【软件名称】：damn.exe

【软件大小】：30KB

【下载地址】：https://github.com/TonyChen56/160-Crackme

【加壳方式】：Aspack

【保护方式】：Name/Serial

【编译语言】：汇编

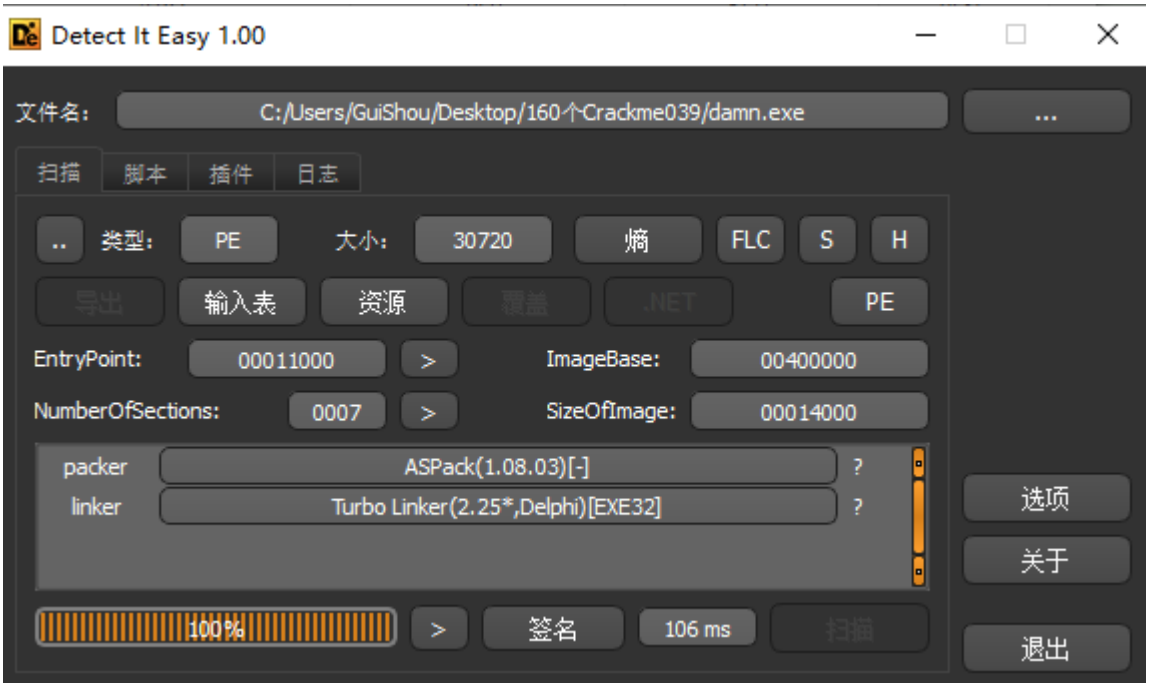
【调试环境】：W10 64

【使用工具】：OD

【破解日期】：2019-05-04

【破解目的】：纯属兴趣

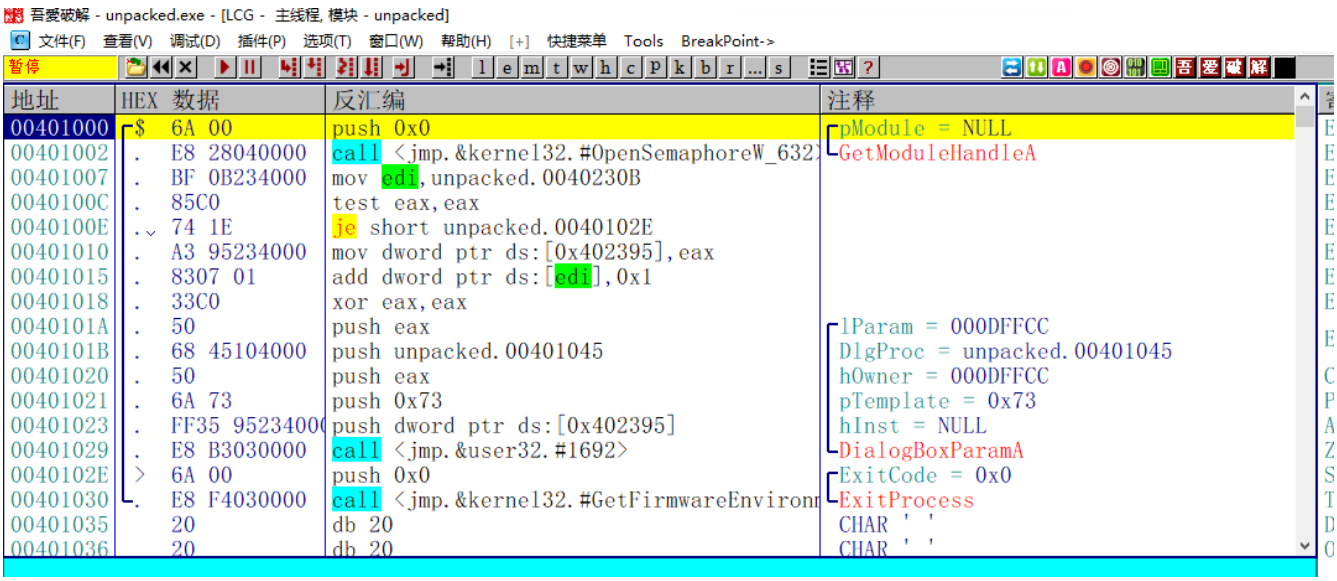
查壳



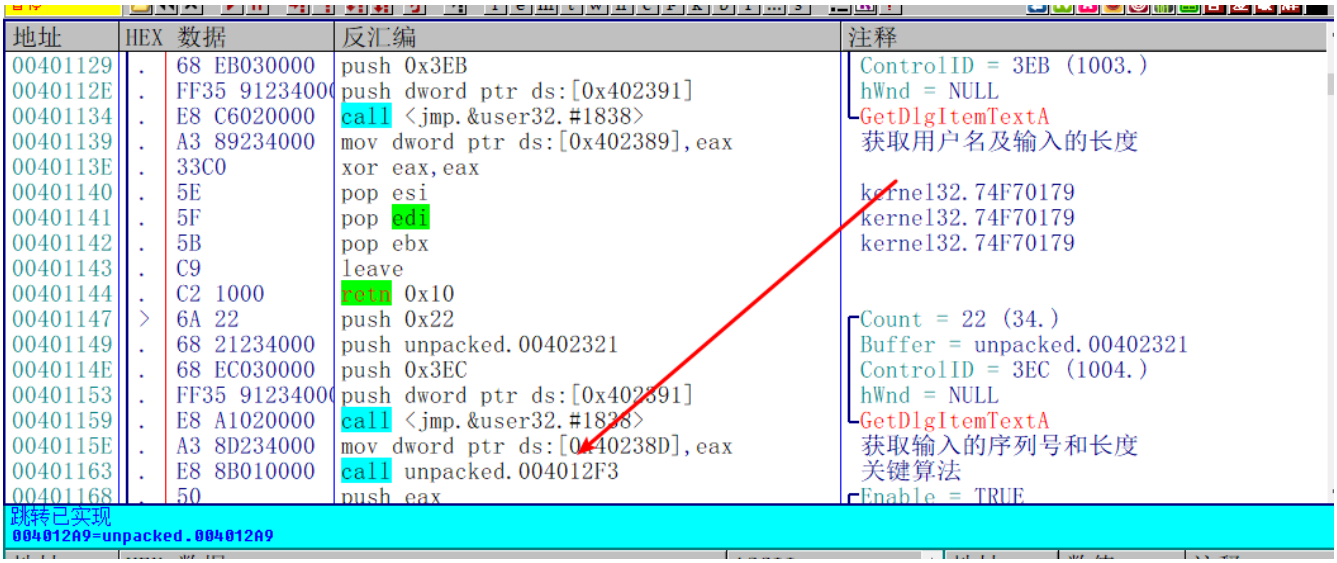
这个目标程序加了个Aspack的壳，用脱壳机或者ESP定律手动脱都行。

寻找突破口

这个程序按钮是被禁用的，所以不太好找突破口。

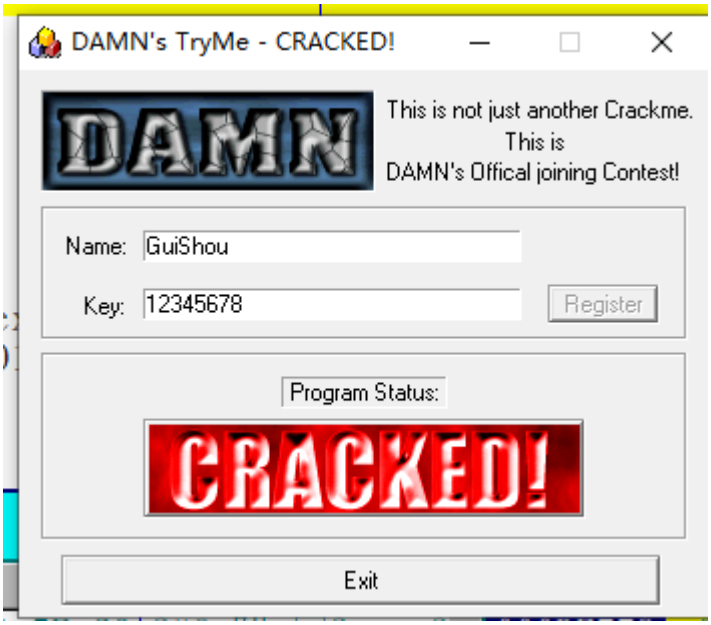


直接载入程序，发现是汇编写的。汇编写的程序不会有编译器生成的无用代码，所以可以一直往下拉，找到关键代码处，当然也可以在**GetDlgItemTextA**函数下断。



一直往下拉，你会发现有两个获取用户输入的函数，分别是获取用户名和序列号。下面那个函数就是关键的算法函数了。

分析算法



随便输入一个用户名，开始分析算法

第一部分

```
地址      HEX 数据    反汇编                                  注释
004012F3  $  90       nop
004012F4  .  8B0D 89234000 mov ecx,dword ptr ds:[0x402389]    ecx用户名长度
004012FA  .  85C9     test ecx,ecx                            unpacked.<ModuleEntryPoint>
004012FC  .v 74 71    je short unpacked.0040136F              用户名不为空
004012FE  .  49       dec ecx                                 长度-1
004012FF  .  8BF1     mov esi,ecx                             unpacked.<ModuleEntryPoint>
00401301  .  BF 53234000 mov edi,unpacked.00402353            edi=username
00401306  .  BB 4E4D4144 mov ebx,0x44414D4E                   ebx=0x44414D4E
0040130B  .  33D2     xor edx,edx                             unpacked.<ModuleEntryPoint>
0040130D  .  8BCA     mov ecx,edx                             unpacked.<ModuleEntryPoint>
0040130F  >  33C0     xor eax,eax
00401311  .  8A040F   mov al,byte ptr ds:[edi+ecx]            al=username[i]
00401314  .  03D0     add edx,eax                             用户名ASCII值累加
00401316  .  D1CB     ror ebx,1                               ebx循环右移1位
00401318  .  D3CB     ror ebx,cl                              ebx循环右移i位
0040131A  .  33DA     xor ebx,edx                             ebx异或username[i]
0040131C  .  3BCE     cmp ecx,esi                             比较循环次数
0040131E  .v 74 03    je short unpacked.00401323
```

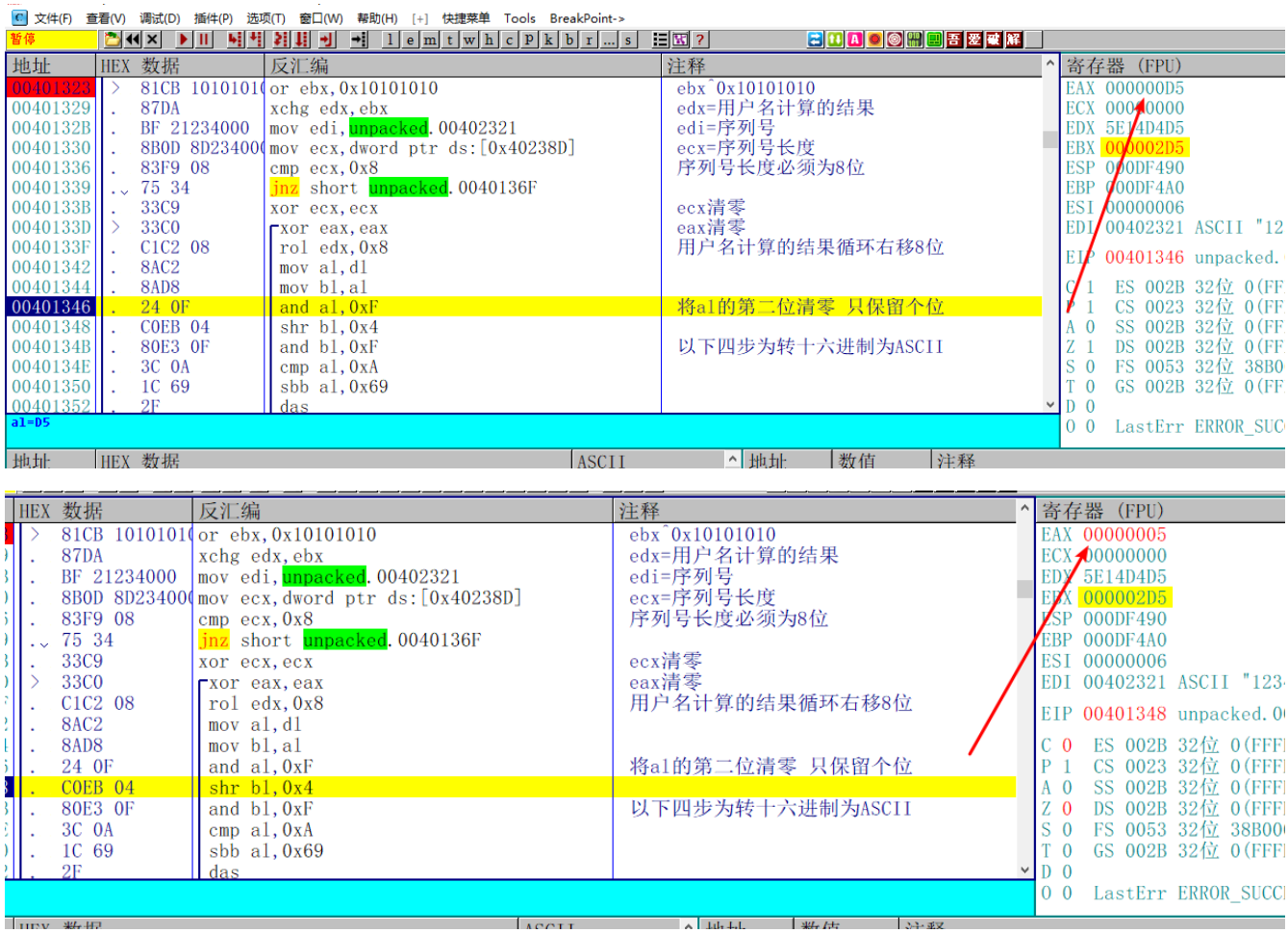
这个算法分为两个部分，第一个部分是对用户名进行处理，算出一个数。这个部分没什么好说的，直接看注释，无非是将用户名的ASCII值经过各种运算计算出来一个数而已。

第二部分

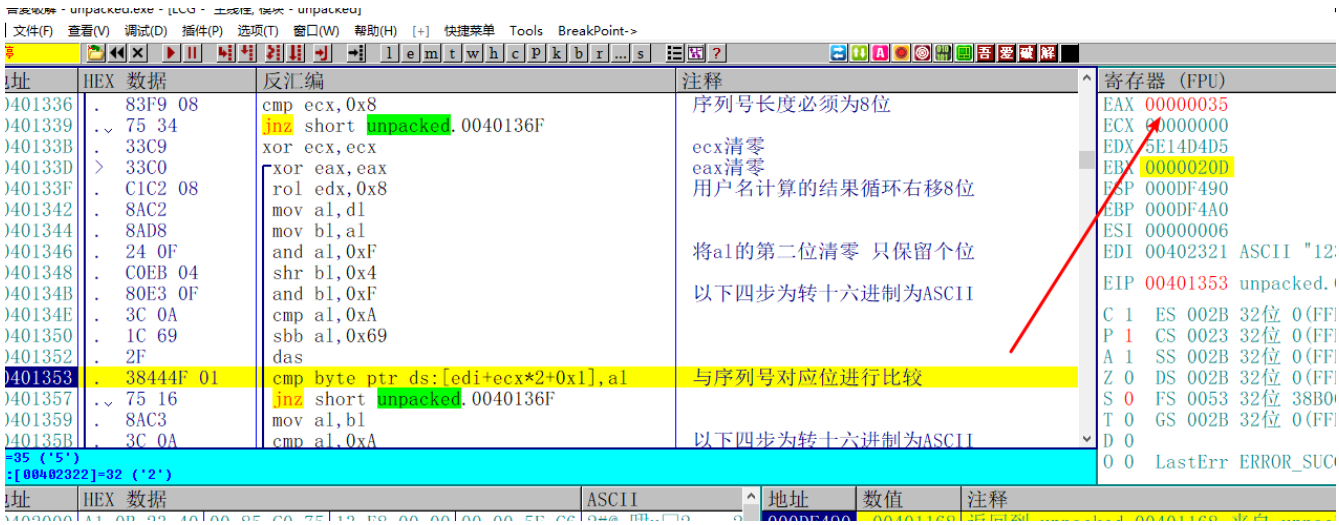
```
00401320  .  41       inc ecx                                 i++
00401321  .^ EB EC    jmp short unpacked.0040130F
00401323  >  81CB 10101010 or ebx,0x10101010                  ebx^0x10101010
00401329  .  87DA     xchg edx,ebx                            edx=用户名计算的结果
0040132B  .  BF 21234000 mov edi,unpacked.00402321            edi=序列号
00401330  .  8B0D 8D234000 mov ecx,dword ptr ds:[0x40238D]    ecx=序列号长度
00401336  .  83F9 08  cmp ecx,0x8                             序列号长度必须为8位
00401339  .v 75 34    jnz short unpacked.0040136F
0040133B  .  33C9     xor ecx,ecx                             ecx清零
0040133D  >  33C0     xor eax,eax                             eax清零
0040133F  .  C1C2 08  rol edx,0x8                             用户名计算的结果循环右移8位
00401342  .  8AC2     mov al,dl
00401344  .  8AD8     mov bl,al
```

程序首先会将第一部分用户名计算的结果和0x10101010进行异或，然后比较序列号的长度是否为8位，不是则报错。

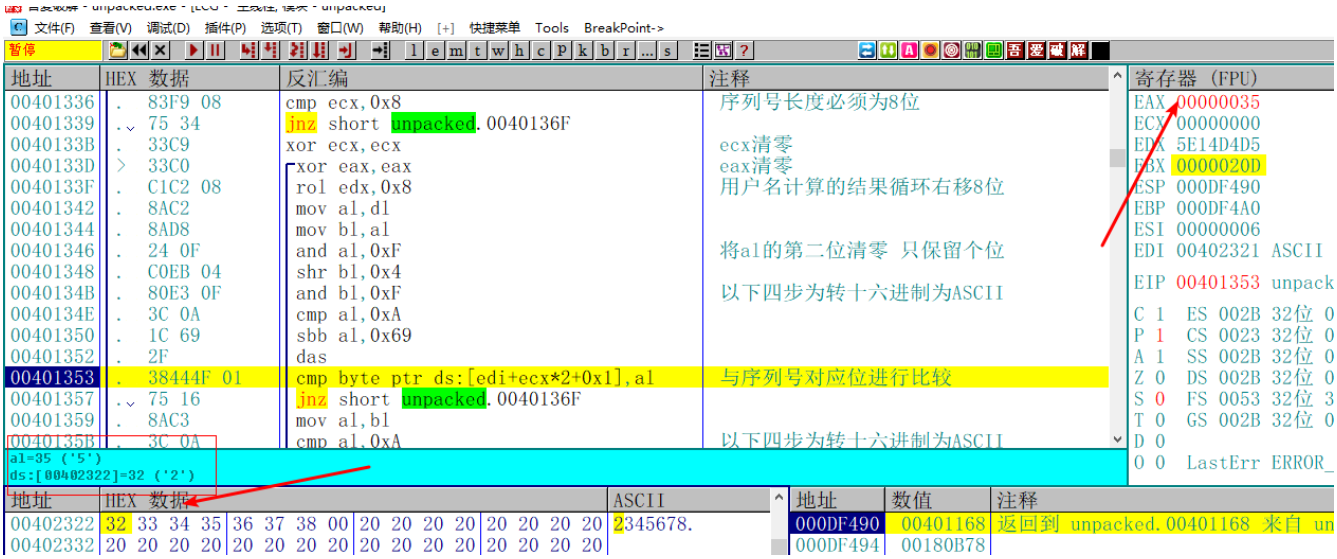
接着将用户名计算的结果循环右移8位，然后拆分为高位和低位，



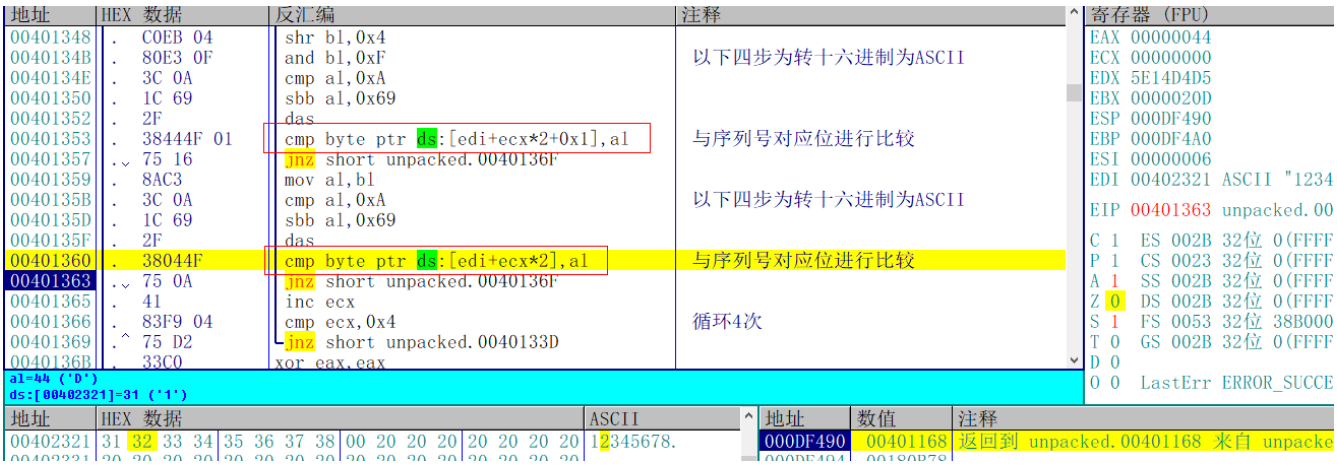
然后将al和0xF进行与运算，清掉十位，保留个位->5



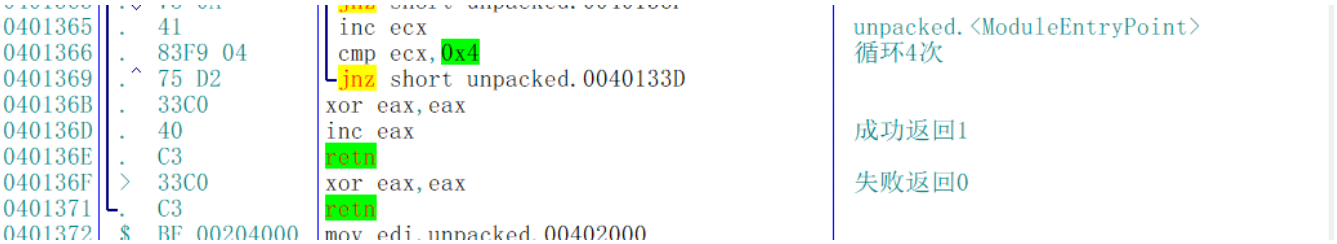
将保留位转成ASCII值->35



和输入的序列号对应的位进行比较



一次循环会比较两位，总共四次循环，一共比较8位，对应前面的序列号长度为8位



最后，根据比较的结果决定是否将eax置1

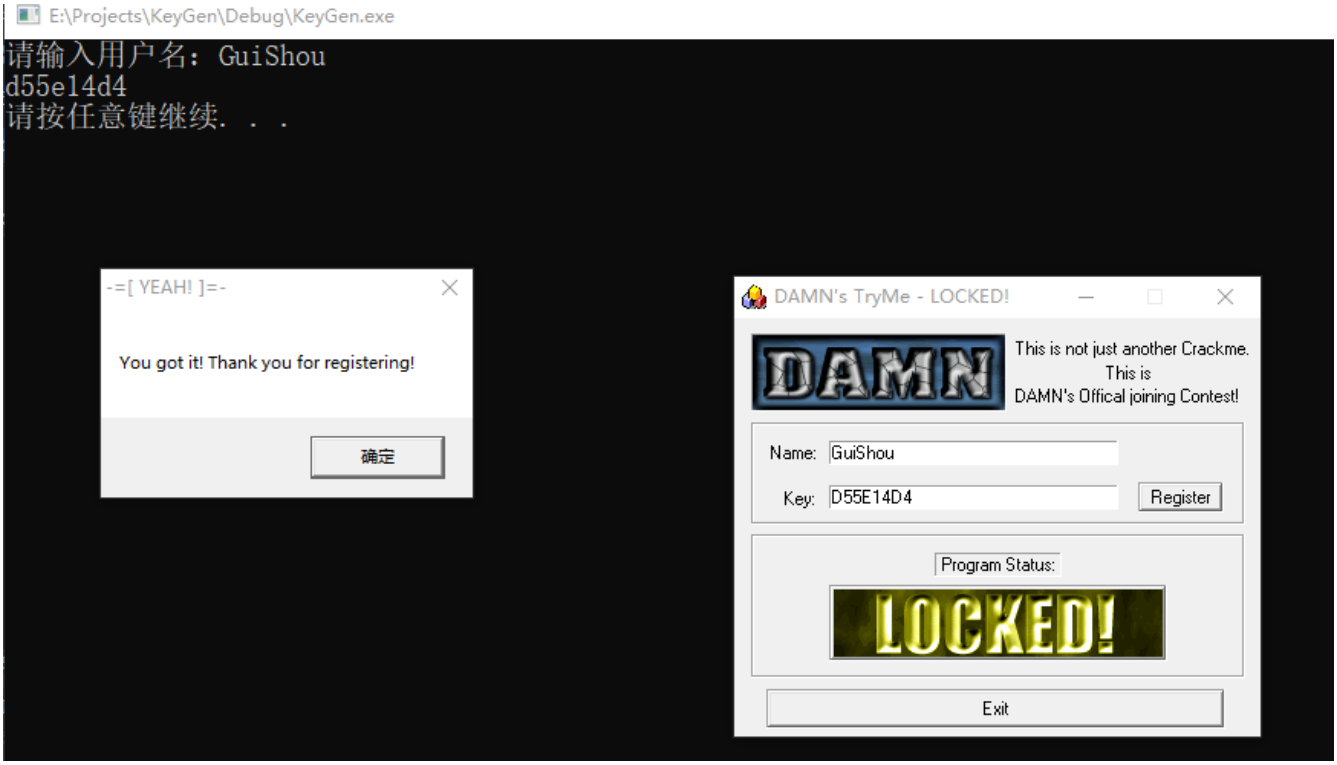
写出注册机

```
#include "pch.h"
#define _CRT_SECURE_NO_WARNINGS
#include <windows.h>
#include <stdio.h>
```

```
void ROR(DWORD * myd, unsigned char n)
{
    _asm
    {
        MOV EBX, myd;
        MOV eax, [EBX];
        MOV cl, n;
        ROR EAX, cl;
        MOV[EBX], eax;
    }
}


int main()
{
    char *username;
    unsigned long serial;
    serial = 0x44414D4E;
    unsigned long nTemp = 0;
    username = new char[260];
    memset(username, 0, 260);
    printf("请输入用户名: ");
    scanf_s("%s", username, 260);
    for (int i = 0; i < strlen(username); i++)
    {
        nTemp += username[i];
        ROR(&serial, 1);
        ROR(&serial, i);
        serial ^= nTemp;
    }

    serial |= 0x10101010;
    delete[] username;

    printf("%x\n", serial);
    system("pause");
    return 0;
}
```

校验结果

输入用户名和计算的序列号，提示注册成功



破解完成，最后需要相关文件可以到我的Github下载：

https://github.com/TonyChen56/160-Crackme