查壳
分析程序
分析算法
写出注册机
校验结果

查壳

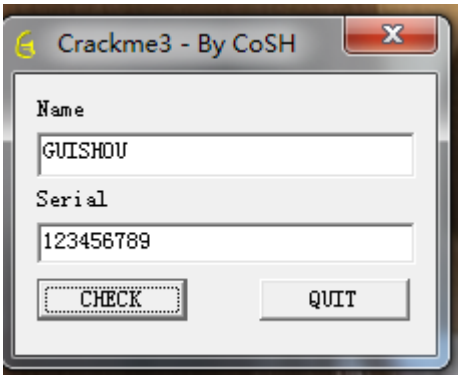
这个Crackme跟027和028是同一个作者，VC6写的，难度为一颗星

分析程序

```
地址      HEX 数据        反汇编                                  注释
004015C3  >   6A 00       push 0x0
004015C5  .   68 6C304000 push cosh_3.0040306C                    ERROR
004015CA  .   68 40304000 push cosh_3.00403040                    One of the Details you entered was wrong
004015CF  .   8B4D E0     mov ecx,[local.8]
004015D2  .   E8 BB020000 call <jmp.&MFC42.#CWnd::MessageBoxA_422
004015D7  .v  EB 14       jmp short cosh_3.004015ED
004015D9  >   6A 00       push 0x0
004015DB  .   68 34304000 push cosh_3.00403034                    YOU DID IT
004015E0  .   68 20304000 push cosh_3.00403020                    Well done,Cracker
004015E5  .   8B4D E0     mov ecx,[local.8]
004015E8  .   E8 A5020000 call <jmp.&MFC42.#CWnd::MessageBoxA_422
004015ED  >   6A 64       push 0x64                               Timeout = 100. ms
004015EF  .   FF15 0020400( call dword ptr ds:[<&KERNEL32.Sleep>] Sleep
004015F5  .   C645 FC 00  mov byte ptr ss:[ebp-0x4],0x0
004015F9  .   8D4D F0     lea ecx,[local.4]
004015FC  .   E8 65010000 call <jmp.&MFC42.#CString::~CString_800
00401601  .   C745 FC FFFF mov [local.1],-0x1
00401608  .   8D4D E4     lea ecx,[local.7]
0040160B  .   E8 56010000 call <jmp.&MFC42.#CString::~CString_800
00401610  .   8B4D F4     mov ecx,[local.3]                       mfc42.722340A5
00401613  .   64:890D 0000( mov dword ptr fs:[0],ecx
0040161A  .   5F          pop edi                                 mfc42.722340A5
0040161B  .   5E          pop esi                                 mfc42.722340A5
```

同样，根据字符串的错误提示，来到函数头的位置，配合IDA的伪代码分析整个算法，

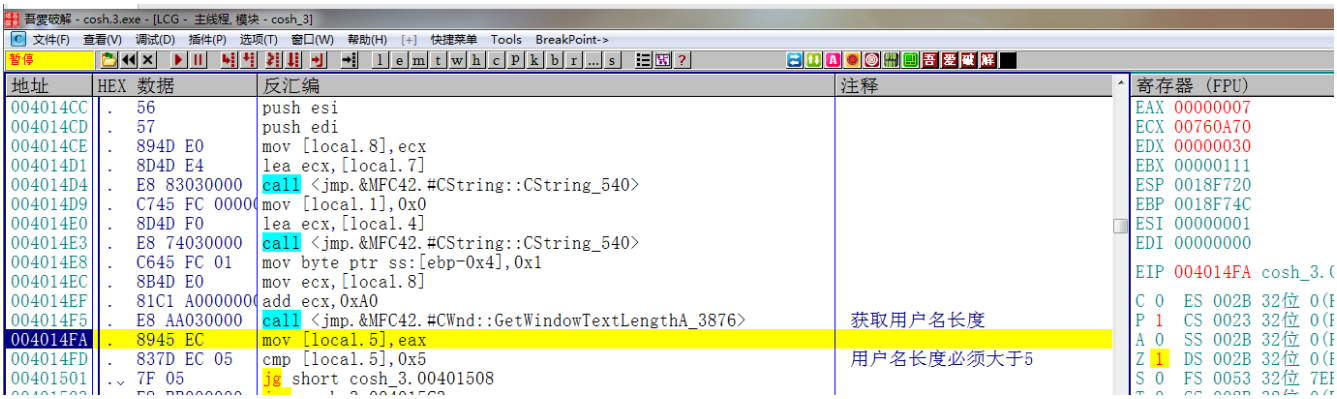
分析算法



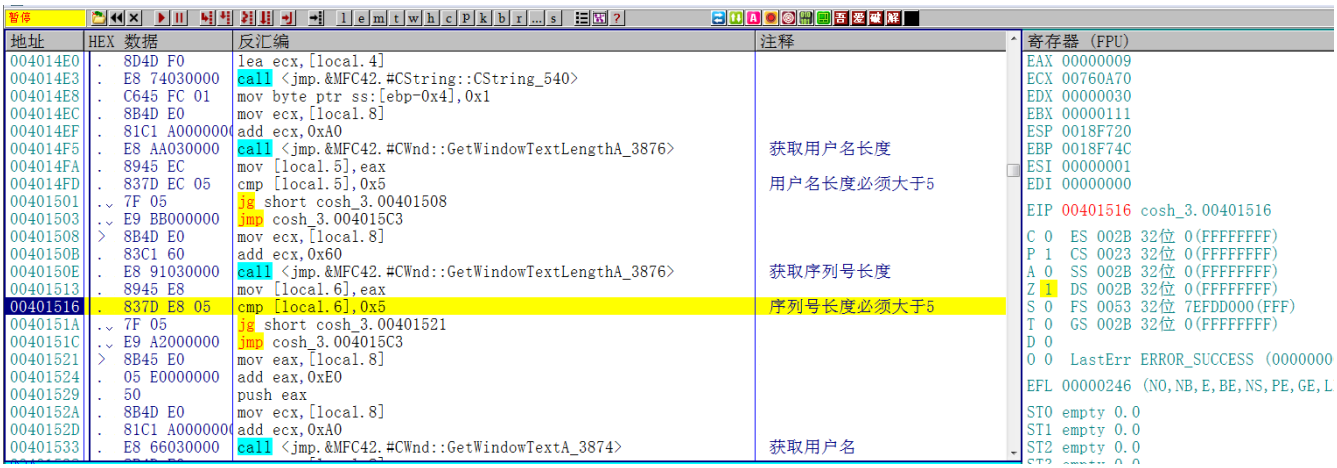
随便输入一个用户名和序列号，算法的校验过程如下。这个Crackme跟028一样有花指令的干扰，如果想在IDA中看到F5的伪代码，需要手动去除花指令，去除的方法请参考我的Crackme027的分析

```
15
16  this_1 = this;
17  CString::CString((CString *)&username_1);
18  v13 = 0;
19  CString::CString((CString *)&Serial);
20  LOBYTE(v13) = 1;
21  UsernameLength = CWnd::GetWindowTextLengthA((CWnd *)((char *)this_1 + 160));// 获取用户名长度
22  if ( UsernameLength > 5 )                     // 用户名长度必须大于5
23  {
24    v10 = CWnd::GetWindowTextLengthA((CWnd *)((char *)this_1 + 96));// 获取序列号长度
25    if ( v10 > 5 )
26    {
27      CWnd::GetWindowTextA((CWnd *)((char *)this_1 + 160), (CWnd *)((char *)this_1 + 224));// 获取用户名
28      CWnd::GetWindowTextA((CWnd *)((char *)this_1 + 96), (CWnd *)((char *)this_1 + 228));// 获取序列号
29      CString::operator=(&username_1, (char *)this_1 + 224);
30      CString::operator=(&Serial, (char *)this_1 + 228);
31      index = 1;
32      username = username_1;
33      do
34        *username++ ^= index++;                 // 用户名第i位的ASCII值和i进行异或
35      while ( *username );
36      i = 10;
37      Serial_1 = Serial;
38      do
39        *Serial_1++ ^= i++;                     // 序列号第i位的ASCII值和i进行异或
40      while ( *Serial_1 );
41      CalcUsername = username_1;
42      CalcSerial = Serial;
43      while ( *CalcUsername == *CalcSerial )    // 比较用户名和序列号的每一位
44      {
45        ++CalcUsername;
46        ++CalcSerial;
47        if ( !*CalcUsername )
48        {
49          CWnd::MessageBoxA(this_1, aWellDoneCracke, aYouDidIt, 0);// 比较通过则提示正确
50          goto LABEL_12;
51        }
52      }
53    }
54  }
55  CWnd::MessageBoxA(this_1, aOneOfTheDetail, aError, 0);// 比较不通过则提示错误
56LABEL_12:
57  Sleep(0x64u);
58  LOBYTE(v13) = 0;
59  CString::~CString((CString *)&Serial);
60  v13 = -1;
61  return CString::~CString((CString *)&username_1);
62}
```

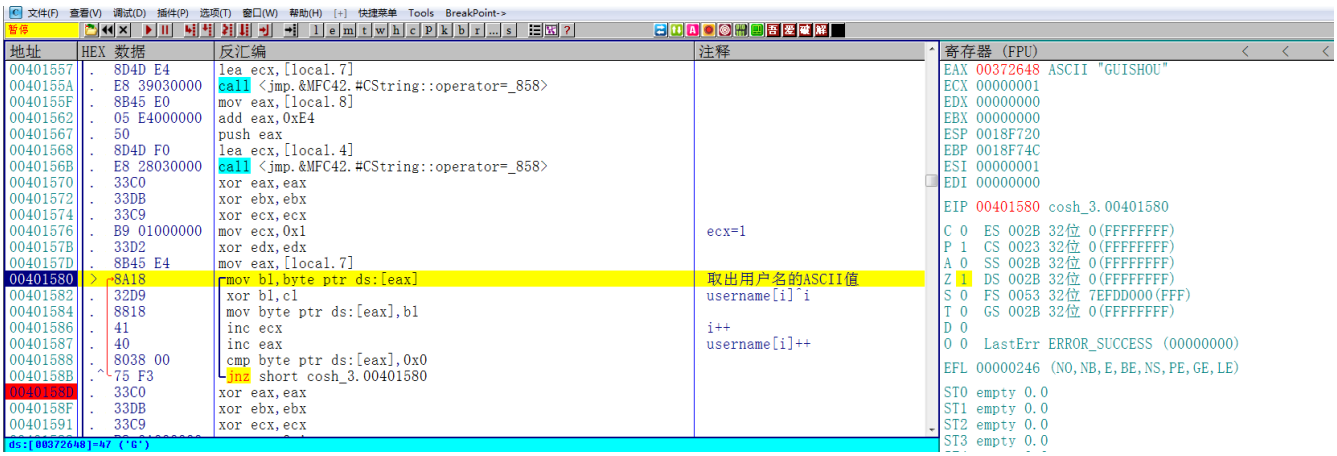
1. 获取用户名长度 比较是否大于5



2. 获取序列号长度 比较是否大于5

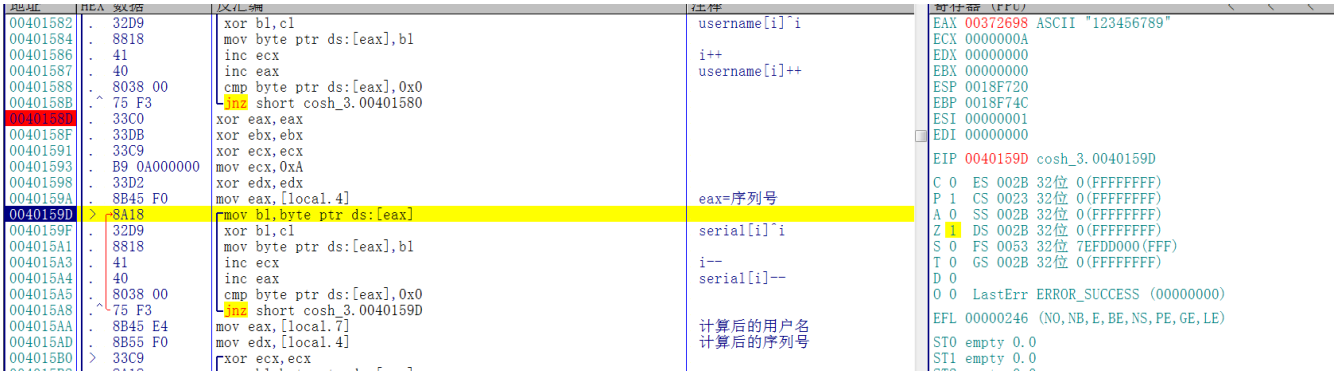


3. 根据用户名计算，得出结果，算法转换为C++代码如下：




```
do
    *username++ ^= index++;
while ( *username );
```

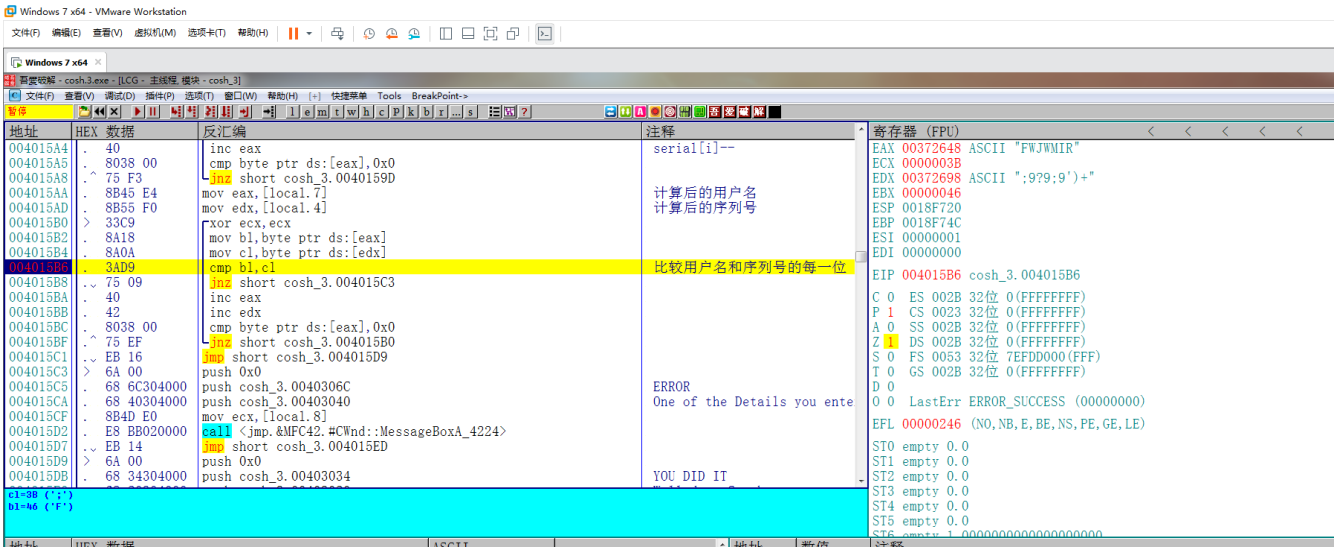
4. 根据序列号计算，得出结果，算法转换为C++代码如下：




```
i = 10;
do
    *Serial_1++ ^= i++;
while ( *Serial_1 );
```

对 没错 两个算法都是直接从IDA的伪代码中拷出来的 直接就能用

5. 循环比较用户名的计算结果和序列号的计算结果是否相等，根据比较的结果提示跳转与否



也就是说用户名和序列号必须要满足程序中的等式才能注册成功

写出注册机

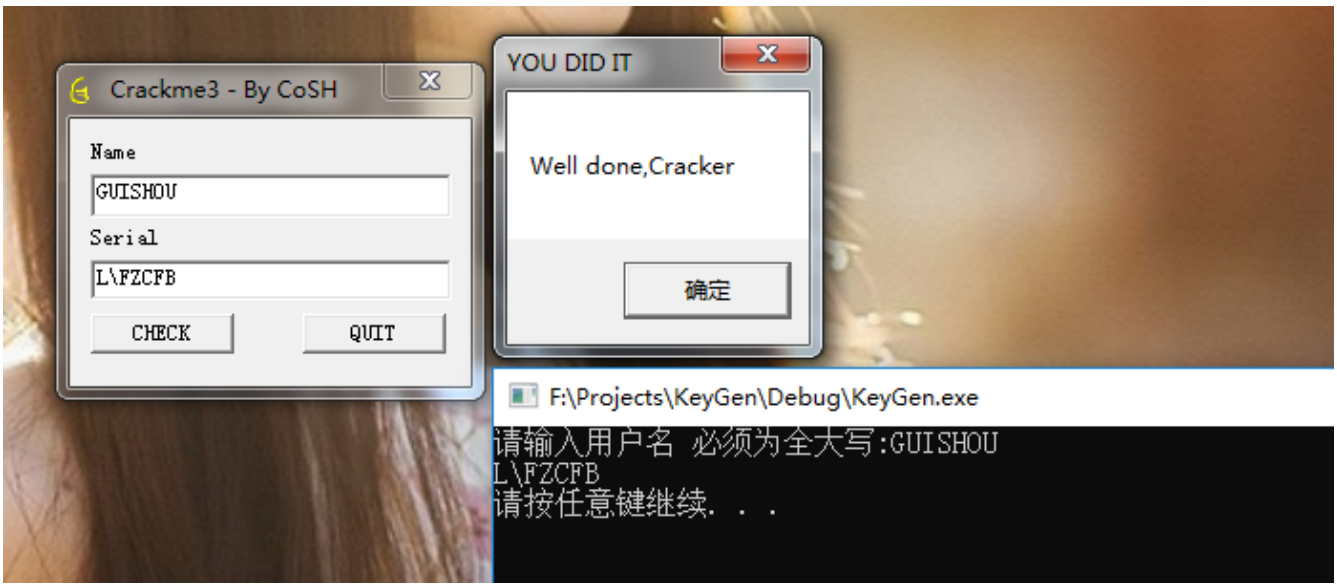
这个程序的注册机也比较好写，首先根据用户名计算出中间结果，然后再根据结果反向逆推出注册码，代码如下:

```
#include <iostream>
#include <windows.h>

using namespace std;

int main()
{
    char username[20] = { 0 };
    char result[20] = { 0 };
    char key[20] = { 0 };
    printf("请输入用户名 必须为全大写:");
```

```
    scanf_s("%s", username, 20);
    int usernameLength = strlen(username);
    if (usernameLength <= 5)
    {
        printf("用户名长度必须大于5");
    }
    //计算中间结果
    for (int i=0;i<usernameLength;i++)
    {
        username[i] ^= i+1;
        result[i] = username[i];
    }

    //根据结果逆推注册码

    for (int i = 10; i < usernameLength+10; i++)
    {
        result[i-10] ^= i;
        key[i-10] = result[i-10];
    }

    printf("%s\n", key);
    system("pause");
    return 0;
}
```

校验结果

输入用户名和计算出来的序列号



提示正确 破解成功

需要相关文件可以到我的Github下载:https://github.com/TonyChen56/160-Crackme