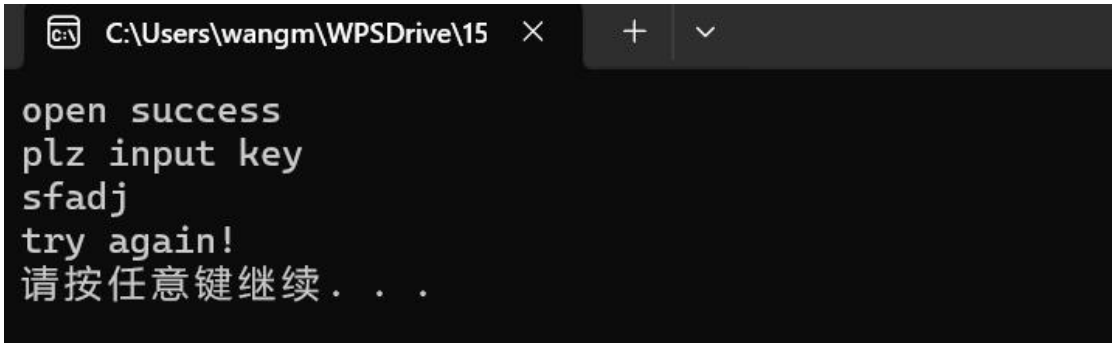
汇编语言与逆向工程第三次作业-HOOK 技术逆向分析

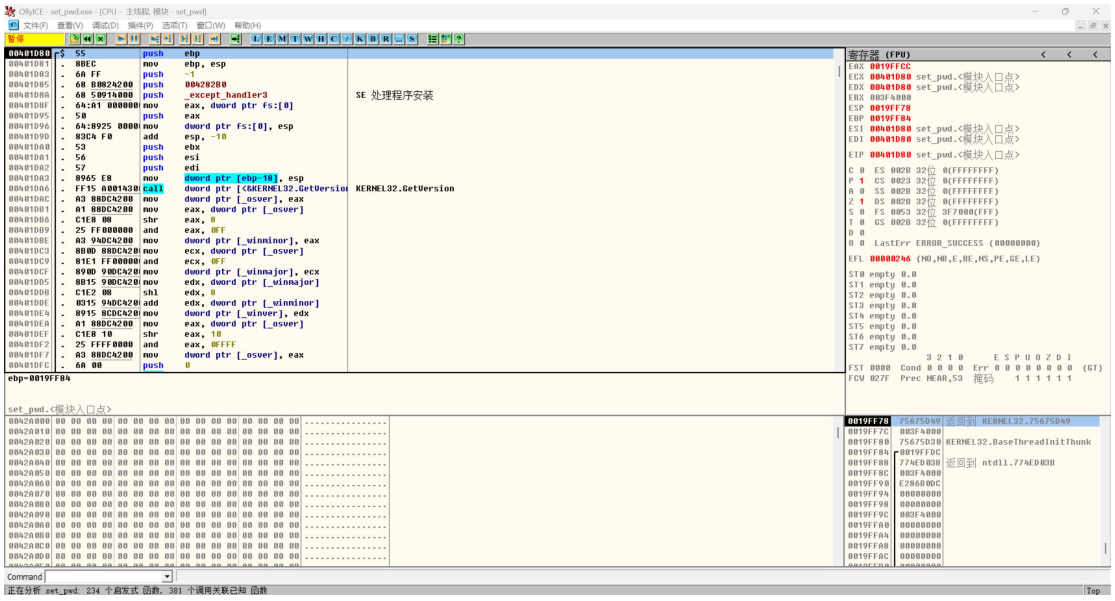
选择 WriteFileAPIHook1 文件逆向分析

首先找到要分析的程序：set_pwd.exe，尝试运行



随便输入字符，返回“try again!”

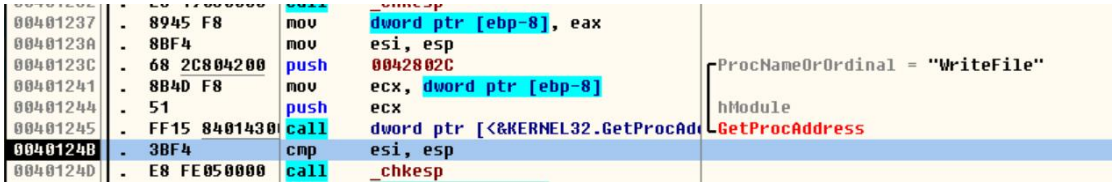
用 OllyICE 打开：



在 main 函数处设计断点，

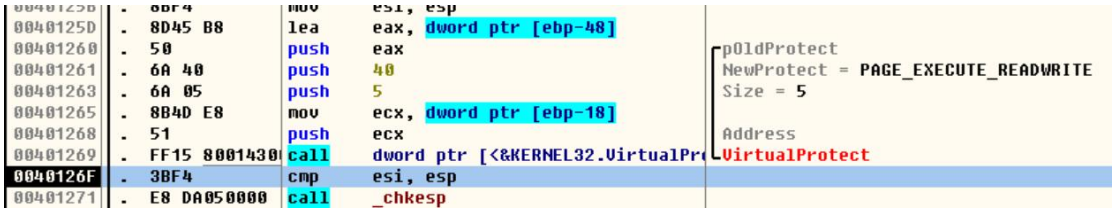
运行，来到 main 函数开始的位置，

发现 GetModuleHandleA 函数，取 kernel32.dell 句柄，



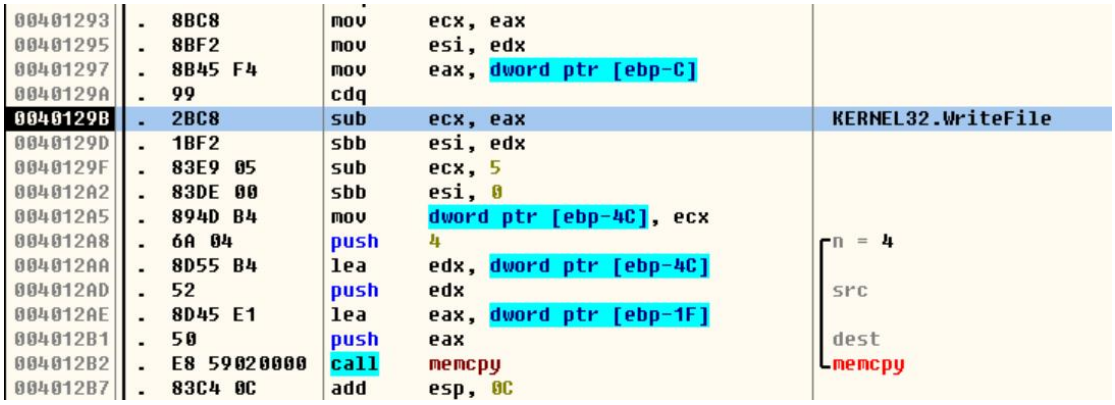
句 GetProcAddress 函数取 WriteFile 函数地址，

从寄存器区发现 WriteFile 函数地址为 7567EED0



VirtualProtect 函数将 WriteFile 函数权限更改为“可执行可读写”

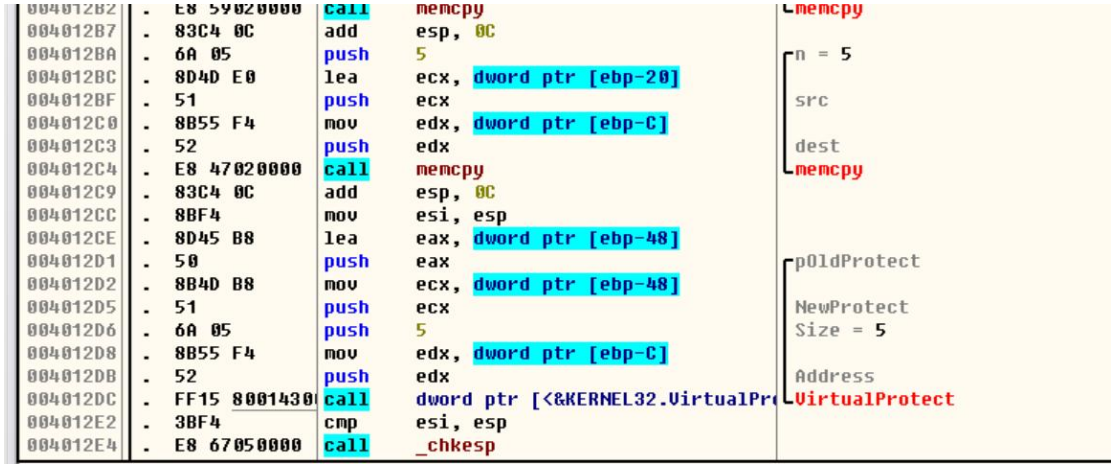
这个 memcpy 函数将 WriteFile 函数地址复制到 pOrgByte 地址中备份，



然后计算偏移地址，也就是在 WriteFile 函数第一句要跳转到的地址

连续的减法中，ecx 为目标地址，eax 是 WriteFile 函数地址，再减去 5，最后目标地址 ecx 放入 ebp-4C 的位置。

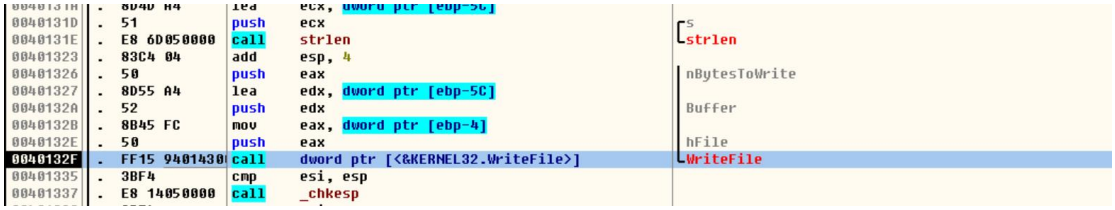
然后再 memcpy 函数中，将目标地址放入 ebp-1F 中



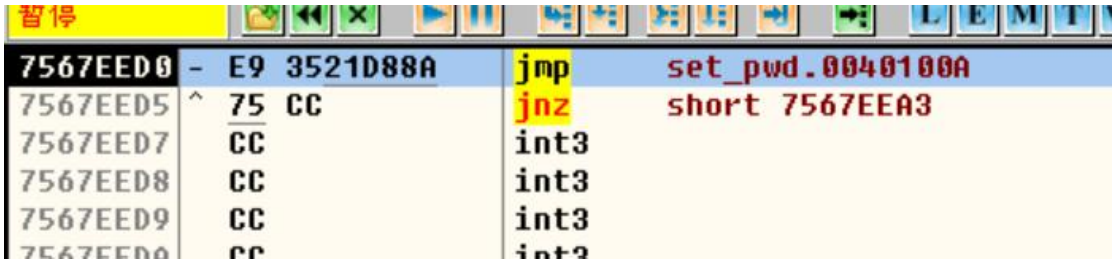
再调用 memcpy 函数中，将“跳转到目标地址”这条指令放入 ebp-C 中，

再用 VirtualProtect 函数将 WriteFile 函数权限更改回权限保护。

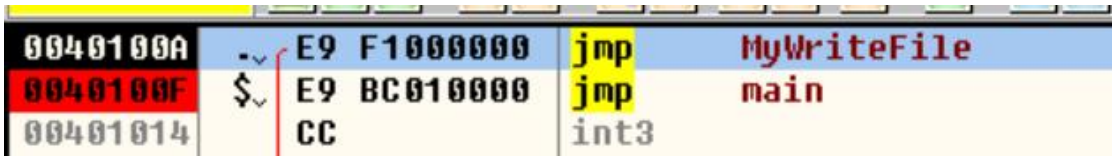
这里的 CreateFileA 函数猜测为创建文件函数，



在这里调用 WriteFile 函数，运行应该会跳转到目标地址

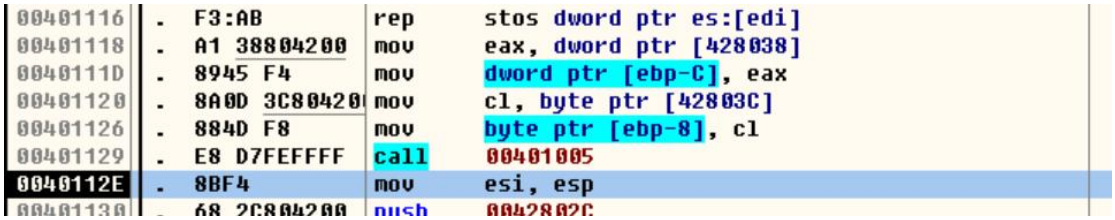


步入发现为 E9 3521D88A，跳转到目的地址

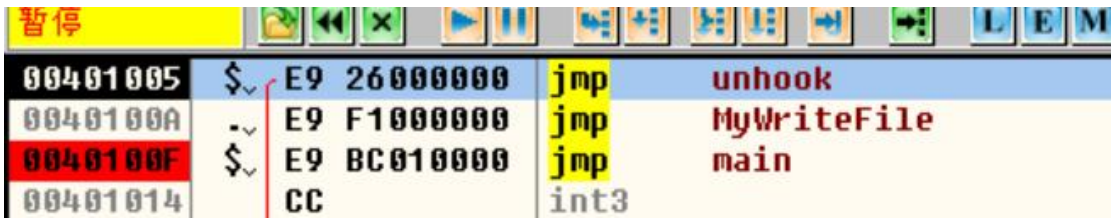


再次步入来到 MyWriteFile 的函数跳转处，步入

进入 MyWriteFile 函数

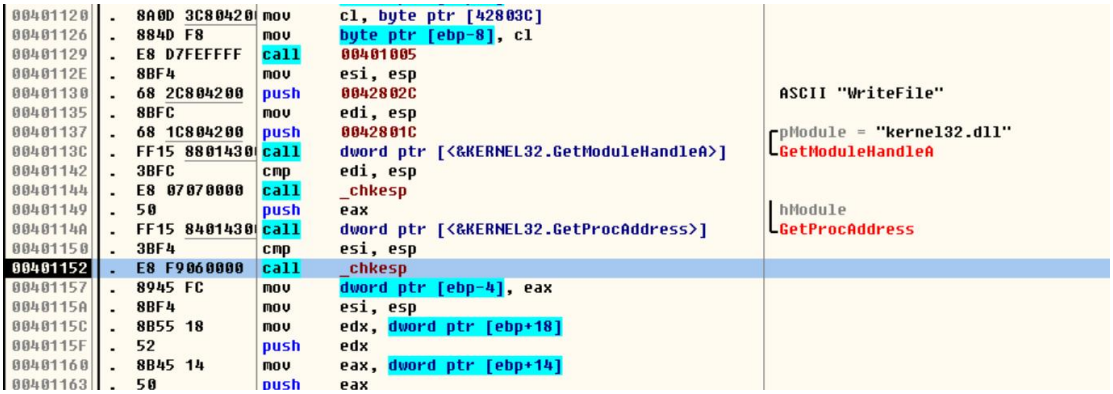


一步步运行，发现 call 00401005 函数，步入

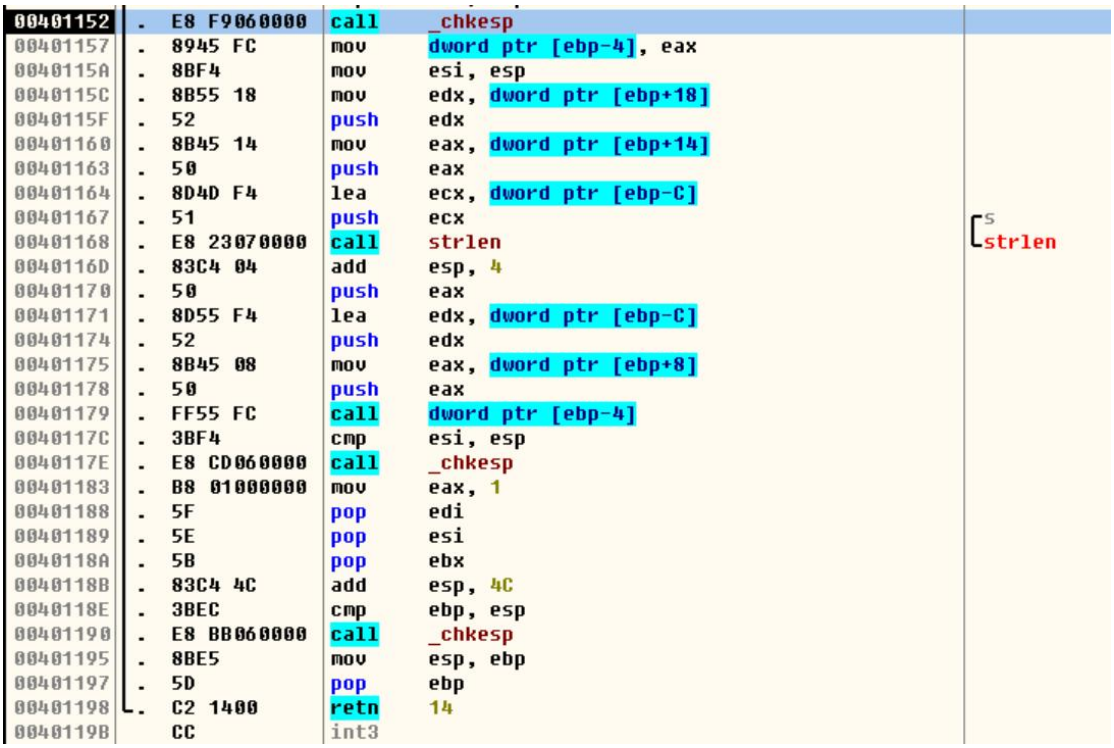


步入后发现调用 unhook 函数

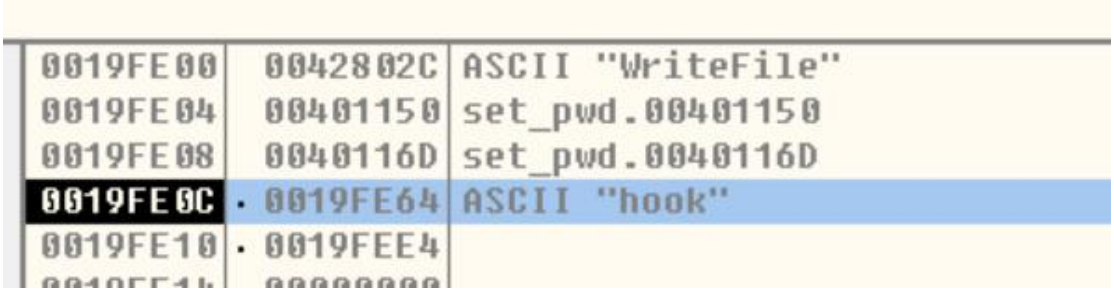
执行到返回，回到 MyWriteFile 函数



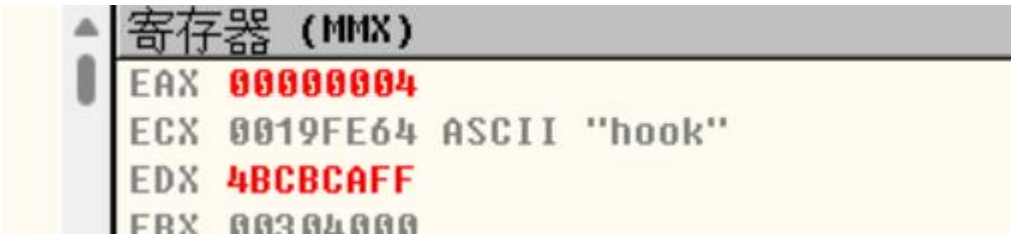
这里的函数调用，再次得到 WriteFile 函数地址，为继续完成 WriteFile 函数的功能



下面这里完成 WriteFile 函数的写入数据功能



运行到压栈后，发现字符串为“hook”



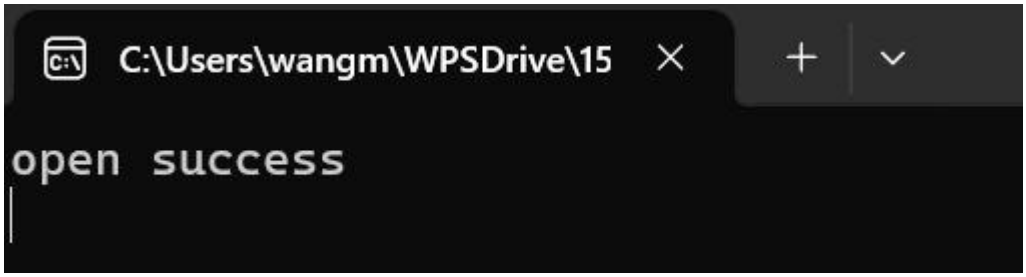
且字符串长度为 4，正确

```
0040132F   .  FF15 94014300  call    dword ptr [<&KERNEL32.WriteFile>]      WriteFile
00401335   .  3BF4           cmp     esi, esp
00401337   .  E8 14050000    call    _chkesp
0040133C   .  8BF4           mov     esi, esp
0040133E   .  8B4D FC        mov     ecx, dword ptr [ebp-4]
00401341   .  51             push    ecx                                    hObject
00401342   .  FF15 90014300  call    dword ptr [<&KERNEL32.CloseHandle>]     CloseHandle
00401348   .  3BF4           cmp     esi, esp
0040134A   .  E8 01050000    call    _chkesp
0040134F   .  68 B8804200    push    004280B8                               mode = "r"
00401354   .  68 BC804200    push    004280BC                               path = "pwd.txt"
00401359   .  E8 020A0000    call    fopen                                  fopen
0040135E   .  83C4 08        add     esp, 8
00401361   .  8945 F0        mov     dword ptr [ebp-10], eax
```

执行完返回 main 函数，发现是 cmp 指令，此时 esi 和 esp 相同，都为 0019FE8C,故，eax=1，

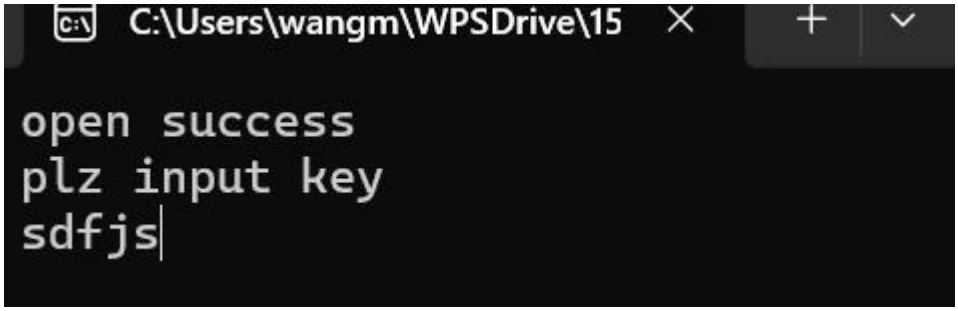
```
00401337   .  E8 14050000    call    _chkesp
0040133C   .  8BF4           mov     esi, esp
0040133E   .  8B4D FC        mov     ecx, dword ptr [ebp-4]
00401341   .  51             push    ecx                                    hObject
00401342   .  FF15 90014300  call    dword ptr [<&KERNEL32.CloseHandle>]     CloseHandle
00401348   .  3BF4           cmp     esi, esp
0040134A   .  E8 01050000    call    _chkesp
0040134F   .  68 B8804200    push    004280B8                               mode = "r"
00401354   .  68 BC804200    push    004280BC                               path = "pwd.txt"
00401359   .  E8 020A0000    call    fopen                                  fopen
0040135E   .  83C4 08        add     esp, 8
00401361   .  8945 F0        mov     dword ptr [ebp-10], eax
00401364   .  837D F0 00     cmp     dword ptr [ebp-10], 0
00401368   .  74 0F          je      short 00401379
0040136A   .  68 A8804200    push    004280A8                               format = "open success",LF,""
0040136F   .  E8 7C080000    call    printf                                 printf
00401374   .  83C4 04        add     esp, 4
00401377   .  EB 0D          jmp     short 00401386
00401379   >  68 98804200    push    00428098                               format = "open fail",LF,""
0040137E   .  E8 6D080000    call    printf                                 printf
00401383   .  83C4 04        add     esp, 4
00401386   >  837D F0 00     cmp     dword ptr [ebp-10], 0
0040138A   .  74 17          je      short 004013A3
0040138C   .  8D55 D0        lea     edx, dword ptr [ebp-30]
0040138F   .  52             push    edx
```

继续运行，命令行打印“open success”



又打印“plz input key”的字样

```
004013BA   .  83C4 04      add     esp, 4
004013BD   .  8D4D C0      lea     ecx, dword ptr [ebp-40]
004013C0   .  51           push    ecx
004013C1   .  68 94804200  push    00428094                 Format = "%s"
004013C6   .  E8 35070000  call    scanf                    scanf
004013CB   .  83C4 08      add     esp, 8
```



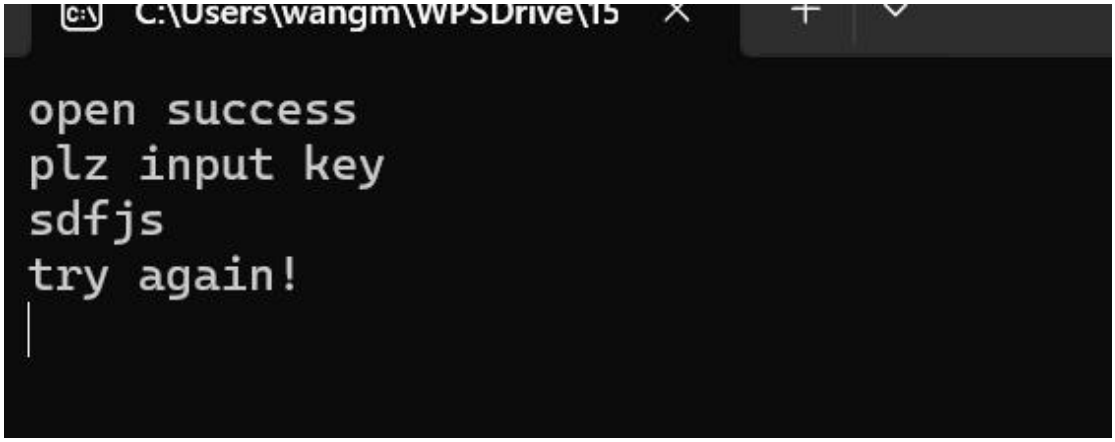
运行搭配 scanf 函数处要求输入字符，随便输入“sdfjs”

```
004013D4   .  8B55 F0      mov     edx, dword ptr [ebp-10]
004013D7   .  52           push    edx                      stream
004013D8   .  E8 43060000  call    fclose                   fclose
004013DD   .  83C4 04      add     esp, 4
004013E0   >  8BF4         mov     esi, esp
004013E2   .  8D45 D0      lea     eax, dword ptr [ebp-30]
004013E5   .  50           push    eax                      String2
004013E6   .  8D4D C0      lea     ecx, dword ptr [ebp-40]
004013E9   .  51           push    ecx                      String1
004013EA   .  FF15 8C01430 call    dword ptr [<&KERNEL32.lstrcmpA>]  lstrcmpA
004013F0   .  3BF4         cmp     esi, esp
004013F2   .  E8 59040000  call    _chkesp
004013F7   .  8945 BC      mov     dword ptr [ebp-44], eax
004013FA   .  837D BC 00   cmp     dword ptr [ebp-44], 0
004013FE   .  75 0F        jnz     short 0040140F
00401400   .  68 58804200  push    00428058                 Format = "congratulations!",LF,""
00401405   .  E8 E6070000  call    printf                   printf
```

继续步过，完成 lstrcmpA 函数比较

```
004013F2   .  E8 59040000  call    _chkesp
004013F7   .  8945 BC      mov     dword ptr [ebp-44], eax
004013FA   .  837D BC 00   cmp     dword ptr [ebp-44], 0
004013FE   .  75 0F        jnz     short 0040140F
00401400   .  68 58804200  push    00428058                 Format = "congratulations!",LF,""
00401405   .  E8 E6070000  call    printf                   printf
0040140A   .  83C4 04      add     esp, 4
0040140D   .  EB 0D        jmp     short 0040141C
0040140F   >  68 48804200  push    00428048                 Format = "try again!",LF,""
00401414   .  E8 D7070000  call    printf                   printf
00401419   .  83C4 04      add     esp, 4
```

由于 eax=1，不为 0，我们跳转，输出“try again！”




```
00401419  .  83C4 04      add     esp, 4
0040141C  >  68 40804200  push    00428040                 command = "Pause"
00401421  .  E8 EA040000  call    system                   system
00401426  .  83C4 04      add     esp, 4
```

最后调用 system 程序执行完成；

因为最后没有输出“congratulations！”，我们在输入函数 scanf 之前设置断点，运行到断点处，查看如何才能输出“congratulations！”

```
00401399  .  E8 C2070000  call    fscanf                   fscanf
0040139E  .  83C4 0C      add     esp, 0C
004013A1  .  EB 0D        jmp     short 004013B0
004013A3  >  68 84804200  push    00428084                 format = "scan fail",LF,""
004013A8  .  E8 43080000  call    printf                   printf
004013AD  .  83C4 04      add     esp, 4
004013B0  >  68 70804200  push    00428070                 format = "plz input key",LF,""
004013B5  .  E8 36080000  call    printf                   printf
004013BA  .  83C4 04      add     esp, 4
004013BD  .  8D4D C0      lea     ecx, dword ptr [ebp-40]
004013C0  .  51           push    ecx
004013C1  .  68 94804200  push    00428094                 format = "%s"
004013C6  .  E8 35070000  call    scanf                    scanf
004013CB  .  83C4 08      add     esp, 8
004013CE  .  837D F0 00   cmp     dword ptr [ebp-10], 0
004013D2  .  74 0C        je      short 004013E0
004013D4  .  8B55 F0      mov     edx, dword ptr [ebp-10]
004013D7  .  52           push    edx                      stream
004013D8  .  E8 43060000  call    fclose                   fclose
004013DD  .  83C4 04      add     esp, 4
004013E0  >  8BF4         mov     esi, esp
004013E2     8D45 D0      lea     eax, dword ptr [ebp-30]
004013E5     50           push    eax
004013E6     8D4D C0      lea     ecx, dword ptr [ebp-40]
004013E9     51           push    ecx
004013EA     FF15 8C01430 call    dword ptr [<&KERNEL32.lstrcmpA>]   KERNEL32.lstrcmpA
```

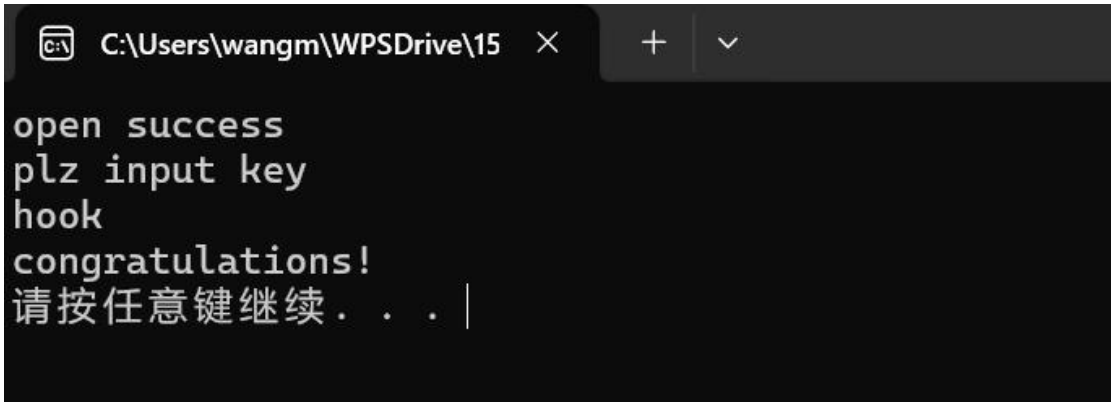
```
EAX 0019FF04 ASCII "hook"
ECX 0019FEF4 ASCII "mnbjg"
EDX 0042AB00 set_pwd.0042AB00
EBX 0025B000
ESP 0019FE88
EBP 0019FF34
ESI 0019FE8C
EDI 0019FF34

EIP 004013E9 set_pwd.004013E9

C 0  ES 002B 32位 0(FFFFFFFF)
P 0  CS 0023 32位 0(FFFFFFFF)
```

随即输入“mnbjg”被复制到 ecx，“hook”被复制到 eax，因此，若我们输入“hook”，下面的 lstrcmpA 函数即可返回‘0’，eax 为 0，则输出“congratulations！”

验证：



验证成功，即该程序的 HOOK 的作用为将 WriteFile 函数原本要写入的“real_pwd”改为“hook”。