buuctf [ACTF新生赛2020]Universe_final_answer wp

原创

43v3rY0unG 于 2020-08-13 21:59:16 发布 805 收藏
分类专栏： # RE

本文链接：https://blog.csdn.net/weixin_43876357/article/details/107989319

ida f5找到主函数，发现关键函数sub_860()：

```
__int64 __fastcall main(__int64 a1, char **a2, char **a3)
{
  __int64 v4; // [rsp+0h] [rbp-A8h]
  char v5; // [rsp+20h] [rbp-88h]
  unsigned __int64 v6; // [rsp+88h] [rbp-20h]

  v6 = __readfsqword(0x28u);
  __printf_chk(1LL, "Please give me the key string:", a3);
  scanf("%s", &v5);
  if ( sub_860(&v5) )
  {
    sub_C50((__int64)&v5, &v5, &v4);
    __printf_chk(1LL, "Judgement pass! flag is actf{%s_%s}\n", &v5);
  }
  else
  {
    puts("False key!");
  }
  return 0LL;
}
```


点进去看一下函数逻辑：

```
14
15  v1 = a1[1];
16  v2 = *a1;
17  v3 = a1[2];
18  v4 = a1[3];
19  v5 = a1[4];
20  v6 = a1[6];
21  v7 = a1[5];
22  v8 = a1[7];
23  v9 = a1[8];
24  result = 0;
25  if ( -85 * v9 + 58 * v8 + 97 * v6 + v7 + -45 * v5 + 84 * v4 + 95 * v2 - 20 * v1 + 12 * v3 == 12613 )
26  {
27    v11 = a1[9];
28    if ( 30 * v11 + -70 * v9 + -122 * v6 + -81 * v7 + -66 * v5 + -115 * v4 + -41 * v3 + -86 * v1 - 15 * v2 - 30 * v8 == -54400
29      && -103 * v11 + 120 * v8 + 108 * v7 + 48 * v4 + -89 * v3 + 78 * v1 - 41 * v2 + 31 * v5 - (v6 << 6) - 120 * v9 == -10283
30      && 71 * v6 + (v7 << 7) + 99 * v5 + -111 * v3 + 85 * v1 + 79 * v2 - 30 * v4 - 119 * v8 + 48 * v9 - 16 * v11 == 22855
31      && 5 * v11 + 23 * v9 + 122 * v8 + -19 * v6 + 99 * v7 + -117 * v5 + -69 * v3 + 22 * v1 - 98 * v2 + 10 * v4 == -2944
32      && -54 * v11 + -23 * v8 + -82 * v3 + -85 * v2 + 124 * v1 - 11 * v4 - 8 * v5 - 60 * v7 + 95 * v6 + 100 * v9 == -2222
33      && -83 * v11 + -111 * v7 + -57 * v2 + 41 * v1 + 73 * v3 - 18 * v4 + 26 * v5 + 16 * v6 + 77 * v8 - 63 * v9 == -13258
34      && 81 * v11 + -48 * v9 + 66 * v8 + -104 * v6 + -121 * v7 + 95 * v5 + 85 * v4 + 60 * v3 + -85 * v2 + 80 * v1 == -1559
35      && 101 * v11 + -85 * v9 + 7 * v6 + 117 * v7 + -83 * v5 + -101 * v4 + 90 * v3 + -28 * v1 + 18 * v2 - v8 == 6308 )
36    {
37      result = 99 * v11 + -28 * v9 + 5 * v8 + 93 * v6 + -18 * v7 + -127 * v5 + 6 * v4 + -9 * v3 + -93 * v1 + 58 * v2 == -1697;
38    }
39  }
40  return result;
41}
```

其实本身逻辑是比较简单的，问题就在于怎么运算：
这里会用到python的z3库
由于我也是第一次接触这个库，所以这里放一些关于这个库的安装和使用的链接：

- z3的安装
- z3的使用

在这里强调一下，安装z3库时需要确保你的python版本为2.x，由于我的windows装的是3.6版本，所以我就下在Linux上了，不过步骤都是一样的，正常下载即可。

exp

```
from z3 import *

#s = Solver()
v1 = Int('v1')
v2 = Int('v2')
v3 = Int('v3')
v4 = Int('v4')
v5 = Int('v5')
v6 = Int('v6')
v7 = Int('v7')
v8 = Int('v8')
v9 = Int('v9')
v11 = Int('v11')

s.add(-85 * v9 + 58 * v8 + 97 * v6 + v7 + -45 * v5 + 84 * v4 + 95 * v2 - 20 * v1 + 12 * v3 == 12613)
s.add(
    30 * v11 + -70 * v9 + -122 * v6 + -81 * v7 + -66 * v5 + -115 * v4 + -41 * v3 + -86 * v1 - 15 * v2 - 30
s.add(-103 * v11 + 120 * v8 + 108 * v7 + 48 * v4 + -89 * v3 + 78 * v1 - 41 * v2 + 31 * v5 - (
            v6 * 64) - 120 * v9 == -10283)
s.add(71 * v6 + (v7 * 128) + 99 * v5 + -111 * v3 + 85 * v1 + 79 * v2 - 30 * v4 - 119 * v8 + 48 * v9 - 16 *
s.add(5 * v11 + 23 * v9 + 122 * v8 + -19 * v6 + 99 * v7 + -117 * v5 + -69 * v3 + 22 * v1 - 98 * v2 + 10 * v
s.add(-54 * v11 + -23 * v8 + -82 * v3 + -85 * v2 + 124 * v1 - 11 * v4 - 8 * v5 - 60 * v7 + 95 * v6 + 100 *
s.add(-83 * v11 + -111 * v7 + -57 * v2 + 41 * v1 + 73 * v3 - 18 * v4 + 26 * v5 + 16 * v6 + 77 * v8 - 63 * v
s.add(81 * v11 + -48 * v9 + 66 * v8 + -104 * v6 + -121 * v7 + 95 * v5 + 85 * v4 + 60 * v3 + -85 * v2 + 80 *
s.add(101 * v11 + -85 * v9 + 7 * v6 + 117 * v7 + -83 * v5 + -101 * v4 + 90 * v3 + -28 * v1 + 18 * v2 - v8 =
s.add(99 * v11 + -28 * v9 + 5 * v8 + 93 * v6 + -18 * v7 + -127 * v5 + 6 * v4 + -9 * v3 + -93 * v1 + 58 * v2

if s.check() == sat:
    result = s.model()

print(result)
```

得到结果：

```
[v6 = 95,
 v9 = 119,
 v11 = 64,
 v7 = 121,
 v2 = 70,
 v4 = 82,
 v3 = 117,
 v1 = 48,
 v5 = 84,
 v8 = 55]
```

按照相应的顺序转成字符串（根据ida内函数逻辑可以知道，v2和v1互换，v7和v6互换）

得到内容：F0uRTy_7w@，将其输入到输入到程序中，即可get flag！

get flag:

```
flag{F0uRTy_7w@_42}
```