

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](#)

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/Palmer9/article/details/105274167>

版权



[ctf小白成长ing](#) 同时被 2 个专栏收录

112 篇文章 6 订阅

订阅专栏



[reverse](#)

113 篇文章 6 订阅

订阅专栏

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

```
unction Data Unexplored External symbol
IDA View-A Pseudocode-B Pseudocode-A Hex View-1 Structures
1 __int64 __fastcall main(__int64 a1, char **a2, char **a3)
2 {
3     __int64 result; // rax
4     __int64 v4; // [rsp+0h] [rbp-40h]
5     char v5; // [rsp+9h] [rbp-37h]
6     char s2[4]; // [rsp+Ah] [rbp-36h]
7     char s[40]; // [rsp+10h] [rbp-30h]
8     unsigned __int64 v8; // [rsp+38h] [rbp-8h]
9
10    v8 = __readfsqword(0x28u);
11    memset(s, 0, 0x19uLL);
12    printf("Tell me the flag:", 0LL);
13    scanf("%s", s);
14    strcpy(s2, "actf{");
15    LODWORD(v4) = 0;
16    while ( (signed int)v4 <= 4 ) // v4为8字节, 低4位存储索引, 高四位存储值
17    {
18        *((_BYTE *)&v4 + (signed int)v4 + 4) = s[(signed int)v4];
19        LODWORD(v4) = v4 + 1;
20    }
21    v5 = 0;
22    if ( !strcmp((const char *)&v4 + 4, s2) ) // 比较flag头部
23    {
24        if ( (unsigned __int8)sub_78A((__int64)s) ) // 主要代码
25            printf("That's True Flag!", s2, v4);
26        else
27            printf("don't stop trying...", s2, v4);
28        result = 0LL;
29    }
30    else
31    {
32        printf("Format false!", s2, v4);
```

进入sub_78A()函数

```

IDA View-A x Pseudocode-B x Pseudocode-A x Hex View-1 x 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;

```

https://blog.csdn.net/Palmer9

```

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 }

```

https://blog.csdn.net/Palmer9

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, 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, 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');
    }
}
```

文件(F) 编辑(E) 视图(V) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T) 扩展(X) 窗口(W) 帮助(H) 搜索 (Ctrl+Q) Cts1

Debug x86 本地 Windows 调试器

Cts1.cpp Cts1 (全局范围) main(void)

```

13 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x4D, 0x4D, 0x4D, 0x00, 0x00, 0x00, 0x45, 0x45,
14 0x00, 0x00, 0x00, 0x30, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x45, 0x00,
15 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x45, 0x45,
16 0x54, 0x54, 0x54, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x45, 0x00,
17 0x00, 0x54, 0x00, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x00, 0x00, 0x45, 0x00,
18 0x00, 0x54, 0x00, 0x49, 0x00, 0x4D, 0x00, 0x4D, 0x00, 0x4D, 0x21, 0x00, 0x00, 0x00, 0x45, 0x45
19 };
20 int main(void)
21 {
22     int i, j;
23     for (i = 0; i < 16; i++)
24     {
25         for (j = 0; j < 16; j++)
26         {
27             if (i == 0 && j == 0)
28                 printf("☆"); //起点
29             else if (maze[16 * i + j] == 0)
30                 printf("□"); //路
31             else if (maze[16 * i + j] == 0x21)
32                 printf("★"); //终点
33             else printf("■"); //障碍物
34         }
35         putchar('\n');
36     }
37 }

```

Microsoft Visual Studio 调试控制台

F:\Acode_Myself\C\Cts1\Debug\Cts1.exe (进程 17180) 已退出, 代码为 0。
按任意键关闭此窗口。

110 % 未找到相关问题

输出
显示输出来源(S): 生成

服务器资源管理器 工具箱

https://blog.csdn.net/Palmer9

则为 MEWEMEWJMEWJM

则flag为 actf{MEWEMEWJMEWJM}