**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：  （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：actual\_size = floor((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  （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：  （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。  **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。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/>  2、https://blog.csdn.net/krrrr/article/details/5557391 |