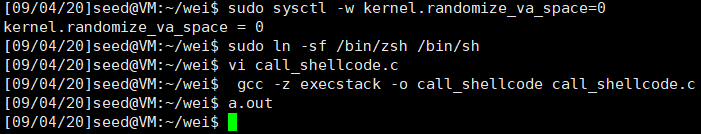
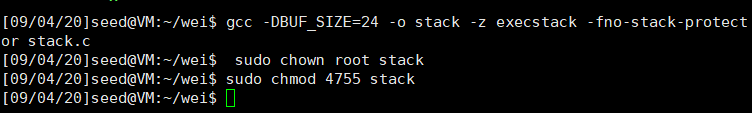
**SeedLab-BufferOverflow**

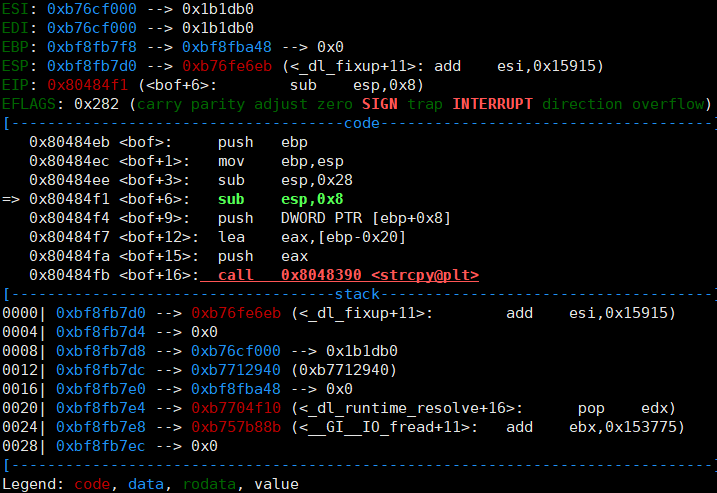
**task1：**



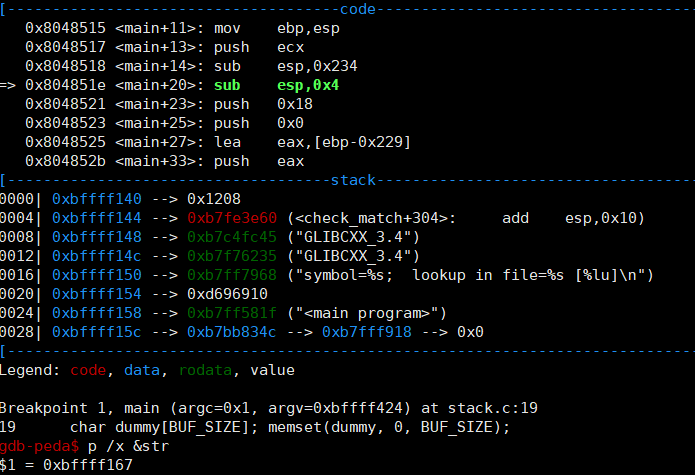


**task2：**

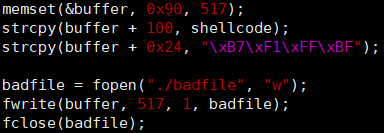
stack.c的bof函数的返回地址为0x24



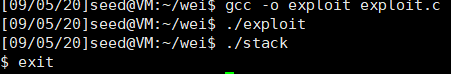
shellcode的返回地址为0xbffff1b7



将exploit.c的后半段代码完善为下图所示

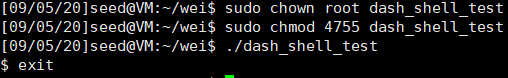


程序执行结果如下

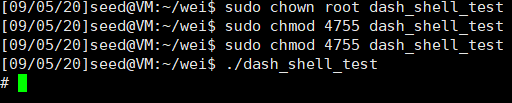


**task3**

未取消注释： 获取的是用户权限的shell



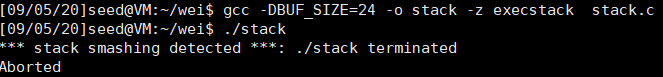
取消注释： 利用setuid(0),获取root权限shell。



在shellcode代码的开头加上如下汇编代码,再次执行stack获得的是root权限

task4:

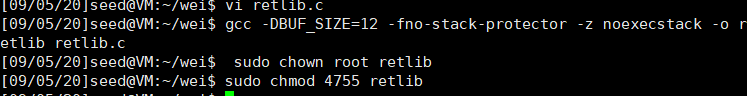
**task5：**去掉-fno-stack-protector选项后stack运行被取消



**task6：**

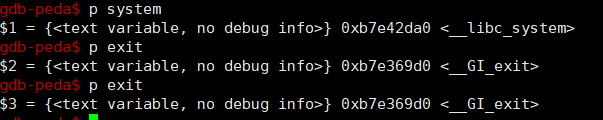
It should be noted that non-executable stack only makes it impossible to run shellcode on the stack, but itdoes not prevent buffer-overflow attacks

**SEED Labs – Return-to-libc Attack Lab**



**task1:**

system函数的地址是 0xb7e42da exit函数的地址是0xb7e369d



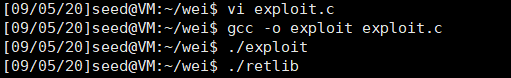
**task2：**



**task3：**

buffer距离ebp的距离为0x14，所以buffer距离return address的距离为0x18（十进制为24），故在exploit.c中将system()在buffer数组中的下标改为24，由于函数调用时在函数自身ebp下 是return address 和参数，故将exit()在buffer数组中的下标置为28，参数/bin/sh在buffer数组中的下标置为32.





**task4：Turning on address randomization**



**task5: Defeat Shell’s countermeasure**



**task6: Defeat Shell’s countermeasure without putting zeros in input**