:00:12:35:02
  src       = 08:00:27:a8:10:49
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0xc0
     len       = 104
     id        = 58366
     flags     =
     frag      = 0
     ttl       = 64
     proto     = icmp
     chksum    = 0x88c7
     src       = 10.0.2.15
     dst       = 1.0.0.1
     \options   \
###[ ICMP ]###
        type      = dest-unreach
        code      = port-unreachable
        chksum    = 0xa56
        reserved  = 0
        length    = 0
        nexthopmtu= 0
###[ IP in ICMP ]###
           version   = 4
           ihl       = 5
           tos       = 0x0
           len       = 76
           id        = 192
           flags     =
           frag      = 0
           ttl       = 64
           proto     = udp
           chksum    = 0x6cd2
           src       = 1.0.0.1
```

- Run without root privilege



```
[10/06/20]seed@VM:~/Desktop$ sniffer.py
Traceback (most recent call last):
  File "./sniffer.py", line 7, in <module>
    pkt = sniff(filter = 'icmp', prn = print_pkt)
  File "/usr/local/lib/python3.5/dist-packages/scapy/sendrecv.py", line 1036, in sniff
    sniffer._run(*args, **kwargs)
  File "/usr/local/lib/python3.5/dist-packages/scapy/sendrecv.py", line 907, in _run
    *arg, **karg)] = iface
  File "/usr/local/lib/python3.5/dist-packages/scapy/arch/linux.py", line 398, in __init__
    self.ins = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.htons(type))  # noqa: E501
  File "/usr/lib/python3.5/socket.py", line 134, in __init__
    _socket.socket.__init__(self, family, type, proto, fileno)
PermissionError: [Errno 1] Operation not permitted
```

在有 root 权限的时候，程序可以正常运行；在没有 root 权限的时候，程序会在调用 sniff 函数的时候因为系统不允许普通用户初始化 socket 而出现 error。

1.1.2 Task 1.1 B

- Filter 1

```python
#!/usr/bin/python3

# sniffer.py

from scapy.all import *


def print_pkt(pkt):

    pkt.show()


pkt = sniff(filter = 'icmp and port 1025', prn = print_pkt)
```

- Filter 2

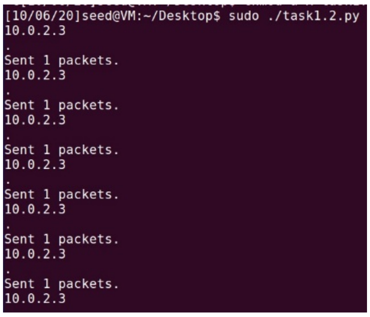
```
#!/usr/bin/python3
# sniffer-1.py
from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt = sniff(filter = 'tcp port 23 and ip host 127.0.0.1', prn = print_pkt)
```

- Filter 3

```
#!/usr/bin/python3
# sniffer-3.py
from scapy.all import *

def print_pkt(pkt):
    pkt.show()

pkt = sniff(filter = 'net 128.230.0.0/16', prn = print_pkt)
```

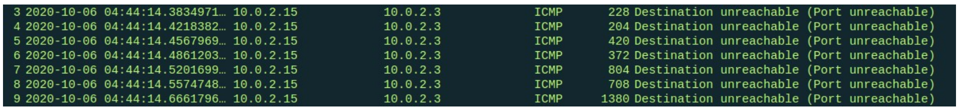
1.2 Task 1.2

```
#!/usr/bin/python3
# task1.2.py
from scapy.all import *

def print_pkt(pkt):
    spoofing_ip = pkt.getlayer(IP)
    spoofing_ip.dst = "10.0.2.3" # change destination
    spoofing_icmp = pkt.getlayer(ICMP)
    spoofing = spoofing_ip/spoofing_icmp
    print(spoofing[IP].dst)
    send(spoofing)

pkt = sniff(filter = 'icmp', prn = print_pkt)
```



```
[10/06/20]seed@VM:~/Desktop$ sudo ./task1.2.py
10.0.2.3
.
Sent 1 packets.
10.0.2.3
.
Sent 1 packets.
10.0.2.3
.
Sent 1 packets.
10.0.2.3
.
Sent 1 packets.
10.0.2.3
.
Sent 1 packets.
10.0.2.3
.
Sent 1 packets.
10.0.2.3
```

WireShark 截图:

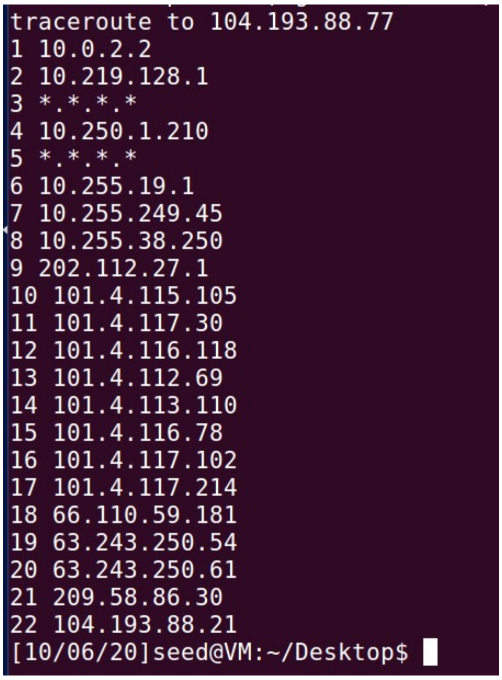


```
 3 2020-10-06 04:44:14.3834971… 10.0.2.15           10.0.2.3            ICMP      228 Destination unreachable (Port unreachable)
 4 2020-10-06 04:44:14.4218382… 10.0.2.15           10.0.2.3            ICMP      204 Destination unreachable (Port unreachable)
 5 2020-10-06 04:44:14.4567969… 10.0.2.15           10.0.2.3            ICMP      420 Destination unreachable (Port unreachable)
 6 2020-10-06 04:44:14.4861203… 10.0.2.15           10.0.2.3            ICMP      372 Destination unreachable (Port unreachable)
 7 2020-10-06 04:44:14.5201699… 10.0.2.15           10.0.2.3            ICMP      804 Destination unreachable (Port unreachable)
 8 2020-10-06 04:44:14.5574748… 10.0.2.15           10.0.2.3            ICMP      708 Destination unreachable (Port unreachable)
 9 2020-10-06 04:44:14.6661796… 10.0.2.15           10.0.2.3            ICMP     1380 Destination unreachable (Port unreachable)
```

1.3 Task 1.3

```python
#!/usr/bin/python3

# task1.3.py

from scapy.all import *


ttl = 1

route = []

while True:

    a = IP()

    a.dst = "104.193.88.77" # www.baidu.com

    a.ttl = ttl

    b = ICMP()

    recv = sr1(a/b, timeout = 5)

    if (str(type(recv)) == "<class 'NoneType'>"): # receive nothing

        route.append("*.*.*.*") # this hop missing

    elif (recv[IP].src == "104.193.88.77"): # reach destination

        break

    else:

        route.append(recv[IP].src)

    ttl += 1


print("traceroute to 104.193.88.77")

for i in range(len(route)):

    print(str(i+1) + " " + route[i])
```

traceroute 104.193.88.77 的程序输出结果：

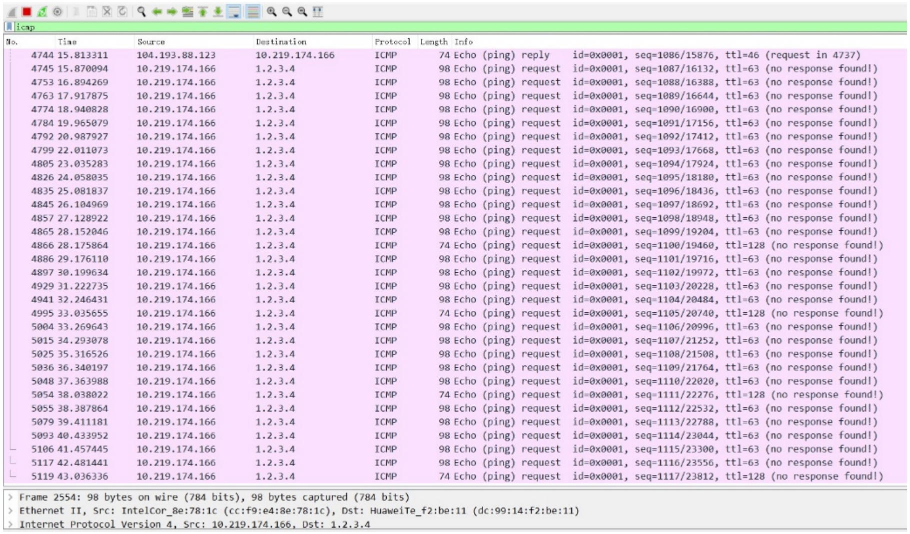


```
traceroute to 104.193.88.77
1 10.0.2.2
2 10.219.128.1
3 *.*.*.*
4 10.250.1.210
5 *.*.*.*
6 10.255.19.1
7 10.255.249.45
8 10.255.38.250
9 202.112.27.1
10 101.4.115.105
11 101.4.117.30
12 101.4.116.118
13 101.4.112.69
14 101.4.113.110
15 101.4.116.78
16 101.4.117.102
17 101.4.117.214
18 66.110.59.181
19 63.243.250.54
20 63.243.250.61
21 209.58.86.30
22 104.193.88.21
[10/06/20]seed@VM:~/Desktop$ 
```

1.4 Task 1.4

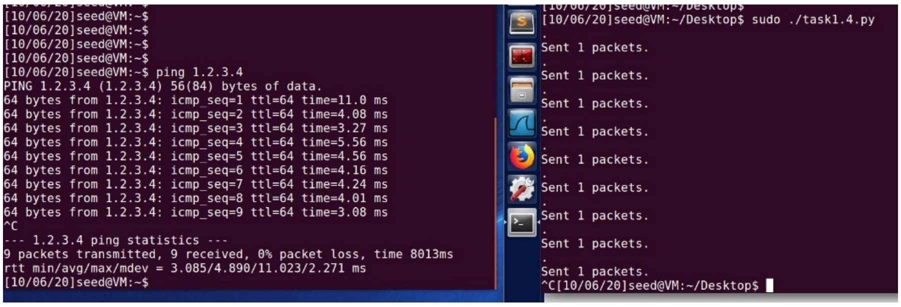
```python
#!/usr/bin/python3
# task1.4.py
from scapy.all import *


def print_pkt(pkt):
    recv_ip = pkt.getlayer(IP)
    a = IP(src = recv_ip.dst, dst = recv_ip.src)
    recv_icmp = pkt.getlayer(ICMP)
    b = ICMP(type = "echo-reply", id = recv_icmp.id, seq = recv_icmp.seq)
    # id and seq are randomly chosen by src
    d = pkt[Raw].load
    s = a/b/d
    send(s)


pkt = sniff(filter = 'icmp[icmptype] == icmp-echo', prn = print_pkt) # want to sniff a echo-request
```

程序未运行时 ping 1.2.3.4 无回应的 WireShark 截图：



```
No.    Time           Source            Destination     Protocol Length Info
  4744 15.813311      104.193.88.123    10.219.174.166  ICMP       74 Echo (ping) reply    id=0x0001, seq=1086/15876, ttl=46 (request in 4737)
  4745 15.870094      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1087/16132, ttl=63 (no response found!)
  4753 16.894269      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1088/16388, ttl=63 (no response found!)
  4763 17.917875      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1089/16644, ttl=63 (no response found!)
  4774 18.940828      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1090/16900, ttl=63 (no response found!)
  4784 19.965079      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1091/17156, ttl=63 (no response found!)
  4792 20.987927      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1092/17412, ttl=63 (no response found!)
  4799 22.011073      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1093/17668, ttl=63 (no response found!)
  4805 23.035283      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1094/17924, ttl=63 (no response found!)
  4826 24.058035      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1095/18180, ttl=63 (no response found!)
  4835 25.081837      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1096/18436, ttl=63 (no response found!)
  4845 26.104969      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1097/18692, ttl=63 (no response found!)
  4857 27.128922      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1098/18948, ttl=63 (no response found!)
  4865 28.152046      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1099/19204, ttl=63 (no response found!)
  4866 28.175864      10.219.174.166    1.2.3.4         ICMP       74 Echo (ping) request  id=0x0001, seq=1100/19460, ttl=128 (no response found!)
  4886 29.176110      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1101/19716, ttl=63 (no response found!)
  4897 30.199634      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1102/19972, ttl=63 (no response found!)
  4929 31.222735      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1103/20228, ttl=63 (no response found!)
  4941 32.246431      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1104/20484, ttl=63 (no response found!)
  4995 33.035655      10.219.174.166    1.2.3.4         ICMP       74 Echo (ping) request  id=0x0001, seq=1105/20740, ttl=128 (no response found!)
  5004 33.269643      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1106/20996, ttl=63 (no response found!)
  5015 34.293078      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1107/21252, ttl=63 (no response found!)
  5025 35.316526      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1108/21508, ttl=63 (no response found!)
  5036 36.340197      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1109/21764, ttl=63 (no response found!)
  5048 37.363988      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1110/22020, ttl=63 (no response found!)
  5054 38.038022      10.219.174.166    1.2.3.4         ICMP       74 Echo (ping) request  id=0x0001, seq=1111/22276, ttl=128 (no response found!)
  5055 38.387864      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1112/22532, ttl=63 (no response found!)
  5079 39.411181      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1113/22788, ttl=63 (no response found!)
  5093 40.433952      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1114/23044, ttl=63 (no response found!)
  5106 41.457445      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1115/23300, ttl=63 (no response found!)
  5117 42.481441      10.219.174.166    1.2.3.4         ICMP       98 Echo (ping) request  id=0x0001, seq=1116/23556, ttl=63 (no response found!)
  5119 43.036336      10.219.174.166    1.2.3.4         ICMP       74 Echo (ping) request  id=0x0001, seq=1117/23812, ttl=128 (no response found!)

> Frame 2554: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: IntelCor_8e:78:1c (cc:f9:e4:8e:78:1c), Dst: HuaweiTe_f2:be:11 (dc:99:14:f2:be:11)
> Internet Protocol Version 4, Src: 10.219.174.166, Dst: 1.2.3.4
```

程序运行后 ping 1.2.3.4 在终端上可以看到回复（回复对应于程序发出的数据包）：



```
[10/06/20]seed@VM:~$
[10/06/20]seed@VM:~$
[10/06/20]seed@VM:~$
[10/06/20]seed@VM:~$
[10/06/20]seed@VM:~$ ping 1.2.3.4
PING 1.2.3.4 (1.2.3.4) 56(84) bytes of data.
64 bytes from 1.2.3.4: icmp_seq=1 ttl=64 time=11.0 ms
64 bytes from 1.2.3.4: icmp_seq=2 ttl=64 time=4.08 ms
64 bytes from 1.2.3.4: icmp_seq=3 ttl=64 time=3.27 ms
64 bytes from 1.2.3.4: icmp_seq=4 ttl=64 time=5.56 ms
64 bytes from 1.2.3.4: icmp_seq=5 ttl=64 time=4.56 ms
64 bytes from 1.2.3.4: icmp_seq=6 ttl=64 time=4.16 ms
64 bytes from 1.2.3.4: icmp_seq=7 ttl=64 time=4.24 ms
64 bytes from 1.2.3.4: icmp_seq=8 ttl=64 time=4.01 ms
64 bytes from 1.2.3.4: icmp_seq=9 ttl=64 time=3.08 ms
^C
--- 1.2.3.4 ping statistics ---
9 packets transmitted, 9 received, 0% packet loss, time 8013ms
rtt min/avg/max/mdev = 3.085/4.890/11.023/2.271 ms
[10/06/20]seed@VM:~$
```

```
[10/06/20]seed@VM:~/Desktop$
[10/06/20]seed@VM:~/Desktop$ sudo ./task1.4.py
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
.
Sent 1 packets.
^C[10/06/20]seed@VM:~/Desktop$
```

2 Task Set 2

2.1 Task 2.1

2.1.1 Task 2.1 A

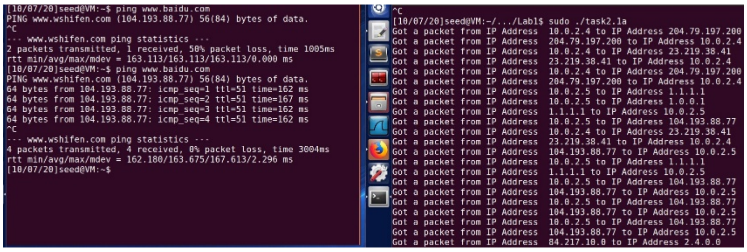
```
// task2.1a.c
#include <pcap.h>
#include <stdio.h>
#include <netinet/ip.h>
#include <netinet/if_ether.h>

void got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char *packet){
    char src_ip[20], dst_ip[20];
    printf("Got a packet ");
    struct ip *ipHeader = (struct ip *)(packet + sizeof(struct ether_header));
    printf("from IP Address ");
    inet_ntop(AF_INET, (void *)&(ipHeader->ip_src), src_ip, 16); // numeric to presentation
    printf(" %s to IP Address ", &src_ip);
    inet_ntop(AF_INET, (void *)&(ipHeader->ip_dst), dst_ip, 16);
    printf("%s\n", &dst_ip);
}

int main(){
    pcap_t *handle;
    char errbuf[PCAP_ERRBUF_SIZE];
    struct bpf_program fp;
    char filter_exp[] = "ip proto icmp";
    bpf_u_int32 net;
    // pcap_t *pcap_open_live(
    // char *device,    Network Interface
    // int snaplen,     maximum bytes captured
    // int promisc,     promiscuous or not ( 1 is on; 0 is off)
    // int to_ms,       timeout (ms)
    // char *ebuf)      convey error message
    handle = pcap_open_live("enp0s3", BUFSIZ, 1, 1000, errbuf);
    // Compile filter_exp into BPF psuedo-code
    pcap_compile(handle, &fp, filter_exp, 0, net);
    pcap_setfilter(handle, &fp);
    // Capture packets
    pcap_loop(handle, -1, got_packet, NULL);
    pcap_close(handle);
    return 0;
}
```

- Answer 1：程序一共使用了 4 个头文件。<stdio.h>提供标准流的输入输出；<pcap.h>提供监听功能需要调用的函数；<netinet/ip.h>提供 IP 层的 C 语言数据结构以及 IP 地址的格式转换函数 inet_ntop(…)，

用于 IP 地址的提取和输出；<netinet/if_ether.h>提供以太网层的 C 语言数据结构，在 Task 1.1 A 的结果截图中可以知道一段 icmp 数据包的结构是以太网层、IP 层、ICMP 层和原始数据，所以通过数据包的地址指针和以太网层数据结构的大小可以推算得出 IP 层数据包在内存中的位置。

- Answer 2: 没有 root 权限运行程序时终端只报了 Segmentation fault 没有具体错误内容。所以根据 Task 1.1 A 中没有 root 权限时的报错，推测程序是在调用 pcap_open_live 方法时需要 root 权限，应该和 scapy 中的 sniff 方法类似，初始化 socket 的步骤涉及到了对底层硬件的访问，需要进入内核模式。
- Answer 3：代码中的注释解释了程序的嗅探如何开启混杂模式。

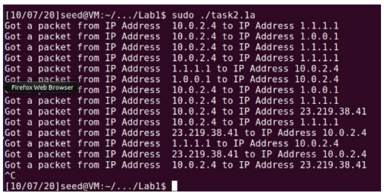
混杂：会嗅探到与本机无关的数据包



```
[10/07/20]seed@VM:~$ ping www.baidu.com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
^C
--- www.wshifen.com ping statistics ---
2 packets transmitted, 1 received, 50% packet loss, time 1005ms
rtt min/avg/max/mdev = 163.113/163.113/163.113/0.000 ms
[10/07/20]seed@VM:~$ ping www.baidu.com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
64 bytes from 104.193.88.77: icmp_seq=1 ttl=51 time=162 ms
64 bytes from 104.193.88.77: icmp_seq=2 ttl=51 time=167 ms
64 bytes from 104.193.88.77: icmp_seq=3 ttl=51 time=162 ms
64 bytes from 104.193.88.77: icmp_seq=4 ttl=51 time=162 ms
^C
--- www.wshifen.com ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3004ms
rtt min/avg/max/mdev = 162.180/163.675/167.613/2.296 ms
[10/07/20]seed@VM:~$
```

```
^C
[10/07/20]seed@VM:~/.../Lab1$ sudo ./task2.1a
Got a packet from IP Address  10.0.2.4 to IP Address 204.79.197.200
Got a packet from IP Address  204.79.197.200 to IP Address 10.0.2.4
Got a packet from IP Address  10.0.2.4 to IP Address 23.219.38.41
Got a packet from IP Address  23.219.38.41 to IP Address 10.0.2.4
Got a packet from IP Address  10.0.2.4 to IP Address 204.79.197.200
Got a packet from IP Address  204.79.197.200 to IP Address 10.0.2.4
Got a packet from IP Address  10.0.2.5 to IP Address 1.1.1.1
Got a packet from IP Address  10.0.2.5 to IP Address 1.0.0.1
Got a packet from IP Address  1.1.1.1 to IP Address 10.0.2.5
Got a packet from IP Address  10.0.2.5 to IP Address 104.193.88.77
Got a packet from IP Address  10.0.2.4 to IP Address 23.219.38.41
Got a packet from IP Address  23.219.38.41 to IP Address 10.0.2.4
Got a packet from IP Address  104.193.88.77 to IP Address 10.0.2.5
Got a packet from IP Address  10.0.2.5 to IP Address 1.1.1.1
Got a packet from IP Address  1.1.1.1 to IP Address 10.0.2.5
Got a packet from IP Address  10.0.2.5 to IP Address 104.193.88.77
Got a packet from IP Address  104.193.88.77 to IP Address 10.0.2.5
Got a packet from IP Address  10.0.2.5 to IP Address 104.193.88.77
Got a packet from IP Address  104.193.88.77 to IP Address 10.0.2.5
Got a packet from IP Address  10.0.2.5 to IP Address 104.193.88.77
Got a packet from IP Address  104.193.88.77 to IP Address 10.0.2.5
Got a packet from IP Address  84.217.10.0 to IP Address 2.4.0.0
```

非混杂：只会嗅探到与本机相关的数据包

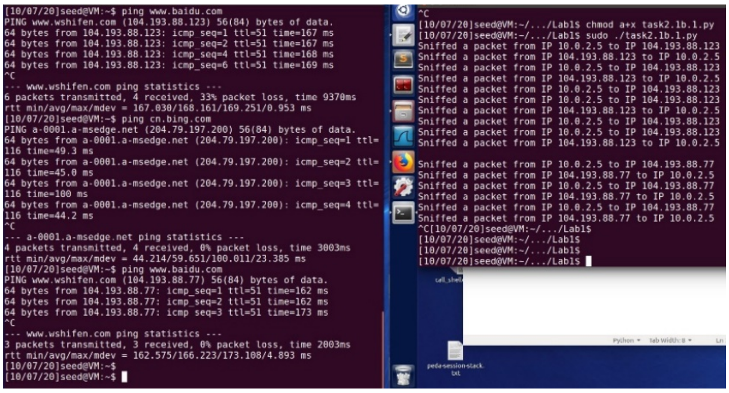


```
[10/07/20]seed@VM:~/.../Lab1$ sudo ./task2.1a
Got a packet from IP Address  10.0.2.4 to IP Address 1.1.1.1
Got a packet from IP Address  10.0.2.4 to IP Address 1.0.0.1
Got a packet from IP Address  10.0.2.4 to IP Address 1.1.1.1
Got a packet from IP Address  10.0.2.4 to IP Address 1.1.1.1
Got a packet from IP Address  1.1.1.1 to IP Address 10.0.2.4
Got a packet from IP Address  1.0.0.1 to IP Address 10.0.2.4
             from IP Address  10.0.2.4 to IP Address 1.0.0.1
Got a packet from IP Address  10.0.2.4 to IP Address 1.1.1.1
Got a packet from IP Address  10.0.2.4 to IP Address 23.219.38.41
Got a packet from IP Address  10.0.2.4 to IP Address 1.1.1.1
Got a packet from IP Address  23.219.38.41 to IP Address 10.0.2.4
Got a packet from IP Address  1.1.1.1 to IP Address 10.0.2.4
Got a packet from IP Address  23.219.38.41 to IP Address 10.0.2.4
Got a packet from IP Address  10.0.2.4 to IP Address 23.219.38.41
^C
[10/07/20]seed@VM:~/.../Lab1$
```

2.1.2 Task 2.1 B

- Filter 1

```
#!/usr/bin/python3

# task2.1b.1.py

from scapy.all import *


def print_pkt(pkt):

    print("Sniffed a packet from IP "+str(pkt[IP].src)+" to IP "+str(pkt[IP].dst))


pkt = sniff(filter = "icmp and host 10.0.2.5 and host www.baidu.com", prn = print_pkt)
```

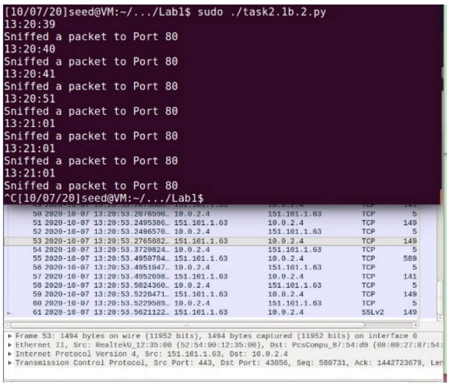


```
[10/07/20]seed@VM:~$ ping www.baidu.com
PING www.wshifen.com (104.193.88.123) 56(84) bytes of data.
64 bytes from 104.193.88.123: icmp_seq=1 ttl=51 time=167 ms
64 bytes from 104.193.88.123: icmp_seq=2 ttl=51 time=167 ms
64 bytes from 104.193.88.123: icmp_seq=4 ttl=51 time=168 ms
64 bytes from 104.193.88.123: icmp_seq=6 ttl=51 time=169 ms
^C
--- www.wshifen.com ping statistics ---
6 packets transmitted, 4 received, 33% packet loss, time 9370ms
rtt min/avg/max/mdev = 167.030/168.161/169.251/0.953 ms
[10/07/20]seed@VM:~$ ping cn.bing.com
PING a-0001.a-msedge.net (204.79.197.200) 56(84) bytes of data.
64 bytes from a-0001.a-msedge.net (204.79.197.200): icmp_seq=1 ttl=
116 time=49.3 ms
64 bytes from a-0001.a-msedge.net (204.79.197.200): icmp_seq=2 ttl=
116 time=45.0 ms
64 bytes from a-0001.a-msedge.net (204.79.197.200): icmp_seq=3 ttl=
116 time=100 ms
64 bytes from a-0001.a-msedge.net (204.79.197.200): icmp_seq=4 ttl=
116 time=44.2 ms
^C
--- a-0001.a-msedge.net ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3003ms
rtt min/avg/max/mdev = 44.214/59.651/100.011/23.385 ms
[10/07/20]seed@VM:~$ ping www.baidu.com
PING www.wshifen.com (104.193.88.77) 56(84) bytes of data.
64 bytes from 104.193.88.77: icmp_seq=1 ttl=51 time=162 ms
64 bytes from 104.193.88.77: icmp_seq=2 ttl=51 time=162 ms
64 bytes from 104.193.88.77: icmp_seq=3 ttl=51 time=173 ms
^C
--- www.wshifen.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2003ms
rtt min/avg/max/mdev = 162.575/166.223/173.108/4.893 ms
[10/07/20]seed@VM:~$
[10/07/20]seed@VM:~$
```

```
^C
[10/07/20]seed@VM:~/.../Lab1$ chmod a+x task2.1b.1.py
[10/07/20]seed@VM:~/.../Lab1$ sudo ./task2.1b.1.py
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 104.193.88.123 to IP 10.0.2.5
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 104.193.88.123 to IP 10.0.2.5
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 104.193.88.123 to IP 10.0.2.5
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.123
Sniffed a packet from IP 104.193.88.123 to IP 10.0.2.5

Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.77
Sniffed a packet from IP 104.193.88.77 to IP 10.0.2.5
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.77
Sniffed a packet from IP 104.193.88.77 to IP 10.0.2.5
Sniffed a packet from IP 10.0.2.5 to IP 104.193.88.77
Sniffed a packet from IP 104.193.88.77 to IP 10.0.2.5
^C[10/07/20]seed@VM:~/.../Lab1$
[10/07/20]seed@VM:~/.../Lab1$
[10/07/20]seed@VM:~/.../Lab1$
```

- Filter 2

```
#!/usr/bin/python3

# task2.1b.2.py

import time

from scapy.all import *


def print_pkt(pkt):

    print(time.strftime("%H:%M:%S", time.localtime()))

    print("Sniffed a packet to Port " + str(pkt[TCP].dport))


pkt = sniff(filter = "tcp", prn = print_pkt)
```

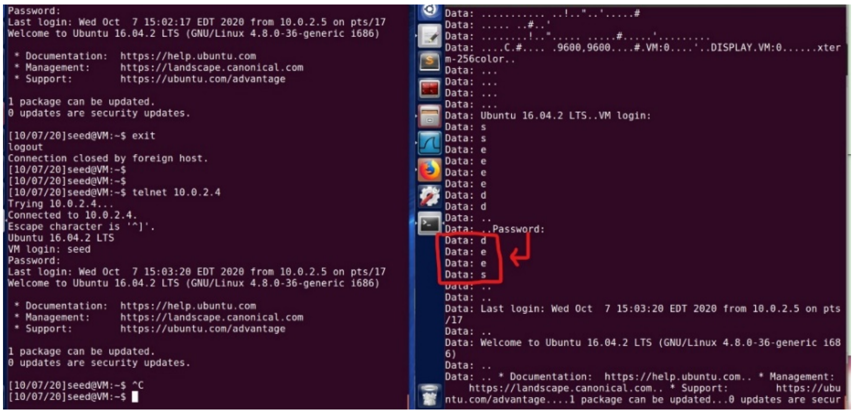


```
[10/07/20]seed@VM:~/.../Lab1$ sudo ./task2.1b.2.py
13:20:39
Sniffed a packet to Port 80
13:20:40
Sniffed a packet to Port 80
13:20:41
Sniffed a packet to Port 80
13:20:51
Sniffed a packet to Port 80
13:21:01
Sniffed a packet to Port 80
13:21:01
Sniffed a packet to Port 80
13:21:01
Sniffed a packet to Port 80
^C[10/07/20]seed@VM:~/.../Lab1$
```

```
50 2020-10-07 13:20:53.2076590… 10.0.2.4        151.101.1.63    TCP      5
51 2020-10-07 13:20:53.2495386… 151.101.1.63    10.0.2.4        TCP    149
52 2020-10-07 13:20:53.2496570… 10.0.2.4        151.101.1.63    TCP      5
53 2020-10-07 13:20:53.2765882… 151.101.1.63    10.0.2.4        TCP    149
54 2020-10-07 13:20:53.3720824… 10.0.2.4        151.101.1.63    TCP      5
55 2020-10-07 13:20:53.4950784… 151.101.1.63    10.0.2.4        TCP    589
56 2020-10-07 13:20:53.4951047… 10.0.2.4        151.101.1.63    TCP      5
57 2020-10-07 13:20:53.4952698… 151.101.1.63    10.0.2.4        TCP    141
58 2020-10-07 13:20:53.5024360… 10.0.2.4        151.101.1.63    TCP      5
59 2020-10-07 13:20:53.5228471… 151.101.1.63    10.0.2.4        TCP    149
60 2020-10-07 13:20:53.5229589… 10.0.2.4        151.101.1.63    TCP      5
61 2020-10-07 13:20:53.5621122… 151.101.1.63    10.0.2.4        SSLv2  149

▸ Frame 53: 1494 bytes on wire (11952 bits), 1494 bytes captured (11952 bits) on interface 0
▸ Ethernet II, Src: RealtekU_12:35:00 (52:54:00:12:35:00), Dst: PcsCompu_87:54:d9 (08:00:27:87:54:
▸ Internet Protocol Version 4, Src: 151.101.1.63, Dst: 10.0.2.4
▸ Transmission Control Protocol, Src Port: 443, Dst Port: 43856, Seq: 580731, Ack: 1442723679, Len
```

2.1.3 Task 2.1 C

```
// task2.1c.c

#include <pcap.h>

#include <stdio.h>

#include <netinet/ip.h>

#include <netinet/tcp.h>

#include <netinet/if_ether.h>


void got_packet(u_char *args, const struct pcap_pkthdr *header, const u_char *packet){

    struct tcphdr *tcpHeader = (struct tcphdr *)(packet + sizeof(struct ether_header) + sizeof(
struct ip)); // Order: Ethernet(IP(TCP(…)))

    int i = 66;

    // learned from packets captured by wireshark

    // if packet includes no information, its length will be 66 bytes

    if (header->len > 66){

        char data[(header->len)-66];

        for (; i < header->len; i ++){

            if(isprint(packet[i])) data[i-66] = packet[i];

            else data[i-66] = '.'; // can't translate to ascii

        }

        data[i-66] = '\0';

        printf("Data: %s\n", data);
    }
}
int main(){
    pcap_t *handle;
    char errbuf[PCAP_ERRBUF_SIZE];
    struct bpf_program fp;
    char filter_exp[] = "tcp";
    bpf_u_int32 net;
    // 此部分注释省略
    handle = pcap_open_live("enp0s3", BUFSIZ, 1, 1000, errbuf);
    // Compile filter_exp into BPF psuedo-code
    pcap_compile(handle, &fp, filter_exp, 0, net);
    pcap_setfilter(handle, &fp);
    // Capture packets
    pcap_loop(handle, -1, got_packet, NULL);
    pcap_close(handle);
    return 0;
}
```



```
Password:
Last login: Wed Oct  7 15:02:17 EDT 2020 from 10.0.2.5 on pts/17
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

1 package can be updated.
0 updates are security updates.

[10/07/20]seed@VM:~$ exit
logout
Connection closed by foreign host.
[10/07/20]seed@VM:~$
[10/07/20]seed@VM:~$
[10/07/20]seed@VM:~$ telnet 10.0.2.4
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
Ubuntu 16.04.2 LTS
VM login: seed
Password:
Last login: Wed Oct  7 15:03:20 EDT 2020 from 10.0.2.5 on pts/17
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

1 package can be updated.
0 updates are security updates.

[10/07/20]seed@VM:~$ ^C
[10/07/20]seed@VM:~$ 
```

```
Data: ........... ..!..".."'.....#
Data: ..... ..#..'
Data: ........!.."..... .....#.....'.........
Data: ....C.#.... .9600,9600....#.VM:0....'..DISPLAY.VM:0......xter
m-256color..
Data: ...
Data: ...
Data: ...
Data: ...
Data: Ubuntu 16.04.2 LTS..VM login:
Data: s
Data: s
Data: e
Data: e
Data: e
Data: e
Data: d
Data: d
Data: ..
Data: ..Password:
Data: d
Data: e
Data: e
Data: s
Data: ..
Data: ..
Data: Last login: Wed Oct  7 15:03:20 EDT 2020 from 10.0.2.5 on pts
/17
Data: ..
Data: Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.8.0-36-generic i68
6)
Data: ..
Data: .. * Documentation:  https://help.ubuntu.com.. * Management:
    https://landscape.canonical.com.. * Support:        https://ubu
ntu.com/advantage....1 package can be updated...0 updates are secur
```

2.2 Task 2.2

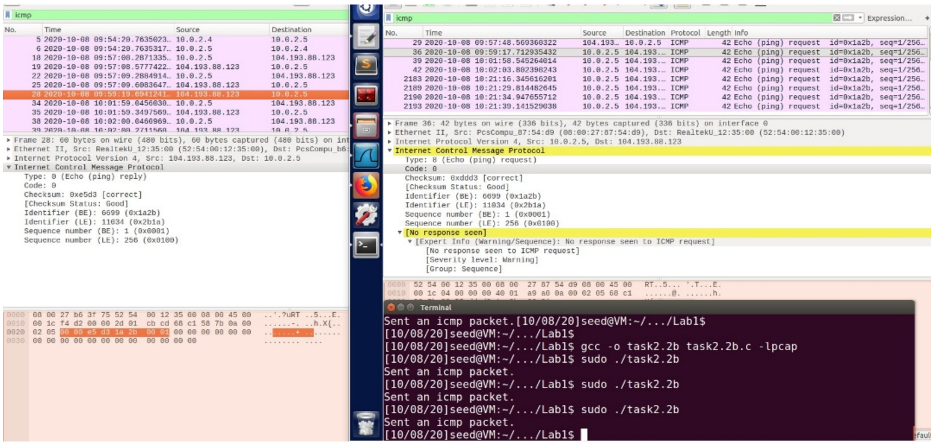
2.2.1 Task 2.2 A

需求包含在 Task 2.2 B 中。

2.2.2 Task 2.2 B

```
// task2.2b.c
#include <pcap.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <netinet/in.h>
#include <sys/socket.h>

#define ICMP_DATA_LENGTH 14
#define SIZE_ETHERNET 14
```

```
typedef struct {
  u_char icmp_type;
  u_char icmp_code;
  u_short checksum;
  u_short id;
  u_short seq;
}ICMP;
typedef struct{
  u_char ver_header_len;
  u_char tos;
  u_short total_len;
  u_short ident;
  u_short flags;
  u_char ttl;
  u_char proto;
  u_short checksum;
  u_char srcIP[4];
  u_char dstIP[4];
}IPHeader;
unsigned short checksum(unsigned short *buffer, int size){
  int checksum = 0;
  while(size > 1){
    checksum += *buffer ++;
    size -= sizeof(unsigned short);
  }
  if(size) checksum += *(unsigned char *)buffer;
  checksum = (checksum >> 16) + (checksum & 0xffff);
  checksum += (checksum >> 16);
  return (unsigned short)(~checksum);
}
int spoof(int sock, u_char ip_src[4], u_char ip_dst[4]) {
  char buffer[1024];
  struct sockaddr_in sin;
  int len = 0;
  sin.sin_family = AF_INET;
  IPHeader *ip = (IPHeader *)buffer; // ip is the beginning, followed by icmp
  ICMP *icmp = (ICMP *)(buffer + sizeof(IPHeader));
  len += sizeof(ICMP);
  icmp->icmp_type = 8; // echo-request
  icmp->icmp_code = 0;
  icmp->id = htons(0x1a2b); // random
  icmp->seq = htons(0x1); // random
  icmp->checksum = checksum((unsigned short*)icmp, sizeof(ICMP));
  len += sizeof(IPHeader);
  ip->ver_header_len = 4 << 4 | 5;
  ip->tos = 0;
  ip->flags = 0;
  ip->ttl = 64;
  ip->proto = 1; // icmp
  for(int i = 0; i < 4; i ++) {
    ip->srcIP[i] = ip_src[i];
    ip->dstIP[i] = ip_dst[i];
  }
  sendto(sock, buffer, len, 0, (struct sockaddr *)&sin, sizeof(sin));
  printf("Sent an icmp packet.\n");
}

int main() {
  int sock = socket(AF_INET, SOCK_RAW, IPPROTO_RAW);
  if(sock < 0){
    perror("socket() error"); exit(-1);
  }
  u_char srcIP[4] = {10, 0, 2, 5};
  u_char dstIP[4] = {104,193,88,123}; // www.baidu.com
  spoof(sock, srcIP, dstIP);
  return 0;
}
```

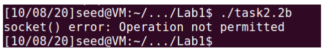


```
No.  Time                           Source          Destination
   5 2020-10-08 09:54:20.7635023…   10.0.2.4        10.0.2.5
   6 2020-10-08 09:54:20.7635317…   10.0.2.5        10.0.2.4
  18 2020-10-08 09:57:08.2871335…   10.0.2.5        104.193.88.123
  19 2020-10-08 09:57:08.5777422…   104.193.88.123  10.0.2.5
  22 2020-10-08 09:57:09.2884914…   10.0.2.5        104.193.88.123
  25 2020-10-08 09:57:09.6083647…   104.193.88.123  10.0.2.5
  28 2020-10-08 09:59:19.6941241…   104.193.88.123  10.0.2.5
  34 2020-10-08 10:01:59.0456030…   10.0.2.5        104.193.88.123
  35 2020-10-08 10:01:59.3497569…   104.193.88.123  10.0.2.5
  38 2020-10-08 10:02:00.0460969…   10.0.2.5        104.193.88.123
▸ Frame 28: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on int
▸ Ethernet II, Src: RealtekU_12:35:00 (52:54:00:12:35:00), Dst: PcsCompu_b6
▸ Internet Protocol Version 4, Src: 104.193.88.123, Dst: 10.0.2.5
▾ Internet Control Message Protocol
    Type: 0 (Echo (ping) reply)
    Code: 0
    Checksum: 0xe5d3 [correct]
    [Checksum Status: Good]
    Identifier (BE): 6699 (0x1a2b)
    Identifier (LE): 11034 (0x2b1a)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
0000  08 00 27 b6 3f 75 52 54  00 12 35 00 08 00 45 00   ..'.?uRT ..5...E.
0010  00 1c f4 d2 00 00 2d 01  cb cd 68 c1 58 7b 0a 00   ......-. ..h.X{..
0020  02 05 00 00 e5 d3 1a 2b  00 01 00 00 00 00 00 00   ........ ........
0030  00 00 00 00 00 00 00 00  00 00 00 00               ........ ....
```

```
No.  Time                           Source    Destination  Protocol Length Info
  29 2020-10-08 09:57:48.569360322  104.193…  10.0.2.5     ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
  36 2020-10-08 09:59:17.712935432  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
  39 2020-10-08 10:01:58.545264014  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
  42 2020-10-08 10:02:03.802398243  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
2183 2020-10-08 10:21:16.345616201  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
2189 2020-10-08 10:21:29.814482645  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
2190 2020-10-08 10:21:34.947655712  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
2193 2020-10-08 10:21:39.141529038  10.0.2.5  104.193...   ICMP     42 Echo (ping) request  id=0x1a2b, seq=1/256…
▸ Frame 36: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface 0
▸ Ethernet II, Src: PcsCompu_87:54:d9 (08:00:27:87:54:d9), Dst: RealtekU_12:35:00 (52:54:00:12:35:00)
▸ Internet Protocol Version 4, Src: 10.0.2.5, Dst: 104.193.88.123
▾ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
    Checksum: 0xddd3 [correct]
    [Checksum Status: Good]
    Identifier (BE): 6699 (0x1a2b)
    Identifier (LE): 11034 (0x2b1a)
    Sequence number (BE): 1 (0x0001)
    Sequence number (LE): 256 (0x0100)
  ▾ [No response seen]
    ▾ [Expert Info (Warning/Sequence): No response seen to ICMP request]
        [No response seen to ICMP request]
        [Severity level: Warning]
        [Group: Sequence]
0000  52 54 00 12 35 00 08 00  27 87 54 d9 08 00 45 00   RT..5... '.T...E.
0010  00 1c 04 00 00 00 40 01  a9 a0 0a 00 02 05 68 c1   ......@. ......h.
```

```
Sent an icmp packet.[10/08/20]seed@VM:~/.../Lab1$
[10/08/20]seed@VM:~/.../Lab1$
[10/08/20]seed@VM:~/.../Lab1$ gcc -o task2.2b task2.2b.c -lpcap
[10/08/20]seed@VM:~/.../Lab1$ sudo ./task2.2b
Sent an icmp packet.
[10/08/20]seed@VM:~/.../Lab1$ sudo ./task2.2b
Sent an icmp packet.
[10/08/20]seed@VM:~/.../Lab1$ sudo ./task2.2b
Sent an icmp packet.
[10/08/20]seed@VM:~/.../Lab1$
```

- Answer 4：不可以，否则的话程序运行时会报 Segmentation fault
- Answer 5：需要计算 checksum，因为使用的是自己定义的数据结构，不会在发送时自动计算 chksum 填入 IP 层和 icmp 层的数据包当中。
- Answer 6：无 root 权限运行的报错信息的最底层函数与之前 python 嗅探程序无 root 权限运行时的报错信息的最底层函数相同，所以此问解答可以详见之前类似的问题。



```
[10/08/20]seed@VM:~/.../Lab1$ ./task2.2b
socket() error: Operation not permitted
[10/08/20]seed@VM:~/.../Lab1$
```

2.3 Task 2.3

由于助教之前说 Lab 没有语言要求，这一部分详见 1.4 节，用 Python 实现。