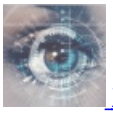
XCTF 3rd-GCTF-2017 hackme

原创

YenKoc 于 2020-03-22 18:53:56 发布 395 收藏

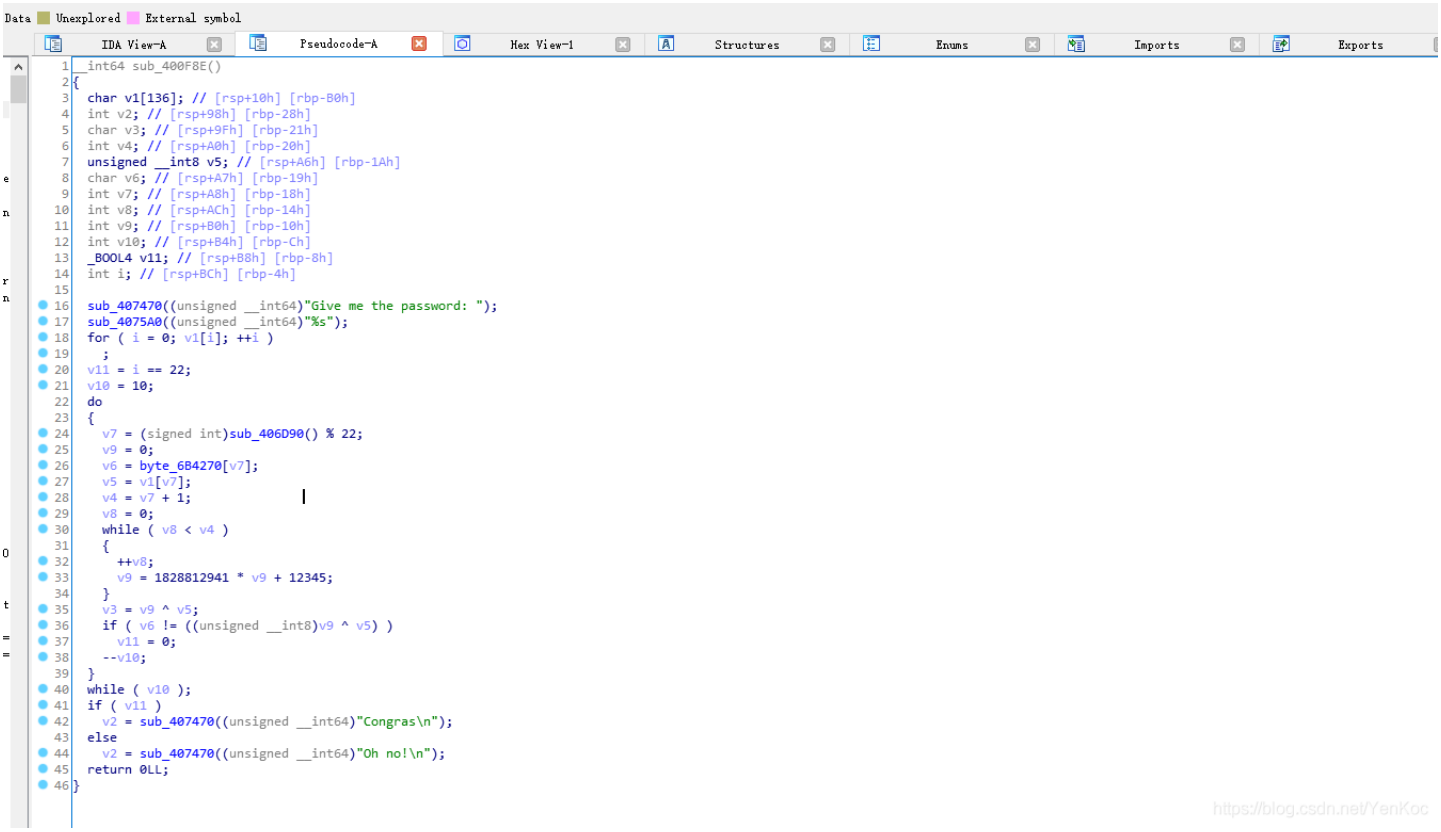
分类专栏： XCTF

本文链接：https://blog.csdn.net/YenKoc/article/details/105032904



一.查壳的老生常谈了。。2分的题目就不多bb了。

二.。elf文件，拖入ida，直接查找字符串，找到对应的函数



```
__int64 sub_400F8E()
{
  char v1[136]; // [rsp+10h] [rbp-B0h]
  int v2; // [rsp+98h] [rbp-28h]
  char v3; // [rsp+9Fh] [rbp-21h]
  int v4; // [rsp+A0h] [rbp-20h]
  unsigned __int8 v5; // [rsp+A6h] [rbp-1Ah]
  char v6; // [rsp+A7h] [rbp-19h]
  int v7; // [rsp+A8h] [rbp-18h]
  int v8; // [rsp+ACh] [rbp-14h]
  int v9; // [rsp+B0h] [rbp-10h]
  int v10; // [rsp+B4h] [rbp-Ch]
  _BOOL4 v11; // [rsp+B8h] [rbp-8h]
  int i; // [rsp+BCh] [rbp-4h]

  sub_407470((unsigned __int64)"Give me the password: ");
  sub_4075A0((unsigned __int64)"%s");
  for ( i = 0; v1[i]; ++i )
    ;
  v11 = i == 22;
  v10 = 10;
  do
  {
    v7 = (signed int)sub_406D90() % 22;
    v9 = 0;
    v6 = byte_6B4270[v7];
    v5 = v1[v7];
    v4 = v7 + 1;
    v8 = 0;
    while ( v8 < v4 )
    {
      ++v8;
      v9 = 1828812941 * v9 + 12345;
    }
    v3 = v9 ^ v5;
    if ( v6 != ((unsigned __int8)v9 ^ v5) )
      v11 = 0;
    --v10;
  }
  while ( v10 );
  if ( v11 )
    v2 = sub_407470((unsigned __int64)"Congras\n");
  else
    v2 = sub_407470((unsigned __int64)"Oh no!\n");
  return 0LL;
}
```

三.直接分析:

```
d int)sub_406D90() % 22;
```

这里讲道理我当时很懵逼，因为进去这个函数后，发现伪码非常复杂，这里困了挺久了，看了wp才知道，这玩意
应该是一个随机数，在0到21中，同时代码还提到了22这个数字，猜测是flag对应的长度，所以应该是这里
然后才能猜测到的，然后这里又有个坑，循环只有10次，肯定是无法得到完整的flag的，所以在脚本中，要凑满22个。

四.

代码：

```
by=[0x5F, 0xF2, 0x5E, 0x8B, 0x4E, 0x0E, 0xA3, 0xAA, 0xC7, 0x93, 0x81, 0x3D, 0x5F, 0x74, 0xA3, 0x09, 0x91, 0x2B,
0x49, 0x28, 0x93, 0x67, 0x00]
flag=""
for i in range(22):
    v9=0
    v8=0
    v4=i+1
    v6=by[i]
    while v8<v4:
        v8+=1
        v9=v9*1828812941+12345

    flag+=chr((v9^v6)&0xff) //起到一个保险作用，这样就不会超过0xff
print(flag)
```