

Problem 1

VM 1:

```
eth14 inet 10.0.2.4
```

```
ping: 5 transmitted, 5 recieved, 0% packet loss, time 3997ms
```

VM 2:

```
eth14 inet 10.0.2.5
```

```
ping: 5 transmitted, 5 recieved, 0% packet loss, time 3998ms
```

The image shows two terminal windows side-by-side, both titled 'SEED Ubuntu 12.04 [Running]'. The left window is for VM 1 and the right window is for VM 2. Both windows show the output of the 'ifconfig' command for the 'eth14' interface and the 'lo' loopback interface. The 'eth14' interface is configured with IP address 10.0.2.4 in VM 1 and 10.0.2.5 in VM 2. The 'lo' interface is configured with IP address 127.0.0.1 in both. Below the 'ifconfig' output, both windows show the output of a 'ping -c 5 10.0.2.5' command (in VM 1) or 'ping -c 5 10.0.2.4' command (in VM 2). The ping results show 5 packets transmitted, 5 received, 0% packet loss, and a time of 3997ms for VM 1 and 3998ms for VM 2. The ping statistics show a minimum round-trip time of 0.210ms for VM 1 and 0.201ms for VM 2.

```
[11/02/2017 19:16] seed@ubuntu:~$ ifconfig
eth14    Link encap:Ethernet  HWaddr 08:00:27:fb:cc:53
          inet addr:10.0.2.4  Bcast:10.0.2.255  Mask:255.255.255
          inet6 addr: fe80::a00:27ff:fe16:95bb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:88 errors:0 dropped:0 overruns:0 frame:0
          TX packets:113 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:1000
          RX bytes:13076 (13.0 KB)  TX bytes:14389 (14.3 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 19:16] 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=0.404 ms
64 bytes from 10.0.2.5: icmp_req=2 ttl=64 time=0.225 ms
64 bytes from 10.0.2.5: icmp_req=3 ttl=64 time=0.223 ms
64 bytes from 10.0.2.5: icmp_req=4 ttl=64 time=0.210 ms
64 bytes from 10.0.2.5: icmp_req=5 ttl=64 time=0.223 ms

--- 10.0.2.5 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3997ms
rtt min/avg/max/mdev = 0.210/0.257/0.404/0.073 ms
[11/02/2017 19:17] seed@ubuntu:~$
```

```
[11/02/2017 19:12] seed@ubuntu:~$ ifconfig
eth14    Link encap:Ethernet  HWaddr 08:00:27:16:95:bb
          inet addr:10.0.2.5  Bcast:10.0.2.255  Mask:255.255.255
          inet6 addr: fe80::a00:27ff:fe16:95bb/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:60 errors:0 dropped:0 overruns:0 frame:0
          TX packets:100 errors:0 dropped:0 overruns:0 carrier:
          collisions:0 txqueuelen:1000
          RX bytes:11751 (11.7 KB)  TX bytes:12934 (12.9 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 19:12] 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=0.201 ms
64 bytes from 10.0.2.4: icmp_req=2 ttl=64 time=0.247 ms
64 bytes from 10.0.2.4: icmp_req=3 ttl=64 time=0.225 ms
64 bytes from 10.0.2.4: icmp_req=4 ttl=64 time=0.207 ms
64 bytes from 10.0.2.4: icmp_req=5 ttl=64 time=0.245 ms

--- 10.0.2.4 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3998ms
rtt min/avg/max/mdev = 0.201/0.225/0.247/0.018 ms
[11/02/2017 19:17] seed@ubuntu:~$
```

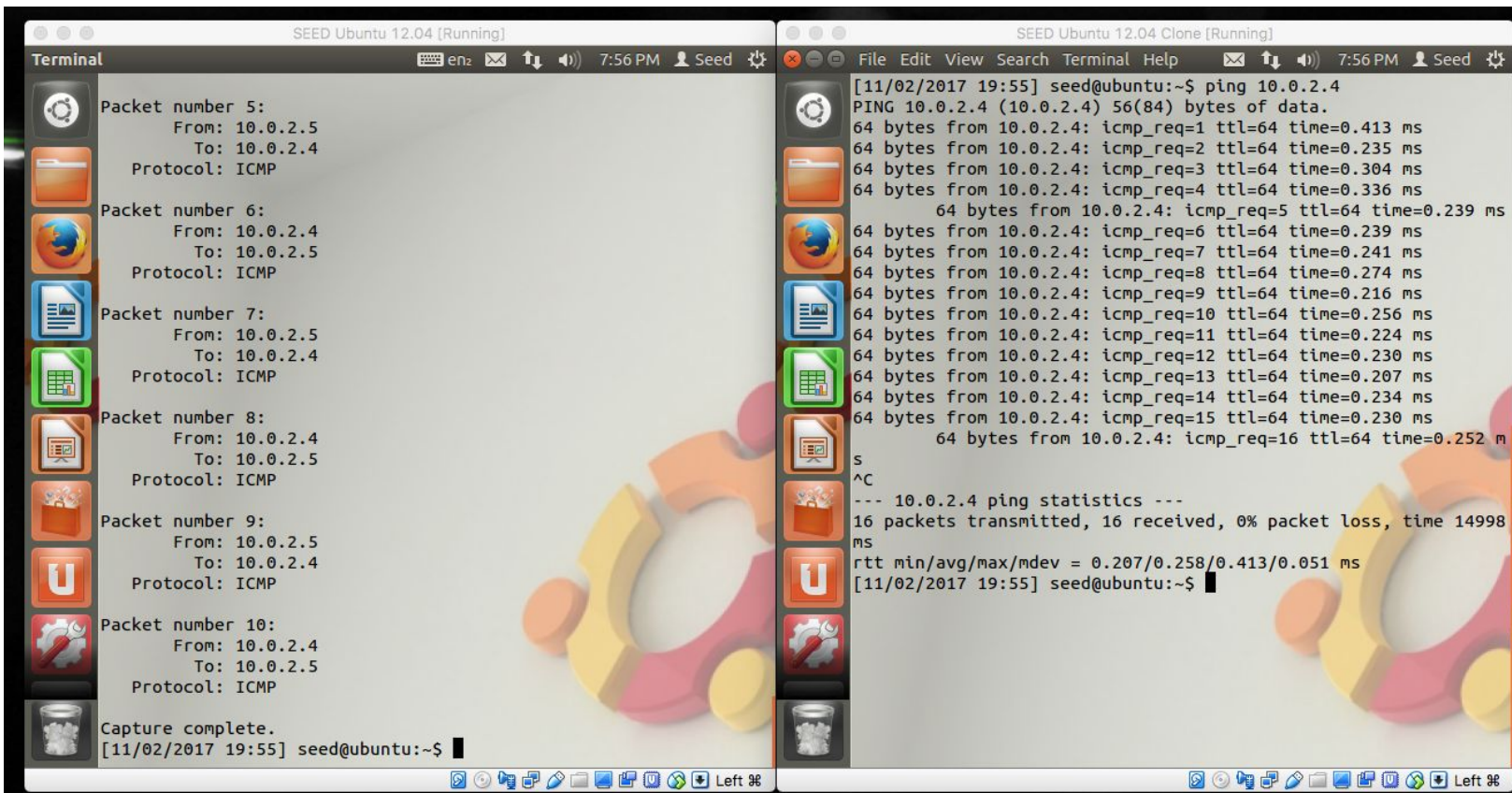
Problem 2

To begin using the `pcap` library, you open a sniffing device by calling the `pcap_open_live()` method. We can use `pcap_compile()` and `pcap_setfilter()` to filter out packets we don't want. We can use this to listen to only traffic on port 22 (SSH) for example. One of the ways we can sniff for packets is by using `pcap_next()` to iteratively capture packets. Alternatively, we can use `pcap_loop()` to continuously sniff packets.

When starting sniffex, we need to run sniffex with `sudo` to grant it root access (to access lower level networking stuff).

Sniffing on VM 1 while pinging VM 1 from VM 2 results in the following:

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



After changing the filter from "ip" to "tcp", we no longer capture the ICMP packets.

SEED Ubuntu 12.04 [Running]

Terminal

GNU nano 2.2.6

File: sniffex.c

Modified

```

        printf("    Payload (%d bytes);\n", size_payload);
        print_payload(payload, size_payload);
    }

return;
}

int main(int argc, char **argv)
{
    char *dev = NULL;                /* capture device */
    char errbuf[PCAP_ERRBUF_SIZE];  /* error buffer */
    pcap_t *handle;                 /* packet capture handle */

    char filter_exp[] = "tcp";      /* filter expression */
    struct bpf_program fp;          /* compiled filter program */
    bpf_u_int32 mask;              /* subnet mask */
    bpf_u_int32 net;               /* ip */
    int num_packets = 10;          /* number of packets to capture */

    print_app_banner();

    /* check for capture device name on command-line */
    if (argc == 2) {
        dev = argv[1];
    }
    else if (argc > 2) {
        fprintf(stderr, "error: unrecognized command-line arguments\n");
        return 1;
    }

    if (dev == NULL) {
        fprintf(stderr, "error: no capture device specified\n");
        return 1;
    }

    if ((handle = pcap_open_live(dev, 4096, 4096, 1000, PCAP_OPENFLAG_PROMISCUOUS)) == NULL) {
        fprintf(stderr, "error: pcap_open_live failed: %s\n", pcap_geterr(handle));
        return 1;
    }

    if (pcap_compile(handle, &fp, filter_exp, 0, PCAP_NETMASK_UNKNOWN) == -1) {
        fprintf(stderr, "error: pcap_compile failed: %s\n", pcap_geterr(handle));
        return 1;
    }

    if (pcap_setfilter(handle, &fp) == -1) {
        fprintf(stderr, "error: pcap_setfilter failed: %s\n", pcap_geterr(handle));
        return 1;
    }

    if (pcap_loop(handle, num_packets, NULL, NULL, 0) == -1) {
        fprintf(stderr, "error: pcap_loop failed: %s\n", pcap_geterr(handle));
        return 1;
    }

    print_stats(handle, num_packets);

    return 0;
}

```

Get Help

Write Out

Read File

Prev Page

Cut Text

Cur Pos

Exit

Justify

Where I Am

Next Page

UnCut Text

To Spell

Left %

SEED Ubuntu 12.04 Clone [Running]

Terminal

```

[11/02/2017 19:55] seed@ubuntu:~$ ping 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=0.413 ms
64 bytes from 10.0.2.4: icmp_req=2 ttl=64 time=0.235 ms
64 bytes from 10.0.2.4: icmp_req=3 ttl=64 time=0.304 ms
64 bytes from 10.0.2.4: icmp_req=4 ttl=64 time=0.336 ms
64 bytes from 10.0.2.4: icmp_req=5 ttl=64 time=0.239 ms
64 bytes from 10.0.2.4: icmp_req=6 ttl=64 time=0.239 ms
64 bytes from 10.0.2.4: icmp_req=7 ttl=64 time=0.241 ms
64 bytes from 10.0.2.4: icmp_req=8 ttl=64 time=0.274 ms
64 bytes from 10.0.2.4: icmp_req=9 ttl=64 time=0.216 ms
64 bytes from 10.0.2.4: icmp_req=10 ttl=64 time=0.256 ms
64 bytes from 10.0.2.4: icmp_req=11 ttl=64 time=0.224 ms
64 bytes from 10.0.2.4: icmp_req=12 ttl=64 time=0.230 ms
64 bytes from 10.0.2.4: icmp_req=13 ttl=64 time=0.207 ms
64 bytes from 10.0.2.4: icmp_req=14 ttl=64 time=0.234 ms
64 bytes from 10.0.2.4: icmp_req=15 ttl=64 time=0.230 ms
64 bytes from 10.0.2.4: icmp_req=16 ttl=64 time=0.252 ms
s
^C
--- 10.0.2.4 ping statistics ---
16 packets transmitted, 16 received, 0% packet loss, time 14998 ms
rtt min/avg/max/mdev = 0.207/0.258/0.413/0.051 ms
[11/02/2017 19:55] seed@ubuntu:~$

```

Get Help

Write Out

Read File

Prev Page

Cut Text

Cur Pos

Exit

Justify

Where I Am

Next Page

UnCut Text

To Spell

Left %

SEED Ubuntu 12.04 [Running]

Terminal

```

[11/02/2017 20:08] seed@ubuntu:~$ 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: tcp

```

Get Help

Write Out

Read File

Prev Page

Cut Text

Cur Pos

Exit

Justify

Where I Am

Next Page

UnCut Text

To Spell

Left %

SEED Ubuntu 12.04 Clone [Running]

Terminal

```

[11/02/2017 20:09] seed@ubuntu:~$ ping 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=0.149 ms
64 bytes from 10.0.2.4: icmp_req=2 ttl=64 time=0.361 ms
64 bytes from 10.0.2.4: icmp_req=3 ttl=64 time=0.222 ms
64 bytes from 10.0.2.4: icmp_req=4 ttl=64 time=0.224 ms
64 bytes from 10.0.2.4: icmp_req=5 ttl=64 time=0.205 ms
64 bytes from 10.0.2.4: icmp_req=6 ttl=64 time=0.357 ms
^C
--- 10.0.2.4 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 4999ms
rtt min/avg/max/mdev = 0.149/0.253/0.361/0.078 ms
[11/02/2017 20:09] seed@ubuntu:~$

```

Get Help

Write Out

Read File

Prev Page

Cut Text

Cur Pos

Exit

Justify

Where I Am

Next Page

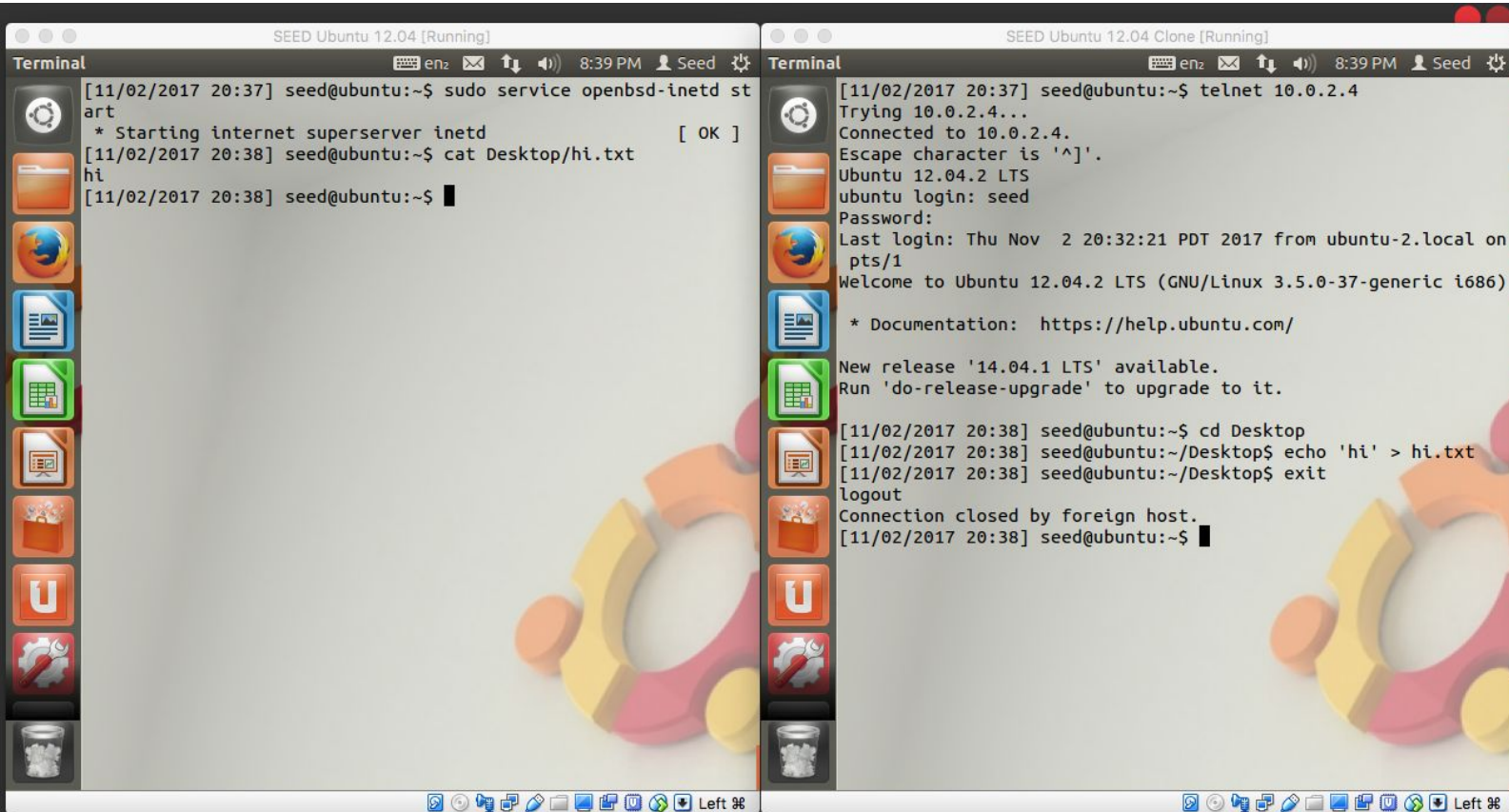
UnCut Text

To Spell

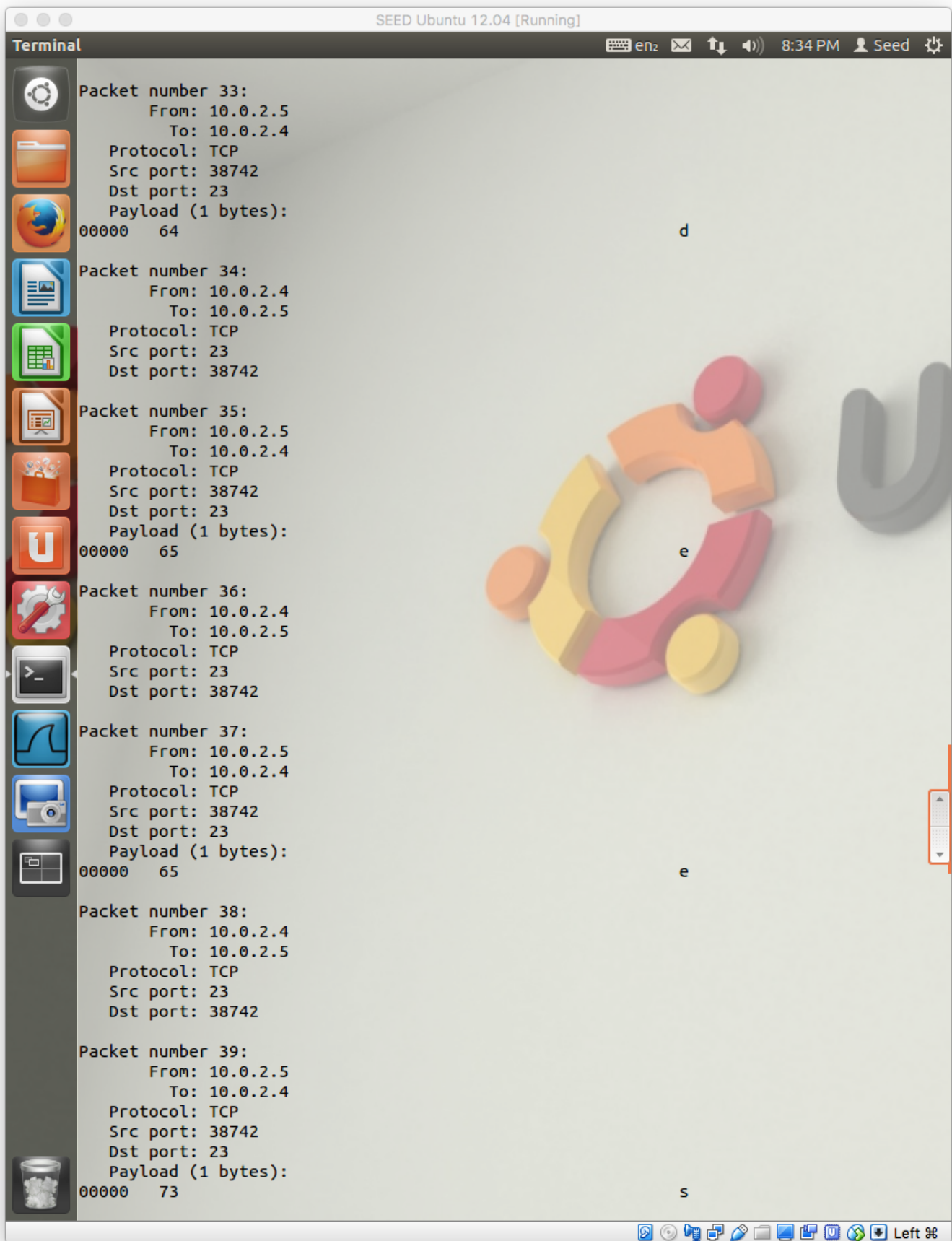
Left %

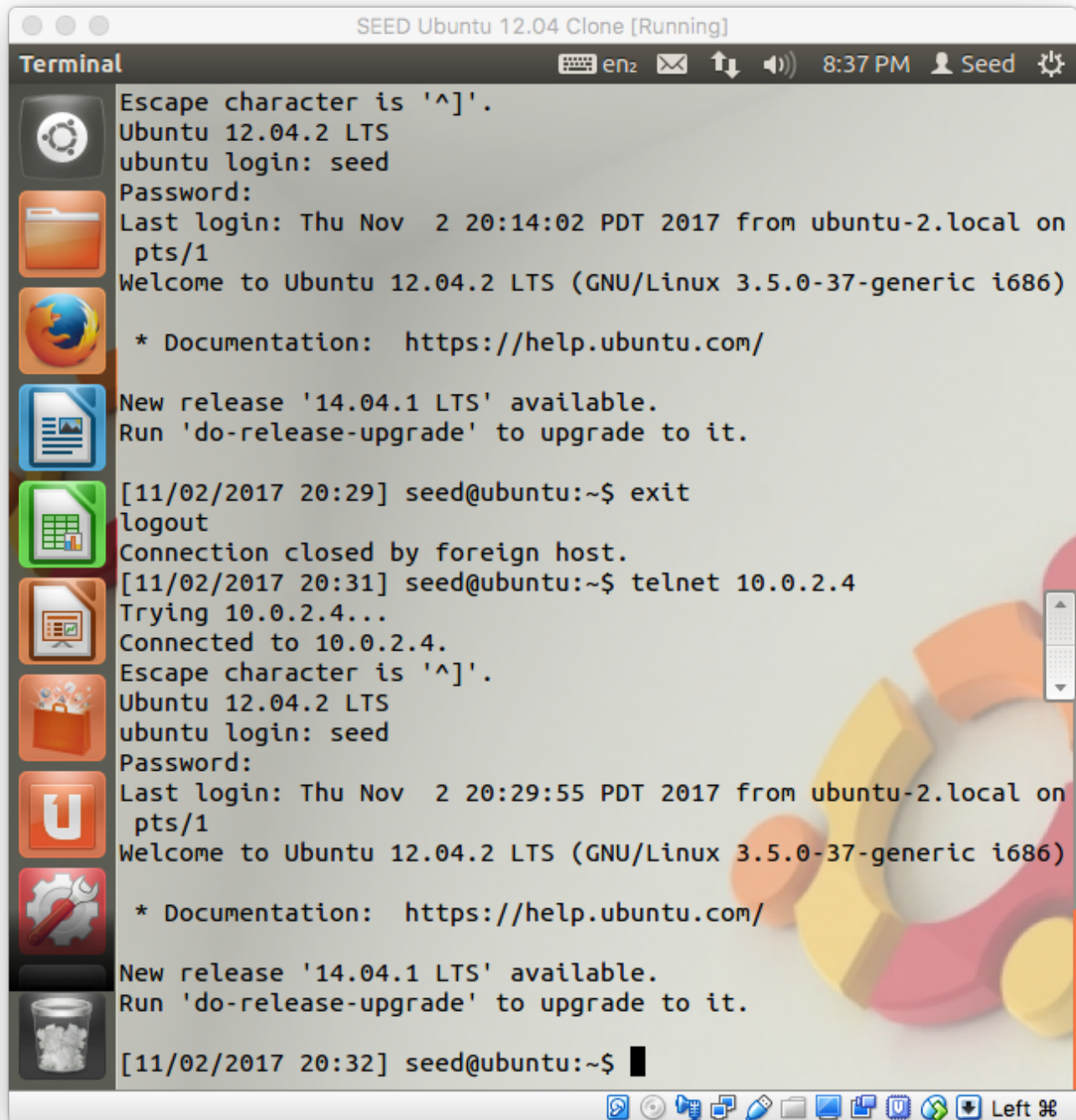
Problem 3

Below is the output for logging in with telnet, creating a file, and verifying the file's existence.



Now we do the same, but capture the packets during login with sniffex. As we can see in the below screenshots, you can see in plain text the password (sent in packets number 33-39)





```
SEED Ubuntu 12.04 Clone [Running]
Terminal en2 8:37 PM Seed

Escape character is '^]'.
Ubuntu 12.04.2 LTS
ubuntu login: seed
Password:
Last login: Thu Nov  2 20:14:02 PDT 2017 from ubuntu-2.local on pts/1
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.

[11/02/2017 20:29] seed@ubuntu:~$ exit
logout
Connection closed by foreign host.
[11/02/2017 20:31] 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:
Last login: Thu Nov  2 20:29:55 PDT 2017 from ubuntu-2.local on pts/1
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.

[11/02/2017 20:32] seed@ubuntu:~$
```

Using Wireshark, we find the password characters in the last byte of the TELNET packets of length 67 as shown below.

SEED Ubuntu 12.04 [Running]

Capturing from eth14 [Wireshark 1.6.7] 8:50 PM Seed

Filter: telnet Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Length	Info
39	2017-11-02 20:43:09.22	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
40	2017-11-02 20:43:09.22	10.0.2.4	10.0.2.5	TELNET	67	Telnet Data ..
42	2017-11-02 20:43:09.36	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
43	2017-11-02 20:43:09.36	10.0.2.4	10.0.2.5	TELNET	67	Telnet Data ..
45	2017-11-02 20:43:09.56	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
46	2017-11-02 20:43:09.56	10.0.2.4	10.0.2.5	TELNET	67	Telnet Data ..
48	2017-11-02 20:43:09.76	10.0.2.5	10.0.2.4	TELNET	68	Telnet Data ..
49	2017-11-02 20:43:09.76	10.0.2.4	10.0.2.5	TELNET	78	Telnet Data ..
51	2017-11-02 20:43:10.73	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
53	2017-11-02 20:43:10.86	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
55	2017-11-02 20:43:11.02	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
57	2017-11-02 20:43:11.13	10.0.2.5	10.0.2.4	TELNET	67	Telnet Data ..
59	2017-11-02 20:43:12.01	10.0.2.5	10.0.2.4	TELNET	68	Telnet Data ..

Frame 51: 67 bytes on wire (536 bits), 67 bytes captured (536 bits)

Ethernet II, Src: CadmusCo_16:95:bb (08:00:27:16:95:bb), Dst: CadmusCo_fb:cc:53 (08:00:27:fb:cc:53)

Internet Protocol Version 4, Src: 10.0.2.5 (10.0.2.5), Dst: 10.0.2.4 (10.0.2.4)

Transmission Control Protocol, Src Port: 38742 (38742), Dst Port: telnet (23), Seq: 12, Ack: 151, Len: 67

```

0000  08 00 27 fb cc 53 08 00 27 16 95 bb 08 00 45 10  ..'.S.. '.....E.
0010  00 35 be 7b 40 00 40 06 64 2f 0a 00 02 05 0a 00  .5.{@.@. d/.....
0020  02 04 97 56 00 17 9d 3b ad dc 62 ef 77 36 80 18  ...V...; ..b.w6..
0030  00 e5 3a 01 00 00 01 01 08 0a 00 13 98 d7 00 13  ....
0040  6a 1c 64                                           j.

```

eth14: <live capture in progress> ... Packets: 135 Displayed: 36 Marked: 0 Profile: Default

This plaintext password communication is an obvious concern, as using telnet means potentially giving an attacker remote root access to your machine.

Problem 4

If we log in with SSH, we can see SSH traffic, however the password is clearly encrypted. Instead of plaintext bytes being sent, we see several key exchange packets along with other encrypted traffic. This is obviously a much more secure method of communication.

The image shows a dual-monitor setup. The left monitor displays Wireshark 1.6.7 capturing traffic on the eth14 interface. The filter is set to 'ssh'. The packet list shows several SSHv2 packets between 10.0.2.5 and 10.0.2.4, including Server Protocol, Client Protocol, Key Exchange Init, and New Keys. The packet details pane shows the raw bytes of the selected packet.

The right monitor displays a terminal window titled 'Terminal' on a 'SEED Ubuntu 12.04 Clone [Running]'. The terminal shows the output of the command 'ssh seed@10.0.2.4'. The output includes a warning about the host's authenticity, the ECDSA key fingerprint, a confirmation to continue connecting, and the password prompt. The password is entered and the user is welcomed to Ubuntu 12.04.2 LTS. The terminal also shows the last login time and the user's prompt.