CSCI 400 - Lab 10

Prof. Faheem Abdur-Razzaaq

November 22, 2019

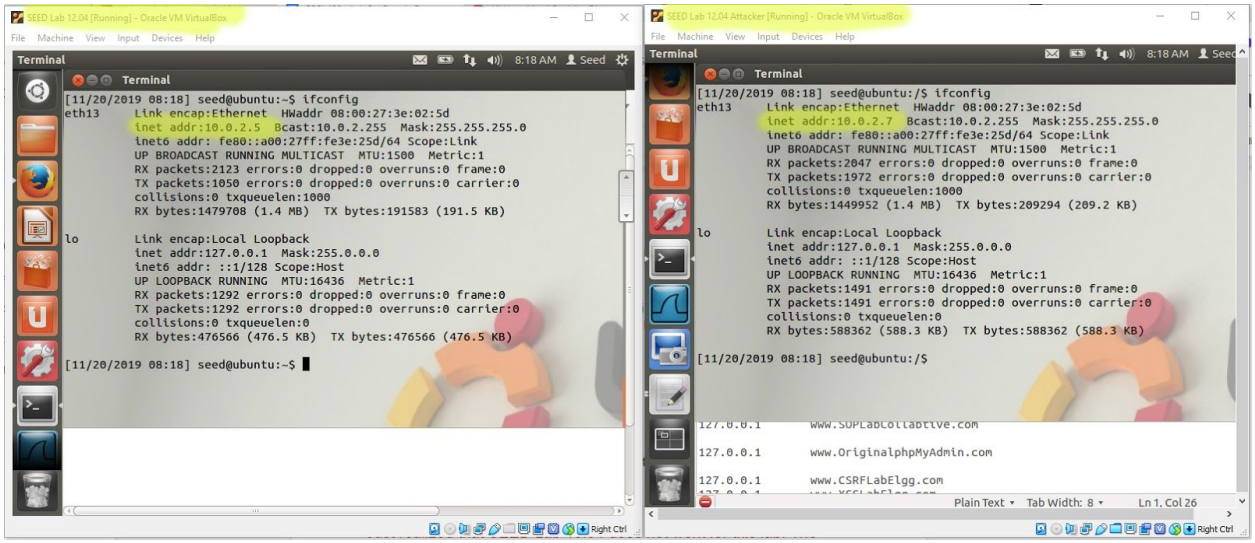
(Priya Thapa, Emranul Hakim, Lakpa Sona Sherpa, Corneliu Raul Nistor)

Heartbleed Attack Lab

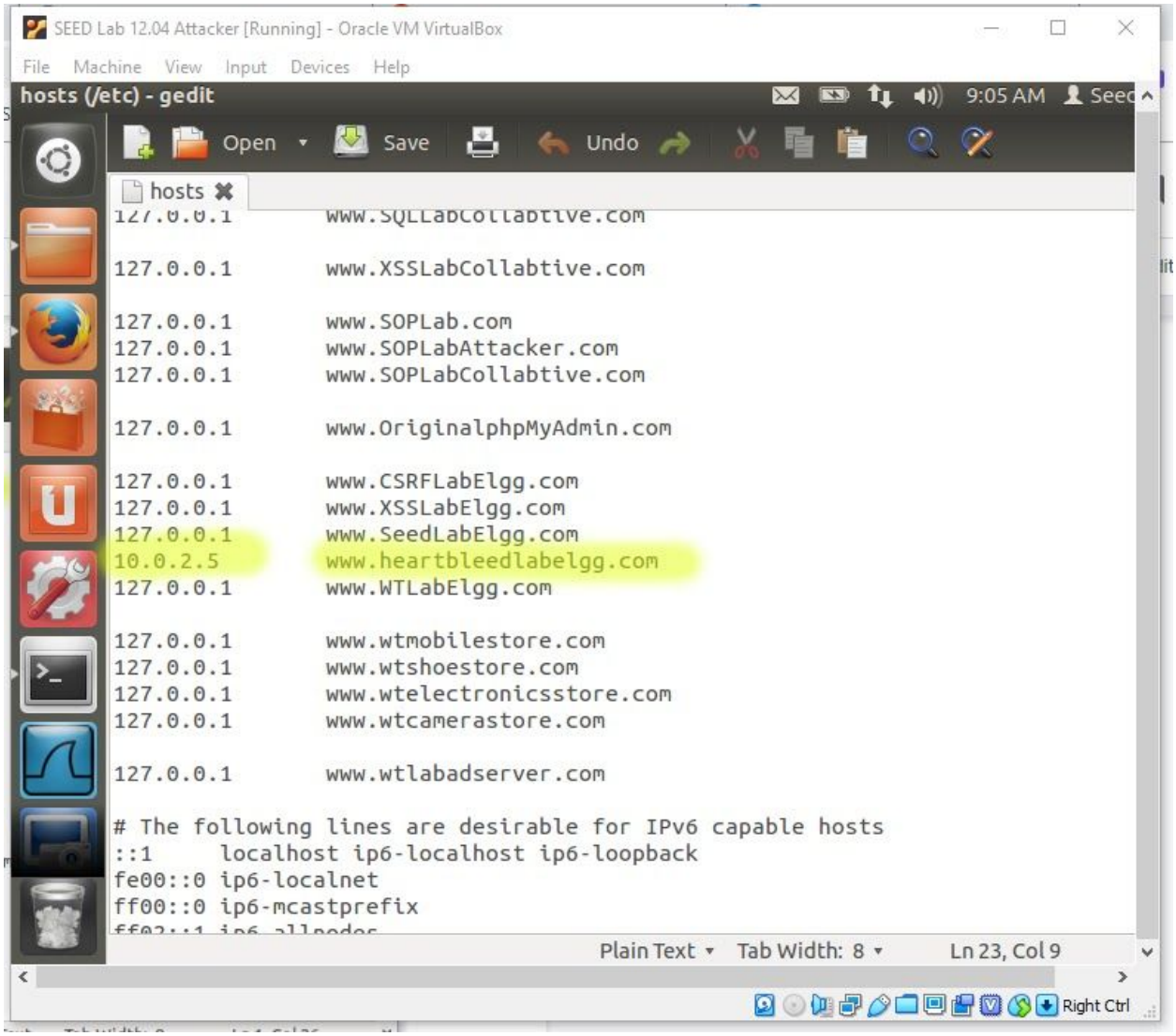
3.1 Task 1: Launch the Heartbleed Attack

As the Lab indicates we have setup two VM machines. One the Attacker and the other as the Victim Server. The Lab tells us to configure both VMs to Nat Network, and as a result we will have the following IP addresses

Attacher: 10.0.2.7
Victim Server: 10.0.2.5

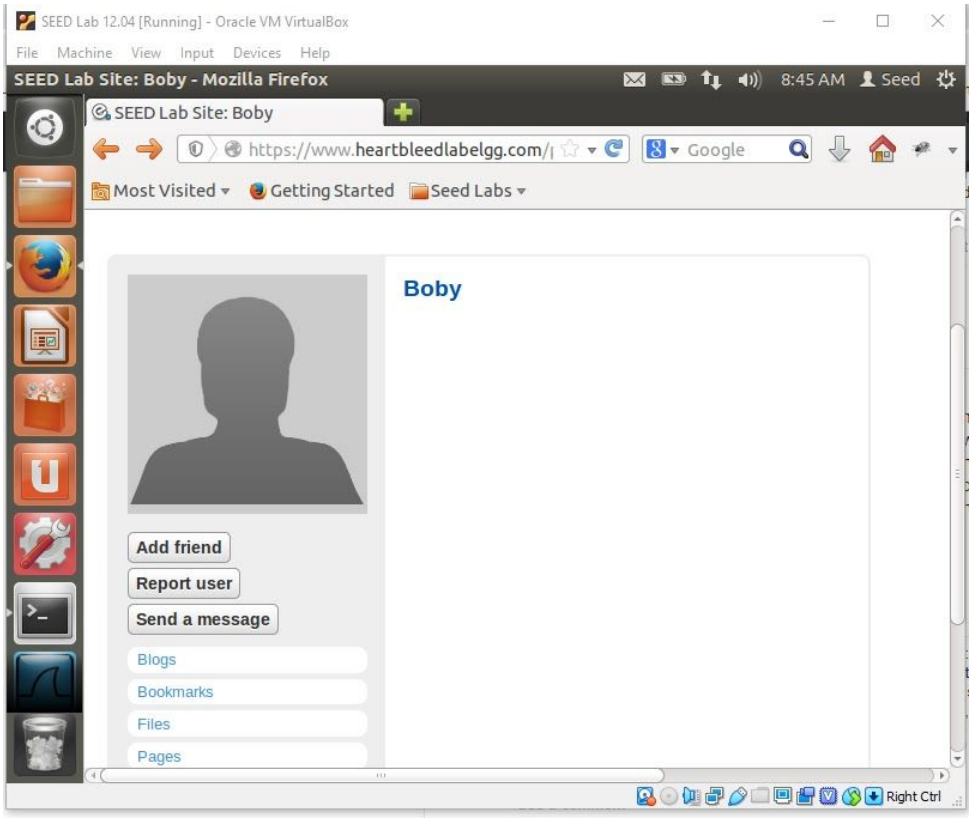


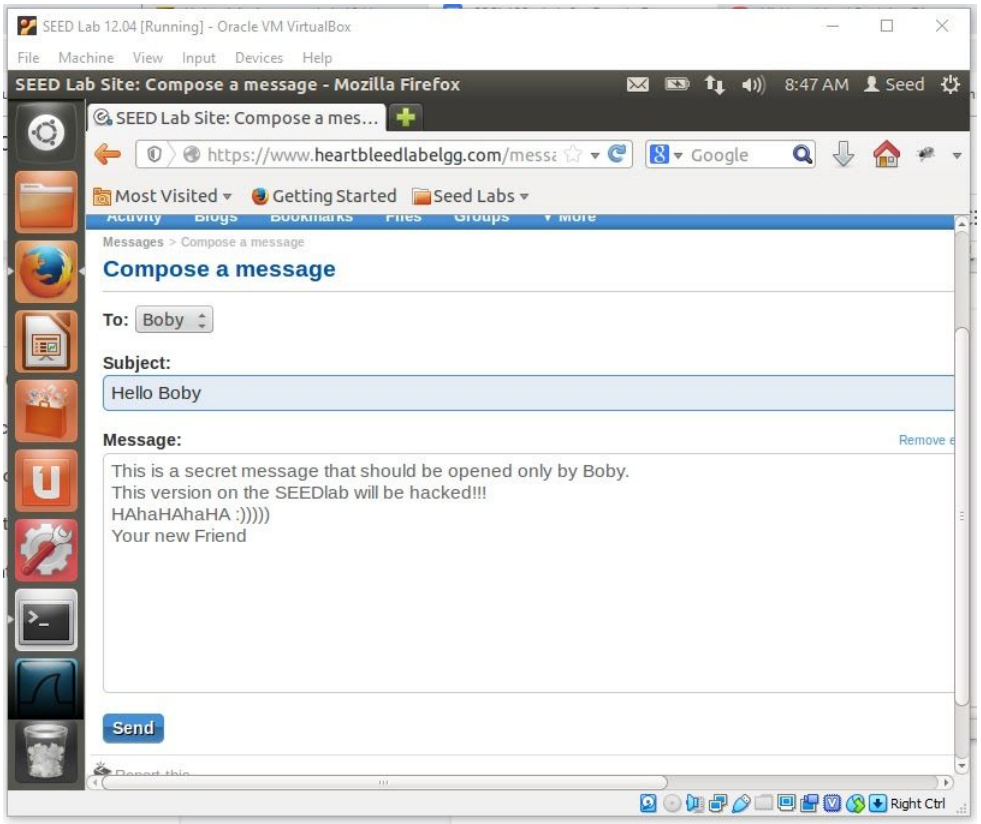
We also have to modify the **etc/hosts** file on the attacker machine with the IP of the victim's server:



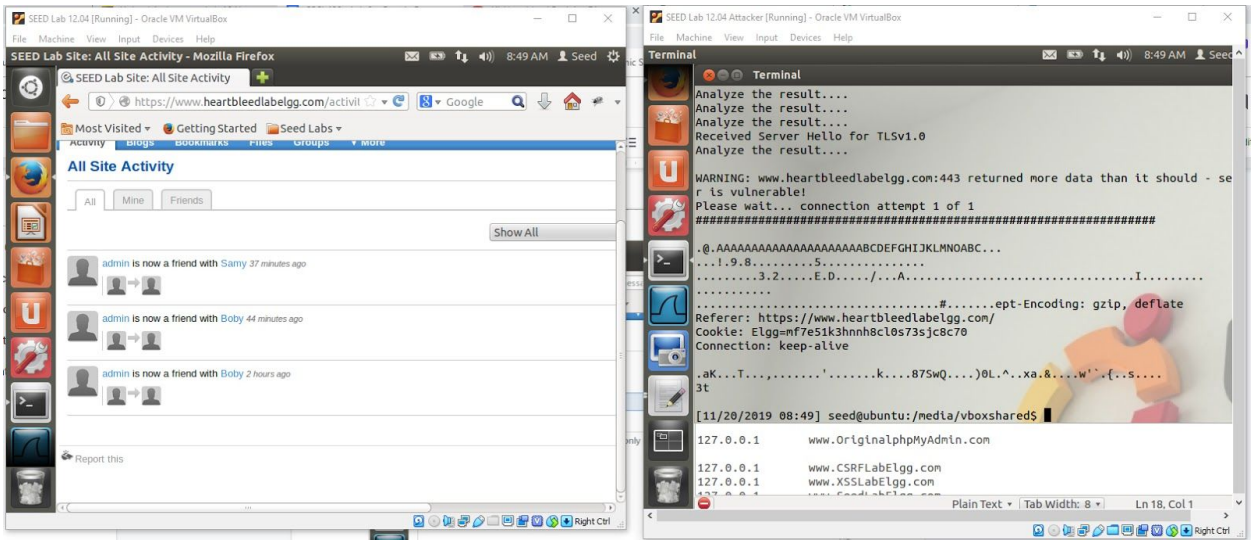
Now we have to do some back and forth on the www.heartbleedlabelgg.com on the Victim's computer. We will login into the social network, send a few messages to members and run the **attack.py** exploit.

Below are some pictures that prove interaction on the Victim's machine:

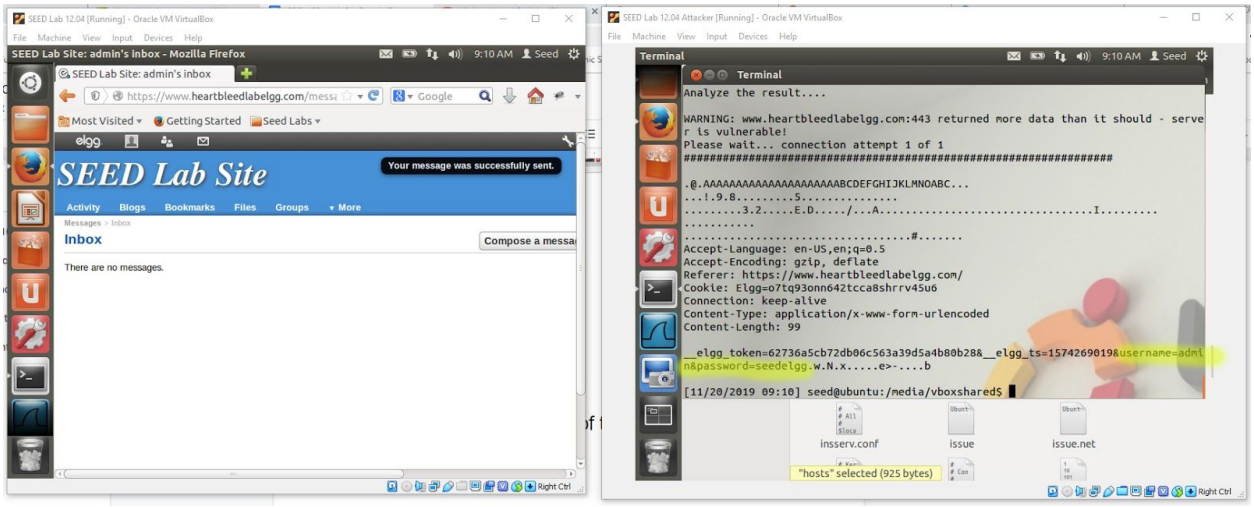

SEED Lab 12.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SEED Lab Site: Boby - Mozilla Firefox
8:45 AM Seed
SEED Lab Site: Boby
https://www.heartbleedlabelgg.com/
Google
Most Visited
Getting Started
Seed Labs
Boby
Add friend
Report user
Send a message
Blogs
Bookmarks
Files
Pages
Right Ctrl


SEED Lab 12.04 [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
SEED Lab Site: Compose a message - Mozilla Firefox
8:47 AM Seed
SEED Lab Site: Compose a mes...
https://www.heartbleedlabelgg.com/messa
Google
Most Visited
Getting Started
Seed Labs
Messages > Compose a message
Compose a message
To: Boby
Subject:
Hello Boby
Message:
This is a secret message that should be opened only by Boby.
This version on the SEEDlab will be hacked!!!
HAhaHAhaHA :)))))
Your new Friend
Send
Right Ctrl

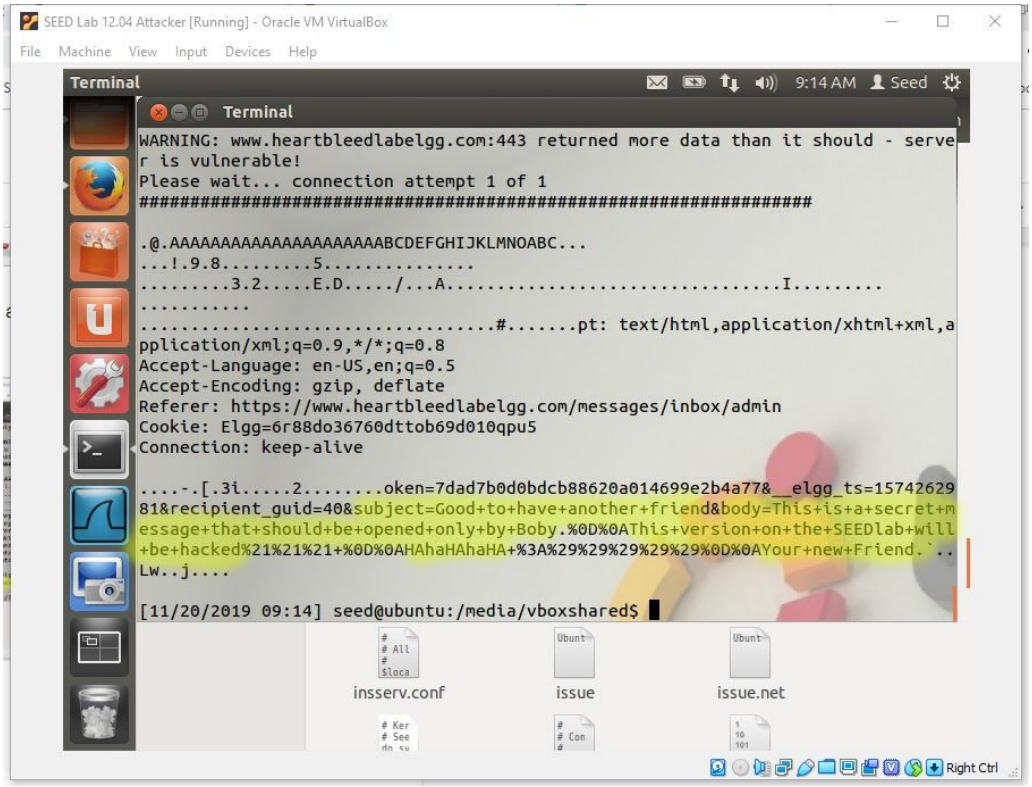
Here begging the attack:



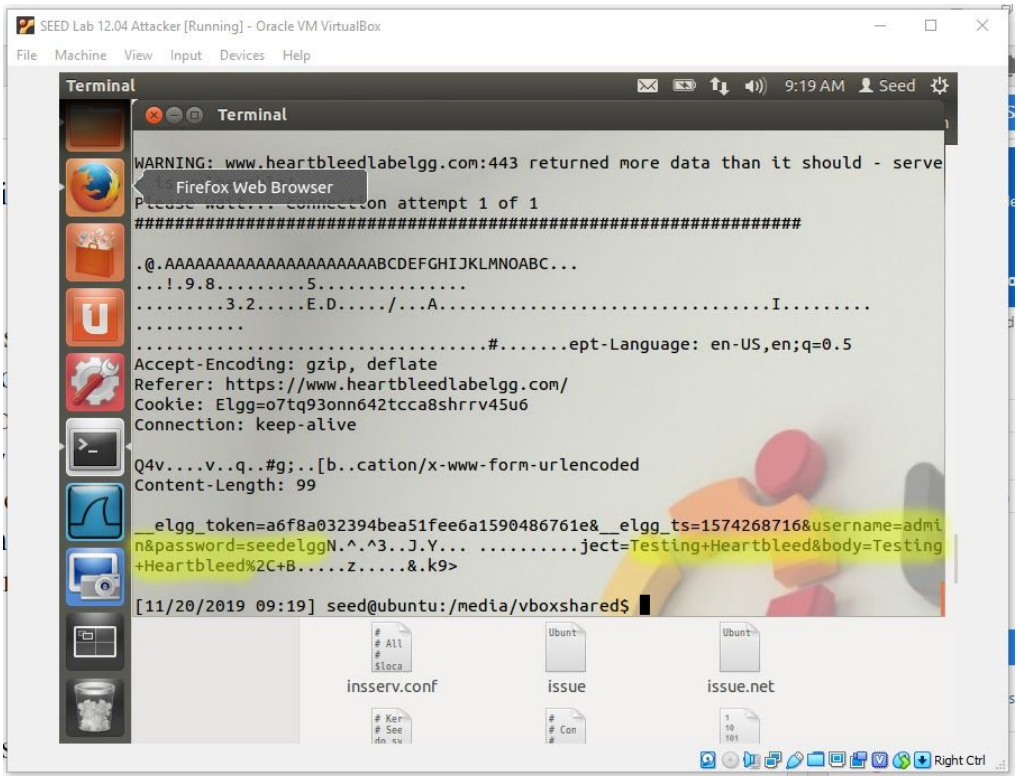
And after a few tries we get to see the **username** and **password** used to Login for www.heartbleedlabelgg.com.



Along with the above message and title:



And yet again another message from Boby as well as the credentials used to login:

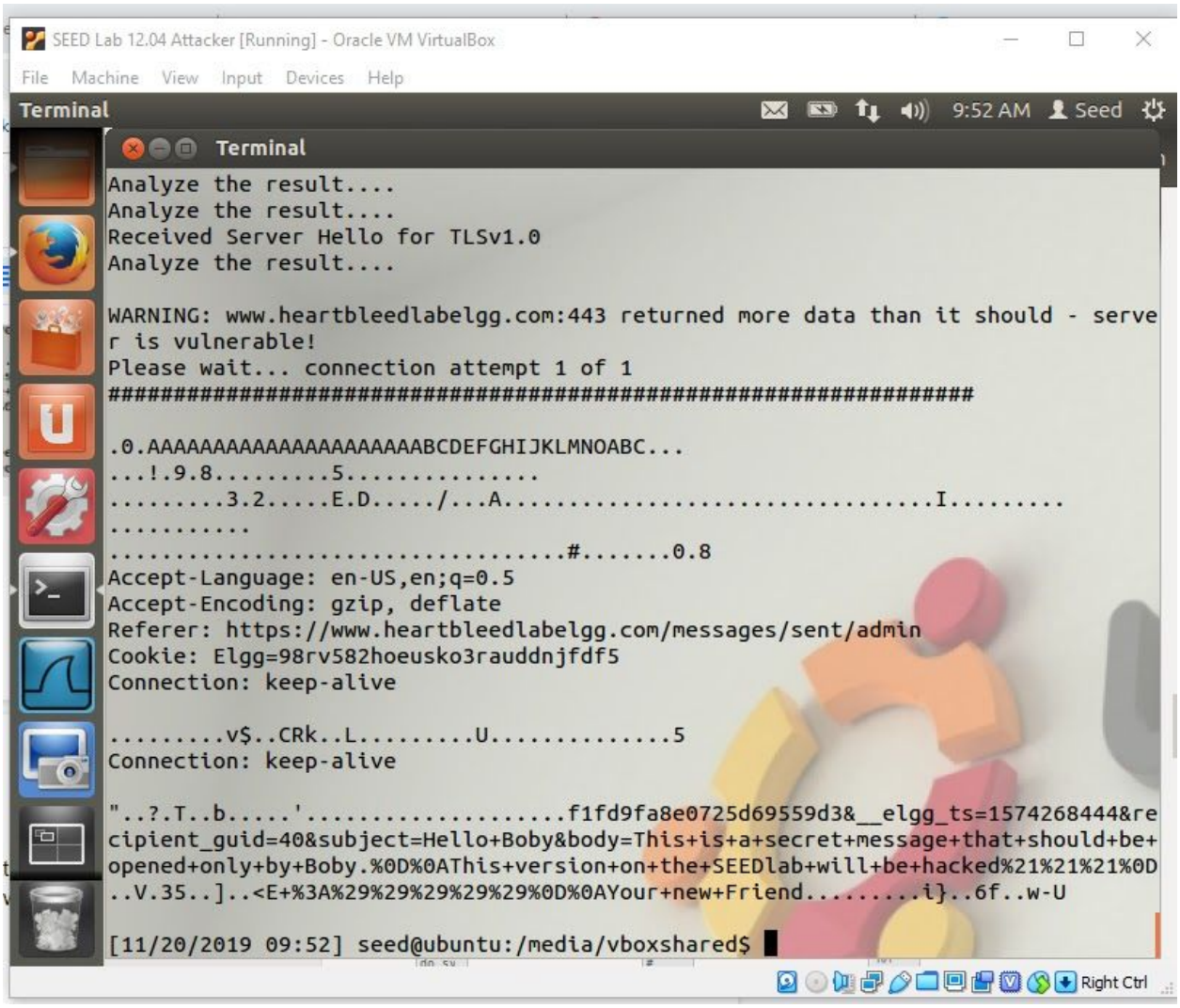


Once we run the exploit, data stored in the memory will appear on the attackers machine. We need to run the exploit multiple times so that we can get to the part in the memory that we are interested in. It seems that data is retrieved in the order it was stored in the memory. If **admin** logs in, then befriends **Boby** and then sends him a message, Heartbleed exploit will retrieve the information to the attacker in the same order; **Login -> Boby -> message**.

3.2 Task 2: Find the Cause of the Heartbleed Vulnerability

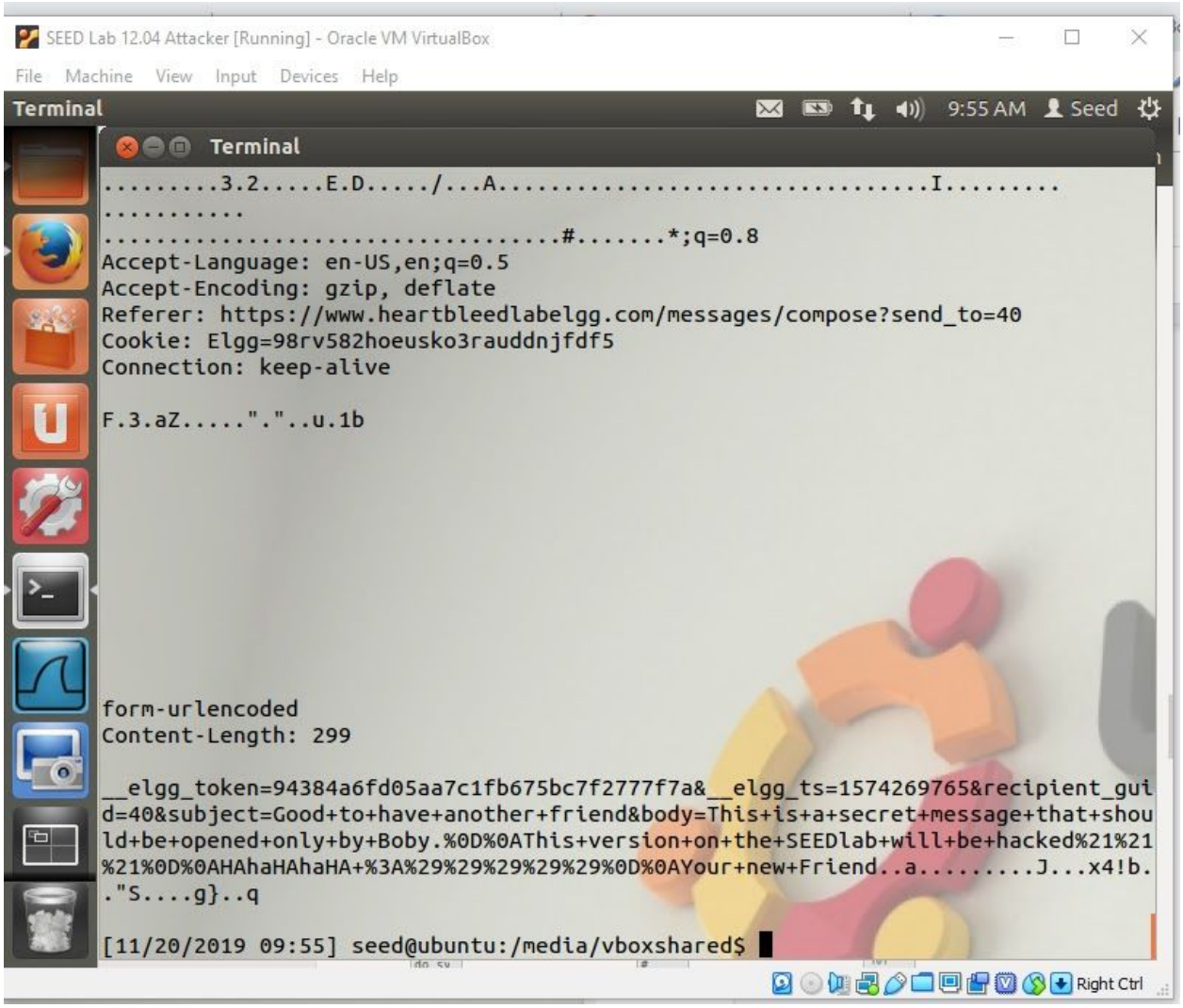
The top image shows the exploit running with extra command **“ -l 0x4000”**, which should display the whole information, and we can see that we have the **username** and **password displayed** along with the whole **message**.

The following image is run with extra command **“ -l 0x3000”** and we can see everything as well.



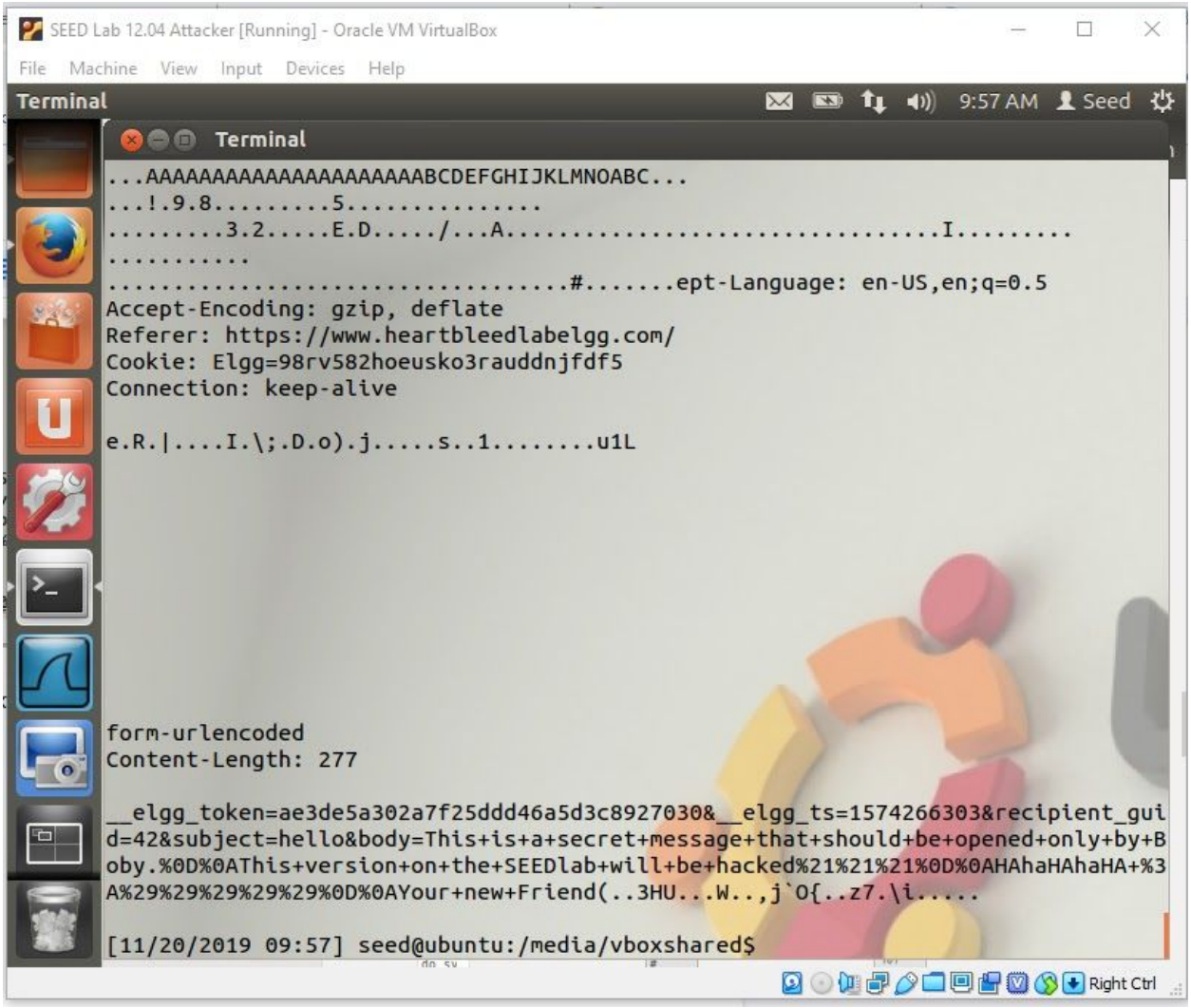
The following image is run with extra command **“ -l 0x2000”** and we can see everything as well.

And as we can see the **Content-Length is 299.** => this might night be actually related with the extra command, but just with the message content.



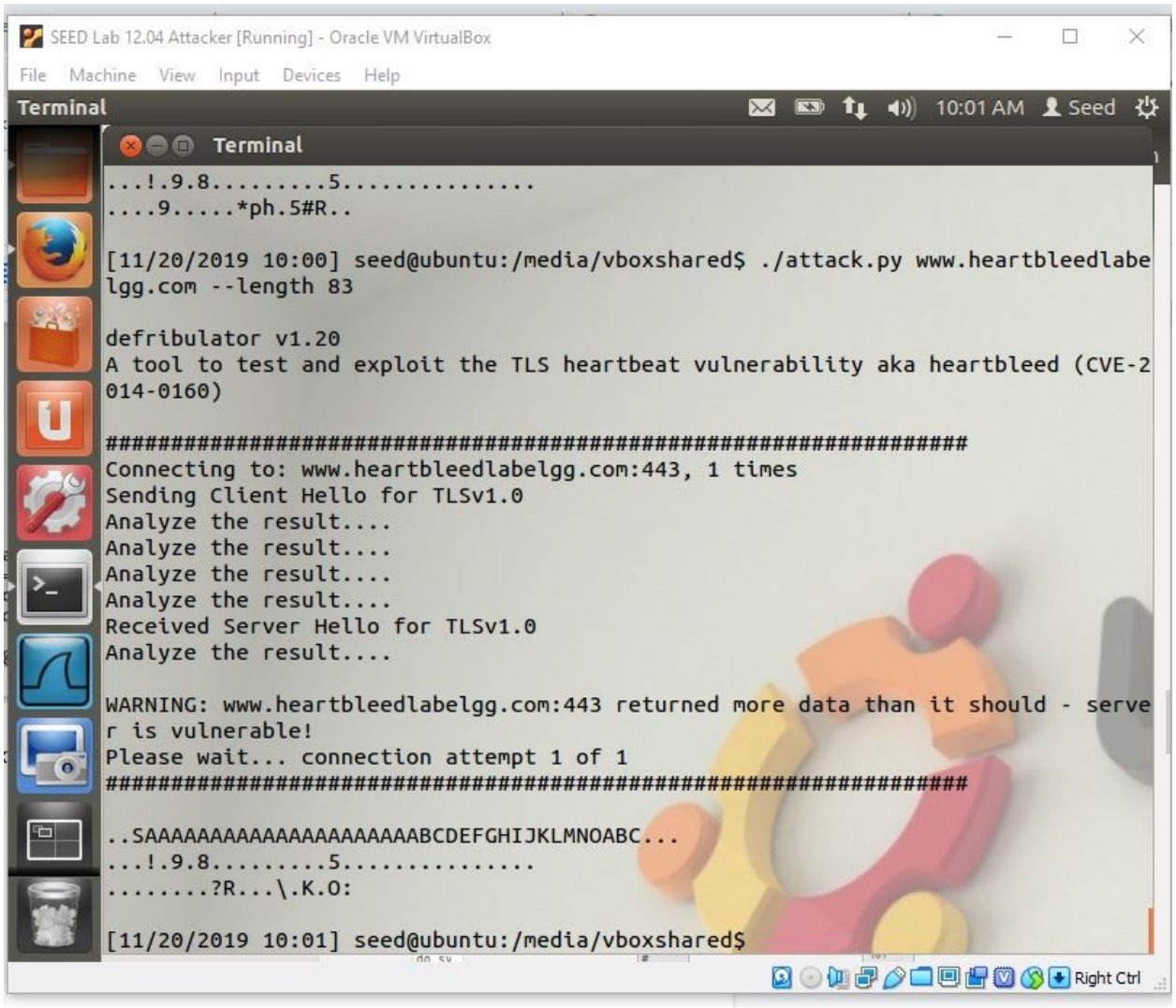
The following image is run with extra command **“ -l 0x158B”** and we can see everything as well.

Here the **Content-Length is 277.** => this might night be actually related with the extra command, but just with the message content.



The following image is run with extra command **“ --length 83”** and we can see everything as well.

Here we didn’t even get to the **Content-Length** part. After running this command multiple times, it stops showing information short.



Question 2.1 - As the length variable decreases, what kind of difference can you observe?

As the length variable decreases the information retrieved is less. In our case the information was completely retrieved, because it was only credentials and few messages, even when the **length** command assigned a lower *Payload_Length*.

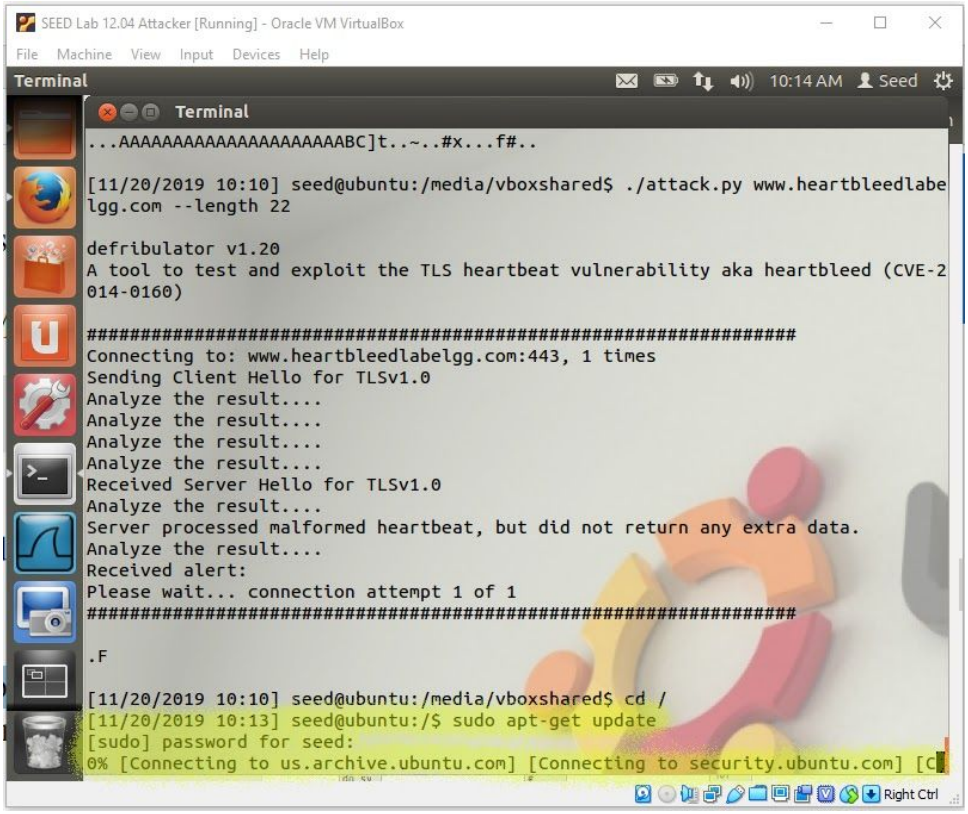
Question 2.2 - As the length variable decreases, there is a boundary value for the input length variable. Please find that boundary length.

$ - ./attack.py www.heartbleedlabelgg.com --length 22 - This was the lowest boundary length for the exploit to display the message:

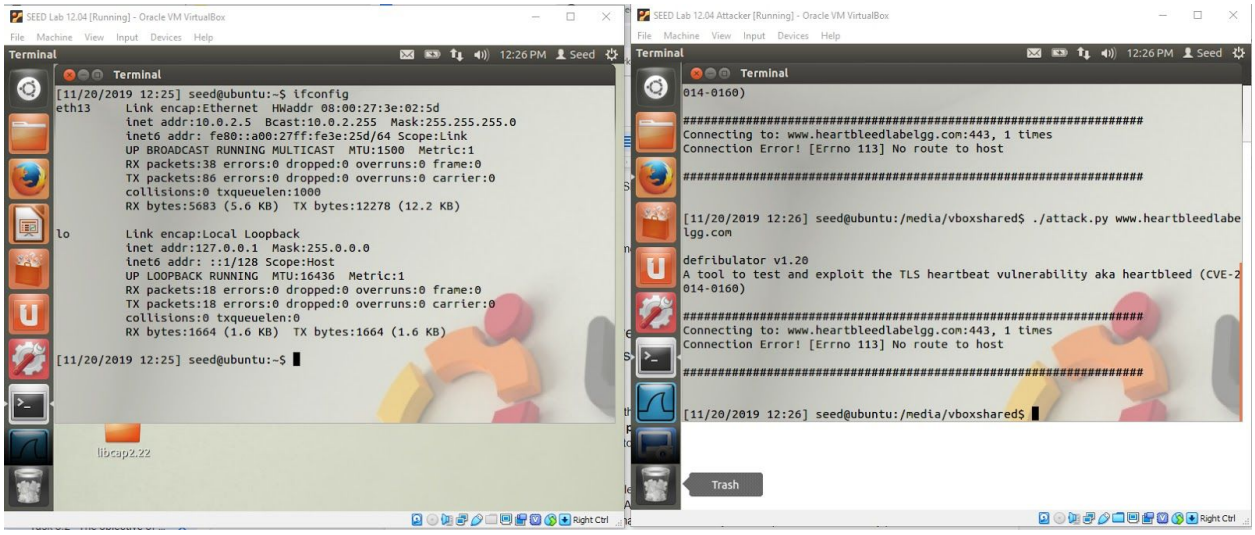
“Server processed malformed Heartbeat, but did not return any extra data.”

3.3 Task 3: Countermeasure and Bug Fix

Task 3.1 - Try your attack again after you have updated the OpenSSL library. Please describe your observations.



After installing and updating the SEEDlab ubuntu the exploit is not working anymore:



As we can see we have tried numerous times but the exploit does not work anymore since the system has been patched.

Task 3.2 - The objective of this task is to figure out how to fix the Heartbleed bug in the source code.

Thanking a look at the Listing 1, this tends to be sort of a **“Buffer Overflow”** situation, since the problem resides in **memcpy(bp, pl, payload)** in which the program copies the payload. The data being copied is not verified to be the same size as the **pl,** and therefore the **payload** could very easily be over that size.

One way to fix the code is to implement a **boundary checking** code that will verify during the buffer copy, and this proves that Alice’s thinking was correct. Another way to protect against **heartbleed attack** would be to make sure that the system is up to date with security patches and updates.

Bob’s thinking about the user input in this circumstance doesn’t have anything to do with the actual attack. In the case of user input the **boundary checking** should be in place and there would be no vulnerability point.

As for Eve, if there is no **length** limit, the **Heartbleed Attack** would just not work.