攻防世界 Reverse高手进阶区 2分题 BABYRE

原创

思源湖的鱼 于 2020-12-26 14:36:35 发布 209 收藏

分类专栏： ctf 文章标签： ctf 攻防世界 reverse

本文链接：https://blog.csdn.net/weixin_44604541/article/details/111741907

ctf 专栏收录该内容

200 篇文章 23 订阅

订阅专栏

前言

继续ctf的旅程

攻防世界Reverse高手进阶区的2分题

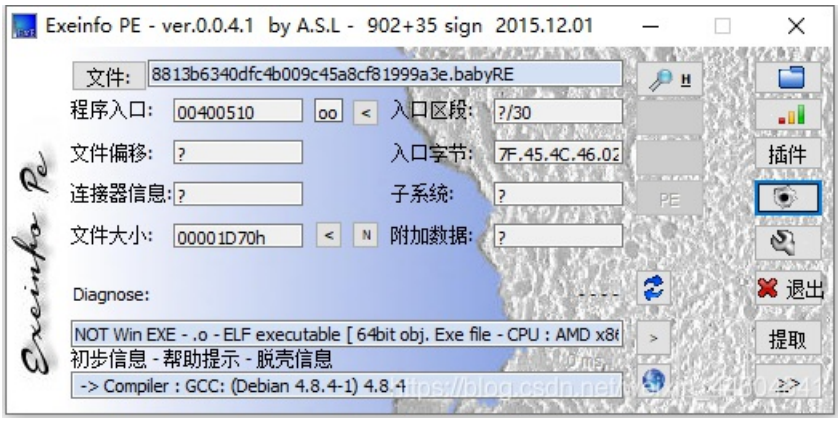
本篇是BABYRE的writeup

发现攻防世界的题目分数是动态的

就仅以做题时的分数为准了

解题过程

PE查壳

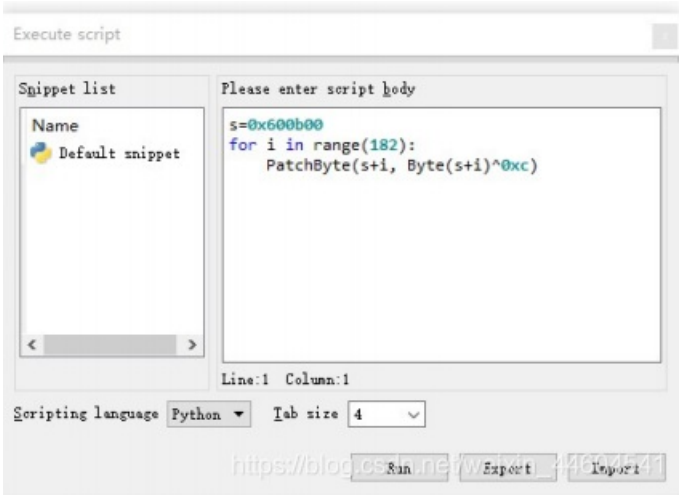


扔进IDA

```
1  int __cdecl main(int argc, const char **argv, const char **envp)
2  {
3    char s; // [rsp+0h] [rbp-20h]
4    int v5; // [rsp+18h] [rbp-8h]
5    int i; // [rsp+1Ch] [rbp-4h]
6
7    for ( i = 0; i <= 181; ++i )
8    {
9      envp = (const char **)(*((unsigned __int8 *)judge + i) ^ 0xCu);
10     *((_BYTE *)judge + i) ^= 0xCu;
11   }
12   printf("Please input flag:", argv, envp);
13   __isoc99_scanf("%20s", &s);
14   v5 = strlen(&s);
15   if ( v5 == 14 && (unsigned int)judge(&s) )
16     puts("Right!");
17   else
18     puts("Wrong!");
19   return 0;
```


显然关键是judge
但是直接反编译失败
因为judge被加密了

根据7-11行，
可以解密



然后得到judge

```
signed __int64 __fastcall judge(__int64 a1)
{
  char v2; // [rsp+8h] [rbp-20h]
  char v3; // [rsp+9h] [rbp-1Fh]
  char v4; // [rsp+Ah] [rbp-1Eh]
  char v5; // [rsp+Bh] [rbp-1Dh]
  char v6; // [rsp+Ch] [rbp-1Ch]
  char v7; // [rsp+Dh] [rbp-1Bh]
  char v8; // [rsp+Eh] [rbp-1Ah]
  char v9; // [rsp+Fh] [rbp-19h]
  char v10; // [rsp+10h] [rbp-18h]
  char v11; // [rsp+11h] [rbp-17h]
  char v12; // [rsp+12h] [rbp-16h]
  char v13; // [rsp+13h] [rbp-15h]
  char v14; // [rsp+14h] [rbp-14h]
  char v15; // [rsp+15h] [rbp-13h]
  int i; // [rsp+24h] [rbp-4h]

  v2 = 102;
  v3 = 109;
  v4 = 99;
  v5 = 100;
  v6 = 127;
  v7 = 107;
  v8 = 55;
  v9 = 100;
  v10 = 59;
  v11 = 86;
  v12 = 96;
  v13 = 59;
  v14 = 110;
  v15 = 112;
  for ( i = 0; i <= 13; ++i )
    *(_BYTE *)(i + a1) ^= i;
  for ( i = 0; i <= 13; ++i )
  {
    if ( *(_BYTE *)(i + a1) != *(&v2 + i) )
      return 0LL;
  }
  return 1LL;
}
```

简单

```
v2 = 102
v3 = 109
v4 = 99
v5 = 100
v6 = 127
v7 = 107
v8 = 55
v9 = 100
v10 = 59
v11 = 86
v12 = 96
v13 = 59
v14 = 110
v15 = 112

data=[]
for i in range(2,16):
  data.append(locals()['v'+str(i)])

flag=''
for i in range(14):
  flag+=chr(data[i]^i)

print(flag)
```

得到flag：`flag{n1c3_j0b}`

结语

关键是解密