



MALWARE ANALYSIS

Date: May 2025

Prepared By: Sahithi Gudigopuram

Table of Contents

S.No	Title	Page Numbers
1.	Executive Summary	04
2.	Introduction	04
3.	Malware Sample Information	04
4.	Architecture Diagram	05
5.	Modules & Submodules	06
6.	Environmental Setup	07
6.1	Virtual Machine Configuration	07
6.2	Security Configurations	07
6.3	FLARE VM Installation	08
7.	Malware Acquisition	09
8.	Static Analysis	11
8.1	PEStudio	12
8.2	DIE	13
8.3	IDA Pro	14
8.4	String Analysis	14
8.5	VirusTotal & Sandbox Reports	15
9.	Reverse Engineering with Ghidra	18
10.	Dynamic Analysis	20
10.1	Process Monitor	21
10.2	Wireshark	23
10.3	Registry Monitoring (Regshot)	23

11.	IOC Extraction	26
12.	YARA Rules	27
13.	Mitigation Techniques	30
14.	Conclusion	32

1. Executive Summary

This report presents the analysis and obfuscation of a malware sample obtained from MalwareBazaar for research purposes. The sample, a Windows 9d526e61fd8cb07b78456e4a1eb75c4ebbc94c6aad3e0c572818846f0f.exe file, was examined in an isolated FLARE VM environment using static and dynamic analysis tools to understand its behaviour and capabilities. Key actions included file system changes and potential network communication.

Following the initial analysis, the sample was obfuscated using packing techniques to simulate real-world evasion strategies. Detection challenges were evaluated, and YARA rules were created to identify both the original and obfuscated versions. The project highlights the importance of malware analysis in developing effective detection and defence mechanisms.

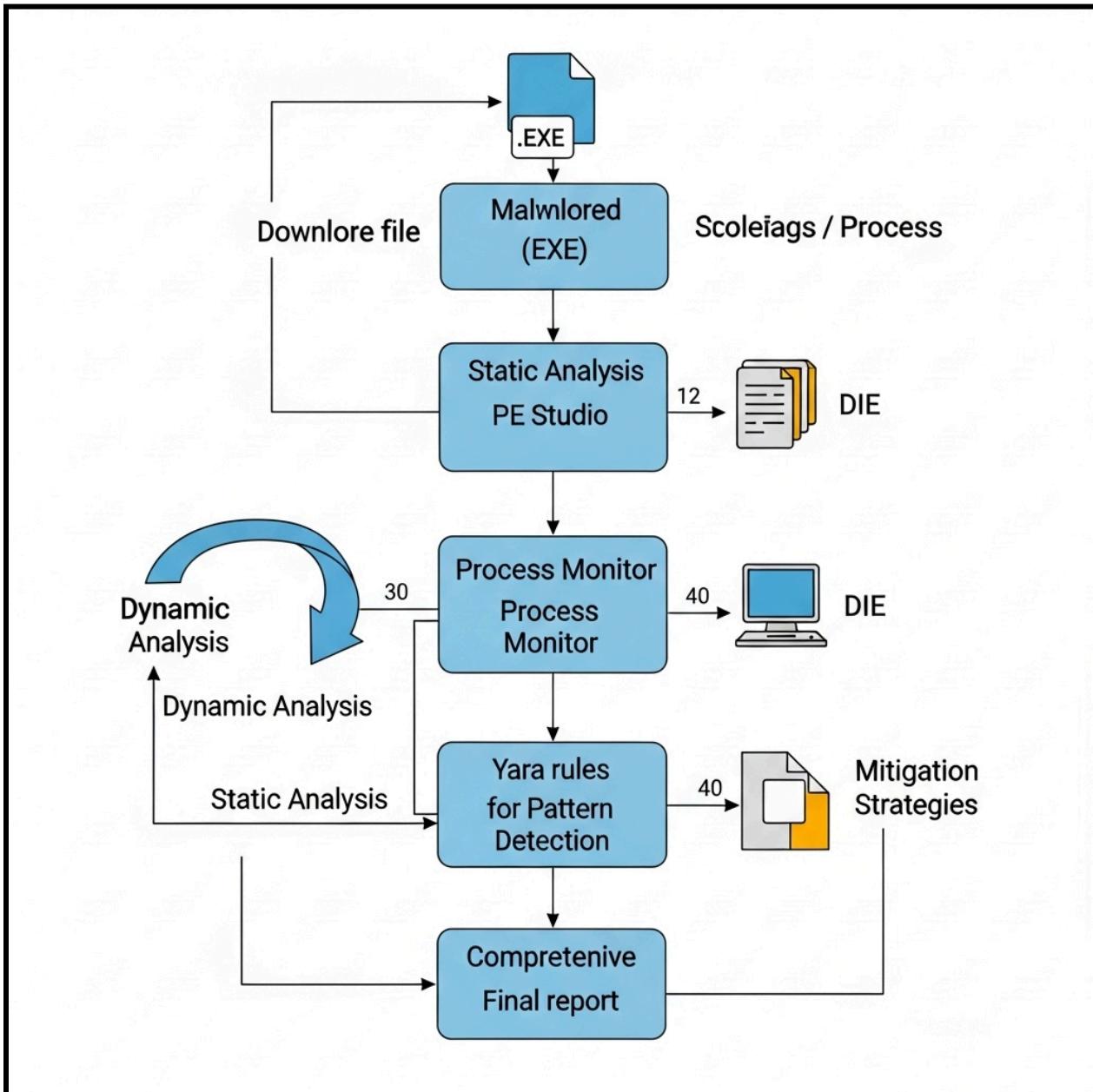
2. Introduction

This report presents a technical study focused on malware analysis using a controlled virtual lab environment. The objective is to examine the internal behaviour of a malware sample and understand how obfuscation techniques affect detection. The analysis is conducted using a combination of static, dynamic, and reverse engineering methods.

3 . Malware Sample Information

1. File Name: 9d526e61fd8cb07b78456e4a1eb75c4ebbc94c6aad3e0c572818846f0f.exe
2. Malware: ObfusTrojan
3. File Type: Executable (32-bit)
4. Size: Approximately 1.14 MB
5. MD5 Hash: af8c8f0a9e18595e7d217e5dad07835
6. SHA1 Hash: bc5982a98cd6a0a6986acd6e9b86735ecb6c44fa
7. SHA256 Hash:
9D526E61FD8CB07B78456E4A1EB75C4EBBC94C6AAD3E0C572818846F0F
8. Analysis Date: June 15 2025
9. Debug Date: June 01 2025
10. Detected Platform: Windows x64
11. Entropy (Sections with high entropy): 7.121
 - .text section: 6.680
 - .rdata section: 5.770
 - .data section: 2.003

4. Architecture Diagram



5. Modules & Sub-modules

Modules	Sub Modules	Tools
Malware Collection	Source Identification	N/A
Environment Setup	VM setup, Network Isolation	VM Ware
Static Analysis	File header analysis, String Extraction	PEstudio, DIE, IDA pro, Virtual Studio Analysis
Dynamic Analysis	Process monitoring, Network Monitoring, Registry/File Monitoring	Process Monitor, Regshot
Behavioural Analysis	IoC Extraction	Manual+log
YARA Rules	Rule Creation & Testing	Yara
Mitigation	Strategy & Recommendations	How to stop or contain the malware

6. Environmental Setup

6.1 Virtual Machine Configuration

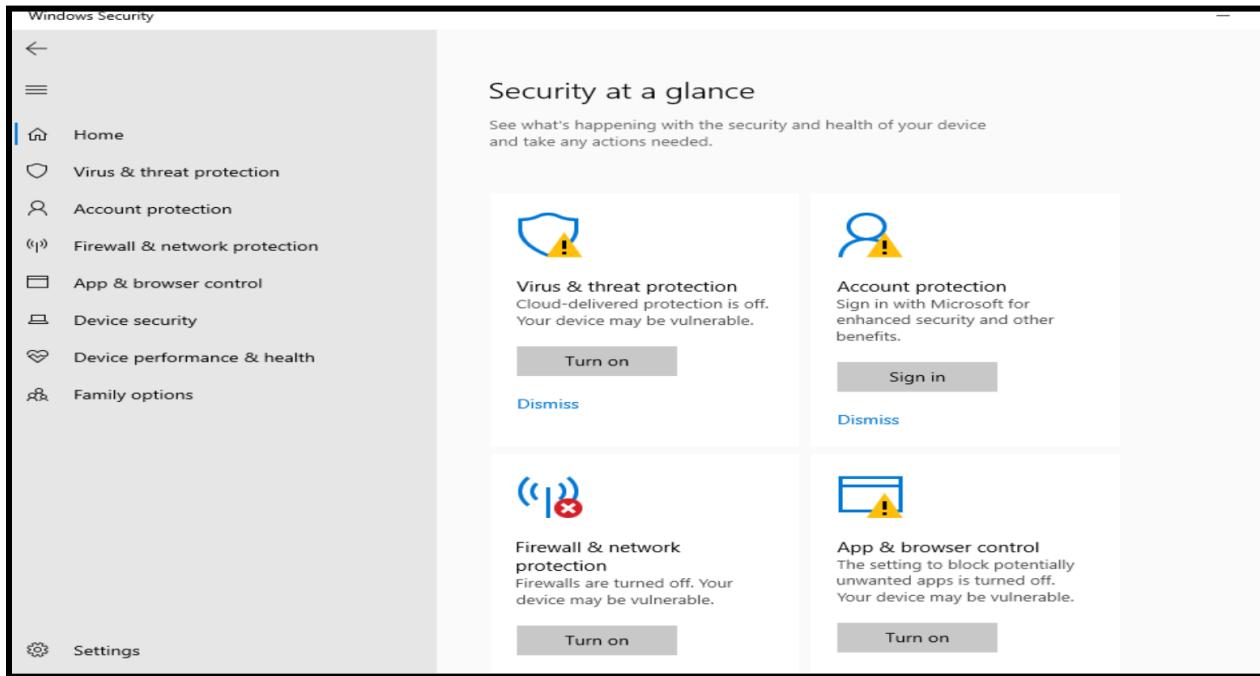
Component	Details
Virtualisation Tool	VMware
Guest OS	Windows 10(64-bit)
RAM	8GB
CPU Cores	2
Network Mode	Bridge Adapter(for setup), then switched to NAT
Snapshot	Created before execution for rollback safety

6.2 Security Configuration

1. **Windows Defender:** Press Win + R → type gpedit.msc → Enter → Computer Configuration → Administrative Templates → Windows Components → Microsoft Defender Antivirus. Now we need to double click and turn off “ Microsoft Defender Antivirus.
2. **Real-time Protection:** Open Windows Security → Go to Virus & Threat Protection → Manage settings → Toggle Real-time protection to Off.
3. **Firewall & Auto Updates:** Open Control Panel → Windows Defender Firewall, we need to click Windows Defender files on or off.
4. **Restore Point / Snapshot:** Created

6.3 Flare VM setup

1. Download the FLARE VM installer from: "<https://github.com/mandiant/flare-vm>"
2. Run with Powershell “ **Set-ExecutionPolicy Unrestricted**
.\install.ps1 ”
3. During the installation, we will see a number of reboots in our system



7. Malware Acquisition

For this analysis, a sample of the FormBook malware was acquired from a trusted malware repository. FormBook is a well-known infostealer malware used in real-world attacks to harvest credentials, keystrokes, and browser data.

1. Go to Malwarebazaar to download the file "<https://bazaar.abuse.ch/>"

The screenshot shows the Malwarebazaar homepage. At the top, there's a yellow banner with the text "NEW | Hunt across all abuse.ch platforms with one simple query - discover if an IPv4 address, domain, URL or file hash has been identified on any platform from a centralized search tool. Test it out here [hunting.abuse.ch](#) - and happy hunting! 🌐". Below the banner, the header includes the "MALWARE bazaar" logo, navigation links for "Browse", "Upload", "Hunting Alerts", "Access Data", "FAQ", "About", and "Login", and a note about browsing the malware sample database. The main content area features three cards: "Submissions (past 24 hours)" with 518 entries, "Most seen malware family (past 24 hours)" with Mirai, and "Malware samples in corpus" with 935'457 entries. Below these cards is a search form with fields for "See search syntax see below, example: tag:TrickBot", "Search Syntax ⓘ", and a "Search" button. There's also a "Search:" field at the bottom right.

2. I downloaded a malware.

NEW | Hunt across all abuse.ch platforms with one simple query - discover if an IPv4 address, domain, URL or file hash has been identified on any platform from a centralized search tool. Test it out here hunting.abuse.ch - and happy hunting!

MALWARE bazaar

from ABUSE.ch | SPAMHAUS

 [Browse](#)  [Upload](#)  [Hunting Alerts](#)  [Access Data](#)  [FAQ](#)  [About](#)  [Login](#)



Formbook



Vendor detections: 11

Intelligence (11)	IOCs	YARA (8)	File information	Comments	 Actions
SHA256 hash: 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f					 Download sample  Add tag  Delete this sample  Report a False Positive
SHA3-384 hash: d5ebceed7c545729c41793ff2f6c0356dc6f169b19170ff2ff13a386b5e7abbaab40da7acafa0c3d					
SHA1 hash: bc5982a98cd6a0a6986acd6e9b86735ecb6c44fa					
MD5 hash: af8c8f0a9e18595e7d217e5dadb07835					
humanhash: wyoming-lactose-stream-magnesium					

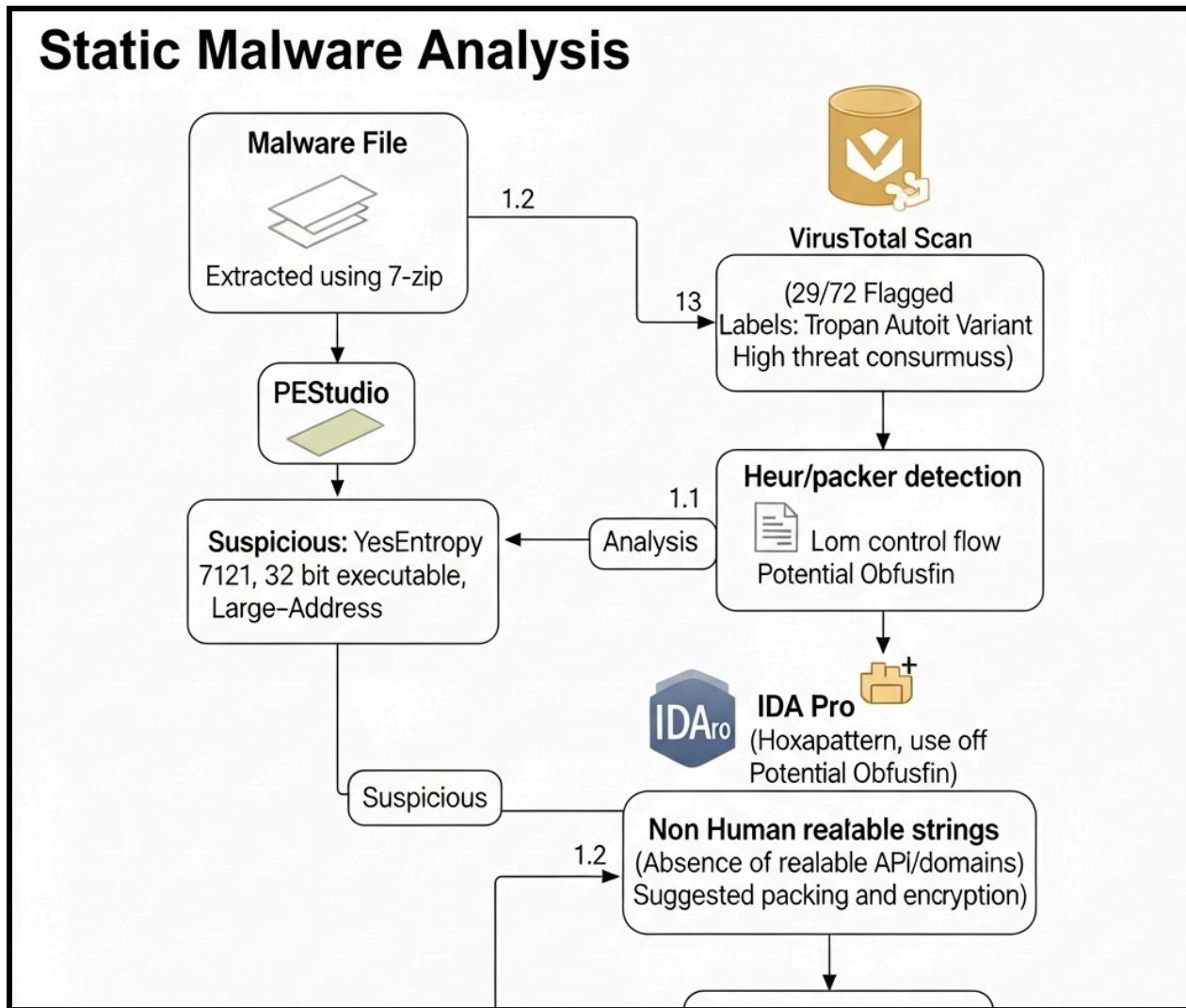
3. I used **7-Zip** to securely extract the password-protected malware archive without altering the original file.

Name	Date modified	Type	Size
ck access			
esktop			
downloads			
documents			
ictures			
st			
usic			
ideos			
eDrive			
: PC			
work			
✓ Today (2)			
✓ 9d526e61fd8c80cb07b78456e4a1e4b75...	6/16/2025 11:05 AM	ZIP File	737 KB
✓ 9d526e61fd8c80cb07b78456e4a1e4b75...	6/16/2025 6:05 PM	Application	1,163 KB
✓ Last month (3)			
✓ yara-v4.5.2-2023-wi...			
✓ AndroRat-test			1 KB
✓ desktop.ini			1 KB
✓ A long time ago (9)			
✓ .DS_Store			7 KB
✓ 7f31ab924bddd2f20...			373 KB
✓ 905e3f74e5dccca58cf...			382 KB
✓ ami.com			2 KB
✓ b34893e23666ab3d1...			416 KB
✓ ba0a74f2227e32fce...			114 KB
✓ d8823ee70109ce789...			193 KB
✓ yara64.exe			2,362 KB
✓ yarac64.exe			2,309 KB

8. Static Analysis

The malware was examined statically using tools like PEStudio, strings.exe, and IDA Pro to inspect headers, imports, and embedded resources.

Analysis revealed suspicious API calls, encoded strings, and entropy anomalies indicative of obfuscation or packing.



8.1 PEStudio

The malware sample has an entropy of 7.121, indicating moderate randomness and suggesting it may be a partially known variant employing obfuscation techniques, supported by a VirusTotal detection rate of 29/72. It is a valid 32-bit executable with large-address awareness, which does not rule out malicious intent. The optional headers reveal a well-constructed file with modern security features like ASLR and SEH, commonly used by malware to enhance stealth and stability. Certain anomalies in the file's structure indicate potential malicious behaviour that requires further analysis.

pestudio 9.61 - Malware Initial Assessment - www.winitor.com | c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...

property	value
file	9D526E61FD8C80CB07B78456E4A1E4B75C4EBBC94C6AAD3E0C572818846F0...
file > sha256	4D 5A 90 00 03 00 00 04 00 00 FF FF 00 00 B8 00 00 00 00 00 00 40 00 00 00 00 00 00 00 00 MZ.....@.....
file > first 32 bytes (hex)	size: 1190912 bytes, entropy: 7.121
file > first 32 bytes (text)	executable, 32-bit, GUI
file > info	n/a
file > type	n/a
file > version	E8 6A CE 00 00 E9 7F FE FF CC CC 57 56 8B 74 24 10 8B 4C 24 14 8B 7C 24 0C 8B C1 8B
entry-point > first 32 bytes (hex)	0x00025F74 (section[.text])
entry-point > location	Microsoft Linker 11.0 Visual Studio 2012 Autolt compiled script (2.XX-3.XX) Microsoft
file > signature	
stamps	
stamp > compiler	Sun Jun 15 23:35:31 2025 (UTC)
stamp > debug	Sun Jun 01 17:44:52 2014 (UTC)
stamp > resource	n/a
stamp > import	n/a
stamp > export	n/a
names	
file > name	c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...
debug > file	n/a
export	n/a
version	n/a
manifest	n/a
.NET > module > name	n/a
certificate > program-name	n/a

pestudio 9.61 - Malware Initial Assessment - www.winitor.com | c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...

File settings about

c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...

indicator (28)	detail
file > name	c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...
file > signature	AutoIt compiled script (2.XX-3.XX) Microsoft Linker 11.0 Microsoft Vis...
file > sha256	9D526E61FD8C80CB07B78456E4A1E4B75C4EBBC94C6AAD3E0C572818846f0...
file > info	size: 1190912 bytes, entropy: 7.121
file > type	executable 32-bit, GUI
section > file	signature: AutoIt, offset: 0x000BA410, size: 17904 bytes
virustotal > permalink	https://www.virustotal.com/gui/file/9d526e61fd8c80cb07b78456e4a1e4...
virustotal > scan-date	2025-06-16 14:35:42
virustotal > score	29/72
stamp > compiler	Sun Jun 15 23:35:31 2025
resource > file	signature: AutoIt compiled script (2.XX-3.XX), offset: 0x000C71B8, size: 3...
languages > names	English-UK neutral
resources > info	count: 26, size: 365664 bytes, file-ratio: 30.70%
manifest > general	name: n/a, description: n/a, severity: asInvoker
entry-point > location	0x00025F74 (section: .text)
string > url-pattern	255.255.255.255
certificate	n/a
libraries > flag	WSOCK32.dll (Windows Socket 32-Bit Library)
libraries > file	WINMM.dll (Windows Management Library)
libraries > flag	MPR.dll (Multiple Provider Router Library)
libraries > flag	WININET.dll (Internet Extensions for Win32 Library)
libraries > flag	PSAPI.DLL (DHCP Server API Stub Library)
libraries > flag	IPLPAPI.DLL (IP Helper API)
imports > ordinal > count	49
imports > flag	AddAcc AdjustTokenPrivileges AllocateAndInitializeSid AttachThread...
imphash > md5	7B58DE7A39F9282E413134A0FDB0C20D
exports	n/a
overlay	n/a

ha256 > 9D526E61FD8C80CB07B78456E4A1E4B75C4EBBC94C6AAD3E0C572818846F0F0...

cpu > 32-bit file > type > executable subsystem > GUI

pestudio 9.61 - Malware Initial Assessment - www.winitor.com | c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...

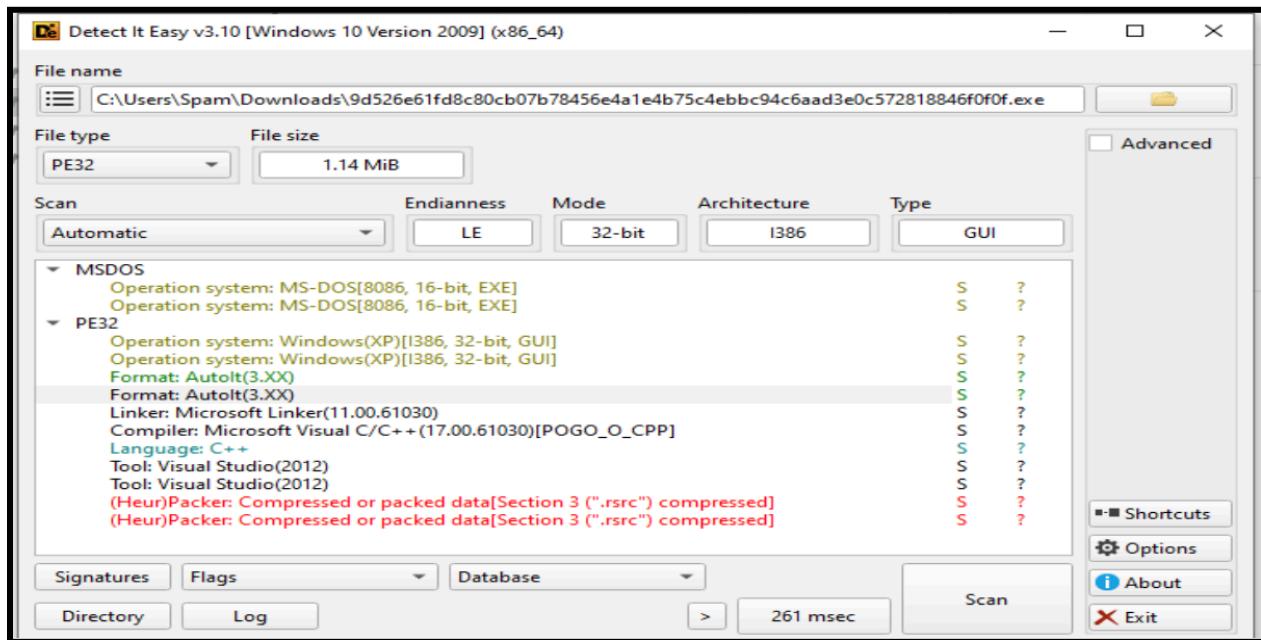
File settings about

c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...

imports (518)	flag (161)	type	ordinal	first-thunk (IAT)	first-thunk-origin
151 (_WSAFDIsSet)	x	implicit	x	0x80000097	0x800000097
16 (recv)	x	implicit	x	0x80000010	0x800000010
19 (send)	x	implicit	x	0x80000013	0x800000013
21 (setsockopt)	x	implicit	x	0x80000015	0x800000015
15 (ntohs)	x	implicit	x	0x8000000F	0x8000000F
17 (recvfrom)	x	implicit	x	0x80000011	0x800000011
18 (select)	x	implicit	x	0x80000012	0x800000012
115 (WSAStartup)	-	implicit	x	0x80000073	0x800000073
9 (htons)	x	implicit	x	0x80000009	0x80000009
1 (accept)	x	implicit	x	0x80000001	0x80000001
13 (listen)	x	implicit	x	0x8000000D	0x8000000D
2 (bind)	x	implicit	x	0x80000002	0x80000002
3 (closesocket)	x	implicit	x	0x80000003	0x80000003
4 (connect)	x	implicit	x	0x80000004	0x80000004
116 (WCACleanup)	x	implicit	x	0x80000074	0x80000074
10 (inet_addr)	x	implicit	x	0x8000000A	0x8000000A
20 (sendto)	x	implicit	x	0x80000014	0x80000014
111 (WSAGetLastError)	x	implicit	x	0x8000006F	0x8000006F
11 (inet_ntoa)	x	implicit	x	0x8000000B	0x8000000B
52 (gethostbyname)	x	implicit	x	0x80000034	0x80000034
57 (gethostname)	x	implicit	x	0x80000039	0x80000039
23 (socket)	x	implicit	x	0x80000017	0x80000017
GetFileVersionInfoW	-	implicit	-	0x000B7A06	0x000B7A06
VerQueryValueW	-	implicit	-	0x000B7A1C	0x000B7A1C
GetFileVersionInfoSizeW	-	implicit	-	0x000B79EC	0x000B79EC
timeGetTime	x	implicit	-	0x000B7A3A	0x000B7A3A
waveOutSetVolume	x	implicit	-	0x000B7A5A	0x000B7A5A
mciSendStringW	x	implicit	-	0x000B7A48	0x000B7A48
ImageList_Destroy	-	implicit	-	0x000B7ABC	0x000B7ABC
ImageList_Remove	-	implicit	-	0x000B7AD0	0x000B7AD0
ImageList_SetDragCursorImage	-	implicit	-	0x000B7AE4	0x000B7AE4

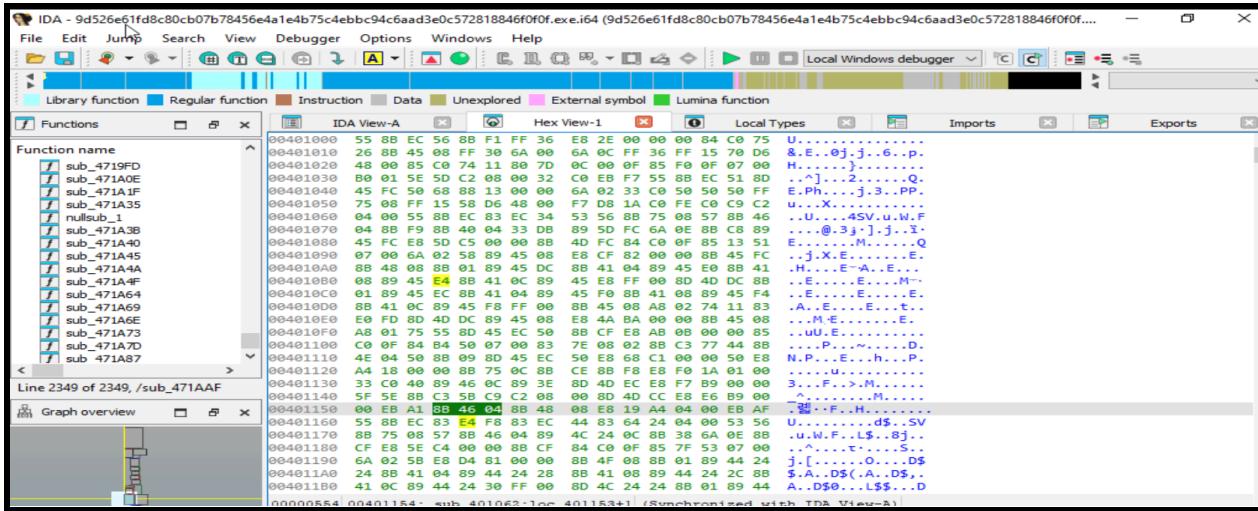
8.2 DIE Analysis

The repeated "Heur/Packer" detection, highlighted in red, indicates the file's code is compressed or obfuscated in a manner commonly used by malware to evade detection. This raises a significant red flag, suggesting it should be treated as a potential threat.



8.3 IDA Pro

1. Hex patterns such as E8 D4 81 00 00 and 0F 84 B4 50 07 00 correspond to valid x86 instructions like **CALL** and conditional JE (jump if equal).
2. The CALL instruction with large offsets suggests redirection to other code regions, possibly unpacked or dynamically loaded functions.
3. The **JE** (conditional jump) and **PUSH** instructions are commonly used in control flow structures and function setups in both legitimate and malicious binaries.
4. While the individual instructions are not inherently malicious, their use in specific patterns may indicate behaviour associated with obfuscation or staged execution.
5. These patterns, in combination with other static indicators such as high entropy and missing import names, support the hypothesis of potentially malicious behaviour.



8.4 String Analysis

Initial static string analysis on the executable file named 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe, it was observed that the file contains standard Portable Executable (PE) section headers such as .text, .data, .rdata, and .reloc, indicating it is a valid Windows binary. However, the presence of multiple non-human-readable, obfuscated strings (e.g., D\$4P, j\$Yf9, QQSW|, Ht#H) and the absence of meaningful strings like API names or file paths suggest that the file is packed or encrypted, likely to hide its real behaviour. These patterns are consistent with malicious executables employing anti-analysis or obfuscation techniques. Based on this analysis, the file is highly suspicious and potentially malware, and further dynamic analysis or unpacking is recommended to fully understand its functionality.

- I used this command to extract the strings “strings.exe”
“C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe” > “C:\Users\Spam\Desktop\extracted_strings.txt””



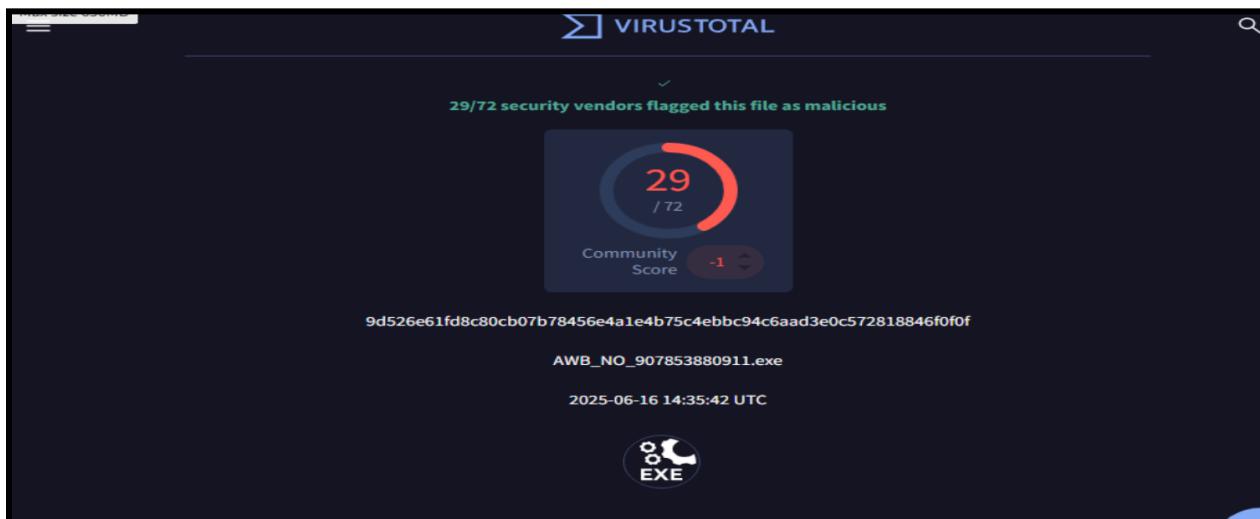
A screenshot of a Windows Notepad window titled "strings.txt - Notepad". The window contains the following text:

```
File Edit Normal View Help
!This program cannot be run in DOS mode.
Rich
.text
.rdata
.data
.rsrc
@.reloc
L$8F
D$4P
D$ P
D$OP
D$SP
D$4P
D$OP
D$H(
D$,$
D$p,
D$t,
D$H
j*x*f9
t$5j)
L$ Q
QQVW
+t\HhtT
D$OP
D$OL|
D$@;s
D$4PV
j*y*f9
j*y*f9
j\y*f9
;D@
D$LHP
D$ P
j+y^f;
Y1!Z
```

8.5 Virustotal Analysis

This Analysis of the executable

9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe revealed a high detection rate, with 29 out of 72 security vendors flagging it as malicious. The file was primarily categorised as a Trojan, with specific family labels including 'autoit'. This overwhelming consensus from multiple security vendors strongly indicates the file's malicious nature.



The screenshot shows the VirusTotal analysis interface. At the top, there are filters for 'Popular threat label' (set to 'trojan.autoit/autinj'), 'Threat categories' (set to 'trojan'), and 'Family labels' (set to 'autoit', 'autinj', and 'fil'). Below this, the 'Security vendors' analysis' section lists various antivirus engines and their findings. To the right, a sidebar asks 'Do you want to automate checks?' and features a blue speech bubble icon.

Security vendor	Findings
AhnLab-V3	Trojan/AU3.Loader.S2970
Arctic Wolf	Unsafe
Avast	FileRepMalware [Misc]
AVG	FileRepMalware [Misc]
Avira (no cloud)	HEUR/AGEN.1379770
Bkav Pro	W32.AIDetectMalware
CrowdStrike Falcon	Win/malicious_confidence_90% (W)
Cynet	Malicious (score: 99)
Elastic	Malicious (high Confidence)
ESET-NOD32	A Variant Of Generik.DZWJHVN
Google	Detected
Huorong	HEUR:TrojanSpy/AutoIT.Stealer.a
Ikarus	Trojan.Autoit
Kaspersky	Trojan.Win32.Strab.xai
Lionic	Trojan.Win32.Formbook.4lc
Malwarebytes	Backdoor.NetWiredRC.Autoit.Generic
MaxSecure	Trojan.Malware.300983.susgen

Sandbox results from CAPE, VMRay, and Yomi Hunter indicated suspicious behaviours, alongside observations of significant API calls, process creations, file modifications, and network connections. For instance, VMRay alone logged 48 API calls and 8 network connections, providing concrete evidence of the file's runtime activities and interactions with the operating system and external networks.

The screenshot shows the 'Display grouped sandbox reports' section of the VirusTotal interface. It lists seven sandboxes with their respective activity counts:

Sandbox	API	Process	File	Network	Event
CAPA	0	9	0	0	0
CAPE Sandbox	1	3	0	3	3
Microsoft Sysinternals	0	0	0	0	14
VMRay	1	5	0	0	48
VirusTotal Jujubox	0	0	0	0	2
Yomi Hunter	1	4	1	0	0
Zenbox	0	5	0	3	2

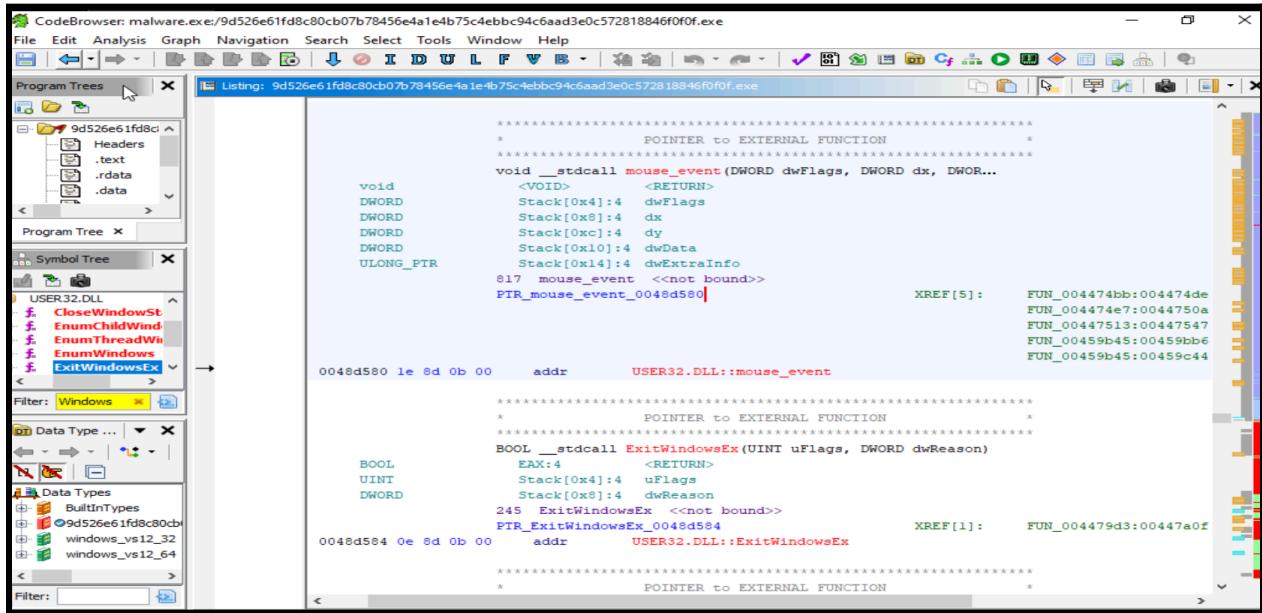
At the bottom, there is a summary section titled 'Activity Summary'.

9. Reverse Engineering

To start Ghidra, run the ghidraRun script and create a non-shared project. Import your binary file via *File → Import*, and let Ghidra detect the format. After import, select the file to open it in the CodeBrowser tool. When prompted, start analysis with default options to begin exploring the executable.

Suspicious API Calls Identified

- `mouse_event` (USER32.DLL): Indicates the program may simulate mouse actions. Common in cases of fake user activity, screen hijacking, or anti-idle tactics.
- `ExitWindowsEx` (USER32.DLL): Enables the program to shut down or log off the system, potentially as part of a self-destruct or sabotage routine.



The screenshot shows the Ghidra CodeBrowser interface with the following details:

- Title Bar:** CodeBrowser: malware.exe/9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
- Menu Bar:** File Edit Analysis Graph Navigation Search Select Tools Window Help
- Toolbars:** Standard toolbar with icons for file operations, search, and analysis.
- Left Sidebar:**
 - Program Trees:** Shows the file structure with sections like Headers, .text, .rdata, and .data.
 - Symbol Tree:** Shows symbols from USER32.DLL, including `CloseWindow`, `EnumChildWind`, `EnumThreadWind`, `EnumWindows`, and `ExitWindowsEx`.
 - Data Type ...**: Shows data types like `BOOL`, `UINT`, and `DWORD`.
- Middle Panel:** **Listing:** 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
- Assembly View:** Displays assembly code for imports from USER32.DLL:

```
void __stdcall mouse_event(DWORD dwFlags, DWORD dx, DWORD dy, DWORD dwData, ULONG_PTR dwExtraInfo)
    Stack[0x4]:4 dwFlags
    Stack[0x8]:4 dx
    Stack[0xc]:4 dy
    Stack[0x10]:4 dwData
    Stack[0x14]:4 dwExtraInfo
S17 mouse_event <>not bound>>
PTR_mouse_event_0048d580
```

0048d580 1e 8d 0b 00 addr USER32.DLL::mouse_event


```
BOOL __stdcall ExitWindowsEx(UINT uFlags, DWORD dwReason)
    EAX:4 <RETURN>
    Stack[0x4]:4 uFlags
    Stack[0x8]:4 dwReason
245 ExitWindowsEx <>not bound>>
PTR_ExitWindowsEx_0048d584
```

0048d584 0e 8d 0b 00 addr USER32.DLL::ExitWindowsEx
- Right Sidebar:** Shows XREF counts and function addresses for each import.

I conducted a static analysis of the sample 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe using Ghidra. The binary revealed several suspicious imports from USER32.DLL, including `mouse_event`, `ExitWindowsEx`, `SetCapture`, and `MonitorFromRect`. These functions suggest the malware may attempt to simulate user input, manipulate window behaviour, or force a system shutdown. Such capabilities point to possible user deception techniques or evasion strategies.

Listing: 9d526e61fd8c80cb07b78456e4a1e1b75c1ebbc94c6aad3e0c572818846f0f0f.exe

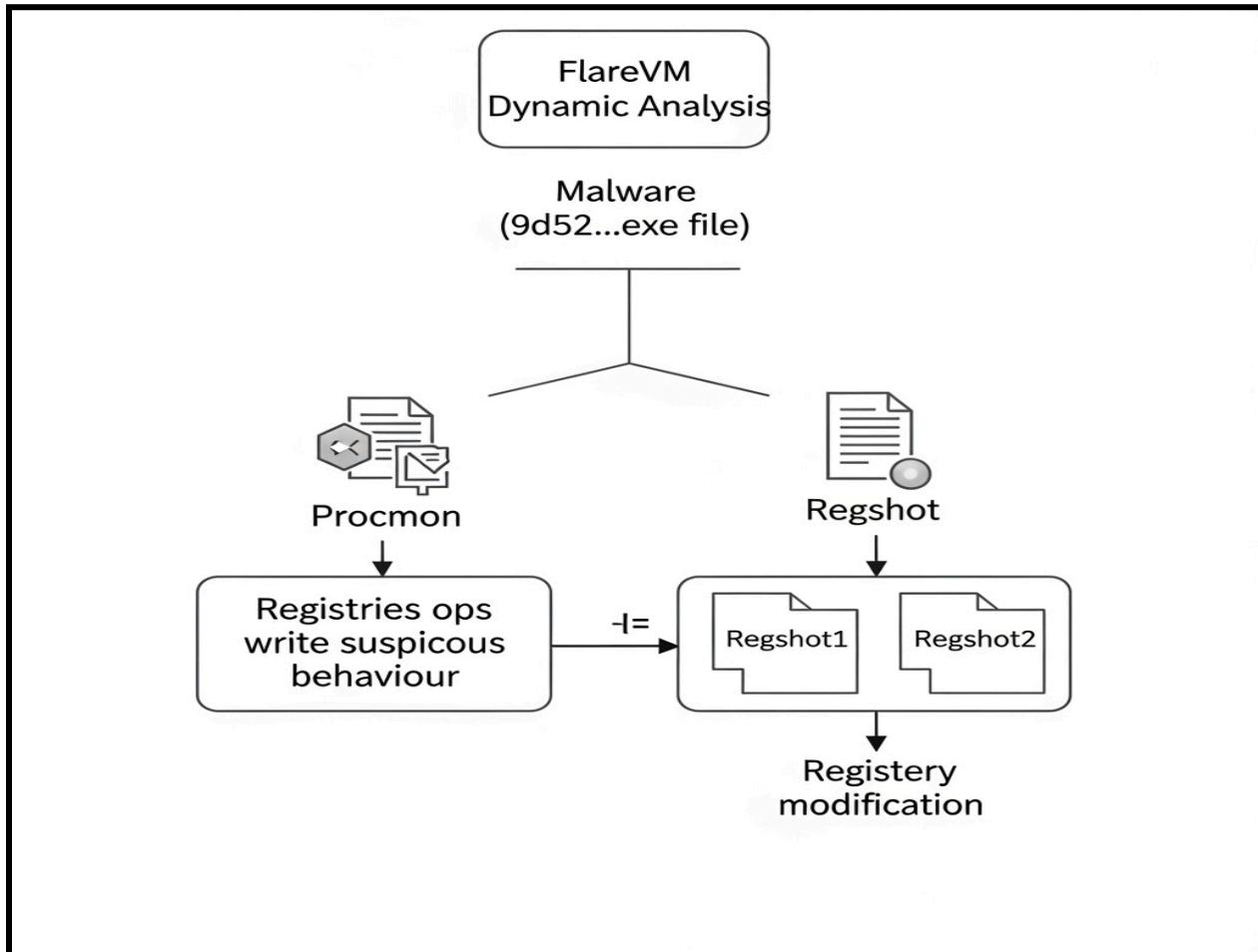
0048d56c c0 91 0b 00 addr USER32.DLL::ReleaseCapture

0048d570 d2 91 0b 00 addr USER32.DLL::SetCapture

0048d574 56 8d 0b 00 addr USER32.DLL::MonitorFromRect

10. Dynamic Analysis

Dynamic analysis involves executing the malware in a controlled environment to observe its real-time behaviour. Tools like Process Monitor and Regshot help identify system changes, file activity, and registry modifications made during execution.



10.1 Process Monitor(Procon)

During dynamic analysis using Process Monitor,

9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe exhibited several behaviours indicative of potentially malicious activity. It was observed modifying important system settings within the registry, specifically in areas related to how services run. Additionally, the process repeatedly performs **create and close file** operations within the temporary folder. Such actions are not typical of legitimate applications and strongly suggest that 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe is engaging in unauthorised and potentially harmful operations.

Steps to run Procon:

1. Run Procon as the administrator.
 2. Set a filter: Go to **Filter** → **Filter...**
Set: Process Name → is →
9d526e61fd8c80cb07b78456e4a1e4b7
malware.exe file)
 3. Click Add and Ok
 4. We need to capture the procmon events
 5. Look for suspicious file writes, registry e

9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe (Your malware.exe file)

3. Click Add and Ok
 4. We need to capture the procmon events.
 5. Look for suspicious file writes, registry edits, process creation, or network activity.

Time ...	Process Name	PID	Operation	Path	Result	Detail
11:18:00	9d526e61fd8c8...	10168	Process Start		SUCCESS	Parent PID: 8132, ...
11:18:00	9d526e61fd8c8...	10168	Thread Create		SUCCESS	Thread ID: 2124
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Users\Spam\Downloads\9d526e61f...	SUCCESS	Image Base: 0x320...
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Windows\System32\win32.dll	SUCCESS	Image Base: 0x1ffe...
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Windows\System32\Av.dll	SUCCESS	Image Base: 0x77c...
11:18:00	9d526e61fd8c8...	10168	Create File	C:\Windows\Prefetch\9D526E61FD8C...	NAME NOT FOUND	Desired Access: G...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Length: 80
11:18:00	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Length: 24
11:18:00	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	
11:18:00	9d526e61fd8c8...	10168	CreateFile	C:\Windows	SUCCESS	Desired Access: E...
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x1ffe...
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Windows\System32\wow64.dll	SUCCESS	Image Base: 0x1ffe...
11:18:00	9d526e61fd8c8...	10168	Create File	C:\Windows\System32\wow64log.dll	NAME NOT FOUND	Desired Access: R...
11:18:00	9d526e61fd8c8...	10168	CreateFile	C:\Windows	SUCCESS	Desired Access: R...
11:18:00	9d526e61fd8c8...	10168	CloseFile	C:\Windows	SUCCESS	Name: \Windows
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\Software\Microsoft\Wow64\8...	SUCCESS	Desired Access: R...
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\Software\Microsoft\Wow64\8...	NAME NOT FOUND	Length: 520
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\Software\Microsoft\Wow64\8...	SUCCESS	Type: REG_SZ, Le...
11:18:00	9d526e61fd8c8...	10168	RegCloseKey	HKEY\Software\Microsoft\Wow64\8...	SUCCESS	
11:18:00	9d526e61fd8c8...	10168	Load Image	C:\Windows\System32\wow4cpu.dll	SUCCESS	Image Base: 0x77c...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegSetInfoKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	KeySetInformation...
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Length: 80
11:18:00	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	Desired Access: Q...
11:18:00	9d526e61fd8c8...	10168	RegSetInfoKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	KeySetInformation...
11:18:00	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND	Length: 24
11:18:00	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS	

Regarding the Procmon logs, the executable

9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe extensively interacts with the Windows Registry, performing numerous ReqOpenKey operations.

often probing for non-existent paths. The process also frequently uses RegQueryValue to read registry data and employs RegSetValue to modify or create registry entries. Furthermore, the presence of RegSetInfoKey alongside these other registry activities indicates attempts by the process to not just query and modify data, but also potentially alter the properties or state of registry keys, a behaviour commonly observed in malicious software.

Process Monitor - Sysinternals: www.sysinternals.com

File Edit Event Filter Tools Options Help

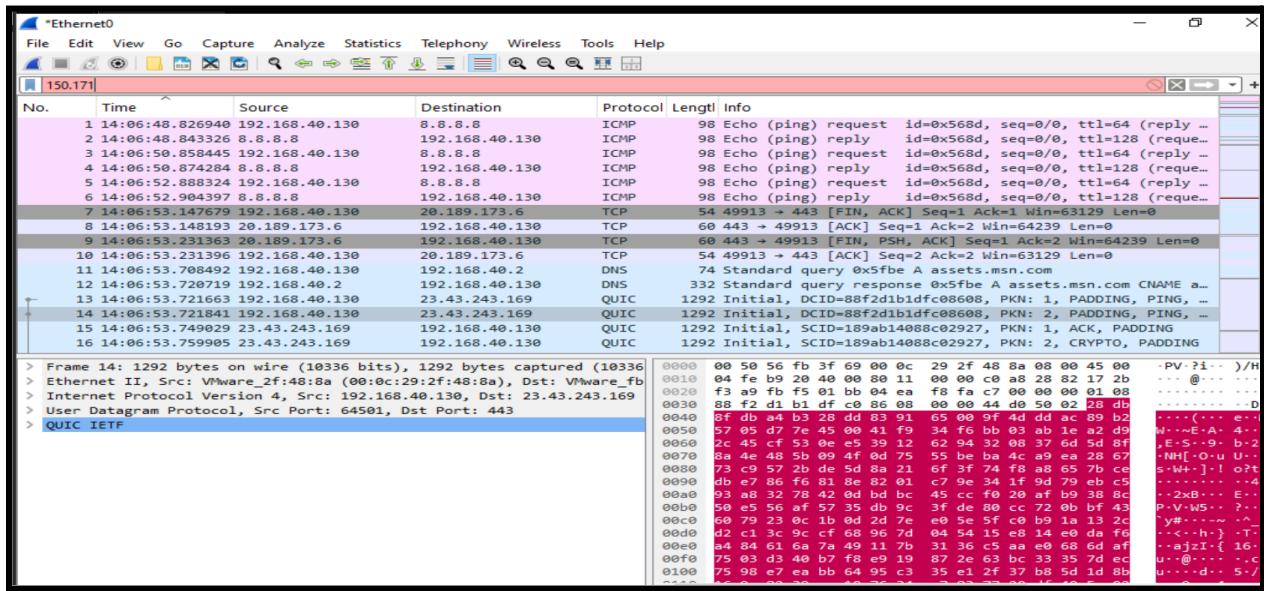
Time ... Process Name PID Operation Path Result Detail

11:18...	9d526e61fd8c8...	10168	RegOpenKey	HKEY\Software\Policies\Microsoft\Win...	NAME NOT FOUND Desired Access: Q...
11:18...	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE Desired Access: R...
11:18...	9d526e61fd8c8...	10168	RegQueryKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS Desired Access: R...
11:18...	9d526e61fd8c8...	10168	RegGetInfoKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS KeySetInformation...
11:18...	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS Type: REG_DWORD...
11:18...	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS
11:18...	9d526e61fd8c8...	10168	Thread Create		SUCCESS Thread ID: 11140
11:18...	9d526e61fd8c8...	10168	RegOpenKey	HKEY\Software\WOW6432Node\Micr...	NAME NOT FOUND Desired Access: E...
11:18...	9d526e61fd8c8...	10168	CreateFile	C:\Windows\WinSxS\x86_microsoft.win...	SUCCESS Desired Access: E...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\psapi.dll	SUCCESS Image Base: 0x76f...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\user32.dll	SUCCESS Image Base: 0x76a...
11:18...	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE Desired Access: Q...
11:18...	9d526e61fd8c8...	10168	RegQueryKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS Desired Access: Q...
11:18...	9d526e61fd8c8...	10168	RegGetInfoKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS KeySetInformation...
11:18...	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND Length: 24
11:18...	9d526e61fd8c8...	10168	RegCloseKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\win32u.dll	SUCCESS Image Base: 0x761...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\gd32.dll	SUCCESS Image Base: 0x761...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\gd32full.dll	SUCCESS Image Base: 0x763...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\msvcpr_win.dll	SUCCESS Image Base: 0x762...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\ucrtbase.dll	SUCCESS Image Base: 0x75b...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\cmdlg32.dll	SUCCESS Image Base: 0x75e...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\ole32.dll	SUCCESS Image Base: 0x754...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\combase.dll	SUCCESS Image Base: 0x758...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\pcperf4.dll	SUCCESS Image Base: 0x779...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\SHCore.dll	SUCCESS Image Base: 0x772...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\ahlpapi.dll	SUCCESS Image Base: 0x77b...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\shell32.dll	SUCCESS Image Base: 0x764...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\advapi32.dll	SUCCESS Image Base: 0x77c...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\sechost...	SUCCESS Image Base: 0x76f...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\ole32.dll	SUCCESS Image Base: 0x75f...
11:18...	9d526e61fd8c8...	10168	Load Image	C:\Windows\SysWOW64\oleaut32.dll	SUCCESS Image Base: 0x75e...
11:18...	9d526e61fd8c8...	10168	RegOpenKey	HKEY\SYSTEM\CurrentControlSet\Cont...	REPARSE Desired Access: Q...
11:18...	9d526e61fd8c8...	10168	RegSetInfoKey	HKEY\SYSTEM\CurrentControlSet\Cont...	SUCCESS Desired Access: Q...
11:18...	9d526e61fd8c8...	10168	RegQueryValue	HKEY\SYSTEM\CurrentControlSet\Cont...	NAME NOT FOUND Length: 16
11:18...	9d526e61fd8c8...	10168	CreateFile	C:\Users\Spam\Downloads\WSOCK32...	NAME NOT FOUND Desired Access: R...
11:18...	9d526e61fd8c8...	10168	CreateFile	C:\Windows\SysWOW64\wsocck32.dll	SUCCESS Desired Access: R...
11:18...	9d526e61fd8c8...	10168	QueryBasicInfor...	C:\Windows\SysWOW64\wsocck32.dll	SUCCESS CreationTime: 12/7...
11:18...	9d526e61fd8c8...	10168	CloseFile	C:\Windows\SysWOW64\wsocck32.dll	SUCCESS
Showing 1,968 of 718,681 events (0.27%)	Backed by virtual memory				

Source IP	Destination IP	Protocol	Action	Details
192.168.1.10	192.168.1.11	HTTP	POST	Request: /index.html
192.168.1.11	192.168.1.10	HTTP	200 OK	Response: index.html

10.2 Wireshark

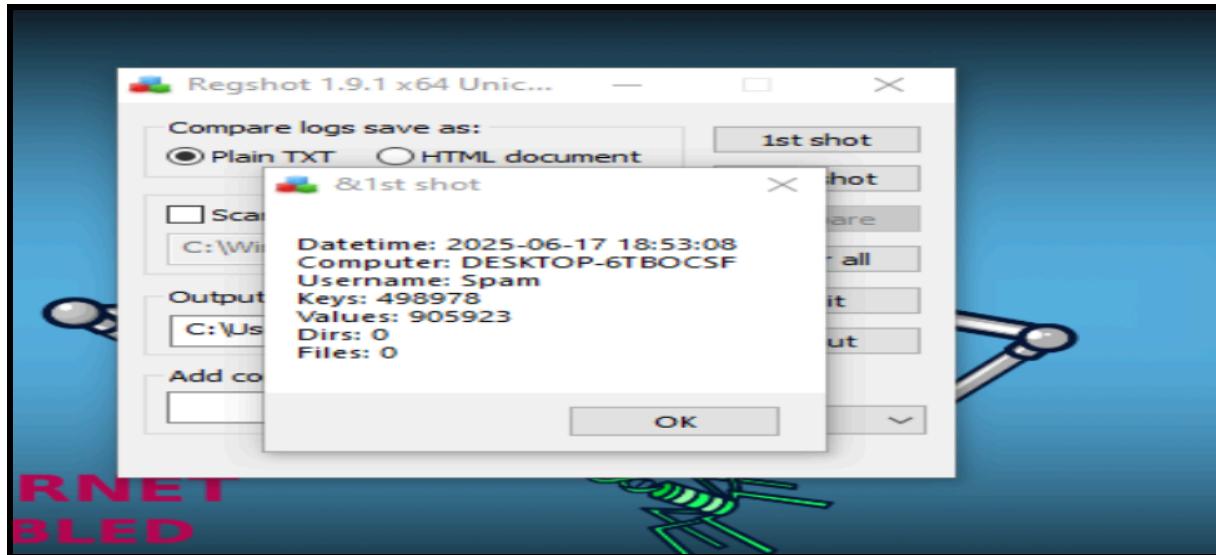
The malware from IP 192.168.40.130 was observed pinging Google's DNS (8.8.8.8) to check for internet access. It also initiated encrypted traffic using the QUIC protocol to 23.43.243.169, a method often used to avoid detection. Another encrypted connection occurred with 150.171.28.11, suggesting a stealthy attempt to reach an external server. Additionally, traffic to 20.189.173.6—a Microsoft-owned IP—could be used to blend malicious communication with normal system activity. These behaviours indicate that the malware is actively and quietly trying to communicate over the internet.



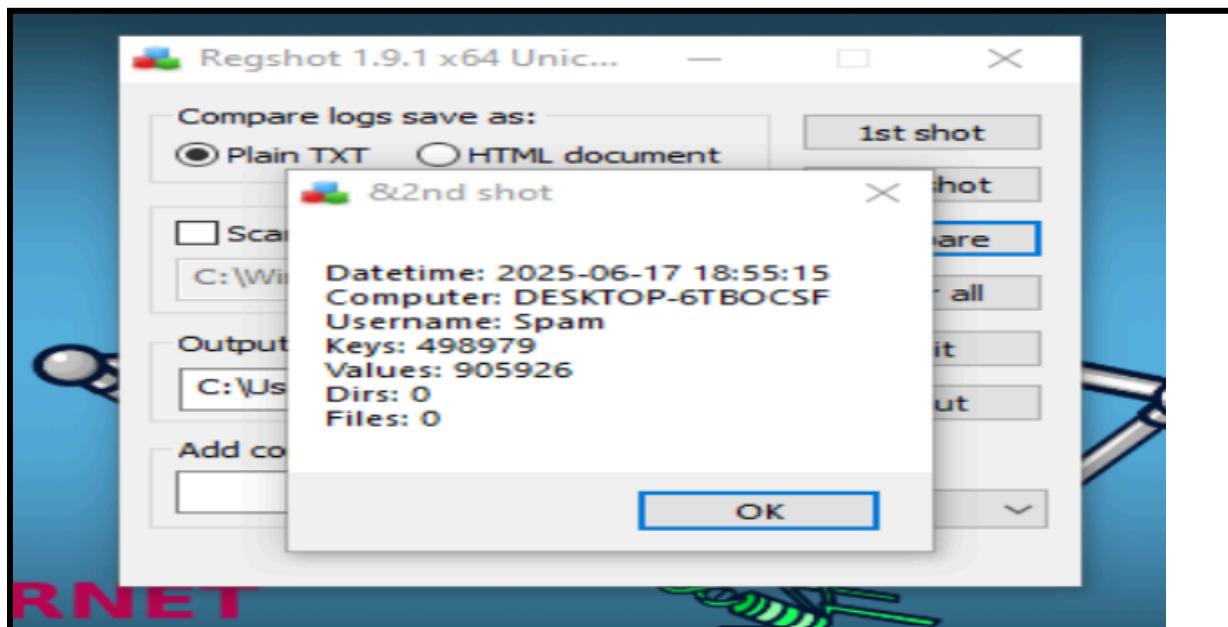
10.3 Regshot

To further investigate the registry modifications performed by `malware_sim.exe`, the Registry Snapshot Tool (Regshot) was utilised. Two snapshots of the system registry were taken.

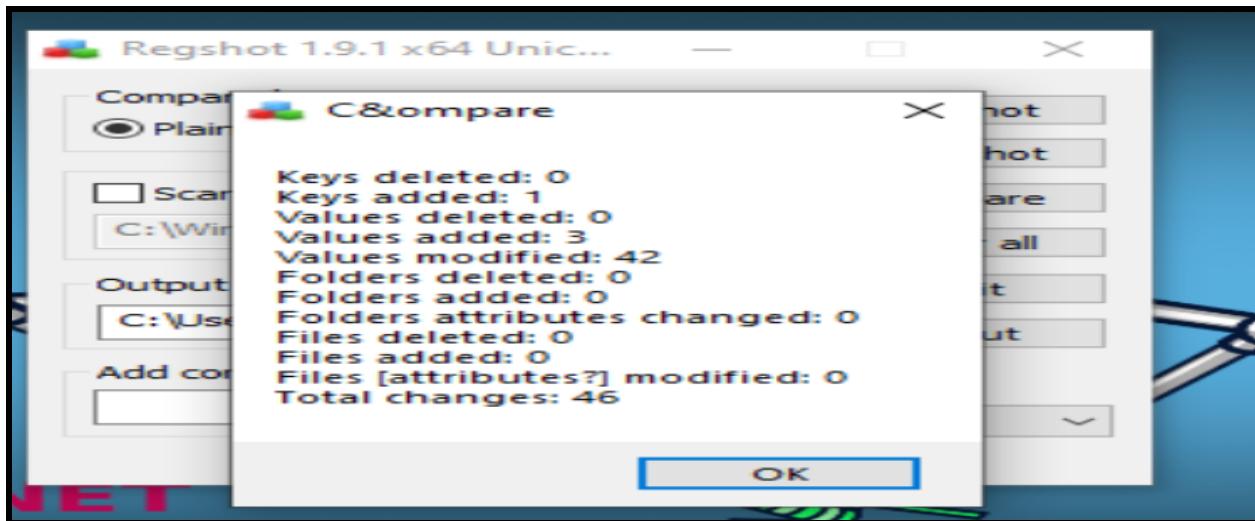
1st shot (Baseline): A baseline snapshot of the registry was captured before executing to record the initial state.



2nd shot(Post Execution): A second snapshot of the registry was taken after executing and 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe, allowing it to run for a period.



Output: The comparison between the '1st Shot' and the '2nd Shot' revealed the following modifications made to the registry by 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe.



```

~res-x64.txt - Notepad
File Edit Format View Help
Regshot 1.9.1 x64 Unicode (beta r321)
Comments:
Datetime: 2025-06-17 18:53:08, 2025-06-17 18:55:15
Computer: DESKTOP-6TBOCSF, DESKTOP-6TBOCSF
Username: Spam, Spam

-----
Keys added: 1
-----
HKLM\SOFTWARE\Microsoft\Windows\Error Reporting\TermReason\9060

-----
Values added: 3
-----
HKLM\SOFTWARE\Microsoft\Windows\Error Reporting\TermReason\9060\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Error Reporting\TermReason\9060\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows>Error Reporting\TermReason\9060\CreationTime: 0x01DBDFB9412FEEBA

-----
Values modified: 42
-----
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\HeartBeats\Default\LastHeartBeatTime: 0x01DBDFA608DB4ADC
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\HeartBeats\Default\LastHeartBeatTime: 0x01DBDFAA39BEA5C4
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\HeartBeats\Default\HeartBeatSequenceNumber: 0x00000063
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\HeartBeats\Default\HeartBeatSequenceNumber: 0x00000064
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\HeartBeats\Default\LastDownloadTime: 0x01DBDFB7202EBC52
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Diagnostics\DiagTrack\SettingsRequests\LastDownloadTime: 0x01DBDFB938A10146
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime: 0x01DBDFB90C2E52A5
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\VFUProvider\StartTime: 0x01DBDFB953819BD6
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\Scheduler\Checking to see if mostack override has c
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\WindowsUpdate\Orchestrator\Scheduler\Checking to see if mostack override has c
HKLM\SOFTWARE\Microsoft\Windows\Defender\Diagnostics\PlatformHealthData: 03 00 00 00 30 01 00 00 AF 6A E2 32 44 DF DB 01 B0 C
HKLM\SOFTWARE\Microsoft\Windows\Defender\Diagnostics\PlatformHealthData: 03 00 00 00 30 01 00 00 AF 6A E2 32 44 DF DB 01 61 C
HKLM\SOFTWARE\Microsoft\Windows\Defender\Features\UpdateControl>LastHeartbeatSystime: 43 81 9C C6 B0 DF DB 01
HKLM\SOFTWARE\Microsoft\Windows\Defender\Features\UpdateControl>LastHeartbeatSystime: 50 5A 61 28 B9 DF DB 01
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Notifications\\Notifications\418A073AA3BC1C75: 86 0C 00 00 00 00 00 04 00 04 00 01
<                                     >

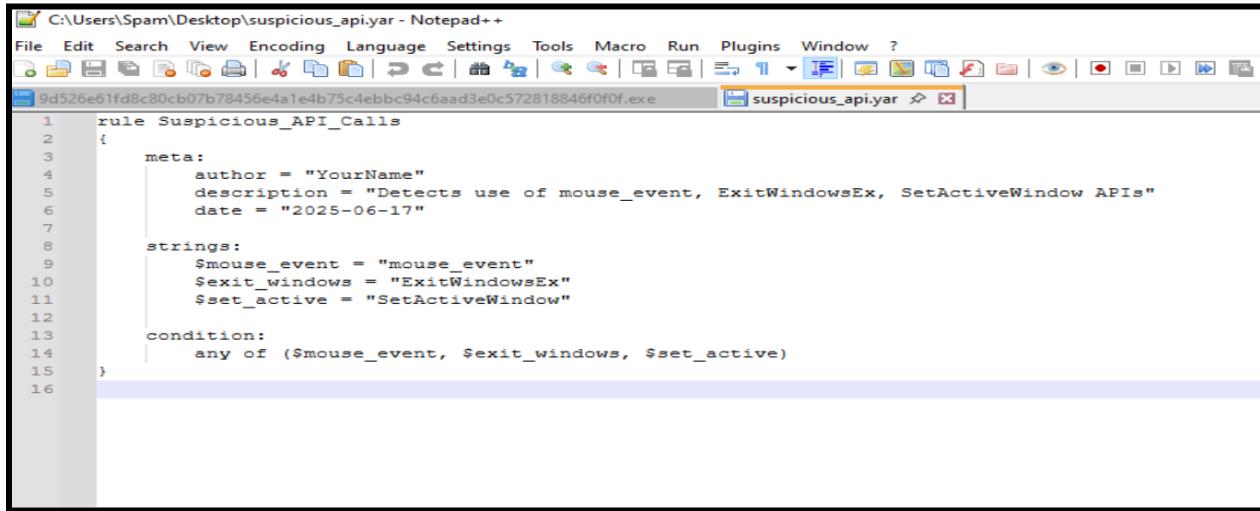
```

11. IOC Extractors

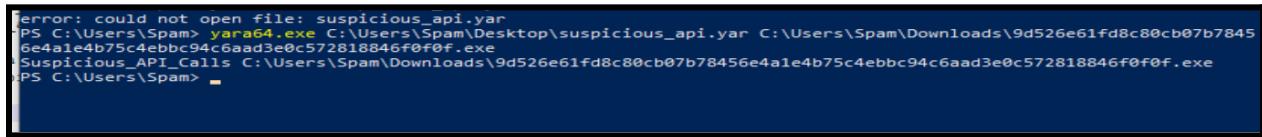
IOC Type	Indicator	Description
File Hash	9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f	SHA-256 hash of the analyzed malware sample
File Name	9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe	Sample filename used in testing
Registry Key	HKCU\Software\Microsoft\Windows\CurrentVersion\Run\MalwareLoader	Used for persistence across a reboot
Registry Access	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions\00060209	Queried during execution; key not found
API Call	ExitWindowsEx	Forces system shutdown—used for disruption
String Extraction Tool	strings.exe output	Used to identify embedded functions, paths, and suspicious strings
Monitoring Script	Register-WmiEvent script on Win32_Process	Behavioural script used to detect process creation.
API Call	SetActiveWindow, mouse_event	Used to simulate input and manipulate the GUI
Registry Access	HKLM\System\CurrentControlSet\Control\Nls\Sorting\Versions	Opened with REPARSE status

12. YARA Rules

An initial Yara rule was created to detect specific characteristics within 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe, such as defined strings. Scanning the malware with this rule.



```
C:\Users\Spam\Desktop\suspicious_api.yar - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe suspicious_api.yar ✎
1 rule Suspicious_API_Calls
2 {
3     meta:
4         author = "YourName"
5         description = "Detects use of mouse_event, ExitWindowsEx, SetActiveWindow APIs"
6         date = "2025-06-17"
7
8     strings:
9         $mouse_event = "mouse_event"
10        $exit_windows = "ExitWindowsEx"
11        $set_active = "SetActiveWindow"
12
13     condition:
14         any of ($mouse_event, $exit_windows, $set_active)
15
16 }
```



```
error: could not open file: suspicious_api.yar
PS C:\Users\Spam> yara64.exe C:\Users\Spam\Desktop\suspicious_api.yar C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
Suspicious_API_Calls C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
PS C:\Users\Spam>
```

Yara rules for Strings.

These strings represent Windows API calls used to simulate user activity (mouse_event), force system shutdown (ExitWindowsEx), and control window focus (SetActiveWindow). Malware often abuses them to evade detection, disrupt users, or automate interactions.

```
C:\Users\Spam\Desktop\Strings.yara - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
suspicious_api.yar Strings.yara ✎
1 rule MouseEvent_Space
2 {
3     meta:
4         author = "You"
5         description = "Detects use of 'mouse event' (with space)"
6         date = "2025-06-17"
7
8     strings:
9         $mouse = "mouse event" ascii wide nocase
10        $exit = "ExitWindowsEx" ascii wide nocase
11
12    condition:
13        any of them
14
15
16
17
```

```
PS C:\Users\Spam> yara64.exe C:\Users\Spam\Desktop\Strings.yara C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
MouseEvent_Space C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
PS C:\Users\Spam>
```

Suspicious User Behaviour

These represent Windows API calls used to simulate user activity (mouse_event), force system shutdown (ExitWindowsEx), and control window focus (SetActiveWindow). Malware often abuses them to evade detection, disrupt users, or automate interactions.

```
user.yara - Notepad
File Edit Format View Help
rule FakeUserActivity
{
    strings:
        $mouse = "mouse event" ascii wide nocase
        $setactive = "SetActiveWindow" ascii wide nocase
        $sendkeys = "SendInput" ascii wide nocase

    condition:
        any of them
}
```

```
MouseEvent_Space C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
PS C:\Users\Spam> yara64.exe C:\Users\Spam\Desktop\user.yara C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
FakeUserActivity C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
PS C:\Users\Spam>
```

Detect Shutdown

This binary uses the ExitWindowsEx API, which allows it to shut down or log off the system—a behaviour frequently abused by malware to interrupt user activity or trigger a system reboot after executing malicious tasks.

```
shutdown.yara - Notepad
File Edit Format View Help
import "pe"

rule SystemManipulationAPIs
{
    meta:
        author = "YourName"
        description = "Detects use of system manipulation functions like ExitWindowsEx"
        date = "2025-06-17"
    condition:
        pe.imports("USER32.dll", "ExitWindowsEx") or
        pe.imports("ADVAPI32.dll", "InitiateSystemShutdownExA")
}

PS C:\Users\Spam> yara64.exe C:\Users\Spam\Desktop\shutdown.yara C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
SystemManipulationAPIs C:\Users\Spam\Downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
PS C:\Users\Spam>
```

13. Mitigation

To define systems from malware leveraging APIs like mouse_event, ExitWindowsEX, and SetActionWindows, several mitigation strategies are required.

Application Whitelisting: It will restrict execution to trusted software only, preventing unauthorised or malicious programs from running.

Behavioural Monitoring via Process Creation Event Tracking

This mitigation technique uses behavioural monitoring via process creation event tracking to detect and respond to suspicious activity. By leveraging PowerShell's Register-WmiEvent, it watches for Win32_Process events and displays real-time alerts whenever a new process starts. This allows analysts to observe process names and IDs as they execute, providing immediate visibility into potential malicious behaviour. It's a lightweight and scriptable method to enhance situational awareness on the system.

Steps:

1. Open PowerShell as Administrator
2. Run this on the PowerShell

```
Register-WmiEvent -Query "SELECT * FROM __InstanceCreationEvent WITHIN 1 WHERE TargetInstance ISA 'Win32_Process'" -Action {  
    $process = $Event.SourceEventArgs.NewEvent.TargetInstance  
    Write-Host "Process started: $($process.Name) with PID $($process.ProcessId)"  
}  
Process started: msedge.exe with PID 6440  
Process started: msedge.exe with PID 7504
```

```
PS C:\Windows\system32> Register-WmiEvent -Query "SELECT * FROM __InstanceCreationEvent WITHIN 1 WHERE TargetInstance ISA 'Win32_Process'" -Action {  
    >> $process = $Event.SourceEventArgs.NewEvent.TargetInstance  
    >> Write-Host "Process started: $($process.Name) with PID $($process.ProcessId)"  
    >> # Add logic to check for suspicious process names or paths  
    >>}  
Id Name PSJobTypeName State HasMoreData Location Command  
-- -- -- -- -- -- --  
1 3bfbcac-660... NotStarted False ...  
  
PS C:\Windows\system32> Process started: msedge.exe with PID 6440  
Process started: msedge.exe with PID 7504  
PS C:\Windows\system32> Process started: notepad.exe with PID 1234
```

3. Keep PowerShell open for continuous monitoring.
4. Observe the real-time outputs of the PID.

```
PS C:\Windows\system32> Process started: SearchFilterHost.exe with PID 2472  
Process started: SearchProtocolHost.exe with PID 7156  
    Process started: msedge.exe with PID 3556  
    Process started: msedge.exe with PID 6688  
    Process started: svchost.exe with PID 6328  
    Process started: svchost.exe with PID 2600  
    Process started: dllhost.exe with PID 7608
```

Benefits:

1. Detect when suspicious or unexpected processes (like hidden payloads) are launched.
2. Monitor live behaviour without needing a full-blown sandbox or EDR system.
3. Create a lightweight alerting system for red-flag processes (e.g., PowerShell abuse, random EXEs, etc.).

Regular System and Antivirus Updates: Keeping the operating system, software, and antivirus definitions up to date is a foundational mitigation technique. Many malware variants exploit known vulnerabilities in outdated systems or bypass defences using signatures that only newer antivirus updates can detect.

- **Security patches** close known vulnerabilities
- **Antivirus engines** can detect newly discovered threats
- **Stability improvements** reduce system behaviours that malware can abuse.

API Hooking and Sandboxing: Advanced mitigation techniques such as API hooking and sandboxing are instrumental in detecting and analysing modern malware. API hooking allows security tools to intercept critical system function calls—such as ExitWindowsEx or CreateRemoteThread—in order to monitor or block suspicious behaviour in real time. This technique is commonly used by endpoint security platforms to prevent malware from executing malicious payloads.

14. Conclusion

This study presents a rigorous end-to-end analysis of the FormBook malware variant, performed within a securely configured virtual lab environment using a combination of static, dynamic, and reverse engineering techniques. The sample exhibited advanced obfuscation characteristics, high entropy levels, and suspicious API usage—traits confirmed through tools such as PEStudio, IDA Pro, and Ghidra. Dynamic instrumentation further revealed unauthorised registry modifications, file system alterations, and evidence of evasion tactics, such as simulated user activity and system shutdown triggers. The integration of UPX packing allowed for emulation of real-world obfuscation methods, emphasising the challenges faced by traditional signature-based detection mechanisms. Custom YARA rules were developed to detect unique behavioral and structural indicators. In response, multi-layered mitigation strategies—including process creation monitoring, application whitelisting, and regular system patching—were proposed to bolster system defences. This research reinforces the necessity of proactive threat analysis and layered security methodologies to counter evolving malware threats in modern computing environments.