

a. Virtualization phase:

Virtualization was fairly simple. We initially used VMware Workstation 17 Player for both the Windows 10 and Windows XP machines. Windows XP worked just fine, but our Windows 10 machine was very slow and laggy and sometimes crashed, and after looking online it looked like that was a problem a lot of people faced, so we decided to use Oracle VirtualBox for our Windows 10 machine. It was much faster and did not crash.

**Note: most tools are used on Windows 10 unless otherwise stated.*

b. Malware collection phase:

We found our malware on <https://bazaar.abuse.ch/> (Malware Bazaar). Finding an infected .exe file was very straightforward, many samples were available online.

c. Static malware analysis phase:

1. General analysis:

- **VirusTotal:**

Immediately, the file displays extremely suspicious behavior:

55 security vendors and 2 sandboxes flagged this file as malicious

672ac1422873d7481b4d37c1c79b4818a96815c7271b0598ef01bdd9dabe3f74

nXw.exe

Size: 418.50 KB | Last Analysis Date: 1 hour ago

peexe assembly checks-cpu-name detect-debug-environment long-sleeps check-user-input calls-wmi spreader persistence

Community Score

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Popular threat label: Trojan:msil:agenda Threat categories: trojan Family labels: mail agenda peexe

Security vendors' analysis

Vendor	Detection	Family
AhnLab-V3	Trojan:Win.PWSX-gen.C5546910	Alibaba
ALYac	Trojan.GenericKD.70505219	ArcaBit
Avast	Win32:PWSX-gen [Trj]	AVG
BitDefender	Trojan.GenericKD.70505219	BitDefender.Theta
BitDefender	Trojan.GenericKD.70505219	GenNN.ZemaisICO.3660B.AmO@48k35f

As we can see, most malware analysis websites classified the file as malicious. If we take a closer look at the file's details:

DETECTION DETAILS RELATIONS BEHAVIOR COMMUNITY 4

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

Basic properties

MD5	e3ff1e89d54c3469f11b3340c5d83f12
SHA-1	1ce54890e5f1a9076c54464527d88c7bb8931b88
SHA-256	672ac1422873d7481b4d37c1c79b4818a96815c7271b0598ef01bdd9dabe3f74
Vhash	245036751512708218293021
Authenticity hash	89819d7411c7a28ff5a59a2d93f08adb9d78e1c08448e8bc9f200aaf5808272
Imphash	f34d5f2d4577edd9ceec516c1f5a744
SSDEEP	6144:poKALTom4JkwXIGRLOYbtyO147nIX2pMx48yJ1SEVToyENhVUPEf7/QMAezXEZ:pGoB1EL4fD7W2p144WNhVUPLU1ACQK
TLSH	T1EF94123072FE6B3E5B553F15826620443F5726F6231E62A2CC680DA9A92F414F1FB3
File type	Win32 EXE (executable) windows win32 pe peexe
Magic	PE32 executable (GUI) Intel 80386 Mono/Net assembly, for MS Windows
TrID	Generic CIL Executable (.NET, Mono, etc.) (63%) Windows screen saver (11.2%) Win4 Executable (generic) (9%) Win32 Dynamic Link Library (generic) (5.6%) Win32 Executable (generic) (3.8%)
DetectItEasy	PE32 Library: .NET (v4.0.30319) Compiler: VB.NET Compiler: VB.NET Library: .NET (v4.0.30319) Linker: Microsoft Linker
File size	418.50 KB (428544 bytes)

Nothing suspicious in basic properties. If we move on to the history of the file, however:

History ⓘ

Creation Time	2095-12-16 23:44:49 UTC
First Seen In The Wild	2023-11-29 12:32:14 UTC
First Submission	2023-11-21 11:40:42 UTC
Last Submission	2023-11-28 13:54:45 UTC
Last Analysis	2023-11-29 21:00:21 UTC

The creation time of the file is set to be in 2095, this is the first clear sign of malicious activity and file obfuscation.

If we look at the names for the file:

Names ⓘ

672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74.exe
nXw.exe
njrat_qui_a_casse_extracteur_nsm.bin
purches order.exe
USVrid.exe

“purches order.exe” raises some suspicion, file may be linked to some shopping website, attacker also possibly misspelled purchase to avoid the file being detected.

Digital signature:

Signature info ⓘ

Signature Verification

⚠ File is not signed

No digital signature, typical for malware to avoid detection, makes it challenging to verify if this file came from a legitimate source.

Sections in the PE info:

Sections

Name	Virtual Address	Virtual Size	Raw Size	Entropy	MD5	Chi2
.text	8192	425800	425984	7.91	4295b0effeb077573655dc55fe3a0af9	136136.12
.rsrc	434176	1452	1536	4.08	f6177b481524abbefafa5ce5d97e91a9	78507.83
.reloc	442368	12	512	0.1	28d762c37b2f106bb87e683103d96ca3	128015

Three things can be found from the screenshot above:

- a. Virtual address for .text section is very far from other two sections. In a well-structured and properly compiled executable file, the section addresses are typically contiguous; this address seems suspicious.
- b. Entropy in .text section is extremely high, big indicator of malicious activity
- c. Very big difference in Virtual vs. Raw size in .reloc section, also indicates malicious behavior

These were the most obvious signs found using VirusTotal.

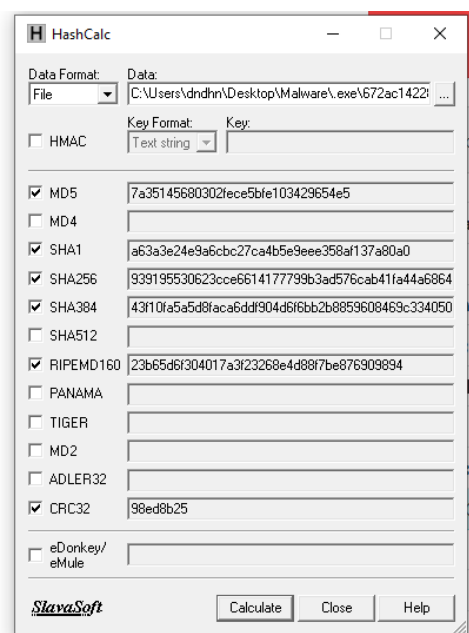
2. Hash calculations:

▪ HashCalc:

The original hashes provided by the author of the file were as follows:

SHA256 hash:	672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74
SHA3-384 hash:	4b0ee649547bfc1d444646946cc811f0e4ffb7d4e5fa7214ab1a707ee9bb837ba9ea9418e1f8b20cc4e2be2cbd3faab3
SHA1 hash:	1ce54890e5f1a9076c54464527d88cfbb8931b88
MD5 hash:	e3ff1e89d54c3469f11b3340c5d83f12
humanhash:	artist-paris-robert-orange
File name:	purches order.exe
Download:	download sample
Signature ©	njrat Alert

The hashes provided by the HashCalc tool, however, did not match the original checksums:



- **MD5Deep:**

The screenshot displays a file's metadata on the left and a command prompt window on the right. The file's hashes are as follows:

Hash Type	Value
SHA3-384 hash:	4b0ee649547bfc1d444646946cc811f0e...
SHA1 hash:	1ce54890e5f1a9076c54464527d88cfbb...
MD5 hash:	e3ff1e89d54c3469f11b3340c5d83f12...
humanhash:	artist-paris-robert-orange
File name:	purchases order.exe
Download:	download sample
Signature	njrat
File size:	428'544 bytes
First seen:	2023-11-27 09:20:26 UTC
Last seen:	2023-11-27 11:53:54 UTC
File type:	exe
MIME type:	application/x-dosexec
imphash	f34d5f2d4577ed6d9ceec516c1f5a744...
ssdeep	6144:poKALTom4JkwXIGRL0Ybtyf0/47n...

The command prompt window shows the following commands and output:

```

C:\Users\dndhn\Desktop\Malware\md5deep>md5deep64 C:\Users\dndhn\Desktop\Malware\...exe
672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74.exe
C:\Users\dndhn\Desktop\Malware\md5deep>md5deep-4.3 C:\Users\dndhn\Desktop\Malware\...exe
672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74.exe
  
```

In the screenshot above, we can see the original hash (*left*) and the resulting hash from the MD5Deep tool (*right*) also don't match.

Although there might be other reasons for hash differences, such as file corruption or a software patch, it could also be a sign of file tampering and obfuscation.

Comparing hashes is a great start to detecting malicious activity, but it's not enough on its own to prove it. We need additional information to determine whether or not the file is really infected.

3. Strings and binary analysis:

- **Strings:**

Using the strings command, our file appeared to have a very large number of strings, most of which were pretty insignificant to our analysis. We limited the length of the strings shown to 10 or more characters, in hopes of getting some human-readable text:

```

C:\Users\dndhn\Desktop\Malware>strings64 -n 10 C:\Users\dndhn\Desktop\Malware\...exe

Strings v2.54 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2021 Mark Russinovich
Sysinternals - www.sysinternals.com

!This program cannot be run in DOS mode.
v4.0.30319
B6D84637B6260209ABE087B8BFCDD8287730EB84D569DD20FD02E585F030A1F51
backgroundWorker1
pictureBox1
__StaticArrayInitTypeSize=22
<PrivateImplementationDetails>
System.Data
  
```

Some findings:

- The file contained many SQL commands, mostly SELECT statements, which indicates that the file might be a script related to some database.

```
dgVACCESSLOG
SELECT TENDONVI FROM QLHT.NHAN_VIEN NV JOIN QLHT.DON_VI DV ON NV.MADONVI = DV.MADONVI WHERE USERNAME = '
btnXemThongTinNhanVien
btnXemThongBao
btnXemHoSoBenhAn
frmNhanVien
SELECT * FROM QLHT.V_HOSOENHAN
TINHTRANGBANDAU
dgVDSHoSoBenhAn
frmXemHoSoBenhAn
XemHoSoBenhAn
SELECT * FROM QLHT.THONG_BAO
dgVDSThongBao
frmXemThongBao
XemThongBao
dgVDSNhanVien
frmXemThongTinNhanVien
Data Source=(DESCRIPTION =(ADDRESS = (PROTOCOL = TCP)(HOST =
))(CONNECT_DATA = (SERVER = DEDICATED)(SERVICE_NAME =
))) ;Password=
```

- The words “Role”, “Privilege” and “Password” appeared multiple times in the file, which suggests the file may be responsible for access control and security configuration. This could be related to defining user roles and their associated privileges.

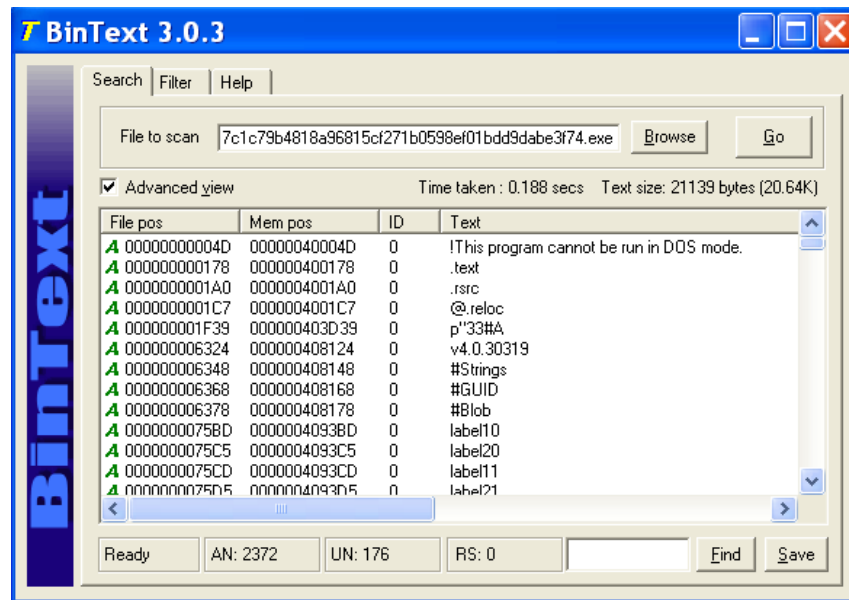
```
CreateNewRole
UpdateRole
GrantObject
RevokeObject
ng tin user
bntLoadUser
ng tin quy
a role/user
bntLoadRoleUser
dtgvRoleUser
User/Role
bntRunRole
txtRolePass
txtRoleName
bntRunUser
txtUserPass
txtRevokeTable
txtRevokePrivilege
n Role/ User
txtRevokeRoleUser
```

- .NET framework v4 is used for the application. mscore.dll file is also present, which is a critical component for managing the execution of .NET applications. It helps handle tasks such as memory management, exception handling, and the loading and execution of managed code. Some copyright information can also be seen.

```
mscoree.dll
AE77Y5Z57548FVG3J5XE54
VS_VERSION_INFO
VarFileInfo
Translation
StringFileInfo
CompanyName
FileDescription
Quanlissinhvien
FileVersion
InternalName
LegalCopyright
Copyright
LegalTrademarks
OriginalFilename
ProductName
Quanlissinhvien
ProductVersion
```

Our file seems to be some kind of role and privilege manager, related to a database, which can be seriously dangerous if it is indeed obfuscated. The strings tool helped us learn a lot about the file, but nothing definitively malicious was found using the tool. While some suspicious text, and a lot of potentially encrypted information was found, it's important to remember that these patterns can also exist in typical .exe files, so they are not enough to conclude that a file is malicious, but they are a good start to analyzing a file.

- **BinText:** (*Windows XP*)

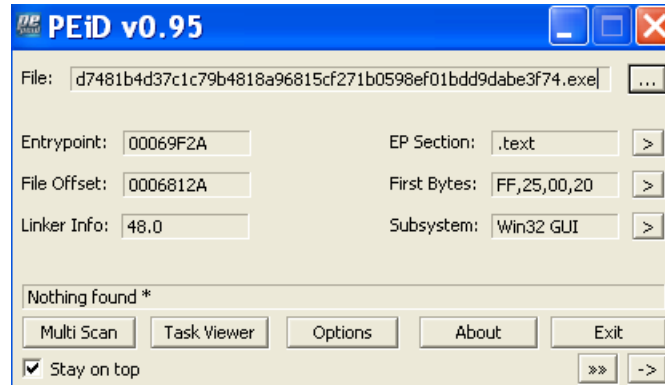


Same outputs as the strings tool, but some things made BinText easier:

- You can see text, function calls, IP addresses.
- You can also filter the way things are viewed using filter button, in a much easier way. You can modify:
 - What characters to include/ not to include in the definition of strings
 - String size (min./Max. length)
 - Essentials

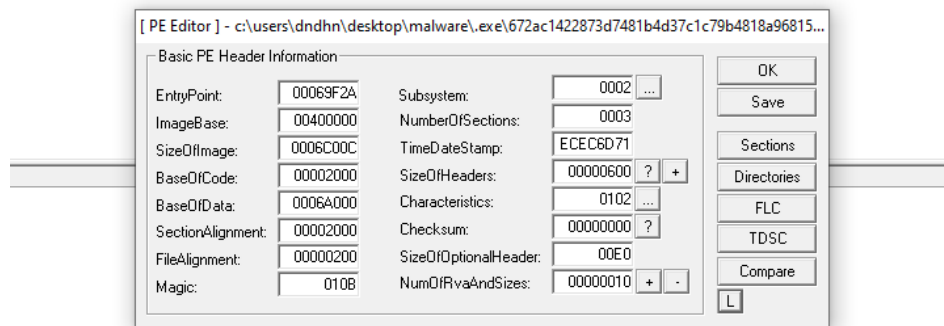
4. Portable Executable file viewers:

- **PEiD:** (*Windows XP*)

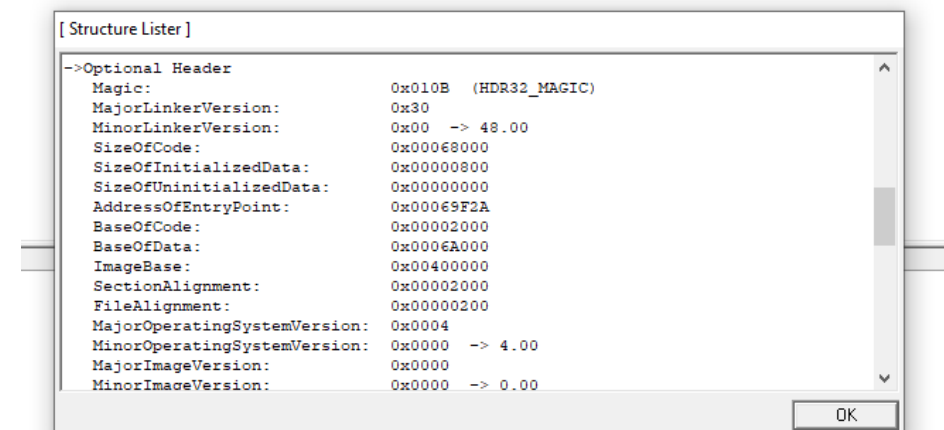
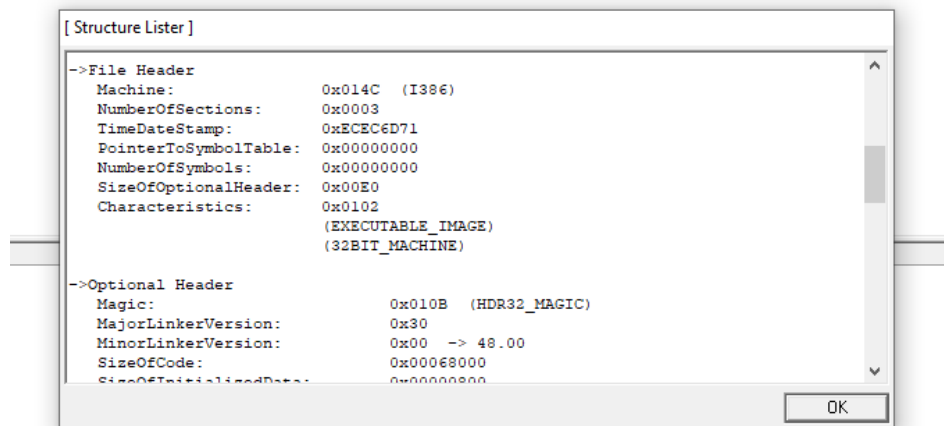


Not a lot of information about the file, however, a good thing about PEiD is that it gives the ability to see the offset and task view. In this malware, when detecting the packers, nothing was found (no files are packed) according to PEiD. It also tells us it is GUI program.

- **LordPE:**



List of some important fields:



After referring to **PE file format**, to compare the values above to the values you would typically get from an executable (image) file, we concluded the following:

Most of the fields' values are valid:

- **EntryPoint** should fall in the range between (ImageBase – ImageBase+SizeOfCode); $0x00069F2A \leq (0x00400000 + 0x00068000)$, so EntryPoint is valid. It's important to look at what this location points to, however.
- **ImageBase** (0x00400000) is the default value for Windows 10: also valid.
- **SectionAlignmnet** should be greater than **FileAlignment**: also both valid.
- **NumberOfSections** is 3, for the .text, .rsrc and .reloc sections: valid.

However, after converting **TimeStamp** field to a human-readable date, we get the following:

epochconverter.com

Convert epoch to human-readable date and vice versa

3974917489 Timestamp to Human date [batch convert]

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **seconds**:

GMT : Friday, December 16, 2095 11:44:49 PM

Your time zone : Saturday, December 17, 2095 2:44:49 AM GMT+03:00

Relative : In 72 years

Yr Mon Day Hr Min Sec
2023 - 11 - 29 20 : 34 : 19 GMT Human date to Timestamp

As we can see in the screenshot above, the supposed time of creation for the file is in the year 2095, this is possibly the biggest sign that there was an attempt to manipulate or obfuscate information.

Lastly, if we take a closer look at the Sections in the file:

Name	VOffset	VSize	ROffset	RSize	Flags
.text	00002000	00067F48	00006000	00068000	60000020
.rsrc	0006A000	000005AC	00068600	00000600	40000040
.reloc	0006C000	0000000C	00068C00	00000200	42000040

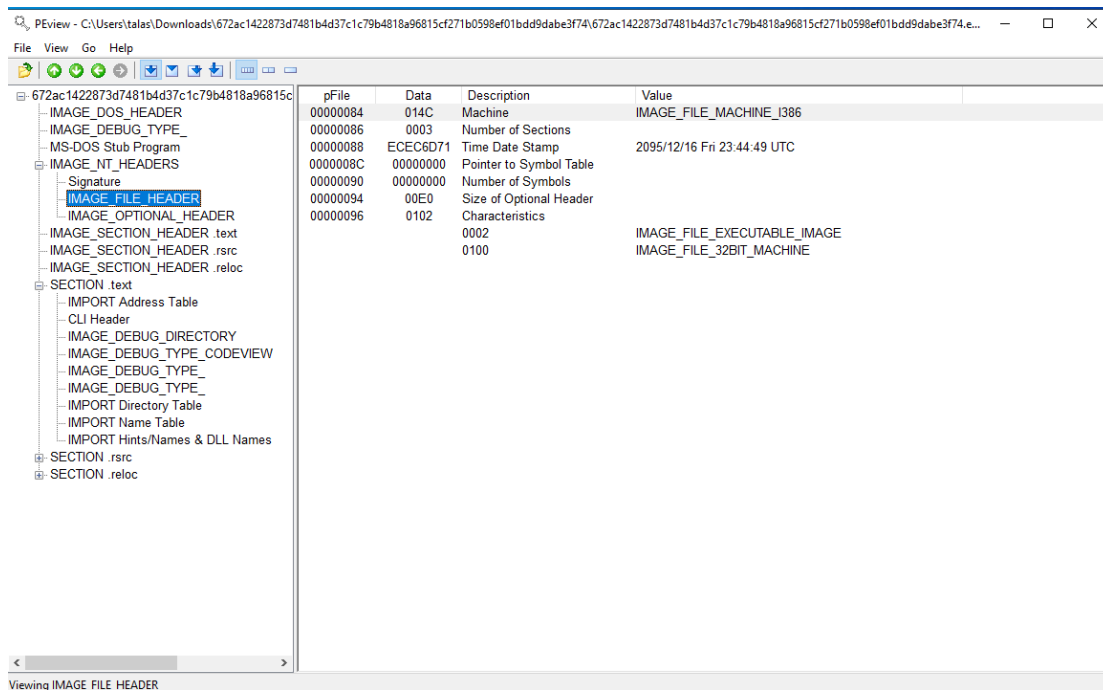
FileAlignment: 00000200 SizeOfOptionalHeader: 00000000
Magic: 010B NumOfRvaAndSizes: 00000010

Compare L

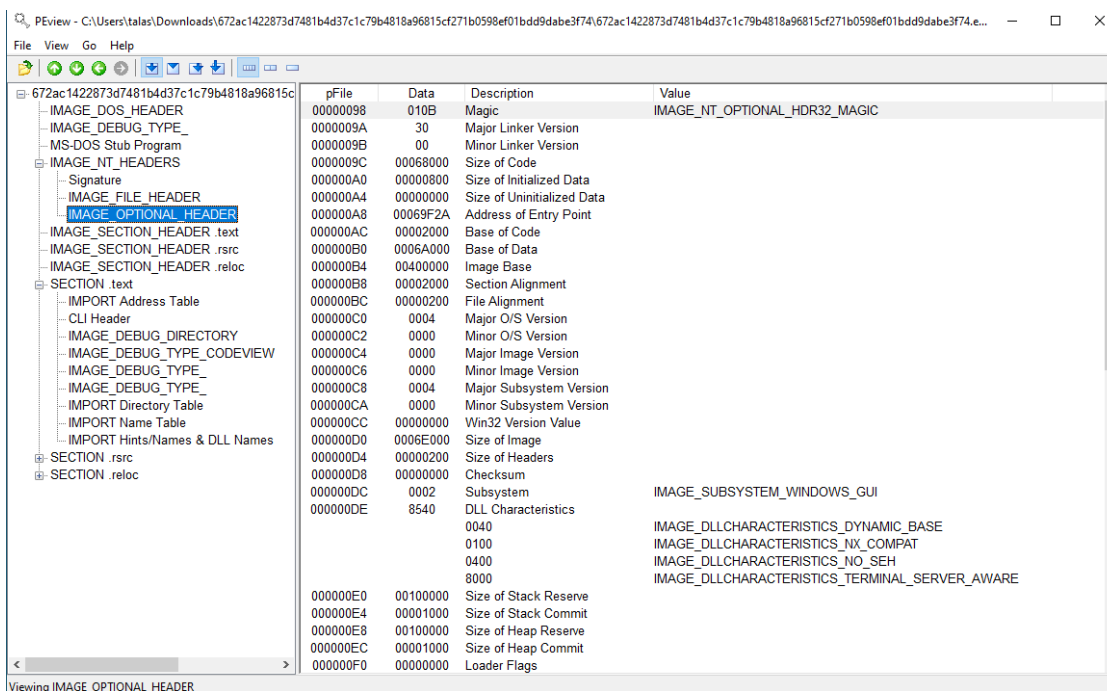
The .rsrc section seems fine, but the **.text** and **.reloc** sections display suspicious behavior: the **.text** section's address is nowhere near the other two sections, and as we mentioned before, these sections are typically contiguous which makes this address suspicious. **.reloc** section: typically, the VSize is equal to or larger than the RSize for sections, especially for the .reloc section, which often involves memory adjustments during runtime. This is also a sign of malicious tampering of the file, possibly obfuscating the information in the .reloc section.

■ PView:

A cleaner look at our findings:



As we mentioned previously, the Time Date Stamp is the most obvious indicator that this file was tampered with. PView makes it a lot easier to detect that as it translates the values for us.



All the information above also matches the information gathered from LordPE, nothing suspicious about the fields here, all values seem to fall within a valid range.

Hex view:

PEview - C:\Users\talas\Downloads\672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74\672ac1422873d7481b4d...

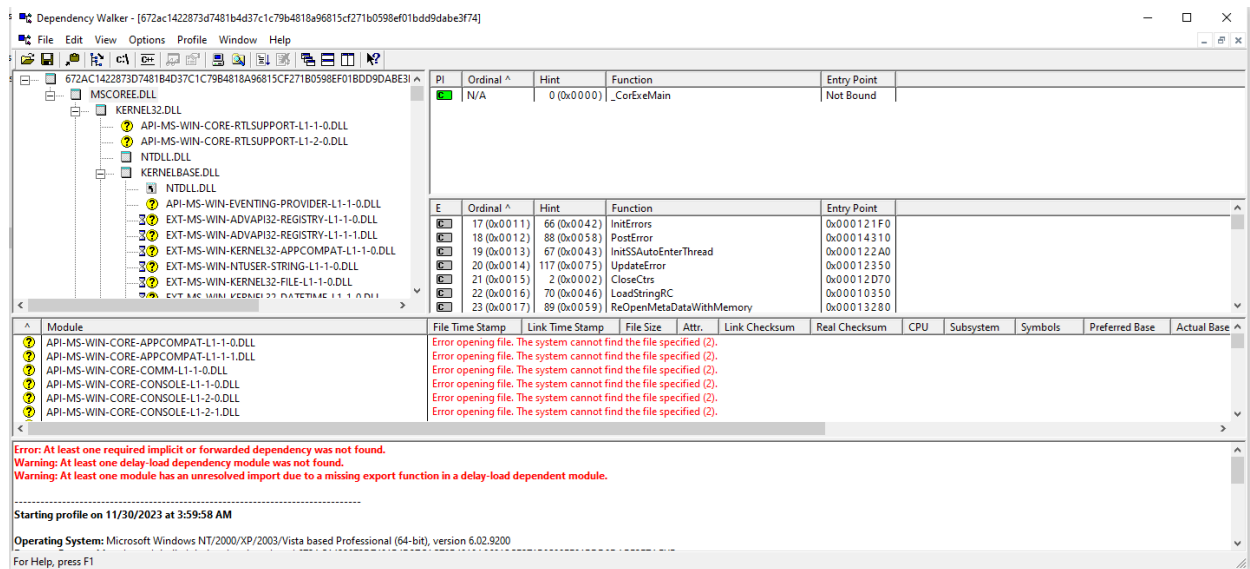
File View Go Help

672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74

	pFile	Raw Data	Value
IMAGE_DOS_HEADER	00000000	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00	MZ.....
IMAGE_DEBUG_TYPE	00000010	B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00@.....
MS-DOS Stub Program	00000020	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
IMAGE_NT_HEADERS	00000030	00 00 00 00 00 00 00 00 00 00 00 80 00 00 00
IMAGE_SECTION_HEADER	00000040	0E 1F BA 0E 00 B4 09 CD 21 B8 01 4C CD 21 54 68!..L..!Th
IMAGE_SECTION_HEADER	00000050	69 73 20 70 72 6F 67 72 61 6D 20 63 61 6E 6E 6F	is program canno
IMAGE_SECTION_HEADER	00000060	74 20 62 65 20 72 75 6E 20 69 6E 20 44 4F 53 20	t be run in DOS
SECTION .text	00000070	6D 6F 64 65 2E 0D 0D 0A 24 00 00 00 00 00 00	mode...\$.....
SECTION .rsrc	00000080	50 45 00 00 4C 01 03 00 71 6D EC EC 00 00 00 00	PE...L...qm.....
SECTION .reloc	00000090	00 00 00 00 E0 00 02 01 0B 01 30 00 00 80 06 000.....
	000000A0	00 08 00 00 00 00 00 00 2A 9F 06 00 00 20 00 00*.....
	000000B0	00 A0 06 00 00 00 40 00 00 20 00 00 00 02 00 00@.....
	000000C0	04 00 00 00 00 00 00 00 04 00 00 00 00 00 00
	000000D0	00 E0 06 00 00 02 00 00 00 00 00 02 00 40 85@.....
	000000E0	00 00 10 00 00 10 00 00 00 00 10 00 00 10 00
	000000F0	00 00 00 00 10 00 00 00 00 00 00 00 00 00 00
	00000100	D8 9E 06 00 4F 00 00 00 00 A0 06 00 AC 05 00 00O.....
	00000110	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00000120	00 C0 06 00 0C 00 00 00 70 84 06 00 70 00 00 00p...p.....
	00000130	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00000140	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00000150	00 00 00 00 00 00 00 00 00 20 00 00 08 00 00
	00000160	00 00 00 00 00 00 00 00 08 20 00 00 48 00 00 00H.....
	00000170	00 00 00 00 00 00 00 00 2E 74 65 78 74 00 00 00text.....
	00000180	48 7F 06 00 00 20 00 00 00 80 06 00 00 02 00 00	H.....
	00000190	00 00 00 00 00 00 00 00 00 00 00 00 20 00 00 60
	000001A0	2E 72 73 72 63 00 00 00 AC 05 00 00 A0 06 00 00	..rsrc.....
	000001B0	00 06 00 00 00 82 06 00 00 00 00 00 00 00 00
	000001C0	00 00 00 00 40 00 00 40 2E 72 65 6C 6F 63 00 00@...@reloc..
	000001D0	0C 00 00 00 00 C0 06 00 00 02 00 00 00 88 06 00
	000001E0	00 00 00 00 00 00 00 00 00 00 00 00 40 00 42@...B.....
	000001F0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
	00000200	0C 9F 06 00 00 00 00 00 48 00 00 00 02 00 05 00H.....

Viewing 672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74.exe

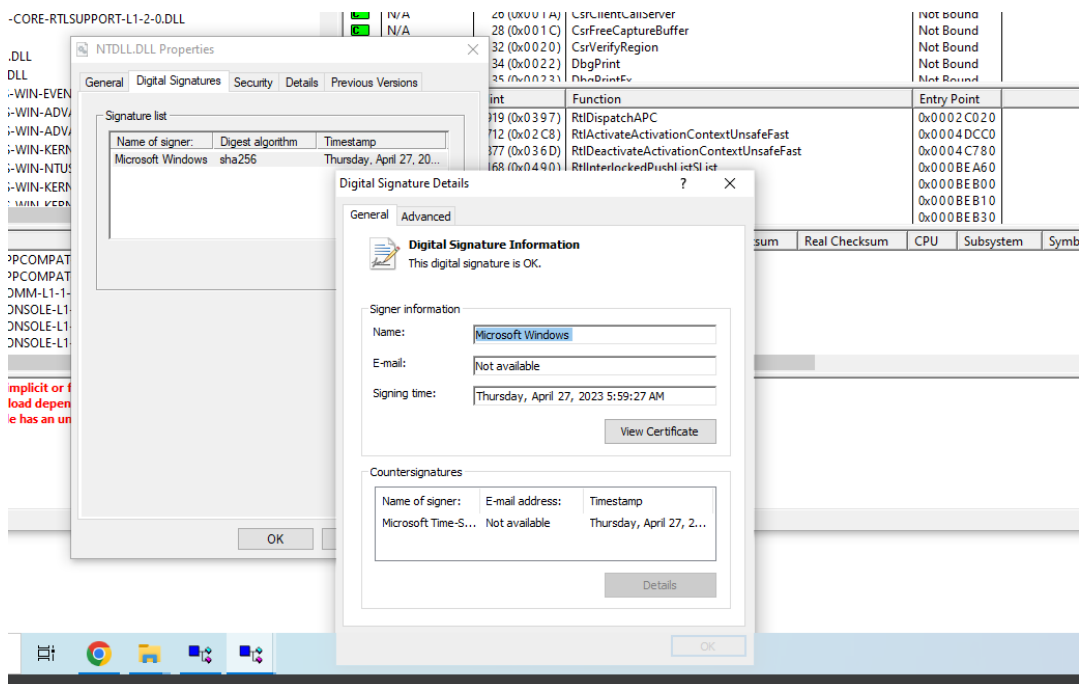
■ Dependency Walker:



*note: these errors are just compatibility errors, they are not indicators of malicious activity.

As shown above, the file uses a lot of .dll modules, mainly *mscorlib.dll*: a critical component of the CLR (Microsoft Common Language Runtime), which is responsible for managing the execution of .NET applications, and *kernel32.dll*.

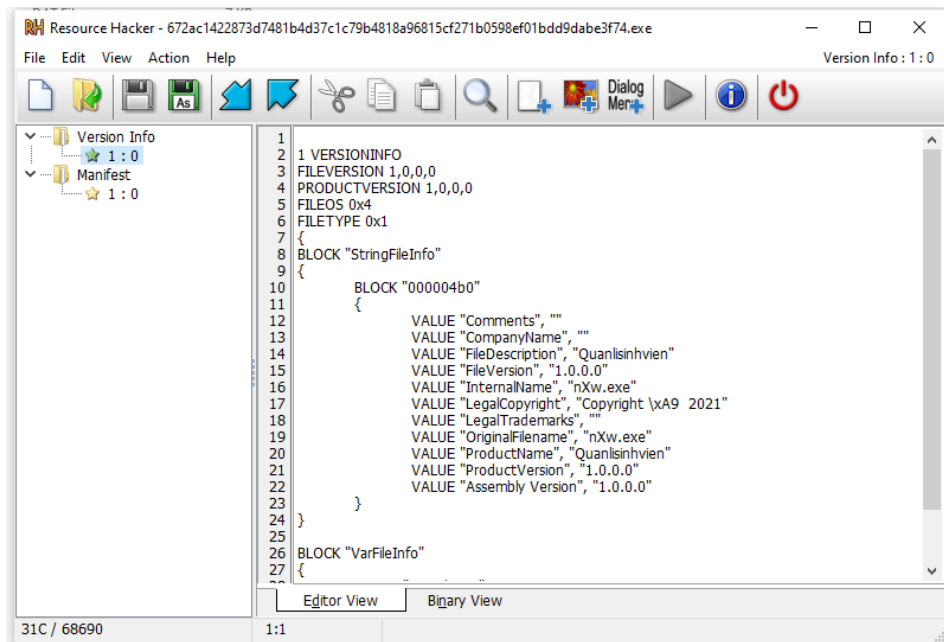
We used dependency walker to check for unusual behavior in these modules, typically when a module is verified it will give us something like this (*NTDLL.DLL*):



However, this signature was not available in a lot of modules, a lot were unsigned which indicates that the file is illegitimate.

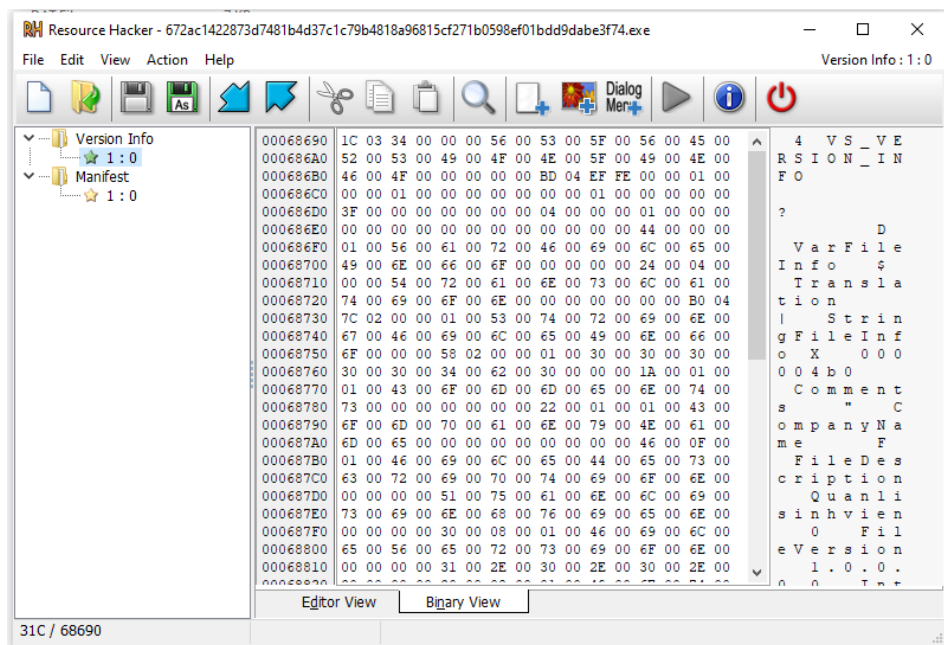
5. Resource analysis:

▪ Resource Hacker:



Version info seems to be standard, the only suspicious thing concluded here is the FileDescription: Quanlinsinhvien, which after some searching it seems it's not a real word, so it could be an indicator but still, information provided about this file is very vague. Manifest file also contained nothing particularly suspicious.

Hexadecimal values:

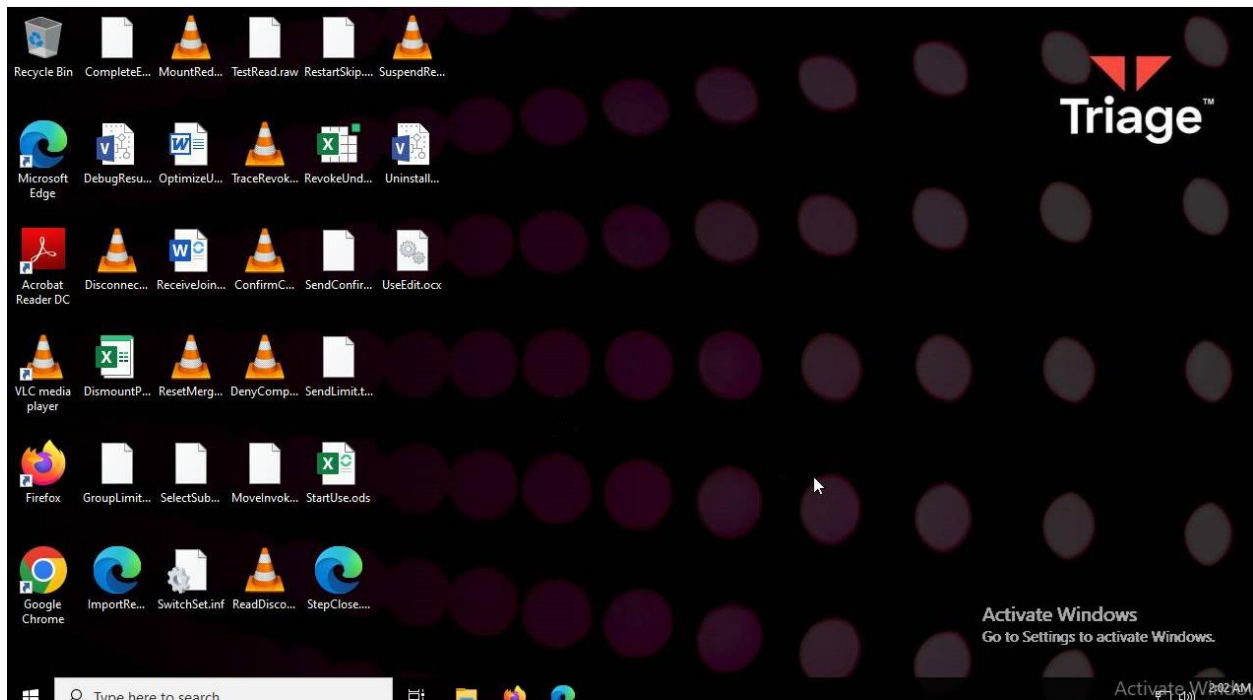


d. Dynamic analysis phase:

▪ Free sandbox tool:

Unfortunately, free versions of sandboxing tools are very limited in what they offer. We tried to use *Any.Run* but the only free operating system it supported was Windows 7. *JoeSandbox* also did not approve our account.

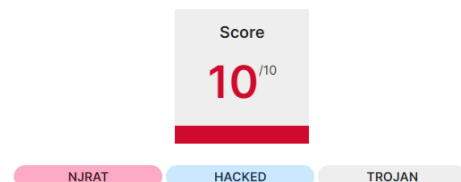
We ended up using Recorded Future's Triage sandbox tool (<https://tria.ge/>). This tool basically opens the executable and gives a full report of its behavior in the sandbox, any changes occurring after executing, what kind of malicious content the file contains... etc. This is what the sandbox looks like (we chose Windows 10 64bit):



After the execution was complete, this is what it reported:

General

Target	672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74.exe	
Size	418KB	
MD5	e3ff1e89d54c3469f11b3340c5d83f12	
SHA1	1ce54890e5f1a9076c54464527d88cfb8931b88	
SHA256	672ac1422873d7481b4d37c1c79b4818a96815cf271b0598ef01bdd9dabe3f74	
SHA512	5814c49ba4e912e71ba7d3d547eefe6cfe485899b8394fd45dad6616a743b55c3a75af9a3e499c936d4bf38b4b08440b7bc20b88ea020769a1b01b9207004c05	
SSDEEP	6144:poKALTom4JkwXIGRL0YbtYf0/47nIX2pMx48yJ15EVToyENhVUPE/f7/QMAezXEZ:pGoB1EL4fD71X2pI44WNhVUPU/1AOzK	



Clearly a very malicious file.

Malware Config

Extracted

Family	njrat
Version	v2.0
Botnet	HacKed
C2	01.92.240.141:5577
Mutex	Windows
Attributes	reg_key Windows

splitter
~F~

System is compromised and part of a botnet.

Signatures

Discovery Execution Persistence Privilege Escalation

njRAT/Bladabindi
Widely used RAT written in .NET.

NJRAT TROJAN

Checks computer location settings • 2 TTPs 1 IoCs
Looks up country code configured in the registry, likely geofence.

Drops startup file • 1 IoCs

Suspicious use of SetThreadContext • 1 IoCs

Enumerates physical storage devices • 1 TTPs
Attempts to interact with connected storage/optical drive(s).

Checks processor information in registry • 2 TTPs 3 IoCs
Processor information is often read in order to detect sandboxing environments.

Creates scheduled task(s) • 1 TTPs 1 IoCs
Schtasks is often used by malware for persistence or to perform post-infection execution.

PERSISTENCE

Enumerates system info in registry • 2 TTPs 3 IoCs

Suspicious behavior: AddClipboardFormatListener • 2 IoCs

Suspicious behavior: EnumeratesProcesses • 4 IoCs

Suspicious behavior: GetForegroundWindowSpam • 1 IoCs

Suspicious use of AdjustPrivilegeToken • 31 IoCs

Suspicious use of FindShellTrayWindow • 14 IoCs

Suspicious use of SendNotifyMessage • 13 IoCs












Suspicious use of SetWindowsHookEx • 15 IoCs


Suspicious use of WriteProcessMemory • 14 IoCs

NJRAT: a type of Remote Access Trojan (RAT), a piece of malicious software that falls into the category of remote administration tools, but is used for malicious purposes, often without the knowledge or consent of the affected user. It provides backdoor functionalities and allows for data theft and surveillance.

Network

Requests TCP UDP

	DNS	136.32.126.40.in-addr.arpa	▼
	DNS	9.228.82.20.in-addr.arpa	▼
	DNS	39.142.81.104.in-addr.arpa	▼
	DNS	55.36.223.20.in-addr.arpa	▼
	DNS	86.23.85.13.in-addr.arpa	▼
	DNS	18.31.95.13.in-addr.arpa	▼
	DNS	107.175.53.84.in-addr.arpa	▼
	DNS	21.236.111.52.in-addr.arpa	▼
	DNS	46.28.109.52.in-addr.arpa	▼
	DNS	170.117.168.52.in-addr.arpa	▼
	DNS	169.117.168.52.in-addr.arpa	▼



- **ProcessMonitor:**

The screenshot displays the Process Monitor application window. The title bar reads "Process Monitor - Sysinternals www.sysinternals.com". The menu bar includes File, Edit, Event Filter, Tools, Options, and Help. Below the menu is a toolbar with icons for various actions like saving, printing, and filtering. The main area shows a log of events with columns for Time, Process Name, PID, Operation, Path, Result, and Detail.

Time	Process Name	PID	Operation	Path	Result	Detail
00:26	Cmsedge.exe	5016	CreateFile	C:\Users\talos\AppData\Local\Microsoft\...	SUCCESS	
00:26	svchost.exe	320	RegOpenKey	HKCU\Software\Classes\AppID\{SE17...}	NAME NOT FOUND Desired Access: M...	
00:26	svchost.exe	320	RegQueryValue	HKCR/AppID/{9E175B9C-F52A-11D8...}	SUCCESS	Query: Name
00:26	svchost.exe	320	RegOpenKey	HKCR/AppID/{9E175B9C-F52A-11D8...}	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	320	RegOpenKey	HKCU\Software\Classes\AppID\{SE17...}	NAME NOT FOUND Desired Access: M...	
00:26	svchost.exe	320	RegQueryValue	HKCR/AppID/{9E175B9C-F52A-11D8...}	NAME NOT FOUND Length: 16	Query: Name
00:26	svchost.exe	320	RegQueryKey	HKCR/AppID/{9E175B9C-F52A-11D8...}	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	320	RegOpenKey	HKCU\Software\Classes\AppID\{SE17...}	NAME NOT FOUND Desired Access: M...	
00:26	svchost.exe	320	RegQueryValue	HKCR/AppID/{9E175B9C-F52A-11D8...}	NAME NOT FOUND Length: 16	Query: Name
00:26	svchost.exe	320	RegQueryKey	HKCR/AppID/{9E175B9C-F52A-11D8...}	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	5016	CreateFile	C:\Users\talos\AppData\Local\Microsoft\...	SUCCESS	Desired Access: R...
00:26	svchost.exe	320	RegOpenKey	HKCU\Software\Classes\AppID\{SE17...}	NAME NOT FOUND Desired Access: M...	
00:26	svchost.exe	320	RegQueryValue	HKCR/AppID/{9E175B9C-F52A-11D8...}	NAME NOT FOUND Length: 16	
00:26	svchost.exe	320	RegCloseKey	HKCR/AppID/{9E175B9C-F52A-11D8...}	SUCCESS	
00:26	svchost.exe	320	RegQueryKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	Query: Name
00:26	Cmsedge.exe	5016	QueryVerbInfo2	C:\Users\talos\AppData\Local\Microsoft\...	SUCCESS	Attributes: A Repa...
00:26	svchost.exe	320	RegQueryKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	Query: Hande Tag...
00:26	Cmsedge.exe	5016	SetDisposition...	C:\Users\talos\AppData\Local\Microsoft\...	SUCCESS	Flags: FILE_DISP...
00:26	svchost.exe	320	RegOpenKey	HKCU\Software\Classes\Wow6432No...	NAME NOT FOUND Desired Access: Q...	
00:26	svchost.exe	320	RegQueryKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	320	RegOpenKey	HKCR\Wow6432Node\CLSID\{7D09...}	NAME NOT FOUND Desired Access: Q...	
00:26	Cmsedge.exe	5016	CloseFile	C:\Users\talos\AppData\Local\Microsoft\...	SUCCESS	
00:26	svchost.exe	320	RegQueryKey	HKCR	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	320	RegQueryKey	HKCR	SUCCESS	Query: Name
00:26	svchost.exe	320	RegOpenKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	Desired Access: R...
00:26	svchost.exe	320	RegSetInfo	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	KeySetInformation...
00:26	svchost.exe	320	RegQueryKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	Query: Hande Tag...
00:26	svchost.exe	320	RegOpenKey	HKCR\Wow6432Node\CLSID\{7D09...}	NAME NOT FOUND Desired Access: R...	
00:26	svchost.exe	320	RegCloseKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	
00:26	svchost.exe	320	RegCloseKey	HKCR\Wow6432Node\CLSID\{7D09...}	SUCCESS	
00:26	svchost.exe	320	RegCloseKey	HKCU\Software\Classes	SUCCESS	

Showing 7,015,869 of 10,917,200 events (64%) Backed by virtual memory

ProcMon captures all events that occur on a file in real-time; for example, if a program is running it captures every single small thing the program does. As we can see from the screenshot, we can even monitor when the file was opened and closed, if that process was successful or not, if it wasn't successful it gives us the reasons it wasn't. Another feature is that it allows you save whatever you want and filter what exactly you want to capture.

For example, when we ran the file on PView and monitored the behavior on ProcMon, this is what came up:

Time	Process Name	PID	Operation	Path	Result	Detail
20:44...	PView.exe	9784	RegOpenKey	HKCU	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Microsoft\Windows\...	SUCCESS	Desired Access: Q...
20:44...	PView.exe	9784	RegSetInfoKey	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	KeySetInformation...
20:44...	PView.exe	9784	RegQueryValue	HKCU\SOFTWARE\Microsoft\Window...	NAME NOT FOUND	Length: 16
20:44...	PView.exe	9784	RegCloseKey	HKCU\SOFTWARE\Microsoft\Window...	SUCCESS	
20:44...	PView.exe	9784	RegOpenKey	HKLM	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKLM	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKLM\Software\WOW6432Node\Micr...	SUCCESS	Desired Access: Q...
20:44...	PView.exe	9784	RegQueryValue	HKLM\SOFTWARE\WOW6432Node\...	NAME NOT FOUND	Length: 16
20:44...	PView.exe	9784	RegOpenKey	HKLM\SOFTWARE\WOW6432Node\...	SUCCESS	
20:44...	PView.exe	9784	RegOpenKey	HKLM	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKLM\Software\Microsoft\Windows\...	SUCCESS	Desired Access: R...
20:44...	PView.exe	9784	RegSetInfoKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	KeySetInformation...
20:44...	PView.exe	9784	RegOpenKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	Query: Cached, Su...
20:44...	PView.exe	9784	RegCloseKey	HKLM\SOFTWARE\Microsoft\Window...	SUCCESS	
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes\WOW6432No...	NAME NOT FOUND	Desired Access: R...
20:44...	PView.exe	9784	RegOpenKey	HKCR\WOW6432Node\CLSID\{F324E...	NAME NOT FOUND	Desired Access: R...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes\WOW6432No...	NAME NOT FOUND	Desired Access: R...
20:44...	PView.exe	9784	RegOpenKey	HKCR\WOW6432Node\CLSID\{F324E...	NAME NOT FOUND	Desired Access: R...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: HandleTag...
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes	SUCCESS	Query: Name
20:44...	PView.exe	9784	RegOpenKey	HKCU\Software\Classes\WOW6432No...	NAME NOT FOUND	Desired Access: Q...
20:44...	PView.exe	9784	RegOpenKey	HKCR\WOW6432Node\CLSID\{F324E...	SUCCESS	Desired Access: Q...

Showing 4,912,055 of 5,914,716 events (83%) Backed by virtual memory

■ Process Explorer:

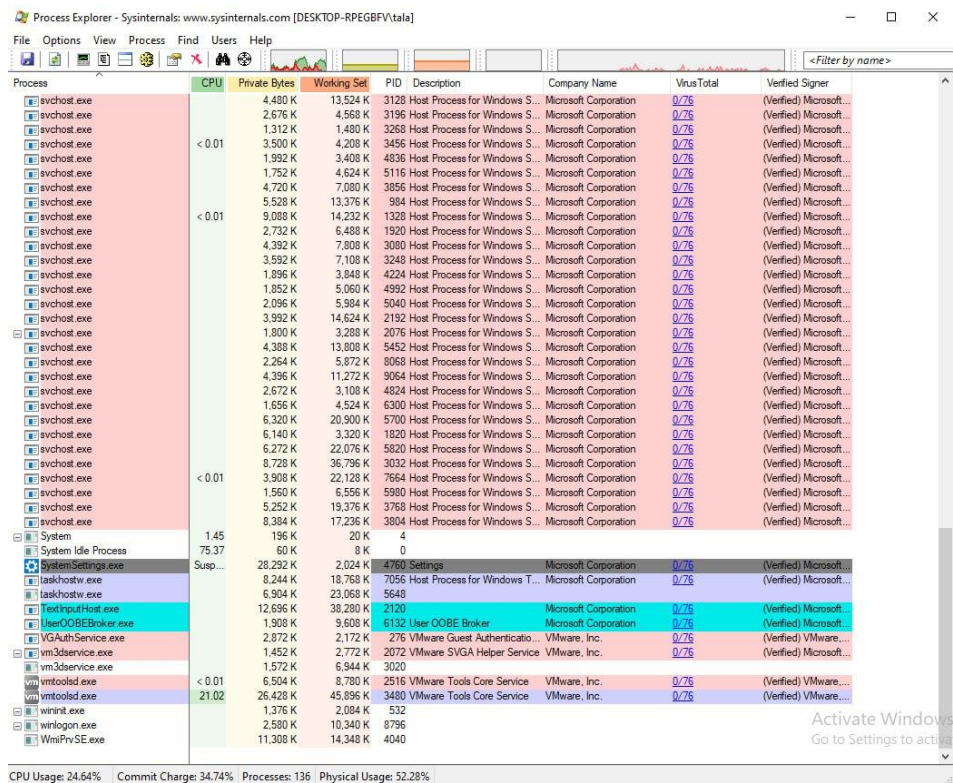
Process	CPU	Private Bytes	Working Set	PID	Description	Company Name
Registry		16,312 K	79,060 K	92		
System Idle Process	< 0.01	60 K	8 K	0		
System	3.62	192 K	44 K	4		
Interrupts	7.25	0 K	0 K	n/a	Hardware Interrupts and DPCs	
smss.exe		1,056 K	672 K	320		
Memory Compression	0.72	1,900 K	498,064 K	1840		
csrss.exe	< 0.01	1,824 K	4,896 K	440		
wininit.exe		1,368 K	6,108 K	524		
services.exe		5,016 K	8,464 K	664		
svchost.exe	< 0.01	12,112 K	29,612 K	784	Host Process for Windows S...	Microsoft Corporation
WmiPrvSE.exe	1.45	11,960 K	17,864 K	3792		
MoUsCoreWorker.exe		14,364 K	21,080 K	2160		
TlWorker.exe		284,560 K	240,540 K	1972		
StartMenuExperienceHost.exe		28,908 K	38,752 K	6384		
backgroundTaskHost.exe		22,500 K	44,820 K	6496	Background Task Host	Microsoft Corporation
RuntimeBroker.exe		6,960 K	23,968 K	6532	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		12,104 K	27,672 K	6620	Runtime Broker	Microsoft Corporation
SearchApp.exe		142,620 K	42,956 K	6672	Search application	Microsoft Corporation
RuntimeBroker.exe		8,908 K	12,136 K	6940	Runtime Broker	Microsoft Corporation
SkypeBackgroundHost.exe		1,984 K	1,528 K	7228	Microsoft Skype	Microsoft Corporation
SkypeApp.exe		148,556 K	6,344 K	7272	SkypeApp	Microsoft Corporation
RuntimeBroker.exe		7,216 K	9,364 K	7616	Runtime Broker	Microsoft Corporation
RuntimeBroker.exe		2,120 K	2,356 K	8116	Runtime Broker	Microsoft Corporation
FileCoAuth.exe		6,692 K	21,784 K	7408	Microsoft OneDriveFile Co-A...	Microsoft Corporation
TextInputHost.exe		12,704 K	23,504 K	7024		
ApplicationFrameHost.exe		14,776 K	45,848 K	7416	Application Frame Host	Microsoft Corporation
WinStoreApp.exe	Susp...	96,756 K	1,876 K	2440	Store	Microsoft Corporation
RuntimeBroker.exe		11,920 K	5,340 K	9144	Runtime Broker	Microsoft Corporation
ShellExperienceHost.exe		17,988 K	22,604 K	3544	Windows Shell Experience H...	Microsoft Corporation

CPU Usage: 99.99% Commit Charge: 84.85% Processes: 161 Physical Usage: 84.19%

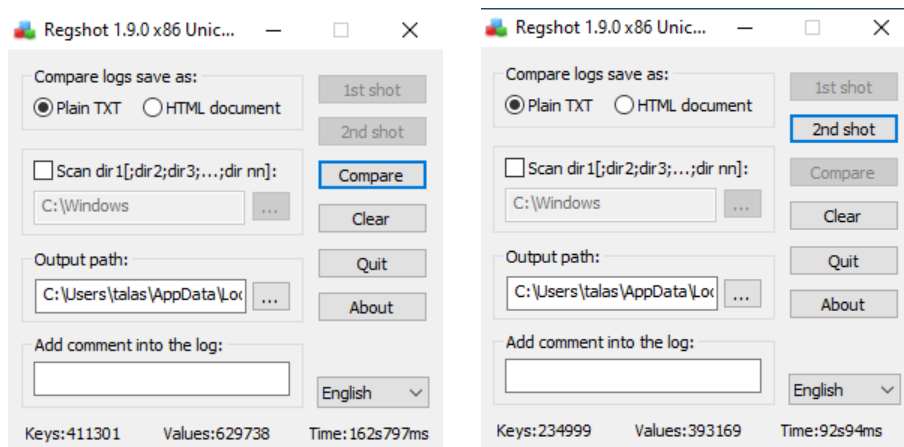
Provides a list of all processes running on the system, helps find out things like the amount of CPU and memory being used. If you are worried about a process, you can easily kill it.

When you click on any process you can see the following things:

- Path
- Command line
- Threads that are associated with it
- TCP/IP (network connections)
- Who owns it
- Strings associated with it



■ RegShot:





```
File Edit Format View Help
Regshot 1.9.0 x86 Unicode
Comments:
Datetime: 2023/11/29 20:15:45 , 2023/11/29 20:36:15
Computer: DESKTOP-RPEGBFV , DESKTOP-RPEGBFV
Username: tala , tala

-----
Keys deleted: 2
-----
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\ServiceInstances\{e53e84be-41d8-4b7f-9d7d-b8d77467d1ed}

-----
Keys added: 176304
-----
HKLM\SYSTEM\ControlSet001\Services\VSS\Diag\SwProvider_{b5946137-7b9f-4925-af80-51abd60b20d5}
HKLM\SYSTEM\ControlSet001\Services\VSS\Diag\VssapiPublisher
HKLM\SYSTEM\CurrentControlSet\Services\VSS\Diag\SwProvider_{b5946137-7b9f-4925-af80-51abd60b20d5}
HKLM\SYSTEM\CurrentControlSet\Services\VSS\Diag\VssapiPublisher
HKLM\SOFTWARE
HKLM\SOFTWARE\Intel
HKLM\SOFTWARE\Intel\PSIS
HKLM\SOFTWARE\Intel\PSIS\PSIS_DECODER
HKLM\SOFTWARE\Microsoft
HKLM\SOFTWARE\Microsoft\NETFramework
HKLM\SOFTWARE\Microsoft\NETFramework\Advised
HKLM\SOFTWARE\Microsoft\NETFramework\Advised\Policy
HKLM\SOFTWARE\Microsoft\NETFramework\Advised\Policy\AppPatch
HKLM\SOFTWARE\Microsoft\NETFramework\Advised\Policy\AppPatch\v2.0.50727.00000
HKLM\SOFTWARE\Microsoft\NETFramework\Advised\Policy\AppPatch\v2.0.50727.00000\BTSNTSvc.exe
HKLM\SOFTWARE\Microsoft\NETFramework\Advised\Policy\AppPatch\v2.0.50727.00000\BTSNTSvc.exe\{CA109828-7CE7-40F4-A073-C7575455A7D5}
```

It shows how making a small change will make a big difference. When clicking the first shot it scans the system, then when you click the second shot it scans again and if any change has happened to the system it will detect it and see how many keys were added and how many deleted, and will calculate the total changes. In our case it is 413001 changes in total. The change we made to the system is: check the show/hide [file name extensions] option.

Insights, conclusions, and recommendations:

After using so many tools, here is what we concluded:

- If you are a beginner in malware analysis, tools with a GUI, and online tools are a good way to start. VirusTotal offers a pretty concise report about infected files, it seemed like all the static analysis tools we used just confirmed everything that VirusTotal reported. If you want a general look into the file, we would recommend it. But if you want a closer look into the specifics of this indicative information, you will have to use some PE viewers to really analyze how a file is structured.
- Hash calculators, and even text viewers to an extent are a great way to look into a file, but they are not enough to determine if a file is infected, additional information is almost always required.
- Dynamic analysis will reveal a lot of things about the file static analysis fails to find. The sandbox tool we used really made a big difference and presented a lot of information that would have otherwise remained unknown. Although they typically need more processing power, and the advanced sandboxes are usually paid, we see why they are so essential to malware analysis, as they give critical insight into files that no other tools can provide.

***Reverse Engineering, Assignment#1**

***Group members:**

- Daniah Imad Al-Gburi, ID: 134347

(VirusTotal, MD5Deep, Strings, LordPE, ResourceHacker, Sandbox)

- Tala Saleh, ID: 153217

(BinText, PEiD, PEview, ProcMon, Process Explorer, RegShot)

- Farah Juneidy, ID: 155270

(Dependency Walker)

- Batool Afeef, ID: 153492

(HashCalc)

***References:**

- *PE Format*: <https://learn.microsoft.com/en-us/windows/win32/debug/pe-format#machine-types>

- *File Entropy*: <https://practicalsecurityanalytics.com/file-entropy/>