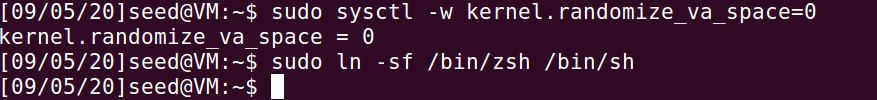
Lab2-report

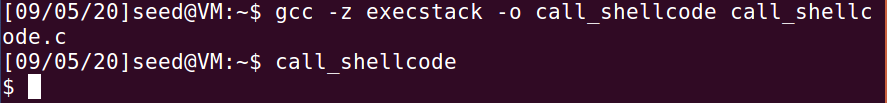
57118113 蔡义涵

**2.1 Turning Off Countermeasures**



**2.2 Task 1: Running Shellcode**

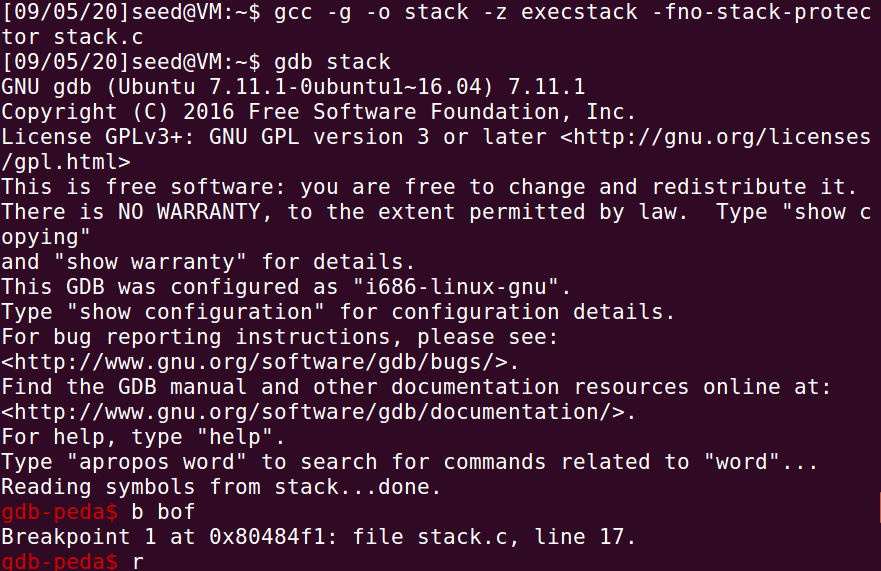


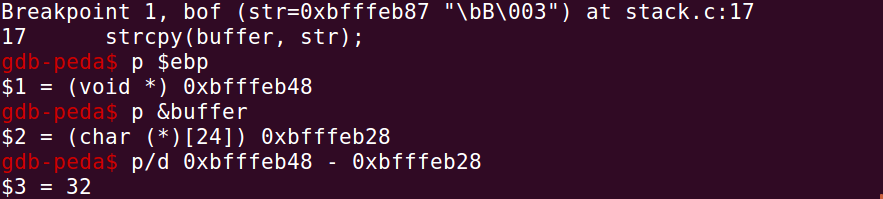


成功启动shell

**2.3 The Vulnerable Program**

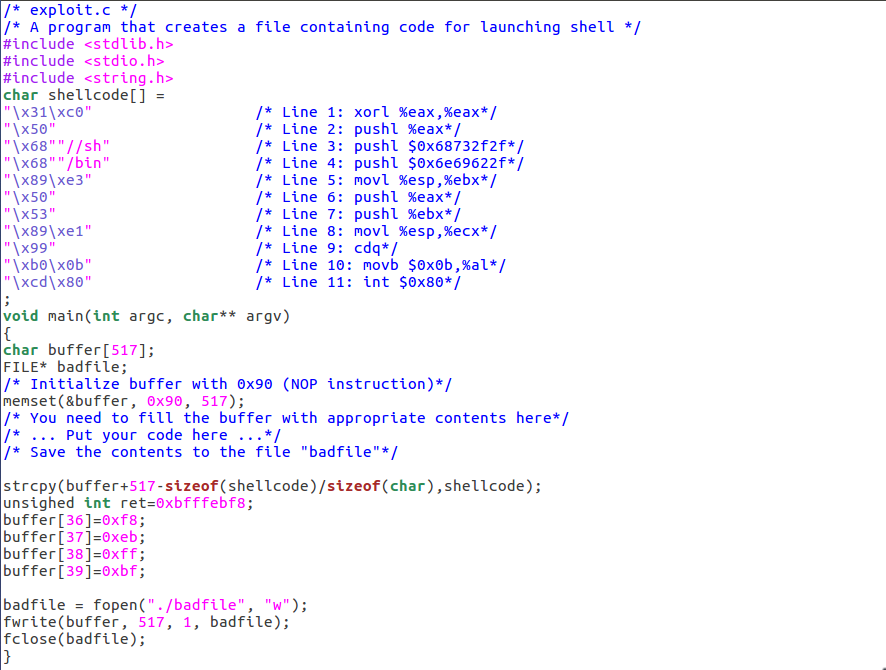






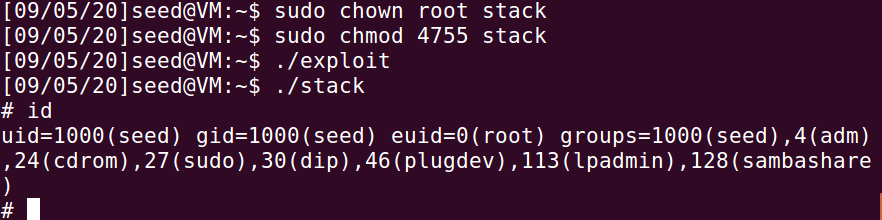
为bof函数设置断点，得到ebp和buffer地址，相差32个字节，考虑在buffer[36]位置发动攻击。

**2.4 Task 2: Exploiting the Vulnerability**



设置用作攻击的return address命令时，只要指向比ebp大8字节以上的地址，比恶意代码小的地址，即可通过NOP指令成功执行恶意代码。





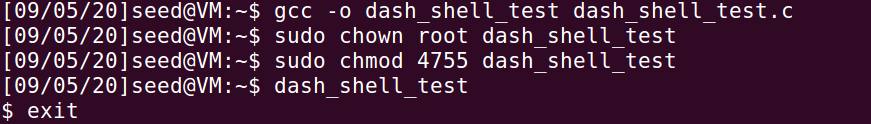
成功启动shell，并获得了root权限

**2.5 Task 3: Defeating dash’s Countermeasure**



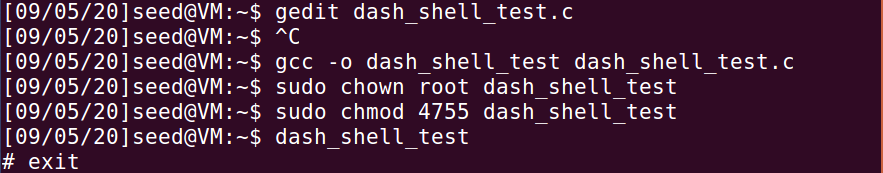
将setuid(0)注释掉：





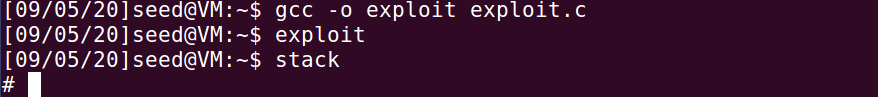
只是获得普通用户的shell

取消注释：



成功获取root权限，证明setuid(0)可以使得RUID和EUID保持一致，从而阻止dash的保护对策

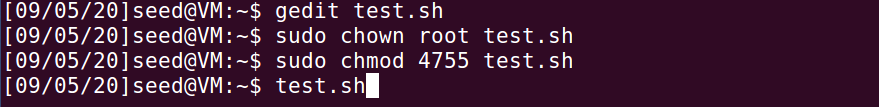
修改exploit.c后，



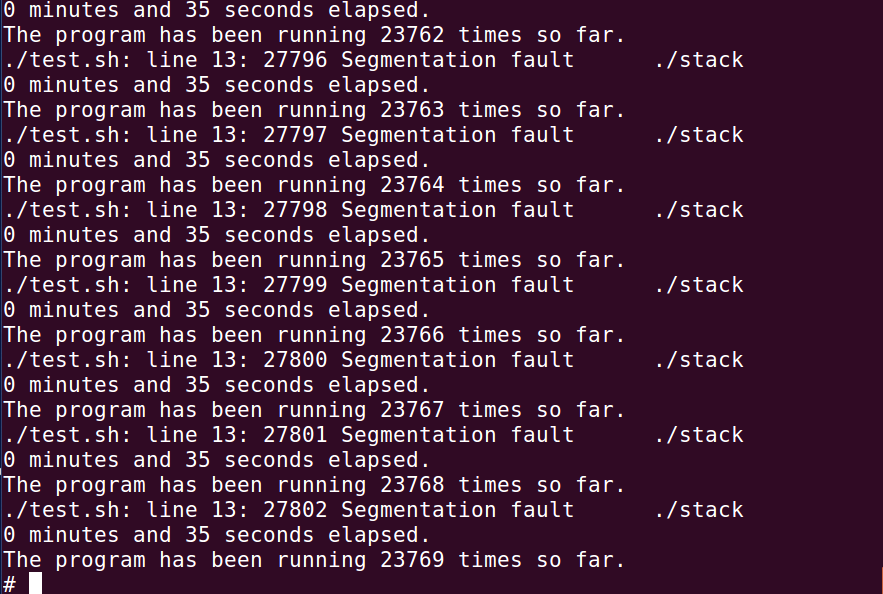
也能成功获取root权限

**2.6 Task 4: Defeating Address Randomization**

编写test.sh脚本，并提升为root所有的setuid程序

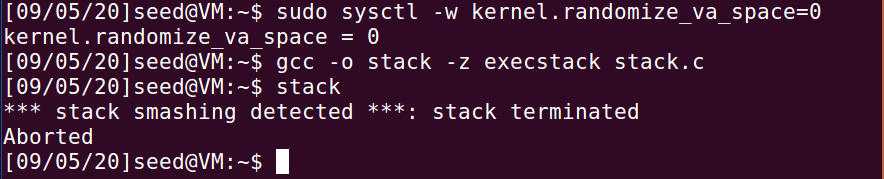


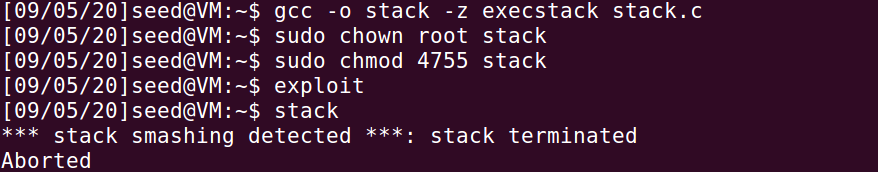
运行test.sh，35s后得到root权限的shell



**2.7 Task 5: Turn on the StackGuard Protection**

关闭地址随机化，并恢复StackGuard保护后：

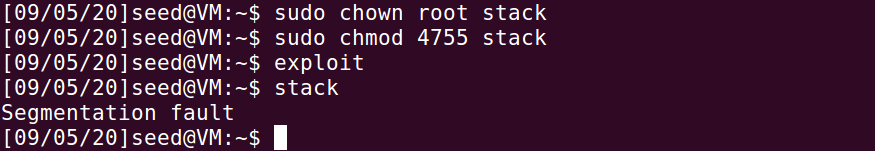




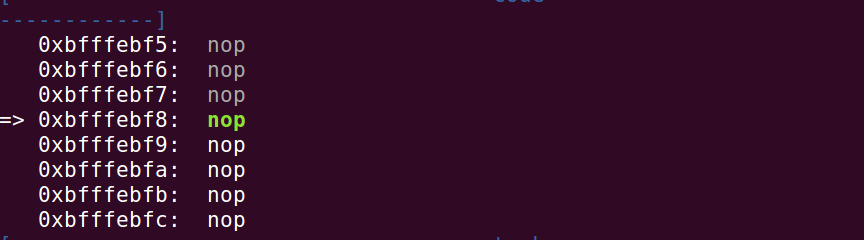
提示栈溢出攻击

**2.8 Task 6: Turn on the Non-executable Stack Protection**





运行时stack报错，进入gdb调试，再次运行



可以看出，错误发生在0xbfffebf8,也正是我们在shellcode中设置的返回地址。