re学习笔记（59）BUUCTF - re - [ACTF新生赛2020]Oruga

原创

Forgo7ten 于 2020-04-03 13:01:36 发布 1451 收藏 1

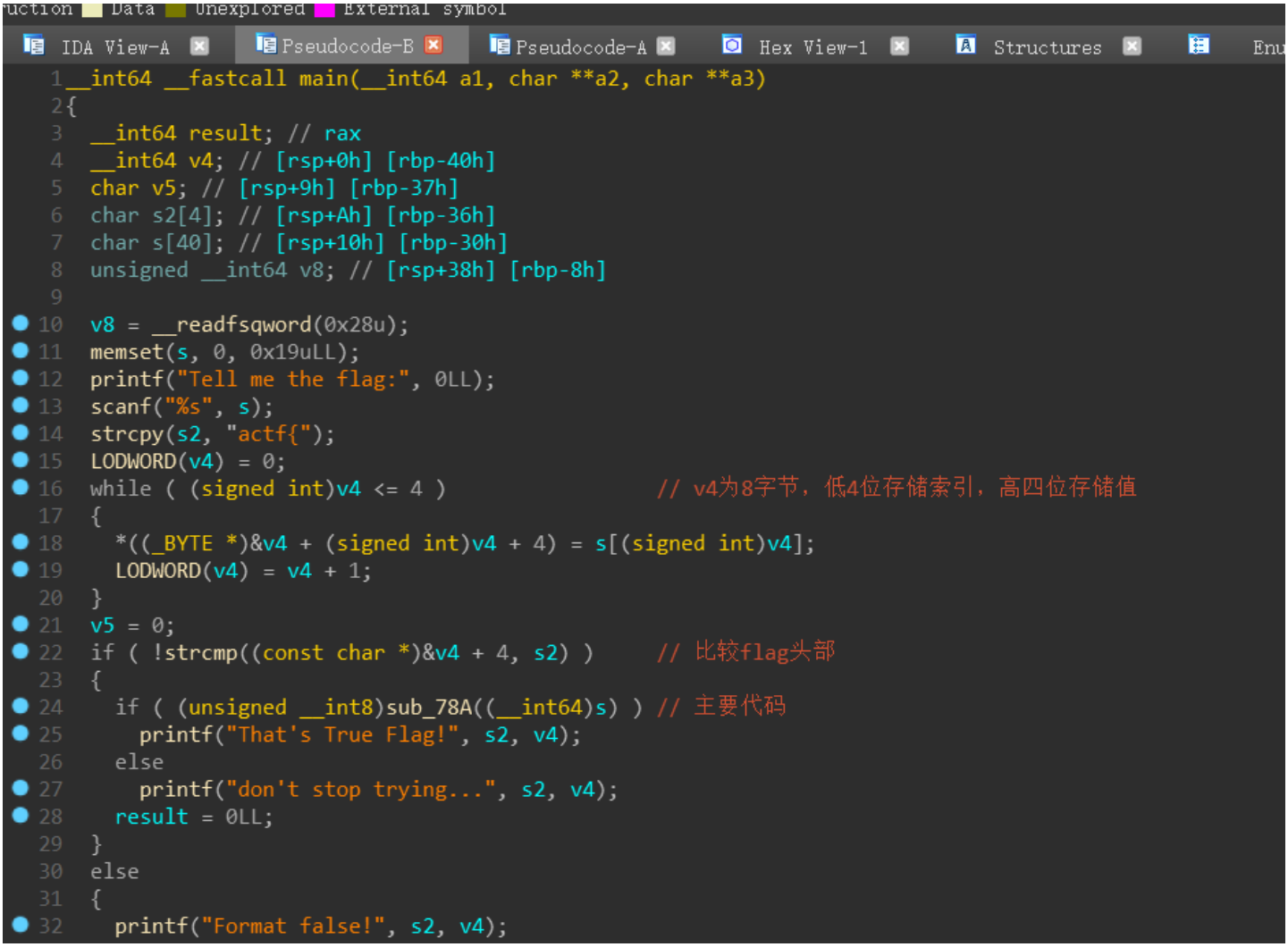
分类专栏： ctf小白成长ing # reverse 文章标签： c语言 信息安全 CTF maze reverse

本文链接：https://blog.csdn.net/Palmer9/article/details/105274167

新手一枚，如有错误（不足）请指正，谢谢！！

题目链接：[ACTF新生赛2020]Oruga

IDA64位载入，进入main函数



```
000009A4|main:24 (9A4)| |
```


进入sub_78A()函数

IDA View-A | Pseudocode-B | Pseudocode-A | Hex View-1 | Structure

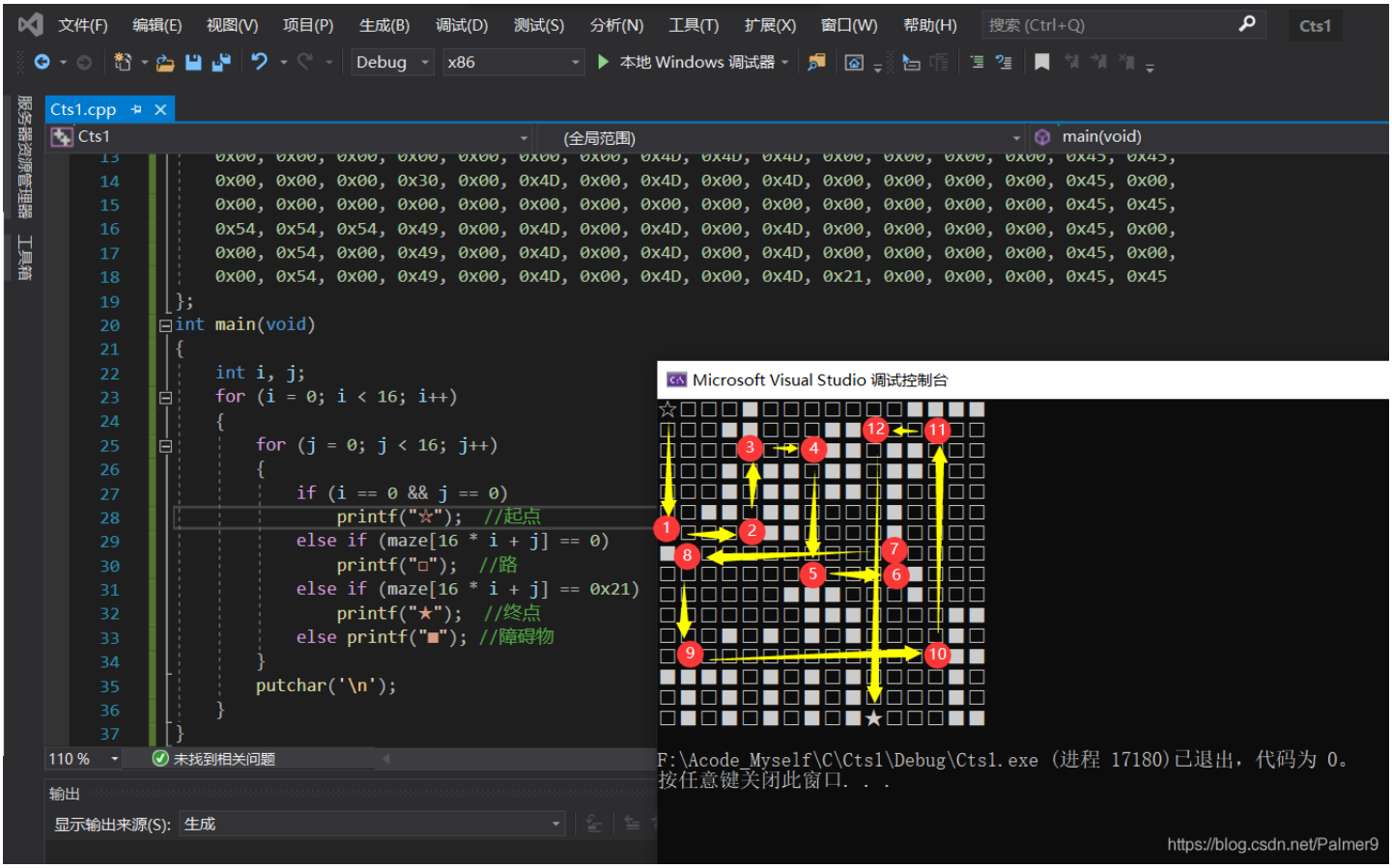
```
 7  v2 = 0;
 8  v3 = 5;
 9  v4 = 0;
10  while ( byte_201020[v2] != 0x21 )
11  {
12    v2 -= v4;                                   // v2 当前坐标
13                                                // 减去上次移动多移动的一次
14    if ( *(_BYTE *)(v3 + a1) != 'W' || v4 == -16 )
15    {
16      if ( *(_BYTE *)(v3 + a1) != 'E' || v4 == 1 )
17      {
18        if ( *(_BYTE *)(v3 + a1) != 'M' || v4 == 16 )
19        {
20          if ( *(_BYTE *)(v3 + a1) != 'J' || v4 == -1 )
21            return 0LL;                         // 判断最后一位是否为 '}'
22          v4 = -1;                              // a1[v3] = 'J'  向左移动
23        }
24        else
25        {
26          v4 = 16;                              // a1[v3] == 'M'  向下移动
27        }
28      }
29      else
30      {
31        v4 = 1;                                 // a1[v3] == 'E'  向右移动
32      }
33    }
34    else
35    {
36      v4 = -16;                                 // a1[v3] == 'W'  向上移动
37    }
38    ++v3;
```



```
34    else
35    {
36      v4 = -16;                                 // a1[v3] == 'W'  向上移动
37    }
38    ++v3;
39    while ( !byte_201020[v2] )                  // 当坐标所在位置为0时
40    {
41      if ( v4 == -1 && !(v2 & 15) )             // 在最左第一列，不能 向左移动
42        return 0LL;
43      if ( v4 == 1 && v2 % 16 == 15 )           // 在最右最后一列，不能向右移动
44        return 0LL;
45      if ( v4 == 16 && (unsigned int)(v2 - 240) <= 0xF )// 在最后一行，不能 向下移动
46        return 0LL;
47      if ( v4 == -16 && (unsigned int)(v2 + 15) <= 30 )// 在第一行，不能 向上移动
48        return 0LL;
49      v2 += v4;                                 // 一直移动
50    }
51  }
52  return *(_BYTE *)(v3 + a1) == '}';
53}                                               // 判断最后一位是否为 '}'
```

类似于象棋里面的‘车’吧，不过这个要走到障碍物才停下来
提取迷宫数据，写脚本

```
#include <stdio.h>
char maze[256] = {
    0x00, 0x00, 0x00, 0x00, 0x23, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x23, 0x23, 0x23, 0x23,
    0x00, 0x00, 0x00, 0x23, 0x23, 0x00, 0x00, 0x00, 0x4F, 0x4F, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4F, 0x4F, 0x00, 0x50, 0x50, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x4C, 0x00, 0x4F, 0x4F, 0x00, 0x4F, 0x4F, 0x00, 0x50, 0x50, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x4C, 0x00, 0x4F, 0x4F, 0x00, 0x4F, 0x4F, 0x00, 0x50, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x4C, 0x4C, 0x00, 0x4F, 0x4F, 0x00, 0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x4F, 0x4F, 0x00, 0x00, 0x00, 0x00, 0x50, 0x00, 0x00, 0x00, 0x00,
    0x23, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x23, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4D, 0x4D, 0x4D, 0x00, 0x00, 0x00, 0x23, 0x00, 0x00, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4D, 0x4D, 0x4D, 0x00, 0x00, 0x00, 0x00, 0x45, 0x45,
    0x00, 0x00, 0x00, 0x30, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x00, 0x45, 0x00,
    0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x45, 0x45,
    0x54, 0x54, 0x54, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x00, 0x45, 0x00,
    0x00, 0x54, 0x00, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x00, 0x45, 0x00,
    0x00, 0x54, 0x00, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x21, 0x00, 0x00, 0x00, 0x45, 0x45
};
int main(void)
{
    int i, j;
    for (i = 0; i < 16; i++)
    {
        for (j = 0; j < 16; j++)
        {
            if (i == 0 && j == 0)
                printf("☆");  //起点
            else if (maze[16 * i + j] == 0)
                printf("□");  //路
            else if (maze[16 * i + j] == 0x21)
                printf("★");  //终点
            else printf("■"); //障碍物
        }
        putchar('\n');
    }
}
```



则为`MEWEMEWJMEWJM`

则flag为`actf{MEWEMEWJMEWJM}`