

## Programming Project 1: Heartbleed Attack

Out: 08/29/18 Due: 09/07/18 11:59pm

### Instructions

1. Strictly adhere to the University of Maryland Code of Academic Integrity.
2. Submit your solutions as a pdf document at Canvas. Include your full name in the solutions document. Name the solutions document as x-project1.pdf, where x is your last name.

*Adjusted from SEED labs <http://www.cis.syr.edu/~wedu/seed/>*

## 1 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 our Ubuntu VM is 1.0.1.

## 2 Lab Setup

In this lab, we need to set up two VMs: one called attacker machine and the other called victim server. We use the pre-built SEEDUbuntu12.04 VM, which you can download from [http://www.cis.syr.edu/~wedu/seed/lab\\_env.html](http://www.cis.syr.edu/~wedu/seed/lab_env.html). To configure two different VMs in VirtualBox, you also need to check this link: <http://www.cis.syr.edu/~wedu/seed/Documentation/VirtualBox/LoadingMultiVMs.pdf>. The VMs need to use the NAT-Network adapter for the network setting. This can be done by going to the VM settings, picking Network, and clicking the Adaptor tag to switch the adapter to NAT-Network. Make sure both VMs are on the same NAT-Network adapter.

The website used in this attack can be any HTTPS website that uses SSL/TLS. However, since it is illegal to attack a real website, we have set up a website in our VM, and conduct the attack on our own VM. We use an open-source social network application called ELGG, and host it in the following URL: <https://www.heartbleedlabelgg.com>. We need to modify the `/etc/hosts` file on the attacker machine to map the server name to the IP address of the server VM. Search the

following line in `/etc/hosts`, and replace the IP address `127.0.0.1` with the actual IP address of the server VM that hosts the ELGG application. You may need to use `sudo` command to modify this file.

### 3 Lab Tasks

Before working on the lab tasks, you need to understand how the heartbeat protocol works. The heartbeat protocol consists of two message types: `HeartbeatRequest` packet and `HeartbeatResponse` packet. Client sends a `HeartbeatRequest` packet to the server. When the server receives it, it sends back a copy of the received message in the `HeartbeatResponse` packet. The goal is to keep the connection alive. For more information you can refer to this URL: [https://en.wikipedia.org/wiki/Heartbeat\\_\(computing\)](https://en.wikipedia.org/wiki/Heartbeat_(computing)).

#### 3.1 Task 1: Launch the Heartbleed Attack

In this task, students will launch the Heartbleed attack (check here: <http://heartbleed.com/> for more details) on our social network site and see what kind of damages can be achieved. The actual damage of the Heartbleed attack depends on what kind of information is stored in the server memory. If there has not been much activity on the server, you will not be able to steal useful data. Therefore, we need to interact with the web server as legitimate users. Let us do it as the administrator, and do the followings:

- Visit `https://www.heartbleedlabelgg.com` from your browser.
- Login as the site administrator. (User Name: `admin`; Password: `seedelgg`)
- Add Bobby as friend. (Go to `More` -> `Members` and click `Bobby` -> `Add Friend`)
- Send Bobby a private message.

After you have done enough interaction as legitimate users, you can launch the attack and see what information you can get out of the victim server. Writing the program to launch the Heartbleed attack from scratch is not easy, because it requires the low-level knowledge of the Heartbeat protocol. Fortunately, other people have already written the attack code. Therefore, we will use the existing code to gain first-hand experience in the Heartbleed attack. The code that we use is called `attack.py`, which was originally written by Jared Stafford. We made some small changes to the code for educational purposes. You can download the code from the lab's web site, change its permission so the file is executable. You can then run the attack code as follows:

```
$ ./attack.py www.heartbleedlabelgg.com
```

You may need to run the attack code multiple times to get useful data. Try and see whether you can get the following information from the target server.

- User name and password.
- User's activity (what the user has done).
- The exact content of the private message.

For each piece of secret that you steal from the Heartbleed attack, you need to show the screen-dump as the proof and explain how you did the attack, and what your observations are.

### 3.2 Task 2: Find the Cause of the Heartbleed Vulnerability

In this task, students will compare the outcome of the benign packet and the malicious packet sent by the attacker code to find out the fundamental cause of the Heartbleed vulnerability. The Heartbleed attack is based on the Heartbeat request. This request just sends some data to the server, and the server will copy the data to its response packet, so all the data are echoed back. In the normal case, suppose that the request includes 3 bytes of data "ABC", so the length field has a value 3. The server will place the data in the memory, and copy 3 bytes from the beginning of the data to its response packet. In the attack scenario, the request may contain 3 bytes of data, but the length field may say 1003. When the server constructs its response packet, it copies from the starting of the data (i.e. "ABC"), but it copies 1003 bytes, instead of 3 bytes. These extra 1000 bytes obviously do not come from the request packet; they come from the server's private memory, and they may contain other user's information, secret keys, password, etc. In this task, we will play with the length field

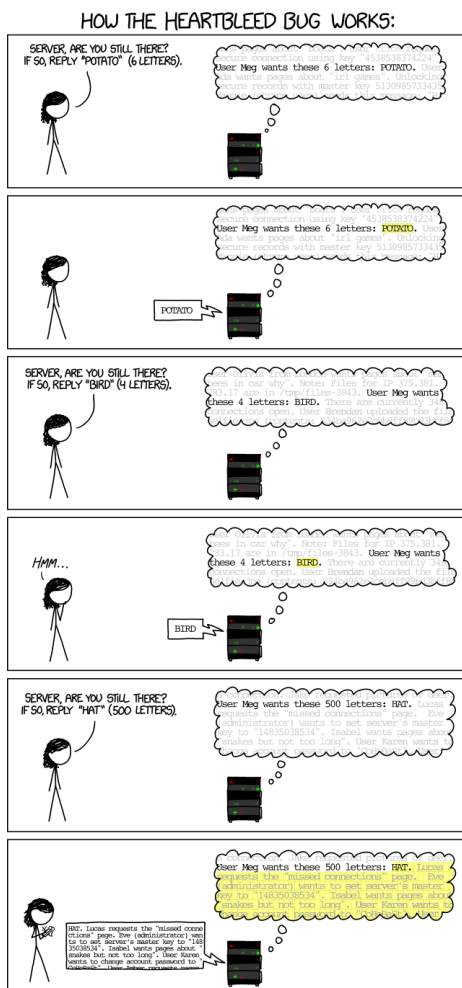


Figure 1: Heartbleed Attack

of the request. First, let's understand how the Heartbeat response packet is built from Figure 2. When the Heartbeat request packet comes, the server will parse the packet to get the payload and the Payload length value. Here, the payload is only a 3-byte string "ABC" and the Payload

length value is exactly 3. The server program will blindly take this length value from the request packet. It then builds the response packet by pointing to the memory storing "ABC" and copy `Payload length` bytes to the response payload. In this way, the response packet would contain a 3-byte string "ABC". We keep the same payload (3 bytes), but set the `Payload length` field to 1003. The server will again blindly take this `Payload length` value when building the response packet. This time, the server program will point to the string "ABC" and copy 1003 bytes from the memory to the response packet as a payload. Besides the string "ABC", the extra 1000 bytes are copied into the response packet, which could be anything from the memory, such as secret activity, logging information, password and so on. Our attack code allows you to play with different `Payload length` values. By default, the value is set to a quite large one (0x4000), but you can reduce the size using the command option "-l" (letter ell) or "--length" as shown in the following examples:

```

$ ./attack.py www.heartbleedlabelgg.com -l 0x015B
$ ./attack.py www.heartbleedlabelgg.com --length 83

```

Note that if the attack does not return additional information then you will observe a message: "Server processed malformed Heartbeat, but did not return any extra data." and then the legitimate message followed by some random padding. On the other hand, if the attack returns more information you will observe a Warning message. Your task is to play with the attack program with different payload length values and answer the following questions:

- As the length variable decreases, what kind of difference can you observe?
- As the length variable decreases, there is a boundary value for the input length variable. At or below that boundary, the Heartbeat query will receive a response packet without attaching any extra data (which means the request is benign). Please find that boundary length. to try many different length values until the web server sends back the reply without extra data. When the number of returned bytes is smaller than the expected length, the program will print "Server processed malformed Heartbeat, but did not return any extra data."

### 3.3 Fix the bug

The objective of this task is to figure out how to fix the Heartbleed bug in the source code of OpenSSL Library. This task will give you some practical experience on how to patch crypto libraries in a linux system. Please find the problem and recompile the OpenSSL Library code by following the following commands:

```

$ sudo apt-get install libssl-dev

$ cd openssl-1.0.1
$ sudo ./config --prefix=/usr/ --openssldir=/etc/ssl/
$ sudo make
$ sudo make install

```

Note that you will need to install `libssl-dev` package first and maybe restart your server VM in order to observe the difference between the unpatched and the patched version of OpenSSL. Please describe your observations and provide us with the corresponding evidence.

## 4 Submission

Students need to submit a detailed lab report to describe what they have done and what they have observed. Report should include the evidences to support the observations. Evidences include packet traces, screen dumps, etc.