[gonbe5] HGAME 2019 week-3 writeup

RE

Math 简单

打开 IDA 发现前面一部分类似于一个投骰子的函数猜中往下进行，猜错直接退出

```
__int64 v26; // [sp+38h] [bp-18h]@1

v26 = *MK_FP(__FS__, 40LL);
LODWORD(v3) = std::operator<<<std::char_traits<char>>(
                &std::cout,
                "to continue, you have to guess the value of my dice first!",
                envp);
std::ostream::operator<<(v3, &std::endl<char,std::char_traits<char>>);
v24 = rolling_dice();
std::operator<<<std::char_traits<char>>(&std::cout, "now the dice have been rolled, guess what it is: ", v4);
std::istream::operator>>(&std::cin, &v23);
v5 = v23;
LODWORD(v7) = std::operator<<<std::char_traits<char>>(&std::cout, "expected: ", v6);
LODWORD(v8) = std::ostream::operator<<(v7, v24);
LODWORD(v10) = std::operator<<<std::char_traits<char>>(v8, ", guess: ", v9);
LODWORD(v11) = std::ostream::operator<<(v10, v5);
std::ostream::operator<<(v11, &std::endl<char,std::char_traits<char>>);
if ( v23 != v24 )
{
  LODWORD(v13) = std::operator<<<std::char_traits<char>>(&std::cout, "you are bad at guessing dice", v12);
  std::ostream::operator<<(v13, &std::endl<char,std::char_traits<char>>);
  exit(0);
}
std::operator<<<std::char_traits<char>>(
  &std::cout,
  "wow, you are good at dice-guessing, now give me your flag: ",
  v12);
std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::basic_string(&v25);
std::operator>><char,std::char_traits<char>,std::allocator<char>>(&std::cin, &v25);
LODWORD(v14) = std::__cxx11::basic_string<char,std::char_traits<char>,std::allocator<char>>::length(&v25);
if ( v14 != 32 )
{
  LODWORD(v16) = std::operator<<<std::char_traits<char>>(&std::cout, "assert len(flag) == 32", v15);
  std::ostream::operator<<(v16, &std::endl<char,std::char_traits<char>>);
  exit(0);
}
LODWORD(v17) = std::operator<<<std::char_traits<char>>(&std::cout, "now the math part...", v15);
std::ostream::operator<<(v17, &std::endl<char,std::char_traits<char>>);
if ( math_part() )
{
  LODWORD(v19) = std::operator<<<std::char_traits<char>>(
```

跟进 math_part()

```
                                                    + 28 * *(flag + 24)
                                                    + 89 * *(flag + 1)
                                                    + 88 * *(flag + 18)
                                                    + 3 * *(flag + 3)
                                                    + 59 * *(flag + 20)
                                                    + 80 * *(flag + 23)
                                                    + 49 * *(flag + 17)
                                                    + 56 * *(flag + 21)
                                                    + 32 * *(flag + 27)
                                                    + 24 * *(flag + 2);
1080                                              if ( 13 * *(flag + 14)
1081                                                 + 73 * *(flag + 19)
1082                                                 + 99 * *(flag + 7)
1083                                                 + 76 * *(flag + 12)
1084                                                 + v32
1085                                                 + 77 * *(flag + 30)
1086                                                 + 18 * *(flag + 6) == 138403 )
1087                                                result = 1;
1088                                            }
```

32 个判别式，都满足则将 result 置 1；很明显解题思路就是解这 32 元一次方程组
刚开始想手动的去扒数据，结果看到眼睛都疼了……好在 oyeye 提供了一个解题脚本
再通过自己写一个正则才最后解出这道题

附上简陋正则脚本（扒数据用）

```
import re
text = ''
file = open('num.txt')
for line in file:
    text = text + line
file.close()
result = re.sub(r"\*\(\w+\s\+\s", "flag[", text)
result2 = re.sub(r"\)", "]", result)
print(result2)
f = open(r'E:\Pycharm\project\num.txt', 'w')
f.write(result2)
f.close()
```

将提取出来的数据填充到 oyeye 提供的脚本中

```
from z3 import *

solver = Solver()
flag = [Int('flag%d' % i) for i in range(32)]

for i in flag:
    solver.add(i >= 32)
    solver.add(i <= 128)

solver.add(19*flag[8] + 33*flag[20] + ..... + 71*flag[0] == 145397)
solver.add(81*flag[10] + 54*flag[14] + .....  63*flag[22] == 127517)
solver.add(45*flag[30] + 94*flag[28] + .....  61*flag[18] == 141411)
solver.add(91*flag[20] + 9*flag[4] + 3 ..... *flag[0] == 117383)
solver.add(91*flag[21] + 92*flag[15] + ..... + 63*flag[10] == 156152)
# 省略
solver.add(3*flag[22] + 93*flag[8] + 7 ..... 31*flag[16] == 117383)
solver.add(90*flag[9] + 29*flag[4] + 3 ..... 56*flag[11] == 155741)
solver.add(85*flag[30] + 14*flag[10] + ..... + 32*flag[18] == 132804)
solver.add(45*flag[9] + 42*flag[18] + .....  + 79*flag[0] == 145568)
solver.add(40*flag[13] + 39*flag[10] + ..... 54*flag[30] == 130175)
solver.add(79*flag[0] + 15*flag[21] + .....  + 92*flag[30] == 171986)
solver.add(52*flag[12] + 40*flag[6] + .....  + 93*flag[0] == 151676)
solver.add(72*flag[0] + 8*flag[25] + 1 .....  49*flag[1] == 128223)
solver.add(89*flag[1] + 28*flag[24] + .....  + 18*flag[6] == 138403)

print('prepare okay')
check = solver.check()

print(check)

if check == sat:
    m = solver.model()
    s = []
    for i in range(32):
        s.append(chr(m[flag[i]].as_long()))
    print(''.join(s))

```

得到 flag

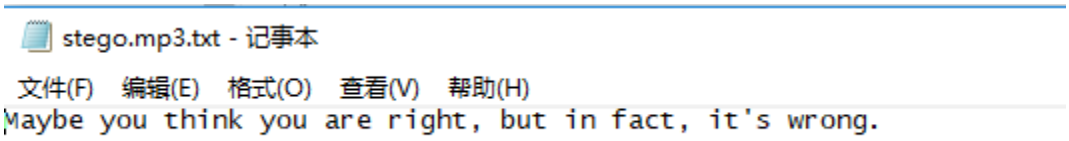
```
hgame{H4ppY#n3w@Y3AR%fr0M-oDiDi}
gonbe5@ubuntu:~/Desktop/RE_week3$
```

hgame{H4ppY#n3w@Y3AR%fr0M-oDiDi}

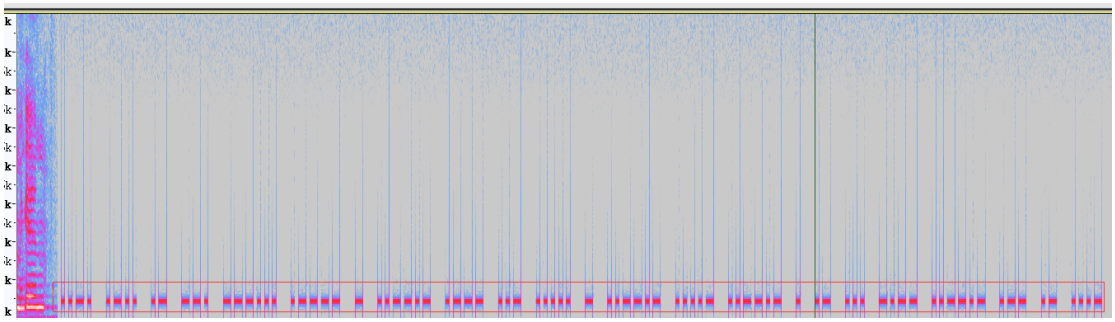
MISC

听听音乐?

将 mp3 下下来发现在音频末尾有一段嘟嘟嘟的类似于敲低电报的声音
先尝试用 MP3stgo 去解



好吧，我就知道没这么简单
接着用 audacity 去分析，查看频谱图



后面这一截有点奇怪，猜测是摩尔斯电码
..-./.-../.-/--./---.../.----/-/..--.-/.---/..-/...../-/..--.-/....-/..--.-/./.-/.../-.--/..--.-/.--/.-/...-/
解出来是：FLAG1TJU5T4EASYWAV
结合题目要求套上 hgame{}和_ 得到 flag:
hgame{1T_JU5T_4_EASY_WAV}