

Heartbleed Attack

实验报告

|  |  |
| --- | --- |
| 学生姓名 | maybeLocalhost |
| 学 号 |  |
| 专业班级 |  |
| 指导教师 |  |
| 学 院 | 计算机学院 |
| 完成时间 | 2020.10 |

目录

[一、实验简介 1](#_Toc58507491)

[1.1 实验背景 1](#_Toc58507492)

[1.2 实验目的 1](#_Toc58507493)

[1.3 实验内容 1](#_Toc58507494)

[1.4 实验环境 1](#_Toc58507495)

[二、实验原理 2](#_Toc58507496)

[三、实验过程 2](#_Toc58507497)

[3.1 Task1：发动Heartbleed攻击 2](#_Toc58507498)

[3.2 Task2：找到Heartbleed漏洞成因 4](#_Toc58507499)

[3.3 Task3：修复Heartbleed漏洞 6](#_Toc58507500)

[四、实验结果与结论 7](#_Toc58507501)

[五、实验思考与讨论、问题的解决 8](#_Toc58507502)

[5.1 实验思考题 8](#_Toc58507503)

[六、参考文献 8](#_Toc58507504)

# 一、实验简介

## 1.1 实验背景

Heartbleed错误（CVE-2014-0160）是OpenSSL库中的一个严重实施缺陷，攻击者可以利用该缺陷从受害者服务器的内存中窃取数据。被盗数据的内容取决于服务器内存中的内容。它可能包含私钥，TLS会话密钥，用户名，密码，信用卡等。漏洞存在于Heartbeat协议的实现中，SSL / TLS使用该协议使连接保持活动状态。

## 1.2 实验目的

本实验的目的是让学生了解此漏洞的严重程度，攻击的工作方式以及如何解决问题。受影响的OpenSSL版本范围是1.0.1至1.0.1f。SEEDUbuntu 12.04 VM中的版本为1.0.1。

## 1.3 实验内容

本实验涵盖以下主题：

1. 发动Heartbleed攻击
2. 找到Heartbleed漏洞成因
3. 修复Heartbleed漏洞

## 1.4 实验环境

Ubuntu 12.04

攻击机IP：192.168.120.140

受害机IP：192.168.120.141

# 二、实验原理

心跳协议由两种消息类型组成：HeartbeatRequest数据包和HeartbeatResponse数据包。客户端将HeartbeatRequest数据包发送到服务器。当服务器接收到它时，它将在HeartbeatResponse数据包中发回接收到的消息的副本。目的是使连接保持活动状态。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。

Heartbleed漏洞之所以得名，是因为用于安全传输层协议（TLS）及数据包传输层安全协议（DTLS）的 Heartbeat扩展存在漏洞。Heartbeat扩展为TLS/DTLS提供了一种新的简便的连接保持方式，但由于OpenSSL 1.0.2-beta与OpenSSL 1.0.1在处理TLS heartbeat扩展时的边界错误，攻击者可以利用漏洞披露连接的客户端或服务器的存储器内容，导致攻击者不仅可以读取其中机密的加密数据，还能盗走用于加密的密钥。

# 三、实验过程

## 3.1 Task1：发动Heartbleed攻击

在此任务中，我们将在试验机的社交网站上发起Heartbleed攻击，并查看可以实现什么样的损害。Heartbleed攻击的实际损害取决于服务器内存中存储的信息类型。发动攻击的前提是, 受害者的内存中有相关的内容。如果服务器上没有太多活动，则将无法窃取有用的数据。因此，我们需要作为合法用户与Web服务器进行交互。让我们以管理员身份进行操作，然后执行以下操作：

1. 访问 https://www.heartbleedlabelgg.com
2. User Name：admin; Password：seedelgg 登录
3. Go to More -> Members and click Boby -> Add Friend 添加朋友
4. 给 Boby 发一条私人信息



图 1 给Boby发送消息

现在受害者的内存充实了起来，可以开始进行攻击。多次运行SEEDLab提供的attack.py，并寻找以下三个内容：

* 用户名和密码；
* 用户的活动（用户所做的事情）；
* 私人消息的确切内容。

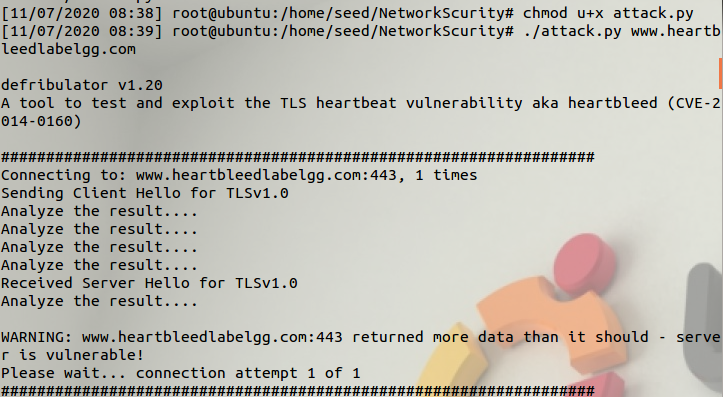


图 2 运行attack.py

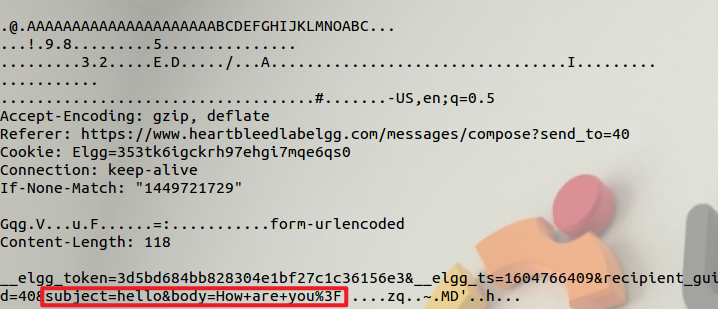


图 3 找到消息内容



图 4 找到消息内容

## 3.2 Task2：找到Heartbleed漏洞成因

Heartbleed攻击基于Heartbeat请求.这个请求只是发送一些数据到服务器,服务器会将数据复制到它的响应数据包中,所有的数据都会被回显.

在正常情况下,假设请求包含3个字节的数据“ABC”,长度字段的值为3.服务器将数据放入内存中,并从数据的开头复制3个字节到其响应包.

在攻击场景中,请求可能包含3个字节的数据,但长度字段可能表示为1003.当服务器构造其响应数据包时,它从数据的起始处(即“ABC”)复制,但它复制1003字节,这些额外的1000字节显然不是来自请求包,它们来自服务器的私有内存,并且可能包含用户名,密码等隐私数据.

* **良性请求：**

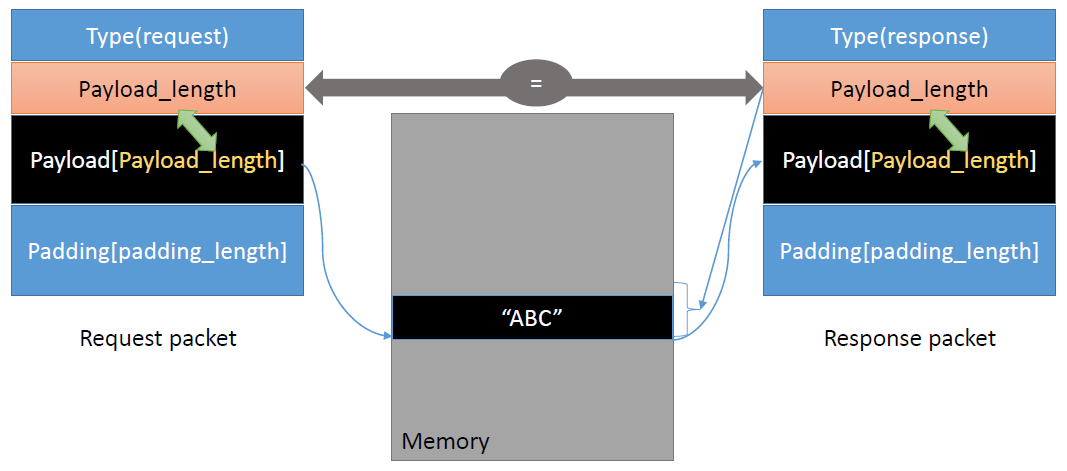


图 5 良性请求

* **恶意请求：**

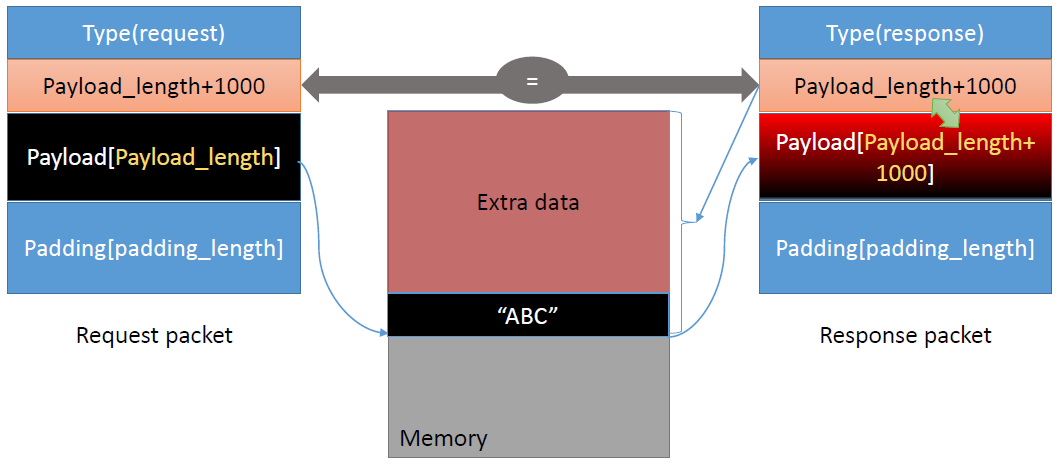


图 6 恶意请求

尝试不同的payload长度值,当载荷长度减小的时候,我们获取到的额外数据量在减少，当载荷值大于等于22的时候,获取不到额外的数据：

$./attack.py www.heartbleedlabelgg.com --length 22

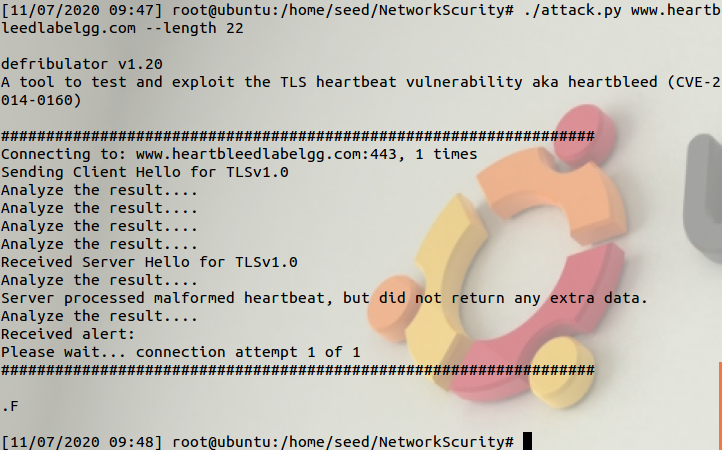


图 7 长度为22时无法获取消息内容

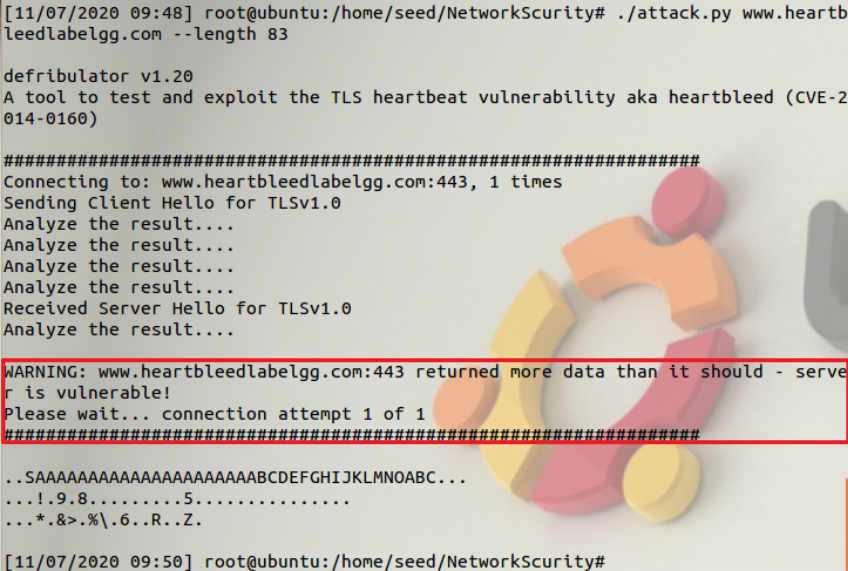


图 8 长度大于83时报错

## 3.3 Task3：修复Heartbleed漏洞

升级OpenSSL：

$ sudo apt-get update

$ sudo apt-get upgrade

$ sudo apt-get dist-upgrade

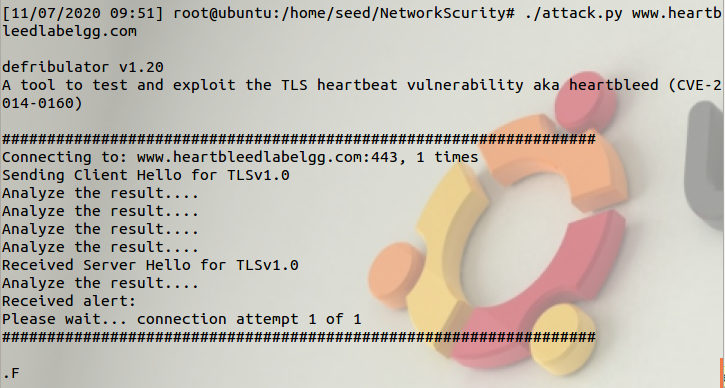


图 9 升级OpenSSL后攻击失败

# 四、实验结果与结论

通过本次实验，我学习了心脏出血攻击的基本使用方法，加深了对边界检查的重要性的认识，并能够进行简单的Heartbleed攻击。根据实验过程，我们可以根据该攻击的过程来制定相应的预防措施。目前，Heartbleed攻击的防范主要从以下四个方面进行：

1. 使用Heartbleed漏洞检测服务；

如Heartbleed test、ossible.lv Heartbleed test、LastPass Heartbleed Checker、Qualys SSL Labs Server Test和Chrombleed等检测工具。

1. 不要在受该漏洞影响的网站上登录帐号，一收到网站的安全修补确认，就立即修改银行、电子邮件等敏感帐号的密码；
2. 修复SSL/TLS的证书架构，尤其是默认证书的吊销过程。这就要协调一致的努力才能得到真正的解决方案中的规定，实施和部署；
3. 使用权限分离的临界加密用于其它的代码部分。

# 五、实验思考与讨论、问题的解决

## 5.1 实验思考题

1. **请评论以下由Alice，Bob和Eva讨论的有关Heartbleed漏洞的根本原因的讨论：Alice认为根本原因是在缓冲区复制过程中缺少边界检查。Bob认为原因是缺少用户输入验证；Eva认为我们只需删除数据包中的长度值即可解决所有问题。**

* 对于爱丽丝：如爱丽丝所说，该漏洞直接由缓冲区复制期间缺少边界检查引起。OpenSSL团队报告中说明了这是当时原因的问题。
* 对于鲍勃：规范化和验证用户输入是一个合理的想法。但是，即使在传输过程中没有恶意意图，数据包中的比特（包括长度字段值）也可能作为任何意外比特而损坏。除了在服务器上完成验证之外，在响应之前在服务器上实施检查仍然至关重要。
* 对于Eva：尽管数据包中某些字段声明的长度不太可靠，但在某种程度上还是有用的，例如数据包验证。尽管OpenSSL团队计划删除心跳内容TLS / SSL的更高版本中的数据包，并且最终实现，有效载荷长度字段（或类似内容）保留在其他SSL / TLS消息数据包结构中。

# 六、参考文献

1. 杜文亮.计算机安全导论：深度实践[M].高等教育出版社:北京,2020:1-
2. 马赛克. 关于HeartBleed漏洞的总结. https://blog.csdn.net/qq\_39112646/article/details/88761331
3. IDF\_Lab. 如何阻止下一次心脏出血漏洞. https://www.freebuf.com/articles/network/38436.html
4. ustc\_sec. 网络安全-Heartbleed攻击. https://www.jianshu.com/p/e92a73e2657e