

CS 380 Exercise 5

My repository for this class is under CS 380 – Computer Networks

<https://github.com/jarodNakamoto/College-CS-Courses.git>

3.1 Verifying the Network

IP config

```

[10/31/2017 14:03] seed@ubuntu:~$ ifconfig
eth14
  Link encap:Ethernet  HWaddr 08:00:27:6c:57:e2
  inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255.0
  inet6 addr: fe80::a00:27ff:fe6c:57e2/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:226 errors:0 dropped:0 overruns:0 frame:0
  TX packets:248 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:91178 (91.1 KB)  TX bytes:31287 (31.2 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:65536  Metric:1
  RX packets:0 errors:0 dropped:0 overruns:0 frame:0
  TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:6583 (6.5 KB)  TX bytes:6583 (6.5 KB)

[10/31/2017 14:03] seed@ubuntu:~$

[10/31/2017 14:02] seed@ubuntu:~$ ifconfig
eth14
  Link encap:Ethernet  HWaddr 08:00:27:b4:5d:a0
  inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255.0
  inet6 addr: fe80::a00:27ff:feb4:5da0/64 Scope:Link
  UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
  RX packets:99 errors:0 dropped:0 overruns:0 frame:0
  TX packets:164 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:48154 (48.1 KB)  TX bytes:18602 (18.6 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:65536  Metric:1
  RX packets:30 errors:0 dropped:0 overruns:0 frame:0
  TX packets:30 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:0
  RX bytes:2791 (2.7 KB)  TX bytes:2791 (2.7 KB)

[10/31/2017 14:03] seed@ubuntu:~$

```

Ping

```

[10/31/2017 14:10] 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=0.563 ms
64 bytes from 10.0.2.5: icmp_req=2 ttl=64 time=1.13 ms
64 bytes from 10.0.2.5: icmp_req=3 ttl=64 time=0.422 ms
64 bytes from 10.0.2.5: icmp_req=4 ttl=64 time=0.798 ms
64 bytes from 10.0.2.5: icmp_req=5 ttl=64 time=0.783 ms
--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 399ms
rtt min/avg/max/mdev = 0.422/0.741/1.139/0.243 ms
[10/31/2017 14:10] seed@ubuntu:~$

[10/31/2017 14:03] 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=0.494 ms
64 bytes from 10.0.2.4: icmp_req=2 ttl=64 time=0.951 ms
64 bytes from 10.0.2.4: icmp_req=3 ttl=64 time=0.784 ms
64 bytes from 10.0.2.4: icmp_req=4 ttl=64 time=0.638 ms
64 bytes from 10.0.2.4: icmp_req=5 ttl=64 time=0.845 ms
--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 400ms
rtt min/avg/max/mdev = 0.494/0.782/0.951/0.156 ms
[10/31/2017 14:10] seed@ubuntu:~$

```

3.2 Packet Sniffer

```
[11/02/2017 17:31] seed@ubuntu:~/Desktop$ gcc -o sniffex sniffex.c -lpcap
[11/02/2017 17:32] seed@ubuntu:~/Desktop$ ./sniffex NatNetwork
sniffex - Sniffer example using libpcap
Copyright (c) 2005 The Tcpdump Group
THERE IS ABSOLUTELY NO WARRANTY FOR THIS PROGRAM.

Couldn't get netmask for device NatNetwork: SIOCGIFADDR: NatNetwork: No such device
Device: NatNetwork
Number of packets: 10
Filter expression: ip
Couldn't open device NatNetwork: NatNetwork: You don't have permission to capture on that device (socket: Operation not permitted)
```

When pinged it with normal privileges, it says that we have insufficient privileges to run this program which means that the OS blocks it for security reasons.

```
[11/02/2017 17:35] seed@ubuntu:~/Desktop$ sudo ./sniffex eth14
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: 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:
    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
```



It works after running it in admin, so it was just a security block by the kernel.

```

int main(int argc, char **argv)
{
    char *dev = NULL;
    char errbuf[PCAP_ERRBUF_SIZE];
    pcap_t *handle;

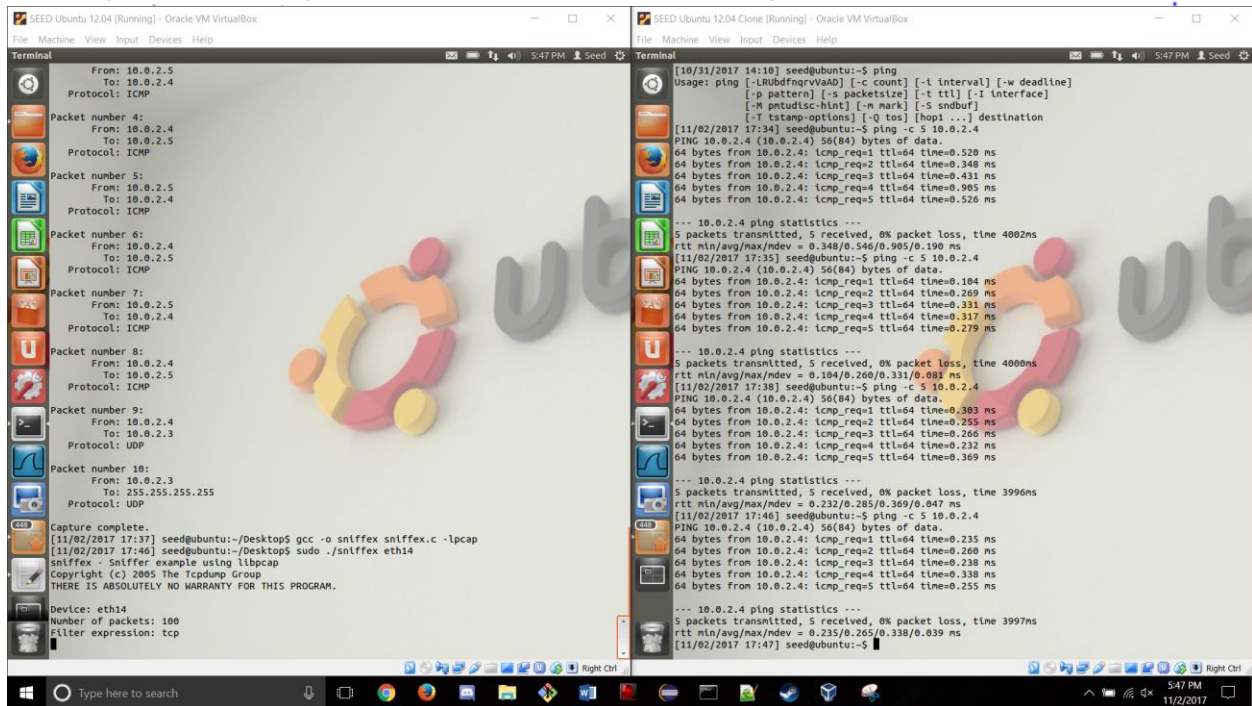
    char filter_exp[] = "tcp";
    struct bpf_program fp;
    bpf_u_int32 mask;
    bpf_u_int32 net;
    int num_packets = 10;

    print_app_banner();

    /* check for capture device name on command-line */
    /* capture device name */
    /* error buffer */
    /* packet capture handle */

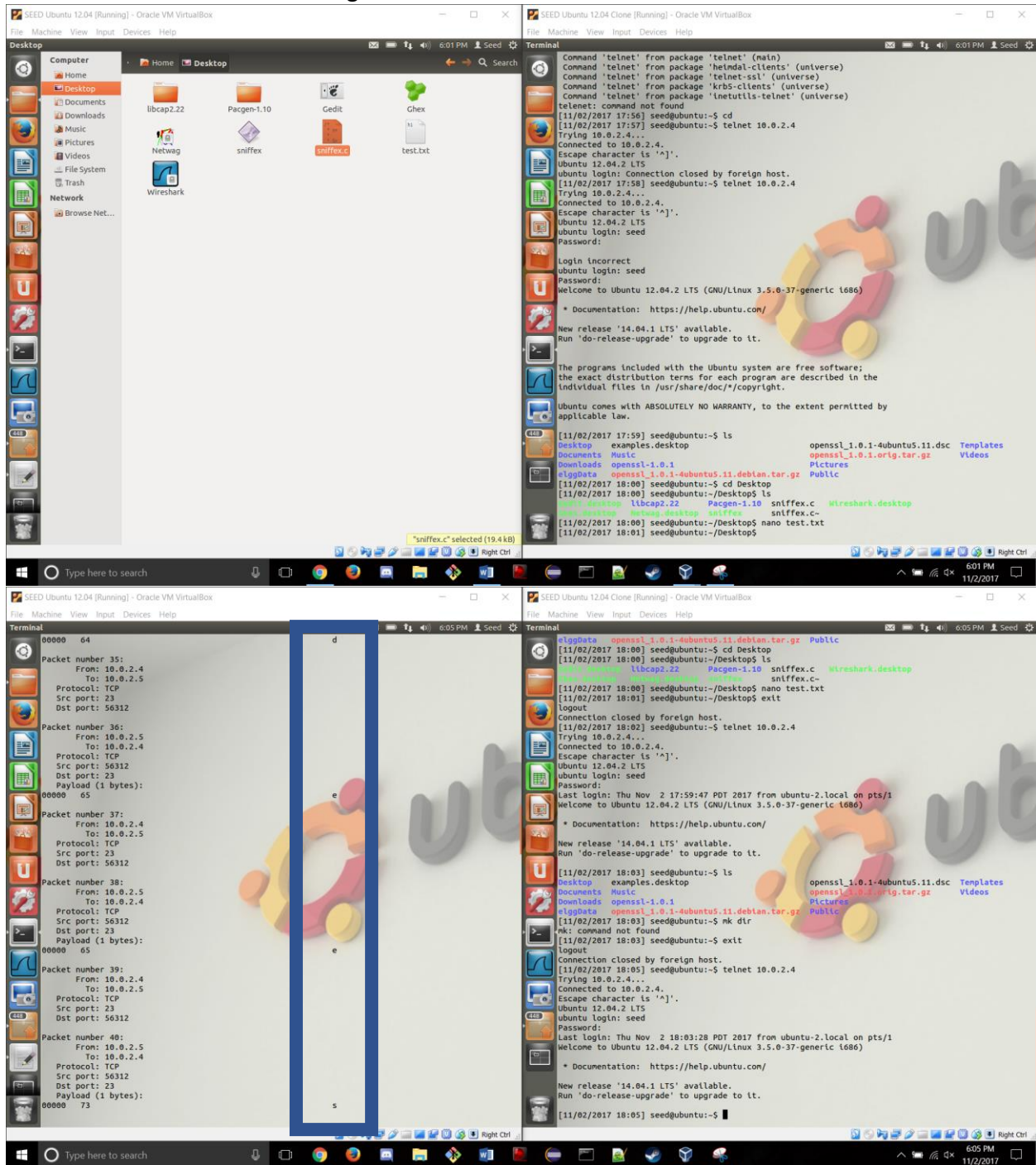
    /* filter expression [3] */
    /* compiled filter program (expression) */
    /* subnet mask */
    /* ip */
    /* number of packets to capture */
}

```

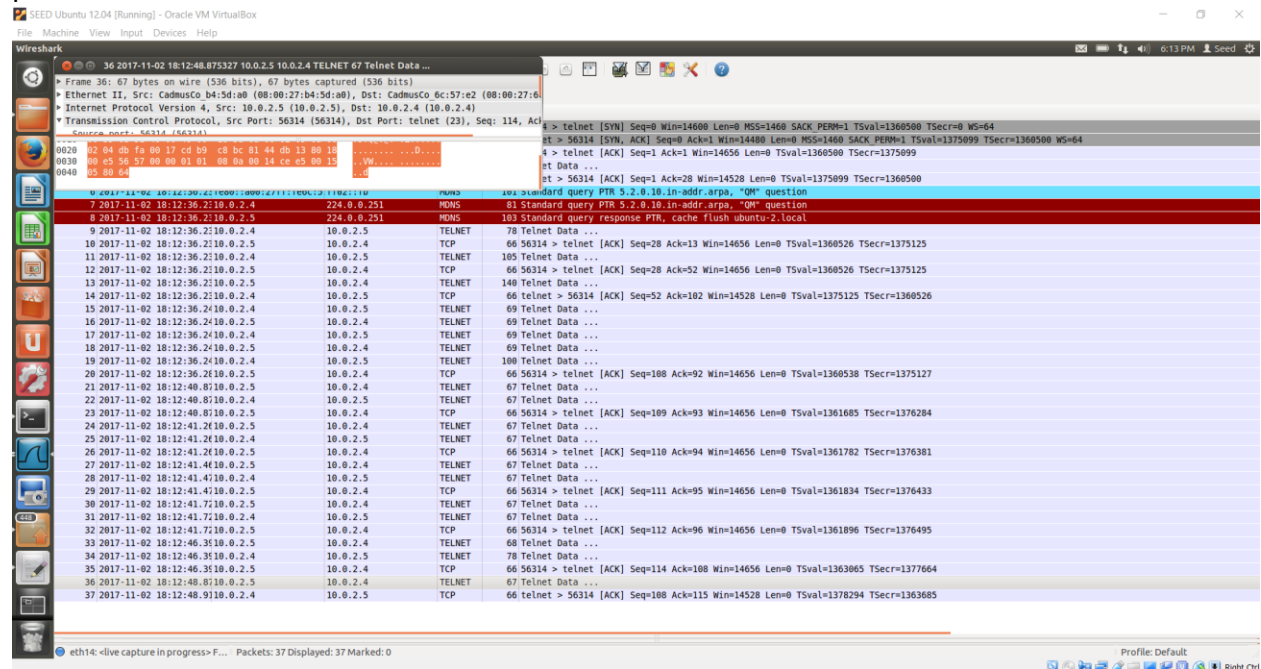


After pinging it twice, there was no change which means there are no tcp packets sent by ping.

3.3 Problem 3: Password Sniffing

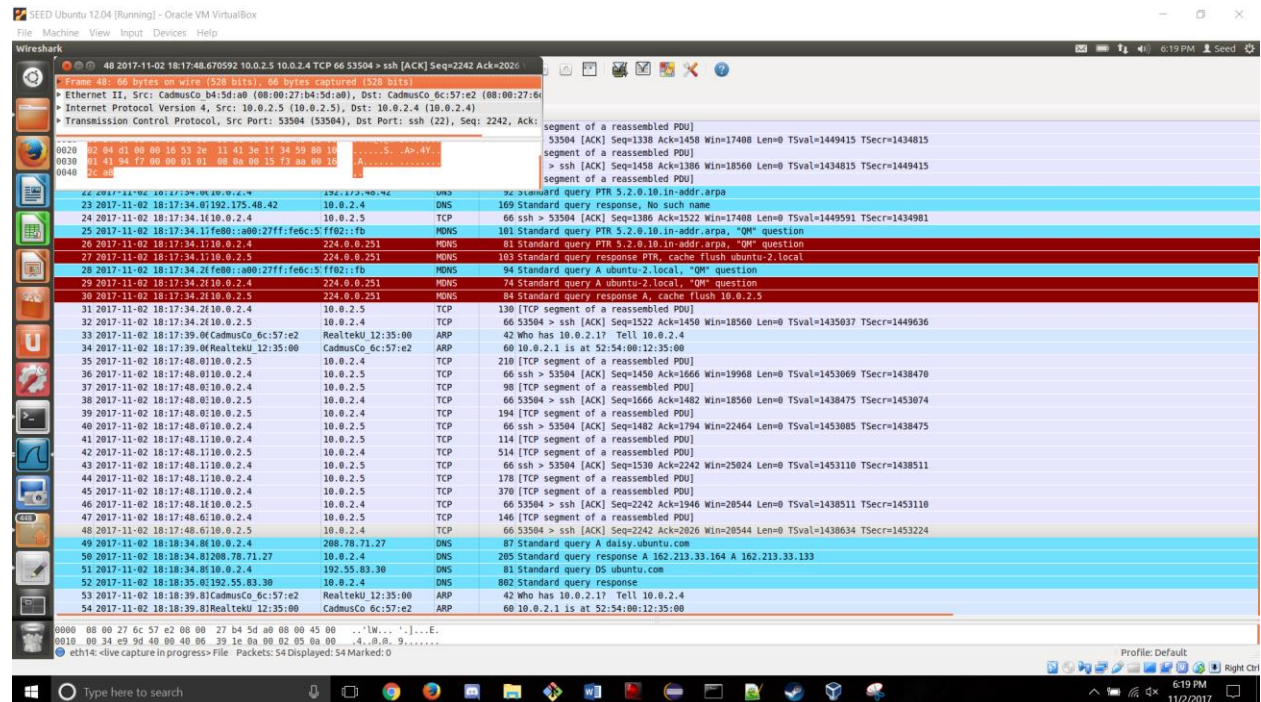


Using Wireshark, it is still possible to find the users passwords at the bottom of the telnetet packets.



Telnet's security is poor as anyone with enough knowledge and a sniffer can see the contents of the data.

3.4 Problem 4 SSH



Even where the password characters should be it is encrypted so the actual password is not found.