实验吧 该题不简单 OD详解 （flag 和分析内部算法）

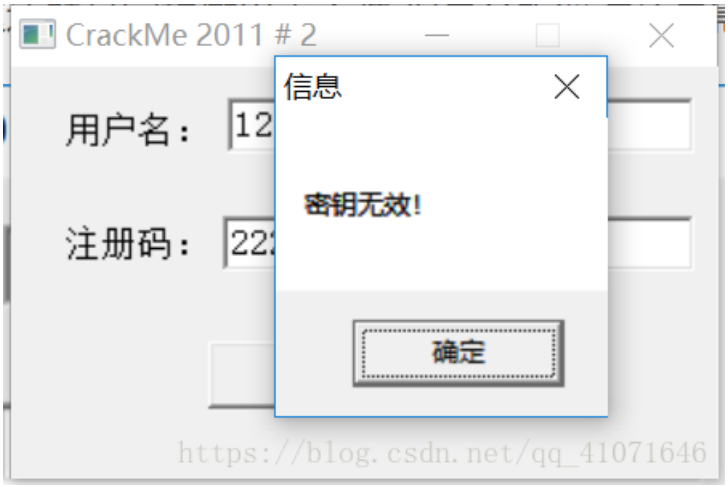
原创

pipixia233333 于 2018-08-17 20:53:54 发布 1313 收藏
分类专栏： 逆向之旅

这个题是偶然在实验吧想起来做的题 本来打算的是分析一晚上的 但是发现经过这一段时间的学习 自己进步很大 然后在半小时的时候就做出来了这一道题 然后 分享给大家

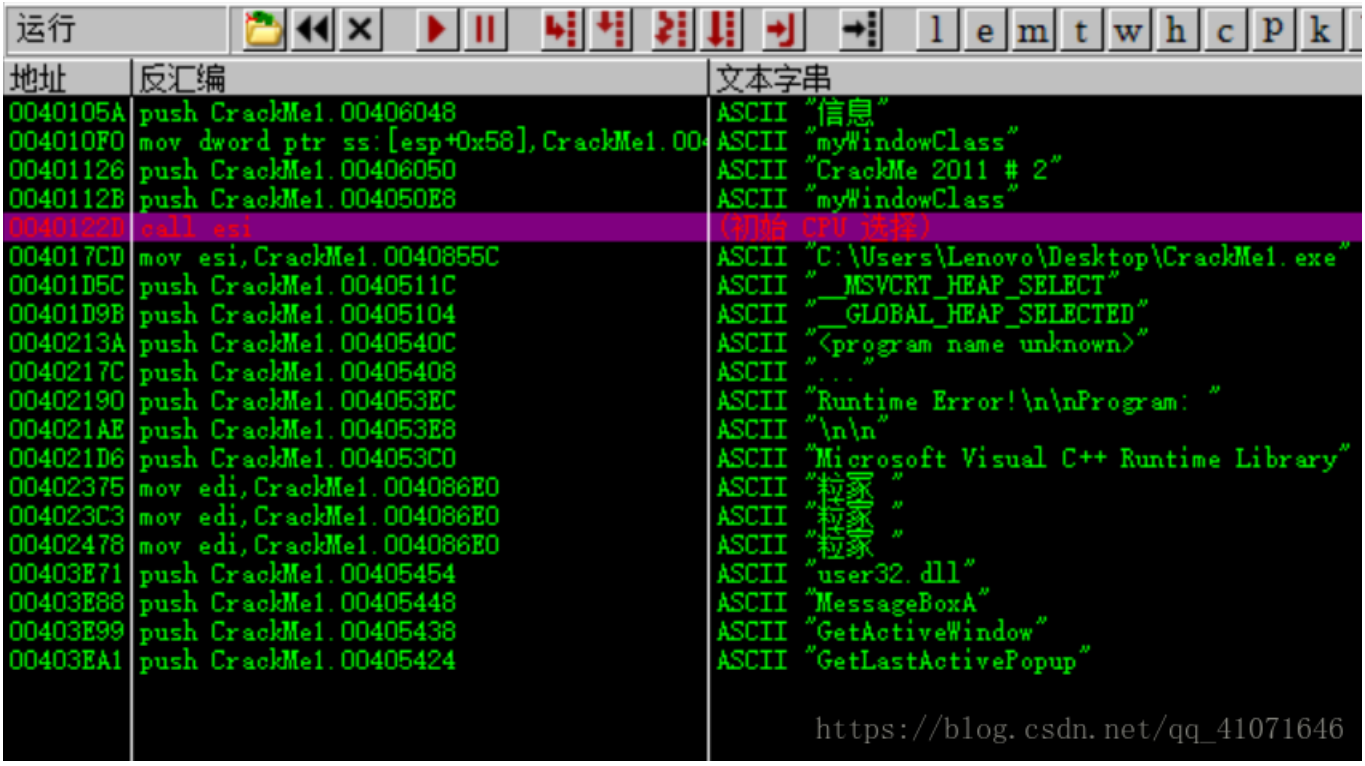
实验吧原题网址 这道题其实想做出来不难 但是分析算法还是需要一些基础的

点开就是这个样子 随便输入一下 就是这个样子

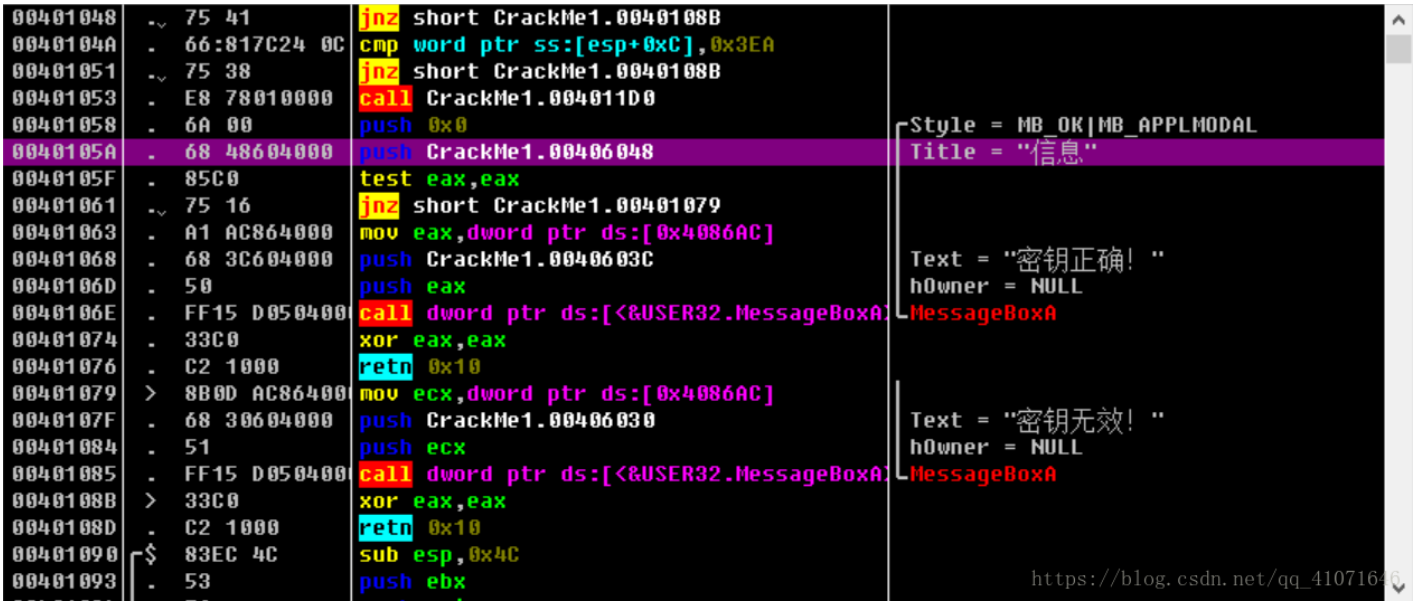


知道了程序大概是什么样子 然后开始分析算法 （题目要求的flag 是 用户名为 hello 的注册码）

然后打开OD 这里有几种方法断到我们想分析的地方 有一点就是 API GetDlgItemTextA 这个API 就是获取控件中的用户输入 然后我们先输入后查看 字符串



我们点开信息那一行



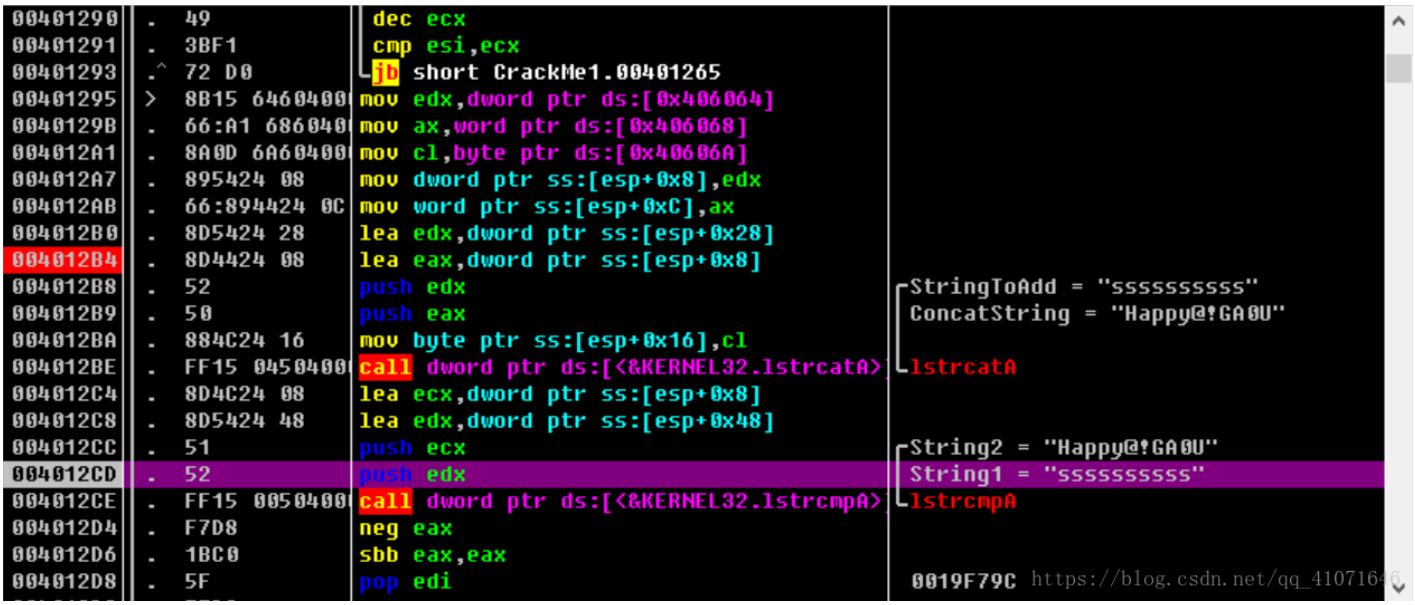
我们可以初步判定这是我们的响应信息后的代码 我找到那个函数 其实有点运气成分 因为我是往下看的时候正好看见了 这个函数

```
004011F1  .  8D7C24 49     lea edi,dword ptr ss:[esp+0x49]
004011F5  .  C64424 48 00  mov byte ptr ss:[esp+0x48],0x0
004011FA  .  F3:AB         rep stos dword ptr es:[edi]
004011FC  .  66:AB         stos word ptr es:[edi]
004011FE  .  AA            stos byte ptr es:[edi]
004011FF  .  B9 07000000   mov ecx,0x7
00401204  .  33C0          xor eax,eax
00401206  .  8D7C24 29     lea edi,dword ptr ss:[esp+0x29]
0040120A  .  C64424 28 00  mov byte ptr ss:[esp+0x28],0x0
0040120F  .  F3:AB         rep stos dword ptr es:[edi]
00401211  .  8B0D B086400  mov ecx,dword ptr ds:[0x4086B0]
00401217  .  8B35 A4504000 mov esi,dword ptr ds:[<&USER32.GetDlgIte  user32.GetDlgItemTextA
0040121D  .  66:AB         stos word ptr es:[edi]
0040121F  .  AA            stos byte ptr es:[edi]
00401220  .  8D4424 08     lea eax,dword ptr ss:[esp+0x8]
00401224  .  6A 10         push 0x10                                Count = 10 (16.)
00401226  .  50            push eax                                 Buffer = 000016C8
00401227  .  68 E8030000   push 0x3E8                               ControlID = 3E8 (1000.)
0040122C  .  51            push ecx                                 hWnd = 00000007
0040122D  .  FFD6          call esi                                 GetDlgItemTextA
0040122F  .  83F8 05       cmp eax,0x5
00401232  .v 73 0B         jnb short CrackMe1.0040123F
```

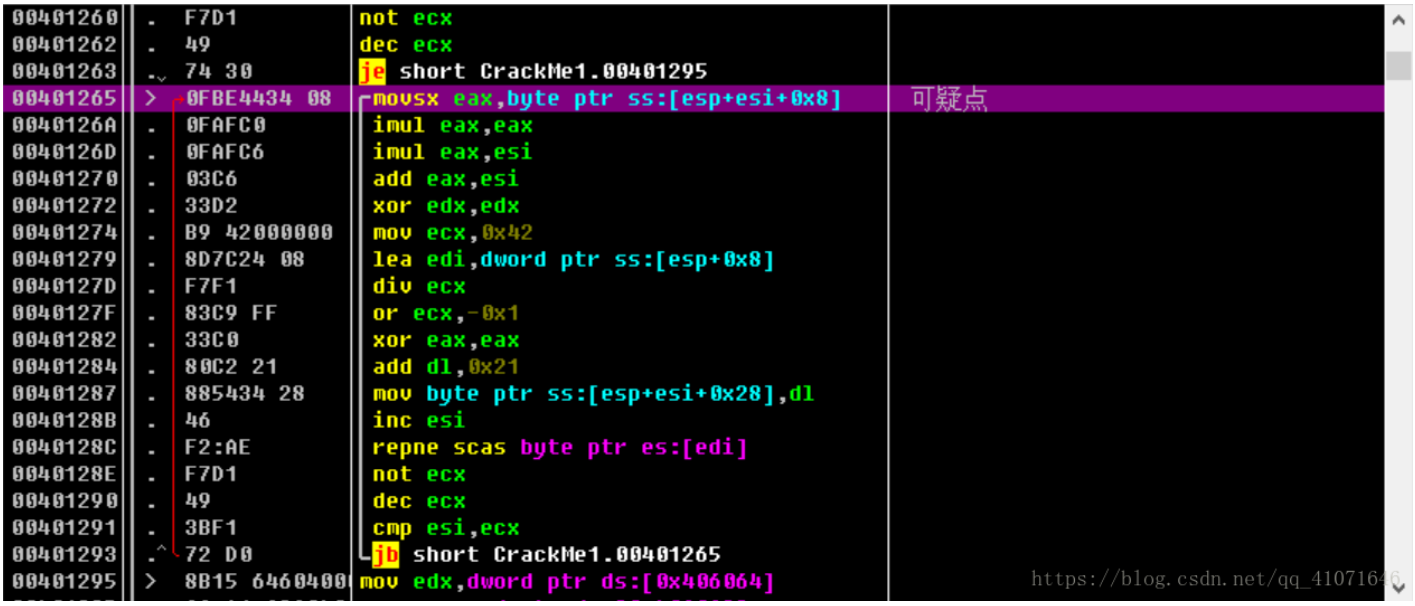
其实真正的做法应该是 用API断点 我用的吾爱破解的OD 有断点插件 插件就是第二个项 然后点下面的就可以了

然后我们断下来后 可以 为所欲为了（能断下来剩下来的分析就很好说了）

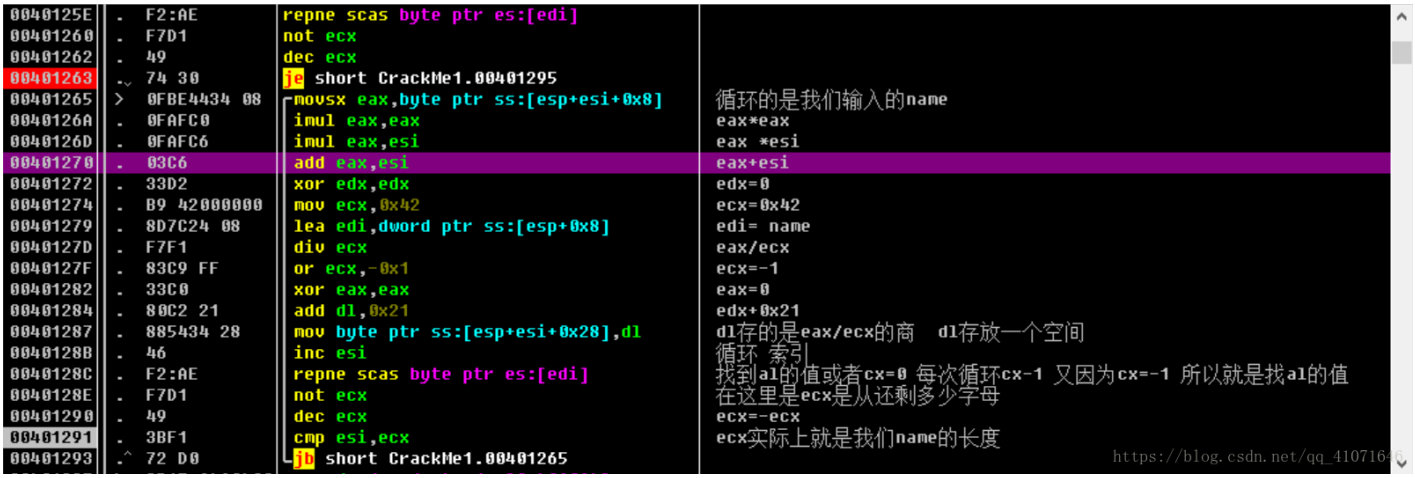
然后 我们先不管算法 随便输入试试 然后往下走看看



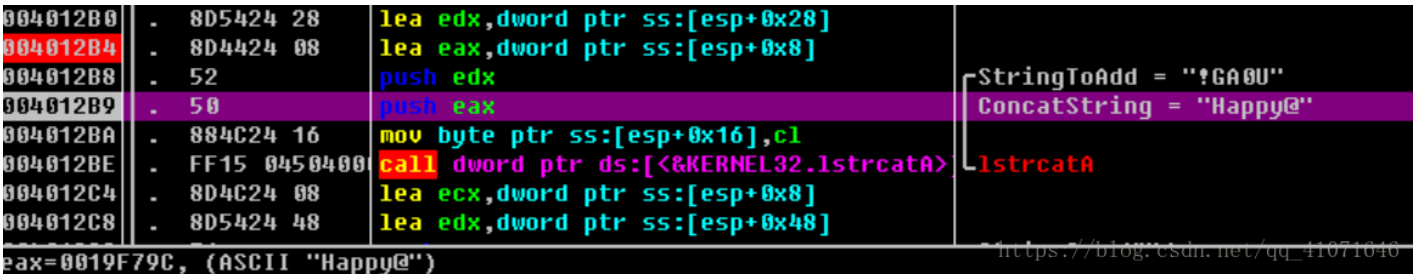
到了这一步应该有所警觉 是不是到了比较的部分 然后我们试着输入 看着flag对不对 然后输入了Happy@!GA0U 发现对了 我们可以知道我们地方找对了 然后就是算法分析部分了 （大家应该有一个好习惯 要保持一定的好奇心） 然后我们往上走 发现了



这里是可疑点 然后经过分析 大概每一步是这样的



其中的 循环 为什么 按位取反然后-1为啥等于相反数 这个可以自己慢慢体会 其实这就是esi是个索引值 ecx就是len（name）然后



这个是将与我们得到的合为一个字符串 那么 我们可以用python来表达这个过程

```
lists="Happy@"
name="hello"

for i in range(len(name)):
    lists+=chr((i+i*ord(name[i])*ord(name[i]))%0x42+0x21)

print(lists)
```

感觉这个题也不是很难