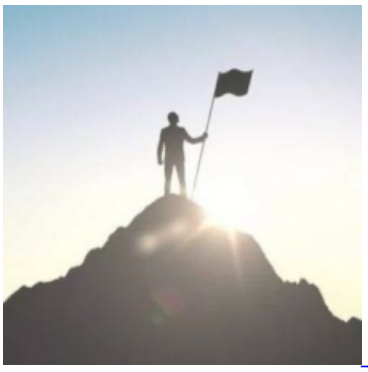
XCTF 进阶 RE crackme

原创

A_dmins 于 2019-07-05 21:07:52 发布 569 收藏

分类专栏： CTF题 XCTF 一天一道CTF

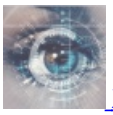
本文链接：https://blog.csdn.net/qq_42967398/article/details/94759944



CTF题 同时被 3 个专栏收录

115 篇文章 11 订阅

订阅专栏



XCTF

24 篇文章 0 订阅

订阅专栏

一天一道CTF

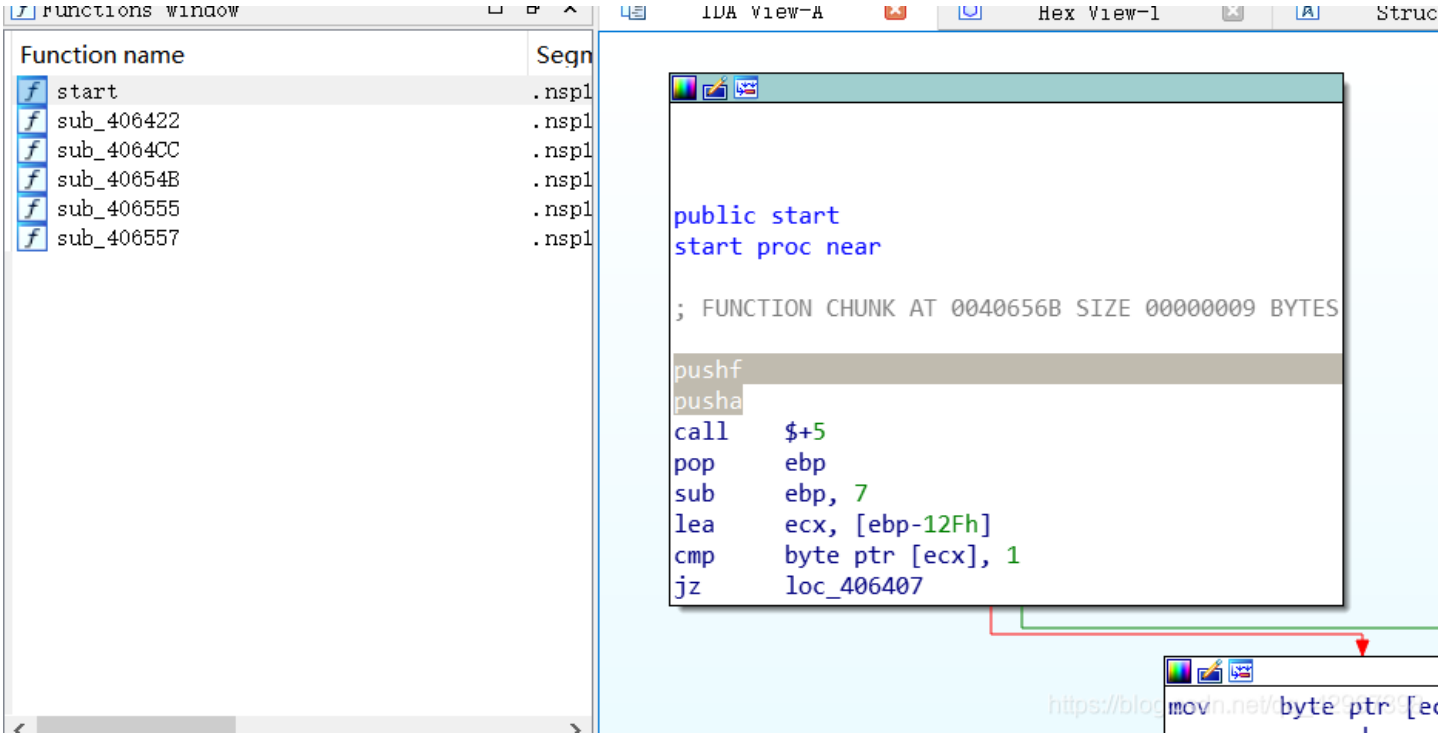
52 篇文章 5 订阅

订阅专栏

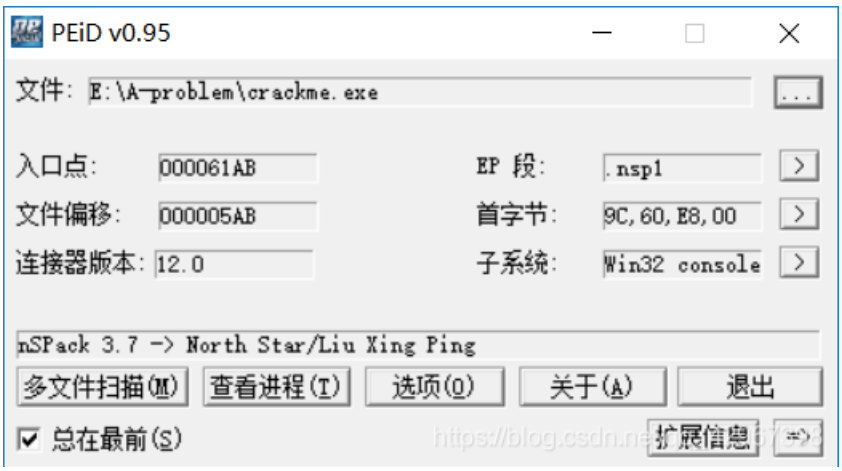
XCTF 进阶 RE crackme

一天一道CTF题目，能多不能少

这道题确实费了些手脚~
关键就是手工去壳，不过还是学到了
题目的逻辑不难，关键就是考察脱壳能力吧
下载文件，打开能正常运行，放入到IDA（32）：



这个有点不对劲啊，怎么可能是这样，主函数都没有，怀疑有壳，查他：



果然有一个壳：`nSPack 3.7 -> North Star/Liu Xing Ping`
好像是什么北斗的壳？？？
去搜索了一波工具~~，但是我好像没找到
于是决定手工去壳了~

这个过程个人还是觉得有必要另外写一篇博客
脱壳过程链接：脱壳过程解析链接

经过脱壳后得到了主函数的源码：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax
  int v4; // eax
```

```
  char Buf; // [esp+4h] [ebp-38h]
  char Dst; // [esp+5h] [ebp-37h]

  Buf = 0;
  memset(&Dst, 0, 0x31u);
  printf("Please Input Flag:");
  gets_s(&Buf, 0x2Cu);
  if ( strlen(&Buf) == 42 )
  {
    v4 = 0;
    while ( (*(&Buf + v4) ^ byte_402130[v4 % 16]) == dword_402150[v4] )
    {
      if ( ++v4 >= 42 )
      {
        printf("right!\n");
        goto LABEL_8;
      }
    }
    printf("error!\n");
LABEL_8:
    result = 0;
  }
  else
  {
    printf("error!\n");
    result = -1;
  }
  return result;
}
```


经过观察，找到主要代码：

```
while ( (*(&Buf + v4) ^ byte_402130[v4 % 16]) == dword_402150[v4] )
{
```

Buf是我们输入的，查看byte_402130和dword_402150，得到：

```
byte_402130     db 't'                  ; DATA XREF: _main:loc_40107F↑r
aHisIsNotFlag   db 'his_is_not_flag',0
                align 10h
; int dword_402150[]
dword_402150    dd 12h                  ; DATA XREF: _main+8D↑r
                dd 4, 8, 14h, 24h, 5Ch, 4Ah, 3Dh, 56h, 0Ah, 10h, 67h, 0
                dd 41h, 0
                dd 1, 46h, 5Ah, 44h, 42h, 6Eh, 0Ch, 44h, 72h, 0Ch, 0Dh
                dd 40h, 3Eh, 4Bh, 5Fh, 2, 1, 4Ch, 5Eh, 5Bh, 17h, 6Eh, 0Ch
                dd 16h, 68h, 5Bh, 12h, 2 dup(0)
                dd 48h, 0Eh dup(0)
                dd offset dword_403000
                dd offset dword_4022B0
                dd 1, 53445352h, 41D713B4h, 4CDD5318h, 12DCFFBAh, 0D5AF8709h
```


按照对代码的理解，写出如下py：

```
s1 = "this_is_not_flag"
s2 = [0x12, 4, 8, 0x14, 0x24, 0x5c, 0x4a, 0x3d, 0x56, 0xa, 0x10, 0x67, 0,
  0x41, 0, 1, 0x46, 0x5a, 0x44, 0x42, 0x6e, 0x0c,
  0x44, 0x72, 0x0c, 0x0d, 0x40, 0x3e, 0x4b, 0x5f, 2, 1, 0x4c, 0x5e,
  0x5b, 0x17, 0x6e, 0xc, 0x16, 0x68, 0x5b, 0x12, 0x48, 0x0e]

flag = ""
for i in range(0,len(s2)):
 flag += chr(s2[i] ^ ord(s1[i % 16]))

print(flag)
```

最后运行得到flag：`flag{59b8ed8f-af22-11e7-bb4a-3cf862d1ee75}`