

# 花指令专栏

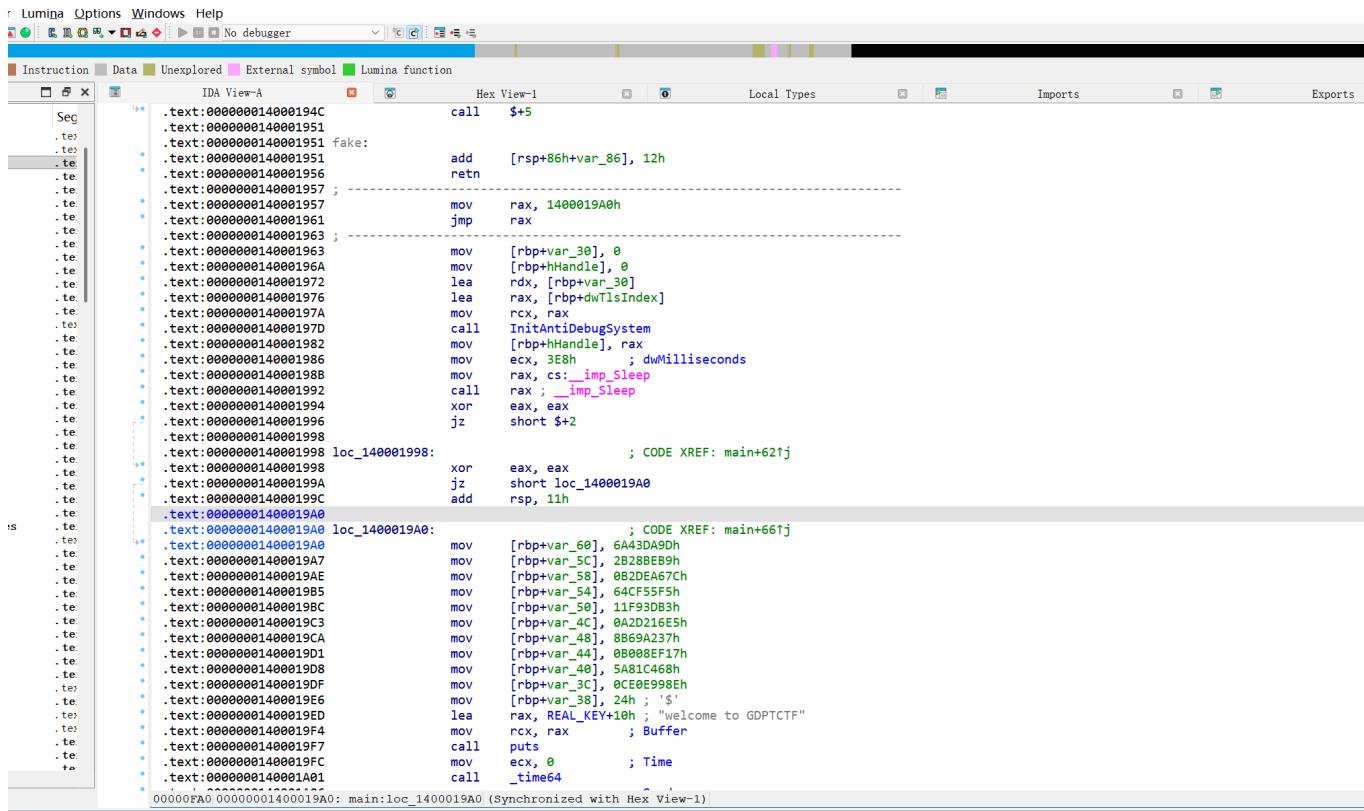
# 花指令

张程思

注:本栏只去除花指令,后面的内容没继续进行

## 一,gdptctf

1,打开后f5编译不出来什么东西,猜测有花指令



```
l Lumiga Options Windows Help
File Edit View Tools Options Help
Instruction Data Unexplored External symbol Lumina function
IDA View-A Hex View-1 Local Types Imports Exports
Seg .text:00000014000194C call    $+5
      .text:000000140001951
      .text:000000140001951 fake:
      .text:000000140001951 add    [rsp+86h+var_86], 12h
      .text:000000140001956 retn
      .text:000000140001957 ;-----.
      .text:000000140001957 ;-----.
      .text:000000140001957 mov    rax, 1400019A0h
      .text:000000140001961 jmp    rax
      .text:000000140001963 ;-----.
      .text:000000140001963 mov    [rbp+var_30], 0
      .text:000000140001964 mov    [rbp+hHandle], 0
      .text:000000140001972 lea    rdx, [rbp+var_30]
      .text:000000140001976 lea    rax, [rbp+dwTlsIndex]
      .text:00000014000197A mov    rcx, rax
      .text:00000014000197D call   InitAntiDebugSystem
      .text:000000140001982 mov    [rbp+hHandle], rax
      .text:000000140001986 mov    ecx, 3E8h ; dwMilliseconds
      .text:000000140001988 mov    rax, cs:_imp_Sleep
      .text:000000140001992 call   rax ; _imp_Sleep
      .text:000000140001994 xor    eax, eax
      .text:000000140001996 jz    short $+2
      .text:000000140001998 loc_140001998: ; CODE XREF: main+62j
      .text:000000140001998 xor    eax, eax
      .text:00000014000199A jz    short loc_1400019A0
      .text:00000014000199C add    rsp, 11h
      .text:0000001400019A0
      .text:0000001400019A0 loc_1400019A0: ; CODE XREF: main+66j
      .text:0000001400019A0 mov    [rbp+var_60], 6A43DA90h
      .text:0000001400019A7 mov    [rbp+var_5C], 2B28BE89h
      .text:0000001400019AE mov    [rbp+var_58], 0B2DEA67Ch
      .text:0000001400019B5 mov    [rbp+var_54], 64CF55F5h
      .text:0000001400019BC mov    [rbp+var_50], 11F93DB3h
      .text:0000001400019C3 mov    [rbp+var_4C], 0A2D216E5h
      .text:0000001400019CA mov    [rbp+var_48], 8869A237h
      .text:0000001400019D1 mov    [rbp+var_44], 0B008EF17h
      .text:0000001400019D8 mov    [rbp+var_40], 5A81C468h
      .text:0000001400019DF mov    [rbp+var_3C], 0CE0E998Eh
      .text:0000001400019E6 mov    [rbp+var_38], 24h ; '$'
      .text:0000001400019ED lea    rax, REAL_KEY+10h ; "welcome to GDPTCTF"
      .text:0000001400019F4 mov    rcx, rax ; Buffer
      .text:0000001400019F7 call   puts
      .text:0000001400019FC mov    ecx, 0 ; Time
      .text:000000140001A01 call   _time64
      .text:00000FA0 000000001400019A0: main:loc_1400019A0 (Synchronized with Hex View-1)
```

5, Apr 2 2024, 10:12:12) [MSC v.1938 64 bit (AMD64)]  
APython Team <idapython@googlegroups.com>  
-----  
4 7-14  
been proximated

The screenshot shows the IDA Pro interface with two windows open. The top window, titled 'IDA View-A', displays the C code for the main function:

```
1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     _main();
4     return 0;
5 }
```

The bottom window, also titled 'IDA View-A', displays the assembly code for the \_main function:

```
1 void __cdecl _main()
2 {
3     if ( !initialized )
4     {
5         initialized = 1;
6         _do_global_ctors();
7     }
8 }
```

The assembly code is highlighted in yellow, indicating it is the current view. The left sidebar shows a list of sections, with the '.text' section selected.

2,断点调试,打上nop,根据f8的跳跃位置来nop

```
.text:00007FF76D0D193F call    __main
.text:00007FF76D0D1944 xor     eax, eax
.text:00007FF76D0D1946 jz      short loc_7FF76D0D194C
.text:00007FF76D0D1948 nop
.text:00007FF76D0D1949 nop
.text:00007FF76D0D194A nop
.text:00007FF76D0D194B nop
.text:00007FF76D0D194C loc_7FF76D0D194C:           ; CODE XREF: main+12tj
.text:00007FF76D0D194D nop
.text:00007FF76D0D194E nop
.text:00007FF76D0D194F nop
.text:00007FF76D0D1950 nop
.text:00007FF76D0D1951
.text:00007FF76D0D1951 fake:
.text:00007FF76D0D1951 nop
.text:00007FF76D0D1952 nop
.text:00007FF76D0D1953 nop
.text:00007FF76D0D1954 nop
.text:00007FF76D0D1955 nop
.text:00007FF76D0D1956 nop
.text:00007FF76D0D1957 nop
.text:00007FF76D0D1958 nop
.text:00007FF76D0D1959 nop
.text:00007FF76D0D195A nop
.text:00007FF76D0D195B nop
.text:00007FF76D0D195C nop
00007FF76D0D1951: main:fake (Synchronized with RIP)
```

```

.text:0000FF76D0D1968 nop
.text:0000FF76D0D1969 nop
.text:0000FF76D0D196A mov [rbp+hHandle], 0
.text:0000FF76D0D1972 lea rdx, [rbp+var_30] ; _QWORD
.text:0000FF76D0D1976 lea rax, [rbp+var_30+4]
.text:0000FF76D0D197A mdv rcx, rax ; p_dwTlsIndex
.text:0000FF76D0D197D call InitAntiDebugSystem
.text:0000FF76D0D1982 mov [rbp+hHandle], rax
.text:0000FF76D0D1986 mov ecx, 3E8h ; dwMilliseconds
.text:0000FF76D0D198B mov rax, cs:_imp_Sleep
.text:0000FF76D0D1992 call rax ; _imp_Sleep
.text:0000FF76D0D1994 xor eax, eax
.text:0000FF76D0D1996 jz short $+2
.text:0000FF76D0D1998
.text:0000FF76D0D1998 loc_7FF76D0D1998: ; CODE XREF: main+62↑j
    xor eax, eax
.text:0000FF76D0D199A jz short loc_7FF76D0D19A0
.text:0000FF76D0D199C nop
.text:0000FF76D0D199D nop
.text:0000FF76D0D199E nop
.text:0000FF76D0D199F nop
.text:0000FF76D0D19A0
.text:0000FF76D0D19A0 loc_7FF76D0D19A0: ; CODE XREF: main+66↑j
    dword ptr [rbp+var_60], 6A43DA9Dh
.text:0000FF76D0D19A7 mov dword ptr [rbp+var_60+4], 2B288EB9h
.text:0000FF76D0D19AE mov [rbp+var_58], 0B2DEA67Ch
.text:0000FF76D0D19B5 mov [rbp+var_54], 64CF55F5h
.text:0000FF76D0D19BC mov [rbp+var_50], 11F93DB3h
.text:0000FF76D0D19C3 mov [rbp+var_4C], 0A2D216E5h

```

00000F7A 00007FF76D0D197A: main+46 (Synchronized with RIP)

```

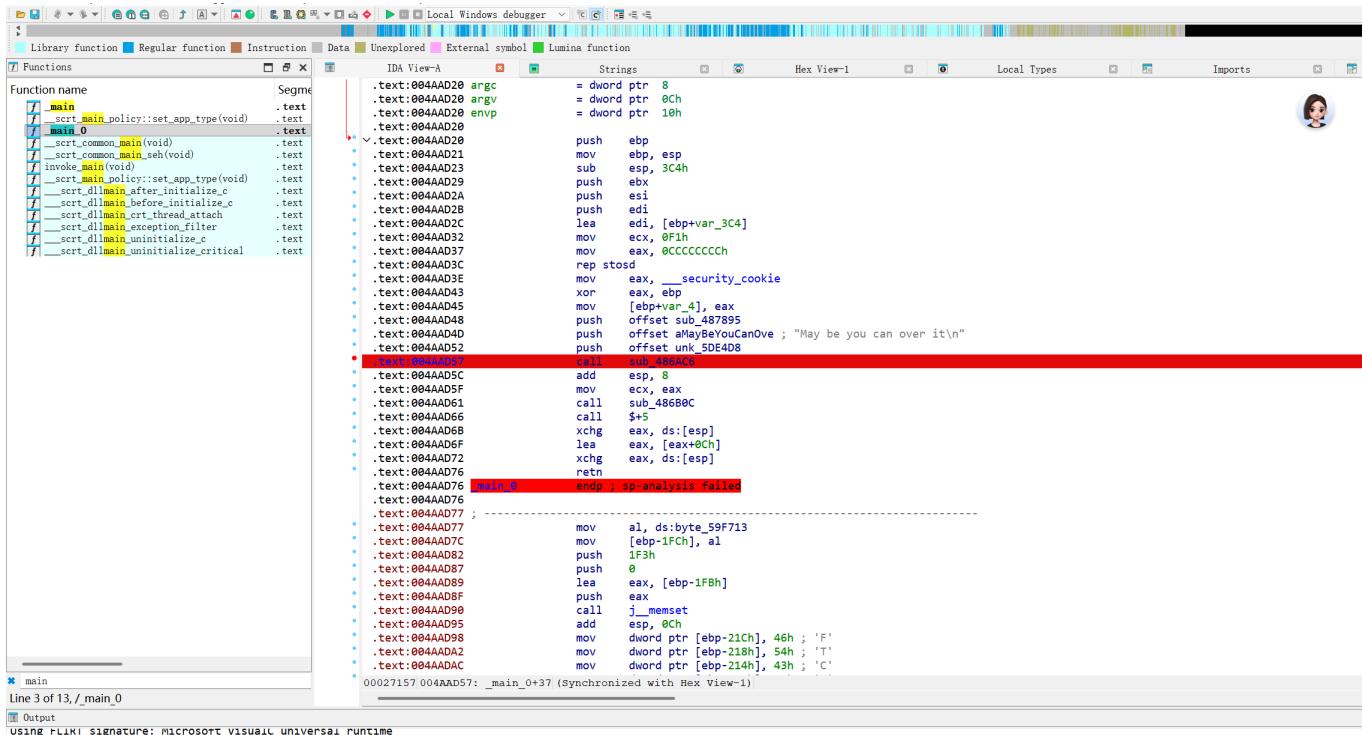
00007FF76D0D1934
00007FF76D0D1934 ; Attributes: bp-based frame
00007FF76D0D1934
00007FF76D0D1934 ; int __fastcall main(int argc, const char **argv, const char **envp)
00007FF76D0D1934 public main
00007FF76D0D1934 main proc near ; CODE XREF: __tmainCRTStartup+B4↑p
00007FF76D0D1934 ; DATA XREF: .pdata:00007FF76D0E506C↓o ...
00007FF76D0D1934
00007FF76D0D1934 var_64= dword ptr -64h
00007FF76D0D1934 var_60= qword ptr -60h
00007FF76D0D1934 var_58= dword ptr -58h
00007FF76D0D1934 var_54= dword ptr -54h
00007FF76D0D1934 var_50= dword ptr -50h
00007FF76D0D1934 var_4C= dword ptr -4Ch
00007FF76D0D1934 var_48= dword ptr -48h
00007FF76D0D1934 var_44= dword ptr -44h
00007FF76D0D1934 var_40= dword ptr -40h
00007FF76D0D1934 var_3C= dword ptr -3Ch
00007FF76D0D1934 var_38= dword ptr -38h
00007FF76D0D1934 var_30= qword ptr -30h
00007FF76D0D1934 var_28= qword ptr -28h
00007FF76D0D1934 var_20= dword ptr -20h
00007FF76D0D1934 n10= qword ptr -1Ch
00007FF76D0D1934 var_14= dword ptr -14h
00007FF76D0D1934 hHandle= qword ptr -10h
00007FF76D0D1934 var_4= dword ptr -4
00007FF76D0D1934
00007FF76D0D1934 push rbp
00007FF76D0D1935 mov rbp, rsp

```

00007FF76D0D1934: main (Synchronized with RIP)

## 二,一杯花茶

1,ida中main无法正常编译,并且看到有call加几这样类似的东西,直接断点调试



The screenshot shows the IDA Pro interface with the assembly view open. The assembly code for the function `_main_0` is displayed. A red arrow highlights the `call` instruction at address `004AAAD52`, which is annotated with the text "endp ; sp-analysis failed". The assembly code includes various standard C runtime initialization and cleanup routines like `__scrt_common_main`, `__scrt_common_main_seh`, and `__scrt_dllmain_before_initialize_c`.

```
.text:004AAAD20 argc      = dword ptr 8
.text:004AAAD20 argv      = dword ptr 0Ch
.text:004AAAD20 envp     = dword ptr 10h
.text:004AAAD20
    push    ebp
    mov     ebp, esp
    sub    esp, 3C4h
    push    ebx
    push    esi
    push    edi
    lea    edi, [ebp+var_3C4]
    mov    ecx, 0Fin
    mov    eax, 0CCCCCCCCh
    rep    stosd
    mov    eax, __security_cookie
    xor    eax, ebp
    mov    [ebp+var_4], eax
    push    offset sub_487895
    push    offset unk_5DE4D8 ; "May be you can over it\n"
    push    offset unk_5DE4D8
    .text:004AAAD52 endp ; sp-analysis failed
```

2,nop掉一个call和一个return再创建个函数就可以看到正确的伪代码了

IDA View-EIP

```

.text:004AAD45 mov    [ebp+var_4], eax
.text:004AAD48 push   offset p_sub_487895      ; p_sub_487895
.text:004AAD4D push   offset aMayBeYouCanOve    ; "May be you can over it\n"
.text:004AAD52 push   offset off_5DE4D8        ; _DWORD
• .text:004AAD57 call   sub_486AC6
.text:004AAD5C add    esp, 8
.text:004AAD5F mov    ecx, eax
.text:004AAD61 call   sub_486B0C
.text:004AAD66 nop
.text:004AAD67 nop
• .text:004AAD68 nop |
.text:004AAD69 nop
• .text:004AAD6A nop
.text:004AAD6B xchng eax, ds:[esp]
.text:004AAD6F lea    eax, [eax+0Ch]
.text:004AAD72 xchng eax, ds:[esp]
• .text:004AAD76 nop
• .text:004AAD77 mov    al, ds:byte_59F713
• .text:004AAD7C mov    [ebp+Str], al
• .text:004AAD82 push   1F3h                      ; Size
• .text:004AAD87 push   0                         ; Val
• .text:004AAD89 lea    eax, [ebp+var_1FB]
• .text:004AAD8F push   eax
• .text:004AAD90 call   j__memset                ; void *
• .text:004AAD95 add   esp, 0Ch

```

EIP

Pseudocode-A

```

1 int __cdecl main_0(int argc, const char **argv, const char **envp)
2 {
3     _BYTE v5[12]; // [esp+0h] [ebp-3D0h] BYREF
4     size_t n23; // [esp+190h] [ebp-240h]
5     _DWORD v7[6]; // [esp+19Ch] [ebp-234h] BYREF
6     _DWORD v8[8]; // [esp+1B4h] [ebp-21Ch] BYREF
7     char Str[504]; // [esp+1D4h] [ebp-1FCh] BYREF
8
9     sub_486AC6(&off_5DE4D8, "May be you can over it\n");
10    _InterlockedExchange(MK_FP(_DS_, v5), _InterlockedExchange(MK_FP(_DS_, v5), sub_486B0C(sub_487895)) + 12);
11    memset(Str, 0, 500);
12    v5[0] = 70;
13    v5[1] = 84;
14    v5[2] = 67;
15    v5[3] = 83;
16    v5[4] = 65;
17    v5[5] = 68;
18    v7[0] = -1161949172;
19    v7[1] = -98537090;
20    v7[2] = -1796982650;
21    v7[3] = 539168800;
22    j_puts("Give me your flag:");
23    sub_48531F(&dword_5DE430, Str);
24    n23 = j_strlen(Str);
25    if ( n23 == 23 )
26    {
27        sub_488123(Str, v8, v7);
28        if ( (unsigned __int8)sub_485C84(Str) )
29            sub_486AC6(&off_5DE4D8, "Good");

```

00027120\_main\_0:1 (4AAD20)

Hex View-1

|          |   |               |
|----------|---|---------------|
| 004AAD50 | 59 00 68 D8 E4 5D 00 E8 6A BD FD FF 83 C4 08 8B Y.h..].....     | 00:0000 0019F |
| 004AAD60 | C8 E8 A6 BD FD FF 90 90 90 90 3E 87 04 24 8D .....>..\$.....    | 01:0004 0019F |
| 004AAD70 | 40 0C 3E 87 04 24 90 A0 13 F7 59 00 88 85 04 FE @.>..\$.....    | 02:0008 0019F |
| 004AAD80 | FF FF 68 F3 03 00 00 6A 00 8D 85 05 FE FF FF 50 ..h....j....P   | 03:000C 0019F |
| 004AAD90 | E8 50 A8 FD FF 83 C4 0C C7 85 E4 FD FF FF 46 00 .....h....F.    | 04:0010 0019F |
| 004AADA0 | 00 00 C7 85 E8 FD FF FF 54 00 00 00 C7 85 EC FD ..h....T...h..  | 05:0014 0019F |
| 004AADB0 | FF FF 43 00 00 00 C7 85 F0 FD FF FF 53 00 00 00 ..C...h....S... | 06:0018 0019F |

### 三, litectf

# 1,ida分析找不到main函数,shift+f12跟踪查找一下,发现了一堆花指令

IDA View-A

| Address         | Length     | Type  | String                        |
|-----------------|------------|---|-------------------------------|
| .text:0000000D  | C          | _Psave_guard  |                               |
| .text:00000007  | C          | _Psave  |                               |
| .text:00000006  | C          | _Lock   |                               |
| .text:00000006  | C          | Input   |                               |
| .rdata:00000012 | C          | Unknown exception   |                               |
| .rdata:00000009 | C          | bad cast  |                               |
| .rdata:00000006 | C          | flag:   |                               |
| .rdata:0000003A | C          | D:\a\work\1\s\src\vctools\crt\github\stl\src\locale0.cpp  |                               |
| .rdata:0000001C | C          | Stack around the variable '   |                               |
| .rdata:00000011 | C          | ' was corrupted.  |                               |
| .rdata:0000000F | C          | The variable '  |                               |
| .rdata:0000002B | C          | ' is being used without being initialized.  |                               |
| .rdata:000000D4 | C          | The value of ESP was not properly saved across a function call. This is usually a result of calling a function declared with one calling convention |                               |
| .rdata:0000011D | C          | A cast to a smaller data type has caused a loss of data. If this was intentional, you should mask the source of the cast with the appropriate bitma |                               |
| .rdata:0000010D | C          | Stack memory was corrupted\r\n  |                               |
| .rdata:00000036 | C          | A local variable was used before it was initialized\r\n   |                               |
| .rdata:0000002C | C          | Stack memory around _alloca was corrupted\r\n   |                               |
| .rdata:0000001E | C          | Unknown Runtime Check Error\r\n   |                               |
| .rdata:00000011 | C          | Unknown Filename  |                               |
| .rdata:00000014 | C          | Unknown Module Name   |                               |
| .rdata:00000020 | C          | Run-Time Check Failure # <b>%d - %s</b>   |                               |
| .rdata:00000026 | C          | Stack corrupted near unknown variable   |                               |
| .rdata:00000006 | C          | %.2X  |                               |
| .rdata:00000049 | C          | Stack area around _alloca memory reserved by this function is corrupted\n   |                               |
| .rdata:00000009 | C          | \nData: <   |                               |
| .rdata:0000002A | C          | \nAllocation number within this function:   |                               |
| .rdata:00000008 | C          | \nSize:   |                               |
| .rdata:0000000D | C          | \nAddress: 0x   |                               |
| .rdata:00000048 | C          | Stack area around _alloca memory reserved by this function is corrupted   |                               |
| .rdata:0000001A | C          | %\$%\$%\$%\$%\$%\$%\$%\$%   |                               |
| .rdata:00000034 | C          | A variable is being used without being initialized.   |                               |
| .rdata:00000019 | C          | Stack pointer corruption  |                               |
| .rdata:0000002A | C          | Cast to smaller type causing loss of data   |                               |
| .rdata:00000018 | C          | Stack memory corruption   |                               |
| .rdata:0000002A | C          | Local variable used before initialization   |                               |
| .rdata:0000001F | C          | Stack around _alloca corrupted  |                               |
| .rdata:00000008 | C          | RegOpenKeyExW   |                               |
| .rdata:0041DC5C |            |   | ; sub_41114A                  |
| .rdata:0041DC60 | dd offset  | sub_41114A  |                               |
| .rdata:0041DC64 | align 8    |   |                               |
| .rdata:0041DC68 | aBadCast   | db 'bad cast',0   | ; DATA XREF: sub_413D30+2D↑o  |
| .rdata:0041DC71 | align 4    |   |                               |
| .rdata:0041DC74 | unk_41DC74 | db 0CAh   | ; DATA XREF: .text:0041503C↑o |
| .rdata:0041DC75 | db 0E4h    |   |                               |
| .rdata:0041DC76 | db 0C8h    |   |                               |
| .rdata:0041DC77 | db 0EBh    |   |                               |
| .rdata:0041DC78 | db 0C4h    |   |                               |
| .rdata:0041DC79 | db 0E3h    |   |                               |
| .rdata:0041DC7A | db 0B5h    |   |                               |
| .rdata:0041DC7B | db 0C4h    |   |                               |
| .rdata:0041DC7C | db 66h ; f |   |                               |
| .rdata:0041DC7D | db 6Ch ; l |   |                               |
| .rdata:0041DC7E | db 61h ; a |   |                               |
| .rdata:0041DC7F | db 67h ; g |   |                               |
| .rdata:0041DC80 | db 3Ah ; : |   |                               |
| .rdata:0041DC81 | db 0       |   |                               |
| .rdata:0041DC82 | db 0       |   |                               |
| .rdata:0041DC83 |            |   |                               |
| .rdata:0041DC84 |            |   |                               |
| .rdata:0041DC85 |            |   |                               |
| .rdata:0041DC86 |            |   |                               |
| .rdata:0041DC87 |            |   |                               |
| .rdata:0041DC88 |            |   |                               |
| .rdata:0041DC89 |            |   |                               |
| .rdata:0041DC8A |            |   |                               |
| .rdata:0041DC8B |            |   |                               |
| .rdata:0041DC8C |            |   |                               |
| .rdata:0041DC8D |            |   |                               |
| .rdata:0041DC8E |            |   |                               |
| .rdata:0041DC8F |            |   |                               |
| .rdata:0041DC90 |            |   |                               |
| .rdata:0041DC91 |            |   |                               |
| .rdata:0041DC92 |            |   |                               |
| .rdata:0041DC93 |            |   |                               |

xrefs to unk\_41DC74

| Direct | Typ | Address        | Text                   |
|--------|-----|----------------|------------------------|
| Up     | o   | .text:0041503C | push offset unk_41DC74 |

Line 1 of 1

OK Cancel Search Help

```

ext:00414FB5      mov    [ebp-4], eax
ext:00414FB8      mov    dword ptr [ebp-18h], 11223344h
ext:00414FBF      mov    dword ptr [ebp-14h], 55667788h
ext:00414FC6      mov    dword ptr [ebp-10h], 99AABBCCh
ext:00414FC0      mov    dword ptr [ebp-0Ch], 0DDEEFF11h
ext:00414FD4      mov    dword ptr [ebp-50h], 977457FEh
ext:00414FDB      mov    dword ptr [ebp-4Ch], 0DA3E1880h
ext:00414FE2      mov    dword ptr [ebp-48h], 0B8169188h
ext:00414FE9      mov    dword ptr [ebp-44h], 1E95285Ch
ext:00414FF0      mov    dword ptr [ebp-40h], 1FE7E6F2h
ext:00414FF7      mov    dword ptr [ebp-3Ch], 2BC5FC57h
ext:00414FFE      mov    dword ptr [ebp-38h], 0B28F0FA8h
ext:00415005      mov    dword ptr [ebp-34h], 8E0E0644h
ext:0041500C      mov    dword ptr [ebp-30h], 6B454425h
ext:00415013      mov    dword ptr [ebp-2Ch], 0C57740D9h
ext:0041501A      xor    eax, eax
ext:0041501C      mov    [ebp-28h], eax
ext:0041501F      mov    [ebp-24h], eax
ext:00415022      mov    dword ptr [ebp-5Ch], 0
ext:00415029      push   64h ; 'd'
ext:0041502B      push   0
ext:0041502D      lea    eax, [ebp-0C8h]
ext:00415033      push   eax
ext:00415034      call   j_memset
ext:00415039      add    esp, 0Ch
ext:0041503C      push   offset unk_41DC74
ext:00415041      mov    eax, ds::?cout@std@@3V?$basic_ostream@DU?$char_traits@D@std@@@1@A ; std::ostream std::cout
ext:00415046      push   eax
ext:00415047      call   sub_41123A
ext:0041504C      add    esp, 8
ext:0041504F      lea    eax, [ebp-0C8h]
ext:00415055      push   eax
ext:00415056      mov    ecx, ds::?cin@std@@3V?$basic_istream@DU?$char_traits@D@std@@@1@A ; std::istream std::cin
ext:0041505C      push   ecx
ext:0041505D      call   sub_4110C8
ext:00415062      add    esp, 8
ext:00415065      jz    short near ptr loc_415069+1
ext:00415067      jnz   short near ptr loc_415069+1
ext:00415069      loc_415069:           ; CODE XREF: .text:00415065!j
ext:00415069          ; .text:00415067!j
J43B8 00414FB8: .text:00414FB8 (Synchronized with Hex View-1)

```

2,还是动态调试把花指令nop掉就行了