

XCTF-EasyRE

原创

永远是深夜有多好。 于 2022-01-19 21:14:13 发布 135 收藏

分类专栏: [XCTF](#) 文章标签: [其他](#)

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/qq_37370714/article/details/122588834

版权



[XCTF 专栏收录该内容](#)

17 篇文章 0 订阅

订阅专栏

```
12  __int16 v13; // [esp+1Eh] [ebp-8h]
13
14  sub_FC1020(Format); // input
15  v12 = 0;
16  v13 = 0;
17  *(_OWORD *)Arglist = 0i64;
18  v11 = 0i64;
19  sub_FC1050("%s", Arglist); // scanf
20  v3 = strlen(Arglist);
21  if ( v3 >= 16 && v3 == 24 )
22  {
23      v4 = 0;
24      v5 = (char *)&v11 + 7;
25      do
26      {
27          v6 = *v5--;
28          byte_FC336C[v4++] = v6;
29      }
30      while ( v4 < 24 );
31      for ( i = 0; i < 0x18; ++i )
32          byte_FC336C[i] = (byte_FC336C[i] + 1) ^ 6;
33      v8 = strcmp(byte_FC336C, aXircjR2twsv3pt);
34      if ( v8 )
35          v8 = v8 < 0 ? -1 : 1;
36      if ( !v8 )
37      {
38          sub_FC1020("right\n");
39          system("pause");
40      }
41  }
42  return 0;
43 }
```

CSDN @永远是深夜有多好。

一进来就发现一个假的flag不管他, 开始分析代码, 发现这句代码很重要, 两个相等为0才能满足right

```
v8 = strcmp(byte_FC336C, aXircjR2twsv3pt);
```

```
v3 = strlen(Arglist);
if ( v3 >= 16 && v3 == 24 )
{
    v4 = 0;
    v5 = (char *)&v11 + 7;
    do
    {
```

```
{
    v6 = *v5--;
    byte_FC336C[v4++] = v6;
}
while ( v4 < 24 );
for ( i = 0; i < 0x18; ++i )
    byte_FC336C[i] = (byte_FC336C[i] ^ 6);
```

继续往上面看

先进行逆序运算，然后再加一做异或运算，反过来就可以解。

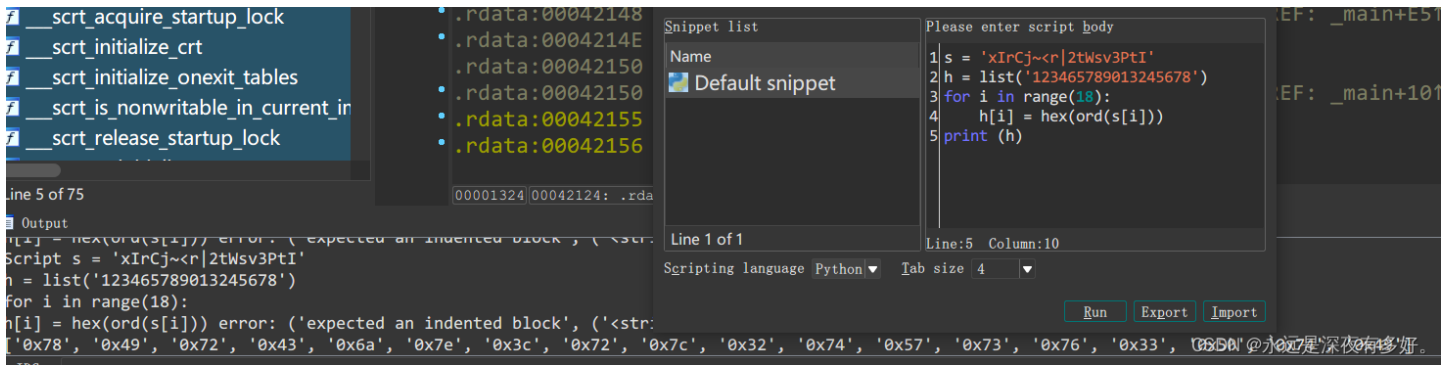
仔细观察发现

```
XircjR2twsv3pt);
const char[]
"xIrcj~<r|2tWsv3PtI"
```

aXircjR2twsv3pt应该有24位

```
• .rdata:00042124 aXircjR2twsv3pt db 'xIrcj~<r|2tWsv3PtI' ; DATA XREF: _main+A8↑o
  .rdata:00042124 | ; .rdata:0004215C↓o
• .rdata:00042136 db 7Fh ;
• .rdata:00042137 db 7Ah ; z
• .rdata:00042138 db 6Eh ; n
• .rdata:00042139 db 64h ; d
• .rdata:0004213A db 6Bh ; k
• .rdata:0004213B db 61h ; a
```

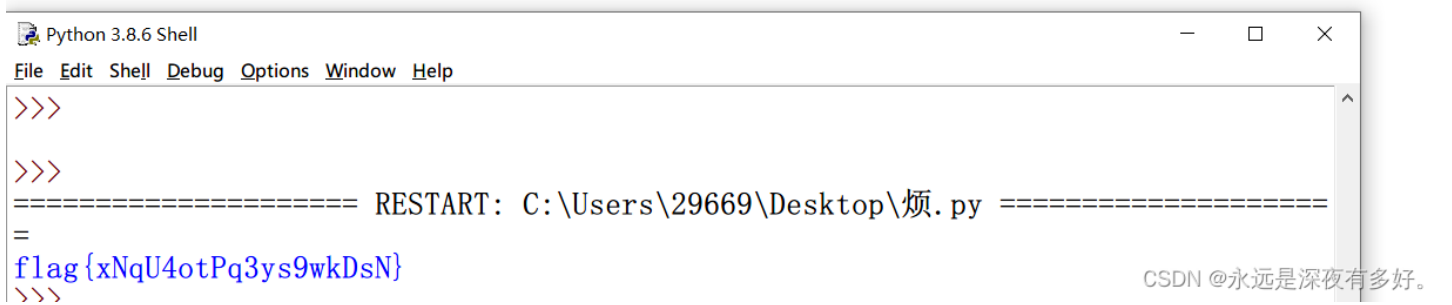
xIrcj~<r|2tWsv3PtI加上后面的7F 7A 6E 64 6B 61构成24位
这时候需要把xIrcj~<r|2tWsv3PtI转成十六进制



```

File Edit Format Run Options Window Help
flag = [0x78, 0x49, 0x72, 0x43, 0x6a, 0x7e, 0x3c, 0x72, 0x7c, 0x32, 0x74, 0x57,
        0x73, 0x76, 0x33, 0x50, 0x74, 0x49, 0x7f, 0x7a, 0x6e, 0x64, 0x6b, 0x61]
for i in range(24):
    flag[i] = chr((flag[i] ^ 6) - 1)
print ("".join(flag[::-1])) #[::-1]也就是逆序

```



最开始以为xIrcjR2twsv3pt只有xIrcj~<r|2tWsv3PtI这十八位后面才发现底下还有6位