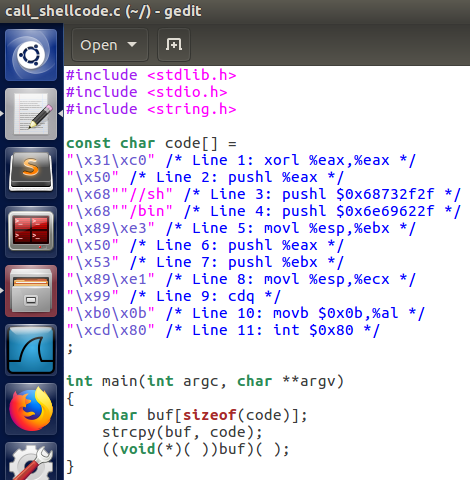
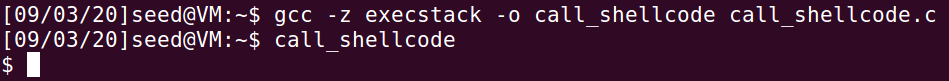
**Lab[2]-report**

57118123刘康辉

**Task 1**

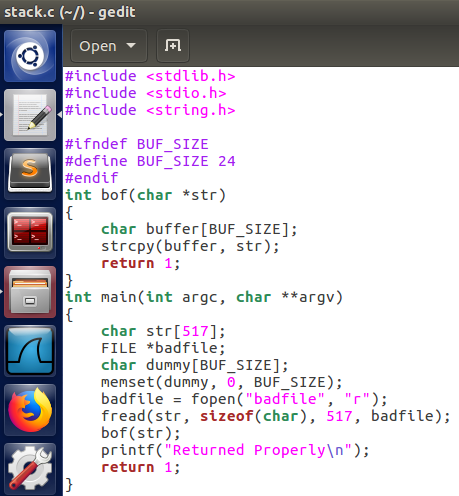


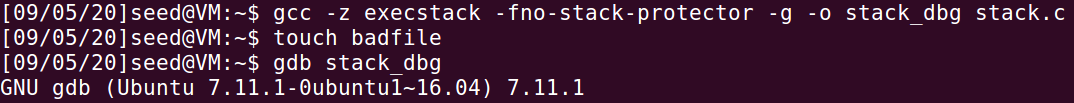
复制实验手册call\_shellcode.c的代码，然后进行编译运行。



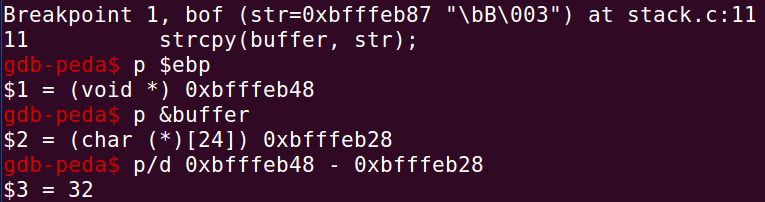
根据实验结果可知，此程序成功调用了shell。

**Task 2**

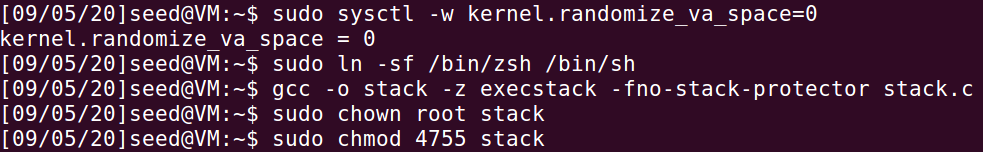




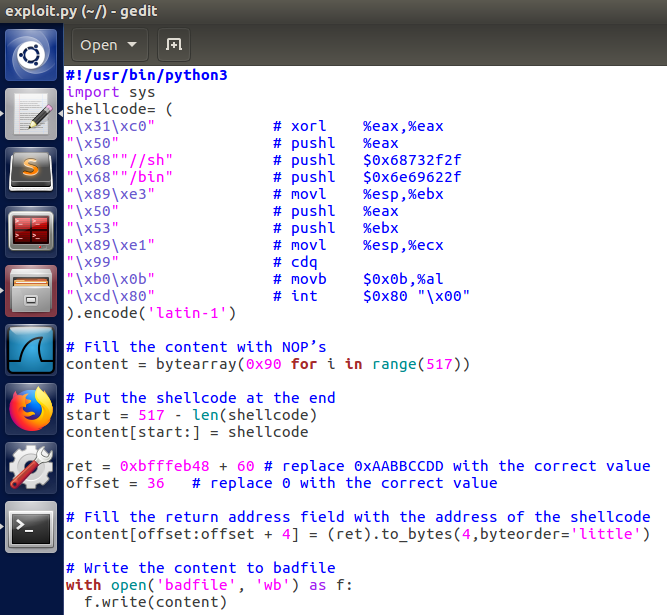




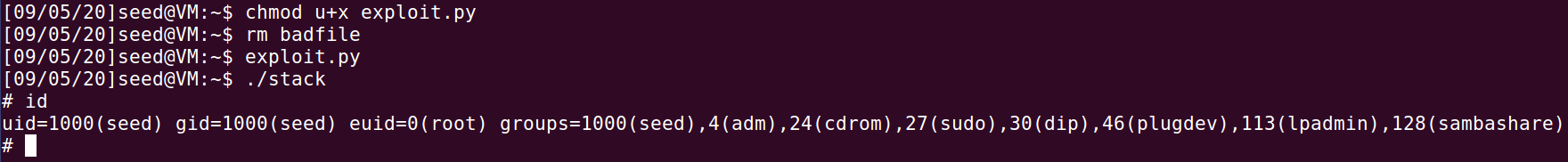
复制实验手册stack.c的代码，数组大小默认为24，然后进入gdb在bof()函数设置断点获取ebp和buffer的地址，计算地址的差值。



关闭Address Space Randomization，将/bin/sh改为指向/bin/zsh，然后编译stack.c文件，设置可执行程序为Set-uid程序。

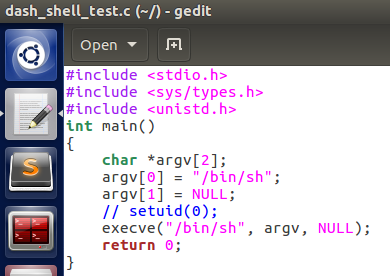


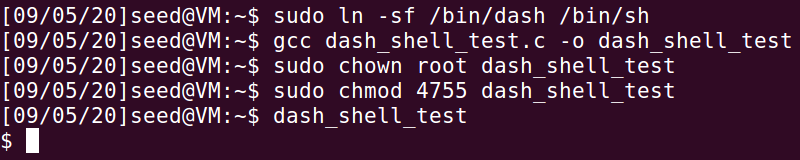
复制实验手册exploit.py的代码，然后根据gdb获取的地址进行修改。

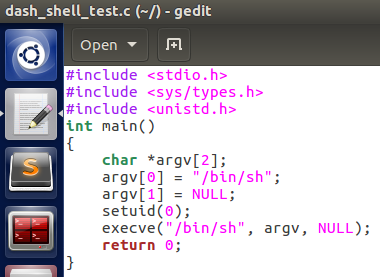


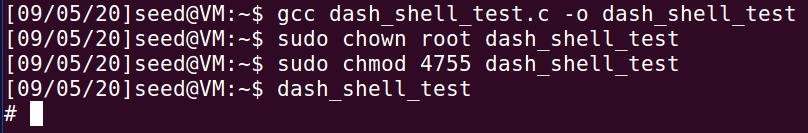
根据实验结果可知，先编译运行过设置为Set-uid程序的exploit可运行程序，然后运行stack程序，发现能够进入root的shell中。

**Task 3**

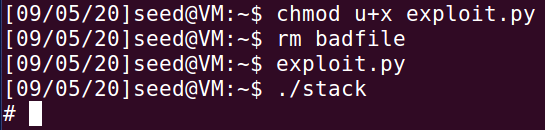
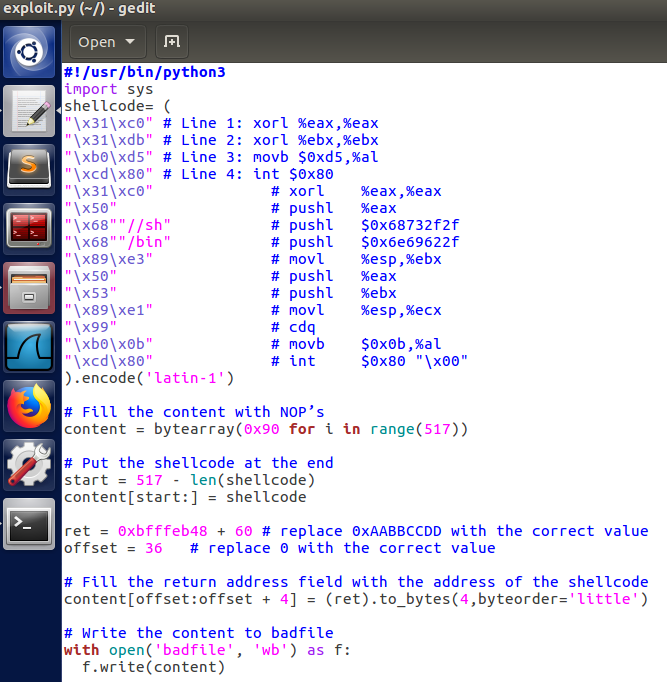






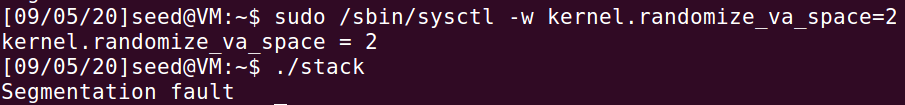


根据实验结果可知，在/bin/dash中，setuid(0)被注释的程序被设置为Set-uid程序后运行不能进入到root的shell中；而setuid(0)未被注释的程序被设置为Set-uid程序后运行能进入到root的shell中。

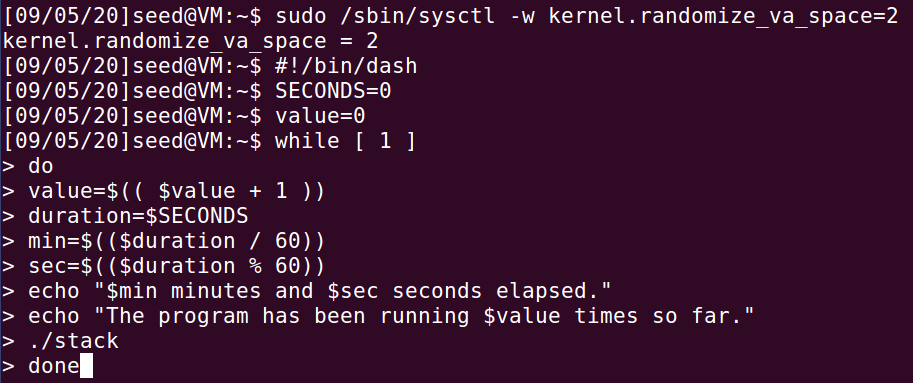


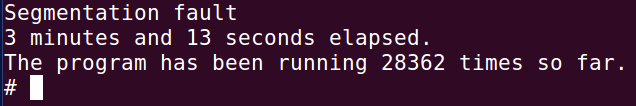
根据实验结果可知，增加实验手册上的代码，然后重新编译运行exploit.py文件，再运行stack能够进入到root的shell中。

**Task 4**



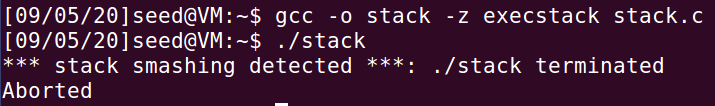
根据实验结果可知，在开启Address Randomization运行stack时，会出现错误，无法进入到shell当中。原因是栈的地址被随机化，根据之前的地址无法找到。





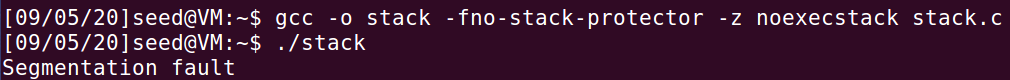
根据实验结果可知，经过3分钟13秒的28362次后攻击成功，进入到shell当中。

**Task 5**



根据实验结果可知，不使用-fno-stack-protector选项，即开启StackGuard Protection进行编译运行stack.c时，会出现错误，无法进入到shell当中。

**Task 6**



根据实验结果可知，使用noexecstack选项，即开启Non-executable Stack Protection进行编译运行stack.c时，会出现错误，无法进入到shell当中。