COMPUTER NETWORK SECURITY LAB - UE18CS335

Heartbleed Attack Lab

Name: Ankitha P **Class:** 6 'D'

Date : 17/04/2021

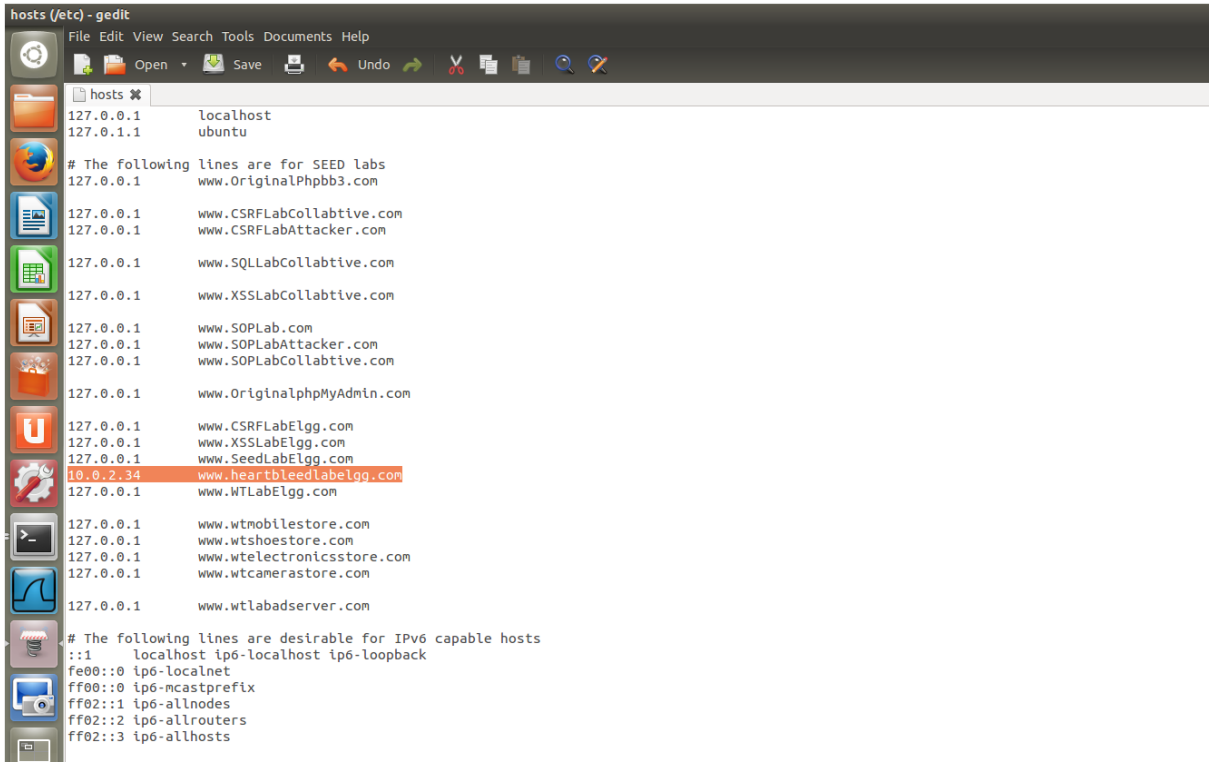
The Heartbleed bug (CVE-2014-0160) is a severe implementation flaw in the OpenSSL library, which enables attackers to steal data from the memory of the victim server. The contents of the stolen data depend on what is there in the memory of the server. It could potentially contain private keys, TLS session keys, user names, passwords, credit cards, etc. The vulnerability is in the implementation of the Heartbeat protocol, which is used by SSL/TLS to keep the connection alive.

LAB SETUP

Victim : 10.0.2.34

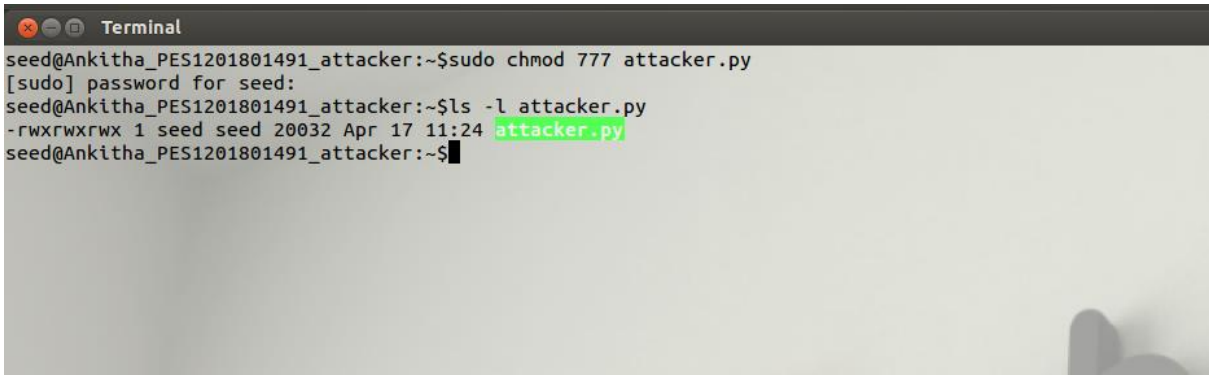
Attacker : 10.0.2.33

We modify the /etc/hosts on the Attacker's machine (10.0.2.33) to make the website www.heartbleedlabelegg.com seem to exist on the victim machine (10.0.2.34) as shown below.



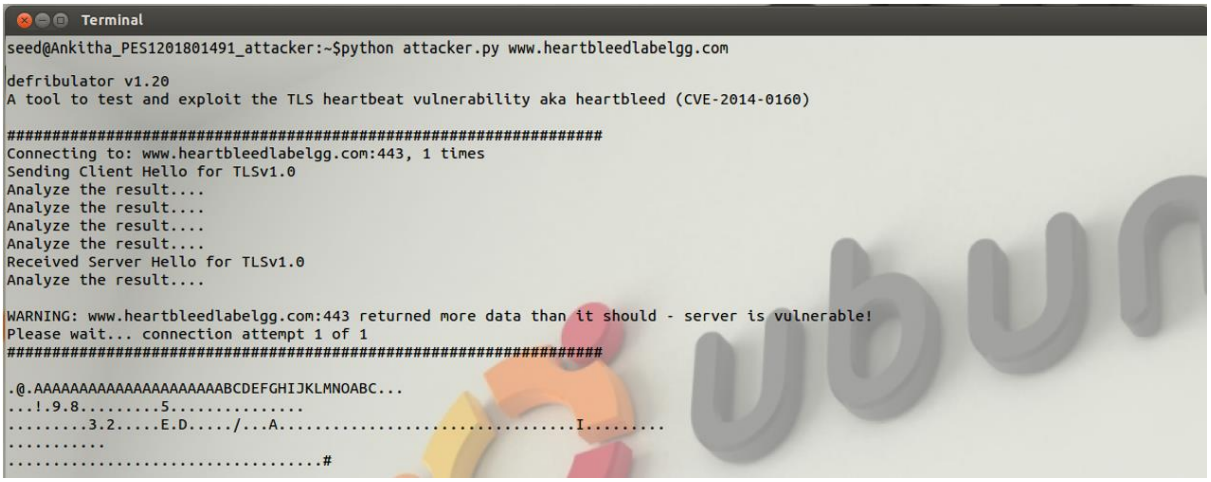
Step 2: Lab Tasks

Using the attacker.py program, we send out the malicious heartbeat request to the server www.heartbleedlabelgg.com and in response we get random data from the server. In order to run it we make it executable as shown below.



As warm up task, use the following command to run the attacker.py code on Attacker machine:

$ python attacker.py www.heartbleedlabelgg.com



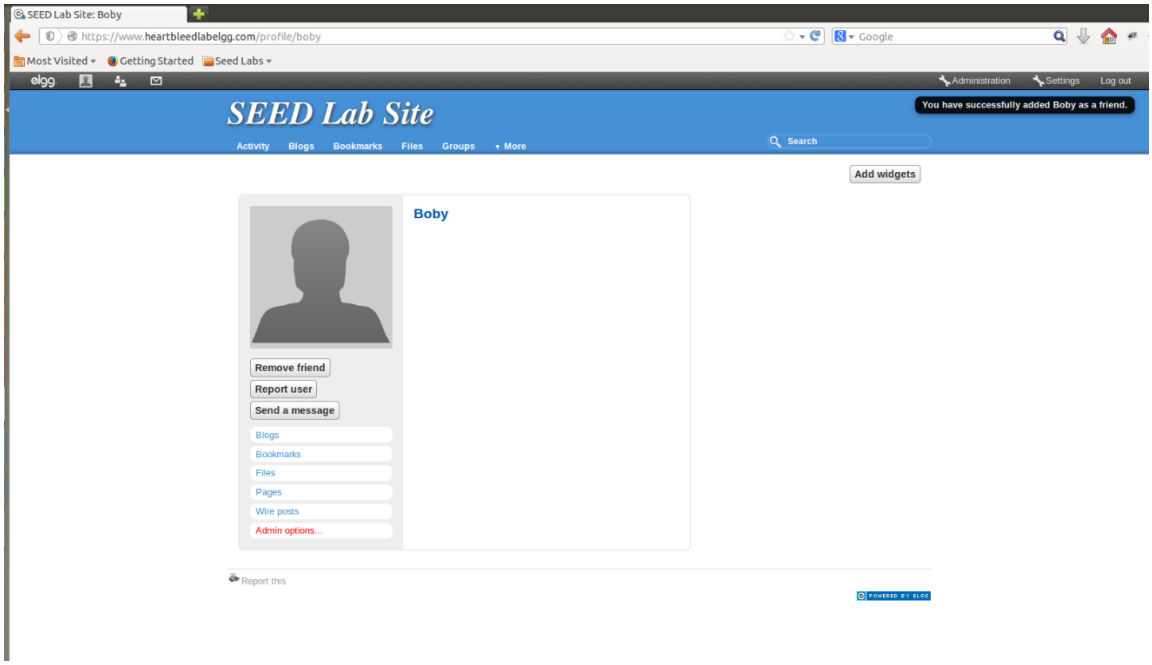
Basically, attacker.py is a program that will send out the malicious heartbeat request to the server www.heartbleedlabelgg.com and in response it will get random data from the server. From the random-data, we could see that no matter how many times we try, we always receive saying something similar to this that the server is vulnerable because it is sending more data than it should. We can see this in the above figure. 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

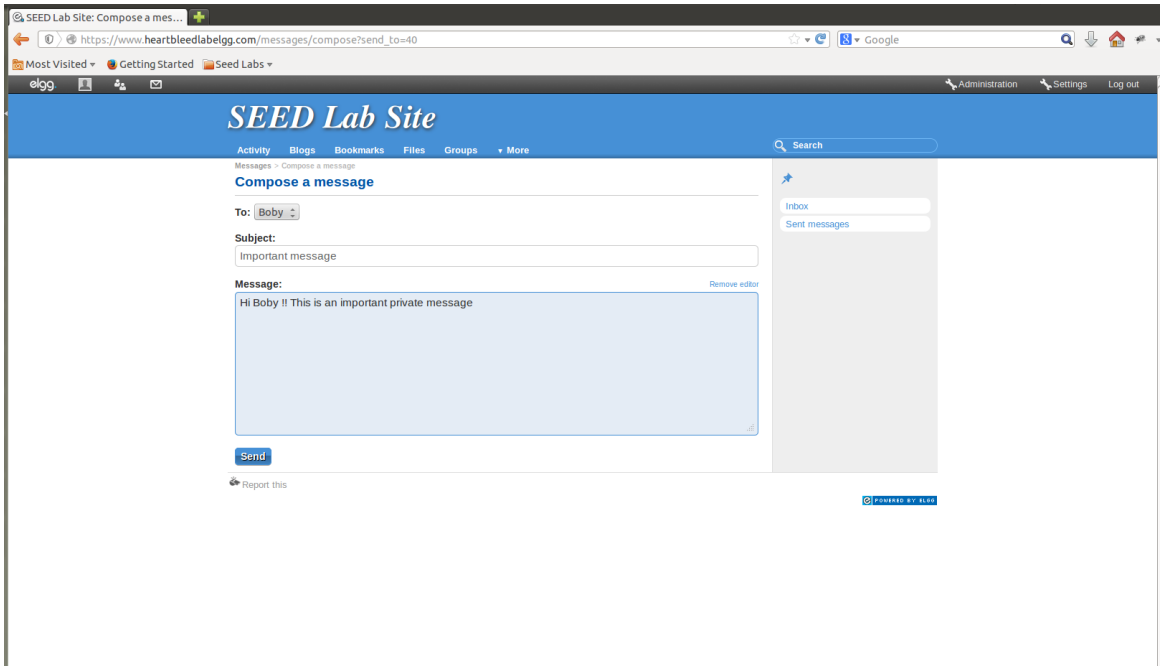
Step 2(a): On the Victim Server:

We visit the www.heartbleedlabelgg.com site and login as admin with the password as seedelgg. We add Boby as a friend and send him a private message as shown below:

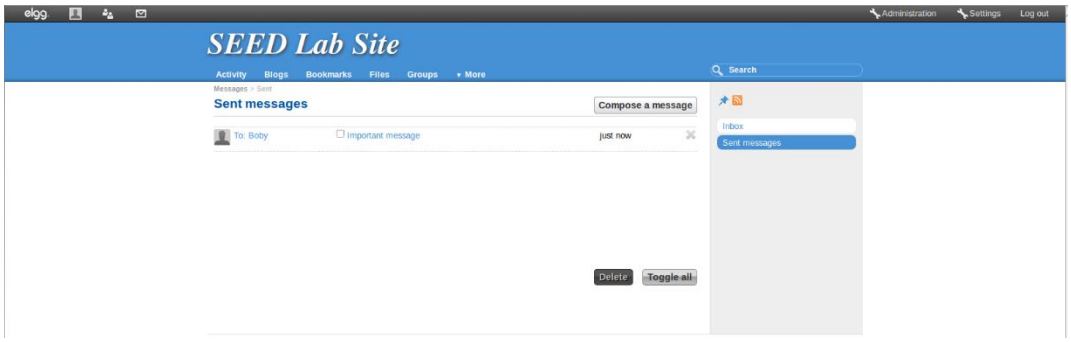
Boby added as a friend:



Sending a personal message to Boby:



Message successfully sent to Boby:



The above screenshot shows the private message sent from the admin account to Boby. These activities will be recorded by the server and can be obtained as saved secret data in the heartbleed attack.

Step 2(b): On Attacker machine:

We run the attack.py code on the attacker machine to find out user activity, password, username and the content of the user's private message. We run it multiple times to get the data we require. The result is different from the previous because the server's memory is not empty anymore after the site was used by the victim to login and send a message to Boby. As the memory allocation is random, the result or dumped data obtained is random each time too.

1) The below screenshot shows that we have obtained the username as password as admin and seedelgg

2)The below screenshot shows that we were able to obtain the exact content of the private message as 'Hi Boby! This is an important private message.'

Step 3: Investigate the fundamental cause of the Heartbleed attack

The fundamental cause of the Heartbleed attack vulnerability is that there is a missing user input validation while constructing the Heartbeat response packet. Based on the payload length, the extent of the attack, i.e., amount of data obtained from the vulnerable server is decided.

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

On trial and error, we find that the boundary length for which the server will not return any extra data is 22.

We can also see that anything beyond this payload length, using length as 23, will leak data from the server.