buu练题记录5-[ACTF新生赛2020]Universe_final_answer

原创

Asteri5m 于 2021-04-14 21:51:35 发布 74 收藏

分类专栏： buuctf Reserve

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_45892237/article/details/115710522

版权

buuctf 同时被 2 个专栏收录

16 篇文章 0 订阅

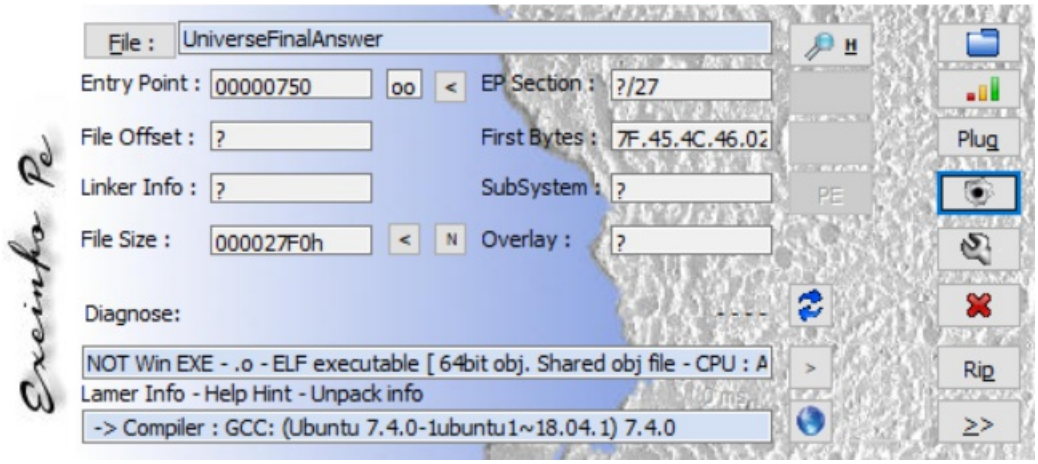
订阅专栏

Reserve

18 篇文章 1 订阅

订阅专栏

0x00 查壳



没有壳，是ELF文件，上IDA64

0x01 IDA分析

在函数列表很容易可以找到main函数，查看的确是关键函数：

```
__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 ( (unsigned __int8)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;
}
```

这里有且只调用了sub_860和sub_C50两个函数，当sub_860返回真值就能输出flag，所以先分析sub_860：

```
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;
}
```

这里就是对输入的v5进行效验，核心是输入的10个字符满足这个十元一次方程组即可。这里解这个方程组用暴力算法比较废时间而且不大可能。所以我们需要用到python的z3库

0x02 安装z3库

```
pip install z3-solver
```

```
C:\Users\15973>pip install z3-solver
Collecting z3-solver
  Downloading z3_solver-4.8.10.0-py2.py3-none-win_amd64.whl (35.5 MB)
     |████████████████████████████████| 35.5 MB 112 kB/s
Installing collected packages: z3-solver
Successfully installed z3-solver-4.8.10.0
```

安装成功之后就可以写exp解input了

z3使用教程：Z3Py教程(翻译)

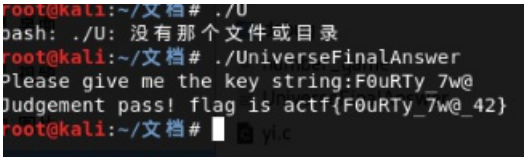
0x03 exp

```
from z3 import *

F = [Int('F[%d]' % i) for i in range(10)]

s = Solver()
s.add(-85 * F[8] + 58 * F[7] + 97 * F[6] + F[5] + -45 * F[4] + 84 * F[3] + 95 * F[0] - 20 * F[1] + 12 * F[2] ==
12613)
s.add(30 * F[9] + -70 * F[8] + -122 * F[6] + -81 * F[5] + -66 * F[4] + -115 * F[3] + -41 * F[2] + -86 * F[1] - 1
5 * F[0] - 30 * F[7] == -54400)
s.add(-103 * F[9] + 120 * F[7] + 108 * F[5] + 48 * F[3] + -89 * F[2] + 78 * F[1] - 41 * F[0] + 31 * F[4] - (F[6]
 * 64) - 120 * F[8] == -10283)
s.add(71 * F[6] + (F[5] * 128) + 99 * F[4] + -111 * F[2] + 85 * F[1] + 79 * F[0] - 30 * F[3] - 119 * F[7] + 48 *
 F[8] - 16 * F[9] == 22855)
s.add(5 * F[9] + 23 * F[8] + 122 * F[7] + -19 * F[6] + 99 * F[5] + -117 * F[4] + -69 * F[2] + 22 * F[1] - 98 * F
[0] + 10 * F[3] == -2944)
s.add(-54 * F[9] + -23 * F[7] + -82 * F[2] + -85 * F[0] + 124 * F[1] - 11 * F[3] - 8 * F[4] - 60 * F[5] + 95 * F
[6] + 100 * F[8] == -2222)
s.add(-83 * F[9] + -111 * F[5] + -57 * F[0] + 41 * F[1] + 73 * F[2] - 18 * F[3] + 26 * F[4] + 16 * F[6] + 77 * F
[7] - 63 * F[8] == -13258)
s.add(81 * F[9] + -48 * F[8] + 66 * F[7] + -104 * F[6] + -121 * F[5] + 95 * F[4] + 85 * F[3] + 60 * F[2] + -85 *
 F[0] + 80 * F[1] == -1559)
s.add(101 * F[9] + -85 * F[8] + 7 * F[6] + 117 * F[5] + -83 * F[4] + -101 * F[3] + 90 * F[2] + -28 * F[1] + 18 *
 F[0] - F[7] == 6308)
s.add(99 * F[9] + -28 * F[8] + 5 * F[7] + 93 * F[6] + -18 * F[5] + -127 * F[4] + 6 * F[3] + -9 * F[2] + -93 * F[
1] + 58 * F[0] == -1697)

Input = []
if s.check() == sat:
    result = s.model()
for i in F:
    Input.append(chr(eval(str(result[i]))))
print(''.join(Input))
```

运行得到input，既v5

```
>>>
==== RESTART
F0uRTy_7w@
>>>
```

后面的sub_C50函数就可以直接跳过不分析，把文件甩进虚拟机直接运行即可得到flag



到此，此题解完。

其实我也有想着把sub_C50分析一下也用exp写出解，但是相比于直接运行，就有些费时费力了，而且过多的往算法里面钻就有点偏离玩破解的本意了，不如干脆去研究算法算了。这里只是个人的一点见解，要是有大佬写了exp的可以在我下面评论我好康(bai)康(piao)。嘿嘿~