**1、题目名称（给选手看的，中文/英文，但是不要使用 类型+数字的形式，如web1）**

project

**2、题目类型**

Reverse

**3、题目描述（给选手看的，不可以“直接下载附件”等这样的敷衍描述）**

环境：windows

**4、知识考察点**

AES、vm、自定义检验逆向

**5、解题思路**

题目文件为单个 exe，没有加壳，直接IDA即可。

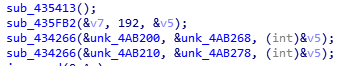
题目思路是要求用户输入一个长度为64的字串，然后将这个字串每两位转换成对应字符的16进制数，得到了一个元素个数为32的数组，将数组放入程序中的AES解密，得到一个长度为24的字串，所以要还原就要写对应的加密函数，再将长24的字串放入VM中处理，又得到一个新的长24的字串，再将这个字串放入检验函数中检验。

IDA打开后分析完成定位到主函数

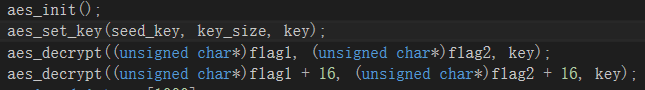


首先是scanf()，然后判断字串长度再判断字串字符的范围，满足即可继续。

其中

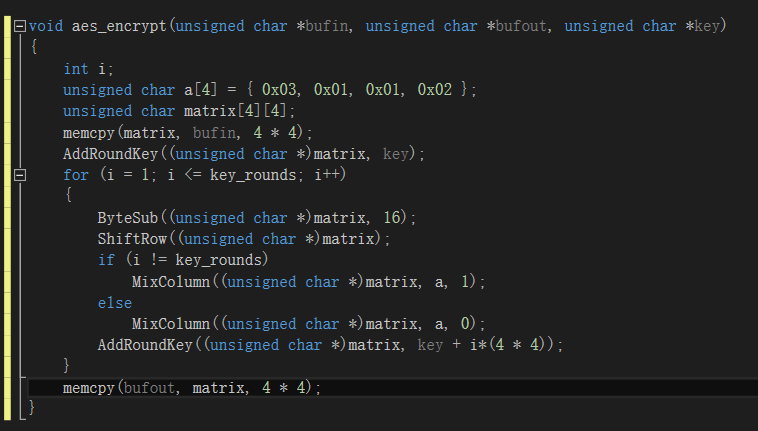


四个函数分别为

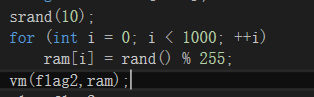


将输入字串放入AES解密，

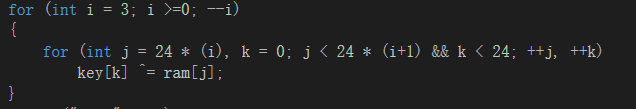
通过逆向可以得到加密还原脚本



然后进入VM，VM模拟了简单的汇编指令，简单的VM很容易通过动态黑盒得到处理过程，所以这次稍微弄复杂了一点，生成了一个数组作为处理的条件



VM对字串进行的是异或处理，但是这次将VM写了4次递归，增加了黑盒难度，要黑盒测试就至少需要知道VM的少量具体动作。这里要节省时间还是得黑盒，通过粗略的审计可以知道VM是通过递归将字串和ram的数组的值以某个方式进行异或，即可写出脚本。



接下来将字串放入检验函数检验。

检验函数有三关，模拟一个弹幕游戏（东方？），只不过是超低配，破产版弹幕游戏。

有起始点，程序有固定变化模式，要求玩家不能碰到固定字符，既需要玩家通过字串进行移动，第一关只能上下左右，第二关可以斜走，第三关可以跳跃。地图环境变化的周期是用户的一次移动，也就是说玩家每次移动，地图就变化一次。具体变化情况就不截图了，有点多。

通过各种逆向可以得到操作方法。不需要脚本解决。

（讲道理我自己都觉得有点难。。。。。

得到最后flag：bctf{181088143FF8C3B07E1F1EBE1543C0AC4C5D8A8258EC5E317FED2E80B060966E}