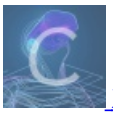
XCTF-EasyRE

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

本文链接：https://blog.csdn.net/qq_37370714/article/details/122588834




```
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}
```


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);
```

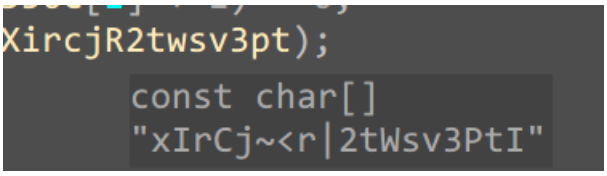
```
v3 = strlen(Arglist);
if ( v3 >= 16 && v3 == 24 )
{
  v4 = 0;
  v5 = (char *)&v11 + 7;
  do
```

```
  {
    v6 = *v5--;
    byte_FC336C[v4++] = v6;
  }
  while ( v4 < 24 );
  for ( i = 0; i < 0x18; ++i )
    byte_FC336C[i] = (byte_FC336C[i] - 1) ^ 5;
```

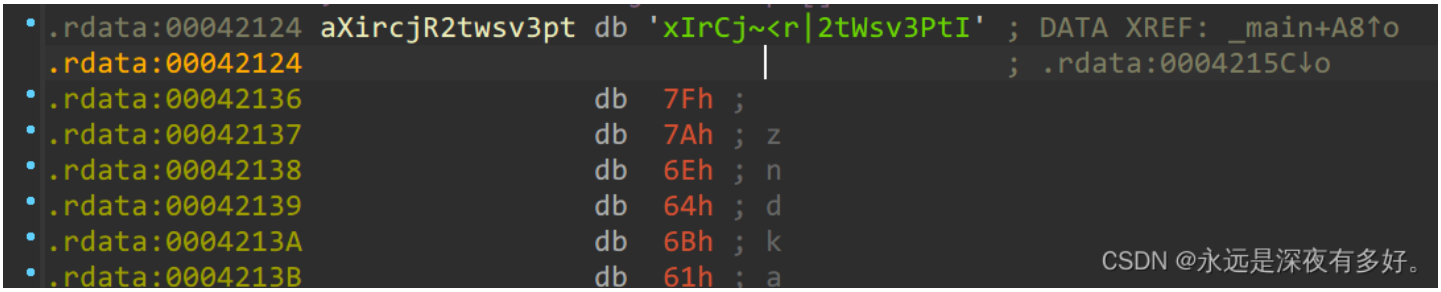

继续往上面看

先进行逆序运算，然后再加一做异或运算，反过来就可以解。

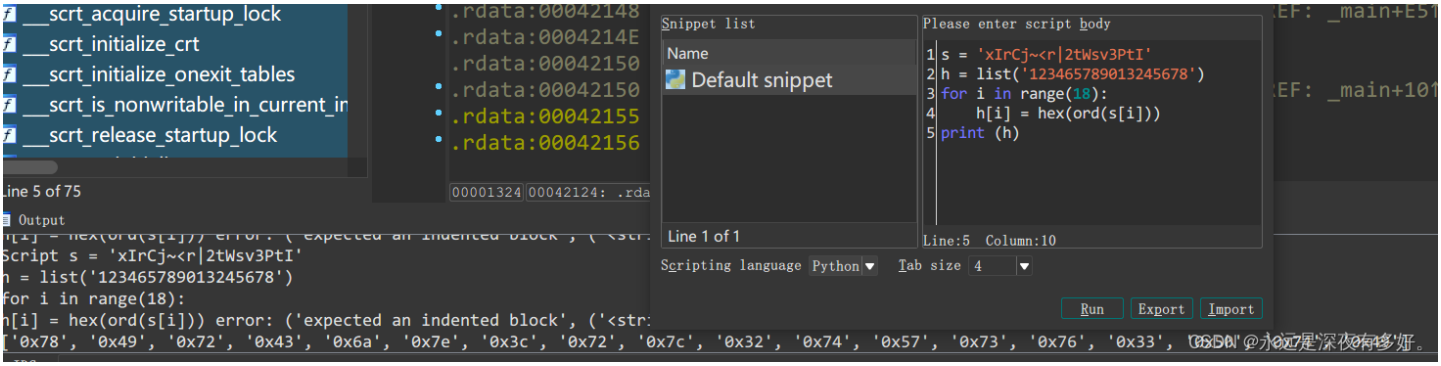
仔细观察发现

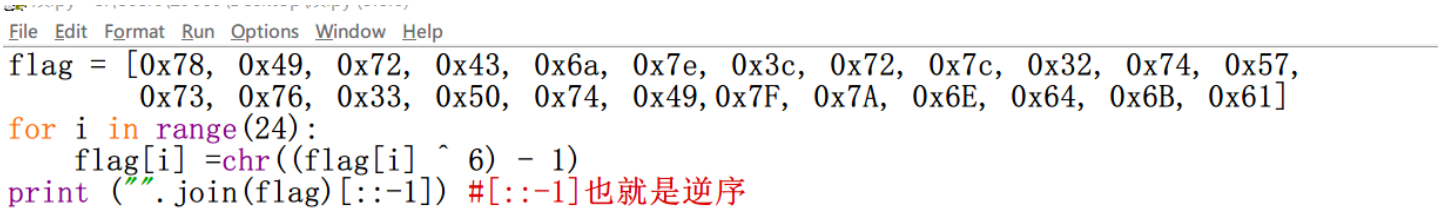


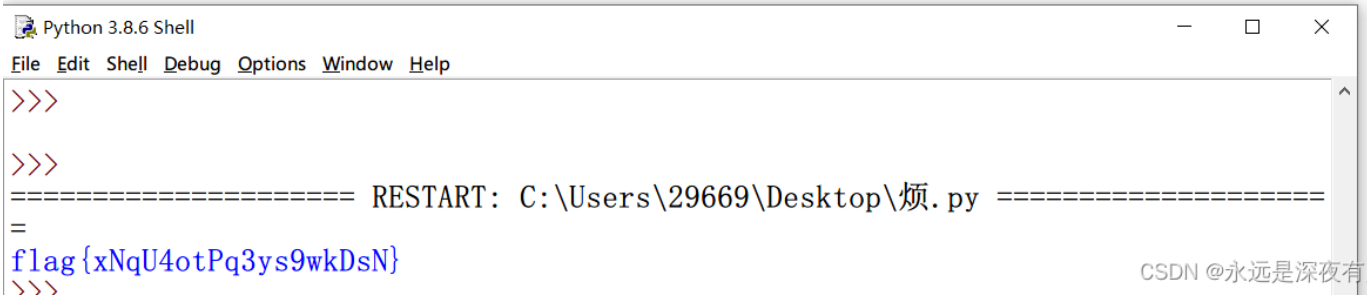
> aXircjR2twsv3pt应该有24位



> xlrCj~<r|2tWsv3Ptl加上后面的7F 7A 6E 64 6B 61构成24位
> 这时候需要把xlrCj~<r|2tWsv3Ptl转成十六进制







CSDN @永远是深夜有多好。

最开始以为aXircjR2twsv3pt只有xlrCj~<r|2tWsv3Ptl这十八位后面才发现底下还有6位