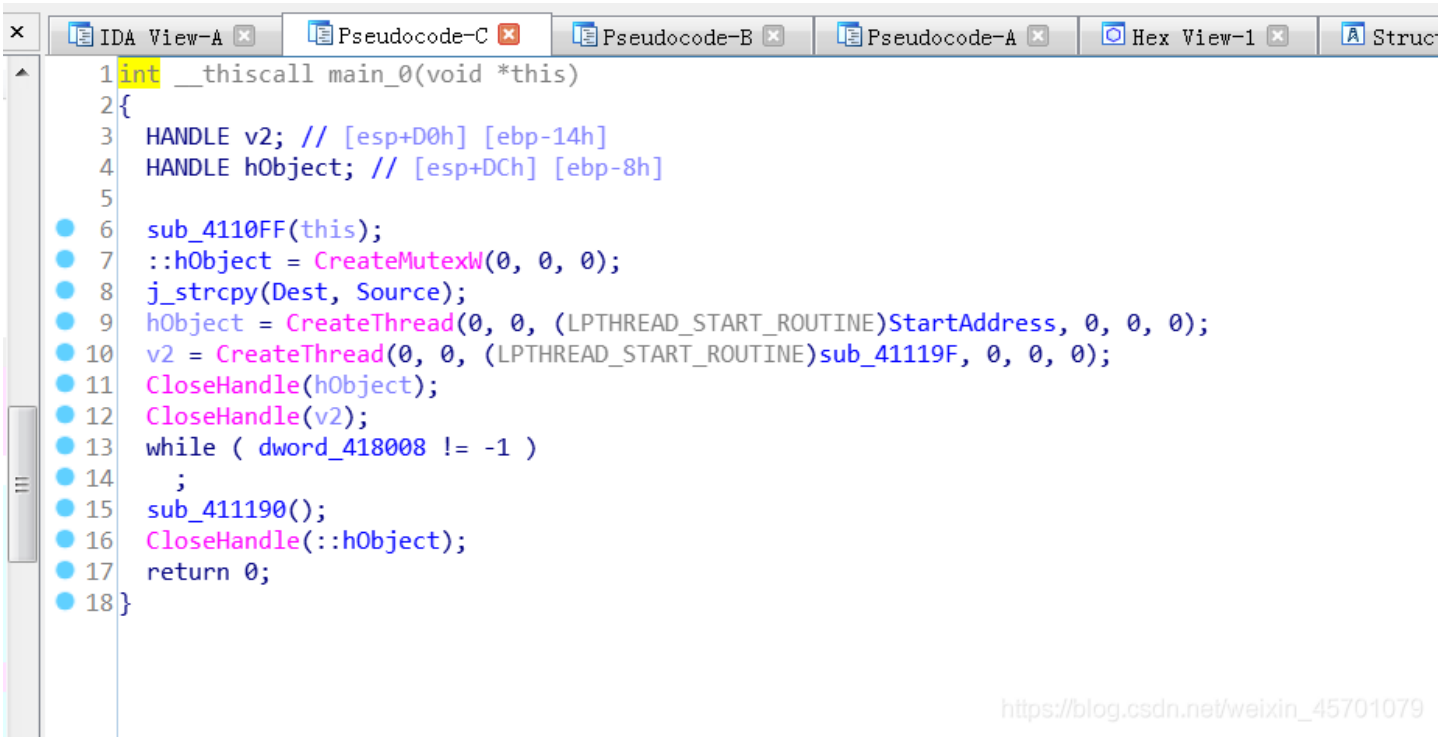
buuctf Youngter-drive

原创

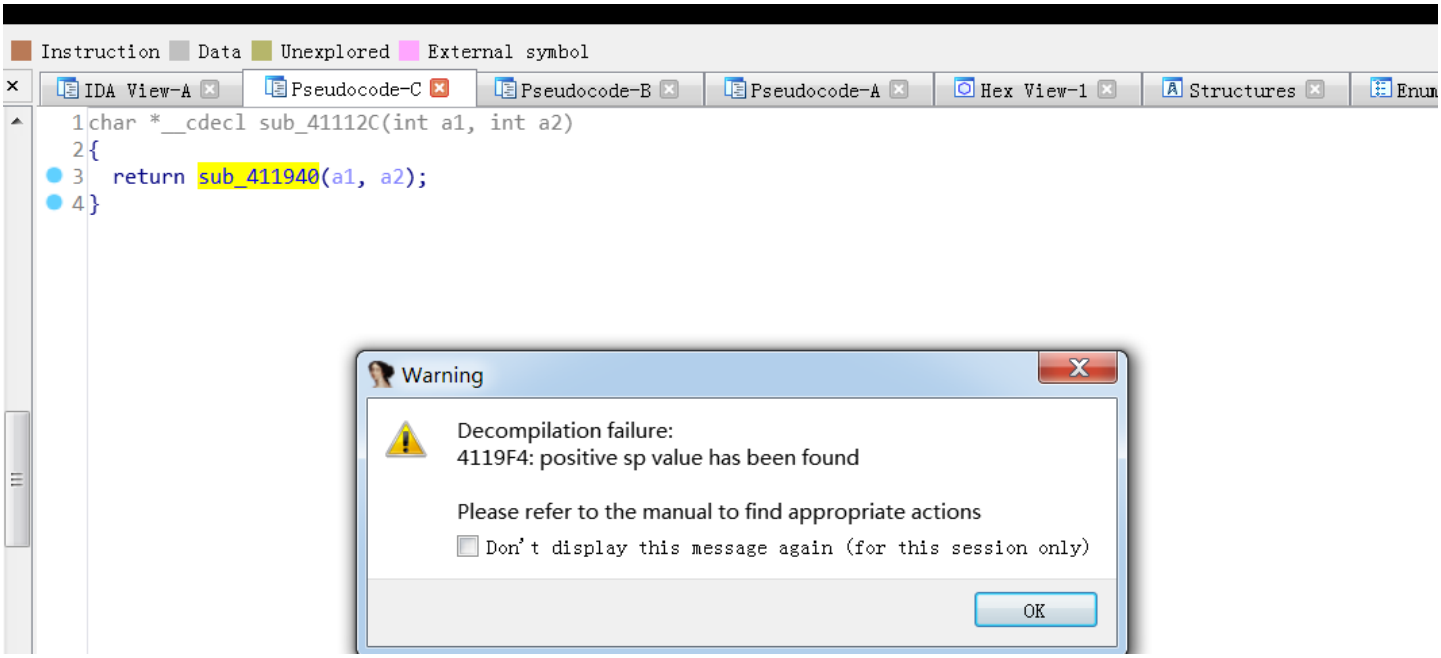
菜逼的ctf之路 于 2020-10-31 15:26:05 发布 1145 收藏 1

buuctf Youngter-drive

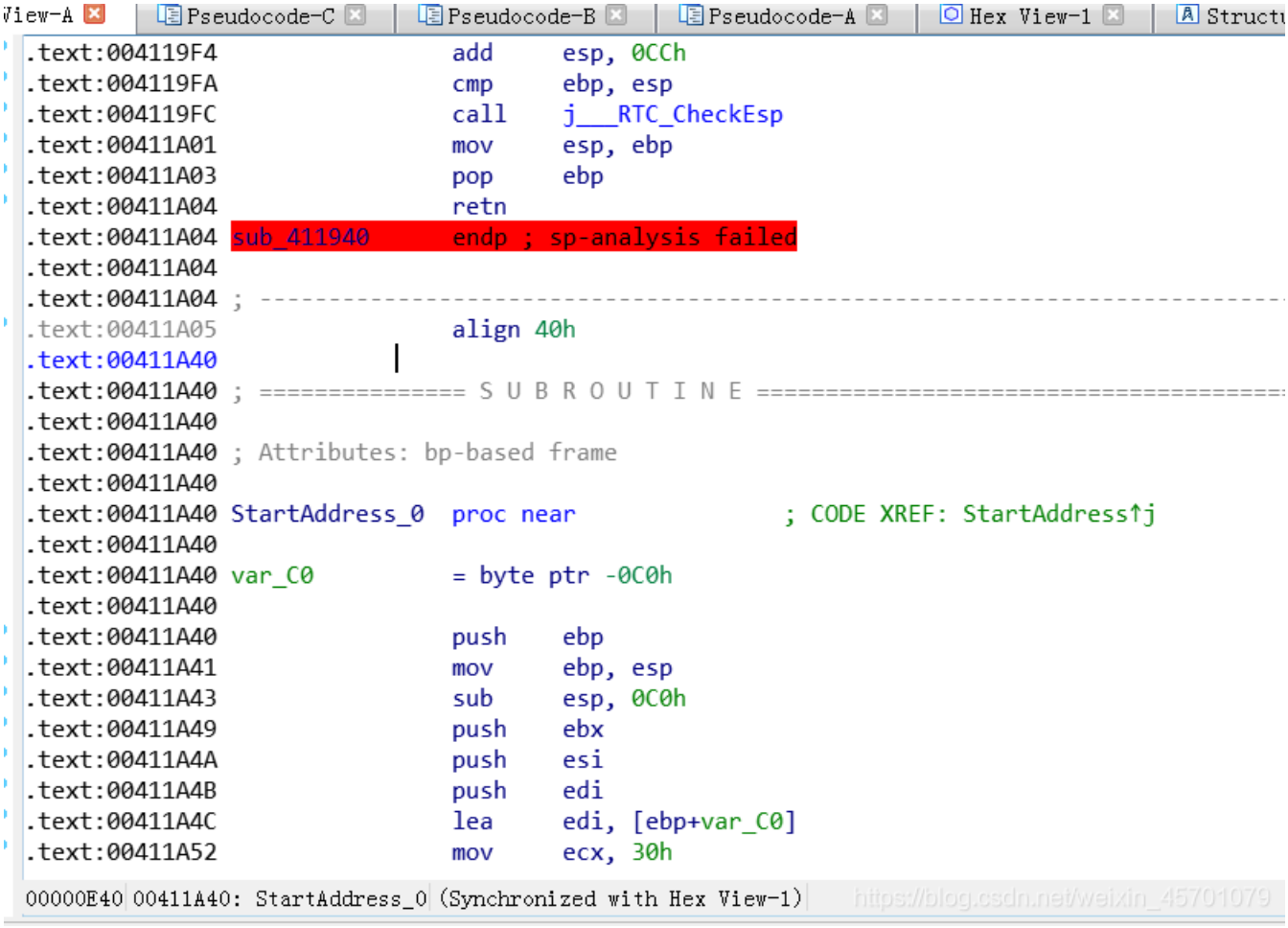
拿到题目，查壳(upx),首先脱壳，(不会脱壳自行百度,很多教程),脱壳之后是不能运行的，好像是因为文件重定向，但是不影响ida静态分析，ida打开



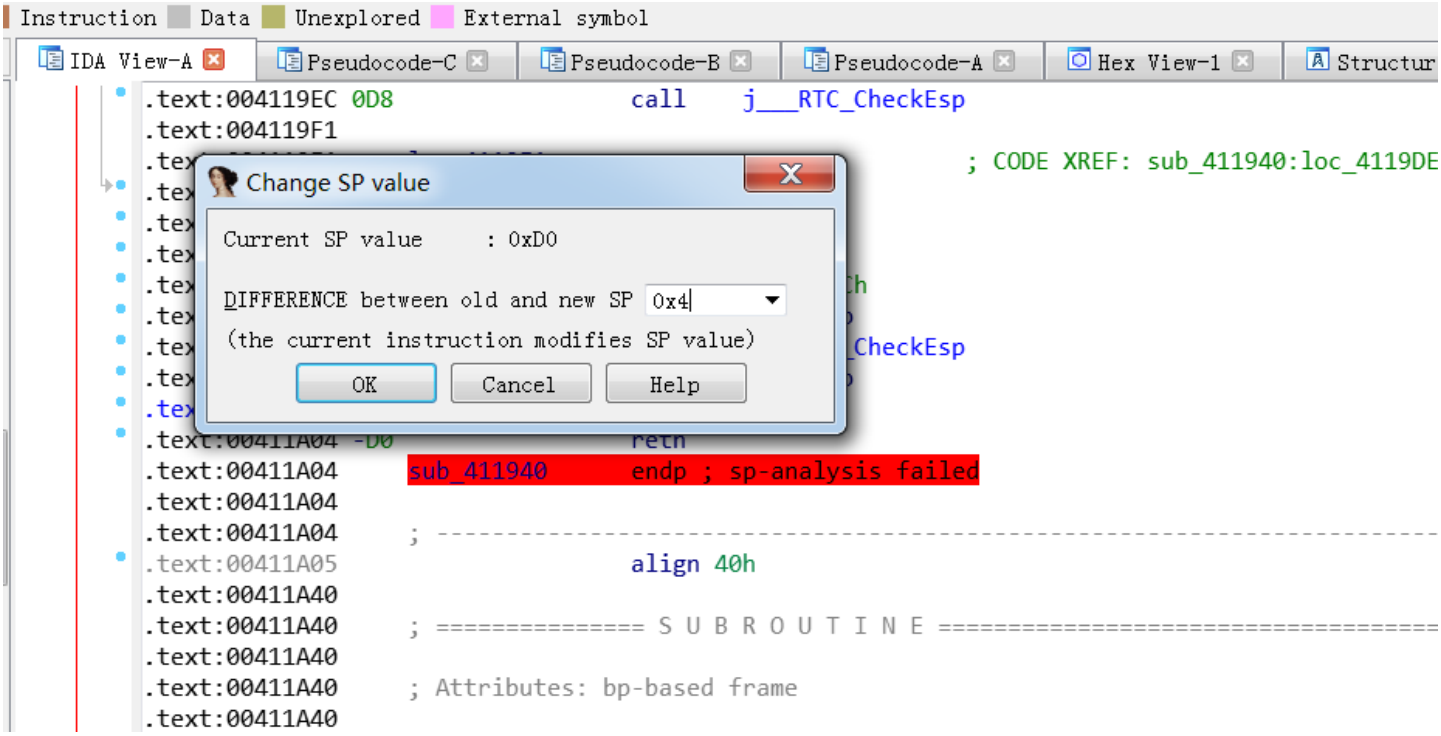
分析，多线程，最近刚好在学多线程，第一个线程h0bject会执行StartAddress函数，第二个线程v2执行sub_41119f函数,然后挨个分析打开StartAddress函数



会出现这种情况，原因是堆栈不平衡利用ida调堆栈就好，



在红色部分用alt+k，打开![在这里插入图片描述]



```
.text:00411A40     StartAddress_0  proc near               ; CODE XREF: StartAddress↑j
.text:00411A40
.text:00411A40     var_C0          = byte ptr -0C0h
.text:00411A40
.text:00411A40 000                 push    ebp
.text:00411A41 004                 mov     ebp, esp
00000E03 00411A03: sub_411940+C3 (Synchronized with Hex View-1)
```


把原来的0x-4改成0x00，堆栈平衡，然后就可以打开函数了

IDA View-A | Pseudocode-C | Pseudocode-B | Pseudocode-A | Hex View-1 | Structure

```
1  char *__cdecl sub_411940(int a1, int a2)
2  {
3    char *result; // eax
4    char v3; // [esp+D3h] [ebp-5h]
5
6    v3 = *(_BYTE *)(a2 + a1);
7    if ( (v3 < 97 || v3 > 122) && (v3 < 65 || v3 > 90) )
8      exit(0);
9    if ( v3 < 97 || v3 > 122 )
10   {
11     result = off_418000[0];
12     *(_BYTE *)(a2 + a1) = off_418000[0][*(char *)(a2 + a1) - 38];
13   }
14   else
15   {
16     result = off_418000[0];
17     *(_BYTE *)(a2 + a1) = off_418000[0][*(char *)(a2 + a1) - 96];
18   }
19   return result;
20 }
```

00000E01 sub_411940:19 (411A01)

逻辑就出现了

然后有坑的地方出现了

他是两个线程同时工作的，所以要交替处理数据

看另一个线程，发现没有处理任何数据

```
1  void __stdcall sub_411B10(int a1)
2  {
3    while ( 1 )
4    {
5      WaitForSingleObject(hObject, 0xFFFFFFFF);
6      if ( dword_418008 > -1 )
7      {
8        Sleep(0x64u);
9        --dword_418008;
10     }
11     ReleaseMutex(hObject);
12   }
13 }
```

只是把原来的字符串向前移动
注意数据是从后向前处理的，查看数据大小发现是0x1d，就是29个，然后分析就行
最后贴上脚本

```
str="0abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ"
text1='TOiZiZtOrYaToUwPnToBsOaOapsyS'
text2='QWERTYUIOPASDFGHJKLZXCVBNMqwertyuiopasdfghjklzxcvbnm'
flag=''
s=0
for i in range(len(text1)):
    if(i%2==0):
        flag+=text1[i]
    else:
        s=text2.index(text1[i])
        flag+=str[s]
print(flag)
```

然后要求有30位数据，发现有一位数据没有对比，一个一个试，发现"E"成立