jarvis oj 软件密码破解-3 Writeup

原创

charlie_heng 于 2017-11-25 19:14:12 发布 448 收藏

分类专栏： 二进制-逆向工程

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/charlie_heng/article/details/78633602

版权

二进制-逆向工程 专栏收录该内容

34 篇文章 3 订阅

订阅专栏

打开软件，首先用mfc工具找一下那几个按钮的处理函数

发现确定按钮默认是不能点击的，看了下函数，发现要输入字符长度为16，且只包含A-F和0-9

先输一下16个1，找了几个比较可疑的函数下断点，发现有个函数断成功了sub_401B80

```
int __stdcall sub_401B80(int a1)
{
  sub_401970();
  if ( (unsigned __int8)(HIBYTE(dword_571458) + BYTE2(dword_571458) + BYTE1(dword_571458) + dword_571458) == 71
    && (unsigned __int8)(HIBYTE(dword_57145C) + BYTE2(dword_57145C) + BYTE1(dword_57145C)) == 3
    && (unsigned __int8)dword_571458 == BYTE1(dword_571458) + 68
    && BYTE1(dword_571458) == BYTE2(dword_571458) + 2
    && BYTE2(dword_571458) == HIBYTE(dword_571458) - 59
    && BYTE2(dword_57145C) == (unsigned __int8)dword_57145C + 10
    && BYTE2(dword_57145C) == HIBYTE(dword_57145C) + 9
    && (unsigned __int8)dword_57145C == BYTE1(dword_57145C) + 52 )
  {
    JUMPOUT(__CS__, 0x1947 + 0x400000);
  }
  return 0;
}
```

http://blog.csdn.net/charlie_heng

首先是调用了一个函数，对输入的字符处理一波，然后在后面一长串判断条件里面判断处理后的东西是否是对的

这里很简单就能解出这八个字符的ASCII码是
119,51,49,108,100,48,110,101

接下来就要看看sub_401970干了什么操作

点进去发现，是一大堆替换操作，感觉像某些比较复杂的算法。

用搜索加密算法的工具发现了AES的sbox，于是去看一波AES的资料，但是发现很奇怪，这里只是进行了AES的sbox操作

于是动态跟一波，发现又是一个魔改算法。。。。

只用到了AES的sbox替换，而且对每个字符替换了4次，替换64轮，位置没有变化

这里还有一个小坑点就是，本来想看看这软件有没有魔改了sbox，于是把sbox提出来，发现的确有些小的不同，于是换了一下，结果发现64轮替换之后的结果和软件输出的结果不同，然后把sbox换回来之后正确了。。。

至于解密的代码我就不写在这里了，根据我上面说的很容易就能写出来