**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 | Linux栈溢出远程利用 | 成绩 |  |   **一、实验目的**  （1）了解C语言程序中函数调用时程序栈的变化情况和栈溢出的基本原理  （2）通过实验掌握如何使用远程栈溢出漏洞程序获取远程服务器上带有root权限的shell  （3）熟悉一些基本的linux命令，了解linux下的编程和调试基本工具。  **二、实验内容**  （1）在非root用户下编写exploit程序连接运行在目标机器5050端口的echo\_server程序，该echo\_server程序由echo\_server.c编译得到，有栈溢出漏洞并具备suid标志位属性。通过socket向该程序发送带有shellcode的buffer进行溢出攻击，获取目标机上具有root权限的shell  （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echo\_server程序的源代码echo\_server.c可以发现该程序的主要功能是绑定本机的5050端口进行tcp通信，把从客户端发送过来的字符串再发送回客户端。其中将recv\_buf拷贝到send\_buf时用到了c语言中的sprintf函数，存在栈溢出漏洞。在本实验中，可以利用该栈溢出漏洞构造shellcode填充到程序栈中，并修改sprintf函数执行完后的返回地址至shellcode的起始地址使得程序从sprintf函数返回后继续执行构造的shellcode从而达到攻击目的。  **五、实验步骤及结果**  （1）安装虚拟机：  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作为登入帐号（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echo\_server程序。可使用ls -l echo\_server命令查看，结果如下图：  （2）阅读源码，寻找溢出漏洞存在的地方：  阅读echo\_server.c源码文件，main函数中无溢出点。在其调用的函数中，echo\_sever函数中分别构造了大小为200的send\_buf和大小为2000的recv\_buf。并且echo\_server函数中有两处调用了sprintf函数，如下图所示：左图是第一次调用，右图是第二次调用。第一次调用拷贝的是定长字符串，不会产生溢出。第二次调用拷贝的rcv\_buf大小（2000）远大于send\_buf（200），所以溢出漏洞在第二个sprintf函数上。    （3）编写exploit程序：  使用C语言编写exploit程序echo\_server\_exploit.c对echo\_server程序进行栈溢出攻击。  此程序的大致逻辑是构造需要发送给目标机的buffer，然后创建套接字连接目标机并向echo\_server运行的5050端口发送构造好的buffer。下图是构造buffer部分的代码：  1、大小为1600的buf用于构造带有shellcode的发送内容，此处需要注意的是buf的大小必须是4的倍数（内存对齐原则，下同）。  2、调用sscanf函数从命令行参数中的第三个参数获取修改后的返回地址，此地址根据echo\_server给的esp值计算。  3、调用memset函数将buf的值全部初始化为0x90，即nop指令（程序执行此指令时什么也不做，并继续接下来的指令）。  4、调用memcpy函数将shellcode拷贝到buf末尾的sizeof（shellcode）个字节中。  5、使用一个for循环将buf前360个字节赋值为返回地址（ret），此步需要注意的是覆盖的范围应该大于200（echo\_server中send\_buf的大小），还需要注意的就是这个值也需要是4的倍数。由此buf中的内容设置完毕，其布局如下图所示：  6、创建tcp套接字连接远程服务器并将buf内容发送给服务器的5050端口。  （4）编译运行echo\_server\_exploit.c获取root权限的shell并验证：  使用gcc -o echo\_server\_exploit echo\_server\_exploit.c命令编译得到攻击程序echo\_server\_exploit，然后使用./echo server &命令后台运行echo\_server程序。得到esp值为：0xbfffea78，如下图。  最后使用./ echo\_server\_exploit 127.0.0.1 5050 0xbfffeb78（0xbfffea78 + 0x100（4的倍数））命令运行攻击程序，结果如下图：  从上图中可以看到数据的发送和接收是正常的：服务端收到客户端的数据并发送回去。exploit程序没有成功拿到shell，失败原因应该是新的返回地址没有计算正确。  **六、实验总结**  通过本次实验，我了解了远程栈溢出的基本原理，并尝试编写exploit程序利用漏洞获得了远程目标机上具有root权限的shell。但是最后没有成功。本次实验的难点主要有两个：一是新返回地址的计算，即exploit程序中的第三个参数值。二是需要将echo\_server程序后台运行，解决方法是在./echo\_server命令末尾加一个&。  **七、参考资料**  1、<https://blog.csdn.net/tyskfs2/article/details/42318531> |