

buuctf——（ACTF新生赛2020）Oruga

原创

re3sry 于 2021-06-09 17:00:00 发布 88 收藏 1

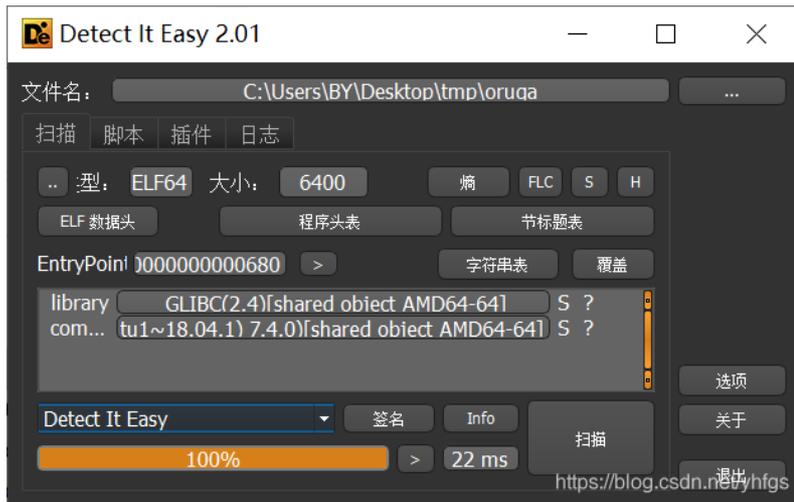
版权声明：本文为博主原创文章，遵循 [CC 4.0 BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) 版权协议，转载请附上原文出处链接和本声明。

本文链接：<https://blog.csdn.net/yhfgs/article/details/117746803>

版权

1.查壳。

无壳，64位。



2. IDA反编译。

分析代码，关键函数为sub_78A,并且flag以actf{ }包裹。

```
1
2
3 v8 = __readfsqword(0x28u);
4 memset(s, 0, 0x19uLL);
5 printf("Tell me the flag:");
6 scanf("%s", s);
7 strcpy(s2, "actf{");
8 for ( i = 0; i <= 4; ++i )
9     s1[i] = s[i];
10 s1[5] = '\0';
11 if ( !strcmp(s1, s2) )
12 {
13     if ( sub_78A((__int64)s) )
14         printf("That's True Flag!");
15     else
16         printf("don't stop trying...");
17     result = 0LL;
18 }
19 else
20 {
21     printf("Format false!");
22     result = 0LL;
23 }
24 return result;
25 }
```

<https://blog.csdn.net/yhfgs>

进入sub_78A函数。

分析代码可知这是个迷宫题

map为word_201020且每16个为一行。

W,E,M,J分别是 上, 右, 下, 左

从左上角开是到“! (33)”结束。

```
5
7 v2 = 0;
8 v3 = 5;
9 v4 = 0;
0 while ( word_201020[v2] != 33 )
1 {
2     v2 -= v4;
3     if ( *(_BYTE *)(v3 + a1) != 'W' || v4 == -16 )
4     {
5         if ( *(_BYTE *)(v3 + a1) != 'E' || v4 == 1 )
6         {
7             if ( *(_BYTE *)(v3 + a1) != 'M' || v4 == 16 )
8             {
9                 if ( *(_BYTE *)(v3 + a1) != 'J' || v4 == -1 )
0                 return 0LL;
1                 v4 = -1;
2             }
3             else
4             {
5                 v4 = 16;
6             }
7         }
8         else
9         {
0             v4 = 1;
1         }
2     }
3     else
4     {
5         v4 = -16;
6     }
7     ++v3;
8     while ( !word_201020[v2] )
9     {
0         if ( v4 == -1 && (v2 & 0xF) == 0 )
1         return 0LL;
2         if ( v4 == 1 && v2 % 16 == 15 )
3         return 0LL;
4         if ( v4 == 16 && (bool)igned int)(v2 - 240) <= 0xF )
5         return 0LL;
6         if ( v4 == -16 && (unsigned int)(v2 + 15) <= 0x1E )
7         return 0LL;
8         v2 += v4;
9     }
0 }
1 return *(_BYTE *)(v3 + a1) == 125;
2 }
```

<https://blog.csdn.net/yhfgs>

<https://blog.csdn.net/yhfgs>

第二个while是说:

在最左的一列不能左走

在最右的一列不能右走

在最下的一列不能下走

在最上的一列不能上走

3.map:

```
File Edit Format Run Options Window I
0 0 0 0 # 0 0 0 0 0 0 0 # # # #
0 0 0 # # 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 P P 0 0 0
0 0 0 L 0 0 0 0 0 0 0 P P 0 0 0
0 0 0 L 0 0 0 0 0 0 0 P 0 0 0 0
0 0 L L 0 0 0 0 0 0 0 P 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 P 0 0 0 0
# 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 # 0 0 0
0 0 0 0 0 0 M M M 0 0 0 # 0 0 0
0 0 0 0 0 0 M M M 0 0 0 0 E E
0 0 0 0 0 M 0 M 0 M 0 0 0 0 E 0
0 0 0 0 0 0 0 0 0 0 0 0 0 E E
T T T I 0 M 0 M 0 M 0 0 0 0 E 0
0 T 0 1 0 M 0 M 0 M 0 0 0 0 E 0
0 T 0 1 0 M 0 M 0 M 0 0 0 E E
```

4.get flag

flag{MEWEMEWJMEWJM}