课程名称	保密技术基础 A		实验课时	4
实验项目名称和序号	实恶意代码攻防实验	4	同组者姓　名	无
实验目的	通过本实验初步了解缓冲区溢出攻击的基本实现方法。实验过程中，学生需 要将实验的结果记录下来，并回答相关思考题，填写到实验报告中。			
实验环境	实验设备：Windows XP 系统，VMWare 系统，Windows 2000/XP 虚拟机。			
实验内容和原理	**实验内容：** 以下实验内容可根据实验室的具体情况和课时安排的变化进行适当的调整， 实验内容中的思考题以书面形式解答并附在实验报告的后面。本次实验的主要内 容为溢出攻击模拟程序的编写、调试； 编写简单的溢出攻击程序，编译后分别在实验主机和虚拟机中运行。 **实验原理：** 通过往程序的缓冲区写超出其长度的内容，造成缓冲区的溢出，从而破坏程序的堆栈，造成程序崩溃或使程序转而执行其它指令，以达到攻击的目的。造成缓冲区溢出的原因是程序中没有仔细检查用户输入的参数。 下面通过一个示例（图 1）来详细看看什么是缓冲区溢出。程序的缓冲区就像一个个格子，每个格子中存放不同的东西，有的是命令，有的是数据，当程序需要接收用户数据，程序预先为之分配了 4 个格子（下图 1 中黄色的 0~3 号格子）。按照程序设计，就是要求用户输入的数据不超过 4 个。而用户在输入数据时，假设输入了 16 个数据，而且程序也没有对用户输入数据的多少进行检查，就往预先分配的格子中存放，这样不仅 4 个分配的格子被使用了，其后相邻的 12 个格子中的内容都被新数据覆盖了。这样原来 12 个格子中的内容就丢失了。这时就出现了缓冲区（0~3 号格子）溢出了。			

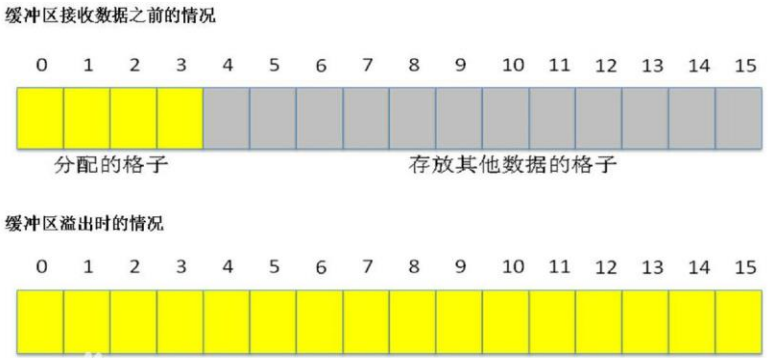


图 1 缓冲区溢出样例

在上面示例的基础上来看一个代码实例，程序如下：

```
void function(char *str) {
char buffer[16];
strcpy(buffer,str);
}
```

上面的 strcpy()将直接把 str 中的内容 copy 到 buffer 中。这样只要 str 的长度大于 16，就会造成 buffer 的溢出，使程序运行出错。存在象 strcpy 这样的问题的标准函数还有 strcat()，sprintf()，vsprintf()，gets()，scanf()等。

当然，随便往缓冲区中填东西造成它溢出一般只会出现“分段错误”（Segmentation fault），而不能达到攻击的目的。最常见的手段是通过制造缓冲区溢出使程序运行一个用户 shell，再通过 shell 执行其它命令。如果该程序有 root 或者 suid 执行权限的话，攻击者就获得了一个有 root 权限的 shell，可以对系统进行任意操作了。

缓冲区溢出攻击之所以成为一种常见安全攻击手段其原因在于缓冲区溢出漏洞太普遍了，并且易于实现。而且，缓冲区溢出成为远程攻击的主要手段其原因在于缓冲区溢出漏洞给予了攻击者他所想要的一切：植入并且执行攻击代码。被植入的攻击代码以一定的权限运行有缓冲区溢出漏洞的程序，从而得到被攻击主机的控制权。

在 1998 年 Lincoln 实验室用来评估入侵检测的的 5 种远程攻击中，有 2 种是缓冲区溢出。而在 1998 年 CERT 的 13 份建议中，有 9 份是是与缓冲区溢出有关的，在 1999 年，至少有半数的建议是和缓冲区溢出有关的。在 Bugtraq 的调查中，有 2/3 的被调查者认为缓冲区溢出漏洞是一个很严重的安全问题。

漏洞

缓冲区溢出攻击的目的在于扰乱具有某些特权运行的程序的功能，这样可以使得攻击者取得程序的控制权，如果该程序具有足够的权限，那么整个主机就被控制了。一般而言，攻击者攻击 root 程序，然后执行类似“exec(sh)”的执行代码来获得 root 权限的 shell。为了达到这个目的，攻击者必须达到如下的两个目标：

1. 在程序的地址空间里安排适当的代码。
2. 通过适当的初始化寄存器和内存，让程序跳转到入侵者安排的地址空间执行。

可以根据这两个目标来对缓冲区溢出攻击进行分类。

实验步骤

方　　法

关键代码

实验步骤：

1. 简单原理示例。

编译以下代码，见图 2：

overflow.c

```
1   #include <stdio.h>
2   #include <string.h>
3   char name[]="abcdefghijklmnopqrstuvwxyz";
4   int main() {
5       char buffer[8];
6       strcpy(buffer,name);
7       return 0;
8   }
```

图 2 简单原理示例代码

由于函数 strcpy 没有溢出检查，所以出现了缓冲区溢出的问题。编译运行之后，出现报错如下，见图 3。

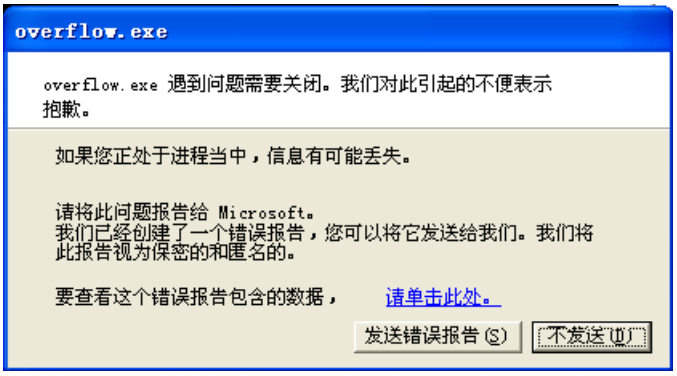


图 3 报错

可以通过查看错误报告中数据（图 4），查看并分析错误。

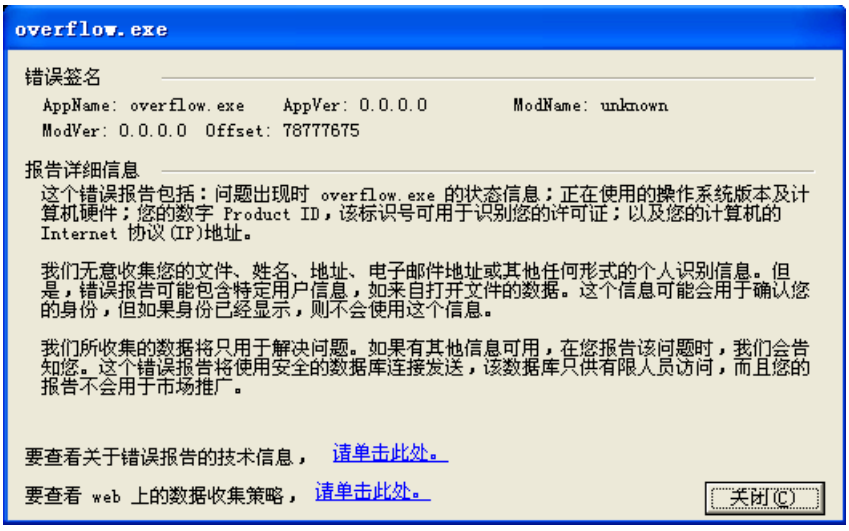


图 4 错误报告

2．溢出攻击模拟示例。

首先写一个 C++程序 2.c，源码如下图 5：

2.cpp

```
1   #include<iostream>
2   using namespace std;
3   int main ( ) {
4       char name[8];
5       cout<<"Please type your name: ";
6       cin>>name;
7       cout<<"Hello, ";
8       cout<<  name;
9       cout<<"\n";
10      return 0;
11  }
```

图 5 2.cpp 源代码

赋值一个名为 name 的字符类型数组（字符串），其内容空间为 8 个字节，运行程序时首先提示使用者输入你的名字，当输入后将该值吸入给 name，然后以“Hello，你的名字\n”的方式输出。编译并运行该程序（编译后的程序为 2.exe）后，此时若“你的名字”小于或等于 8 个字节时程序当然能正 常运行了，但若超过 8 个字节时将出现错误，但是在实际过程中，并没有在超出八个字节报错（图 6），具体原因在阅读汇编代码时再说明。

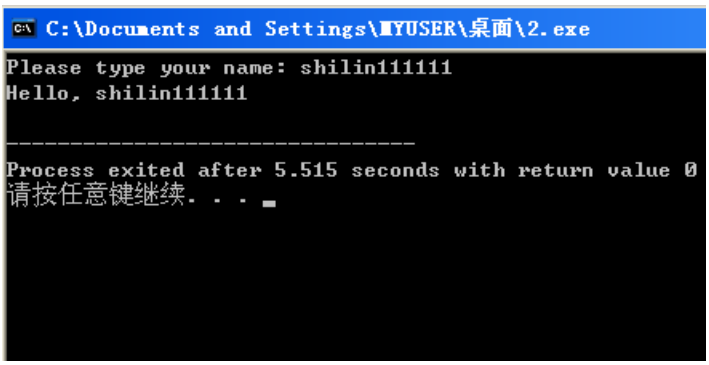


图 6 2.cpp 运行结果（超出 8 字节）

这次我们要做的实验就是让该程序溢出，并能跳转到程序的开头重新运行该 程序。首先我们运行 2.exe，当程序进行至提示用户输入字符串时，输入一个特 殊定制的字符串“aaabbbcccdddeeefffggg”，在弹出的对话框中按“调试”按钮（这里我是用 Ollydbg 作为系统的主调试器的）进入 Ollydbg 调试模式（图 7）：

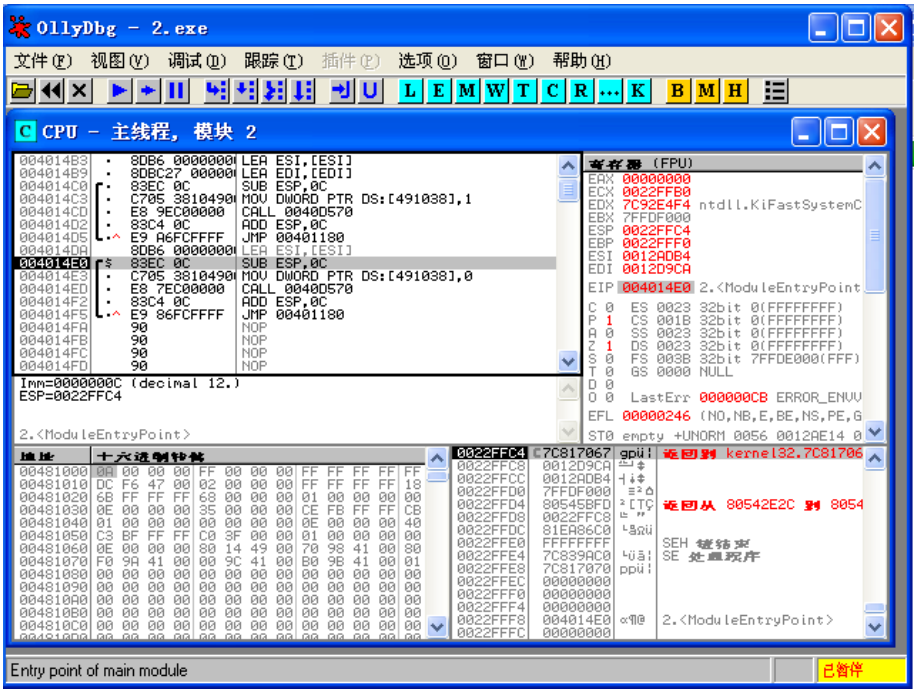


图 7 Ollydbg 调试

这里我们发现负责下一跳的 EIP 寄存器的值被覆盖了，其值为 68686867（图 8），对照 ascii 表后发现其值为“hhhg”，由于小端存储，因此其实是“ghhh” 覆盖了 EIP，现在我们可以确认这个输入的字符串中是从第 21 个字节开始覆盖 EIP 的，共 4 个字节。

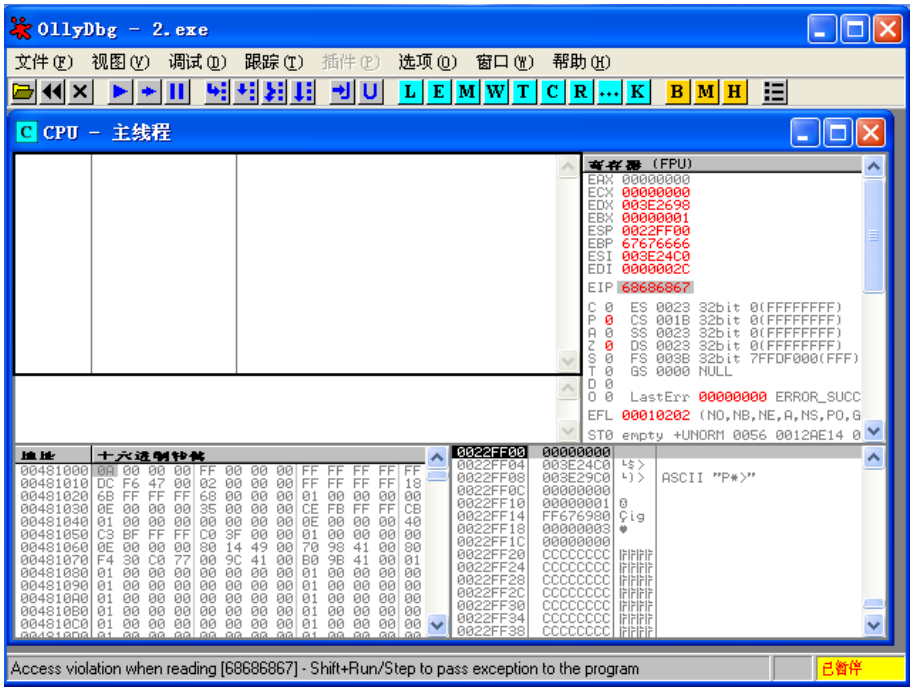


图 8 运行程序

用 Ollydbgc 重新加载 2.exe。

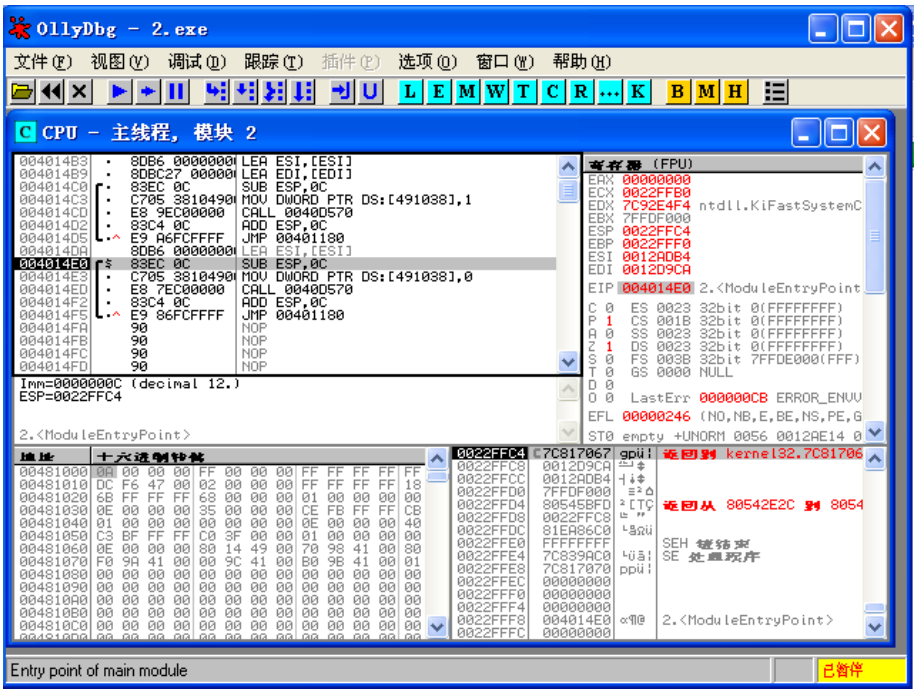
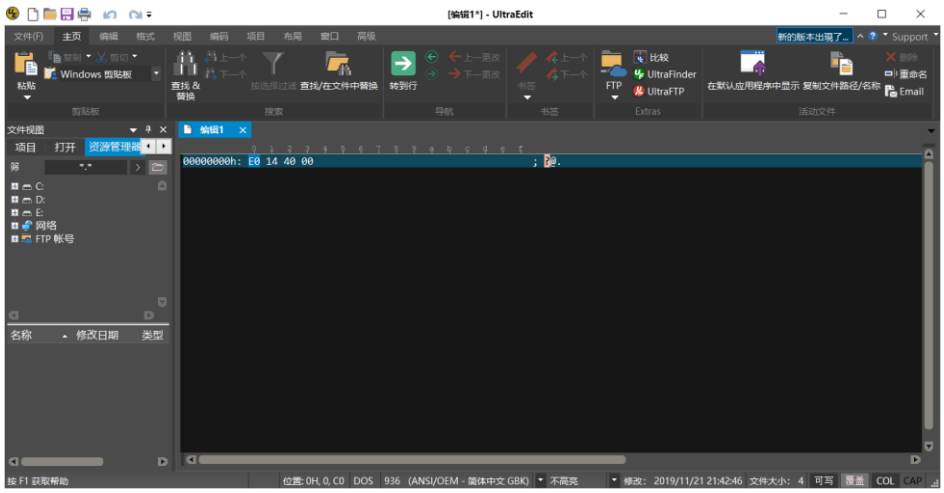


图 9 重新加载 2.exe

我们可以看到该程序起始地址为 004014E0（图 9），由于是小端存储：

00 40 14 E0 等于 E0 14 40 00

此时打开 UltraEdit，输入 1，然后按 Ctrl+H 切换到 HEX 显示模式，然后在 HEX 输出界面中输入 E0 14 40 00，见图 10：



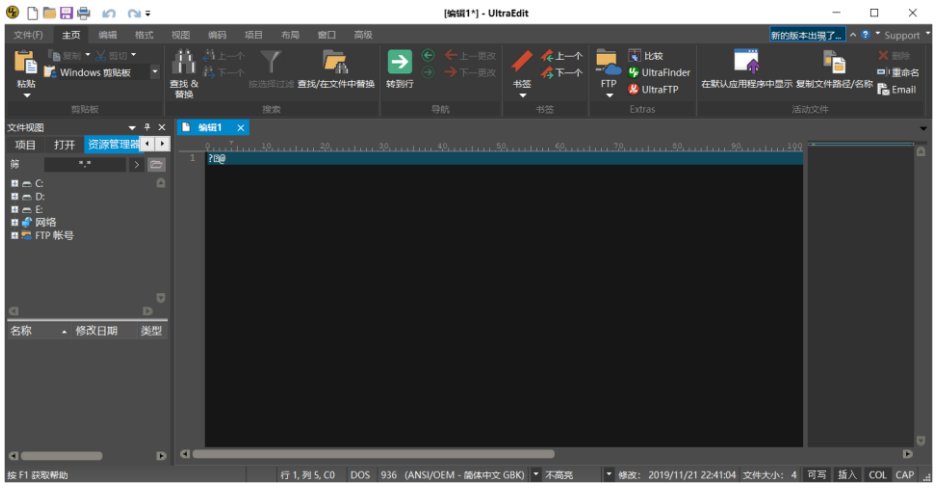


图 10 地址 16 进制转换结果

这时候将原字符串 ghhh 的位置替换成转换出的字符，输出结果见图 11:

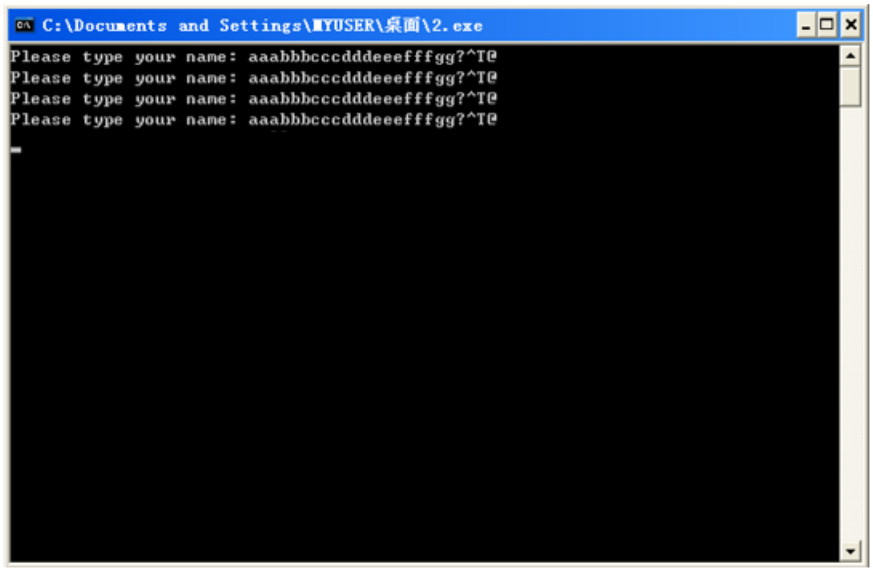
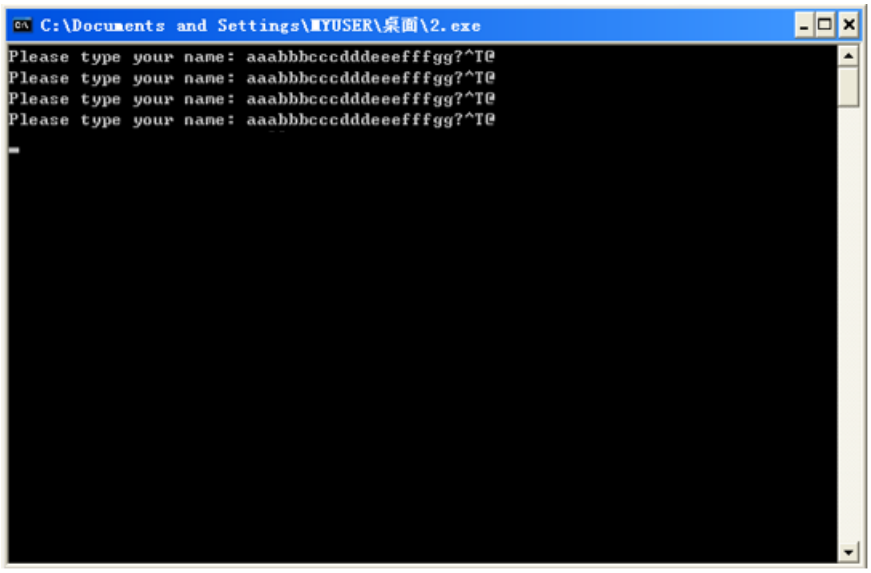


图 11 输出结果

测试记录 分　析 结　论	（具体过程见实验步骤） 对字符串的特定位置字符进行替换再次运行程序，结果如下图所示：



后四位特殊字符覆盖 EIP 的值，而覆盖的值是程序运行的开始地址。这样便会循环执行程序。成功利用缓冲区溢出实现了程序返回地址的更改。

思考题：

溢出攻击提升权限是如何实现的？

最常见提升权限的手段是通过制造缓冲区溢出使程序运行一个用户 shell，再通过 shell 执行其它命令。如果该程序属于 root 且有 suid 权限的话，攻击者就获得了一个有 root 权限的 shell，可以对系统进行任意操作了。

小　结

缓冲区溢出攻击是利用缓冲区溢出漏洞所进行的攻击行动。缓冲区溢出是一种非常普遍、非常危险的漏洞，在各种操作系统、应用软件中广泛存在。利用缓冲区溢出攻击，可以导致程序运行失败、系统关机、重新启动等后果。缓冲区溢出中，最为危险的是堆栈溢出，因为入侵者可以利用堆栈溢出，在函数返回时改变返回程序的地址，让其跳转到任意地址，带来的危害一种是程序崩溃导致拒绝服务，另外一种就是跳转并且执行一段恶意代码。

经过这次实验操作，对缓冲区溢出攻击有更加深刻的了解。对缓冲区溢出的原理及简单溢出过程和程序运行时的堆栈结构等有了更加深入的学习。对计算机中的实际漏洞进行了一定的学习和实现。

参考文献

[1]侯炳辉，曹慈惠，耒珠, et al. 计算机原理与系统结构，（第二版）[M]. 2002.

[2]ken007.缓冲区溢出详解

[EB/OL].https://www.cnblogs.com/kexianting/p/8805591.html,2018-05-17.

[3]怀揣梦想的大鸡腿.Python 经典栈缓冲区溢出获取 root 权限

[EB/OL].https://blog.csdn.net/dajitui2024/article/details/79396339,2018-02-28.

以下由实验教师填写	
记　事 评　议	
成绩评定	平时成绩_____　　实验报告成绩_____　　综合成绩 _____ 指导教师签名：