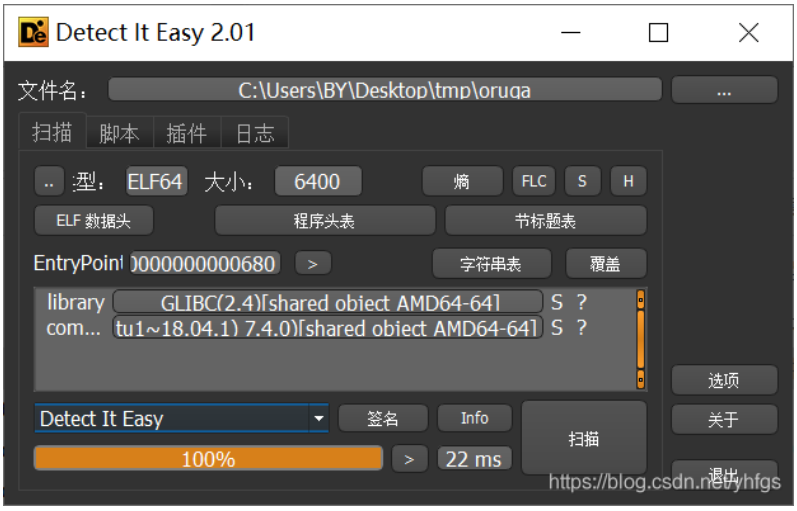
buuctf——（ACTF新生赛2020）Oruga

原创

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

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。

```
  v8 = __readfsqword(0x28u);
  memset(s, 0, 0x19uLL);
  printf("Tell me the flag:");
  scanf("%s", s);
  strcpy(s2, "actf{");
  for ( i = 0; i <= 4; ++i )
    s1[i] = s[i];
  s1[5] = '\0';
  if ( !strcmp(s1, s2) )
  {
    if ( sub_78A((__int64)s) )
      printf("That's True Flag!");
    else
      printf("don't stop trying...");
    result = 0LL;
  }
  else
  {
    printf("Format false!");
    result = 0LL;
  }
  return result;
}
```


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。

```
  v2 = 0;
  v3 = 5;
  v4 = 0;
  while ( word_201020[v2] != 33 )
  {
    v2 -= v4;
    if ( *(_BYTE *)(v3 + a1) != 'W' || v4 == -16 )
    {
      if ( *(_BYTE *)(v3 + a1) != 'E' || v4 == 1 )
      {
        if ( *(_BYTE *)(v3 + a1) != 'M' || v4 == 16 )
        {
          if ( *(_BYTE *)(v3 + a1) != 'J' || v4 == -1 )
            return 0LL;
          v4 = -1;
        }
        else
        {
          v4 = 16;
        }
      }
      else
      {
        v4 = 1;
      }
```



```
    }
    else
    {
      v4 = -16;
    }
    ++v3;
    while ( !word_201020[v2] )
    {
      if ( v4 == -1 && (v2 & 0xF) == 0 )
        return 0LL;
      if ( v4 == 1 && v2 % 16 == 15 )
        return 0LL;
      if ( v4 == 16 && (bool gned int)(v2 - 240) <= 0xF )
        return 0LL;
      if ( v4 == -16 && (unsigned int)(v2 + 15) <= 0x1E )
        return 0LL;
      v2 += v4;
    }
  }
  return *(_BYTE *)(v3 + a1) == 125;
}
```


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

```
0 0 0 0 # 0 0 0 0 0 0 # # # #
0 0 0 # # 0 0 0 o o 0 0 0 0 0 0
0 0 0 0 0 0 0 0 o o 0 P P 0 0 0
0 0 0 L 0 o o 0 o o 0 P P 0 0 0
0 0 0 L 0 o o 0 o o 0 P 0 0 0 0
0 0 L L 0 o o 0 0 0 0 P 0 0 0 0
0 0 0 0 0 o o 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 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 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}