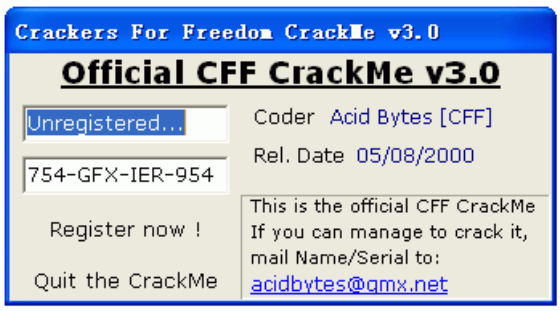
OllyDBG 入门系列（二）—字串参考

转载

子曰小玖 于 2020-10-29 10:42:06 发布 179 收藏 3
分类专栏： OllyDbg
原文链接：https://bbs.pediy.com/thread-21308.htm

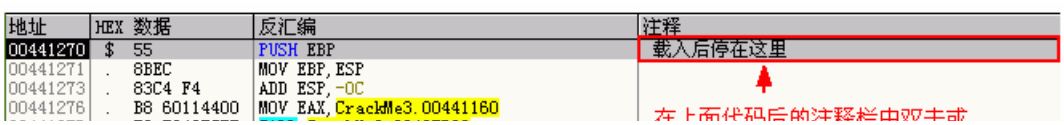
上一篇是使用入门，现在我们开始正式进入破解。今天的目标程序是看雪兄《加密与解密》第一版附带光盘中的 crackmes.cjb.net 镜像打包中的 CFF Crackme #3，采用用户名/序列号保护方式。原版加了个 UPX 的壳。刚开始学破解先不涉及壳的问题，我们主要是熟悉用 OllyDBG 来破解的一般方法。我这里把壳脱掉来分析，附件是脱壳后的文件，直接就可以拿来用。先说一下一般软件破解的流程：拿到一个软件先别接着马上用 OllyDBG 调试，先运行一下，有帮助文档的最好先看一下帮助，熟悉一下软件的使用方法，再看看注册的方式。如果是序列号方式可以先输个假的来试一下，看看有什么反应，也给我们破解留下一些有用的线索。如果没有输入注册码的地方，要考虑一下是不是读取注册表或 Key 文件（一般称 keyfile，就是程序读取一个文件中的内容来判断是否注册），这些可以用其它工具来辅助分析。如果这些都不是，原程序只是一个功能不全的试用版，那要注册为正式版本就要自己来写代码完善了。有点跑题了，呵呵。获得程序的一些基本信息后，还要用查壳的工具来查一下程序是否加了壳，若没壳的话看看程序是什么编译器编的，如 VC、Delphi、VB 等。这样的查壳工具有 PEiD 和 FI。有壳的话我们要尽量脱了壳后再来用 OllyDBG 调试，特殊情况下也可带壳调试。下面进入正题：

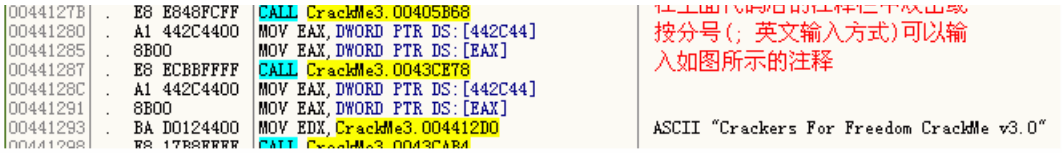
我们先来运行一下这个 crackme（用 PEiD 检测显示是 Delphi 编的），界面如图：



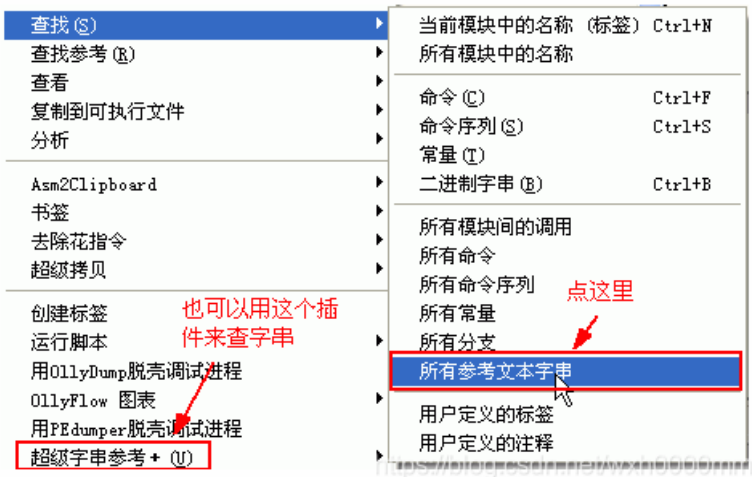
这个 crackme 已经把用户名和注册码都输好了，省得我们动手^_^。我们在那个“Register now !”按钮上点击一下，将会跳出一个对话框：

好了，今天我们就从这个错误对话框中显示的“Wrong Serial, try again!”来入手。启动 OllyDBG，选择菜单 文件->打开 载入 CrackMe3.exe 文件，我们会停在这里：

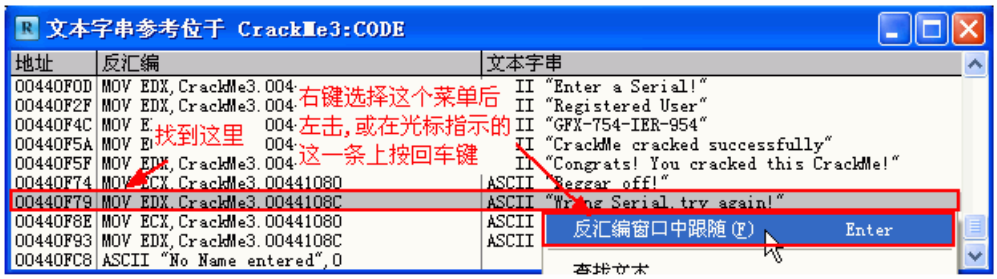




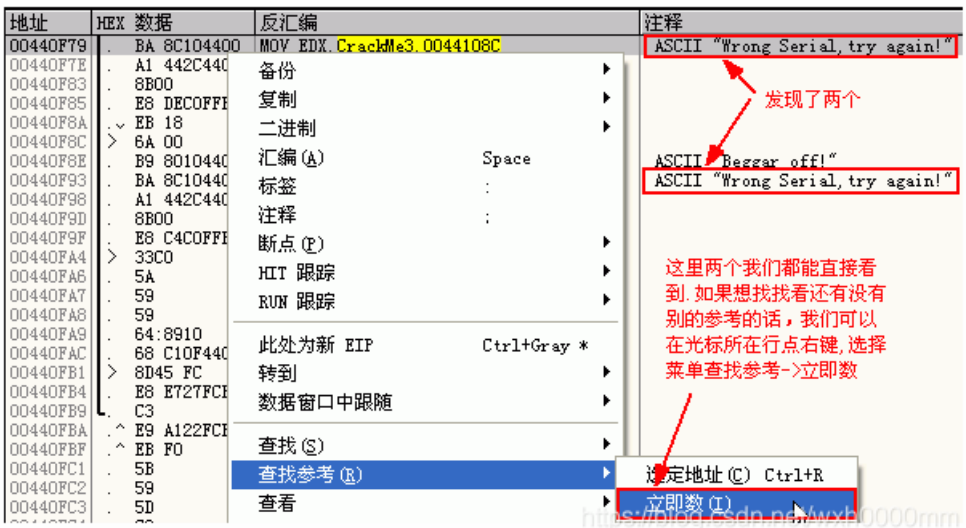
我们在反汇编窗口中右击，出来一个菜单，我们在 查找->所有参考文本字串 上左键点击：



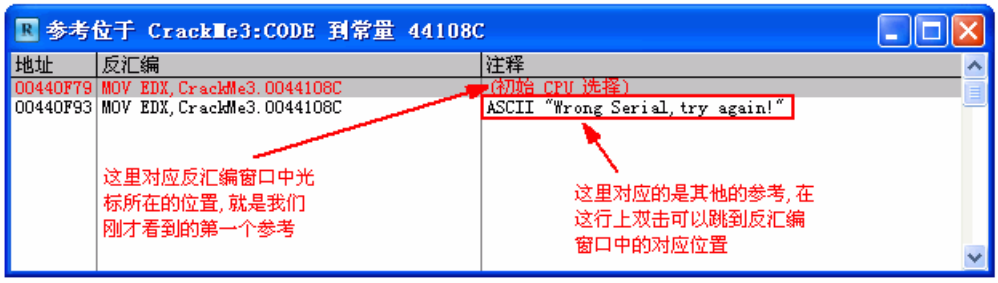
当然如果用上面那个 超级字串参考＋ 插件会更方便。但我们的目标是熟悉 OllyDBG 的一些操作，我就尽量使用 OllyDBG 自带的功能，少用插件。好了，现在出来另一个对话框，我们在这个对话框里右击，选择“查找文本”菜单项，输入“Wrong Serial, try again!”的开头单词“Wrong”（注意这里查找内容要区分大小写）来查找，找到一处：



在我们找到的字串上右击，再在出来的菜单上点击“反汇编窗口中跟随”，我们来到这里：



见上图，为了看看是否还有其他的参考，可以通过选择右键菜单查找参考->立即数，会出来一个对话框：



分别双击上面标出的两个地址，我们会来到对应的位置：

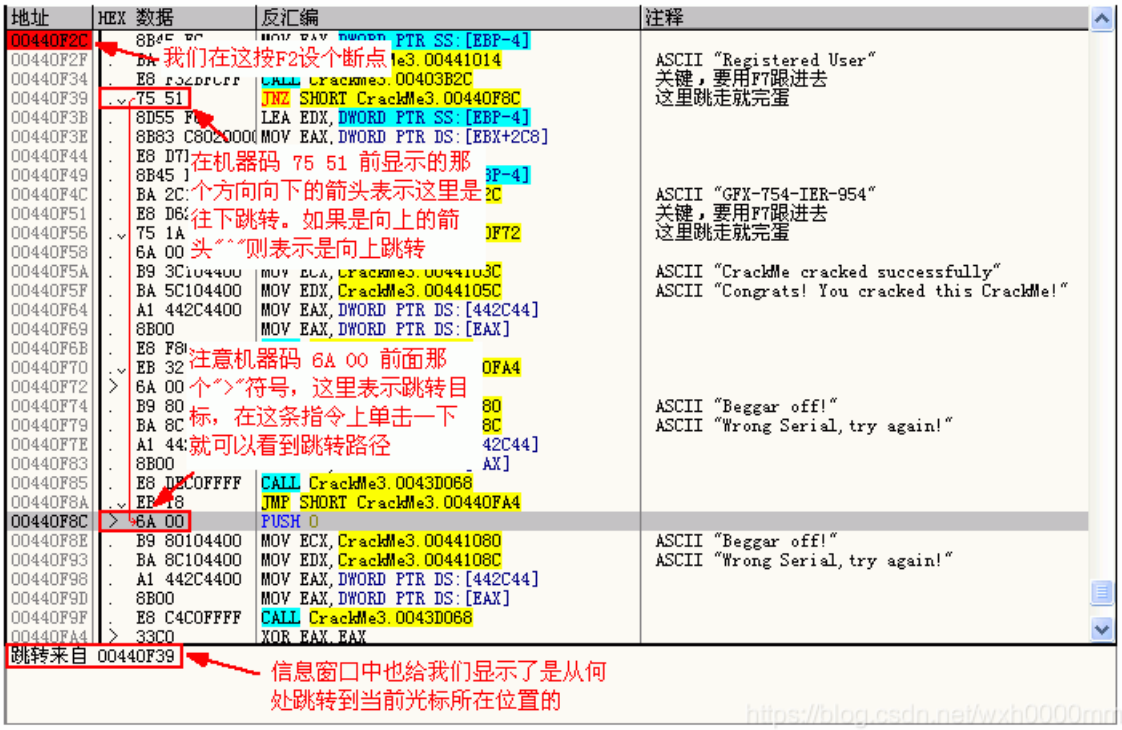
```
00440F79 |. BA 8C104400     MOV EDX,CrackMe3.0044108C            ; ASCII "Wrong Serial,try again!"
00440F7E |. A1 442C4400     MOV EAX,DWORD PTR DS:[442C44]
00440F83 |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
00440F85 |. E8 DEC0FFFF     CALL CrackMe3.0043D068
00440F8A |. EB 18           JMP SHORT CrackMe3.00440FA4
00440F8C |> 6A 00           PUSH 0
00440F8E |. B9 80104400     MOV ECX,CrackMe3.00441080            ; ASCII "Beggar off!"
00440F93 |. BA 8C104400     MOV EDX,CrackMe3.0044108C            ; ASCII "Wrong Serial,try again!"
00440F98 |. A1 442C4400     MOV EAX,DWORD PTR DS:[442C44]
00440F9D |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
00440F9F |. E8 C4C0FFFF     CALL CrackMe3.0043D068
```

我们在反汇编窗口中向上滚动一下再看看：

```
00440F2C |. 8B45 FC         MOV EAX,DWORD PTR SS:[EBP-4]
00440F2F |. BA 14104400     MOV EDX,CrackMe3.00441014            ; ASCII "Registered User"
00440F34 |. E8 F32BFCFF     CALL CrackMe3.00403B2C               ; 关键，要用F7跟进去
00440F39 |. 75 51           JNZ SHORT CrackMe3.00440F8C          ; 这里跳走就完蛋
00440F3B |. 8D55 FC         LEA EDX,DWORD PTR SS:[EBP-4]
00440F3E |. 8B83 C8020000   MOV EAX,DWORD PTR DS:[EBX+2C8]
00440F44 |. E8 D7FEFDFF     CALL CrackMe3.00420E20
00440F49 |. 8B45 FC         MOV EAX,DWORD PTR SS:[EBP-4]
00440F4C |. BA 2C104400     MOV EDX,CrackMe3.0044102C            ; ASCII "GFX-754-IER-954"
00440F51 |. E8 D62BFCFF     CALL CrackMe3.00403B2C               ; 关键，要用F7跟进去
00440F56 |. 75 1A           JNZ SHORT CrackMe3.00440F72          ; 这里跳走就完蛋
00440F58 |. 6A 00 PUSH 0
00440F5A |. B9 3C104400     MOV ECX,CrackMe3.0044103C            ; ASCII "CrackMe cracked successfully"
00440F5F |. BA 5C104400     MOV EDX,CrackMe3.0044105C            ; ASCII "Congrats! You cracked this
CrackMe!"
00440F64 |. A1 442C4400     MOV EAX,DWORD PTR DS:[442C44]
00440F69 |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
00440F6B |. E8 F8C0FFFF     CALL CrackMe3.0043D068
00440F70 |. EB 32           JMP SHORT CrackMe3.00440FA4
00440F72 |> 6A 00           PUSH 0
00440F74 |. B9 80104400     MOV ECX,CrackMe3.00441080            ; ASCII "Beggar off!"
00440F79 |. BA 8C104400     MOV EDX,CrackMe3.0044108C            ; ASCII "Wrong Serial,try again!"
00440F7E |. A1 442C4400     MOV EAX,DWORD PTR DS:[442C44]
00440F83 |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
```

```
00440F85 |. E8 DEC0FFFF     CALL CrackMe3.0043D068
00440F8A |. EB 18           JMP SHORT CrackMe3.00440FA4
00440F8C |> 6A 00           PUSH 0
00440F8E |. B9 80104400     MOV ECX,CrackMe3.00441080          ; ASCII "Beggar off!"
00440F93 |. BA 8C104400     MOV EDX,CrackMe3.0044108C          ; ASCII "Wrong Serial,try again!"
00440F98 |. A1 442C4400     MOV EAX,DWORD PTR DS:[442C44]
00440F9D |. 8B00           MOV EAX,DWORD PTR DS:[EAX]
00440F9F |. E8 C4C0FFFF     CALL CrackMe3.0043D068
```

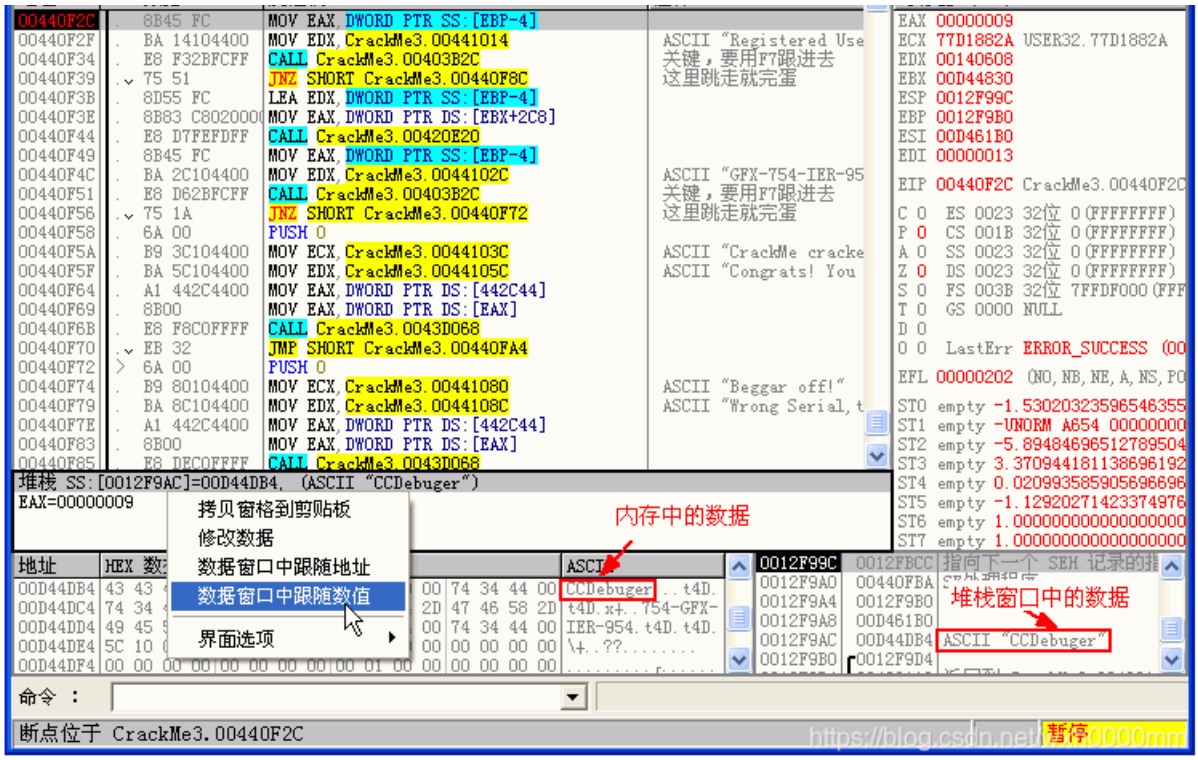
大家注意看一下上面的注释，我在上面标了两个关键点。有人可能要问，你怎么知道那两个地方是关键点？其实很简单，我是根据查看是哪条指令跳到“wrong serial,try again”这条字串对应的指令来决定的。如果你在 调试选项->CPU 标签中把“显示跳转路径”及其下面的两个“如跳转未实现则显示灰色路径”、“显示跳转到选定命令的路径”都选上的话，就会看到是从什么地方跳到出错字串处的：



我们在上图中地址 00440F2C 处按 F2 键设个断点，现在我们按 F9 键，程序已运行起来了。我在上面那个编辑框中随便输入一下，如 CCDebuger，下面那个编辑框我还保留为原来的“754-GFX-IER-954”，我们点一下那个“Register now !”按钮，呵，OllyDBG 跳了出来，暂停在我们下的断点处。我们看一下信息窗口，你应该发现了你刚才输入的内容了吧？我这里显示是这样：

堆栈 SS:[0012F9AC]=00D44DB4, (ASCII "CCDebuger")
EAX=00000009

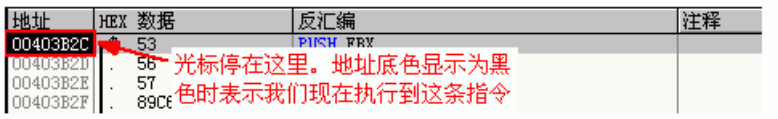
上面的内存地址 00D44DB4 中就是我们刚才输入的内容，我这里是 CCDebuger。你可以在 堆栈 SS:[0012F9AC]=00D44DB4, (ASCII "CCDebuger") 这条内容上左击选择一下，再点右键，在弹出菜单中选择“数据窗口中跟随数值”，你就会在下面的数据窗口中看到你刚才输入的内容。而 EAX=00000009 指的是你输入内容的长度。如我输入的 CCDebuger 是9个字符。如下图所示：



现在我们来按 F8 键一步步分析一下：

00440F2C |. 8B45 FC MOV EAX,DWORD PTR SS:[EBP-4] ；把我们输入的内容送到EAX，我这里是“CCDebuger”
00440F2F |. BA 14104400 MOV EDX,CrackMe3.00441014 ; ASCII "Registered User"
00440F34 |. E8 F32BFCFF CALL CrackMe3.00403B2C ；关键，要用F7跟进去
00440F39 |. 75 51 JNZ SHORT CrackMe3.00440F8C ；这里跳走就完蛋

当我们按 F8 键走到 00440F34 |. E8 F32BFCFF CALL CrackMe3.00403B2C 这一句时，我们按一下 F7 键，进入这个 CALL，进去后光标停在这一句：



我们所看到的那些 PUSH EBX、 PUSH ESI 等都是调用子程序保存堆栈时用的指令，不用管它，按 F8 键一步步过来，我们只关心关键部分：

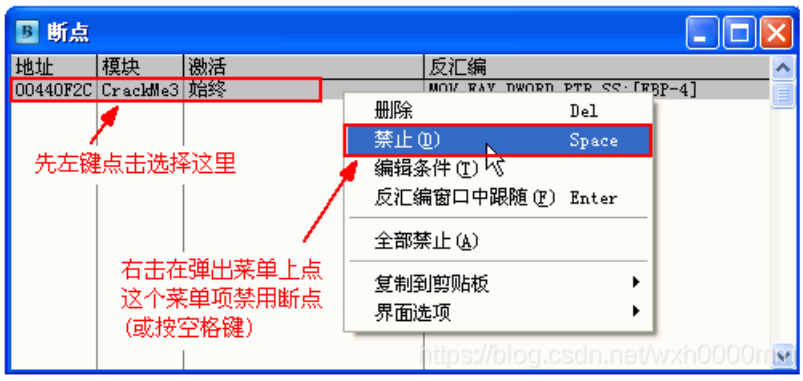
00403B2C /$ 53 PUSH EBX
00403B2D |. 56 PUSH ESI
00403B2E |. 57 PUSH EDI
00403B2F |. 89C6 MOV ESI,EAX ；把EAX内我们输入的用户名送到 ESI
00403B31 |. 89D7 MOV EDI,EDX ；把EDX内的数据“Registered User”送到EDI
00403B33 |. 39D0 CMP EAX,EDX ；用“Registered User”和我们输入的用户名作比较
00403B35 |. 0F84 8F000000 JE CrackMe3.00403BCA ；相同则跳
00403B3B |. 85F6 TEST ESI,ESI ；看看ESI中是否有数据，主要是看看我们有没有输入用户名
00403B3D |. 74 68 JE SHORT CrackMe3.00403BA7 ；用户名为空则跳
00403B3F |. 85FF TEST EDI,EDI
00403B41 |. 74 6B JE SHORT CrackMe3.00403BAE

```
00403B43 |. 8B46 FC        MOV EAX,DWORD PTR DS:[ESI-4]       ; 用户名长度送EAX
00403B46 |. 8B57 FC        MOV EDX,DWORD PTR DS:[EDI-4]       ; “Registered User”字串的长度送EDX
00403B49 |. 29D0          SUB EAX,EDX                        ; 把用户名长度和“Registered User”字串长度相减
00403B4B |. 77 02          JA SHORT CrackMe3.00403B4F         ; 用户名长度大于“Registered User”长度则跳
00403B4D |. 01C2          ADD EDX,EAX                        ; 把减后值与“Registered User”长度相加，即用户名长
度
00403B4F |> 52            PUSH EDX
00403B50 |. C1EA 02       SHR EDX,2                          ; 用户名长度值右移2位，这里相当于长度除以4
00403B53 |. 74 26          JE SHORT CrackMe3.00403B7B         ; 上面的指令及这条指令就是判断用户名长度
最少不能低于4
00403B55 |> 8B0E          MOV ECX,DWORD PTR DS:[ESI]         ; 把我们输入的用户名送到ECX
00403B57 |. 8B1F          MOV EBX,DWORD PTR DS:[EDI]         ; 把“Registered User”送到EBX
00403B59 |. 39D9          CMP ECX,EBX                        ; 比较
00403B5B |. 75 58          JNZ SHORT CrackMe3.00403BB5        ; 不等则完蛋
```

根据上面的分析，我们知道用户名必须是“Registered User”。我们按 F9 键让程序运行，出现错误对话框，点确定，重新在第一个编辑框中输入“Registered User”，再次点击那个“Register now !”按钮，被 OllyDBG 拦下。因为地址 00440F34 处的那个 CALL 我们已经分析清楚了，这次就不用再按 F7 键跟进去了，直接按 F8 键通过。我们一路按 F8 键，来到第二个关键代码处：

```
00440F49 |. 8B45 FC        MOV EAX,DWORD PTR SS:[EBP-4]       ; 取输入的注册码
00440F4C |. BA 2C104400     MOV EDX,CrackMe3.0044102C          ; ASCII "GFX-754-IER-954"
00440F51 |. E8 D62BFCFF     CALL CrackMe3.00403B2C             ; 关键，要用F7跟进去
00440F56 |. 75 1A          JNZ SHORT CrackMe3.00440F72        ; 这里跳走就完蛋
```

大家注意看一下，地址 00440F51 处的 CALL CrackMe3.00403B2C 和上面我们分析的地址 00440F34 处的 CALL CrackMe3.00403B2C 是不是汇编指令都一样啊？这说明检测用户名和注册码是用的同一个子程序。而这个子程序 CALL 我们在上面已经分析过了。我们执行到现在可以很容易得出结论，这个 CALL 也就是把我们输入的注册码与 00440F4C 地址处指令后的“GFX-754-IER-954”作比较，相等则 OK。好了，我们已经得到足够的信息了。现在我们在菜单 查看->断点 上点击一下，打开断点窗口（也可以通过组合键 ALT+B 或点击工具栏上那个“B”图标打开断点窗口）：



为什么要做这一步，而不是把这个断点删除呢？这里主要是为了保险一点，万一分析错误，我们还要接着分析，要是把断点删除了就要做一些重复工作了。还是先禁用一下，如果经过实际验证证明我们的分析是正确的，再删不迟。现在我们把断点禁用，在 OllyDBG 中按 F9 键让程序运行。输入我们经分析得出的内容：
用户名：Registered User
注册码：GFX-754-IER-954
点击“Register now !”按钮，呵呵，终于成功了：


Official CFF CrackMe v3.0
Registered User
Coder Acid Bytes [CFF]
CrackMe cracked successfully
Congrats! You cracked this CrackMe!
确定