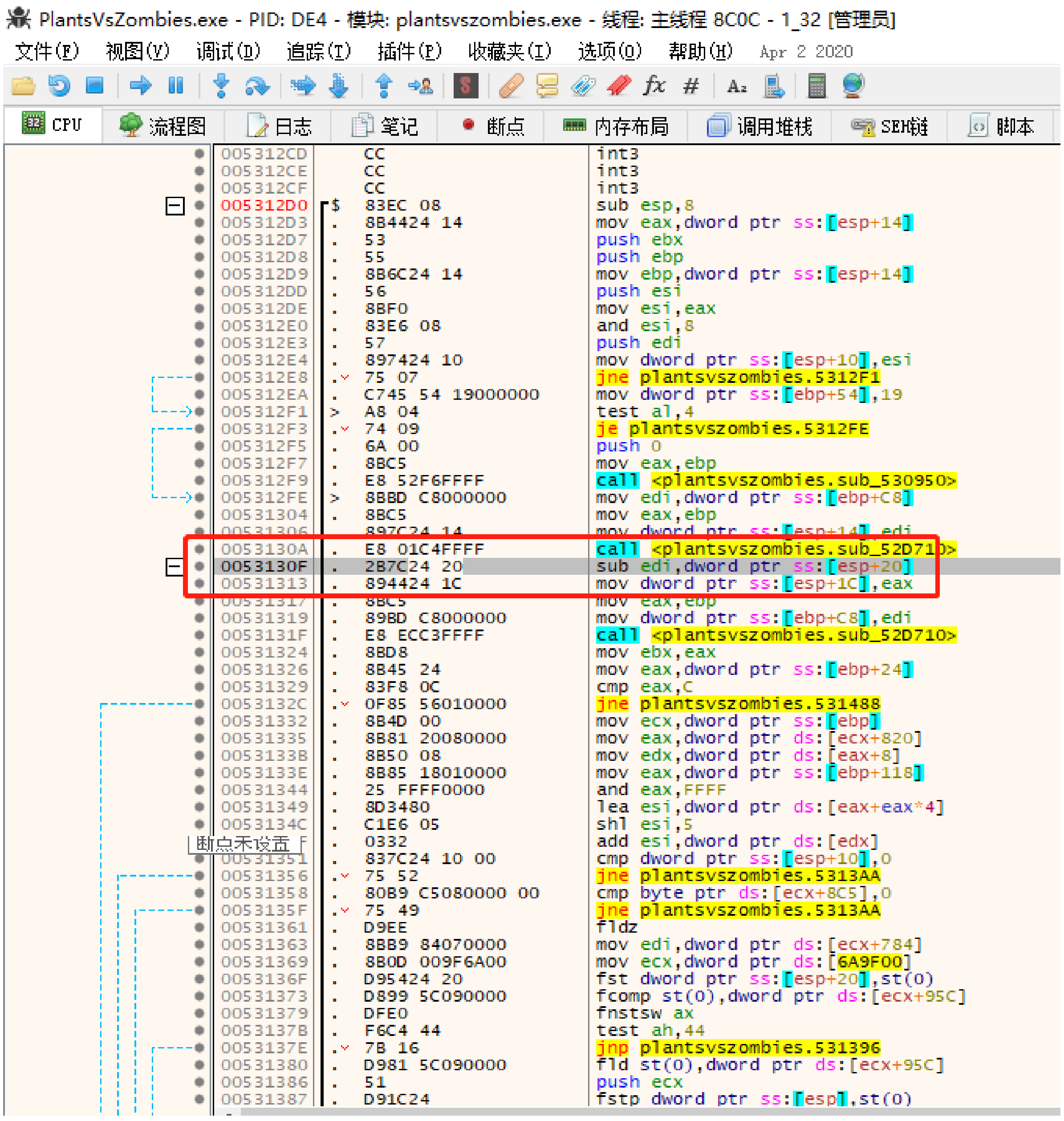
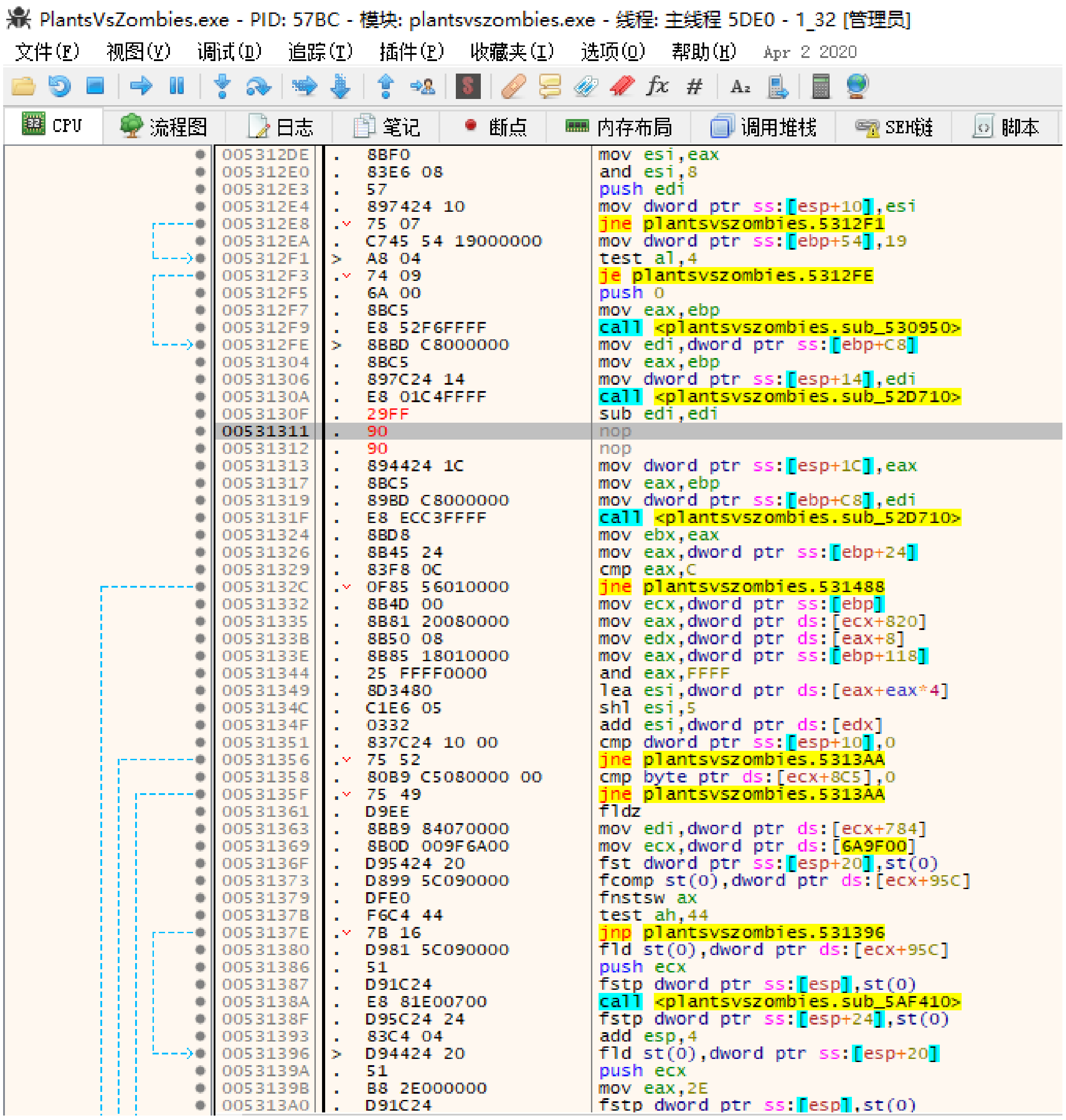
秒杀普通僵尸

```
EAX=1D910C84
EBX=00000000
ECX=000000B4
EDX=000000B4
ESI=00000000
EDI=00000046
EBP=1D910C84
ESP=0019F910
EIP=0053131F

指针基址可能是 = 1D910C84

00531313 - mov [esp+1C],eax
00531317 - mov eax,ebp
00531319 - mov [ebp+000000C8],edi
0053131F - call 0052D710
00531324 - mov ebx,eax
```



PlantsVsZombies.exe - PID: DE4 - 模块: plantsvszombies.exe - 线程: 主线程 8C0C - 1_32 [管理员]
文件(F) 视图(V) 调试(D) 追踪(T) 插件(P) 收藏夹(I) 选项(O) 帮助(H) Apr 2 2020
CPU 流程图 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本

```
005312CD      CC                      int3
005312CE      CC                      int3
005312CF      CC                      int3
005312D0   ┌$ 83EC 08                 sub esp,8
005312D3   .  8B4424 14               mov eax,dword ptr ss:[esp+14]
005312D7   .  53                      push ebx
005312D8   .  55                      push ebp
005312D9   .  8B6C24 14               mov ebp,dword ptr ss:[esp+14]
005312DD   .  56                      push esi
005312DE   .  8BF0                    mov esi,eax
005312E0   .  83E6 08                 and esi,8
005312E3   .  57                      push edi
005312E4   .  897424 10               mov dword ptr ss:[esp+10],esi
005312E8   .v 75 07                   jne plantsvszombies.5312F1
005312EA   .  C745 54 19000000        mov dword ptr ss:[ebp+54],19
005312F1   >  A8 04                   test al,4
005312F3   .v 74 09                   je plantsvszombies.5312FE
005312F5   .  6A 00                   push 0
005312F7   .  8BC5                    mov eax,ebp
005312F9   .  E8 52F6FFFF             call <plantsvszombies.sub_530950>
005312FE   >  8BBD C8000000           mov edi,dword ptr ss:[ebp+C8]
00531304   .  8BC5                    mov eax,ebp
00531306   .  897C24 14               mov dword ptr ss:[esp+14],edi
0053130A   .  E8 01C4FFFF             call <plantsvszombies.sub_52D710>
0053130F   .  2B7C24 20               sub edi,dword ptr ss:[esp+20]
00531313   .  894424 1C               mov dword ptr ss:[esp+1C],eax
00531317   .  8BC5                    mov eax,ebp
00531319   .  89BD C8000000           mov dword ptr ss:[ebp+C8],edi
0053131F   .  E8 ECC3FFFF             call <plantsvszombies.sub_52D710>
00531324   .  8BD8                    mov ebx,eax
00531326   .  8B45 24                 mov eax,dword ptr ss:[ebp+24]
00531329   .  83F8 0C                 cmp eax,C
0053132C   .v 0F85 56010000           jne plantsvszombies.531488
00531332   .  8B4D 00                 mov ecx,dword ptr ss:[ebp]
00531335   .  8B81 20080000           mov eax,dword ptr ds:[ecx+820]
0053133B   .  8B50 08                 mov edx,dword ptr ds:[eax+8]
0053133E   .  8B85 18010000           mov eax,dword ptr ss:[ebp+118]
00531344   .  25 FFFF0000             and eax,FFFF
00531349   .  8D3480                  lea esi,dword ptr ds:[eax+eax*4]
0053134C   .  C1E6 05                 shl esi,5
断点未设置 F .  0332                    add esi,dword ptr ds:[edx]
00531351   .  837C24 10 00            cmp dword ptr ss:[esp+10],0
00531356   .v 75 52                   jne plantsvszombies.5313AA
00531358   .  80B9 C5080000 00        cmp byte ptr ds:[ecx+8C5],0
0053135F   .v 75 49                   jne plantsvszombies.5313AA
00531361   .  D9EE                    fldz
00531363   .  8BB9 84070000           mov edi,dword ptr ds:[ecx+784]
00531369   .  8B0D 009F6A00           mov ecx,dword ptr ds:[6A9F00]
0053136F   .  D95424 20               fst dword ptr ss:[esp+20],st(0)
00531373   .  D899 5C090000           fcomp st(0),dword ptr ds:[ecx+95C]
00531379   .  DFE0                    fnstsw ax
0053137B   .  F6C4 44                 test ah,44
0053137E   .v 7B 16                   jnp plantsvszombies.531396
00531380   .  D981 5C090000           fld st(0),dword ptr ds:[ecx+95C]
00531386   .  51                      push ecx
00531387   .  D91C24                  fstp dword ptr ss:[esp],st(0)
```


PlantsVsZombies.exe - PID: 57BC - 模块: plantsvszombies.exe - 线程: 主线程 5DE0 - 1_32 [管理员]
文件(F) 视图(V) 调试(D) 追踪(T) 插件(P) 收藏夹(I) 选项(O) 帮助(H) Apr 2 2020
CPU 流程图 日志 笔记 断点 内存布局 调用堆栈 SEH链 脚本

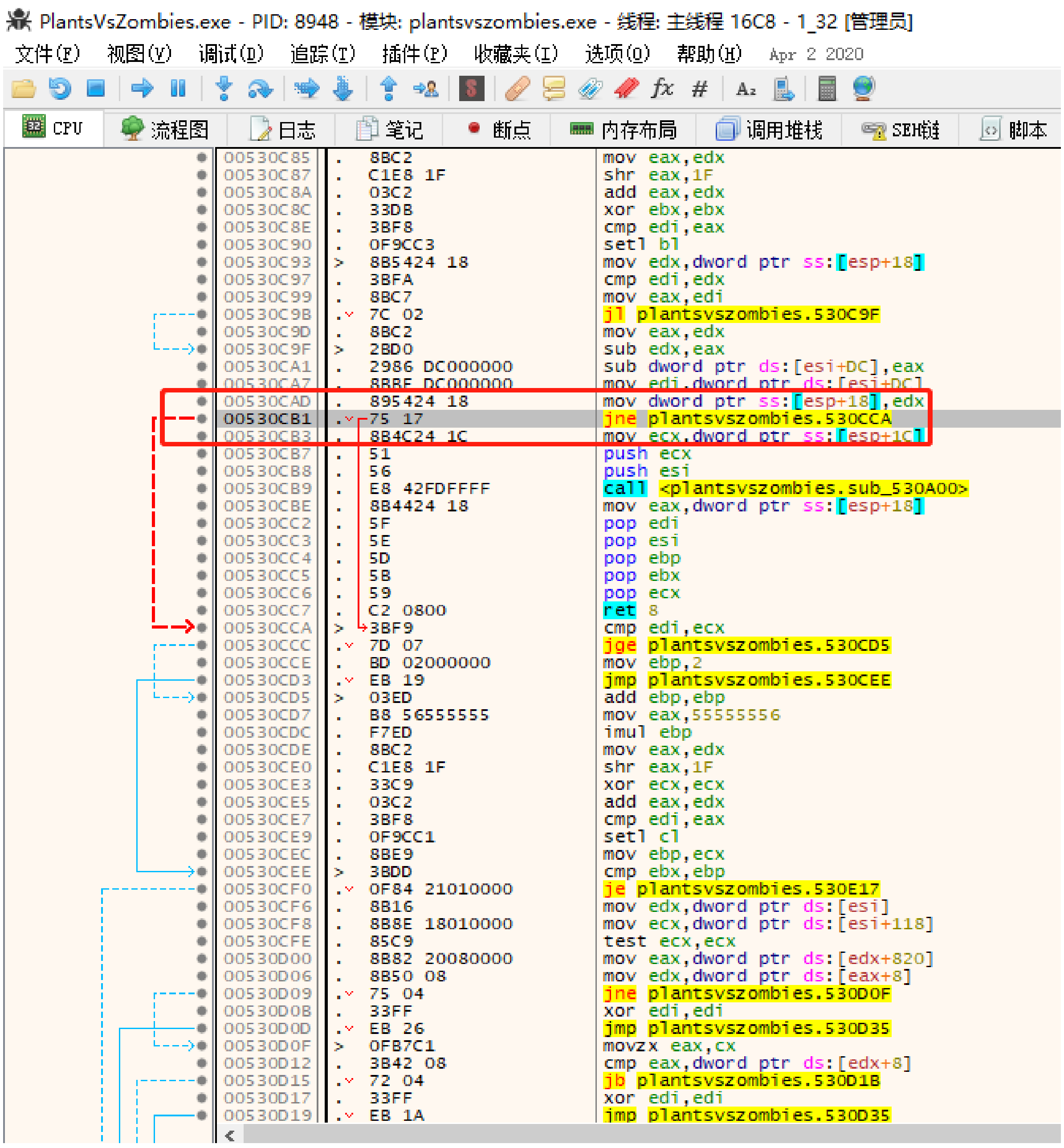
```
005312DE  .   8BF0                 mov esi,eax
005312E0  .   83E6 08              and esi,8
005312E3  .   57                   push edi
005312E4  .   897424 10            mov dword ptr ss:[esp+10],esi
005312E8  .v  75 07                jne plantsvszombies.5312F1
005312EA  .   C745 54 19000000     mov dword ptr ss:[ebp+54],19
005312F1  >   A8 04                test al,4
005312F3  .v  74 09                je plantsvszombies.5312FE
005312F5  .   6A 00                push 0
005312F7  .   8BC5                 mov eax,ebp
005312F9  .   E8 52F6FFFF          call <plantsvszombies.sub_530950>
005312FE  >   8BBD C8000000        mov edi,dword ptr ss:[ebp+C8]
00531304  .   8BC5                 mov eax,ebp
00531306  .   897C24 14            mov dword ptr ss:[esp+14],edi
0053130A  .   E8 01C4FFFF          call <plantsvszombies.sub_52D710>
0053130F  .   29FF                 sub edi,edi
00531311  .   90                   nop
00531312  .   90                   nop
00531313  .   894424 1C            mov dword ptr ss:[esp+1C],eax
00531317  .   8BC5                 mov eax,ebp
00531319  .   89BD C8000000        mov dword ptr ss:[ebp+C8],edi
0053131F  .   E8 ECC3FFFF          call <plantsvszombies.sub_52D710>
00531324  .   8BD8                 mov ebx,eax
00531326  .   8B45 24              mov eax,dword ptr ss:[ebp+24]
00531329  .   83F8 0C              cmp eax,C
0053132C  .v  0F85 56010000        jne plantsvszombies.531488
00531332  .   8B4D 00              mov ecx,dword ptr ss:[ebp]
00531335  .   8B81 20080000        mov eax,dword ptr ds:[ecx+820]
0053133B  .   8B50 08              mov edx,dword ptr ds:[eax+8]
0053133E  .   8B85 18010000        mov eax,dword ptr ss:[ebp+118]
00531344  .   25 FFFF0000          and eax,FFFF
00531349  .   8D3480               lea esi,dword ptr ds:[eax+eax*4]
0053134C  .   C1E6 05              shl esi,5
0053134F  .   0332                 add esi,dword ptr ds:[edx]
00531351  .   837C24 10 00         cmp dword ptr ss:[esp+10],0
00531356  .v  75 52                jne plantsvszombies.5313AA
00531358  .   80B9 C5080000 00     cmp byte ptr ds:[ecx+8C5],0
0053135F  .v  75 49                jne plantsvszombies.5313AA
00531361  .   D9EE                 fldz
00531363  .   8BB9 84070000        mov edi,dword ptr ds:[ecx+784]
00531369  .   8B0D 009F6A00        mov ecx,dword ptr ds:[6A9F00]
0053136F  .   D95424 20            fst dword ptr ss:[esp+20],st(0)
00531373  .   D899 5C090000        fcomp st(0),dword ptr ds:[ecx+95C]
00531379  .   DFE0                 fnstsw ax
0053137B  .   F6C4 44              test ah,44
0053137E  .v  7B 16                jnp plantsvszombies.531396
00531380  .   D981 5C090000        fld st(0),dword ptr ds:[ecx+95C]
00531386  .   51                   push ecx
00531387  .   D91C24               fstp dword ptr ss:[esp],st(0)
0053138A  .   E8 81E00700          call <plantsvszombies.sub_5AF410>
0053138F  .   D95C24 24            fstp dword ptr ss:[esp+24],st(0)
00531393  .   83C4 04              add esp,4
00531396  >   D94424 20            fld st(0),dword ptr ss:[esp+20]
0053139A  .   51                   push ecx
0053139B  .   B8 2E000000          mov eax,2E
005313A0  .   D91C24               fstp dword ptr ss:[esp],st(0)
```

秒杀戴帽子僵尸

```
EAX=00000014
EBX=00000000
ECX=0000006E
EDX=00000001
ESI=00000000
EDI=00000172
EBP=19996960
ESP=0019F920
EIP=00531053

指针基址可能是 = 19996960

00531046 - test bl,04
00531049 - mov [esp+0C],esi
0053104D - mov [ebp+000000D0],ecx
00531053 - je 0053105E
00531055 - push 00
```

秒杀铁丝网僵尸

```
EAX=00000014
EBX=00000000
ECX=0000016E
EDX=00000000
ESI=167933F8
EDI=000002F8
EBP=0000044C
ESP=0019F918
EIP=00530CA7

指针基址可能是 = 167933F8

00530C9D - mov eax,edx
00530C9F - sub edx,eax
00530CA1 - sub [esi+000000DC],eax
00530CA7 - mov edi,[esi+000000DC]
00530CAD - mov [esp+18],edx
```

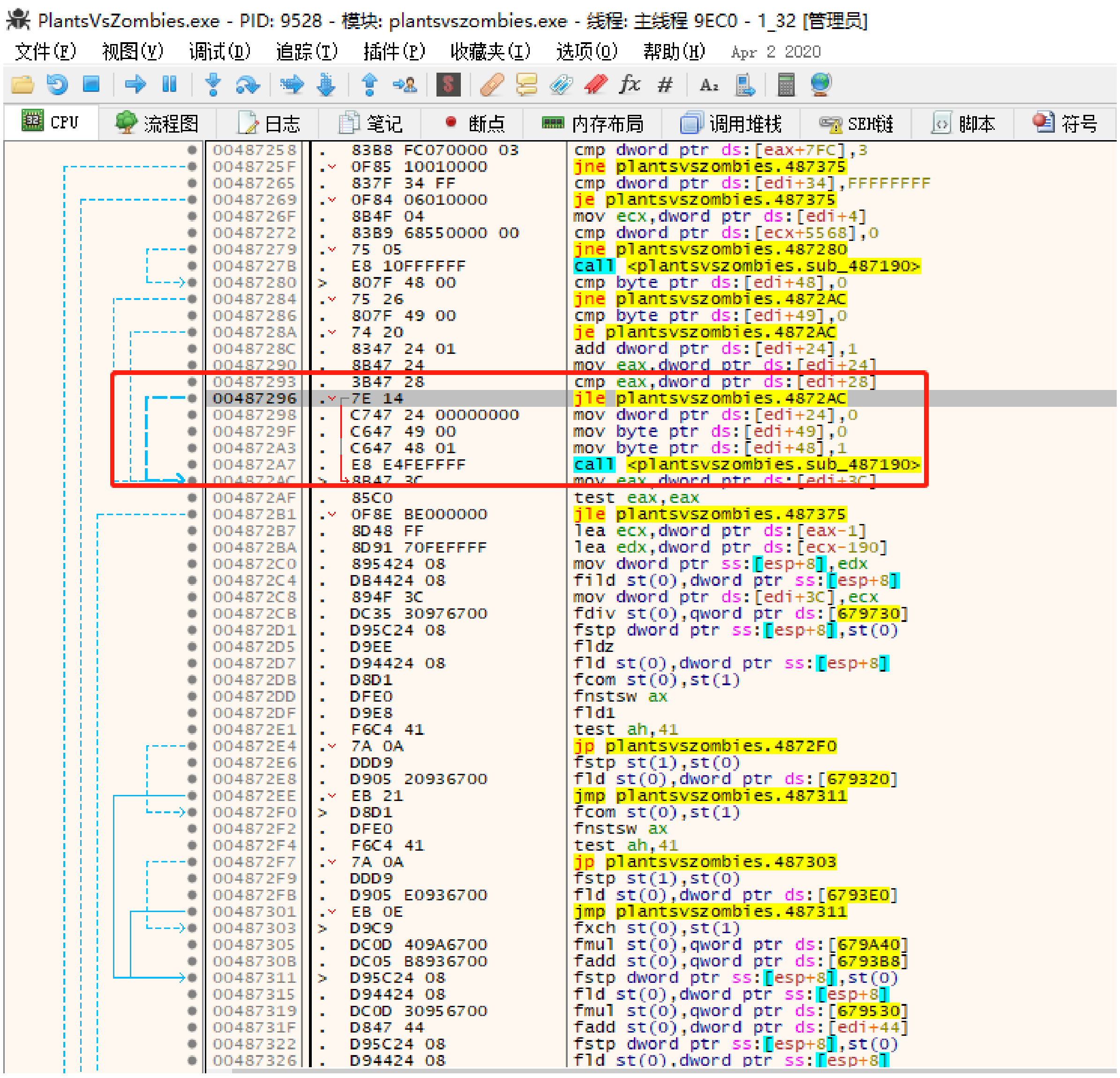


PlantsVsZombies.exe - PID: 8948 - 模块: plantsvszombies.exe - 线程: 主线程 16C8 - 1_32 [管理员]

```
00530C85 .   8BC2           mov eax,edx
00530C87 .   C1E8 1F        shr eax,1F
00530C8A .   03C2           add eax,edx
00530C8C .   33DB           xor ebx,ebx
00530C8E .   3BF8           cmp edi,eax
00530C90 .   0F9CC3         setl bl
00530C93 >   8B5424 18      mov edx,dword ptr ss:[esp+18]
00530C97 .   3BFA           cmp edi,edx
00530C99 .   8BC7           mov eax,edi
00530C9B .v  7C 02          jl plantsvszombies.530C9F
00530C9D .   8BC2           mov eax,edx
00530C9F >   2BD0           sub edx,eax
00530CA1 .   2986 DC000000  sub dword ptr ds:[esi+DC],eax
00530CA7 .   8BBE DC000000  mov edi,dword ptr ds:[esi+DC]
00530CAD .   895424 18      mov dword ptr ss:[esp+18],edx
00530CB1 .v  75 17          jne plantsvszombies.530CCA
00530CB3 .   8B4C24 1C      mov ecx,dword ptr ss:[esp+1C]
00530CB7 .   51             push ecx
00530CB8 .   56             push esi
00530CB9 .   E8 42FDFFFF    call <plantsvszombies.sub_530A00>
00530CBE .   8B4424 18      mov eax,dword ptr ss:[esp+18]
00530CC2 .   5F             pop edi
00530CC3 .   5E             pop esi
00530CC4 .   5D             pop ebp
00530CC5 .   5B             pop ebx
00530CC6 .   59             pop ecx
00530CC7 .   C2 0800        ret 8
00530CCA >   3BF9           cmp edi,ecx
00530CCC .v  7D 07          jge plantsvszombies.530CD5
00530CCE .   BD 02000000    mov ebp,2
00530CD3 .v  EB 19          jmp plantsvszombies.530CEE
00530CD5 >   03ED           add ebp,ebp
00530CD7 .   B8 56555555    mov eax,55555556
00530CDC .   F7ED           imul ebp
00530CDE .   8BC2           mov eax,edx
00530CE0 .   C1E8 1F        shr eax,1F
00530CE3 .   33C9           xor ecx,ecx
00530CE5 .   03C2           add eax,edx
00530CE7 .   3BF8           cmp edi,eax
00530CE9 .   0F9CC1         setl cl
00530CEC .   8BE9           mov ebp,ecx
00530CEE >   3BDD           cmp ebx,ebp
00530CF0 .v  0F84 21010000  je plantsvszombies.530E17
00530CF6 .   8B16           mov edx,dword ptr ds:[esi]
00530CF8 .   8B8E 18010000  mov ecx,dword ptr ds:[esi+118]
00530CFE .   85C9           test ecx,ecx
00530D00 .   8B82 20080000  mov eax,dword ptr ds:[edx+820]
00530D06 .   8B50 08        mov edx,dword ptr ds:[eax+8]
00530D09 .v  75 04          jne plantsvszombies.530D0F
00530D0B .   33FF           xor edi,edi
00530D0D .v  EB 26          jmp plantsvszombies.530D35
00530D0F >   0FB7C1         movzx eax,cx
00530D12 .   3B42 08        cmp eax,dword ptr ds:[edx+8]
00530D15 .v  72 04          jb plantsvszombies.530D1B
00530D17 .   33FF           xor edi,edi
00530D19 .v  EB 1A          jmp plantsvszombies.530D35
```

植物安放无CD

```
EAX=02879CF0
EBX=1A482230
ECX=1A482230
EDX=0287A848
ESI=00000000
EDI=17050E58
EBP=00000000
ESP=0019FA00
EIP=00487290

指针基址可能是 = 17050E58

00487286 - cmp byte ptr [edi+49],00
0048728A - je 004872AC
0048728C - add dword ptr [edi+24],01
00487290 - mov eax,[edi+24]
00487293 - cmp eax,[edi+28]
```



PlantsVsZombies.exe - PID: 9528 - 模块: plantsvszombies.exe - 线程: 主线程 9EC0 - 1_32 [管理员]

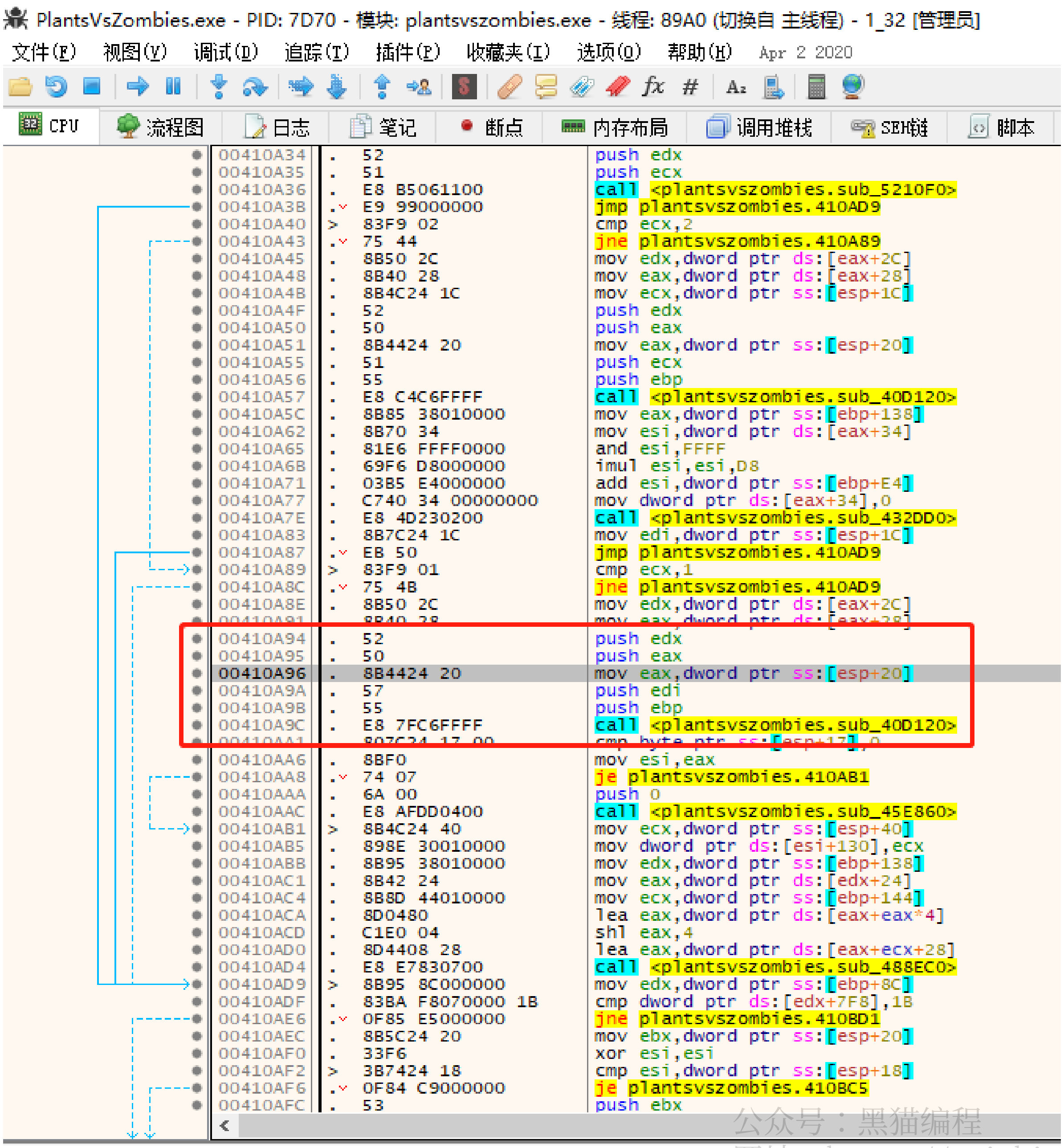
```
00487258 .  83B8 FC070000 03  cmp dword ptr ds:[eax+7FC],3
0048725F .v 0F85 10010000     jne plantsvszombies.487375
00487265 .  837F 34 FF        cmp dword ptr ds:[edi+34],FFFFFFFF
00487269 .v 0F84 06010000     je plantsvszombies.487375
0048726F .  8B4F 04           mov ecx,dword ptr ds:[edi+4]
00487272 .  83B9 68550000 00  cmp dword ptr ds:[ecx+5568],0
00487279 .v 75 05             jne plantsvszombies.487280
0048727B .  E8 10FFFFFF       call <plantsvszombies.sub_487190>
00487280 >  807F 48 00        cmp byte ptr ds:[edi+48],0
00487284 .v 75 26             jne plantsvszombies.4872AC
00487286 .  807F 49 00        cmp byte ptr ds:[edi+49],0
0048728A .v 74 20             je plantsvszombies.4872AC
0048728C .  8347 24 01        add dword ptr ds:[edi+24],1
00487290 .  8B47 24           mov eax,dword ptr ds:[edi+24]
00487293 .  3B47 28           cmp eax,dword ptr ds:[edi+28]
00487296 .v 7E 14             jle plantsvszombies.4872AC
00487298 .  C747 24 00000000  mov dword ptr ds:[edi+24],0
0048729F .  C647 49 00        mov byte ptr ds:[edi+49],0
004872A3 .  C647 48 01        mov byte ptr ds:[edi+48],1
004872A7 .  E8 E4FEFFFF       call <plantsvszombies.sub_487190>
004872AC >  8B47 3C           mov eax,dword ptr ds:[edi+3C]
004872AF .  85C0              test eax,eax
004872B1 .v 0F8E BE000000     jle plantsvszombies.487375
004872B7 .  8D48 FF           lea ecx,dword ptr ds:[eax-1]
004872BA .  8D91 70FEFFFF     lea edx,dword ptr ds:[ecx-190]
004872C0 .  895424 08         mov dword ptr ss:[esp+8],edx
004872C4 .  DB4424 08         fild st(0),dword ptr ss:[esp+8]
004872C8 .  894F 3C           mov dword ptr ds:[edi+3C],ecx
004872CB .  DC35 30976700     fdiv st(0),qword ptr ds:[679730]
004872D1 .  D95C24 08         fstp dword ptr ss:[esp+8],st(0)
004872D5 .  D9EE              fldz
004872D7 .  D94424 08         fld st(0),dword ptr ss:[esp+8]
004872DB .  D8D1              fcom st(0),st(1)
004872DD .  DFE0              fnstsw ax
004872DF .  D9E8              fld1
004872E1 .  F6C4 41           test ah,41
004872E4 .v 7A 0A             jp plantsvszombies.4872F0
004872E6 .  DDD9              fstp st(1),st(0)
004872E8 .  D905 20936700     fld st(0),dword ptr ds:[679320]
004872EE .v EB 21             jmp plantsvszombies.487311
004872F0 >  D8D1              fcom st(0),st(1)
004872F2 .  DFE0              fnstsw ax
004872F4 .  F6C4 41           test ah,41
004872F7 .v 7A 0A             jp plantsvszombies.487303
004872F9 .  DDD9              fstp st(1),st(0)
004872FB .  D905 E0936700     fld st(0),dword ptr ds:[6793E0]
00487301 .v EB 0E             jmp plantsvszombies.487311
00487303 >  D9C9              fxch st(0),st(1)
00487305 .  DC0D 409A6700     fmul st(0),qword ptr ds:[679A40]
0048730B .  DC05 B8936700     fadd st(0),qword ptr ds:[6793B8]
00487311 >  D95C24 08         fstp dword ptr ss:[esp+8],st(0)
00487315 .  D94424 08         fld st(0),dword ptr ss:[esp+8]
00487319 .  DC0D 30956700     fmul st(0),qword ptr ds:[679530]
0048731F .  D847 44           fadd st(0),dword ptr ds:[edi+44]
00487322 .  D95C24 08         fstp dword ptr ss:[esp+8],st(0)
00487326 .  D94424 08         fld st(0),dword ptr ss:[esp+8]
```

大嘴花吞噬无CD

```
EAX=000009D9
EBX=167E2B80
ECX=00000006
EDX=02889CF0
ESI=0019FA20
EDI=167E2B80
EBP=00000000
ESP=0019F9F8
EIP=00463252

指针基址可能是 = 167E2B80

0046324A - jle 00463252
0046324C - add eax,-01
0046324F - mov [edi+54],eax
00463252 - mov ecx,[edi]
00463254 - call 00453840
```

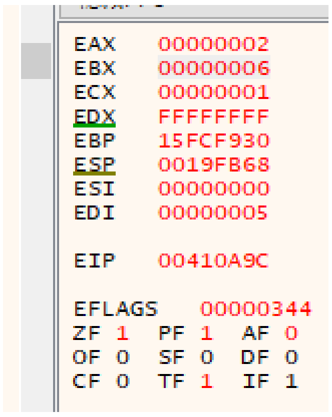
植物安放call

```
EAX=00000006
EBX=00000002
ECX=00000000
EDX=16030010
ESI=16BBA284
EDI=00000006
EBP=16B5D6E0
ESP=0019FB78
EIP=00410AC4

指针基址可能是 = 16030010

00410AB5 - mov [esi+00000130],ecx
00410ABB - mov edx,[ebp+00000138]
00410AC1 - mov eax,[edx+24]
00410AC4 - mov ecx,[ebp+00000144]
00410ACA - lea eax,[eax+eax*4]
```



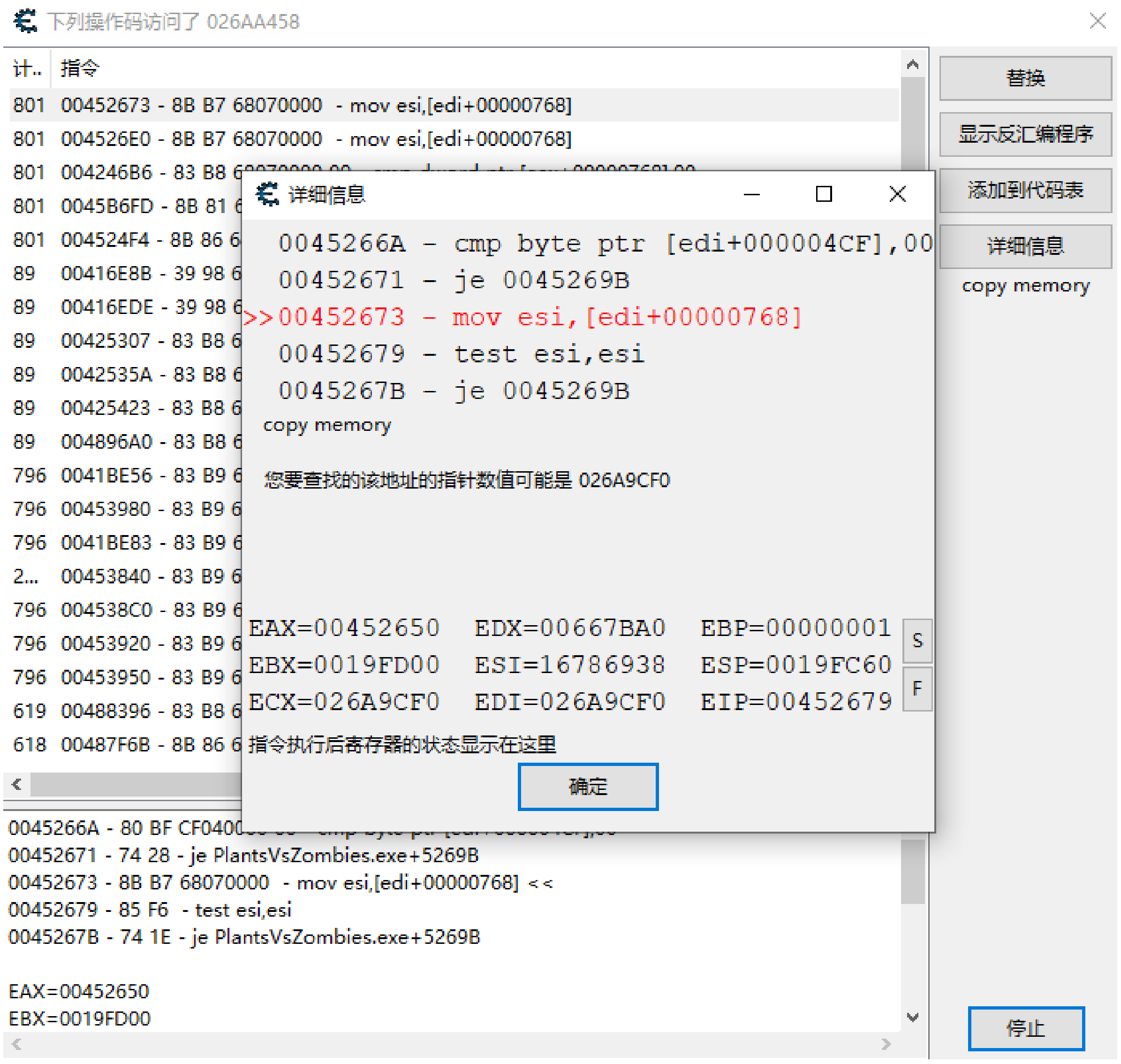
PlantsVsZombies.exe - PID: 7D70 - 模块: plantsvszombies.exe - 线程: 89A0 (切换自 主线程) - 1_32 [管理员]

```
00410A34  .   52              push edx
00410A35  .   51              push ecx
00410A36  .   E8 B5061100     call <plantsvszombies.sub_5210F0>
00410A3B  .v  E9 99000000     jmp plantsvszombies.410AD9
00410A40  >   83F9 02         cmp ecx,2
00410A43  .v  75 44           jne plantsvszombies.410A89
00410A45  .   8B50 2C         mov edx,dword ptr ds:[eax+2C]
00410A48  .   8B40 28         mov eax,dword ptr ds:[eax+28]
00410A4B  .   8B4C24 1C       mov ecx,dword ptr ss:[esp+1C]
00410A4F  .   52              push edx
00410A50  .   50              push eax
00410A51  .   8B4424 20       mov eax,dword ptr ss:[esp+20]
00410A55  .   51              push ecx
00410A56  .   55              push ebp
00410A57  .   E8 C4C6FFFF     call <plantsvszombies.sub_40D120>
00410A5C  .   8B85 38010000   mov eax,dword ptr ss:[ebp+138]
00410A62  .   8B70 34         mov esi,dword ptr ds:[eax+34]
00410A65  .   81E6 FFFF0000   and esi,FFFF
00410A6B  .   69F6 D8000000   imul esi,esi,D8
00410A71  .   03B5 E4000000   add esi,dword ptr ss:[ebp+E4]
00410A77  .   C740 34 00000000 mov dword ptr ds:[eax+34],0
00410A7E  .   E8 4D230200     call <plantsvszombies.sub_432DD0>
00410A83  .   8B7C24 1C       mov edi,dword ptr ss:[esp+1C]
00410A87  .v  EB 50           jmp plantsvszombies.410AD9
00410A89  >   83F9 01         cmp ecx,1
00410A8C  .v  75 4B           jne plantsvszombies.410AD9
00410A8E  .   8B50 2C         mov edx,dword ptr ds:[eax+2C]
00410A91  .   8B40 28         mov eax,dword ptr ds:[eax+28]
00410A94  .   52              push edx
00410A95  .   50              push eax
00410A96  .   8B4424 20       mov eax,dword ptr ss:[esp+20]
00410A9A  .   57              push edi
00410A9B  .   55              push ebp
00410A9C  .   E8 7FC6FFFF     call <plantsvszombies.sub_40D120>
00410AA1  .   807C24 17 00    cmp byte ptr ss:[esp+17],0
00410AA6  .   8BF0            mov esi,eax
00410AA8  .v  74 07           je plantsvszombies.410AB1
00410AAA  .   6A 00           push 0
00410AAC  .   E8 AFDD0400     call <plantsvszombies.sub_45E860>
00410AB1  >   8B4C24 40       mov ecx,dword ptr ss:[esp+40]
00410AB5  .   898E 30010000   mov dword ptr ds:[esi+130],ecx
00410ABB  .   8B95 38010000   mov edx,dword ptr ss:[ebp+138]
00410AC1  .   8B42 24         mov eax,dword ptr ds:[edx+24]
00410AC4  .   8B8D 44010000   mov ecx,dword ptr ss:[ebp+144]
00410ACA  .   8D0480          lea eax,dword ptr ds:[eax+eax*4]
00410ACD  .   C1E0 04         shl eax,4
00410AD0  .   8D4408 28       lea eax,dword ptr ds:[eax+ecx+28]
00410AD4  .   E8 E7830700     call <plantsvszombies.sub_488EC0>
00410AD9  >   8B95 8C000000   mov edx,dword ptr ss:[ebp+8C]
00410ADF  .   83BA F8070000 1B cmp dword ptr ds:[edx+7F8],1B
00410AE6  .v  0F85 E5000000   jne plantsvszombies.410BD1
00410AEC  .   8B5C24 20       mov ebx,dword ptr ss:[esp+20]
00410AF0  .   33F6            xor esi,esi
00410AF2  >   3B7424 18       cmp esi,dword ptr ss:[esp+18]
00410AF6  .v  0F84 C9000000   je plantsvszombies.410BC5
00410AFC  .   53              push ebx
```



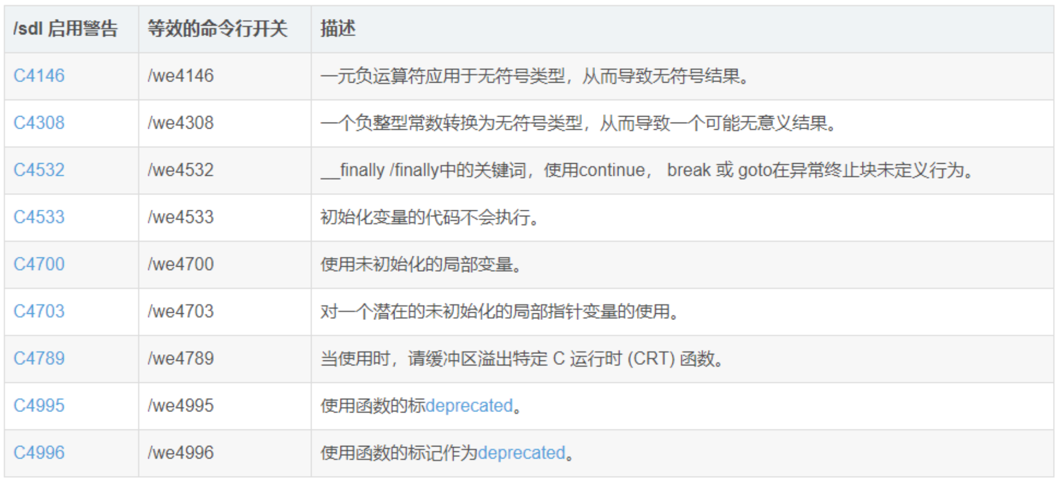
```
EAX     00000002
EBX     00000006
ECX     00000001
EDX     FFFFFFFF
EBP     15FCF930
ESP     0019FB68
ESI     00000000
EDI     00000005

EIP     00410A9C

EFLAGS     00000344
ZF  1   PF  1   AF  0
OF  0   SF  0   DF  0
CF  0   TF  1   IF  1
```

```
00410A94 | 52                                   | push edx
                                                              |
00410A95 | 50                                   | push eax
                                                              |
00410A96 | 8B4424 20                            | mov eax,dword ptr ss:[esp+20]
                                                               |
00410A9A | 57                                   | push edi
                                                              |
00410A9B | 55                                   | push ebp
                                                              |
00410A9C | E8 7FC6FFFF                          | call <plantsvszombies.sub_40D120>

EBP是变化的，需要寻找基址

push -1
push 1
mov eax, 1
push 2
mov ebx, ds:[0x6A9EC0]
mov ebx, ds:[ebx+0x768]
push ebx
mov edx,0x40D120
call edx

__asm {
    pushad
    push -1
    push 2
    mov eax, 0
    push 2
    mov ebx, ds:[0x6A9EC0]
    mov ebx, ds:[ebx+0x768]
    push ebx
    mov edx,0x40D120
    call edx
    popad
    ret
}
```



```
下列操作码访问了 026AA458

计..  指令
801  00452673 - 8B B7 68070000  - mov esi,[edi+00000768]
801  004526E0 - 8B B7 68070000  - mov esi,[edi+00000768]
801  004246B6 - 83 B8 6
801  0045B6FD - 8B 81 6
801  004524F4 - 8B 86 6
89   00416E8B - 39 98 6
89   00416EDE - 39 98 6
89   00425307 - 83 B8 6
89   0042535A - 83 B8 6
89   00425423 - 83 B8 6
89   004896A0 - 83 B8 6
796  0041BE56 - 83 B9 6
796  00453980 - 83 B9 6
796  0041BE83 - 83 B9 6
2... 00453840 - 83 B9 6
796  004538C0 - 83 B9 6
796  00453920 - 83 B9 6
796  00453950 - 83 B9 6
619  00488396 - 83 B8 6
618  00487F6B - 8B 86 6

详细信息
  0045266A - cmp byte ptr [edi+000004CF],00
  00452671 - je 0045269B
>>00452673 - mov esi,[edi+00000768]
  00452679 - test esi,esi
  0045267B - je 0045269B
copy memory

您要查找的该地址的指针数值可能是 026A9CF0

EAX=00452650  EDX=00667BA0  EBP=00000001
EBX=0019FD00  ESI=16786938  ESP=0019FC60
ECX=026A9CF0  EDI=026A9CF0  EIP=00452679
指令执行后寄存器的状态显示在这里

0045266A - 80 BF CF0400
00452671 - 74 28 - je PlantsVsZombies.exe+5269B
00452673 - 8B B7 68070000  - mov esi,[edi+00000768] <<
00452679 - 85 F6  - test esi,esi
0045267B - 74 1E - je PlantsVsZombies.exe+5269B

EAX=00452650
EBX=0019FD00
```

- 注入时一定注意切换为Release模式
    - Debug通常称为调试版本，通过一系列编译选项的配合，编译结果通常包含调试信息，而且不做任何优化，以为开发人员提供强大的应用程序调试能力。但是注入时也包含很多额外信息，进而导致注入失败。
    - Release通常称为发布版本，是为用户使用的，一般客户不允许在发布版本上进行调试，所以不保存调试信息，同时它往往进行了各种优化，以期达到代码最小和速度最优，为用户的使用提供便利。
- 关闭SDL检查
    - SDL检查也叫做`安全开发生命周期检查`，是微软在VS2012推出的，为了能更好的监管开发者的代码安全，如果勾选上这一项，那么他将严格按照SDL的规则编译代码，会有一些以前常用的函数无法通过编译，比如在VS2010中的scanf是warning那么在VS2012中就是error。

◦ 

/sdl 启用警告	等效的命令行开关	描述
C4146	/we4146	一元负运算符应用于无符号类型，从而导致无符号结果。
C4308	/we4308	一个负整型常数转换为无符号类型，从而导致一个可能无意义结果。
C4532	/we4532	__finally /finally中的关键词，使用continue， break 或 goto在异常终止块未定义行为。
C4533	/we4533	初始化变量的代码不会执行。
C4700	/we4700	使用未初始化的局部变量。
C4703	/we4703	对一个潜在的未初始化的局部指针变量的使用。
C4789	/we4789	当使用时，请缓冲区溢出特定 C 运行时 (CRT) 函数。
C4995	/we4995	使用函数的标deprecated。
C4996	/we4996	使用函数的标记作为deprecated。

源码展示

```cpp
HANDLE g_process_handle;    // 游戏进程句柄
HANDLE g_monitor_thread;
BOOL is_collect_sun;

// 向指定内存写入数据
void WriteMemory(HANDLE hProcess, void* value, DWORD valueSize, ...) {
    if (value == NULL || valueSize == 0 || hProcess == NULL) return;

    DWORD tempValue = 0;

    va_list addresses;
    va_start(addresses, valueSize);
    DWORD offset = 0;
    DWORD lastAddress = 0;
    while ((offset = va_arg(addresses, DWORD)) != -1) {
        lastAddress = tempValue + offset;
        ::ReadProcessMemory(hProcess, (LPCVOID)lastAddress, &tempValue, sizeof(DWORD), NULL);
    }
    va_end(addresses);

    ::WriteProcessMemory(hProcess, (LPVOID)lastAddress, value, valueSize, NULL);
}

void WriteMemory(HANDLE hProcess, void* value, DWORD valueSize, DWORD address) {
    WriteMemory(hProcess, value, valueSize, address, -1);
}

// 线程函数
DWORD WINAPI monitorThreadProc(LPVOID lpParameter) {
    while (TRUE) {
        HWND game_hwnd = ::FindWindowA(NULL, "植物大战僵尸中文版");
        if (!game_hwnd) {
            ::MessageBoxA(NULL, "植物大战僵尸游戏未打开", "错误", MB_OK);
        }
        else if (!g_process_handle) {
            DWORD pid;
            ::GetWindowThreadProcessId(game_hwnd, &pid);
            g_process_handle = ::OpenProcess(PROCESS_ALL_ACCESS, NULL, pid);
        }

        WriteMemory(g_process_handle, &is_collect_sun, sizeof(is_collect_sun), 0x6A9EC0, 0x768, 0xe4, 0x50, -1);

        ::Sleep(1000);
    }

    return 0;
}

// 提升权限函数
BOOL ImproveAccessPrivilege()
{
    HANDLE tokenHandle;
    LUID privilegeValue;

    if (!::OpenProcessToken(GetCurrentProcess(), TOKEN_ADJUST_PRIVILEGES | TOKEN_QUERY, &tokenHandle)) return FALSE;

    if (!LookupPrivilegeValue(NULL, SE_DEBUG_NAME, &privilegeValue))
    {
        ::CloseHandle(tokenHandle);
        return FALSE;
    }

    TOKEN_PRIVILEGES privileges;
    privileges.PrivilegeCount = 1;
    privileges.Privileges[0].Luid = privilegeValue;
    privileges.Privileges[0].Attributes = SE_PRIVILEGE_ENABLED;

    if (!::AdjustTokenPrivileges(tokenHandle, FALSE, &privileges, sizeof(privileges), NULL, NULL))
    {
        ::CloseHandle(tokenHandle);
        return FALSE;
    }

    return TRUE;
}
```

```cpp
ImproveAccessPrivilege();
g_monitor_thread = ::CreateThread(NULL, NULL, monitorThreadProc, NULL, NULL, NULL);

// 阳光初始值
m_edit_sun_value = "8000";
m_edit_money_value = "9999";
m_edit_plantX = "5";
m_edit_plantY = "4";

UpdateData(FALSE);
```

```
// 设置阳光值
void CPlantsVsZombiesWGDlg::OnBnClickedBtnSetSun()
{
    UpdateData(TRUE);

    int sun_value = _ttoi(m_edit_sun_value);
    WriteMemory(g_process_handle, &sun_value, sizeof(sun_value), 0x6A9EC0,
0x768, 0x5560, -1);
}


// 设置金币值
void CPlantsVsZombiesWGDlg::OnBnClickedBtnSetmoney()
{
    UpdateData(TRUE);
    int money_value = _ttoi(m_edit_money_value);
    WriteMemory(g_process_handle, &money_value, sizeof(money_value),
0x6A9EC0, 0x82C, 0x28, -1);
}


// 自动收集阳光
void CPlantsVsZombiesWGDlg::OnBnClickedCheckAutoCollect()
{
    if (m_check_auto_collect.GetCheck()) is_collect_sun = 1;
    // else is_collect_sun = 0;
}


// 秒杀僵尸
void CPlantsVsZombiesWGDlg::OnBnClickedCheckKill()
{
    DWORD address1 = 0x53130F;  // 普通僵尸
    DWORD address2 = 0x531044;  // 戴帽子僵尸
    DWORD address3 = 0x530CB1;  // 铁丝网僵尸

    if (m_check_kill.GetCheck()) {

        BYTE data1[] = { 0x29, 0xff, 0x90, 0x90 };
        WriteMemory(g_process_handle, data1, sizeof(data1), address1);

        BYTE data2[] = { 0x29, 0xc9 };
        WriteMemory(g_process_handle, data2, sizeof(data2), address2);

        BYTE data3[] = { 0x90, 0x90 };
        WriteMemory(g_process_handle, data3, sizeof(data3), address3);
        /*::WriteProcessMemory(g_process_handle, (LPVOID)0x53130F,
(LPCVOID)&data1[0], 1, NULL);
        ::WriteProcessMemory(g_process_handle, (LPVOID)0x531310,
(LPCVOID)&data1[1], 1, NULL);
        ::WriteProcessMemory(g_process_handle, (LPVOID)0x531311,
(LPCVOID)&data1[2], 1, NULL);
        ::WriteProcessMemory(g_process_handle, (LPVOID)0x531312,
(LPCVOID)&data1[3], 1, NULL);*/
    }
    else {
        BYTE data1[] = { 0x2b, 0x7c, 0x24, 0x20 };
        WriteMemory(g_process_handle, data1, sizeof(data1), address1);

        BYTE data2[] = { 0x2b, 0xc8 };
        WriteMemory(g_process_handle, data2, sizeof(data2), address2);

        BYTE data3[] = { 0x75, 0x17 };
        WriteMemory(g_process_handle, data3, sizeof(data3), address3);
    }
}

// 植物不死
void CPlantsVsZombiesWGDlg::OnBnClickedCheckPlantsNodeath()
{
    DWORD address1 = 0x52FCF0;
    DWORD address2 = 0x46D7A6;
    DWORD address3 = 0x45EC63;
    DWORD address4 = 0x46CFEB;
    if (m_check_plabts_no_death.GetCheck()) {
        BYTE data1[] = { 0x90, 0x90, 0x90, 0x90 };
        WriteMemory(g_process_handle, data1, sizeof(data1), address1);

        BYTE data2[] = { 0x90, 0x90, 0x90 };
        WriteMemory(g_process_handle, data2, sizeof(data2), address2);

        BYTE data3[] = { 0x90, 0x90, 0x90, 0x90 };
        WriteMemory(g_process_handle, data3, sizeof(data3), address3);

        BYTE data4[] = { 0x90, 0x90, 0x90 };
        WriteMemory(g_process_handle, data4, sizeof(data4), address4);
    }
    else {
        BYTE data1[] = { 0x83, 0x46, 0x40, 0xFC };
        WriteMemory(g_process_handle, data1, sizeof(data1), address1);

        BYTE data2[] = { 0x29, 0x4E, 0x40 };
        WriteMemory(g_process_handle, data2, sizeof(data2), address2);

        BYTE data3[] = { 0x83, 0x46, 0x40, 0xCE };
        WriteMemory(g_process_handle, data3, sizeof(data3), address3);

        BYTE data4[] = { 0x29, 0x50, 0x40 };
        WriteMemory(g_process_handle, data4, sizeof(data4), address4);
    }
}


// 后台运行
void CPlantsVsZombiesWGDlg::OnBnClickedCheckRunInbg()
{
    DWORD address = 0x54E1C2;
    if (m_check_run_inbg.GetCheck()) {
        BYTE data[] = { 0x90, 0x90, 0x90 };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
    else {
        BYTE data[] = { 0x0F, 0x95, 0xC0 };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
}


void CPlantsVsZombiesWGDlg::OnBnClickedCheckPlantsNocd()
{
    DWORD address = 0x487296;
    if (m_check_plants_nocd.GetCheck()) {
        BYTE data[] = { 0x90, 0x90 };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
    else {
        BYTE data[] = { 0x7E, 0x24 };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
}


void CPlantsVsZombiesWGDlg::OnBnClickedCheckBigmouseNocd()
{
    DWORD address = 0x46324c;
    if (m_check_bigmouse_nocd.GetCheck()) {
        BYTE data[] = { 0x29, 0xc0, 0x90 };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
    else {
        BYTE data[] = { 0x83, 0xc0, 0xff };
        WriteMemory(g_process_handle, data, sizeof(data), address);
    }
}

// ----------------植物全屏种植--------------------
typedef struct PutPlantsNode {
    UINT x, y, id;
}PutPlants, *PPutPlants;

// 无参自定义汇编
__declspec(naked) void asmPutPlants() {
    __asm {
        pushad
        push -1
        push 1
        mov eax,1
        push 2
        mov ebx, ds:[0x6A9EC0]
        mov ebx, ds:[ebx+0x768]
        push ebx
        mov edx, 0x40D120
        call edx
        popad
        ret
    }
}

// 有参自定义汇编
DWORD __stdcall asmPutPlants2(LPVOID lpThreadParam) {

    PPutPlants p_param = (PPutPlants)lpThreadParam;
    UINT x = p_param->x;
    UINT y = p_param->y;
    UINT id = p_param->id;
    __asm {
        pushad
        push - 1
        push id
        mov eax, x
        push y
        mov ebx, dword ptr ds: [0x6A9EC0]
        mov ebx, dword ptr ds : [ebx + 0x768]
        push ebx
        mov edx, 0x40D120
        call edx
        popad
    }
    return 0;
}

// 带参数的注入
BOOL injectRemoteThread(LPVOID funcAddr, LPVOID paramAddr, DWORD paramSize)
{

    // 函数所需空间
    LPVOID threadFuncAddr = ::VirtualAllocEx(g_process_handle, NULL, 4096,
MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);
    // 写入函数汇编
    ::WriteProcessMemory(g_process_handle, threadFuncAddr, funcAddr, 4096,
NULL);

    // 参数所需空间
    LPVOID threadParamAddr = ::VirtualAllocEx(g_process_handle, NULL, 4096,
MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);
    // 写入参数汇编
    ::WriteProcessMemory(g_process_handle, threadParamAddr, paramAddr,
paramSize, NULL);

    // 执行注入的函数和参数
    HANDLE remoteThreadRet = ::CreateRemoteThread(g_process_handle, NULL,
0, (LPTHREAD_START_ROUTINE)threadFuncAddr, threadParamAddr, 0, NULL);

    BOOL is_sucess = FALSE;
    if (remoteThreadRet) is_sucess = TRUE;

    DWORD threadWaitRet = ::WaitForSingleObject(remoteThreadRet, 0);

    if (WAIT_TIMEOUT == threadWaitRet)
        ::CloseHandle(remoteThreadRet);
    else {
        ::VirtualFreeEx(g_process_handle, threadFuncAddr, 0, MEM_RELEASE);
        ::VirtualFreeEx(g_process_handle, threadParamAddr, 0, MEM_RELEASE);
    }

    return is_sucess;
}

// 组合框选择植物发生改变
```

```
void CPlantsVsZombiesWGDlg::OnCbnSelchangeCbxPlantsType()
{
    UpdateData(TRUE);
    UINT x = _ttoi(m_edit_plantX) - 1;
    UINT y = _ttoi(m_edit_plantY) - 1;
    UINT id = m_cbx_choose_plant.GetCurSel() + 1;
    PutPlants param = { x, y, id };

    if (!injectRemoteThread(asmPutPlants2, &param, sizeof(param)))
        ::MessageBoxA(NULL, "安放植物失败", "错误", MB_OK);
}

// 范围随机种植
void CPlantsVsZombiesWGDlg::OnBnClickedBtnRandPutPlants()
{
    // 无参注入测试
    /*LPVOID threadAddr = ::VirtualAllocEx(g_process_handle, NULL, 4096, MEM_COMMIT | MEM_RESERVE, PAGE_EXECUTE_READWRITE);
    ::WriteProcessMemory(g_process_handle, threadAddr, asmPutPlants, 4096, NULL);
    HANDLE remoteThread = ::CreateRemoteThread(g_process_handle, NULL, 0, (LPTHREAD_START_ROUTINE)threadAddr, NULL, 0, NULL);*/
    UpdateData(TRUE);
    UINT row = _ttoi(m_edit_plantX);
    UINT col = _ttoi(m_edit_plantY);

    for (int i = 0; i < row; i++) {
        for (int j = 0; j < col; j++) {
            PutPlants param = { i, j, rand() % 8 + 1 };
            if (!injectRemoteThread(asmPutPlants2, &param, sizeof(param))) {
                ::MessageBoxA(NULL, "安放植物失败", "错误", MB_OK);
                return;
            }
            ::Sleep(100);
        }
    }
}


void CPlantsVsZombiesWGDlg::OnClose()
{
    ::TerminateThread(g_monitor_thread, 0);
    ::CloseHandle(g_monitor_thread);
    ::CloseHandle(g_process_handle);

    CDialogEx::OnClose();
}
```