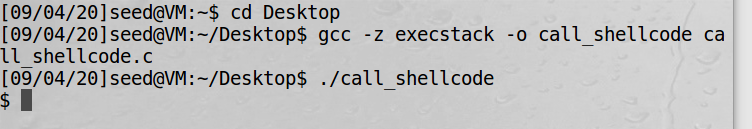
网安实训实验报告

LAB2

57118139顾宸玮

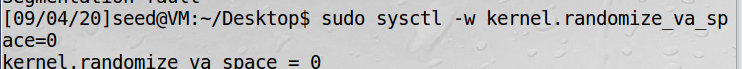
**Task1 Running Shellcode**

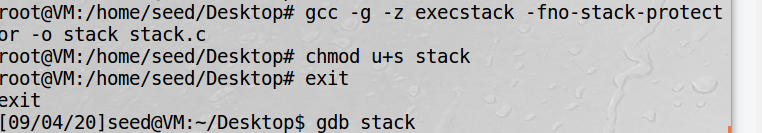


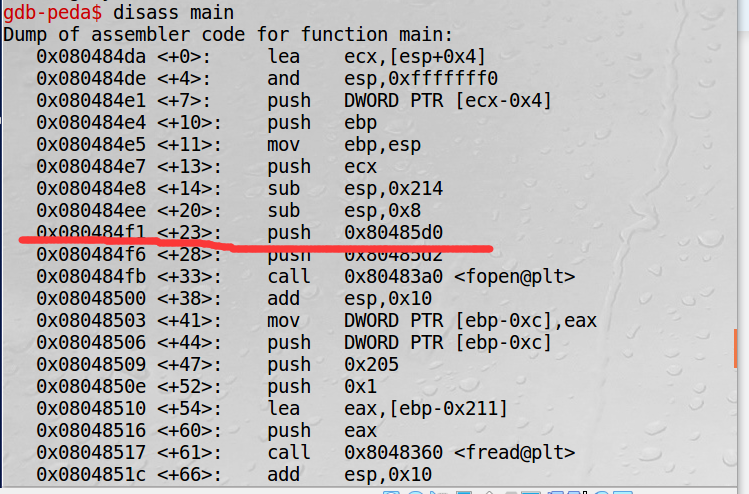
直接编译会出现“segmentation fault”现象，在进行如报告的方法编译后，成功调用shell。Shellcode系统调用了execve()函数来执行/bin/sh。

**Task2 Exploiting the Vulnerability**

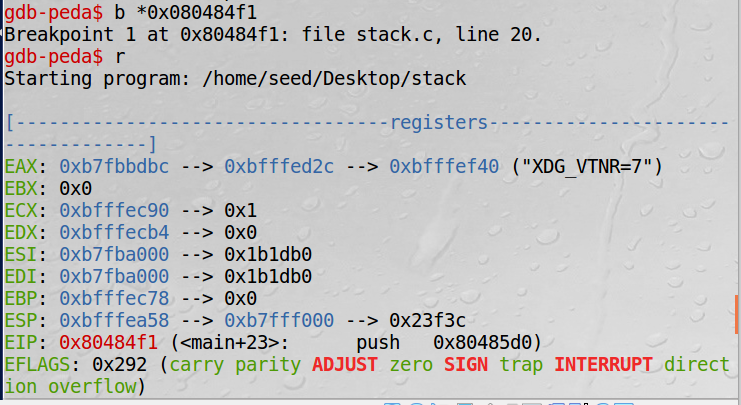
将BUF\_SIZE改为12

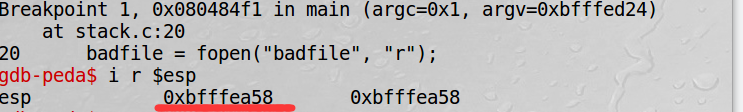




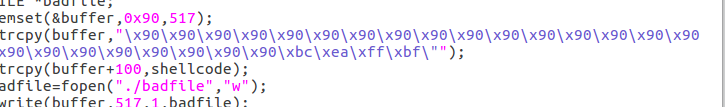


选择如图所示的地址。

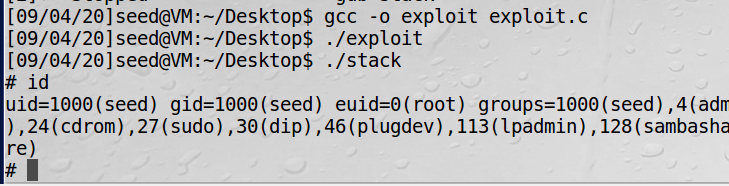




计算0xbfffea58+100（十进制）=0xbfffeabc，填入代码中，



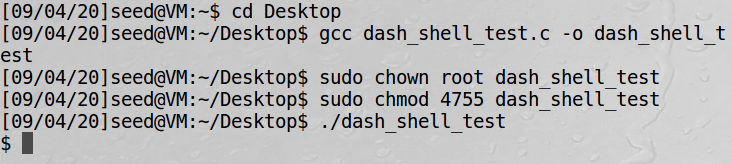
编译成功！



成功获得root权限。

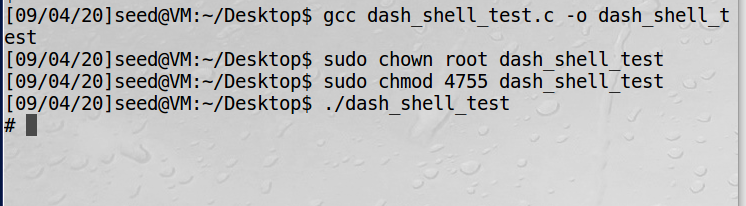
**Task3 Defeating dash’s Countermeasure**





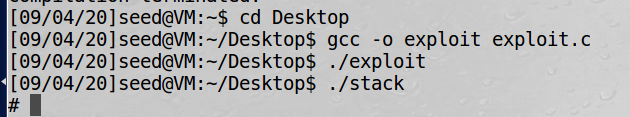
在注释语句前，发现进入了普通用户模式。

下面将语句取消注释：



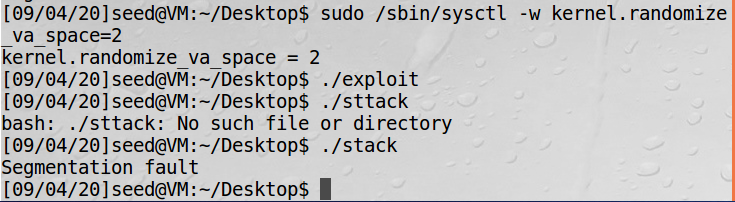
发现进入特权模式。

添加语句后再次进行task2的过程：

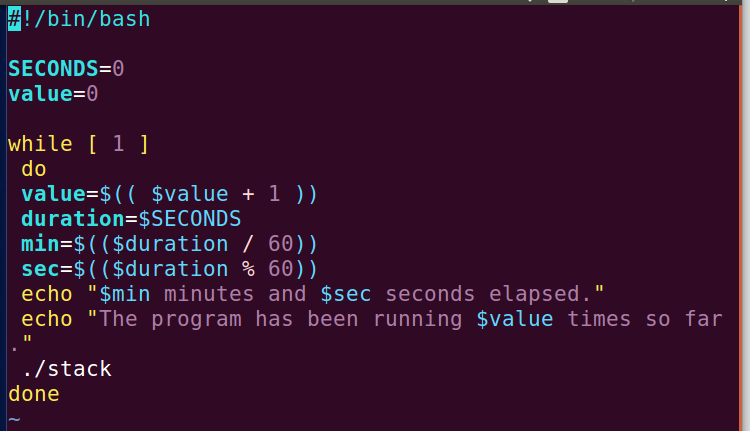


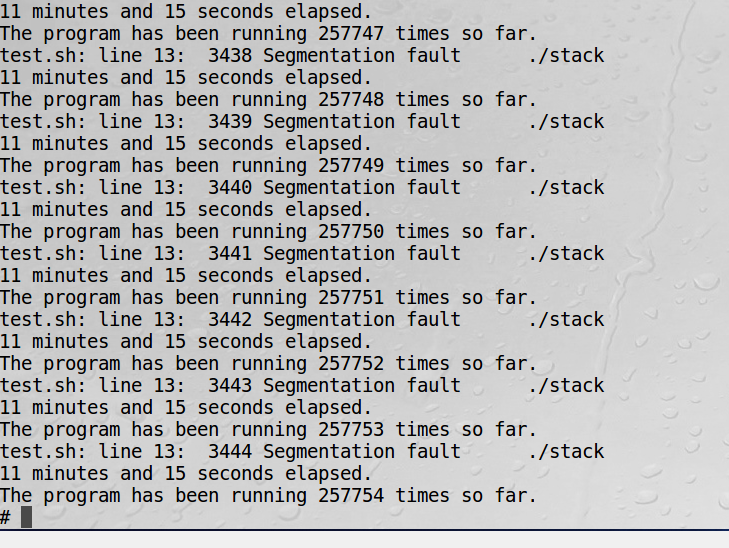
发现进入了特权模式。因为调用了dash程序之前就将victim进程的用户ID改成0，在shell代码中执行execve()前就调用setuid（0）获得权限。从而执行后就进入了root模式。

**Task4 Defeating Address Randomization**



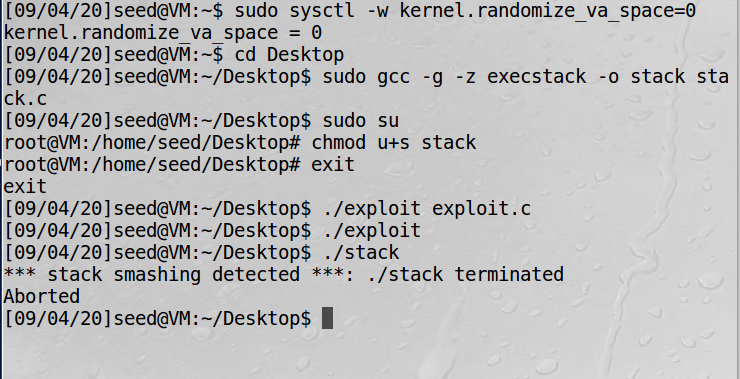
出现段错误，无法获得shell。因为地址随机化，程序运行栈的基地址是随机的，所以执行一次难以执行成功。





开启地址随机化后，在跑了11min,执行了25w次左右命中地址，进入特权模式。

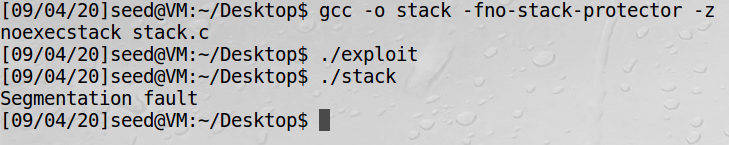
**Task5 Turn on the** **StackGuard Protection**



当启用了StackGuard防止栈溢出，有了保护模式，再次运行task2就不会进入特权模式，而是出现了缓存溢出的错误。

**Task6 Turn on the** **Non-executable Stack Protection**

在关闭了地址随机化后执行以下操作：



不能得到一个shell，显示段错误。通过减小代码段的虚拟地址来区分数据段和代码段，对于缓冲区溢出攻击行为，产生一个通用保护异常而终止进程。

**实验感想**

此次实验主要是针对栈溢出的攻击和防御，Linux有许多防止此攻击的保护模式，比如Non-executable Stack Protection，StackGuard Protection。但是针对某种保护机制也有特殊的攻击方法，所以要多种机制共同作用。Task2让我深切感受到了攻击程序后进入特权模式的危险，对于此课程有了更深的理解。