看雪加密与解密第四版随书chap03练习题解ReverseMes by SantMat #1

原创

jovy-rtt 于 2020-04-23 21:58:15 发布 441 收藏 1

分类专栏： 看雪加密与解密第四版随书程序题解 CTF 文章标签： 逆向

本文链接：https://blog.csdn.net/zmx2473162621/article/details/105715332

看雪加密与解密第四版随书程序题解 同时被 2 个专栏收录

1 篇文章 2 订阅

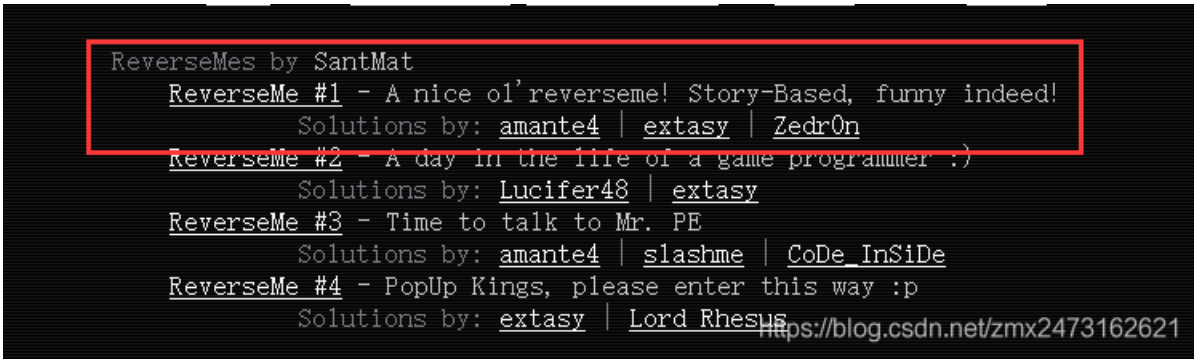
订阅专栏

CTF

28 篇文章 3 订阅

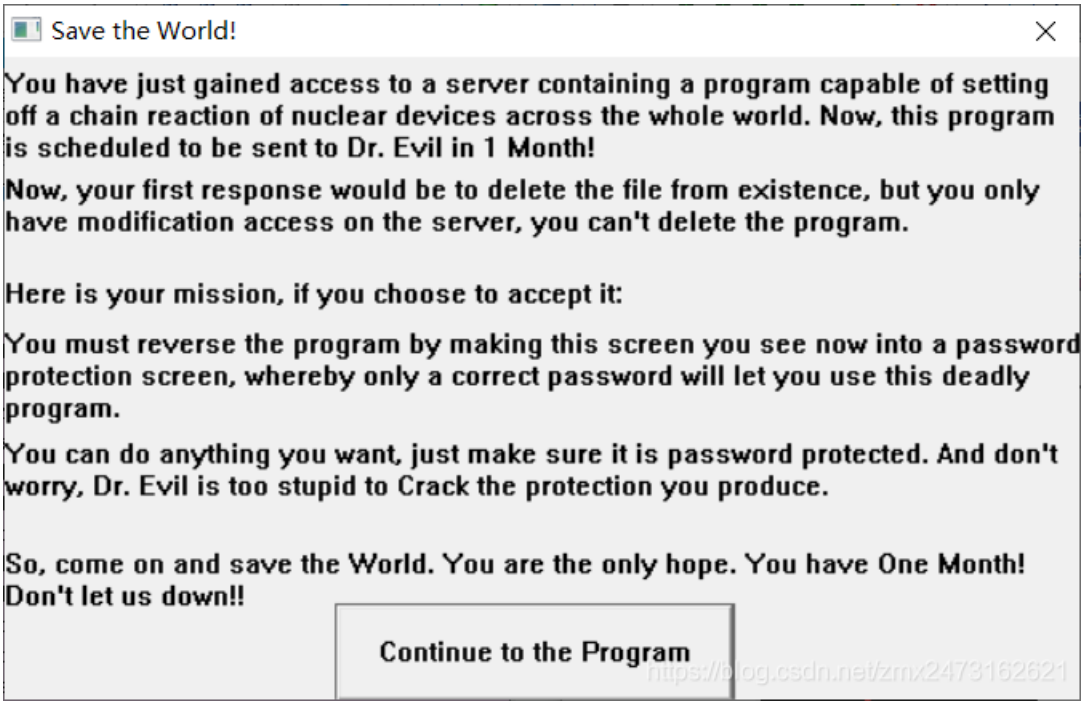
订阅专栏

题目



FIRST

运行一下程序，看一下，这个程序主要的功能，以及我们需要做什么：



我们可以注意到

标题：save the world
内容：分段的 you have …
一个按钮

内容翻译：
您刚刚可以访问包含其中一个程序的服务器，该程序能够在全世界范围内引发核设备的连锁反应。现在，该程序计划在1个月内发送给Evil博士！
现在，您的第一个响应将是删除存在的文件，但是您只有服务器上的修改访问权限，无法删除程序。
如果您选择接受，这是您的任务：
您必须通过使现在看到的该屏幕进入密码保护屏幕来使程序反向，从而只有正确的密码才能使用该致命程序。
您可以做任何您想做的事情，只要确保它受密码保护即可。不用担心，Evil博士太愚蠢而无法破解您提供的保护。
因此，快来拯救世界。您是唯一的希望。你有一个月！不要让我们失望！

然后我们点击继续按钮：

发现两个按钮，一个点击之后会boooom！爆炸，
另一个是退出这个程序的按钮

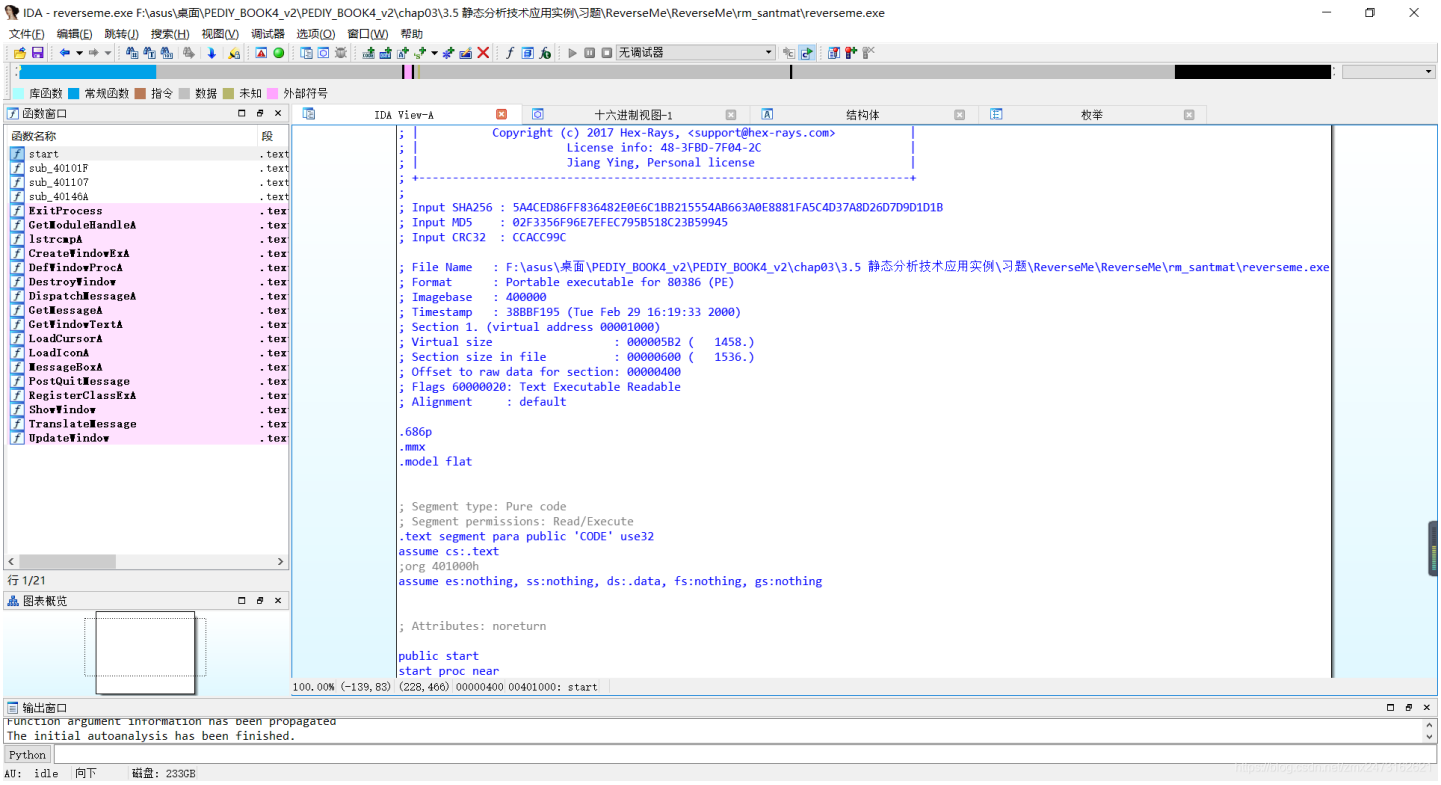
所以我们的目的是

阻止evil进入这个程序，通过修改第一个界面，加上一个输入密码验证

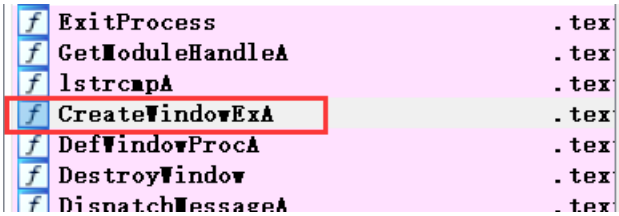
Second

tools：
ida pro
od

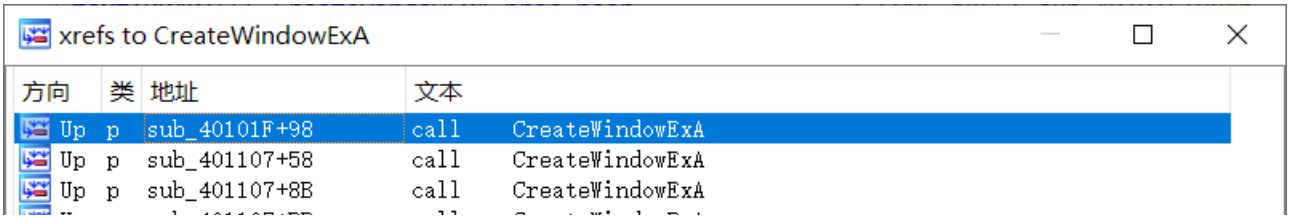
拉入ida中

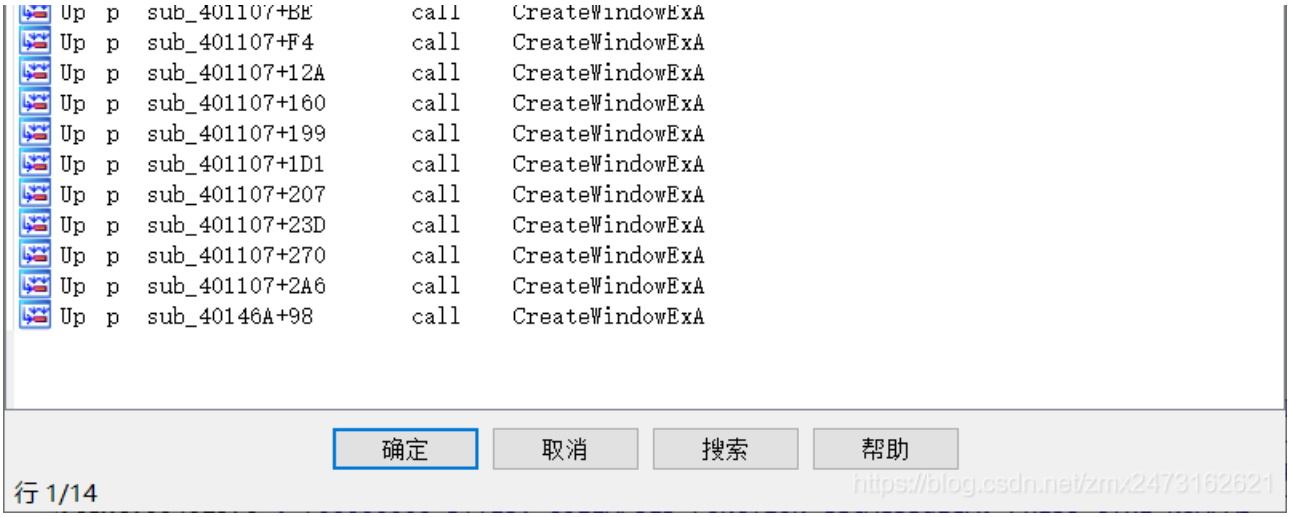


第一个页面是这样的，首先，我们要思考，这是个什么程序，这好像不是用说用codeblock什么的c++写出来的，那么，我们看一下函数窗口，发现都是createwindowexa这些函数，调用的是底层函数api，那么就要去想，我们要做的，和要用的都有哪些
首先，想要定位到第一个窗口对于的信息，所以，我们在后边函数窗口找到(第二种方法，直接打开视图-打开子视图-字符串窗口，找都save the world，进去，然后找这个地方的调用)
CreateWindowExA函数

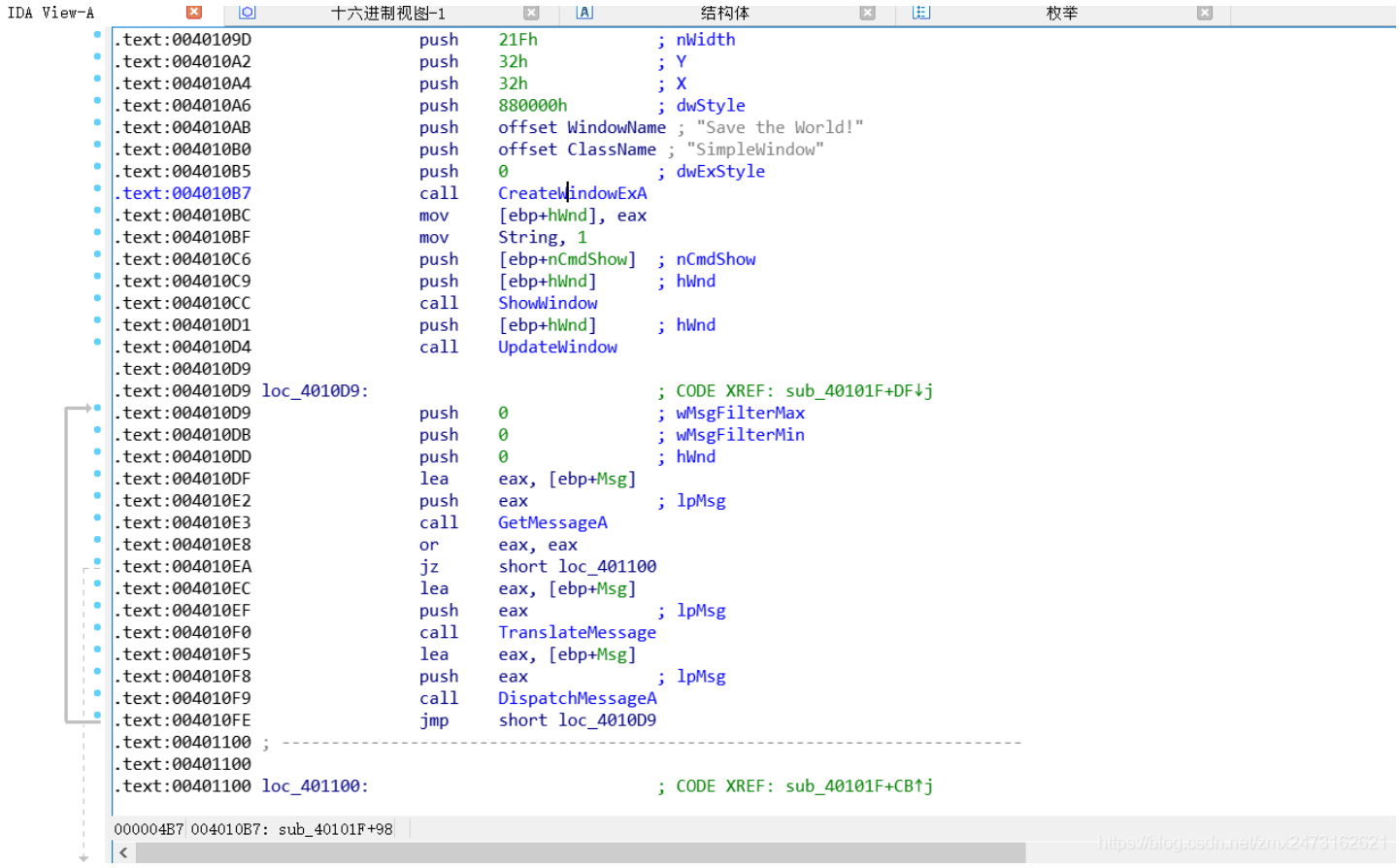


双击进去，
.text:0040155E CreateWindowExA proc near
在这个地方按x弹出交叉参考





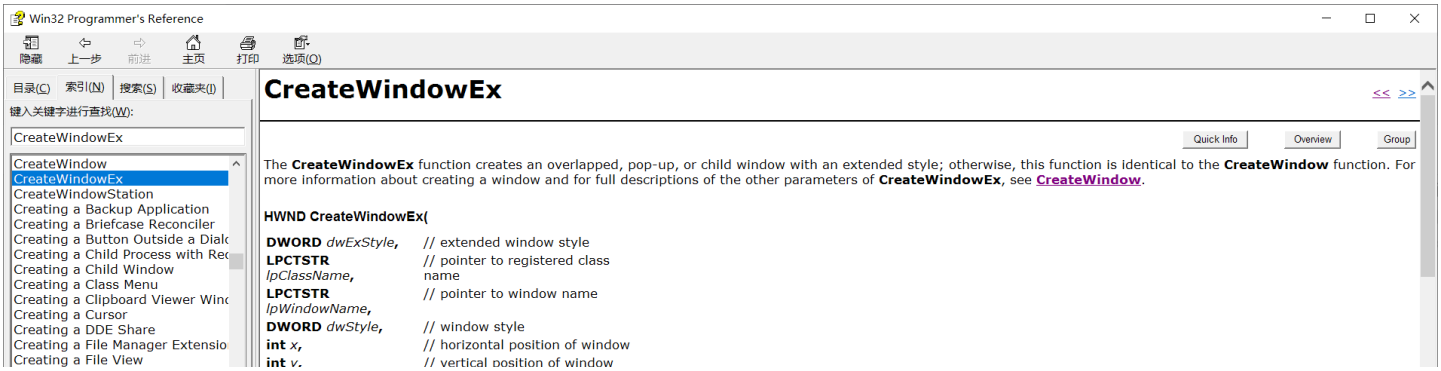
找到了都有哪些地方调用了这个函数，进去第一个

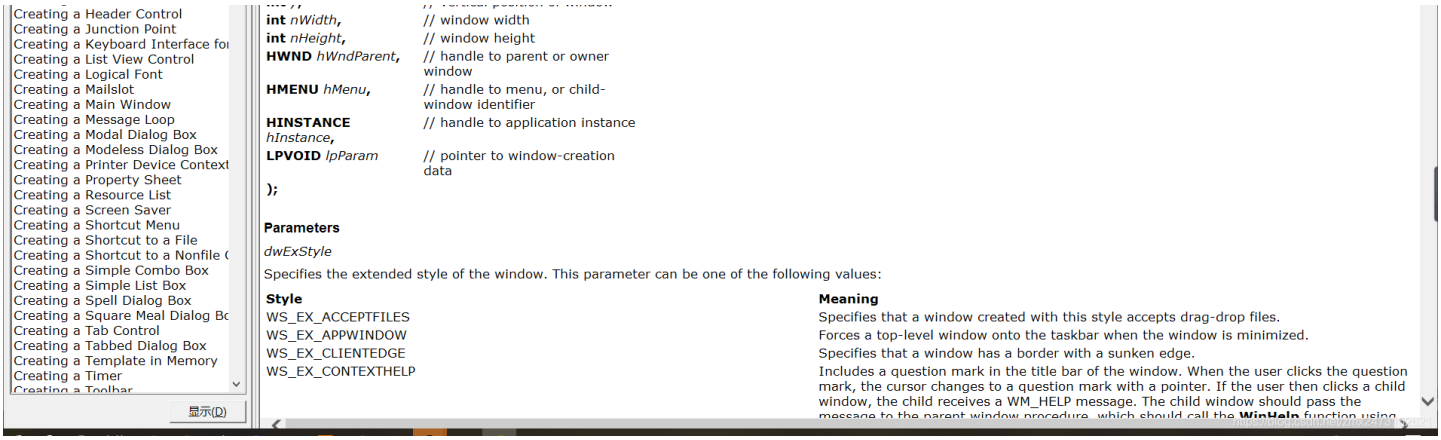


发现了save the world这个关键地方，这不就是我们进去的第一个标题嘛

所以确定了这部分代码就是第一个窗口要创建的地方

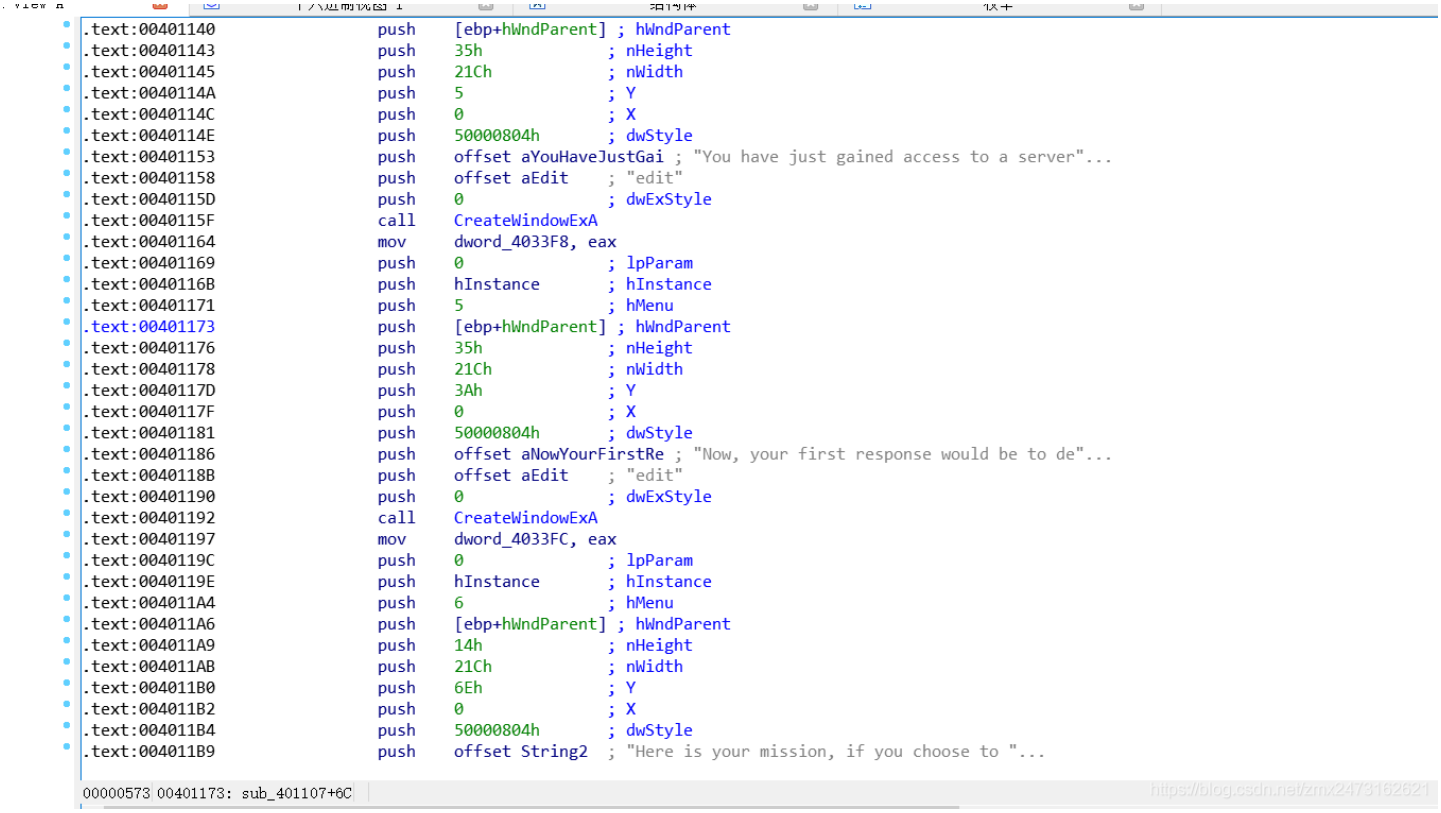
当然有很多备注我们看不懂，没关系，可以看一下win32api的帮助手册





就会发现，备注一般就是这个函数参数

然后我们确定了这一步就是创建，那么我们就往下继续翻



突然发现这个地方，备注很熟悉，这不就是界面上显示的地方嘛
那么我们怎么输入密码呢？
可以编辑box，就可以，但是发现这些创建的窗口exa都可以edit，看备注，都可以edit
那么我们用哪一个作为我们的输入密码的地方呢？
思考一下，咦，输入之后要get，所以我们用函数窗口的get，然后通过调用定位，或者直接f5反汇编

```
                            200,
                            50,
                            hWndParent,
                            (HMENU)1,
                            hInstance,
                            0);
    }
    break;
  case 0x111u:
    v4 = wParam;
    if ( (_WORD)wParam == 1 )
    {
      v4 = wParam >> 16;
      if ( !HIWORD(wParam) )
      {
        GetWindowTextA(hWnd, &String, 512);
        lstrcmpA(&String, String2);
        DestroyWindow(hWndParent);
        v4 = sub_40146A(hInstance, 0, 0, 10);
      }
    }
    if ( (_WORD)v4 == 2 )
    {
      v4 >>= 16;
      if ( !(_WORD)v4 )
        v4 = MessageBoxA(0, Text, Caption, 0);
    }
    if ( (_WORD)v4 == 3 && !HIWORD(v4) )
      ExitProcess(0);
    break;
  default:
    return DefWindowProcA(hWndParent, Msg, wParam, lParam);
 }
 return 0;
}
```

找到get，光标方上，然后找到汇编代码哪里
.text:004013D2 shr eax, 10h
.text:004013D5 or ax, ax
.text:004013D8 jnz short loc_401417
.text:004013DA push 200h ; nMaxCount
.text:004013DF push offset String ; lpString
.text:004013E4 push hWnd ; hWnd
.text:004013EA call GetWindowTextA
.text:004013EF push offset String2 ; "Here is your mission, if you choose to "…
.text:004013F4 push offset String ; lpString1
.text:004013F9 call lstrcmpA
.text:004013FE push [ebp+hWndParent] ; hWnd
.text:00401401 call DestroyWindow
.text:00401406 push 0Ah ; nCmdShow
.text:00401408 push 0 ; int
.text:0040140A push 0 ; int
.text:0040140C push hInstance ; hInstance
.text:00401412 call sub_40146A

然后这里有lstrcmpA，那就证明确实是这个地方，我们找对了，所以我们的目的就是把 "Here is your mission, if you choose to "…这一块变成可输入模式，然后给一个密码，验证，后面的验证部分我们不用操心，因为已经有了strcmp了，我们还要搞输入，然后付一个密码，就好！
回到刚才的creat部分，esc退步操作

```
.text:00401176                push    35h             ; nHeight
.text:00401178                push    21Ch            ; nWidth
.text:0040117D                push    3Ah             ; Y
.text:0040117F                push    0               ; X
.text:00401181                push    50000804h       ; dwStyle
```

```
.text:00401186                push    offset aNowYourFirstRe ; "Now, your first response would be to de"...
.text:0040118B                push    offset aEdit    ; "edit"
.text:00401190                push    0               ; dwExStyle
.text:00401192                call    CreateWindowExA
.text:00401197                mov     dword_4033FC, eax
.text:0040119C                push    0               ; lpParam
.text:0040119E                push    hInstance       ; hInstance
.text:004011A4                push    6               ; hMenu
.text:004011A6                push    [ebp+hWndParent] ; hWndParent
.text:004011A9                push    14h             ; nHeight
.text:004011AB                push    21Ch            ; nWidth
.text:004011B0                push    6Eh             ; Y
.text:004011B2                push    0               ; X
.text:004011B4                push    50000804h       ; dwStyle
.text:004011B9                push    offset String2  ; "Here is your mission, if you choose to "...
.text:004011BE                push    offset aEdit    ; "edit"
.text:004011C3                push    0               ; dwExStyle
.text:004011C5                call    CreateWindowExA
.text:004011CA                mov     hWnd, eax
.text:004011CF                push    0               ; lpParam
.text:004011D1                push    hInstance       ; hInstance
.text:004011D7                push    7               ; hMenu
.text:004011D9                push    [ebp+hWndParent] ; hWndParent
.text:004011DC                push    35h             ; nHeight
00000592 00401192: sub_401107+8B
```


这个地方，Here is your mission, if you choose to 我们要搞成输入模式，查一下createwindowexa（注意大小写，我这里随意了）的参数

把dwstyle由50000804h改为：50800000h 改的时候可以用od吧，我用ida的时候用不得不熟…

od改的时候按照地址就好了

ok

然后

输入的offset string2就没用了，我们把它变成push 0

其他的提示地方同理push 0

然后开始考虑，验证密码

```
.text:004013E4                push    hWnd            ; hWnd
.text:004013EA                call    GetWindowTextA
.text:004013EF                push    offset String2  ; "Here is your mission, if you choose to "...
.text:004013F4                push    offset String   ; lpString1
.text:004013F9                call    lstrcmpA
.text:004013FE                push    [ebp+hWndParent] ; hWnd
.text:00401401                call    DestroyWindow
.text:00401406                push    0Ah             ; nCmdShow
.text:00401408                push    0               ; int
.text:0040140A                push    0               ; int
```


这个地方的返回值，要加一个判断，不通过我们就直接退出程序，通过才可以

Return Values

If the function succeeds and the string pointed to by *lpString1* is less than the string pointed to by *lpString2*, the return value is negative; if the string pointed to by *lpString1* is greater than the string pointed to by *lpString2*, it is positive. If the strings are equal, the return value is zero.

这是lstrcmp的返回值，我们知道，等于0就是相等，所以，等于0才能进入下一步

```
.text:004013F9                call    lstrcmpA
.text:004013FE                push    [ebp+hWndParent] ; hWnd
.text:00401401                call    DestroyWindow
.text:00401406                push    0Ah             ; nCmdShow
.text:00401408                push    0               ; int
.text:0040140A                push    0               ; int
.text:0040140C                push    hInstance       ; hInstance
.text:00401412                call    sub_40146A
```

这一段的代码要等判断正确才可以执行

所以我们要想办法cmp之后跳到一个比较函数的地方

那我们就可以利用之前一些多余的部分，比如字符的编辑

那我们就可以利用之前一些多余的部分，比如空白的编辑栏

```
.text:004011CA                  mov     hWnd, eax
.text:004011CF                  push    0               ; lpParam
.text:004011D1                  push    hInstance       ; hInstance
.text:004011D7                  push    7               ; hMenu
.text:004011D9                  push    [ebp+hWndParent] ; hWndParent
.text:004011DC                  push    35h             ; nHeight
.text:004011DE                  push    21Ch            ; nWidth
.text:004011E3                  push    87h             ; Y
.text:004011E8                  push    0               ; X
.text:004011EA                  push    50000804h       ; dwStyle
.text:004011EF                  push    offset aYouMustReverse ; "You must reverse the program by making "...
.text:004011F4                  push    offset aEdit    ; "edit"
.text:004011F9                  push    0               ; dwExStyle
.text:004011FB                  call    CreateWindowExA
.text:00401200                  mov     dword_403404, eax
.text:00401205                  push    0               ; lpParam
.text:00401207                  push    hInstance       ; hInstance
.text:0040120D                  push    8               ; hMenu
.text:0040120F                  push    [ebp+hWndParent] ; hWndParent
.text:00401212                  push    35h             ; nHeight
.text:00401214                  push    21Ch            ; nWidth
.text:00401219                  push    0BEh            ; Y
.text:0040121E                  push    0               ; X
```

```
000005F4 004011F4: sub_401107+ED
```


我们把这个地方改了，所以在输入字符串后就直接跳到按钮部分的代码，就可以完美避开这一部分，然后这一部分就可以做我们私有地方，想干嘛干嘛

```
.text:004011CA                  mov     hWnd, eax
.text:004011CF                  push    0               ; lpParam
.text:004011D1                  push    hInstance       ; hInstance
.text:004011D7                  push    7               ; hMenu
```

.text:004011CA mov hWnd, eax
改为：jmp 40126c
然后对于

```
.text:004013EF                  push    offset String2  ; "Here is your mission, if you choose to "...
.text:004013F4                  push    offset String   ; lpString1
.text:004013F9                  call    lstrcmpA
.text:004013FE                  push    [ebp+hWndParent] ; hWnd
.text:00401401                  call    DestroyWindow
.text:00401406                  push    0Ah             ; nCmdShow
.text:00401408                  push    0               ; int
.text:0040140A                  push    0               ; int
.text:0040140C                  push    hInstance       ; hInstance
.text:00401412                  call    sub_40146A
```

.text:004013FE push [ebp+hWndParent] ; hWnd
改为
jmp 4011cf
然后把
4011cf之后改成这样（可以不改，直接退出程序，这就比较流氓了，哈哈哈哈哈哈啊哈哈哈）

```
004011CF      83F8 00       CMP EAX,0
004011D2      75 19         JNZ SHORT reversem.004011ED
004011D4      FF75 08       PUSH DWORD PTR SS:[EBP+8]
004011D7      E8 8E030000   CALL <JMP.&USER32.DestroyWindow>
004011DC      6A 0A         PUSH 0A
004011DE      6A 00         PUSH 0
004011E0      6A 00         PUSH 0
004011E2      68 F4334000   PUSH reversem.004033F4
004011E7      E8 7E020000   CALL reversem.0040146A
004011EC      90            NOP
004011ED      6A 00         PUSH 0
004011EF      E8 58030000   CALL <JMP.&KERNEL32.ExitProcess>
```

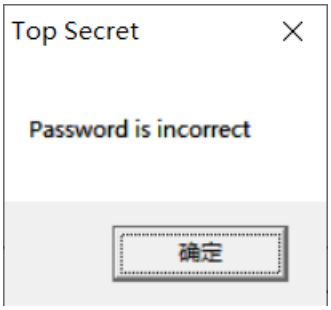
就是输对正常，输错直接退出
然后就可以了
因为我们对于

```
push    offset String2  ; "Here is your mission, if you choose to "...
push    offset String   ; lpString1
call    lstrcmpA
```

这个地方，string密码是空的，所以，这两个比较一定是不对的，所以他就无法进入啦
save the world
！

好丑…点击就退出
可以美化。。。看雪给的题解美化了，而且输错密码有提示
我看了一下，应该能实现出来吧…但我太懒了。。也太菜了


Continue to the Program
https://blog.csdn.net/zmx2473162621


Top Secret
Password is incorrect
确定