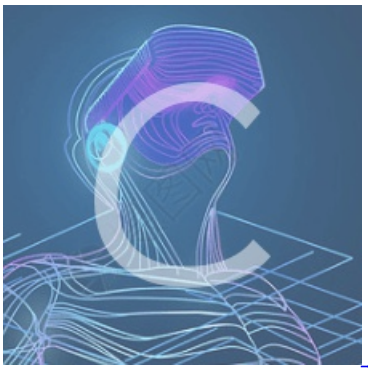
BUUCTF_不一样的 flag

原创

ZYen12138 于 2020-10-16 22:10:14 发布 223 收藏 1

分类专栏： # BUUCTF CTF

本文链接：https://blog.csdn.net/weixin_46009088/article/details/109125500



BUUCTF 同时被 2 个专栏收录

17 篇文章 2 订阅

订阅专栏

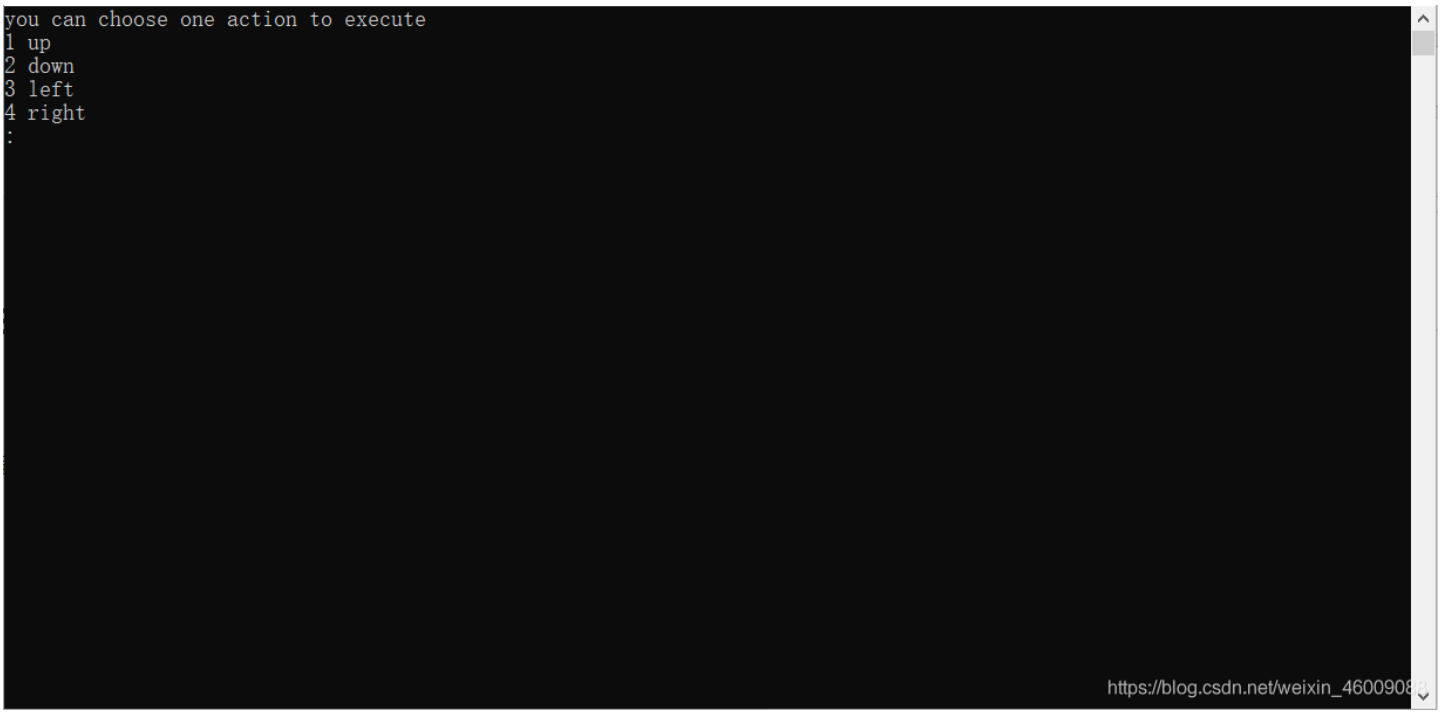
CTF

17 篇文章 0 订阅

订阅专栏

BUUCTF_不一样的 flag

打开程序：



啥玩意？打1，2，3，4 都没用，用IDA加载程序，找到主函数，如下：

```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  char v3; // [esp+17h] [ebp-35h]
  int v4; // [esp+30h] [ebp-1Ch]
  int v5; // [esp+34h] [ebp-18h]
  signed int v6; // [esp+38h] [ebp-14h]
  int i; // [esp+3Ch] [ebp-10h]
  int v8; // [esp+40h] [ebp-Ch]

  __main();
  v4 = 0;
  v5 = 0;
  qmemcpy(&v3, _data_start__, 0x19u);
  while ( 1 )
  {
    puts("you can choose one action to execute");
    puts("1 up");
    puts("2 down");
    puts("3 left");
    printf("4 right\n:");
    scanf("%d", &v6);
    if ( v6 == 2 )
    {
      ++v4;
    }
    else if ( v6 > 2 )
    {
      if ( v6 == 3 )
      {
        --v5;
      }
      else
      {
        if ( v6 != 4 )
LABEL_13:
          exit(1);
        ++v5;
      }
    }
    else
    {
      if ( v6 != 1 )
        goto LABEL_13;
      --v4;
    }
    for ( i = 0; i <= 1; ++i )
    {
      if ( *(&v4 + i) < 0 || *(&v4 + i) > 4 )
        exit(1);
    }
    if ( *((_BYTE *)&v8 + 5 * v4 + v5 - 41) == '1' )
      exit(1);
    if ( *((_BYTE *)&v8 + 5 * v4 + v5 - 41) == '#' )
    {
      puts("\nok, the order you enter is the flag!");
      exit(0);
    }
  }
}
```

下图函数是将_data_start_拷贝到&v3这个地址上，双击_data_start_查看字符串.

```
qmemcpy(&v3, _data_start__, 0x19u);
```

是一串1，0，*，#数字，一共25个，还是□往下看.

```
__data_start__ db '*11110100001010000101111#',0
```

这个 if 语句下面的 puts("\nok, the order you enter is the flag!")，不就是输出"好的，你输入的顺序就是答案"嘛，在下图中把0x31和0x23转化为字符是"1"，"#"，也就是说走到"1"程序将退出，走到"#"就输出字符串，现在懂了原来是迷宫！！！

```
if ( *((_BYTE *)&v8 + 5 * v4 + v5 - 41) == 0x31 )
  exit(1);
if ( *((_BYTE *)&v8 + 5 * v4 + v5 - 41) == 0x23 )
{
  puts("\nok, the order you enter is the flag!");
  exit(0);
```

25个字符，那刚好是一个5*5的迷宫！

```
* 1 1 1 1
0 1 0 0 0
0 1 0 1 0
0 0 0 1 0
1 1 1 1 #
```

我们走0不走1，得到222441144222的结果，这个就是flag！！！