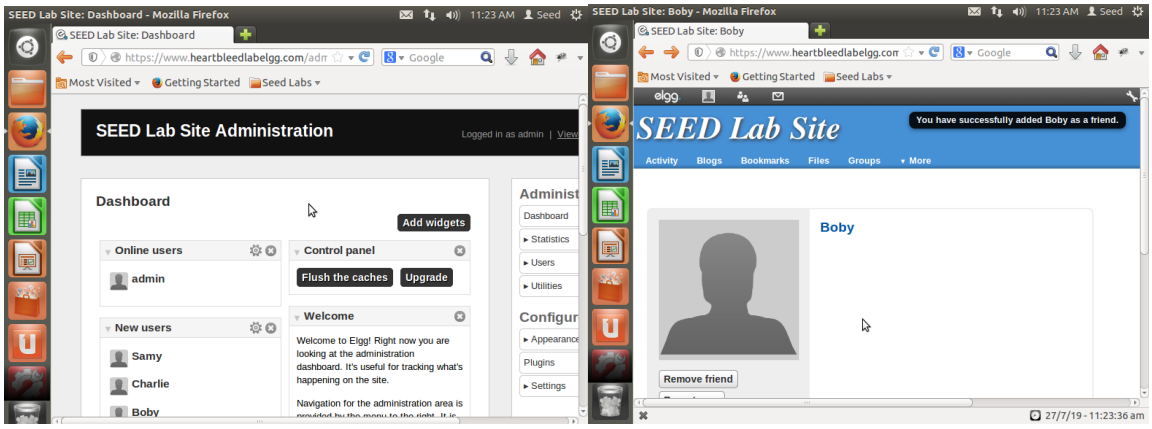
Lab 5: Heartbleed Attack Lab

Initially setting up the lab with nat-network and cloning VM in order to have both attacker and defender running at the same time. I pinged the defender VM from attacker. Ping successful.

Changed to root user and modified /etc/hosts. Set the ip address of www.heartbleedlabelgg.com to that of Defender ip address.

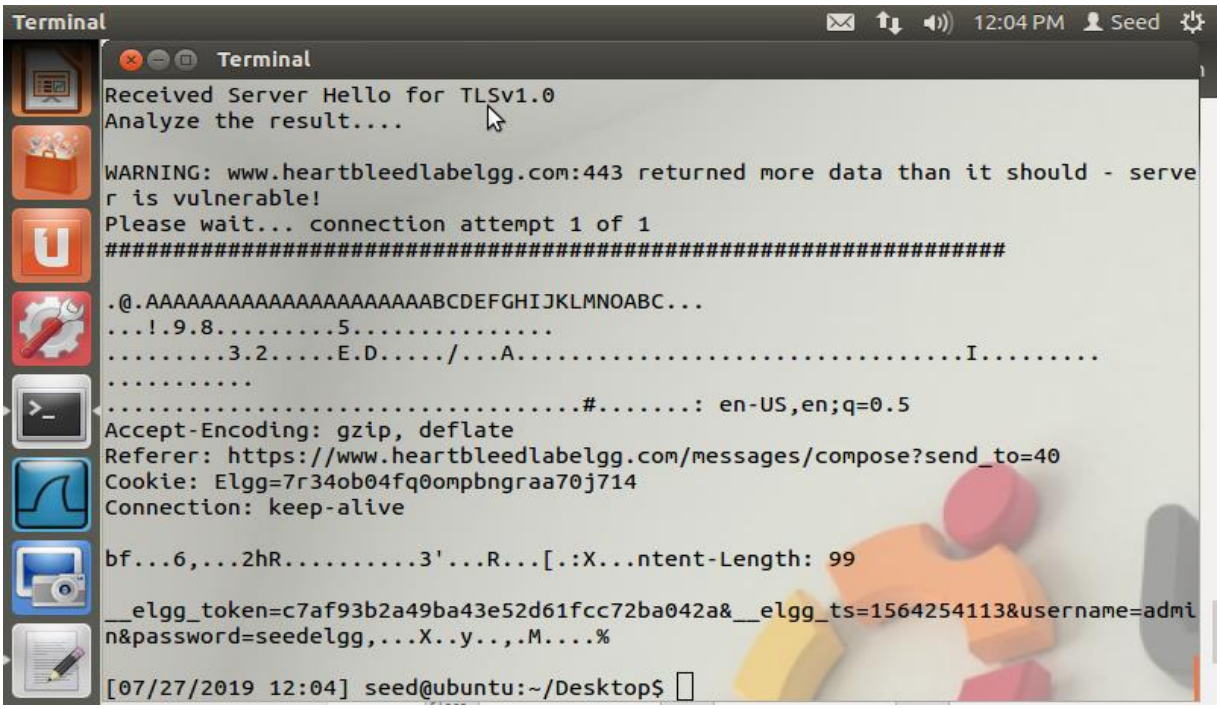
3.1 Task 1: Launch the Heartbleed Attack

I logged into admin account on heartbleedlabelgg.com, sent boby a friend request, and sent boby a message.



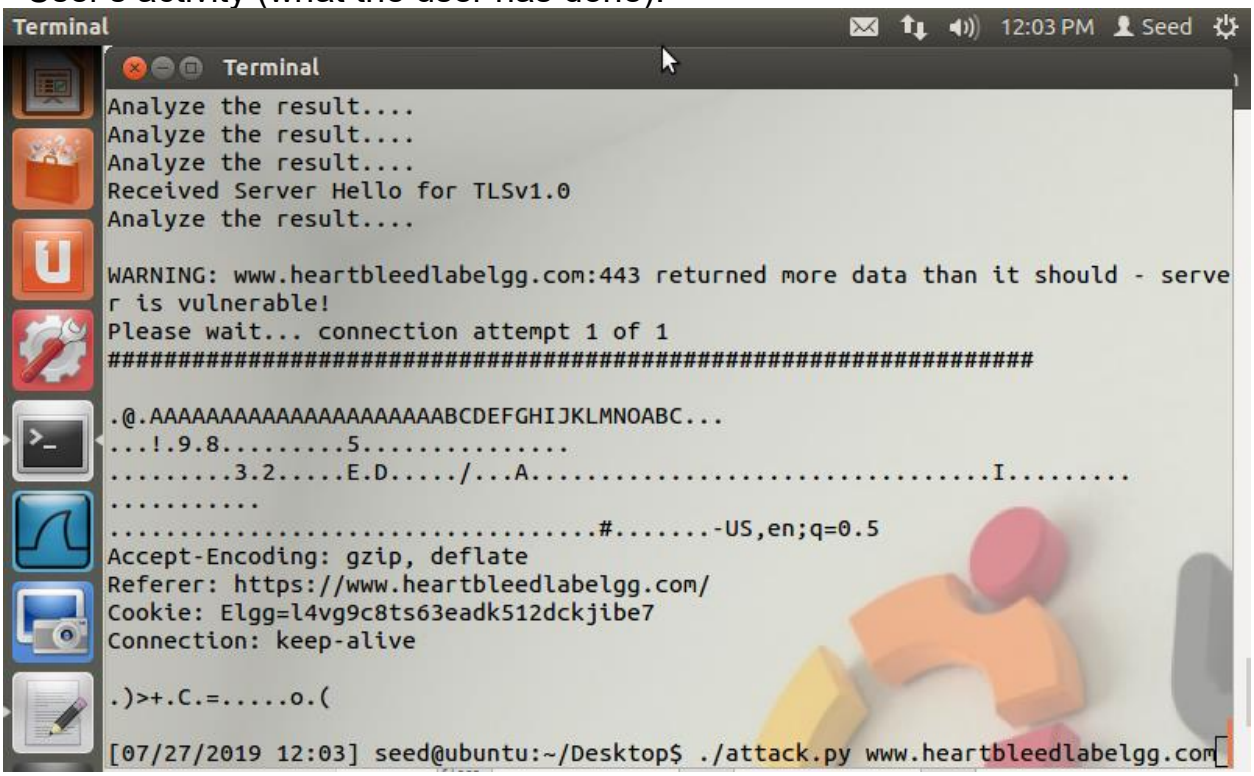
Going back to attack VM, I already have attack.py ready and entered the command ./attack.py www.heartbleedlabelgg.com. Entering the command a few times and finally I can see admins username and password as well as the message sent to boby.

- User name and password.

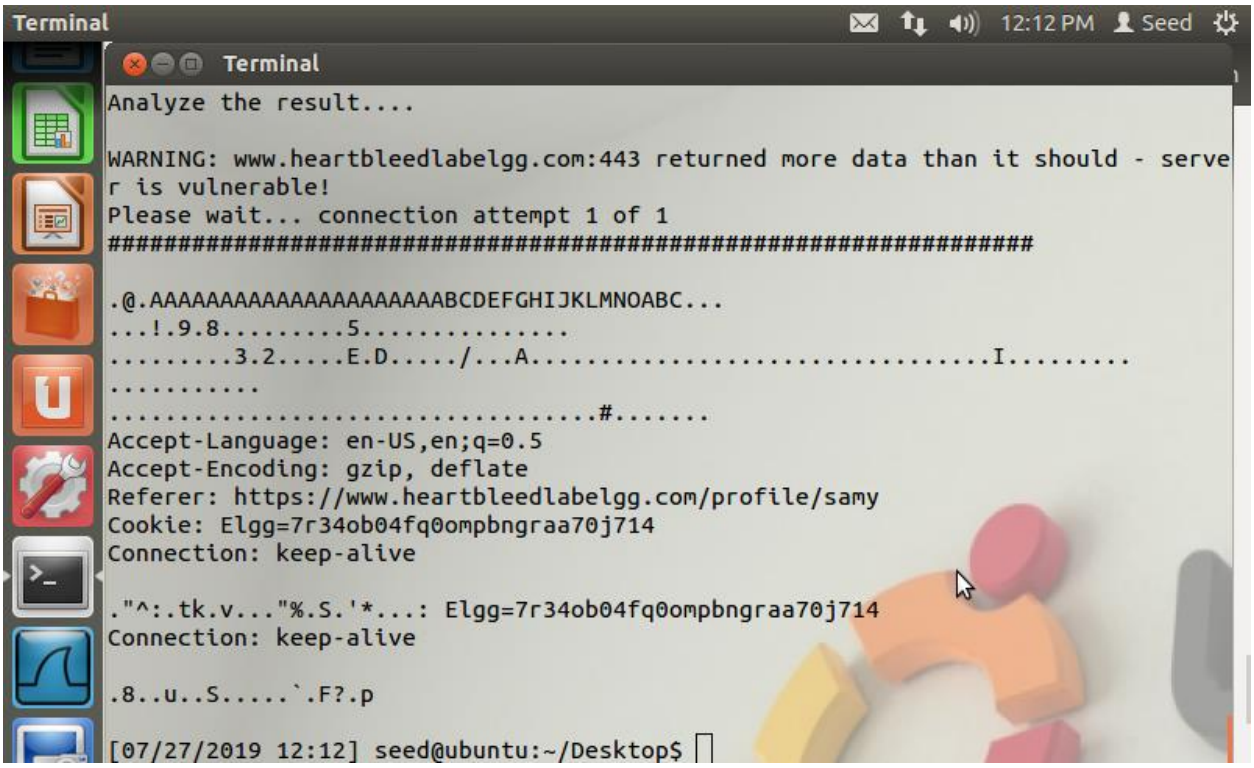


-Here you can see the username and password

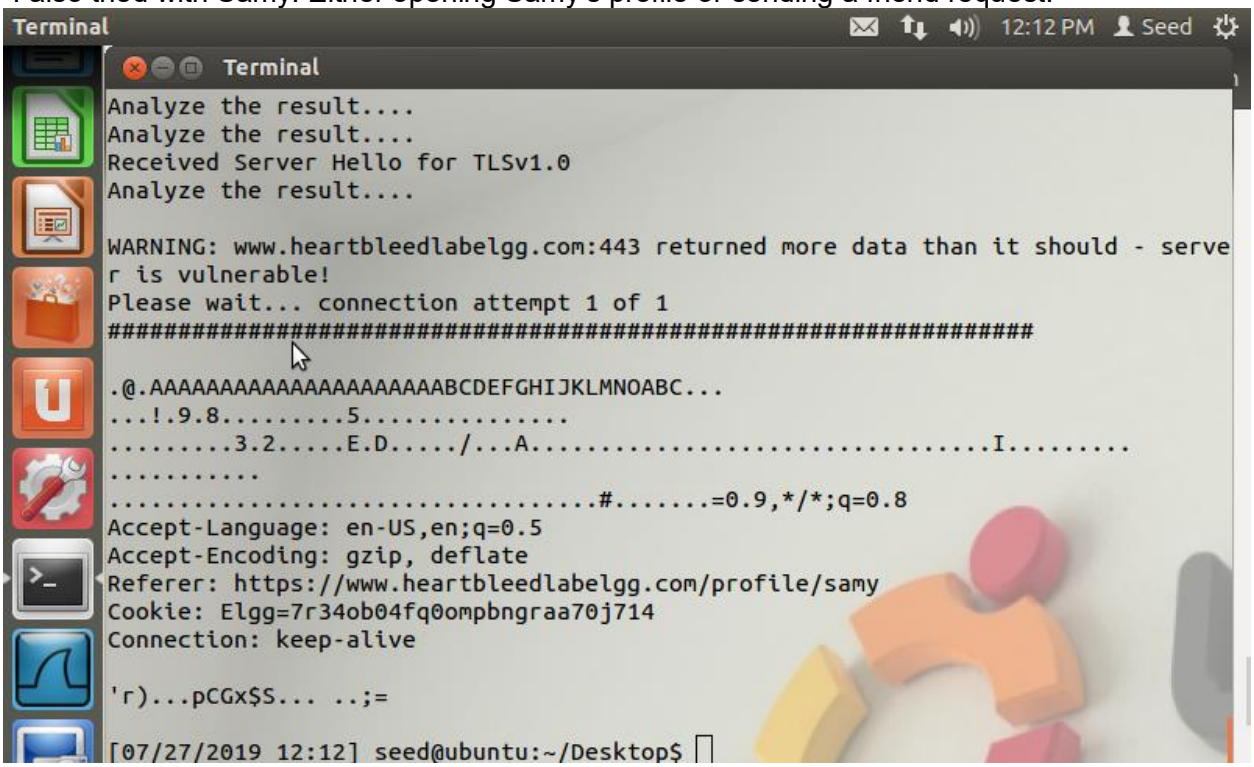
• User's activity (what the user has done).



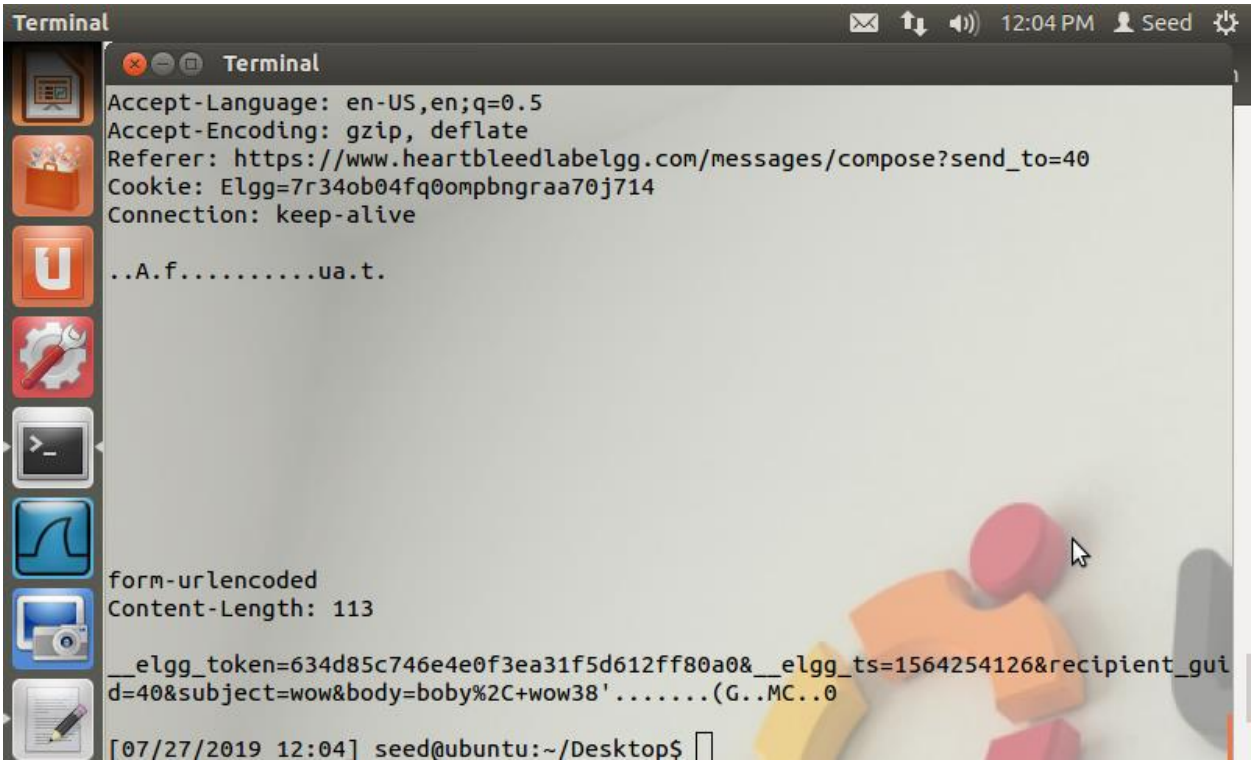
-I believe this is a friend request.



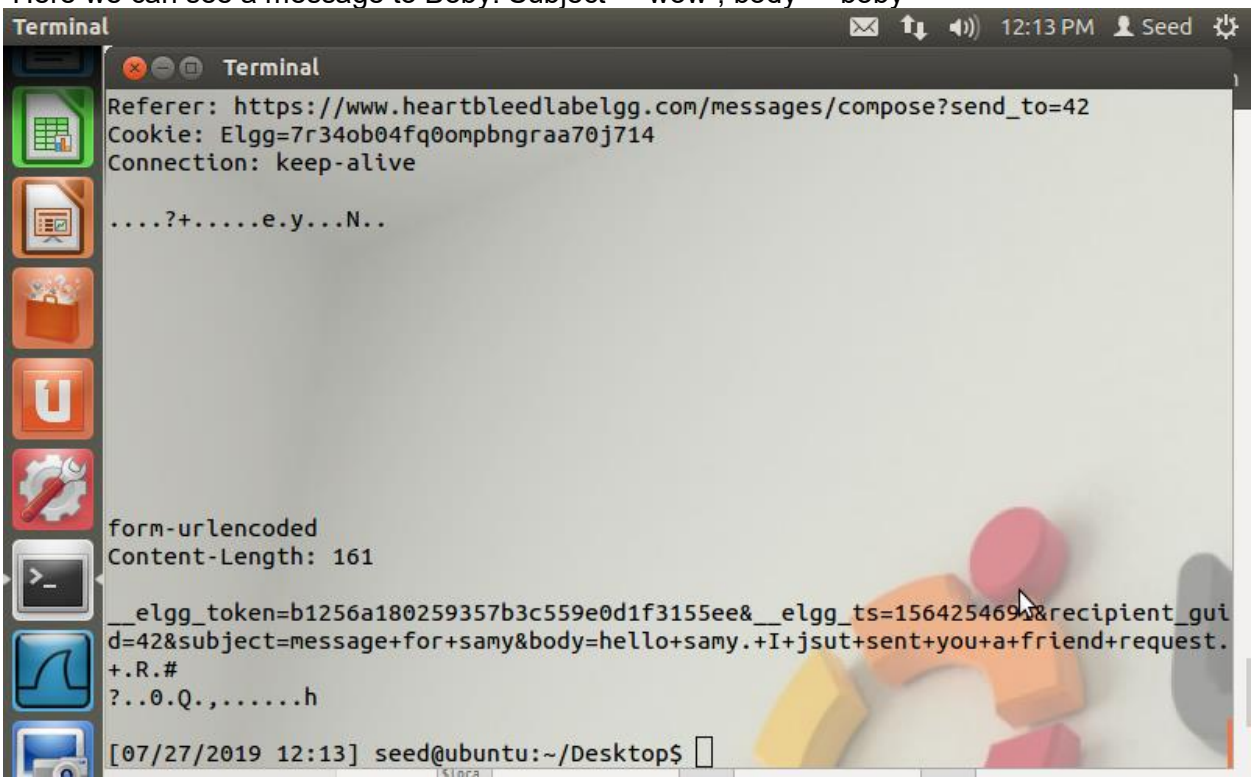
-I also tried with Samy. Either opening Samy's profile or sending a friend request.



• The exact content of the private message.



-Here we can see a message to Boby. Subject = “wow”, body = “boby”

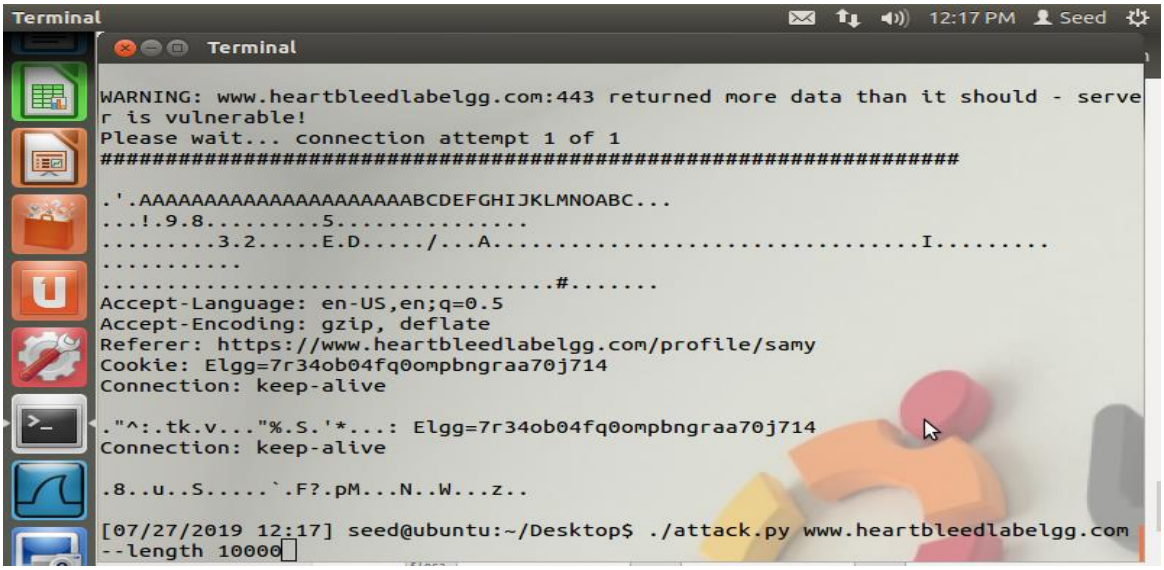


-Message to samy with subject = message, and body = “hello samy. I just sent you a friend request”.

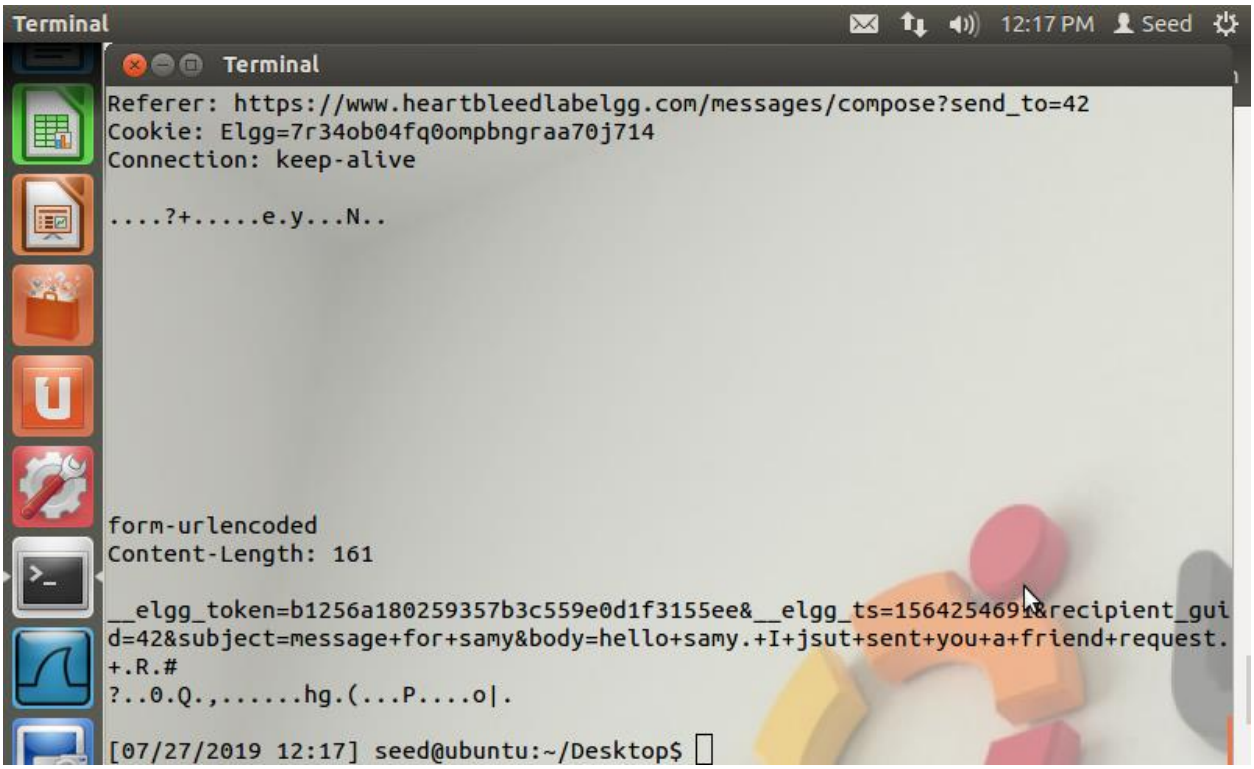
3.2 Task 2: Find the Cause of the Heartbleed Vulnerability

Question 2.1

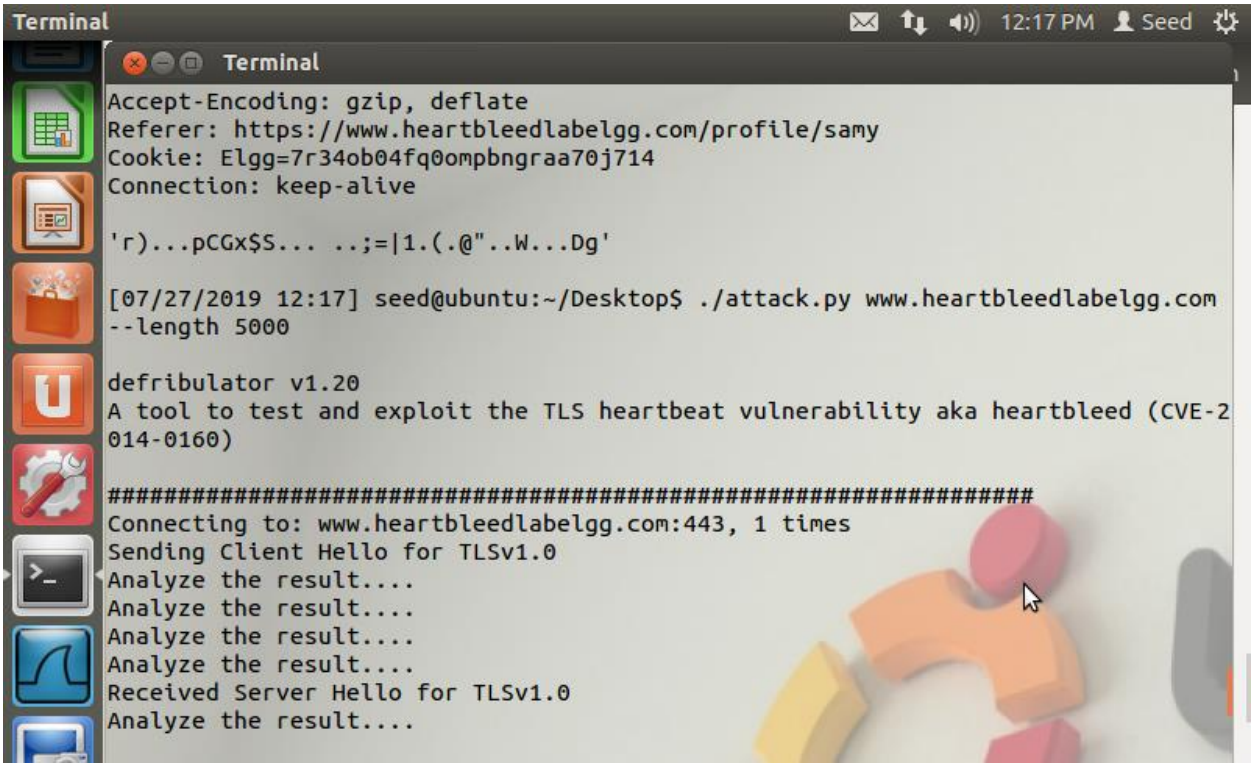
As I decreased length the information that was being retrieved also decreased. Length started at ~16000 in length. I went down to 10000 and was still receiving information. I then went down to 5000 and didn't notice any loss of the useful information I was retrieving. 5000 and 1000 also returned useful information. By the time I was using 500 as the length, I was still retrieving useful information, but I did not see the login information or the body of private messages. I could still see information about URL visited, and cookies. When I used 100 as the length no useful information was retrieved. For this task I stopped at length 100.



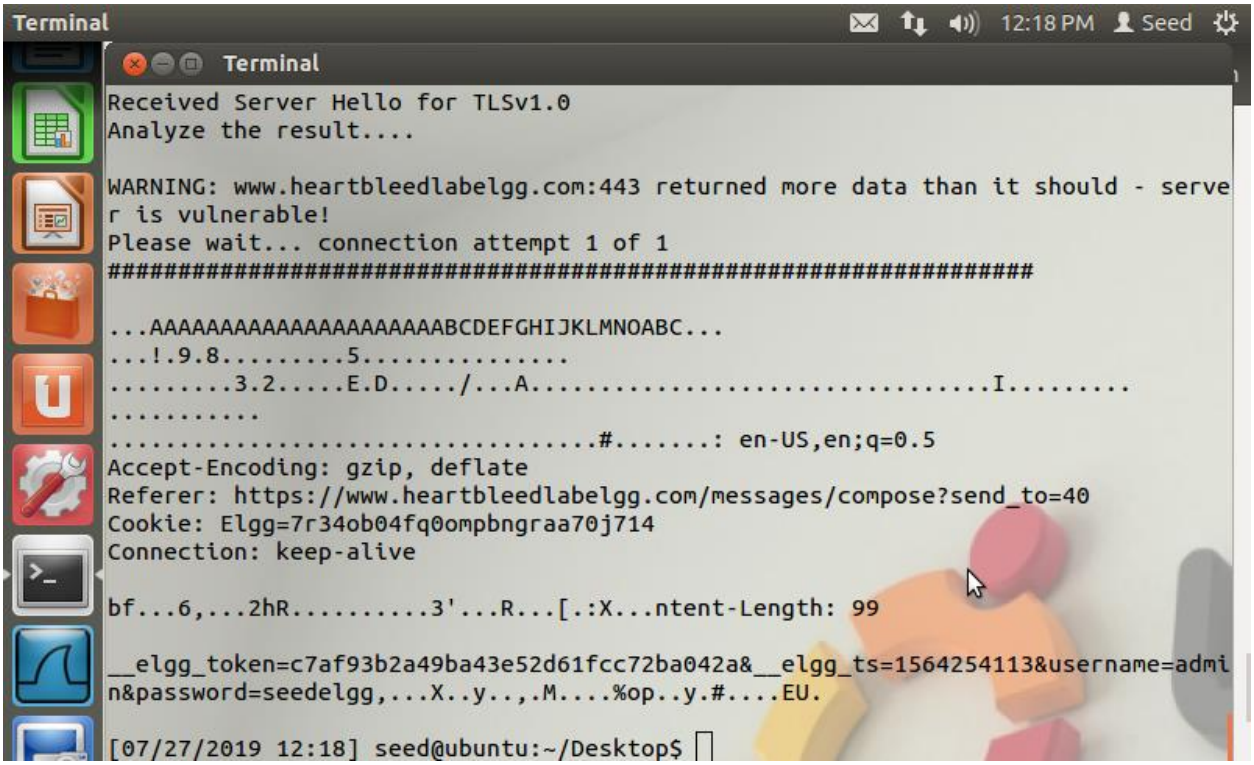
-Length 10000



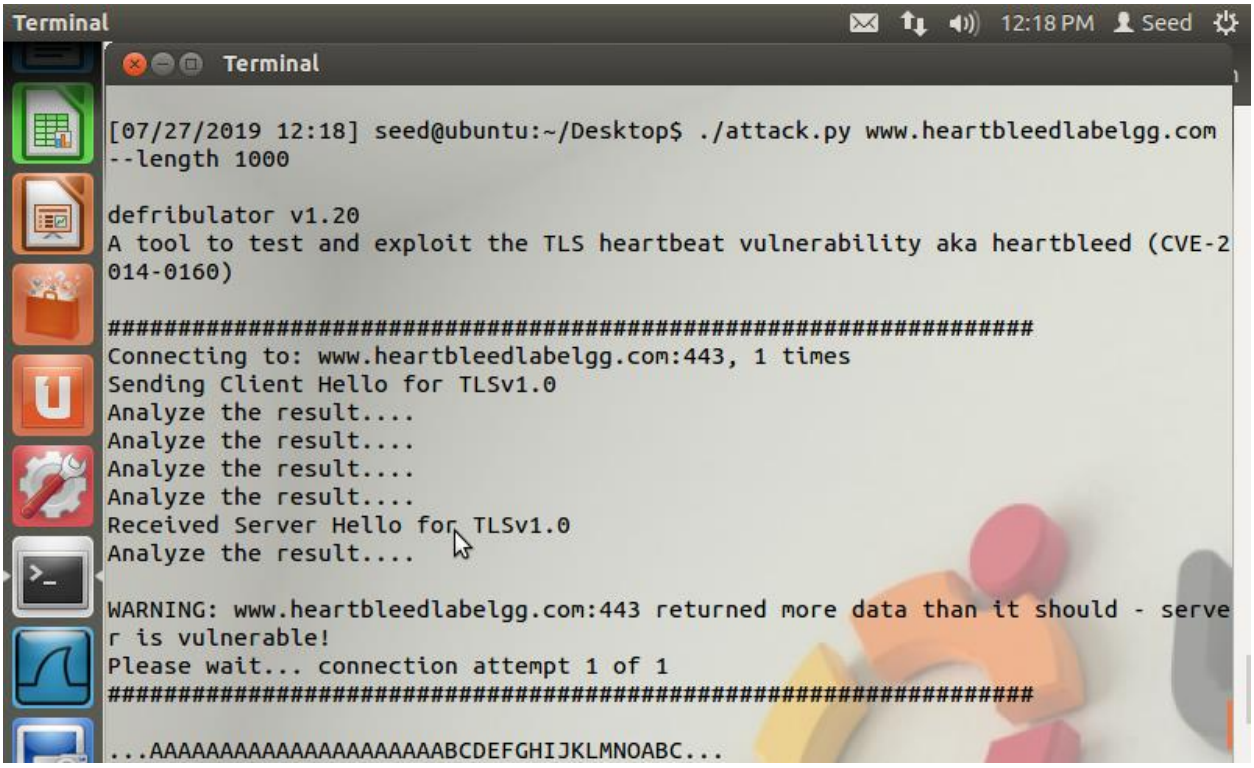
-Length 10000



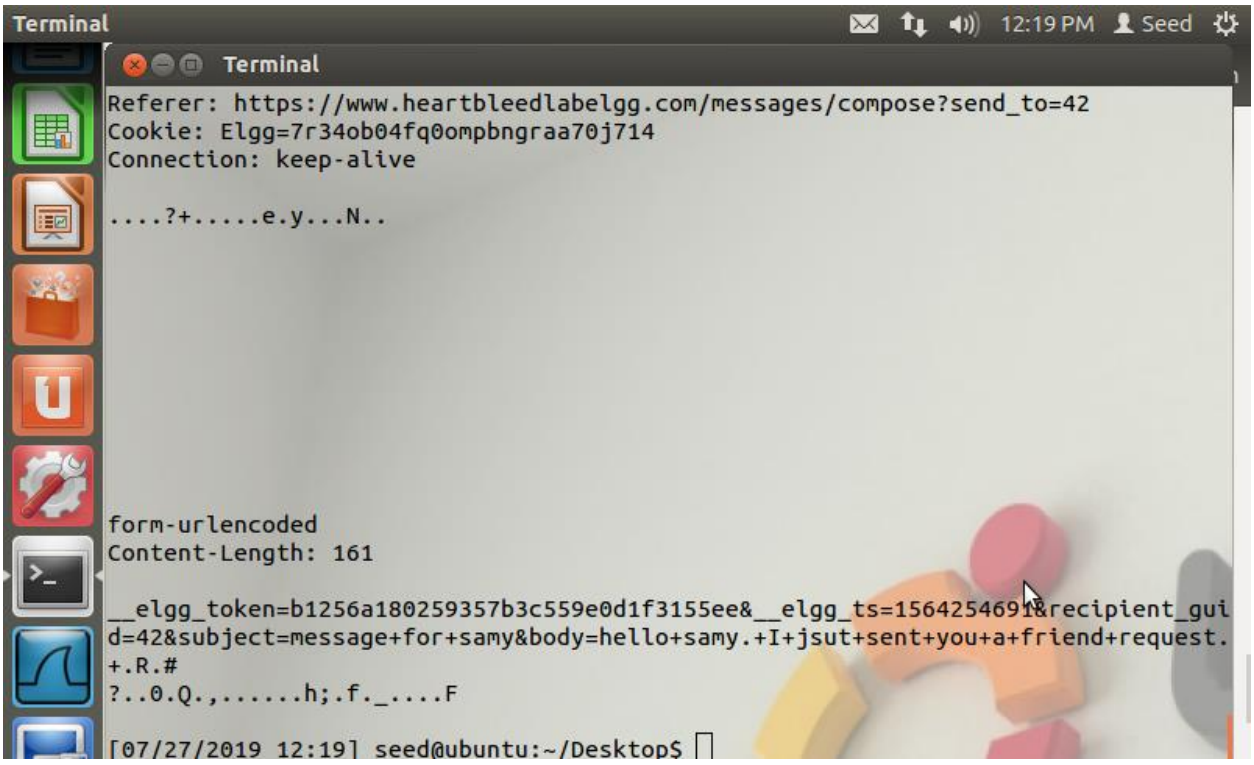
-Length 5000



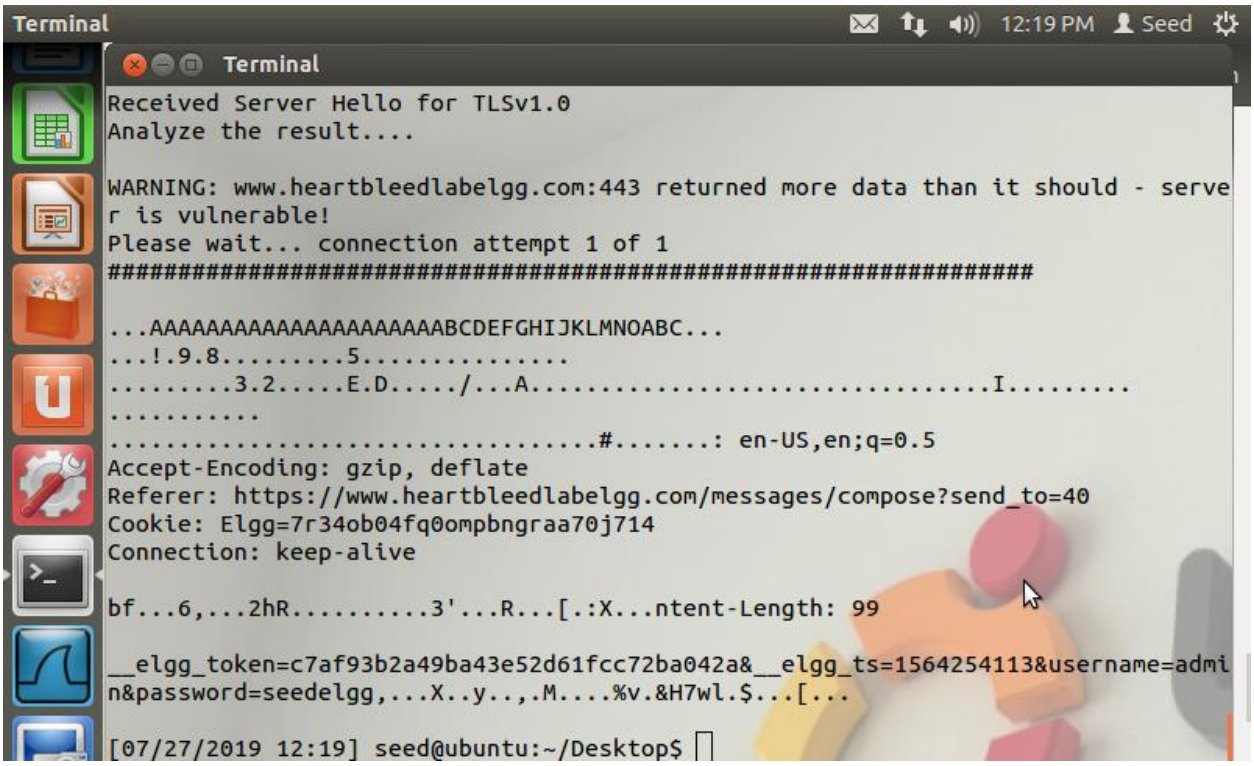
-Length 5000



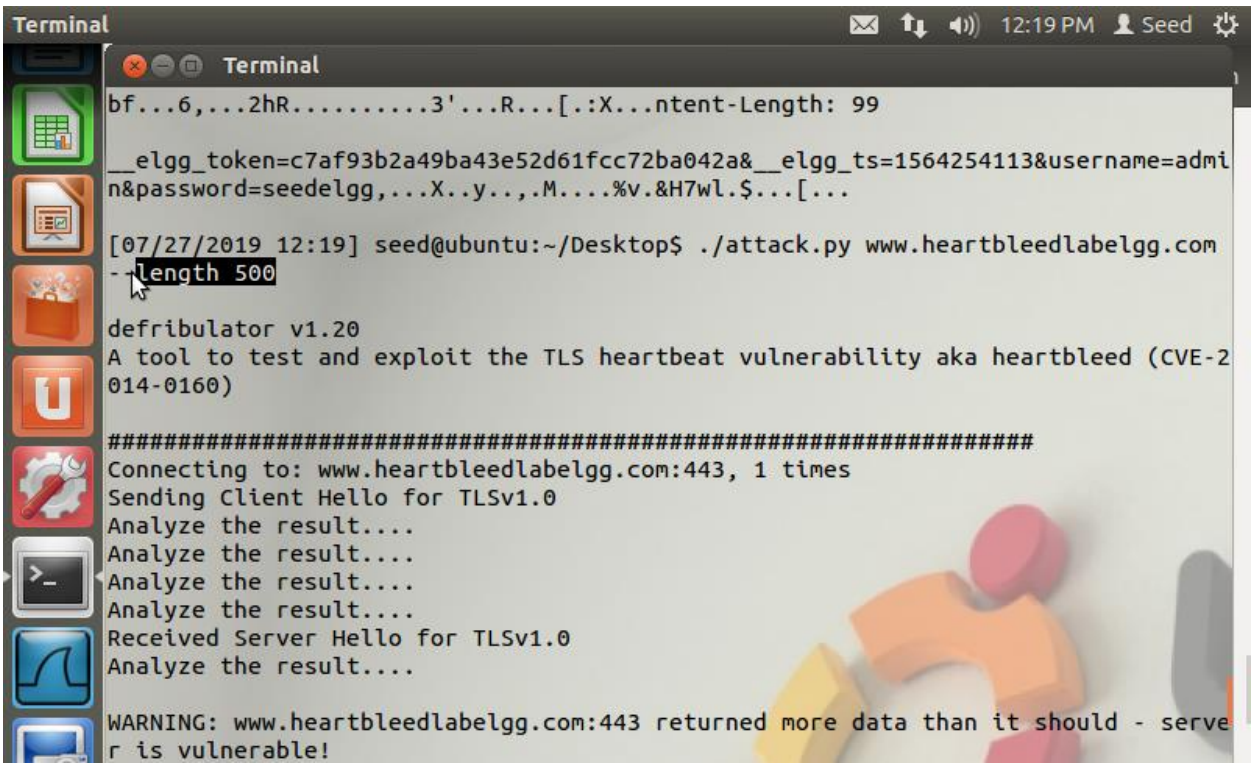
-length 1000



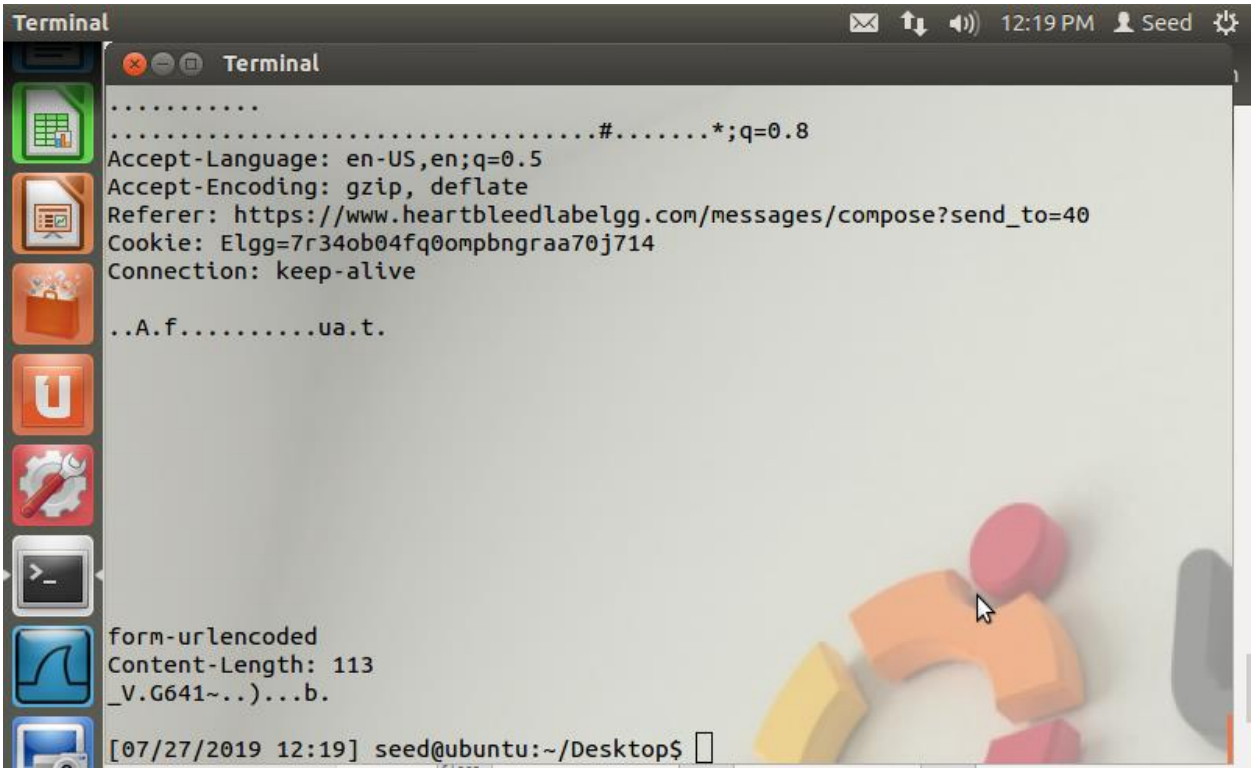
-Length 1000. Can still see contents of a private message



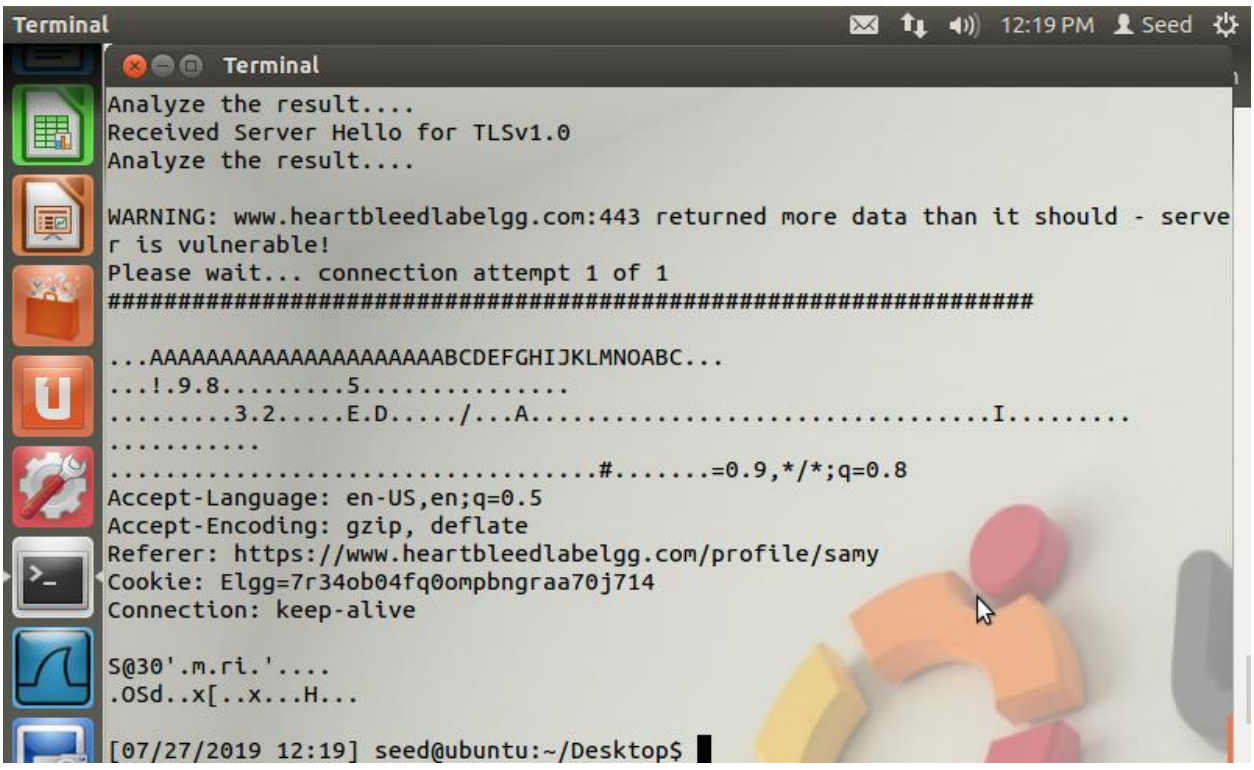
-length 1000. Can still see username and password



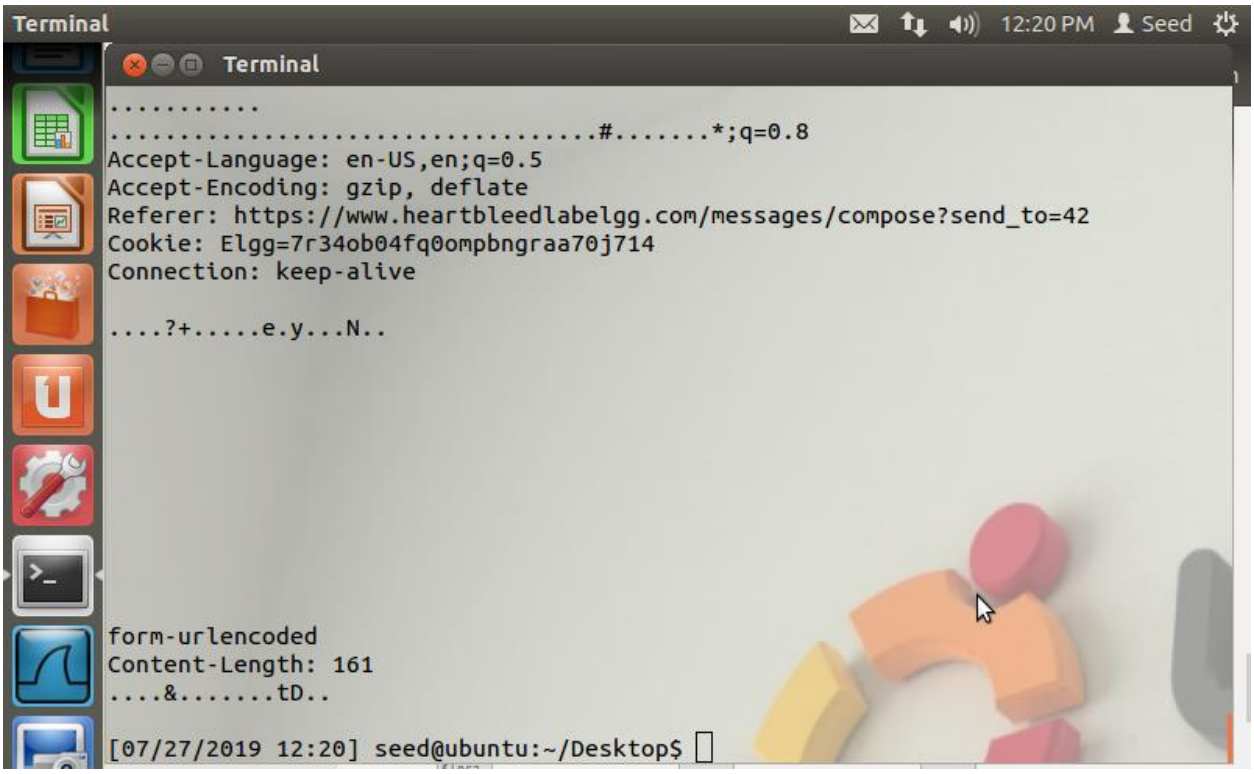
-Now checking with length 500



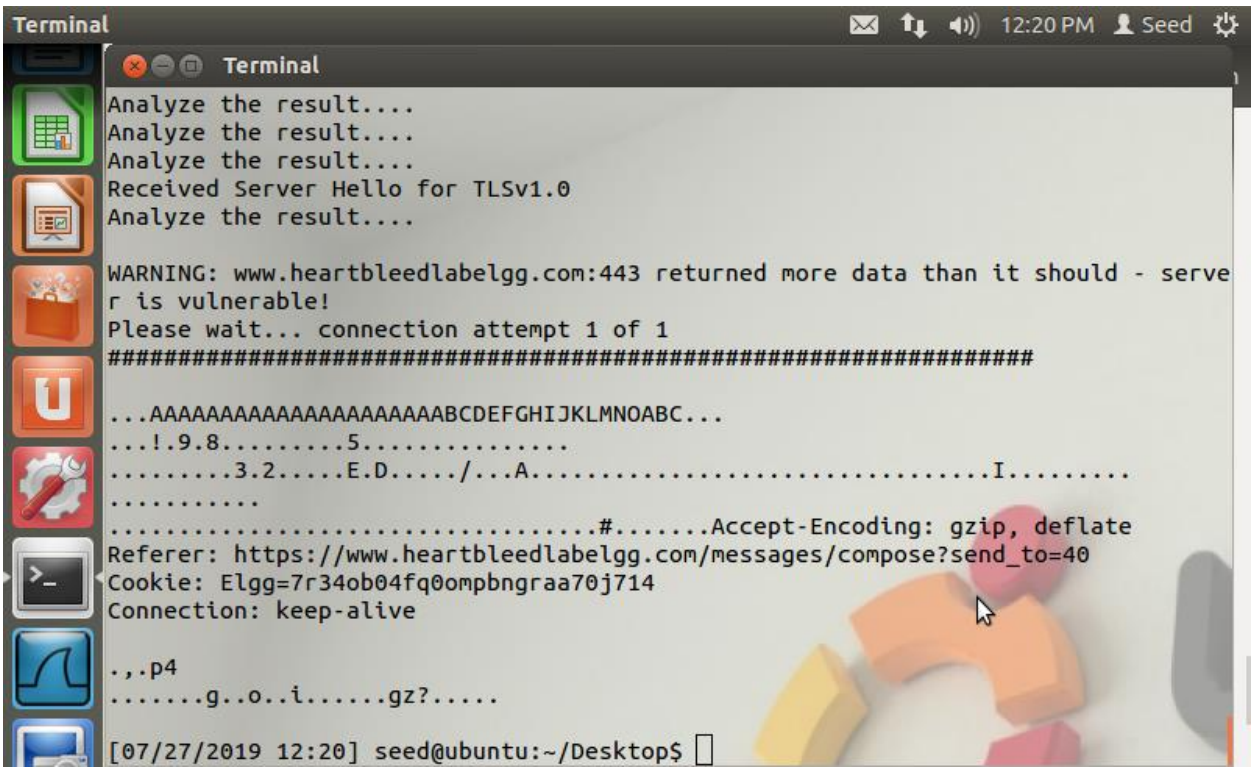
-Some information with length 500.



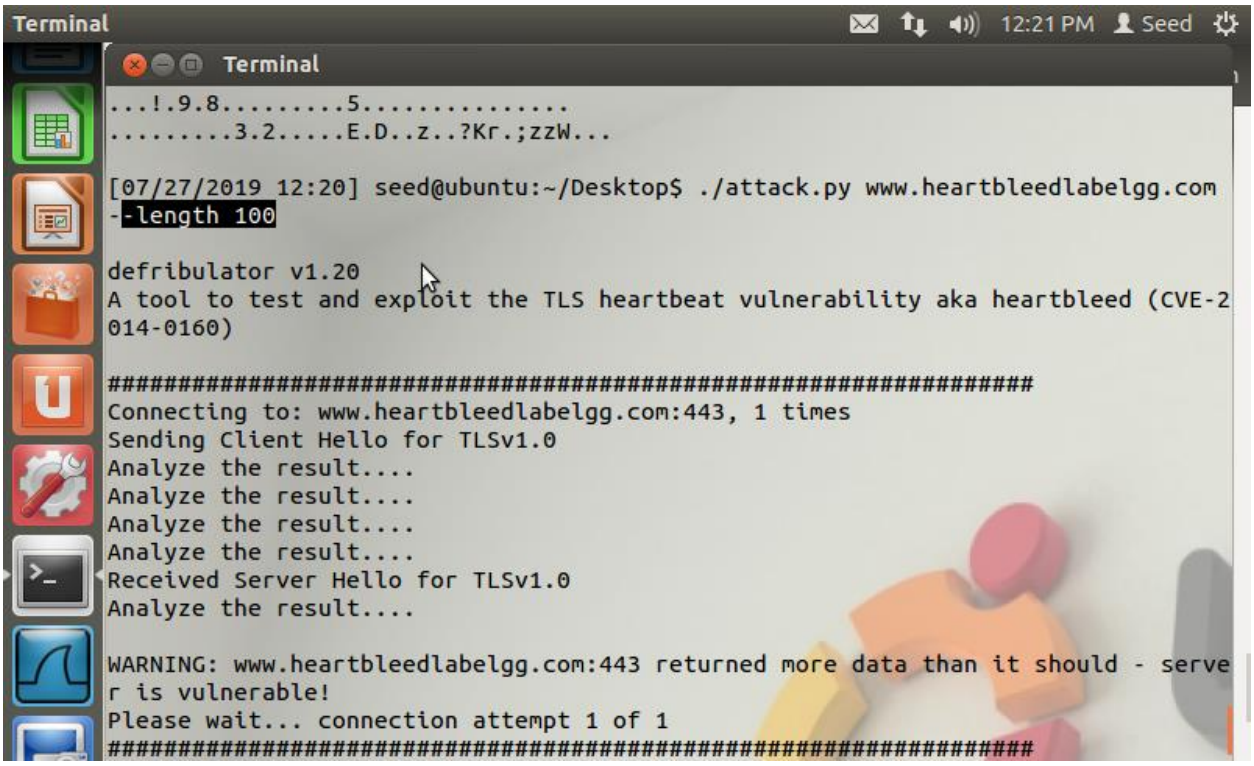
-length 500 still



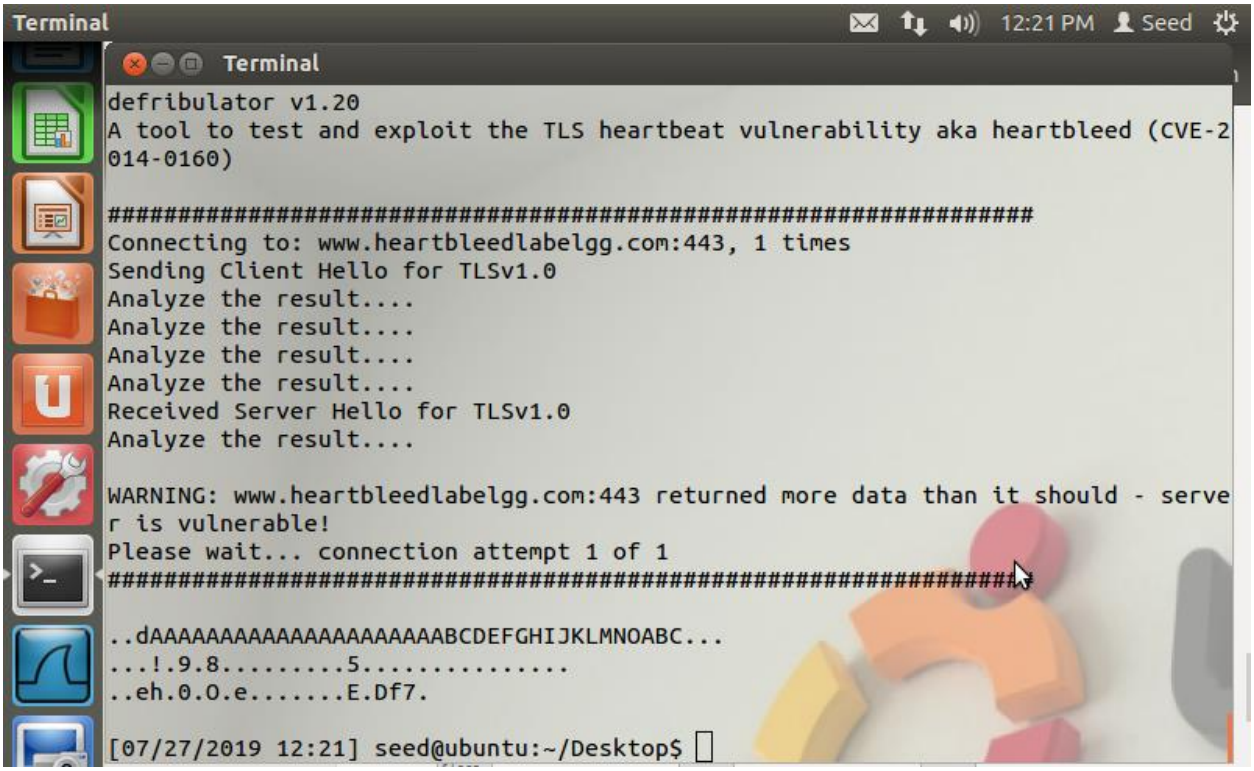
-Length 500. Can see that is sending a message, but not contents of message.



-Length 500, again can not see the contents of message.



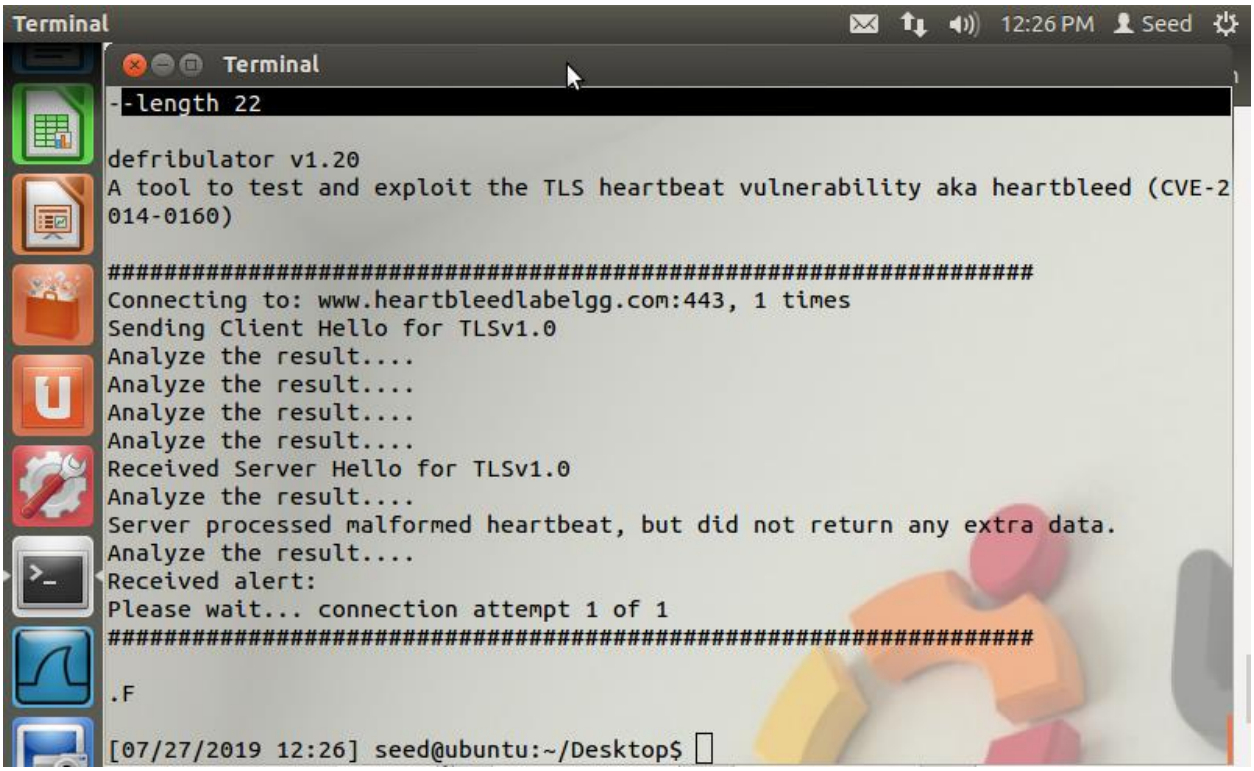
-Length 100 now.



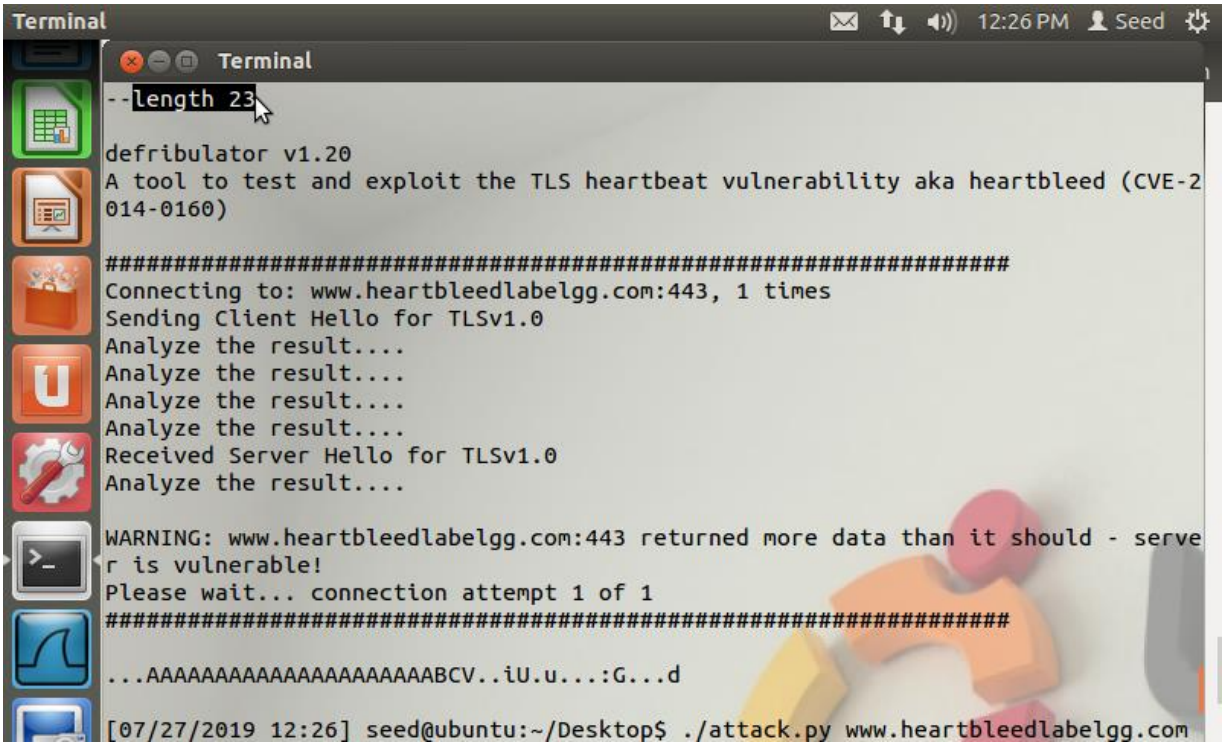
-length 100. No useful information.

Question 2.2:

Length 22 was the largest length in which I receiver the message “Server processed malformed heartbeat but did not return any extra data.”



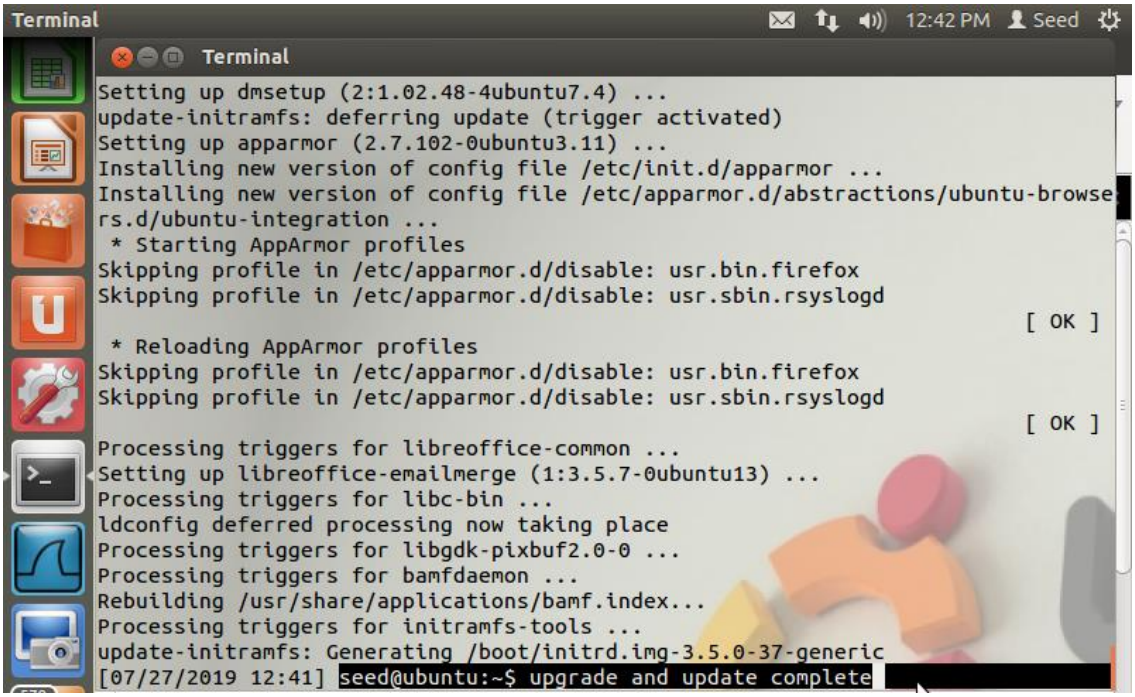
Length 23 was the smallest length in which I received the message “WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!”



3.3 Task 3: Countermeasures and Bug Fix

Task 3.1

After the update and upgrade no vulnerability was exploited by the heartbeat attack.



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....

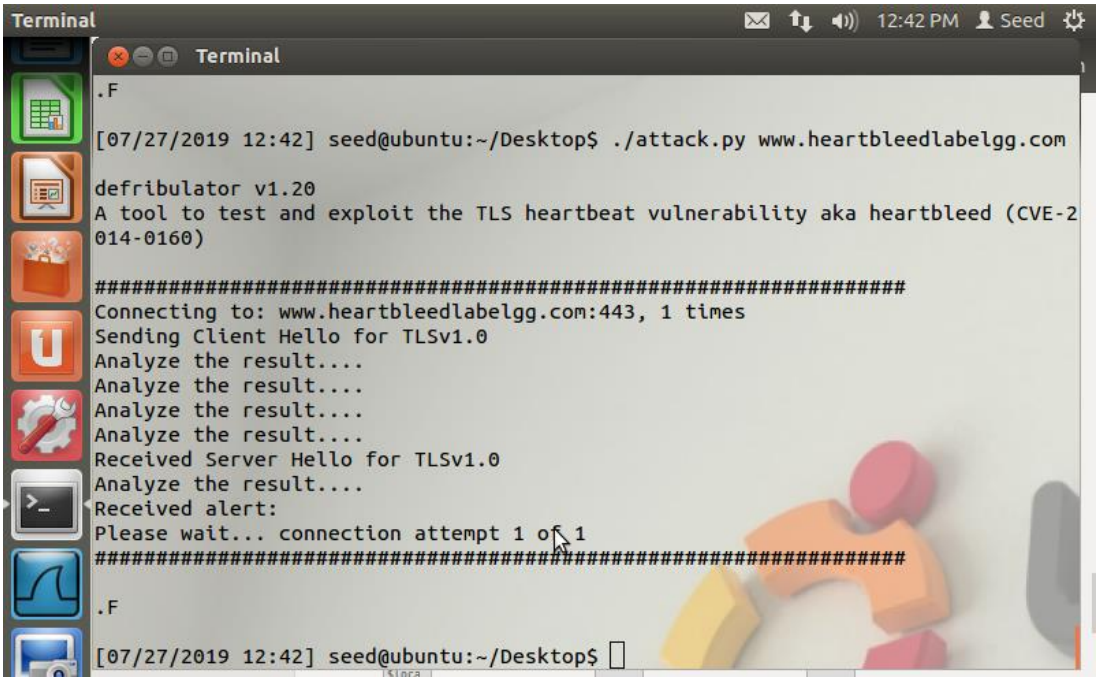
Received alert:

Please wait... connection attempt 1 of 1

##

.F

This is what was observed. No message saying that vulnerability was NOT detected, but no message saying there was a vulnerability was detected. No exploits noticed.



Task 3.2

The problem is the memcpy doesn't have any kind of bound checking. We want to check that "1+2+payload+padding" is the actual size it says it is, and that it isn't some large number. I'm not sure what the exact best size would be, but from a task above, we saw that with length 22 was ineffective, so that could be an approximate upper bound. Or maybe check the size of 1+2+payload and if the size being returned from memcpy is larger than that we do not return contents from memcpy. Bound checking is what we want to do.

Alice is right in her assumption that the problem is that there is no bound checking. In regard to Bob saying we need to user input validation. The normal function of heartbeat is to check if the connection is still active, and typically there is no user input. In a way the heart bleed attack sends a much larger length than the heartbeat message (payload). Eva thinks we could just delete length. That may fix the exploit to some extent, but not vulnerability that heartbeat exploited. We still want to ensure that a buffer overflow isn't being exploited with memcpy.