InvisiMole: Surprisingly equipped spyware, undercover since 2013

welivesecurity.com/2018/06/07/invisimole-equipped-spyware-undercover/

June 7, 2018

Hunting for secrets from high-profile targets while staying in the shadows

Zuzana Hromcová
7 Jun 2018 - 03:00PM

Hunting for secrets from high-profile targets while staying in the shadows

This is the modus operandi of the two malicious components of InvisiMole. They turn the affected computer into a video camera, letting the attackers see and hear what's going on in the victim's office or wherever their device may be. Uninvited, InvisiMole's operators access the system, closely monitoring the victim's activities and stealing the victim's secrets.

Our telemetry indicates that the malicious actors behind this malware have been active at least since 2013, yet the cyber-espionage tool was never analyzed nor detected until discovered by ESET products on compromised computers in Ukraine and Russia.

The campaign is highly targeted – no wonder the malware has a low infection ratio, with only a few dozen computers being affected.

InvisiMole has a modular architecture, starting its journey with a wrapper DLL, and performing its activities using two other modules that are embedded in its resources. Both of the modules are feature-rich backdoors, which together give it the ability to gather as much information about the target as possible.

Extra measures are taken to avoid attracting the attention of the compromised user, enabling the malware to reside on the system for a longer period of time. How the spyware was spread to the infected machines is yet to be determined by further investigation. All infection vectors are possible, including installation facilitated by physical access to the machine.

Installation and persistence

The first part of the malware we are looking at is a wrapper DLL, compiled with the Free Pascal Compiler. From our telemetry, we have observed that this DLL is placed in the Windows folder, masquerading as a legitimate mpr.dll library file with a forged version info resource.

```
1
2  1 VERSIONINFO
3  FILEVERSION 6,1,7600,16385
4  PRODUCTVERSION 6,1,7600,16385
5  FILEOS 0x40004
6  FILETYPE 0x2
7  {
8  BLOCK "StringFileInfo"
9  {
10         BLOCK "040904B0"
11         {
12                 VALUE "CompanyName", "Microsoft Corporation"
13                 VALUE "FileDescription", "Multiple Provider Router DLL"
14                 VALUE "FileVersion", "6.1.7600.16385 (win7_rtm.090713-1255)"
15                 VALUE "InternalName", "mpr.dll"
16                 VALUE "LegalCopyright", "© Microsoft Corporation. All rights reserved."
17                 VALUE "OriginalFilename", "mpr.dll"
18                 VALUE "ProductName", "Microsoft® Windows® Operating System"
19                 VALUE "ProductVersion", "6.1.7600.16385"
20         }
21 }
22
23 BLOCK "VarFileInfo"
24 {
25         VALUE "Translation", 0x0409 0x04B0
26 }
27 }
```

Figure 1 – The wrapper DLL poses as a legitimate mpr.dll library, both by its name and version info

We have not seen a wrapper DLL named differently; however, there are hints in the DLL code that it might be also named fxsst.dll or winmm.dll.

The first way in which the malware can be launched is by hijacking a DLL. Being placed in the same folder as explorer.exe, the wrapper DLL is loaded during the Windows startup into the Windows Explorer process instead of the legitimate library located in the %windir%\system32 folder.

We have found both 32-bit and 64-bit versions of the malware, which makes this persistence technique functional on both architectures.

As an alternative to DLL hijacking, other loading and persistence methods are possible. The wrapper DLL exports a function called GetDataLength. When this function is called, the DLL checks whether it was loaded by the rundll32.exe process with either explorer.exe or svchost.exe as its parent process, and only then does it launch the payload. This suggests other possible persistence methods – by scheduling a task (i.e. having svchost.exe as a parent process) or by installation in a startup registry key (explorer.exe being the parent process).

Regardless of the persistence method, the behavior of the malware and of the actual payload is the same in all cases. The wrapper DLL loads both the modules stored in its resources, named RC2FM and RC2CL, and (if DLL hijacking was used) finally loads the legitimate library into the explorer.exe process, in order not to disrupt the normal operation of the application, and thereby remain hidden.

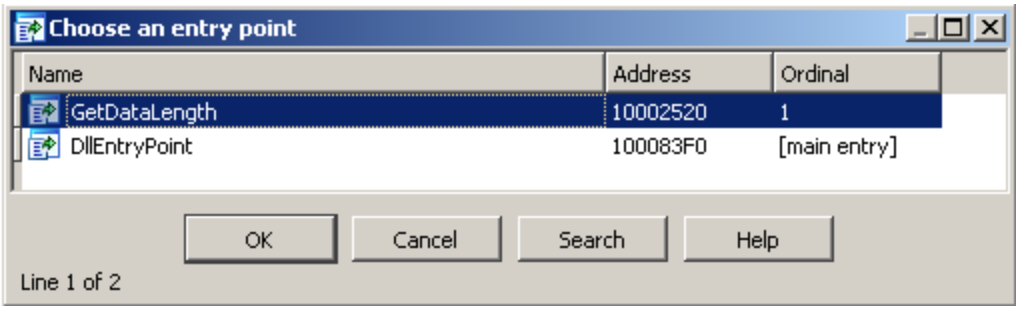


Figure 2 – Exported functions of the wrapper DLL

Technical analysis

The exact date when the malware was compiled is unknown – the recent wrapper DLL samples were tampered with by the malware authors, with the PE timestamps manually set to zero values. However, during our research, we found an earlier version of the malware with a PE timestamp reading Oct 13, 2013, so the compilation date of the later version is almost surely more recent.

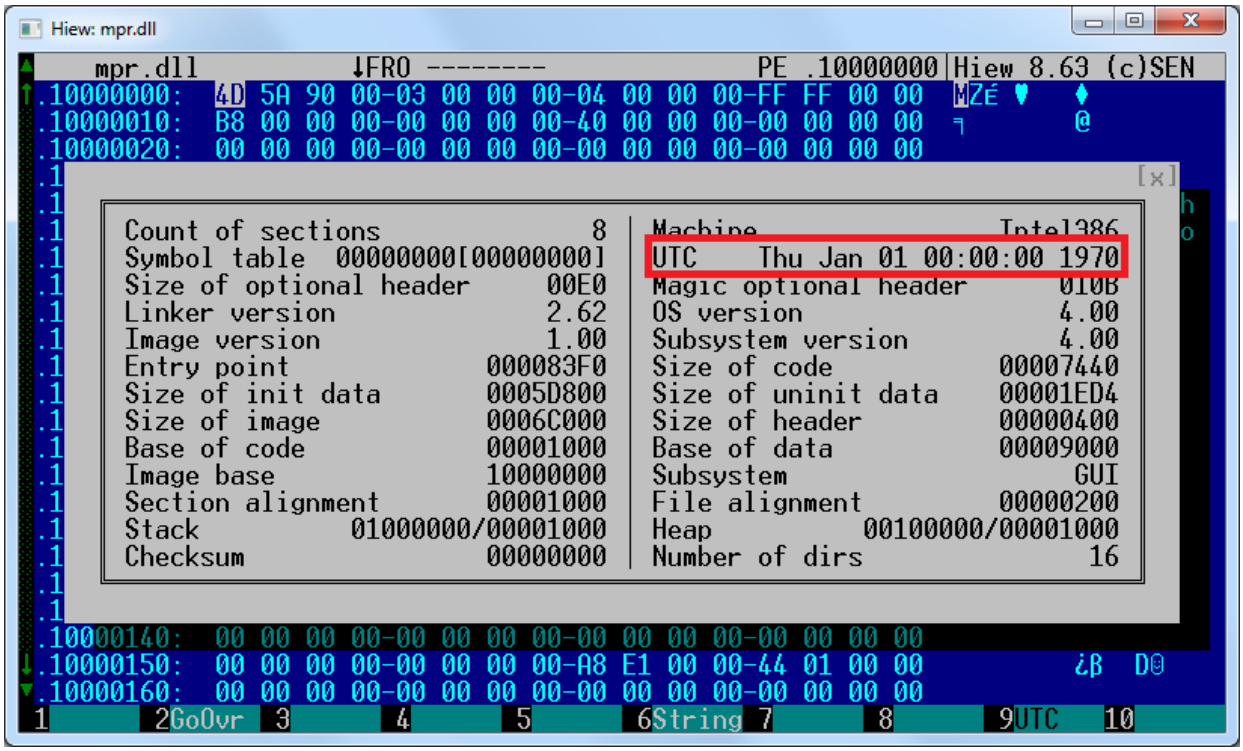
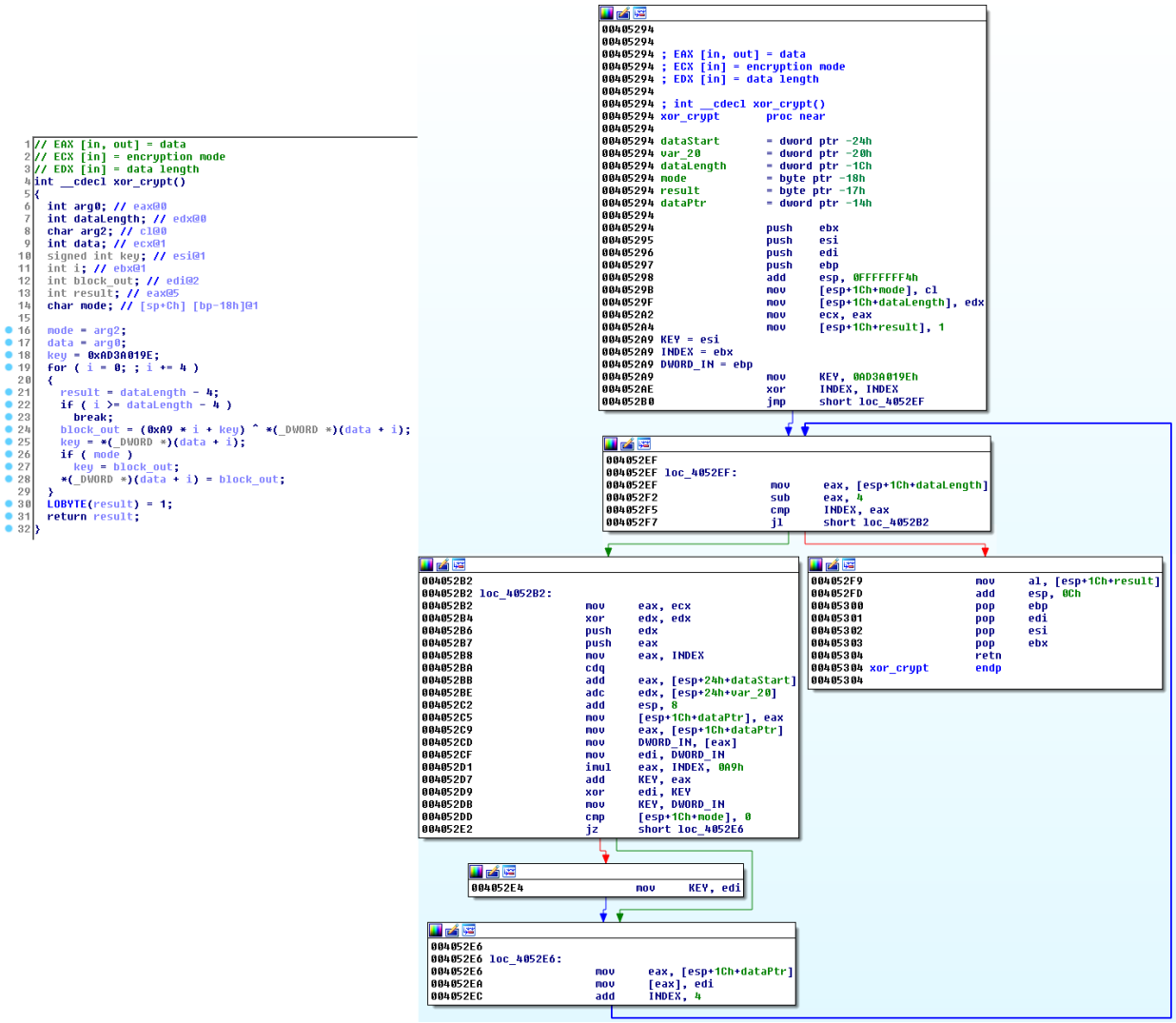


Figure 3 – The compilation timestamp is set to zero in all the latest samples

Encryption and decryption

To increase its level of stealth, the malware protects itself from the eyes of administrators and analysts by encrypting its strings, internal files, configuration data and network communication. While the RC2FM module uses a handful of custom ciphers, the wrapper DLL and the RC2CL module share one particular routine for all purposes, especially for decrypting other malware modules embedded in the wrapper DLL.

A script that is able to extract the embedded modules RC2FM and RC2CL from the wrapper DLL, using this routine, is available on ESET's malware-research GitHub repository.



```
// EAX [in, out] = data
// ECX [in] = encryption mode
// EDX [in] = data length
int __cdecl xor_crypt()
{
  int arg0; // eax@0
  int dataLength; // edx@0
  char arg2; // cl@0
  int data; // ecx@1
  signed int key; // esi@1
  int i; // ebx@1
  int block_out; // edi@2
  int result; // eax@5
  char mode; // [sp+Ch] [bp-18h]@1

  mode = arg2;
  data = arg0;
  key = 0xAD3A019E;
  for ( i = 0; ; i += 4 )
  {
    result = dataLength - 4;
    if ( i >= dataLength - 4 )
      break;
    block_out = (0xA9 * i + key) ^ *(_DWORD *)(data + i);
    key = *(_DWORD *)(data + i);
    if ( mode )
      key = block_out;
    *(_DWORD *)(data + i) = block_out;
  }
  LOBYTE(result) = 1;
  return result;
}
```

```
00405294
00405294
00405294 ; EAX [in, out] = data
00405294 ; ECX [in] = encryption mode
00405294 ; EDX [in] = data length
00405294
00405294 ; int __cdecl xor_crypt()
00405294 xor_crypt       proc near
00405294
00405294 dataStart       = dword ptr -24h
00405294 var_20          = dword ptr -20h
00405294 dataLength      = dword ptr -1Ch
00405294 mode            = byte ptr -18h
00405294 result          = byte ptr -17h
00405294 dataPtr         = dword ptr -14h
00405294
00405294                 push    ebx
00405295                 push    esi
00405296                 push    edi
00405297                 push    ebp
00405298                 add     esp, 0FFFFFFF4h
0040529B                 mov     [esp+1Ch+mode], cl
0040529F                 mov     [esp+1Ch+dataLength], edx
004052A2                 mov     ecx, eax
004052A4                 mov     [esp+1Ch+result], 1
004052A9 KEY = esi
004052A9 INDEX = ebx
004052A9 DWORD_IN = ebp
004052A9                 mov     KEY, 0AD3A019Eh
004052AE                 xor     INDEX, INDEX
004052B0                 jmp     short loc_4052EF

004052EF
004052EF loc_4052EF:
004052EF                 mov     eax, [esp+1Ch+dataLength]
004052F2                 sub     eax, 4
004052F5                 cmp     INDEX, eax
004052F7                 jl      short loc_4052B2

004052B2
004052B2 loc_4052B2:
004052B2                 mov     eax, ecx
004052B4                 xor     edx, edx
004052B6                 push    edx
004052B7                 push    eax
004052B8                 mov     eax, INDEX
004052BA                 cdq
004052BB                 add     eax, [esp+24h+dataStart]
004052BE                 adc     edx, [esp+24h+var_20]
004052C2                 add     esp, 8
004052C5                 mov     [esp+1Ch+dataPtr], eax
004052C9                 mov     eax, [esp+1Ch+dataPtr]
004052CD                 mov     DWORD_IN, [eax]
004052CF                 mov     edi, DWORD_IN
004052D1                 imul    eax, INDEX, 0A9h
004052D7                 add     KEY, eax
004052D9                 xor     edi, KEY
004052DB                 mov     KEY, DWORD_IN
004052DD                 cmp     [esp+1Ch+mode], 0
004052E2                 jz      short loc_4052E6

004052E4                 mov     KEY, edi

004052E6
004052E6 loc_4052E6:
004052E6                 mov     eax, [esp+1Ch+dataPtr]
004052EA                 mov     [eax], edi
004052EC                 add     INDEX, 4

004052F9                 mov     al, [esp+1Ch+result]
004052FD                 add     esp, 0Ch
00405300                 pop     ebp
00405301                 pop     edi
00405302                 pop     esi
00405303                 pop     ebx
00405304                 retn
00405304 xor_crypt       endp
00405304
```

Figure 4 – Encryption routine used across the samples (decompiled and disassembled)

Module RC2FM

The first, smaller module RC2FM contains a backdoor with fifteen supported commands. These are executed on the affected computer when so instructed by the attackers. The module is designed to make various changes to the system but it also offers a bunch of spying commands.

A logging option is implemented throughout the file but the name of the log file is not configured in the analyzed sample. This suggests that it was only used during the development of the malware.

Network communication

This module communicates with C&C servers that are either hardcoded in the sample, or updated later by the attackers.

Moreover, the module is able to reach out to the C&C servers even if there is a proxy configured on the infected computer. If a direct connection is unsuccessful, the module attempts to connect to any of its C&C servers using locally-configured proxies or proxies configured for various browsers (Firefox, Pale Moon, and Opera).

RC2FM can go as far as inspecting the recently executed applications list and look specifically for portable browser executables:

- FirefoxPortable.exe
- OperaPortable.exe
- Run waterfox.exe
- OperaAC.exe
- Palemoon-Portable.exe

Should the victim use one of these portable browsers *with a proxy server configured*, the malware can find that in the user's preferences and use that proxy to communicate with its C&C servers.

C&C communication consists of a series of HTTP GET and POST requests, as shown in Figure 5. The encrypted request includes a PC identifier and timestamp, and optionally some other data. It is worth noting that the RC2FM module uses a number of encryption methods (variations of a simple XOR encryption routine), unlike the other InvisiMole parts.

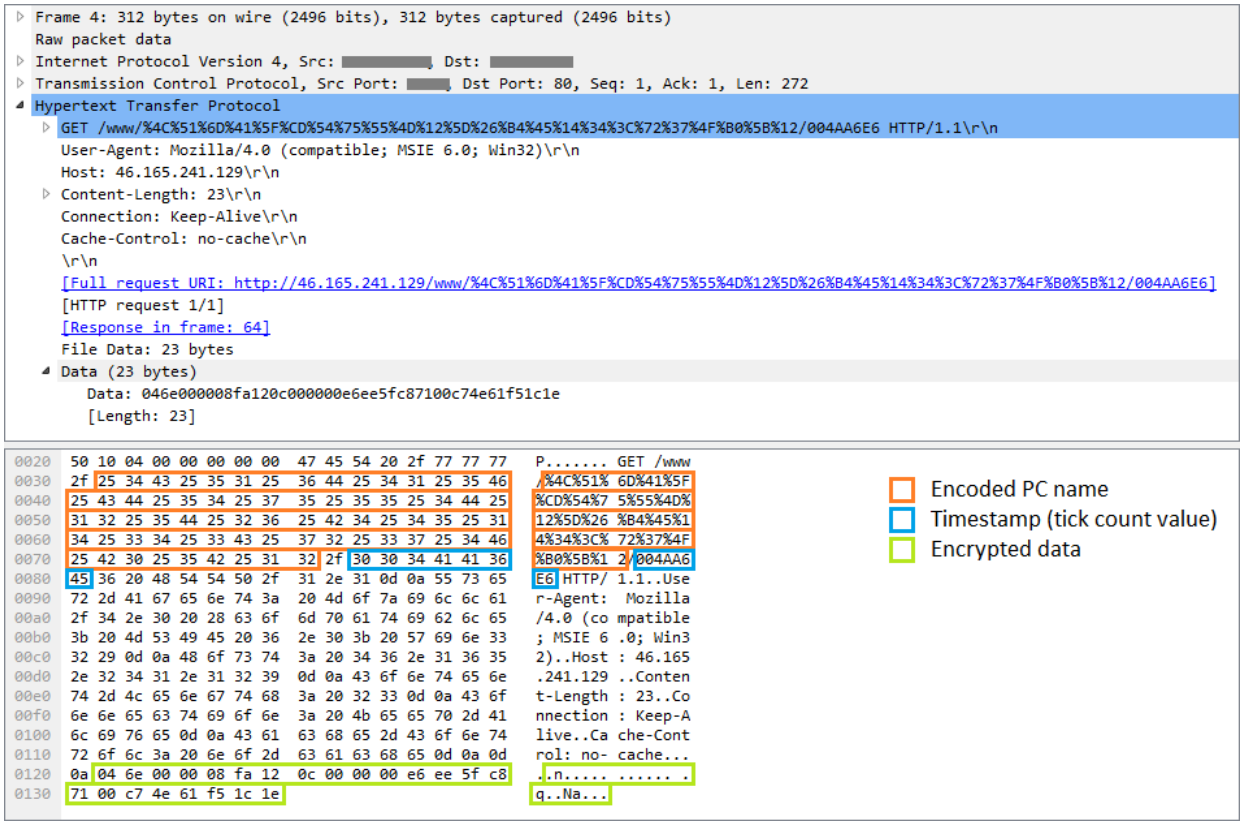


Figure 5 – Example of a request sent to the C&C server by the RC2FM module

After successfully registering the victim with the C&C server, additional data are downloaded, which are to be interpreted on the local computer as backdoor commands.

Capabilities

RC2FM supports commands for listing basic system information and performing simple changes on the system, but also includes a few spyware features. When required by the attacker, it is capable of remotely activating the microphone on the compromised computer and capturing sounds. The audio recordings are encoded to MP3 format using a legitimate lame.dll library, which is downloaded and misused by the malware.

Another way in which the malware can interfere with the victim's privacy is by taking screenshots, which is another of the backdoor commands.

The malware also monitors all fixed and removable drives mapped on the local system. Whenever a new drive is inserted, it creates a list of all the files on the drive and stores it encrypted in a file.

All of the collected data can ultimately be sent to the attackers, when the appropriate command is issued.

Backdoor commands

Fifteen commands are supported, as listed below. The backdoor interpreter function is visualized in Figure 6.

Command ID	Command description
0	List information about mapped drives, list files in a folder, list network shares
2	Create, move, rename, execute or delete a file, delete a directory using the specified path
4	Open a file, set the file pointer to the file beginning
5	Close a previously opened file
6	Write data into a previously opened file
7	Modify file times / delete a file
8	Open a file, set the file pointer to the end of the file
10	Modify file times / delete a file

Command ID	Command description
12	Search files by supplied file mask in a specified directory
13	Take a screenshot
14	Upload or modify files with internal data
15	Record sound using input audio devices, list available devices, send recordings, change configuration
16	Check whether this module currently has any files open
17	Update list of C&C servers
19	Create, set, copy, enumerate or delete the specified registry keys or values

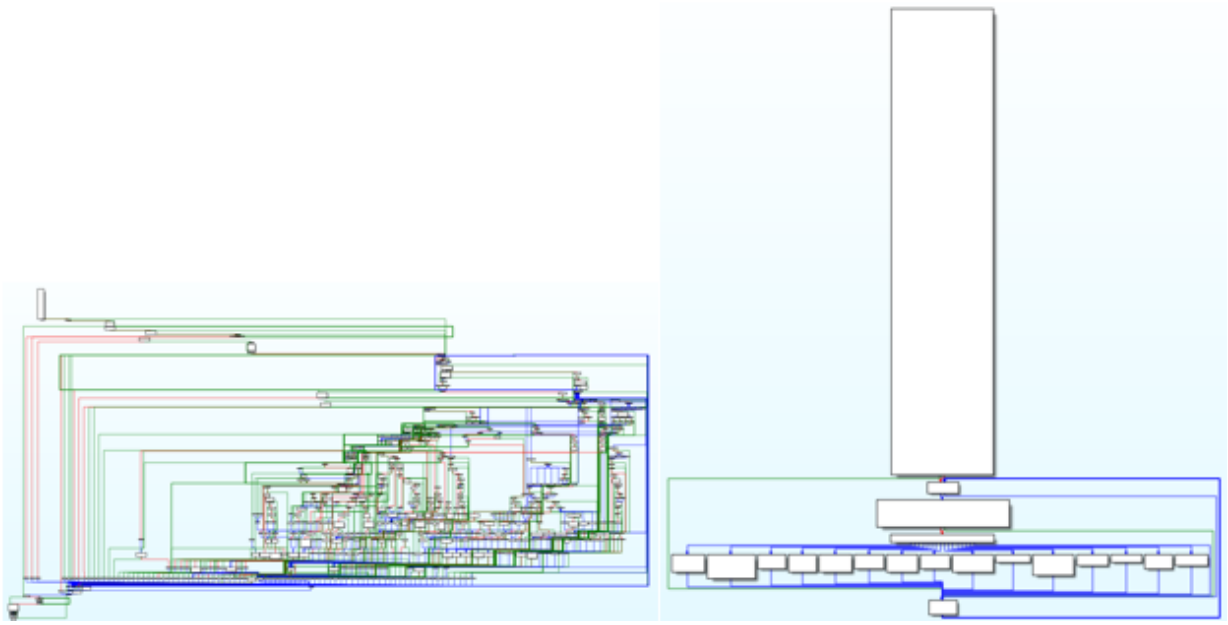


Figure 6 – Backdoor interpreter function (original and after our analysis, changed using Group Nodes functionality of IDA Pro for better readability)

Module RC2CL

The RC2CL module is also a backdoor with extensive spying capabilities. It is started by the wrapper DLL, launched at the same time as the RC2FM module. This one is more complex and offers features for collecting as much data about the infected computer as possible, rather than for making system changes.

Interestingly, there is an option in the RC2CL module to turn off its backdoor functionality and act as a proxy. In this case, the malware turns off the Windows firewall and creates a server that relays communication between a client and C&C server, or between two clients.

Network communication

The malware communicates with its C&C servers through a TCP socket. Messages sent from a client mimic the HTTP protocol, but note the invalid “HIDE” HTTP verb in the example in Figure 7.

These requests comprise an identifier of the compromised PC, the request type, and encrypted data that are to be sent to the attackers, i.e. the results of the backdoor commands or appeals for further instructions.

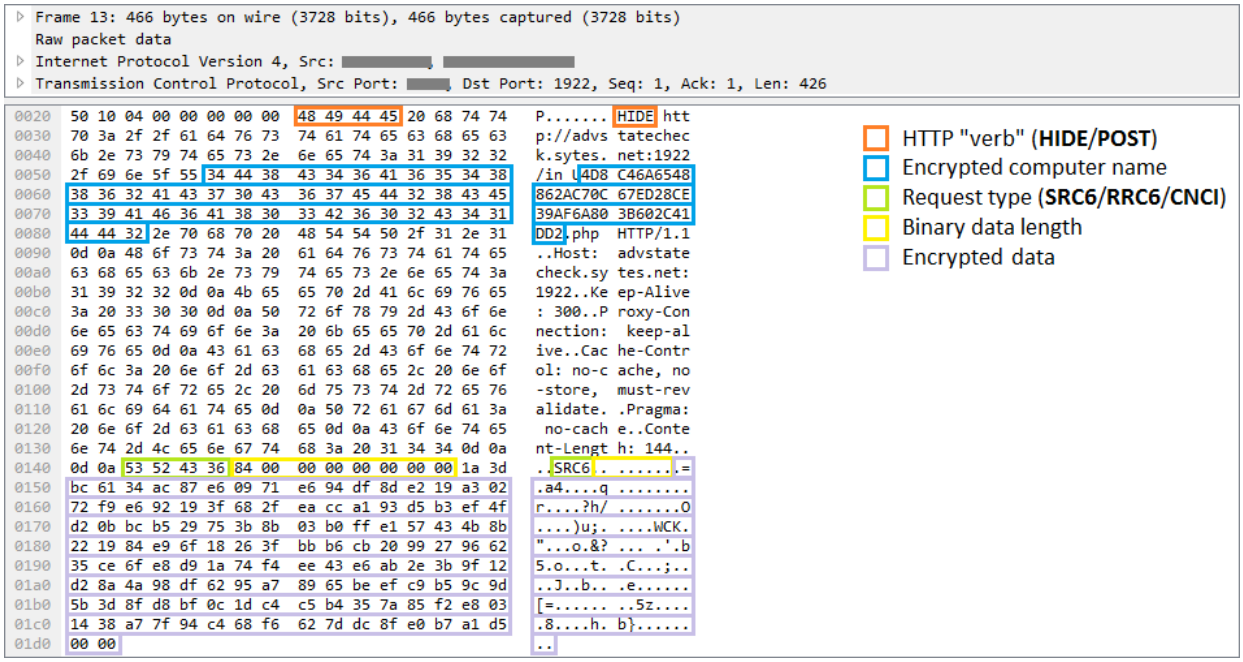


Figure 7 – Example of a request sent to the C&C server by the RC2CL module

Capabilities

Depending on the commands received, the backdoor can perform various actions on the infected computer. Common backdoors often support commands such as file system operations, file execution, registry key manipulation or remote shell activation. This spyware supports all of these instructions and a whole lot more – its 84 commands provide the attackers with all they need to look at their victims more closely.

The malware can inspect the infected computer and provide various data, from system information such as lists of active processes, running services, loaded drivers or available drives, to networking information, including the IP forward table and the speed of the internet connection.

InvisiMole is capable of scanning enabled wireless networks on the compromised system. It records information such as the SSID and MAC address of the visible Wi-Fi access points. These data can then be compared to public databases, letting the attackers track the geolocation of the victim.

Other commands can provide information about the users of the compromised machine, their accounts and previous sessions.

The software installed on the compromised computer is of particular interest. Which programs are installed on the system? Which of them are executed automatically at each system start or user logon? Which programs are used by a particular user? If the attackers are interested, they are only one command away from these valuable data.

The malware can be instructed to search for recently-used documents or other interesting files. It can monitor specific directories and removable devices, report any changes and exfiltrate files of the attackers' choice.

The malware may enable or disable the User Account Control (UAC), or even bypass the UAC and work with the files in secure locations without having administrator privileges (see more at https://wikileaks.org/ciav7p1/cms/page_3375231.html). If the malware is running under the explorer.exe process, which is auto-elevated, it can create an elevated COM object and use it to delete or move files in locations that require admin rights.

What is even more disturbing is that it can remotely activate the victim's webcam and microphone and spy on the victim by taking pictures and recording sound. Screen activity can be monitored by capturing screenshots. What is particularly interesting about InvisiMole is that not only are the usual "whole display" screenshots taken – it can separately capture each window, which helps the attackers gain more information even when the windows are overlapped.

Further, one of the backdoor commands is used to replace the contents of drivers with the following names:

- blbdrive.sys
- compbatt.sys
- secdrv.sys

We have not observed the attackers actually using this command but we can speculate that it does so to achieve additional persistence on 32-bit systems.

Though the backdoor is capable of interfering with the system (e.g. to log off a user, terminate a process or shut down the system), it mostly provides passive operations. Whenever possible, it tries to hide its activities.

For instance, the malware sniffs around interesting places on the system, reads recent documents or even modifies some files. This leaves traces on the system and could raise the victim's suspicions as the time of the last access or modification of the files is changed with each such activity. To prevent this, the malware always restores the original file access or modification times, so that the user is unaware of its operation.

Another example of how the malware authors attempt to act covertly is the way they treat traces left on the disk. The malware collects loads of sensitive data, which are then temporarily stored in files and deleted after they have been successfully uploaded to the C&C servers. Even the deleted files can, however, be recovered by an experienced system administrator, which could help further investigation of the attack – after the victim becomes aware of it. This is possible due to the fact that some data still reside on a disk even after a file is deleted. To prevent this, the malware has the ability to safe-delete all the files, which means it first overwrites the data in a file with zeroes or random bytes, and only then is the file deleted.

Internal storage

The backdoor configuration and the data collected are stored in one of two places – a working directory and working registry keys. A substantial portion of the backdoor command set is dedicated to manipulating these storage locations and their contents.

The location of the working directory is determined by the instructions from the remote server. The directory is used as temporary storage for files containing collected data about the compromised computer. Such files share a common naming convention, encryption algorithm and structure. They are encrypted by a simple variation of the XOR cipher which is used across the malware components. The type of the file can be derived from the 4-byte control sequences placed at the beginning of the file.

Besides being a storehouse for the gathered data, the working directory is also home to a copy of the legitimate WinRAR.exe application. This is copied by the malware and abused by the attackers for compressing the data that are to be exfiltrated.

The working registry keys store configuration data, as well as a list of files in the working directory. The data are packed using a Zlib routine implemented in the malware binary and encrypted with the same cipher as the internal files.

Subdirectory name	File name	Control sequences	File content
\	~mrc_%random%.tmp	932101DA	Audio recordings
\	~src_%random%.tmp	958901DA	Audio recordings
\	~wbc_%random%.tmp	938901DA	Webcam photos
sc\	~sc%random%.tmp	DFE43A08	Screenshots
~zlp\	zdf_%random%.data	B1CBF218	Zlib-compressed packages

Subdirectory name	File name	Control sequences	File content
~lcf\	tfl_%random%	C0AFF208	Internal data
fl_%timestamp%\strcn%num%\	fdata.dat	A1CAF108	Data from removable drives
fl_%timestamp%\strcn%num%\	index.dat	BAAB0019	Data from removable drives
Winrar\	WinRAR.exe	-	Copy of a legitimate application
Winrar\	comment.txt	-	-
Winrar\	descript.ion	-	-
Winrar\	Default.SFX	-	-
Winrar\	main.ico	-	-

Backdoor commands

The backdoor provides more than eighty commands that utilize the working directory and registry keys to store their intermediate results and configuration data. The graph of the backdoor interpreter is shown in Figure 8.

Approximately a third of the commands are dedicated to reading and updating the configuration data stored in the registry. The rest of the commands are listed in the table below.

Command ID(s)	Command description
4	List information about files in a directory
6	Upload a file
20	List information about active processes
22	Terminate a process by ID
24	Execute a file
26	Delete a file

Command ID(s)	Command description
28	Get the IP forward table
30	Write data to a file
31	Sleep
38	List account information
40	List information about services on the system
42	List information about loaded drivers
43	Collect basic system information (computer name, OS version, memory status, local time, drive information, configured proxy information, system and process DEP policy…)
44	List installed software
46	List local users and session information
48	List applications accessed by users
52	Create a directory structure
78	Create a remote shell
81	Execute a command via a remote shell
91	Enable/disable UAC
93	Log off the user/shutdown/restart the system
101	Monitor and record changes in the specified directories
103	Delete directories
109	Turn the monitor on/off/onto standby
120	Capture screenshots of the display/active windows
126	Capture screenshots of the display/active windows & update configuration data
130	List information about resources on unmapped drives
132	Rename/move a file, modify create/access/write times of the file to the given values
134	List information about recently opened files

Command ID(s)	Command description
152	Disconnect (previously connected) remote drives
155	Create/delete a registry key, set/delete a registry key value, or enumerate registry values/keys/data
159, 161	Disable routing/firewall, create a proxy server on a specified port
172	Repeatedly display a dialog requesting the user to reboot the computer
175	Bypass UAC to manipulate a file
177	Create and write a file, set the create/access/modify times
181	Remove all system restore points
183	Drop (legitimate) WinRAR components
185	Add files to a password-protected archive (password = "12KsNh92Dwd")
187	Decrypt, unpack and load a DLL, load executables from its resources RC2CL, RC2FM
189	Create a system restore point
191	Extract a password-protected archive (12KsNh92Dwd)
193	Modify an encrypted file
195	Restart itself after the primary process finishes
197	Send 198 bytes of data hardcoded in the sample
199	Rename/move a file
206	Decrypt, unpack and load a DLL, load executables from its resources RC2CL, RC2FM
211	Upload collected information (captured screenshots, audio recordings, etc.)
213	List information about active windows
218	API for recording input audio devices
220	API for capturing webcam photos
224	List files executed with each system start
226	List information about enabled wireless networks (MAC address, SSID, beacon interval)

Command ID(s)	Command description
228	Drop a Zlib-compressed package

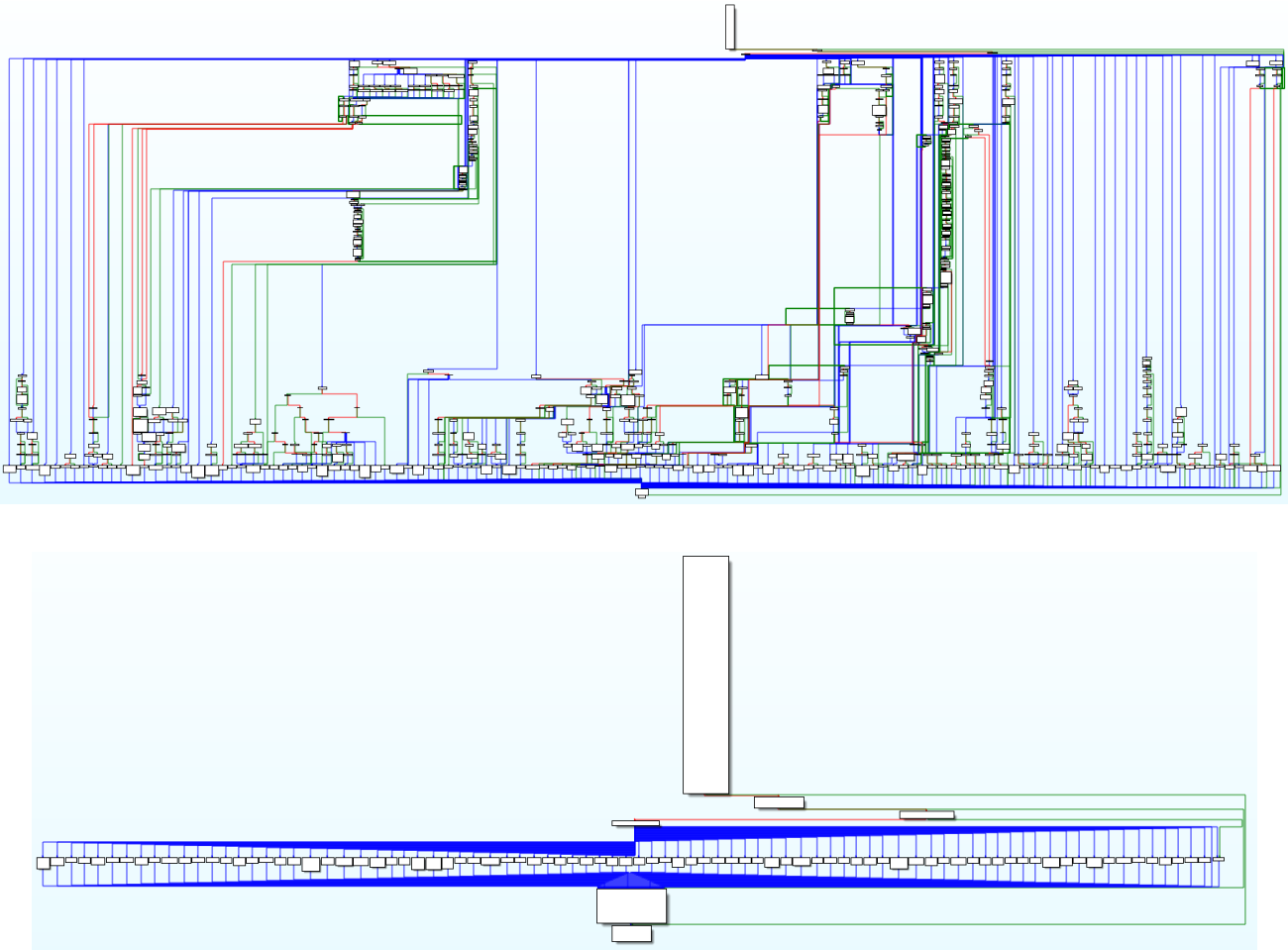


Figure 8 – Backdoor interpreter function (original and after our analysis, changed using Group Nodes functionality of IDA Pro for better readability)

Conclusion

InvisiMole is fully-equipped spyware whose rich capabilities can surely compete with other espionage tools seen in the wild.

We can only wonder why the authors decided to use two modules with overlapping capabilities. One might think the smaller module, RC2FM, is used as an initial reconnaissance tool, while the bigger RC2CL module is only run on interesting targets. This is, however, not the case – both of the modules are launched simultaneously. Another possible explanation is that the modules might have been crafted by various authors and then bundled together to provide the malware operators a more complex range of functionalities.

The malware uses only a few techniques to avoid detection and analysis, yet, deployed against a very small number of high-value targets, it was able to stay under the radar for at least five years.

Indicators of Compromise (IoCs)

A full and comprehensive list of IoCs, C&C servers, along with registry keys and values can be found on GitHub.

7 Jun 2018 - 03:00PM

Sign up to receive an email update whenever a new article is published in our Ukraine Crisis – Digital Security Resource Center

Newsletter

Discussion