2018-12-08

缓冲区溢出漏洞实验报告

# 一、实验背景

1、缓冲区溢出：指程序设计时未考虑缓冲区输入超过最大数据量的数据，使之溢出并破坏程序运行，进而取得程序控制权的异常操作。

2、现有保护机制：

（1）Ubuntu及一些Linux系统使用地址空间随机化来随机堆栈的初始地址，使攻击者无法轻易获得准确内存地址。

（2）shell程序在调用时会放弃特权，这个防护措施在/bin/bash中实现。

# 二、实验目的

利用缓冲区溢出漏洞，通过shellcode获得linux下的root权限。

# 三、实验环境

实验楼（https://www.shiyanlou.com/courses/231）虚拟机

64位Ubuntu Linux

# 四、实验步骤

1、打开终端，输入命令安装用于编译32位C程序的软件包（为了方便观察汇编语句，我们需要做将环境转化为32位）。

*$ sudo apt-get update*

*$ sudo apt-get install -y lib32z1 libc6-dev-i386*

*$ sudo apt-get install -y lib32readline-gplv2-dev*

2、输入命令关闭堆栈随机化。

*$ sudo sysctl -w kernel.randomize\_va\_space=0*

3、输入命令，将/bin/sh删除，同时在/bin下建立一个符号链接文件sh，使它指向目录zsh。从而取消shell程序自动放弃特权（此步防护措施在/bin/bash实现）。

*$ sudo su*

*$ cd /bin*

*$ rm sh*

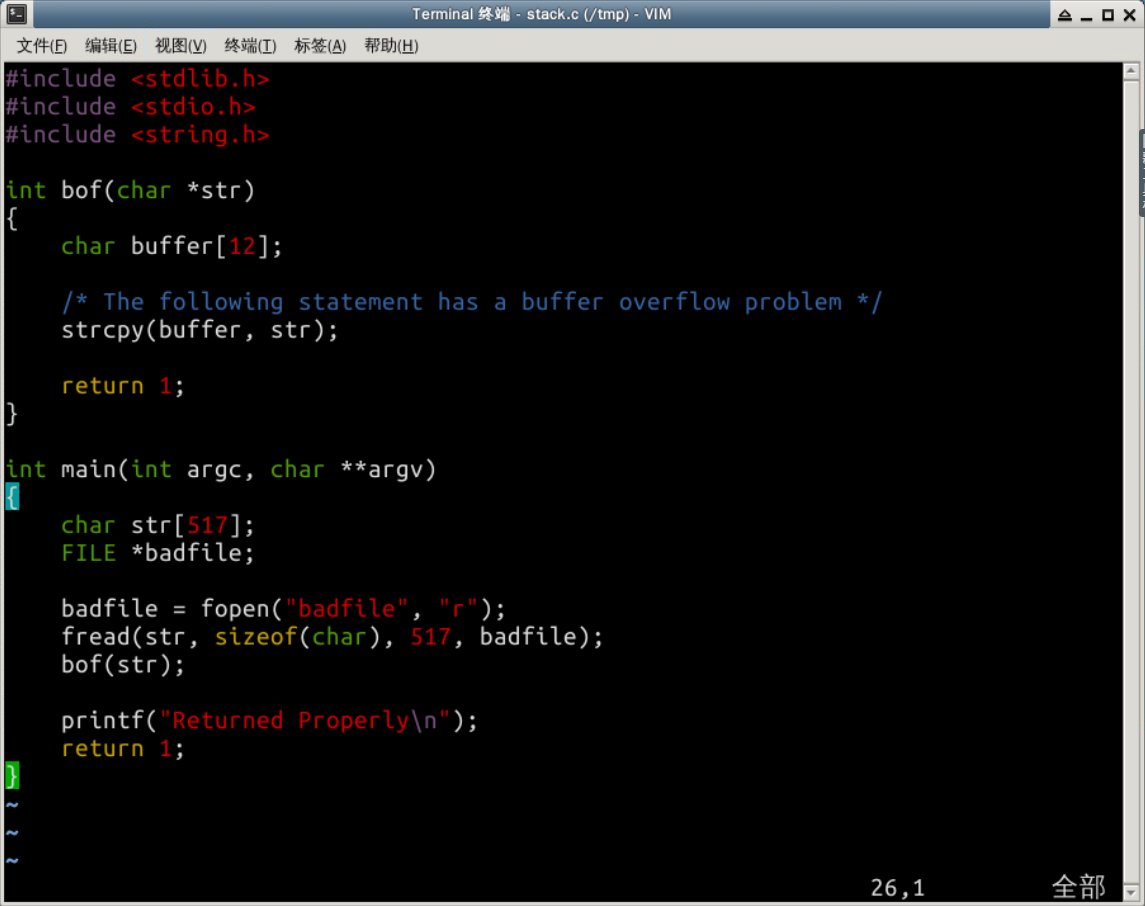
*$ ln -s zsh sh*

*$ exit*

4、输入命令进入32位linux环境并使用bash。

*$ linux32*

*$ /bin/bash*

**5、在/tmp下新建一个stack.c作为我们攻击的目标程序。

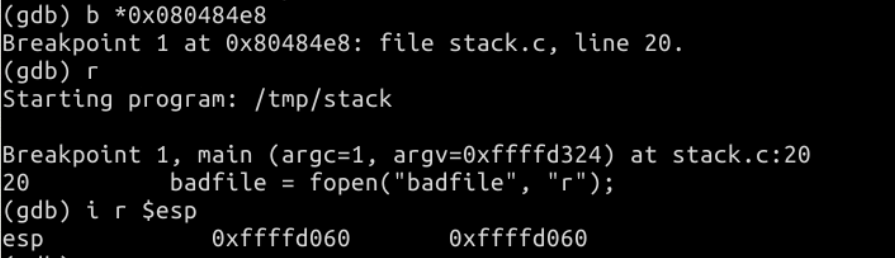
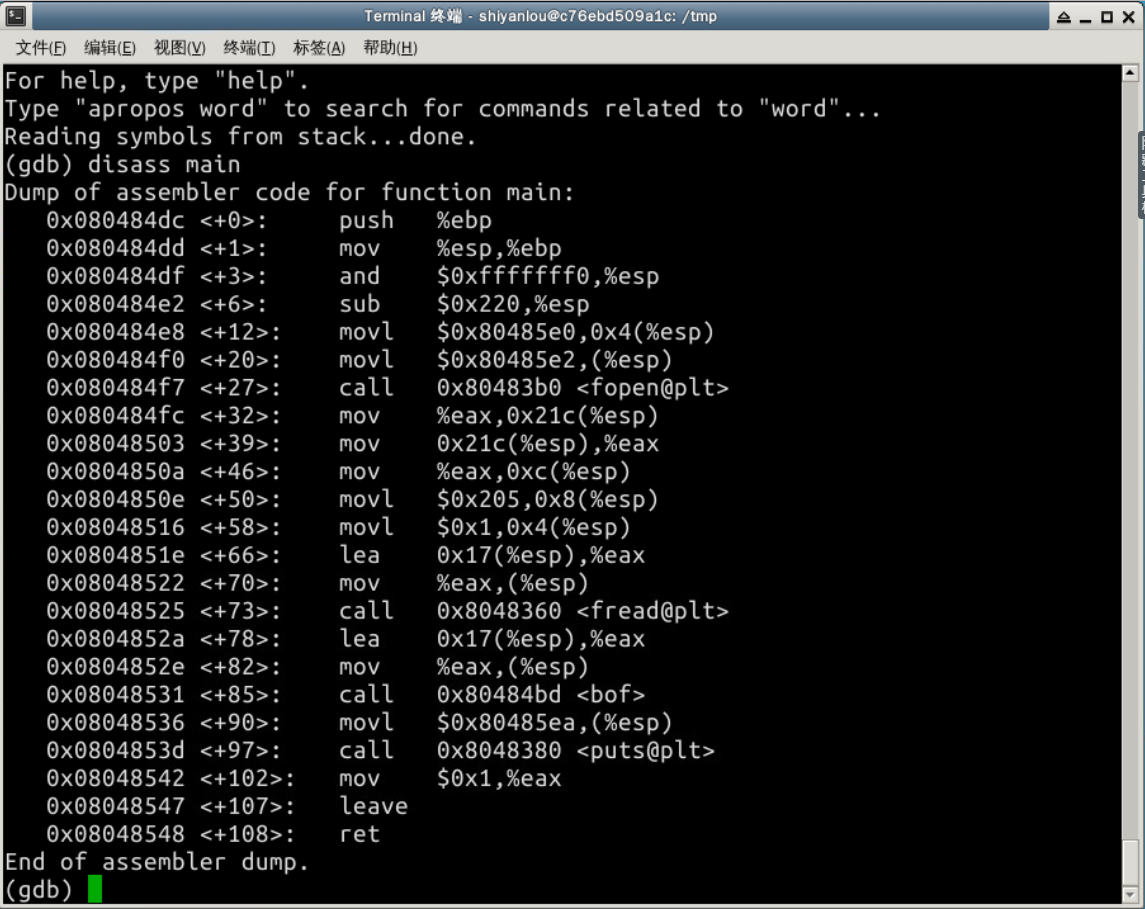
6、输入命令编译stack.c，同时关闭GCC编译器的栈保护机制、允许程序执行栈、使编译后程序可用gdb调试。

*$ sudo su*

*$ gcc -m32 -g -z execstack -fno-stack-protector -o stack stack.c*

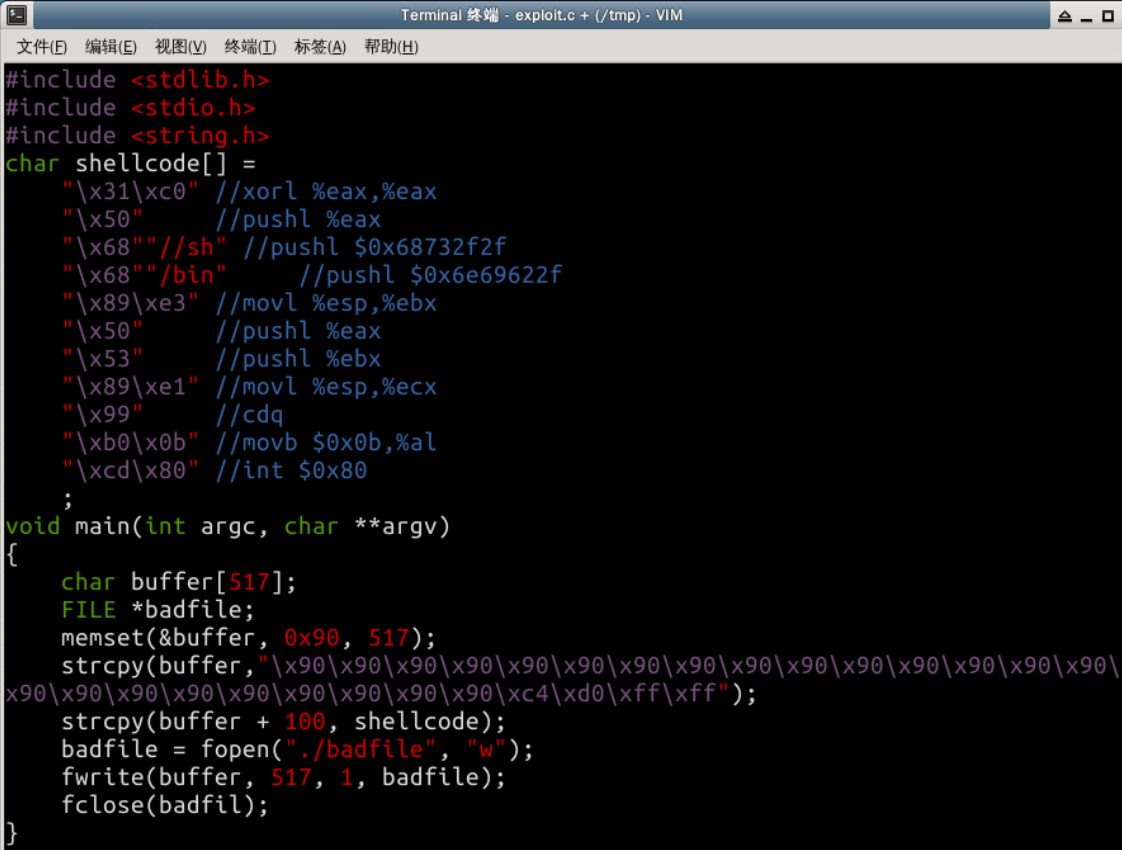
*$ chmod u+s stack*

*$ exit*

**7、使用gdb调试目标程序stack，通过设置断点找到buffer的地址。

如图，buffer地址为0xffffd060

8、通过计算buffer(0xffffd060) + 100(0x64) 得到 shellcode地址为0xffffd0c4

**9、新建一个exploit.c作为我们攻击程序。

10、输入命令编译并运行exploit.c，生成恶意读入文件badfile。

*$ gcc -m32 -o exploit exploit.c*

*$ ./exploit*

11、运行目标程序stack，检查是否获得root权限。

*$ ./stack*

*# whoami*

# 五、实验结果

成功获取root权限。