网络安全课外实验

实验报告

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：0906140130

姓　　名：殷淑杰

目　录

HeartBleed 攻击实验

1. 概要介绍

Heartbleed 漏洞（CVE-2014-0160）是 OpenSSL 库中严重的应用缺陷，攻击者利用此漏洞可以从受害服务器内存中窃取数据。窃取的内容主要是服务器中所存储的信息。它可能包含私钥，TLS 会话密钥，用户名，密码，信用卡等。这个漏洞主要是因为 Heartbeat 协议通过 SSL/TLS 来保持连接状态。本实验的目的是为了让学生意识到这个漏洞的严重性，了解攻击是如何做的以及如何解决这个问题。存在这个漏洞的 OpenSSL 的版本从 1.0.1 到 1.0.1f。在我们的 Ubuntu 虚拟机上是 1.0.1。

2. 实验环境

在本实验中，我们需要建立两个虚拟机，一个称为攻击机器，另一个称为受害服务器。我们使用预先建立的 seedubuntu12.04 VM。虚拟机需要使用 NAT 网络适配器的网络设置。这可以通过点击虚拟机设置，选择网络，然后点击适配器标签切换到 NAT 网络适配器。一定要确保这两个虚拟机在同一 NAT 网络。

本次攻击可以攻击任何使用 SSL/TLS 的 HTTPS 的网站。然而，攻击一个真实的网站是违法行为，所以我们在虚拟机里虚构了一个网站来进行攻击实验。我们使用的是一个开源的社交网络应用程序（ELGG），网址为：https://www.heartbleedlabelgg.com。

我们需要修改攻击机器上的/etc/hosts 文件，将服务器名称映射到受害服务器的 IP 地址上。在/etc/hosts 中找到：127.0.0.1 www.heartbleedlabelgg.com，修改 127.0.0.1 为受害服务器 IP。

3. 实验内容

在实验前，你需要了解 Heartbeat 协议是如何工作的。协议包括两个消息类型：Heartbeat 请求包和 Heartbeat 应答包。客户端发送一个请求包给服务器端。当服务器端收到后，会在应答包中复制收到的消息并回传。目的是保持连接活性。具体方式如图 1 所示。

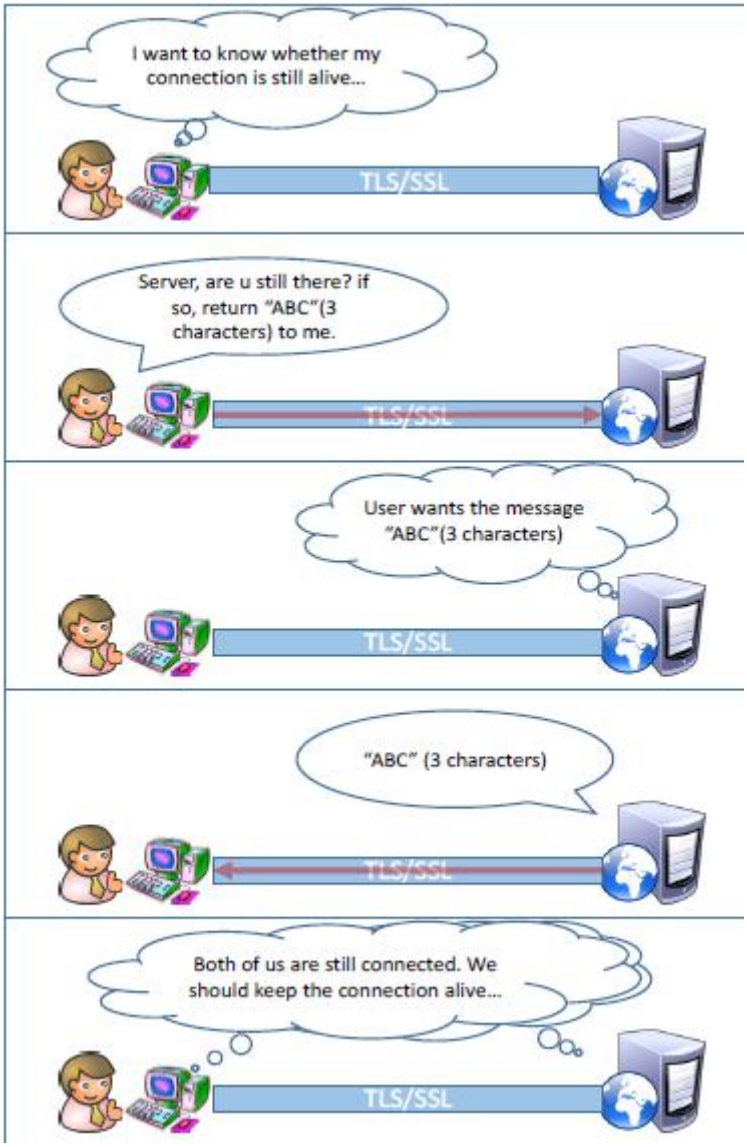


图 1.Heartbeat 协议

3.1 任务 1：启动 Heartbleed 攻击

实验者可以在我们的社交网络中启动攻击，观察会造成什么危害。Heartbleed 攻击所造成的危害严重与否取决于服务器内存中所存储的信息数据重要与否。如果服务器上没有进行太多的活动，那么你将无法取得有用的信息。因此，我们需要作为合法用户与网络服务器进行交互。实验步骤如下：

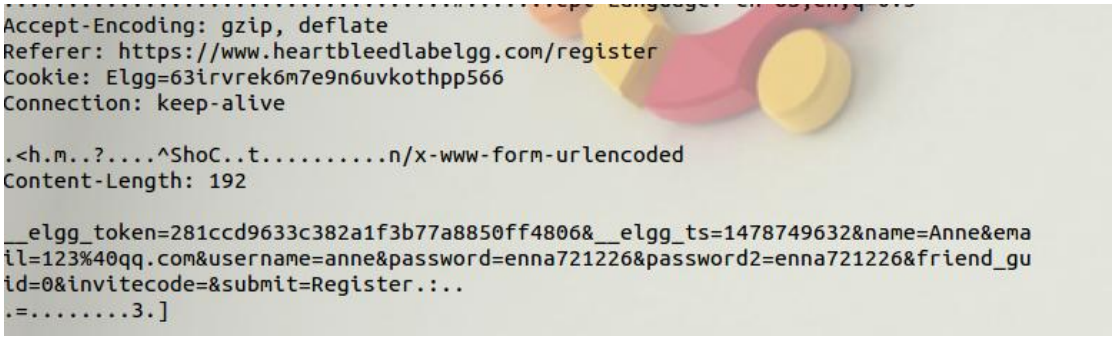
- 通过浏览器访问 https://www.heartbleedlabelgg.com
- 登录（用户名：admin ；密码：seedelgg）
- 添加 Bob 为好友（More -> Members and click Boby -> Add Friend）
- 给 Bob 发一条消息。

至此你已经完成了复杂的交互，你可以开始发动攻击，观察会获取什么信息数据。从头编写启动此攻击的程序并不简单，因为这需要我们对 heartbeat 底层协议有深刻的理解。但是，已经有人写出了这样的攻击代码。因此，我们将使用已有的代码来初试 Heartbleed 攻击。代码名为 attack.py，由 Jared Stafford 编写。为了实验，我们对这个代码进行了些许调整。你可以通过实验室网站直接下载，更改其权限，让其可执行。然后在攻击机器的终端输入以下指令：

$ ./attack.py www.heartbleedlabelgg.com

你需要多次运行来获得有用的数据。观察你是否得到的目标服务器上的以下信息：

- 用户名和密码
- 用户进行的活动
- 消息的确切内容



这是我自己在交互网站注册了一个账号，获取到了包括注册时填写的基本信息，以及用户名密码，以及我所发送的消息。

3.2 任务 2：寻找 Heartbleed 漏洞的原因

任务要求，通过比较正常数据包和恶意数据包的内容，探寻漏洞产生的根本原因。

Heartbleed 攻击是基于 Heartbeat 请求，请求包向服务器发送数据，而服务器复制后进行响应，所以所有数据都会被回传。在一般情况下，假设服务器发送 3 个字节”ABC”,所以数据字段长度为 3,。服务器会存储这个数据，并从头开始复制到响应包中。在攻击场景下，请求包包含 3 字节数据，但是长度字段可以说是 1003。当服务器生成应答包时，会从头复制数据”ABC”，但它会复制 1003 个字节。额外的 1000 字节来自服务器的内存中，其中就包括用户名、密码、用户活动等信息。具体描述如图 2，图 2 所示：

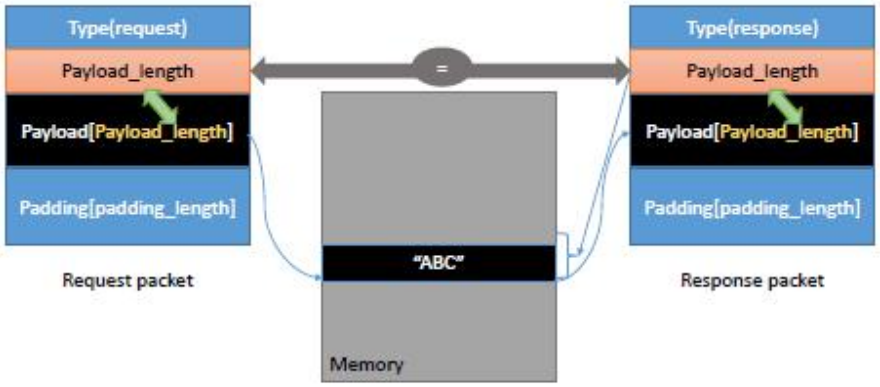


图 2 正常数据包

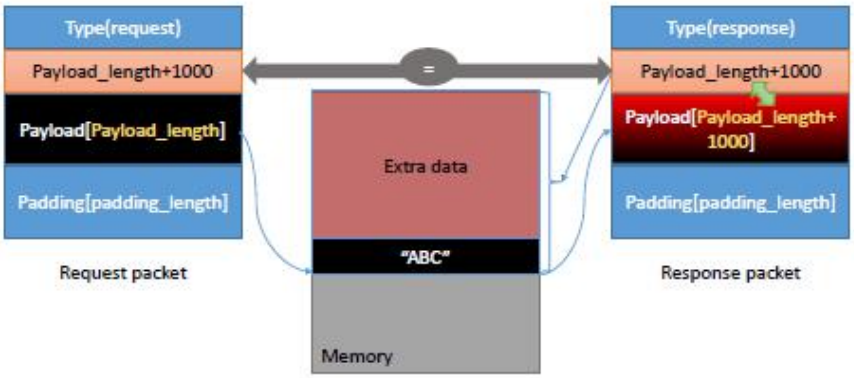


图 3 攻击连接

我们的攻击代码允许实验者使用不同的有效载荷长度，默认为（0x4000）可以通过以下命令来缩小它：

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

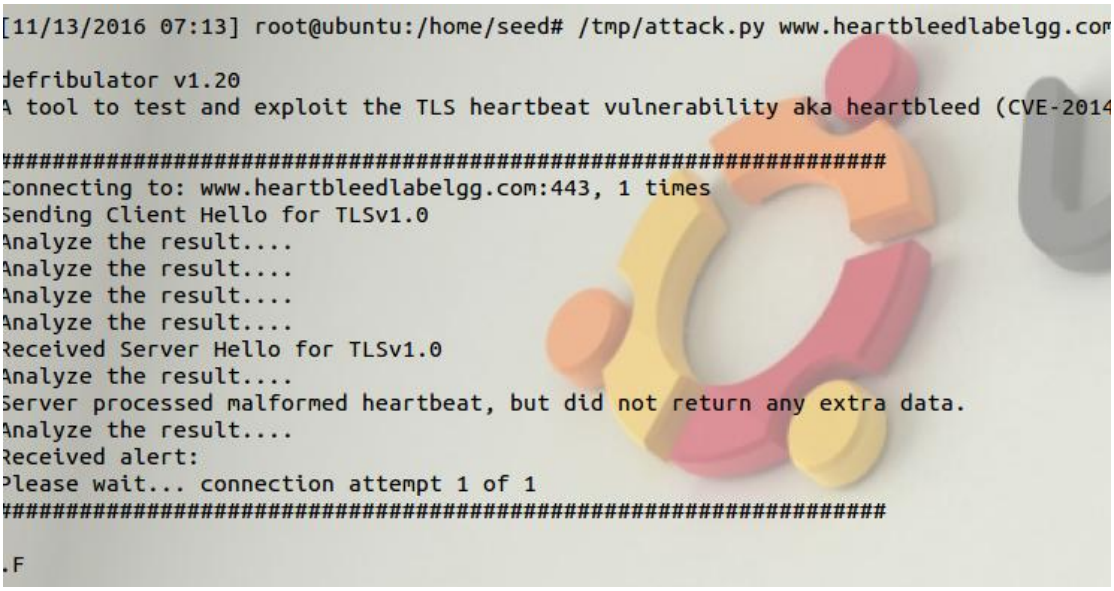
$./attack.py www.heartbleedlabelgg.com --length 83

测试后，请回答下列问题：

• **问题 2.1** 随着长度的减小，你观察到什么差异？

随着长度的减小，可以获取的数据越来越少，包含的敏感信息也随之减少。

• **问题 2.2** 随着长度变量的减小，输入长度会有一个临界值，在低于该临界值时，Heartbeat 应答包不包含任何额外附加信息。请尝试寻找临界值，当返回的长度小于预期的长度时，会出现”Server processed malformed Heartbeat, but did not return any extra data.”



通过测试发现 pl=23 时，还是有数据产生。当 pl=22 时，就出现了问题中所描述的语段。所以临界值是 23。

3.3 任务 3：修复漏洞

修复漏洞的最好方式，是更新到最新版本的 OpenSSL 库。这可以通过以下命令实现更新：

#sudo apt-get update

#sudo apt-get upgrade

• **任务 3.1** 更新后尝试攻击，你发现了什么？

更新后，再用之前的代码，无论尝试多少次，返回值只有.F，并没有攻击成功。

• **任务 3.2** 研究源码，找出问题所在（自己分析）。

（1）下面的代码主要是获取数据的长度，并把报文的类型赋值给 hbtype：

```
// Read from type field first
hbtype = *p++; /* After this instruction, the pointer
  * p will point to the payload_length field *.

// Read from the payload_length field
  // from the request packet
n2s(p, payload); /* Function n2s(p, payload) reads 16 bits
  * from pointer p and store the value
  * in the INT variable "payload". */
  pl=p; // pl points to the beginning of the payload content
```

（2）下面的代码是对收到的报文，进行类型的判断，如果是 request，就构造 response 进行回复：

```
if (hbtype == TLS1_HB_REQUEST)
 {
unsigned char *buffer, *bp;
 int r;

 /* Allocate memory for the response, size is 1 byte
 * message type, plus 2 bytes payload length, plus
 * payload, plus padding
 */

buffer = OPENSSL_malloc(1 + 2 + payload + padding);
    bp = buffer;
```

这里的问题是：当客户端发送的 Request 报文的实际数据部分长度并不等于 payload，而是实际数据的长度小于 payload 值。假设发送 Request 报文时 payload 填充的是 65535，因为长度字段占用两个字节，最大可以是 65535byte=64kB。这样在分配内存的时，payload 的值等于 65535，这里没有对数据包的实际长度进行验证，造成分配的内存大小大于客户端发送过来的数据大小：

buffer = OPENSSL_malloc(1 + 2 + payload + padding)；这段内存区域最大为：65535 + 1 + 2 + 16

在正常的情况下，response 报文中的 data 就是 request 报文中的 data 数据，但是在异常情况下，payload 的长度远大于实际数据的长度，这样就会发生内存的越界访问，但这种越界访问并不会直接导致程序异常，（因为这里直接 memcpy 后，服务器端并没有使用 copy 后的数据，而只是简单的进行了回复报文的填充）这里使用了 memcpy 函数，该函数会直接根据长度把内存中数据复制给另一个变量。这样就给恶意的程序留下了后门，当恶意程序给 data 的长度变量赋值为 65535 时，就可以把内存中 64KB 的数据通过 Response 报文发送给客户端，这样客户端程序就可以获取到一些敏感数据。

所以可以加入输入检测，如果用户的输入长度大于 length，就认为是非法的。

4.参考文献

[1] Heartbleed attack - Implementation:
https://alexandreborgesbrazil.files.wordpress.com/
2014/04/hearbleed attack version a 1.pdf
[2] Heartbleed attack - Interesting explanation: http://xkcd.com/1354/

心得体会

很开心自己真的去做个 seed project 上的实验，这真的是一个很不错的网站，自带的虚拟机配置好环境，同时附带了配置方法，可以供高阶人员使用。这次实验，我最初的目标是做一个简单级别的和一个中等级别的。但最后却成功完成了两个简单级别的实验。

第一个实验我选择了 heartbleed，理解起来很容易，以前的我不是很关注安全方面的事情，这次为了深刻理解 Heartbleed，通过百度了解到这曾是一个很严重的漏洞问题，泄露了不少隐私数据，涉及众多网站。在理解了实验指导书后，我发现我对 Linux 指令了解的不够多，想通过命令行修改 hosts 文件也是通过翻阅书籍才知道的，这方面还有待加强。当成功看到有数据显示时，开心又惊讶，于是又翻阅了一次 heartbeat 协议的工作方式，去理解这个漏洞存在的原因，以及攻击者为何可以获得数据。由于实验室提供了现成的攻击代码，让实验变得简单，浏览攻击代码大体理解了思路，但要真正写出自己的攻击代码，还需对编程语言以及 heartbeat 协议有更进一步的了解。

总之，通过这次实验，发现自己的动手实践能力真的有待提高，学习不能只是理论，要多去动手实践操作才能加深对理论的印象和理解。最后，感谢王老师以及学长们的帮助和点评。