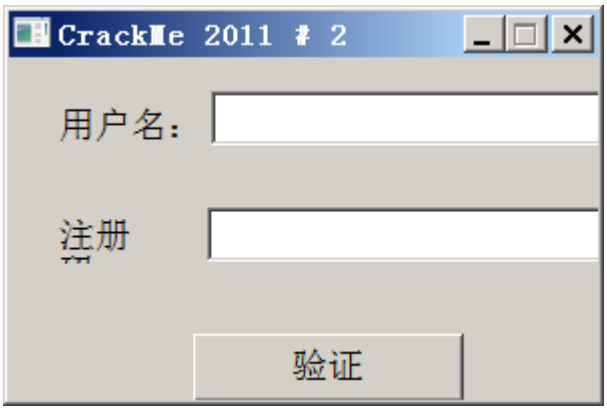
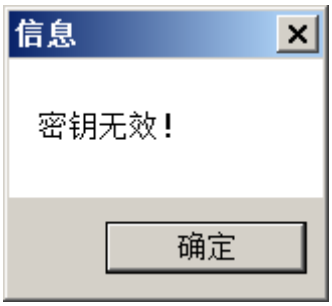
该题不简单 WriteUp

题目地址：http://www.shiyanbar.com/ctf/14
已知用户名为 hello，寻找注册码。





首先用 IDA 打开该文件，寻找文字附近的信息。

```
.text:00401079                          ; ---------------------------------------------------------------------------
.text:00401079
.text:00401079                          loc_401079:                             ; CODE XREF: DialogFunc+21↑j
.text:00401079 8B 0D AC 86 40 00                        mov     ecx, hWnd
.text:0040107F 68 30 60 40 00                           push    offset aZgb     ; "密钥无效！"
.text:00401084 51                                       push    ecx             ; hWnd
.text:00401085 FF 15 D0 50 40 00                        call    ds:MessageBoxA
.text:0040108B
.text:0040108B                          loc_40108B:                             ; CODE XREF: DialogFunc+8↑j
.text:0040108B                                                                  ; DialogFunc+11↑j
.text:0040108B 33 C0                                    xor     eax, eax
.text:0040108D C2 10 00                                 retn    10h
.text:0040108D                          DialogFunc      endp
```

然后利用 loc_401079 寻找上一个被引用的位置（类似于 j loc_401079 之类的），实际上就在上方：

```
.text:00401040                          ; ============== S U B R O U T I N E =======================================
.text:00401040
.text:00401040
.text:00401040                          ; BOOL __stdcall DialogFunc(HWND, UINT, WPARAM, LPARAM)
.text:00401040                          DialogFunc      proc near               ; DATA XREF: WinMain(x,x,x,x)+A8↓o
.text:00401040
.text:00401040                          arg_4           = dword ptr  8
.text:00401040                          arg_8           = word ptr  0Ch
.text:00401040
.text:00401040 81 7C 24 08 11 01 00 00                  cmp     [esp+arg_4], 111h
.text:00401048 75 41                                    jnz     short loc_40108B
.text:0040104A 66 81 7C 24 0C EA 03                     cmp     [esp+arg_8], 3EAh
.text:00401051 75 38                                    jnz     short loc_40108B
.text:00401053 E8 78 01 00 00                           call    sub_4011D0
.text:00401058 6A 00                                    push    0               ; uType
.text:0040105A 68 48 60 40 00                           push    offset Caption  ; "信息"
.text:0040105F 85 C0                                    test    eax, eax
.text:00401061 75 16                                    jnz     short loc_401079
.text:00401063 A1 AC 86 40 00                           mov     eax, hWnd
.text:00401068 68 3C 60 40 00                           push    offset Text     ; "密钥正确！"
.text:0040106D 50                                       push    eax             ; hWnd
.text:0040106E FF 15 D0 50 40 00                        call    ds:MessageBoxA
.text:00401074 33 C0                                    xor     eax, eax
.text:00401076 C2 10 00                                 retn    10h
```

这样一来 DialogFunc 应该是程序的入口了。查看反编译出来的伪 C 码：

```
int __stdcall DialogFunc(int a1, int a2, __int16 a3, int a4)
{
  if ( a2 == 273 && a3 == 1002 )
  {
    if ( !sub_4011D0() )
    {
      MessageBoxA(hWnd, "密钥正确！", "信息", 0);
      return 0;
    }
    MessageBoxA(hWnd, "密钥无效！", "信息", 0);
  }
  return 0;
}
```

可见其中的判断关键方法为 sub_4011d0。寻找到这个方法：

```
signed int __cdecl sub_4011D0()
{
  signed int result; // eax@2
  unsigned int v1; // esi@3
  CHAR String1[4]; // [sp+8h] [bp-60h]@1
  __int16 v3; // [sp+25h] [bp-43h]@1
  char v4; // [sp+27h] [bp-41h]@1
  const CHAR String2; // [sp+28h] [bp-40h]@1
  char v6; // [sp+29h] [bp-3Fh]@1
  __int16 v7; // [sp+45h] [bp-23h]@1
  char v8; // [sp+47h] [bp-21h]@1
  char v9; // [sp+48h] [bp-20h]@1
  char v10; // [sp+49h] [bp-1Fh]@1
  __int16 v11; // [sp+65h] [bp-3h]@1
  char v12; // [sp+67h] [bp-1h]@1

  String1[0] = 0;
  memset(&String1[1], 0, 0x1Cu);
  v3 = 0;
  v4 = 0;
  v9 = 0;
  memset(&v10, 0, 0x1Cu);
  v11 = 0;
  v12 = 0;
  String2 = 0;
  memset(&v6, 0, 0x1Cu);
  v7 = 0;
```

```
  v8 = 0;
  if ( GetDlgItemTextA(dword_4086B0, 1000, String1, 16) >= 5 )
  {
    GetDlgItemTextA(dword_4086B0, 1001, &v9, 16);
    v1 = 0;
    if ( strlen(String1) != 0 )
    {
      do
      {
        *(&String2 + v1) = (v1 + v1 * String1[v1] * String1[v1]) % 0x42 + 33;
        ++v1;
      }
      while ( v1 < strlen(String1) );
    }
    strcpy(String1, "Happy@");
    lstrcatA(String1, &String2);
    result = lstrcmpA(&v9, String1) != 0;
  }
  else
  {
    result = 1;
  }
  return result;
}
```

那么已知的情况为：用户名 hello，此处应为 String1.再看代码的关键部分 if 块，String2 是对 String1，即字符串”hello”进行少量变换获得。将其编成代码块：

```
int v = 0;
String s = "hello";
do{
    System.out.print((char)((v + v * (int)s.charAt(v) * (int)s.charAt(v)) % 0x42 + 33) );
    v += 1;
}while(v < s.length());
```

执行获得一个五位字符串结果（此处结果隐藏）。
然后继续往下：将 Happy@拷给 String1，再将 String2 附在 String1 后面，与 v9 所存储的字符串进行比较，如果结果不同，result 为 0，否则为 1，然后返回 result 给先前的 DialogFunc，答案已经揭晓。