

MANVINDER TOOR  
CS380  
EXERCISE 5

Problem 1: Verifying the Network

```
Terminal
[11/02/2017 21:15] seed@ubuntu:~$ ifconfig
eth13    Link encap:Ethernet  HWaddr 08:00:27:11:f5:29
         inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
         inet6 addr: fe80::a00:27ff:fe11:f529/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:92 errors:0 dropped:0 overruns:0 frame:0
         TX packets:90 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:16334 (16.3 KB)  TX bytes:12572 (12.5 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:22 errors:0 dropped:0 overruns:0 frame:0
         TX packets:22 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:1865 (1.8 KB)  TX bytes:1865 (1.8 KB)

[11/02/2017 21:15] seed@ubuntu:~$
```

10.0.2.4

```
Terminal
[11/02/2017 21:15] seed@ubuntu:~$ ifconfig
eth14    Link encap:Ethernet  HWaddr 08:00:27:a7:bc:82
         inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
         inet6 addr: fe80::a00:27ff:fea7:bc82/64 Scope:Link
         UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
         RX packets:43 errors:0 dropped:0 overruns:0 frame:0
         TX packets:100 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:1000
         RX bytes:8429 (8.4 KB)  TX bytes:13308 (13.3 KB)

lo       Link encap:Local Loopback
         inet addr:127.0.0.1  Mask:255.0.0.0
         inet6 addr: ::1/128 Scope:Host
         UP LOOPBACK RUNNING  MTU:16436  Metric:1
         RX packets:26 errors:0 dropped:0 overruns:0 frame:0
         TX packets:26 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:0
         RX bytes:2222 (2.2 KB)  TX bytes:2222 (2.2 KB)

[11/02/2017 21:15] seed@ubuntu:~$
```

10.0.2.5

Different HWaddr (Mac Addr)

```
[11/02/2017 21:15] seed@ubuntu:~$ ping -c 5 10.0.2.5
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.
64 bytes from 10.0.2.5: icmp_req=1 ttl=64 time=0.344 ms
64 bytes from 10.0.2.5: icmp_req=2 ttl=64 time=0.720 ms
64 bytes from 10.0.2.5: icmp_req=3 ttl=64 time=0.679 ms
64 bytes from 10.0.2.5: icmp_req=4 ttl=64 time=0.532 ms
64 bytes from 10.0.2.5: icmp_req=5 ttl=64 time=0.507 ms

--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.344/0.556/0.720/0.135 ms
[11/02/2017 21:19] seed@ubuntu:~$
```

0% packet loss

```
[11/02/2017 21:15] seed@ubuntu:~$ ping -c 5 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_req=1 ttl=64 time=0.442 ms
64 bytes from 10.0.2.4: icmp_req=2 ttl=64 time=0.684 ms
64 bytes from 10.0.2.4: icmp_req=3 ttl=64 time=0.674 ms
64 bytes from 10.0.2.4: icmp_req=4 ttl=64 time=0.701 ms
64 bytes from 10.0.2.4: icmp_req=5 ttl=64 time=0.696 ms

--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.442/0.639/0.701/0.101 ms
[11/02/2017 21:18] seed@ubuntu:~$
```

0% packet loss

## Problem 2: Writing a Packet Sniffer

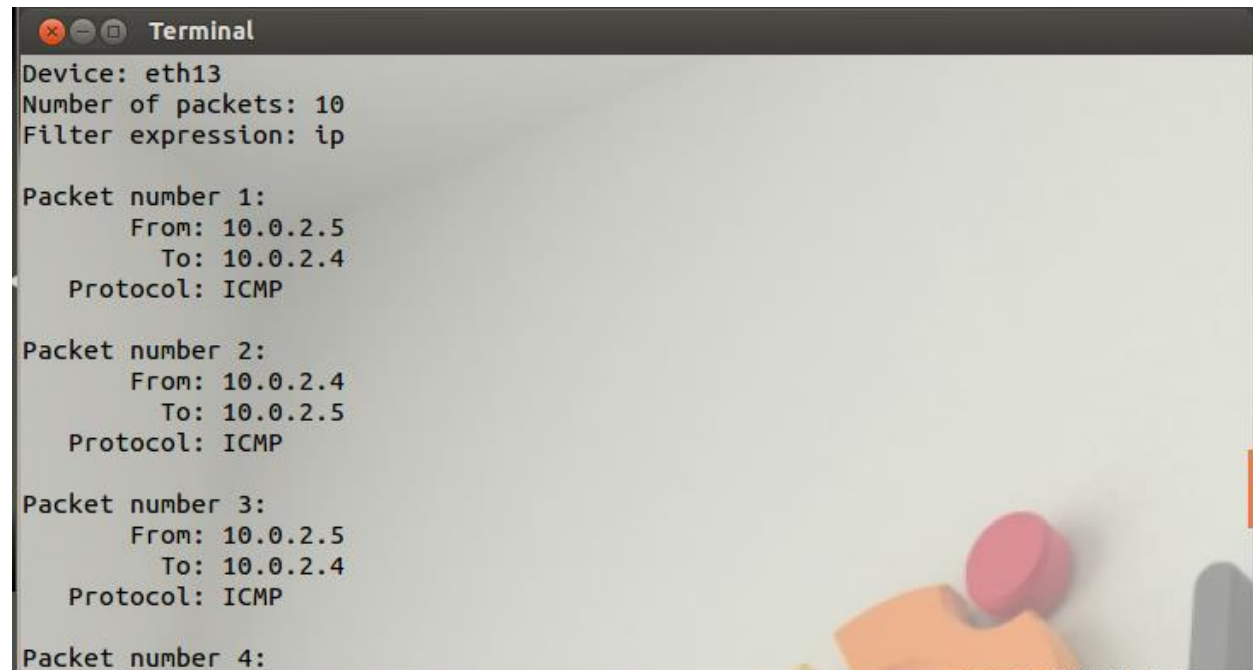
Summarize <http://www.tcpdump.org/pcap.htm>

First we determine the interface we want to sniff on, for example on linux it's the ethX (in our case we are using eth13). After that we initialize pcap and tell it what device we are sniffing on(eth13) we can hold multiple sessions to tell different devices apart. We can then determine if we want to sniff on any particular port or protocol (TCP/UDP/any ports).Next Pcap will wait for a packet and everytime it gets a packet we can do anything we want, we can display it, save it, or do nothing. After it's done the session closes.

```
[11/02/2017 21:38] seed@ubuntu:~/Desktop$ gcc -o sniffex sniffex.c -lpcap
[11/02/2017 21:38] seed@ubuntu:~/Desktop$ ./sniffex eth13
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
Couldn't open device eth13: eth13: You don't have permission to capture on that
device (socket: Operation not permitted)
[11/02/2017 21:38] seed@ubuntu:~/Desktop$
```

Don't have permission to capture on that device, didn't work

A terminal window titled "Terminal" with standard Ubuntu window controls (close, minimize, maximize). The terminal displays the output of the sniffex program after it successfully captured packets. It shows the device (eth13), number of packets (10), and filter expression (ip). It then lists four captured packets, each showing its number, source and destination IP addresses, and the protocol (ICMP).

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

Packet number 1:
    From: 10.0.2.5
    To: 10.0.2.4
    Protocol: ICMP

Packet number 2:
    From: 10.0.2.4
    To: 10.0.2.5
    Protocol: ICMP

Packet number 3:
    From: 10.0.2.5
    To: 10.0.2.4
    Protocol: ICMP

Packet number 4:
```



```
Terminal

Packet number 4:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: ICMP

Packet number 5:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: ICMP

Packet number 6:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: ICMP

Packet number 7:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: ICMP
```

```
Packet number 8:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: ICMP

Packet number 9:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: ICMP

Packet number 10:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: ICMP

Capture complete.
[11/02/2017 21:49] seed@ubuntu:~/Desktop$
```

### Sniffing TCP

```
Capture complete.
[11/02/2017 22:00] seed@ubuntu:~/Desktop$ gcc -o sniffex sniffex.c -lpcap
[11/02/2017 22:10] seed@ubuntu:~/Desktop$ sudo ./sniffex eth13
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: tcp
^C[11/02/2017 22:15] seed@ubuntu:~/Desktop$
```

Nothing happened

### Problem 3: Password Sniffing

Telnetting in

```
[11/02/2017 22:15] seed@ubuntu:~$ telnet 10.0.2.4
Trying 10.0.2.4...
Connected to 10.0.2.4.
Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
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.

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
```

Creating testfile.txt

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

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

[11/02/2017 22:27] seed@ubuntu:~$ ls
Desktop          Music             Pictures
Documents        openssl-1.0.1     Public
Downloads        openssl_1.0.1-4ubuntu5.11.debian.tar.gz  Templates
elggData         openssl_1.0.1-4ubuntu5.11.dsc             Videos
examples.desktop openssl_1.0.1.orig.tar.gz

[11/02/2017 22:27] seed@ubuntu:~$ cd Desktop
[11/02/2017 22:27] seed@ubuntu:~/Desktop$ nano textfile.txt
[11/02/2017 22:28] seed@ubuntu:~/Desktop$
```

Confirming text file

```
Terminal
Capture complete.
[11/02/2017 22:00] seed@ubuntu:~/Desktop$ gcc -o sniffex sniffex.c -lpcap
[11/02/2017 22:10] seed@ubuntu:~/Desktop$ sudo ./sniffex eth13
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: tcp
^C[11/02/2017 22:15] seed@ubuntu:~/Desktop$ sudo service openbsd-inetd start
[sudo] password for seed:
* Starting internet superserver inetd [ OK ]
[11/02/2017 22:26] seed@ubuntu:~/Desktop$ ls
Godit.desktop  libcap2.22  Pacgen-1.10  sniffex.c  Wireshark.desktop
Chex.desktop  Netwag.desktop  sniffex      sniffex.c~
[11/02/2017 22:29] seed@ubuntu:~/Desktop$ ls
Godit.desktop  libcap2.22  Pacgen-1.10  sniffex.c  textfile.txt
Chex.desktop  Netwag.desktop  sniffex      sniffex.c~  Wireshark.desktop
[11/02/2017 22:30] seed@ubuntu:~/Desktop$
```

## Password Sniffing

```
Payload (12 bytes):
00000  0d 0a 50 61 73 73 77 6f 72 64 3a 20      ..Password:

Packet number 31:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
  Src port: 42181
  Dst port: 23

Packet number 32:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
  Src port: 42181
  Dst port: 23
  Payload (1 bytes):
00000  64      d
```

```
Terminal
Packet number 34:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
  Src port: 42181
  Dst port: 23
  Payload (1 bytes):
00000  65                                     e

Packet number 35:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: TCP
  Src port: 23
  Dst port: 42181

Packet number 36:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
```

```
Terminal
Packet number 36:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
  Src port: 42181
  Dst port: 23
  Payload (1 bytes):
00000  65                                     e

Packet number 37:
  From: 10.0.2.4
  To: 10.0.2.5
  Protocol: TCP
  Src port: 23
  Dst port: 42181

Packet number 38:
  From: 10.0.2.5
  To: 10.0.2.4
  Protocol: TCP
```







Telnet doesn't seem that secure since it shows exactly what is happening, doesn't seem to have any encryption

## Problem 4: SSH

| Filter: |                                    | Expression... |             | Clear    | Apply  |
|---------|------------------------------------|---------------|-------------|----------|--|
| No.     | Time                               | Source        | Destination | Protocol | Length Info  |
| 315     | 2017-11-02 23:05:00.810.0.2.4      | 10.0.2.5      | TCP         | 180      | [TCP segment of a reassembled PDU]   |
| 316     | 2017-11-02 23:05:00.810.0.2.4      | 10.0.2.5      | TCP         | 372      | [TCP segment of a reassembled PDU]   |
| 317     | 2017-11-02 23:05:00.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2242 Ack=1946 Win=20544 Len=0 TSval=1575110 TSecr=1239840      |
| 318     | 2017-11-02 23:05:01.010.0.2.4      | 10.0.2.5      | TCP         | 148      | [TCP segment of a reassembled PDU]   |
| 319     | 2017-11-02 23:05:01.110.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2242 Ack=2026 Win=20544 Len=0 TSval=1575178 TSecr=1239899      |
| 320     | 2017-11-02 23:05:10.710.0.2.5      | 10.0.2.4      | TCP         | 116      | [TCP segment of a reassembled PDU]   |
| 321     | 2017-11-02 23:05:10.710.0.2.4      | 10.0.2.5      | TCP         | 116      | [TCP segment of a reassembled PDU]   |
| 322     | 2017-11-02 23:05:10.710.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2290 Ack=2074 Win=20544 Len=0 TSval=1577580 TSecr=1242310      |
| 323     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 116      | [TCP segment of a reassembled PDU]   |
| 324     | 2017-11-02 23:05:11.810.0.2.4      | 10.0.2.5      | TCP         | 116      | [TCP segment of a reassembled PDU]   |
| 325     | 2017-11-02 23:05:11.810.0.2.4      | 10.0.2.5      | TCP         | 100      | [TCP segment of a reassembled PDU]   |
| 326     | 2017-11-02 23:05:11.810.0.2.4      | 10.0.2.5      | TCP         | 228      | [TCP segment of a reassembled PDU]   |
| 327     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2338 Ack=2122 Win=20544 Len=0 TSval=1577862 TSecr=1242591      |
| 328     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2338 Ack=2154 Win=20544 Len=0 TSval=1577862 TSecr=1242591      |
| 329     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2338 Ack=2314 Win=22528 Len=0 TSval=1577862 TSecr=1242591      |
| 330     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 100      | [TCP segment of a reassembled PDU]   |
| 331     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 132      | [TCP segment of a reassembled PDU]   |
| 332     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [FIN, ACK] Seq=2434 Ack=2314 Win=22528 Len=0 TSval=1577862 TSecr=1242591 |
| 333     | 2017-11-02 23:05:11.810.0.2.4      | 10.0.2.5      | TCP         | 68       | ssh > 43243 [ACK] Seq=2314 Ack=2435 Win=25024 Len=0 TSval=1242591 TSecr=1577862      |
| 334     | 2017-11-02 23:05:11.810.0.2.4      | 10.0.2.5      | TCP         | 68       | ssh > 43243 [FIN, ACK] Seq=2314 Ack=2435 Win=25024 Len=0 TSval=1242591 TSecr=1577862 |
| 335     | 2017-11-02 23:05:11.810.0.2.5      | 10.0.2.4      | TCP         | 68       | 43243 > ssh [ACK] Seq=2435 Ack=2315 Win=22528 Len=0 TSval=1577862 TSecr=1242591      |
| 336     | 2017-11-02 23:05:17.5610.0.2.4     | 209.18.47.62  | DNS         | 78       | Standard query A daisy.ubuntu.com  |
| 337     | 2017-11-02 23:05:17.56209.18.47.62 | 10.0.2.4      | DNS         | 110      | Standard query response A 162.213.33.133 A 162.213.33.164                            |

Frame 315: 180 bytes on wire (1440 bits), 180 bytes captured (1440 bits)  
Linux cooked capture  
Internet Protocol Version 4, Src: 10.0.2.4 (10.0.2.4), Dst: 10.0.2.5 (10.0.2.5)  
Transmission Control Protocol, Src Port: ssh (22), Dst Port: 43243 (43243), Seq: 1530, Ack: 2242, Len: 112  
Source port: ssh (22)  
Destination port: 43243 (43243)  
[Stream index: 21]  
Sequence number: 1530 (relative sequence number)  
[Next sequence number: 1642 (relative sequence number)]  
Acknowledgment number: 2242 (relative ack number)

000 00 04 00 01 00 06 08 00 27 11 f5 20 00 00 08 00 ..... '.)....  
010 45 00 00 a4 c9 c1 40 00 40 06 48 6a 09 00 02 04 E....@: 8.H....  
020 0a 00 02 05 00 16 a8 eb 32 cf 36 10 6d 4f 95 a6 ..... 2.6.m0..  
030 80 18 01 07 18 9f 00 00 01 01 08 0a 00 12 eb 20 .....  
040 00 18 00 c5 76 31 42 30 17 ea 56 da 28 a8 2c a1 ...v1800..V{....  
050 04 94 70 27 63 c0 dd 7f 86 6a 6b 3b c0 7f f8 7d ...p'c...;k;....  
060 f5 f3 5a 0b 35 5a 10 e2 9c dc 48 c0 8f 47 75 4f ..2.52...H..0u0  
070 0c 05 08 08 14 63 c3 34 83 f3 f3 8f 1b 6a 36 c3 E...e...e

Ssh seems to not show the data, looks like it's decrypted and harder to decode just by looking at it