BUUCTF Reverse/[ACTF新生赛2020]Universe_final_answer

原创

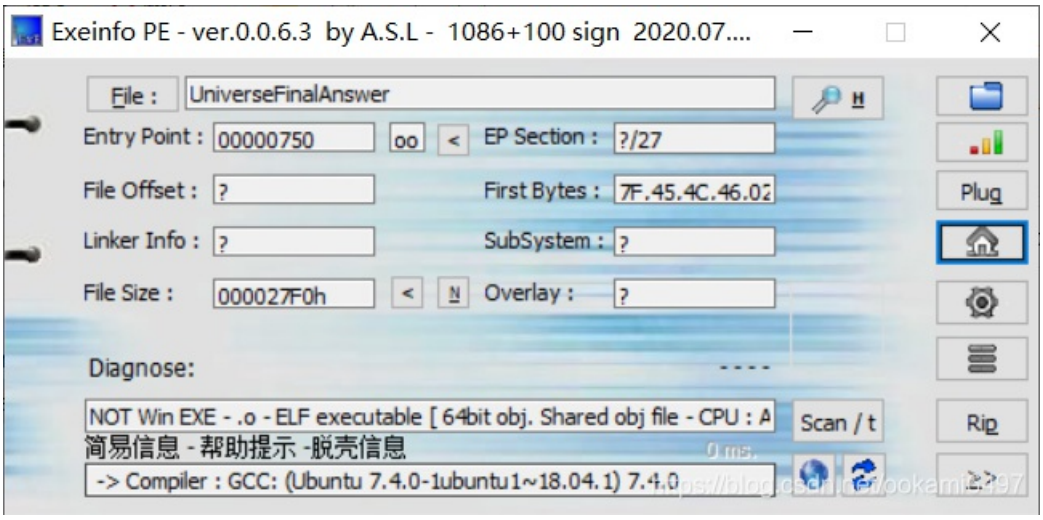
这就是强者的世界么 于 2021-07-29 16:38:16 发布 113 收藏

分类专栏： # BUUCTF Reverse

本文链接：https://blog.csdn.net/ookami6497/article/details/119214347

BUUCTF Reverse/[ACTF新生赛2020]Universe_final_answer

> 先看文件信息，没有加壳



用IDA64位打开，代码很简单，又是输入字符串比较

```
__int64 __fastcall main(int a1, char **a2, char **a3)
{
  char v4[32]; // [rsp+0h] [rbp-A8h] BYREF
  char v5[104]; // [rsp+20h] [rbp-88h] BYREF
  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(v5, v4);
    __printf_chk(1LL, "Judgement pass! flag is actf{%s_%s}\n", v5);
  }
  else
  {
    puts("False key!");
  }
  return 0LL;
}
```

跟进if语句中的 sub_860(v5)，只要算出这10位字符就行

```
bool __fastcall sub_860(char *a1)
{
  int v1; // ecx
  int v2; // esi
  int v3; // edx
  int v4; // er9
  int v5; // er11
  int v6; // ebp
  int v7; // ebx
  int v8; // er8
  int v9; // er10
  bool result; // al
  int v11; // [rsp+0h] [rbp-38h]

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

> 用python计算，参考用python解多元一次方程，用numpy库，按v11，v9，v8 。。。。。v1排列

> 写出脚本，这玩意就是容易看错，，，我检查了好几遍才输出正确答案(看别人wp才知道用z3库要简单好多，我想抽死我自己)

```
import numpy as np

m = np.array([[0,-85,58,1,97,-45,84,12,95,-20],           #第一个式子没有v11，所以用0代替
              [30,-70,-30,-81,-122,-66,-115,-41,-15,-86],
              [-103,-120,120,108,-64,31,48,-89,-41,78],
              [-16,48,-119,128,71,99,-30,-111,79,85],
              [5,23,122,99,-19,-117,10,-69,-98,22],
              [-54,100,-23,-60,95,-8,-11,-82,-85,124],
              [-83,-63,77,-111,16,26,-18,73,-57,41],
              [81,-48,66,-121,-104,95,85,60,-85,80],
              [101,-85,-1,117,7,-83,-101,90,18,-28],
              [99,-28,5,-18,93,-127,6,-9,58,-93]])
n = np.array([12613,-54400,-10283,22855,-2944,-2222,-13258,-1559,6308,-1697])
solution = np.linalg.solve(m,n)
print(solution)
```

> 运行结果：

```
[ 64. 119.  55. 121.  95.  84.  82. 117.  70.  48.]
```

> 根据这个进行对照

```
  v1 = a1[1];
  v2 = *a1;
  v3 = a1[2];
  v4 = a1[3];
  v5 = a1[4];
  v6 = a1[6];
  v7 = a1[5];
  v8 = a1[7];
  v9 = a1[8];
  v11 = a1[9];
```

> 写个脚本输出一下结果

```
a = [ 64, 119,55, 95, 121, 84,82,117, 48,70]
for i in range(10):
    print(chr(a[9-i]),end='')
```

> 结果得到：

```
F0uRTy_7w@
```

> 在Linux环境下运行这个程序，在kali里面运行的话，要先在属性里面将这个允许此文件作为程序运行 的框选上

然后拖到命令行里面运行，输入刚刚得到的字符串，得到flag

flag{F0uRTy_7w@_42}