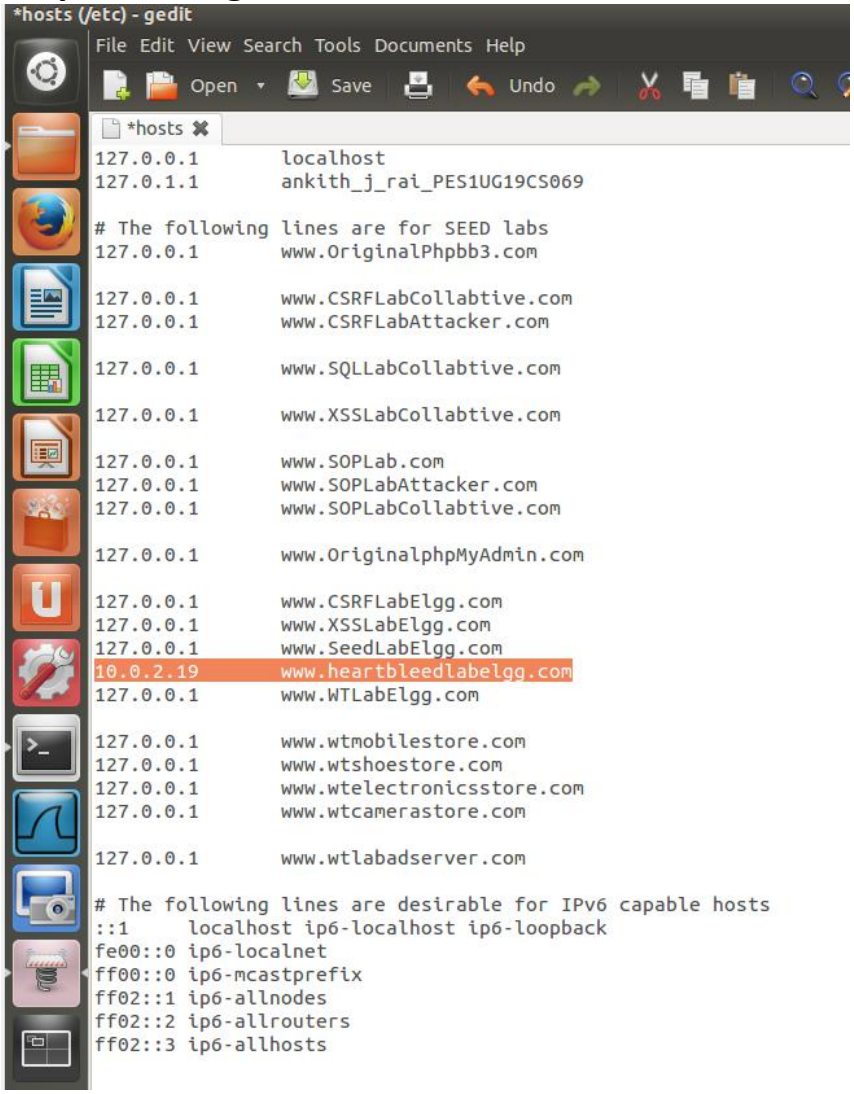
# Heartbleed Attack Lab

Name : Ankith J Rai
SRN : PES1UG19CS069
SEC : B

| Machine | IP address |
|---|---|
| Attacker | 10.0.2.18 |
| Victim | 10.0.2.19 |

## Step 1: Configure the DNS server for Attacker machine

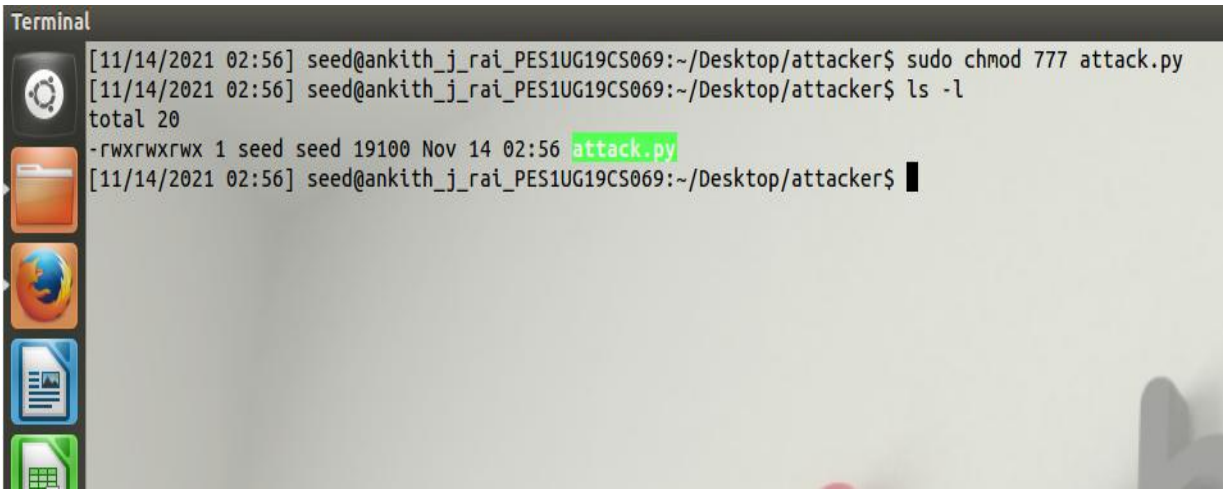



We can see that now we have changed the ip address of www.heartbleedlabelgg.com from 127.0.0.1 to 10.0.2.19(ip address of

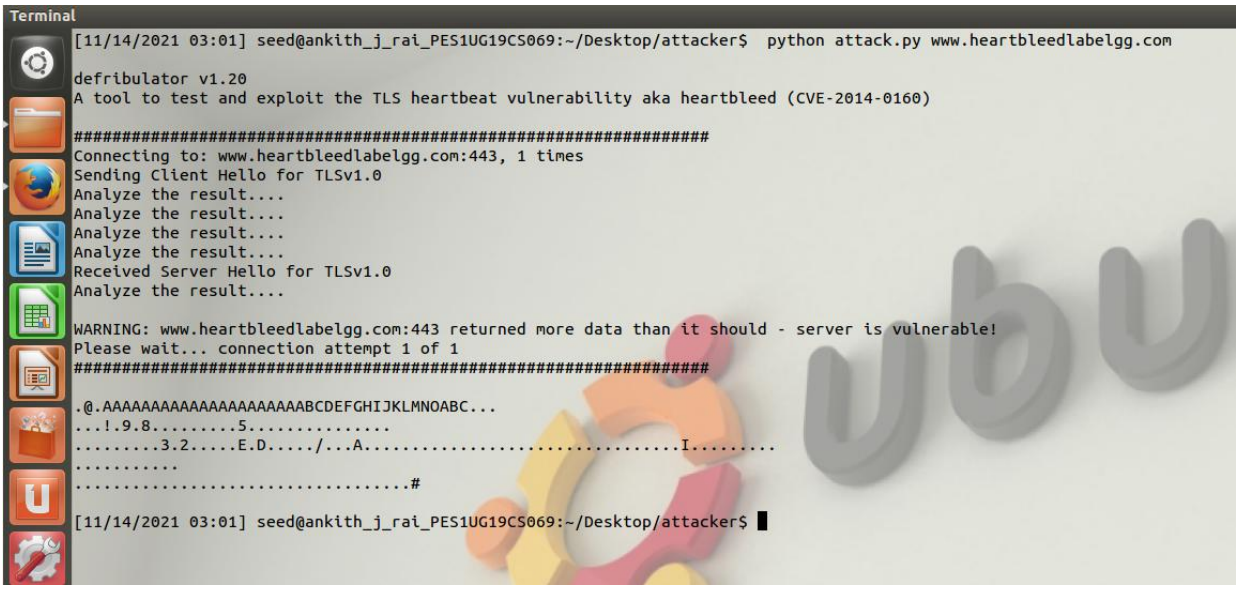
victim machine) .Hence now attacker believes that the www.heartbleedlabelgg.com is at the 10.0.2.19.

## Step 2: Lab Tasks

Now we will run the command **sudo chmod 777 attack.py** on attacker machine.



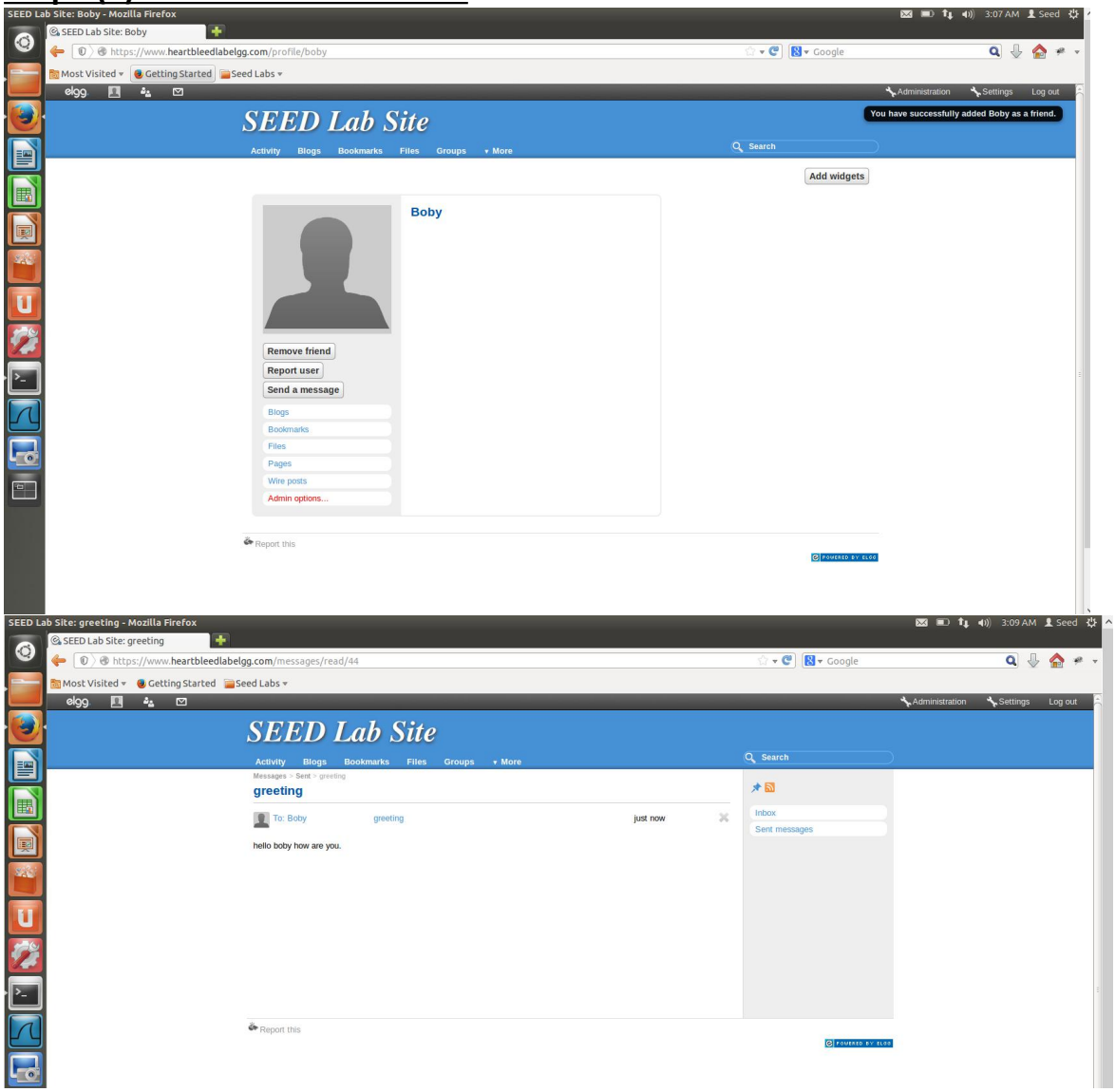

Now running **python attack.py www.heartbleedlabelgg.com** on attcker machine.




The above command sends malicious requests to **www.heartbleedlabelgg.com.** But we can see that what ever data we have received contains no useful/secret information.

## Step 2: Explore the damage of the Heartbleed attack

## Step 2(a): On the Victim Server:

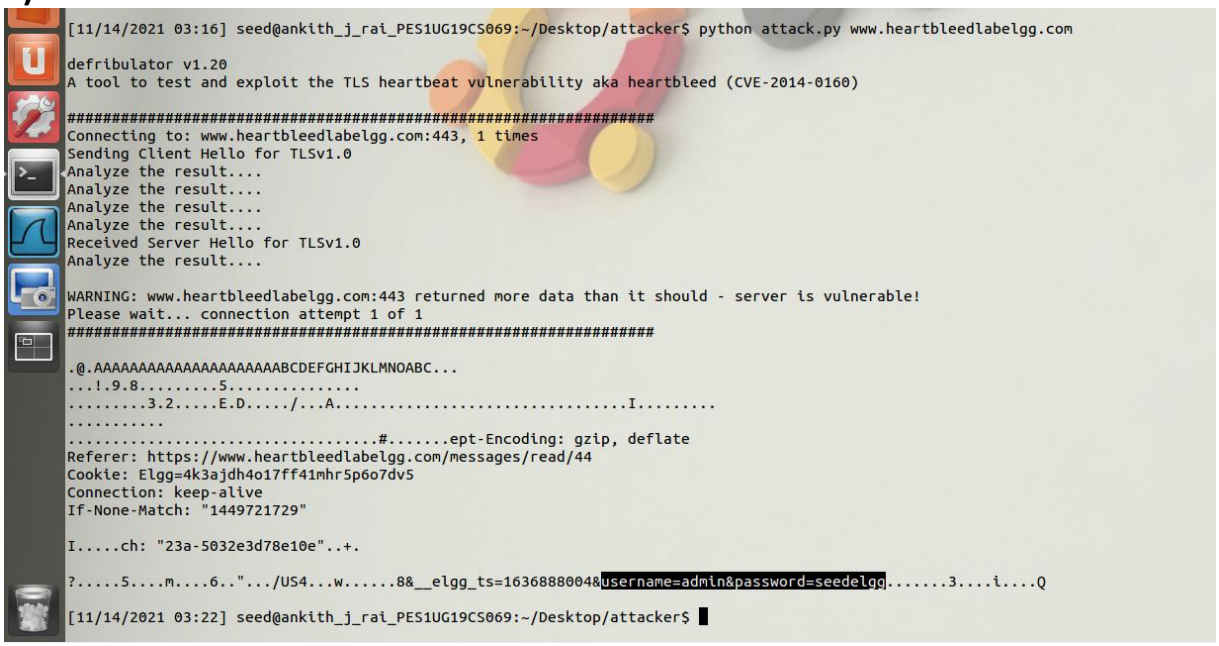



From the above screenshot we can see that we have added **boby** as a friend on the website **www.heartbleedlabelgg.com .** We can also see that we have sent a greeting message to **boby** as " **hello boby how are you**."

## Step 2(b): On Attacker machine:

Here we are going to run the command **python attack.py** [www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com) again and again inorder to get usefull information.

## 1) Find out the Username & Password




We can see from the above screenshot that the username is "**admin**" and the password is "**seedelgg**".

## 2) Find the exact content of the private message

From the above screenshot we can see that the content of the message is that the subject is “**greeting**” and the body contains “**hello how are you.**”

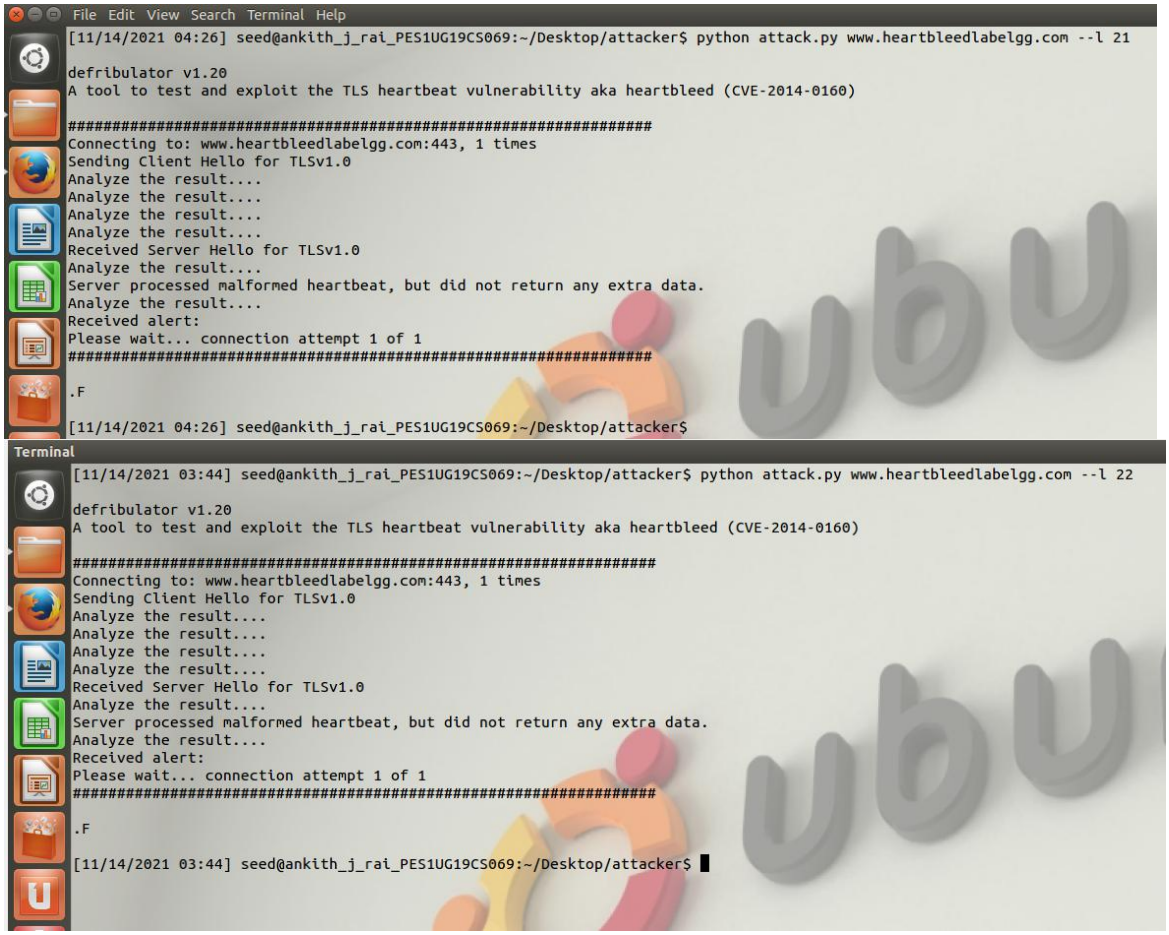
## Step 3: Investigate the fundamental cause of the Heartbleed attack

Now we are going to change the payload length of the heartbleed response packet.

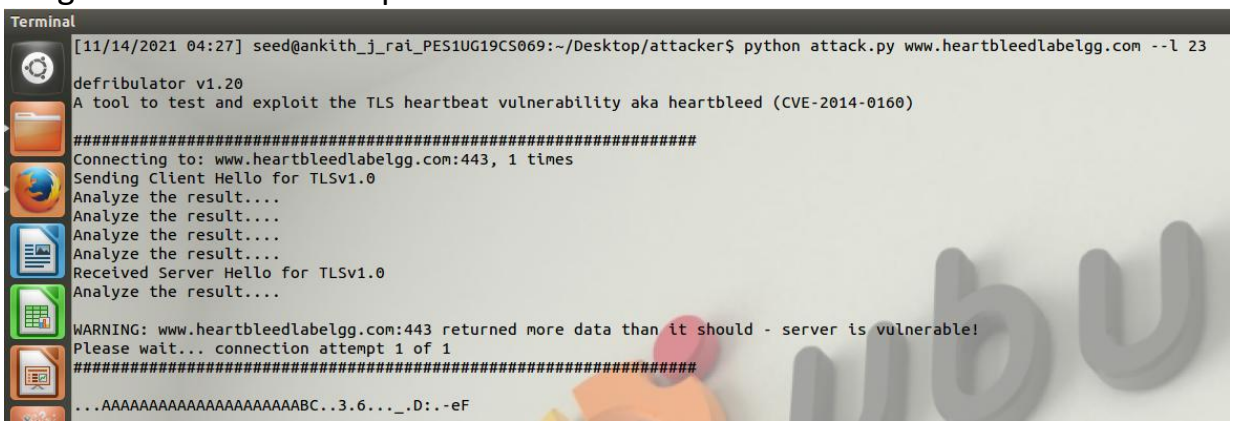

From the above screenshot we can see that I have set the payload length of the response packet to be **40 bytes**.We can see that we are getting random values in the output and this information is not usefull to us at all.

From the above screenshot we can see that I have set the payload length of the response packet to be **0x012B (299)** bytes.We can see that we are getting random values in the output and this information is not usefull to us at all.

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




From repeated trials we get to know that till payload length of 22 bytes we get no data in the response.



Now we can see that when the payload length is 23 bytes we get data in the response.Hence 23 bytes is the boundary value of the payload length.At any value for the payload above this causes the extra data/information coming in the response.

## Step 5: Countermeasure and bug fix

## Methods to prevent this attack:

1) We can update the SSL and use the updated SSL which has solved this problem.
2) In the code perspective we add the below code:

```
...
hbtype =
*p++;
n2s(p,payloa
d);
        if (1 + 2 + payload + 16 >
      sizeof(HeartbeatMessage)) return 0; /* silently
      discard per RFC 6520 sec. 4*/
```

From the above code we can see that in the conditional if statement the condition is (1+2+payload + 16 > sizeof(HeartbeatMessage) ) where value 1 stands storing for 1-byte type , value 2 stands for storing 2-byte type and value 16 is used for padding. If the summation (1+2+payload+16) is greater than the sizeof(HeartbeatMessage) then the request packet is rejected.On this conditions the the Heartbleed attack is prevented.