Date: May 2025
Prepared By: Sahithi Gudigopuram

Table of Contents

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.

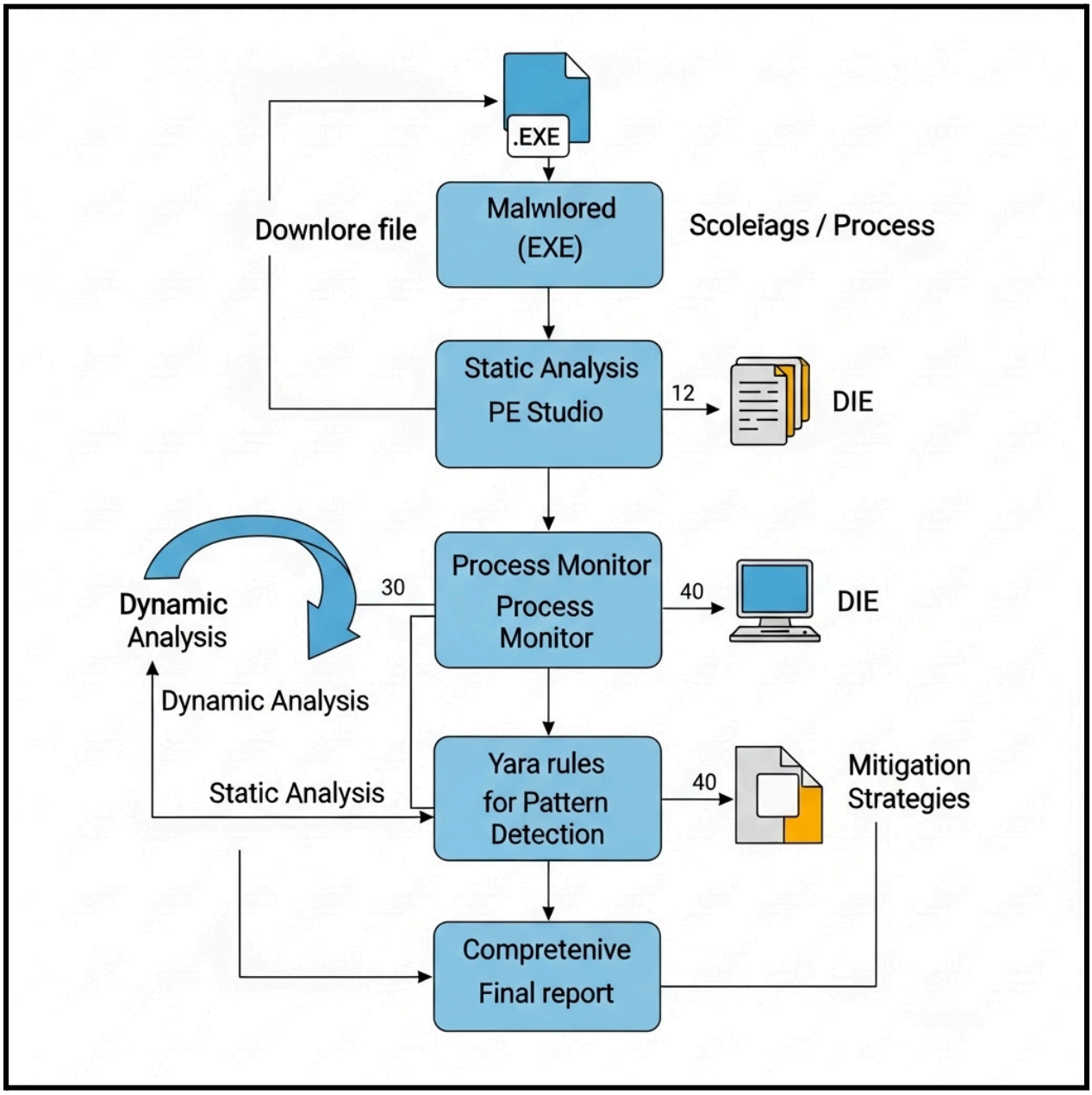
2. Introduction

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: af8c8f0a9e18595e7d217e5dadb07835
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

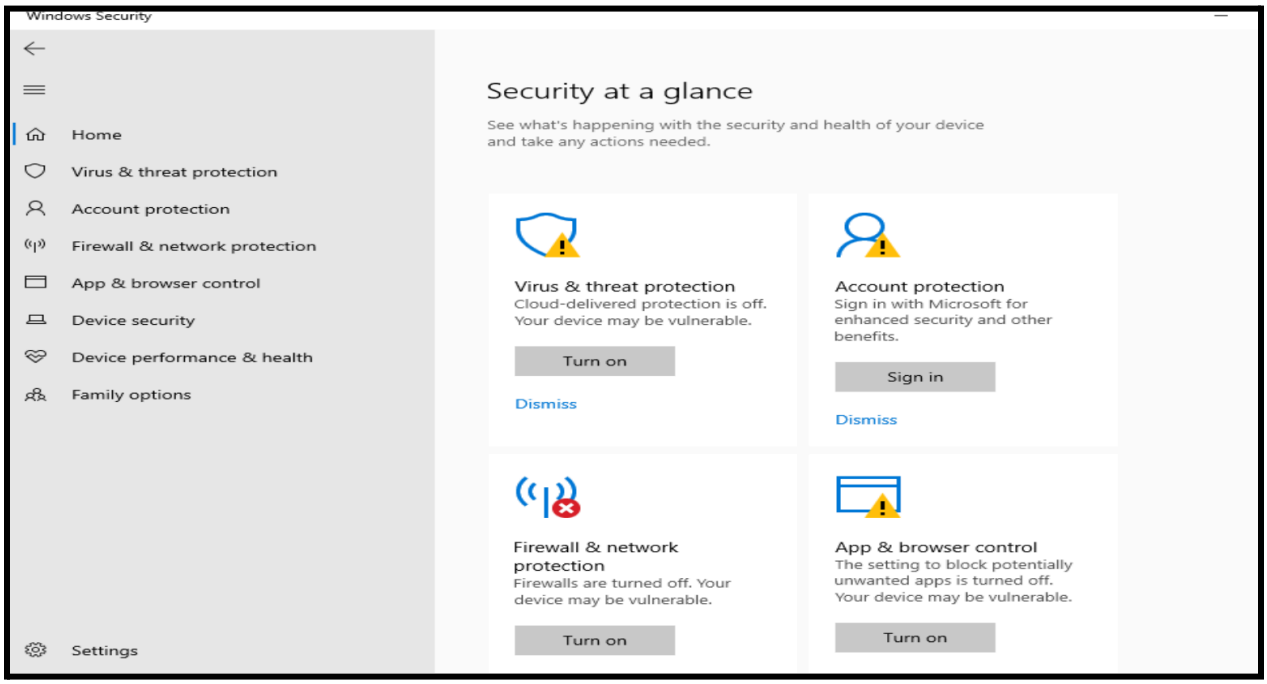
Component	Details
Virtualisation Tool	VMware
Guest OS	Windows 10(64-bit)
RAM	8GB
CPU Cores	2
Network Mode	Bridge Adapter(for setup), then switched to NAT
Snapshot	Created before execution for rollback safety

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

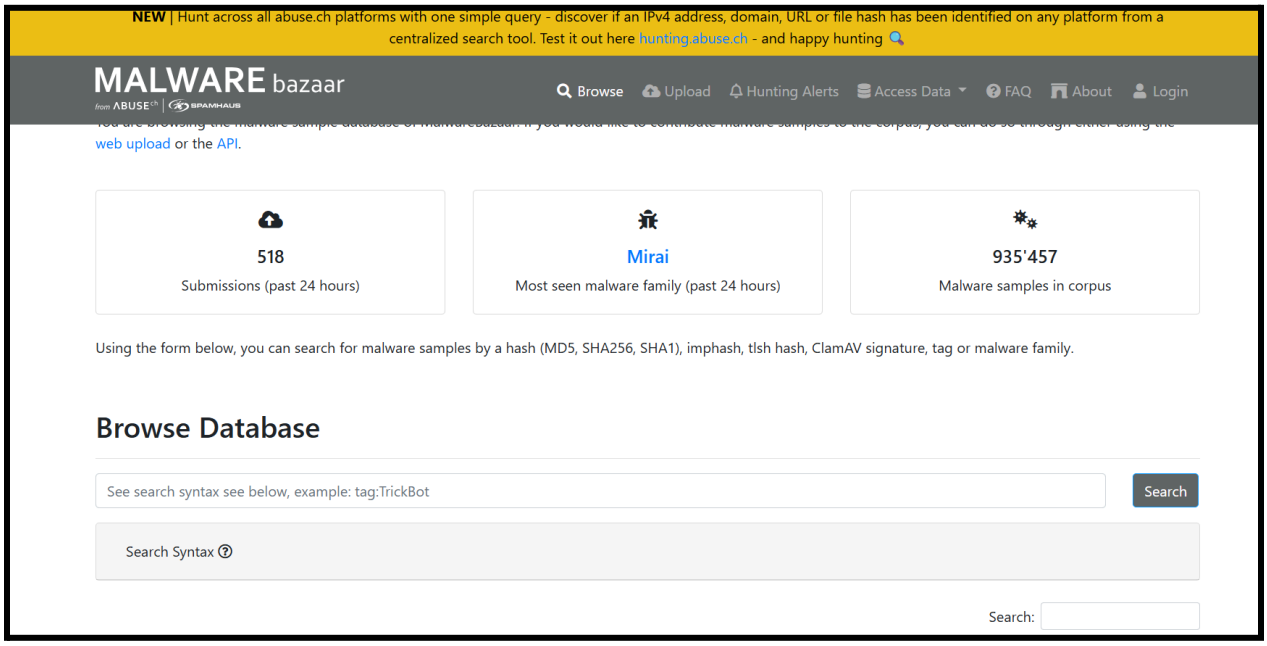
1. Download the FLARE VM installer from:”https://github.com/mandiant/flare-vm”
2. Run with Powershell “ **Set-ExecutionPolicy Unrestricted .\install.ps1 “**
3. During the installation, we will see a number of reboots in our system



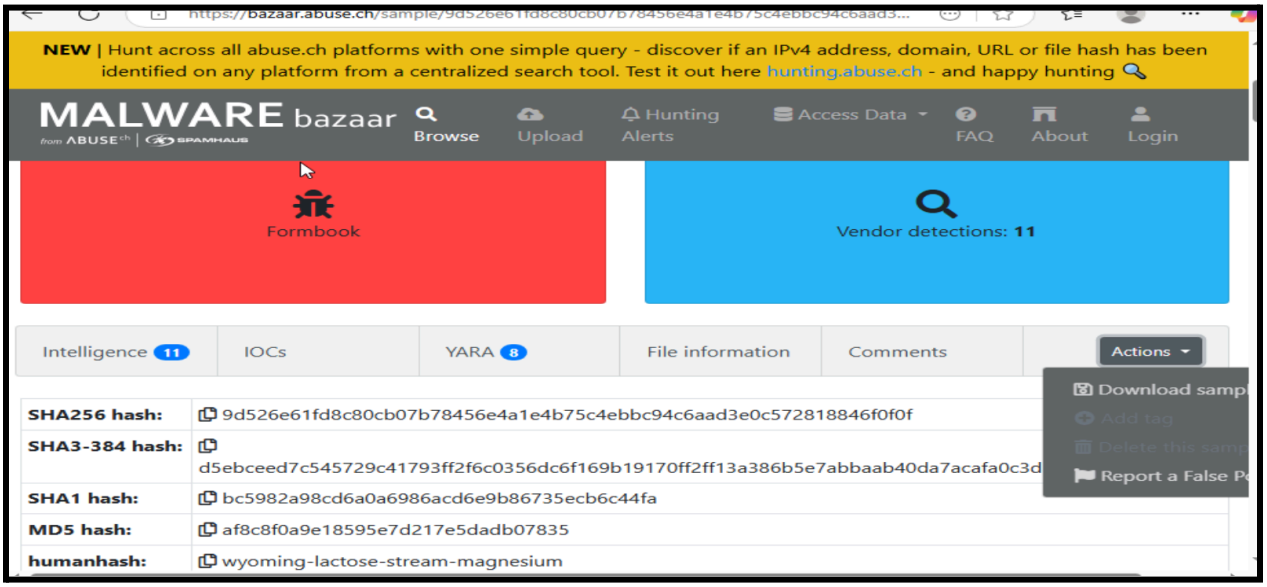
7. Malware Acquisition

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.

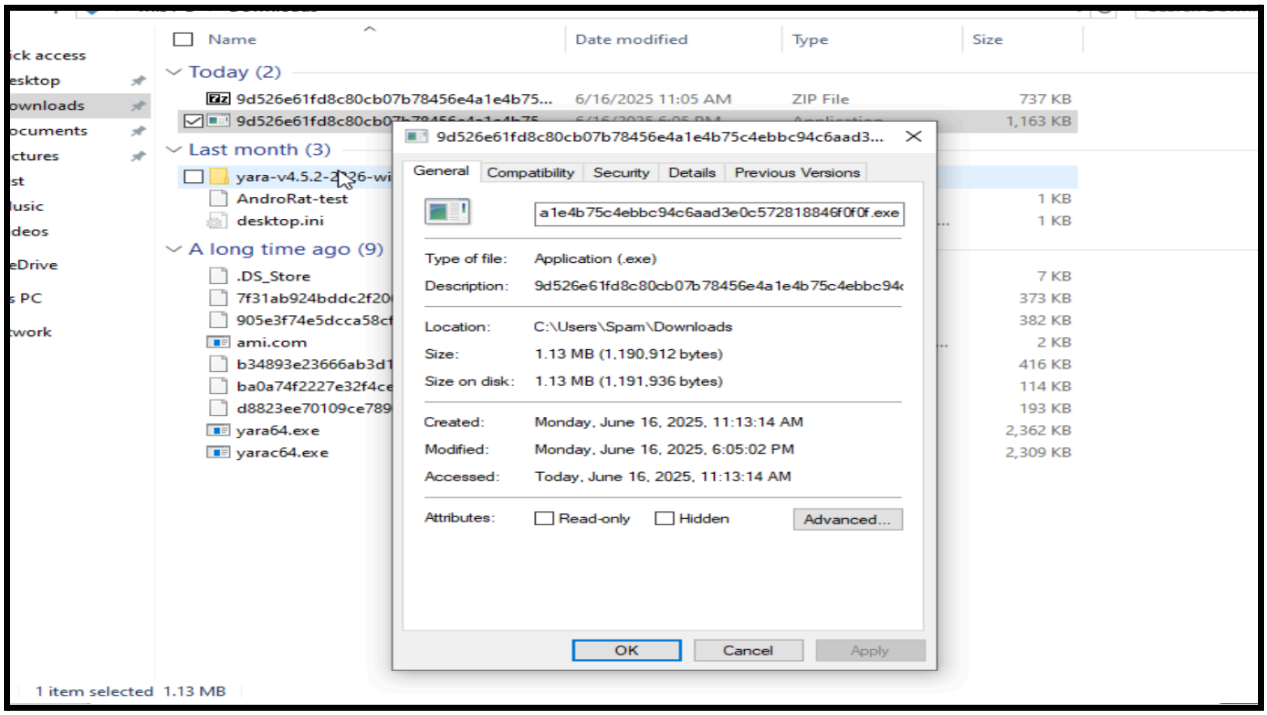
1. Go to Malwarebazaar to download the file "https://bazaar.abuse.ch/"



2. I downloaded a malware.

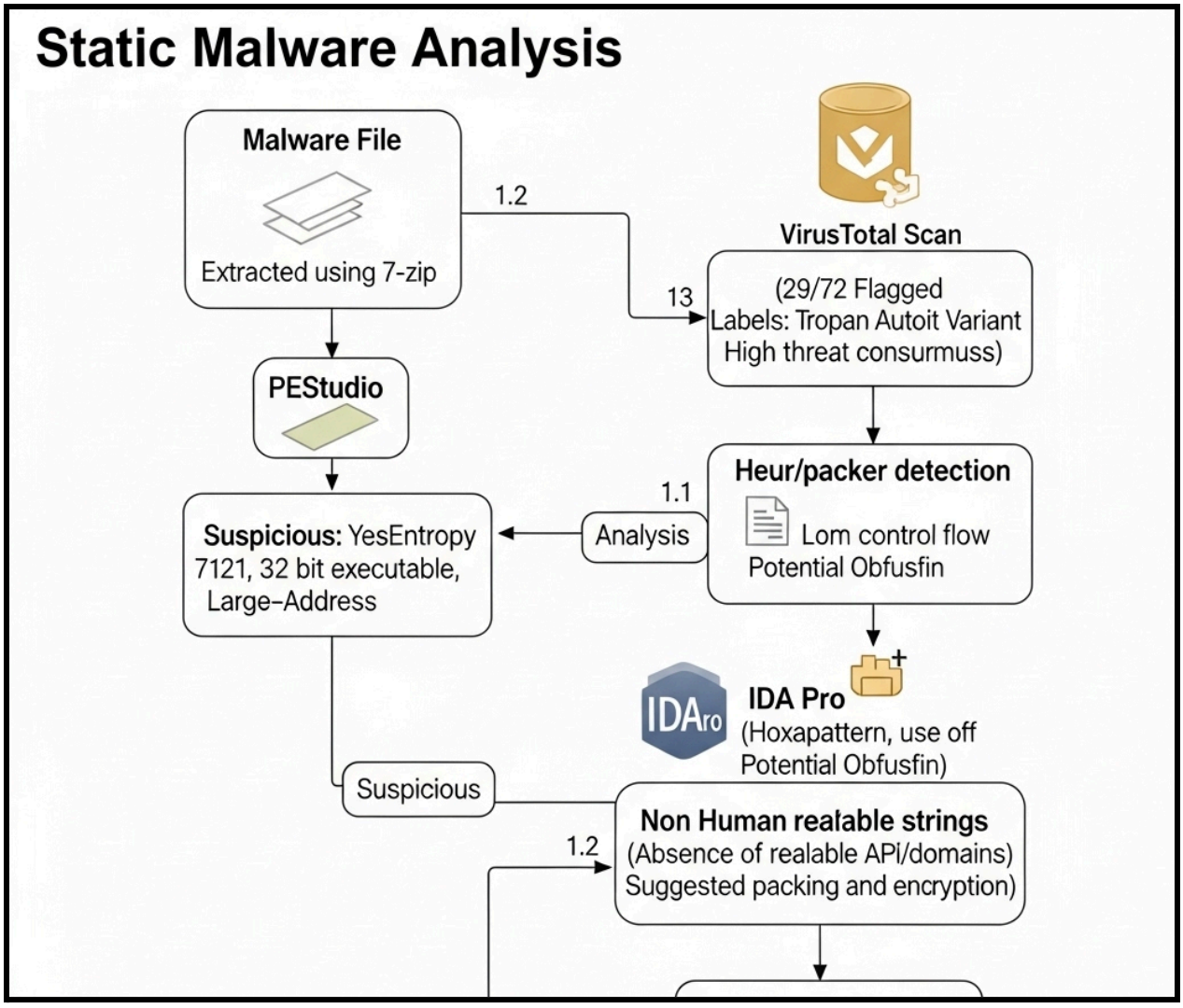


3. I used **7-Zip** to securely extract the password-protected malware archive without altering the original file.



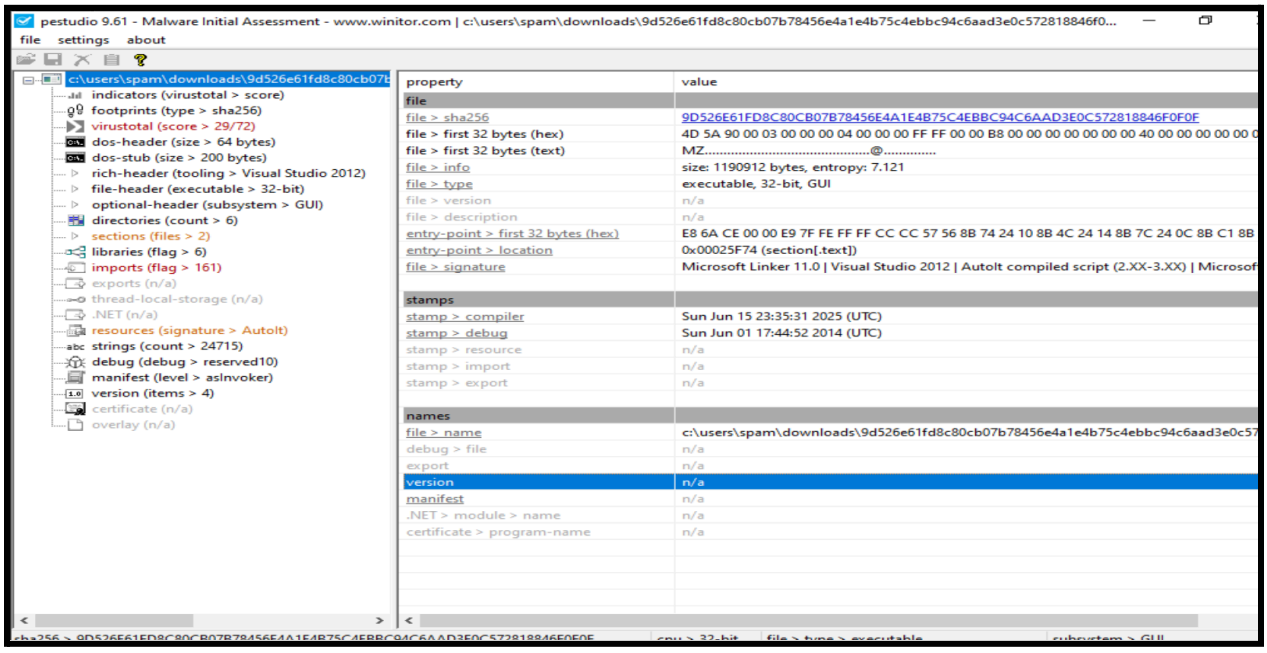
8. Static Analysis

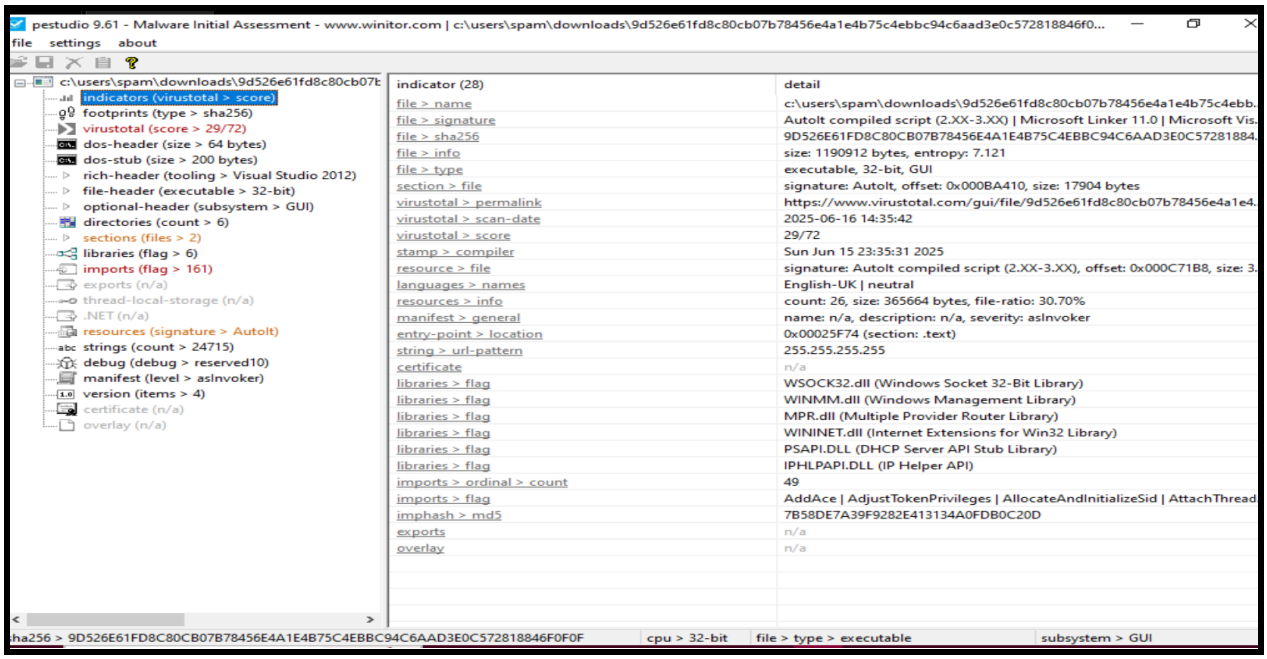
The malware was examined statically using tools like `PEStudio`, `strings.exe`, and `IDA Pro` to inspect headers, imports, and embedded resources.
Analysis revealed suspicious API calls, encoded strings, and entropy anomalies indicative of obfuscation or packing.

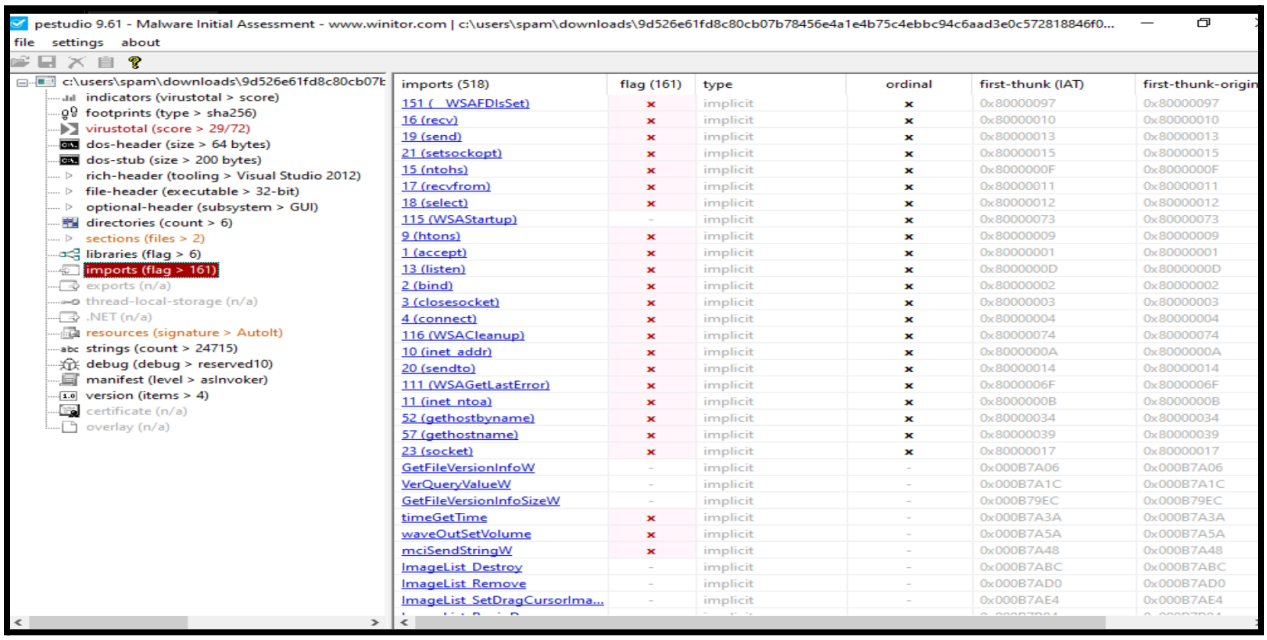


8.1 PEStudio

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.

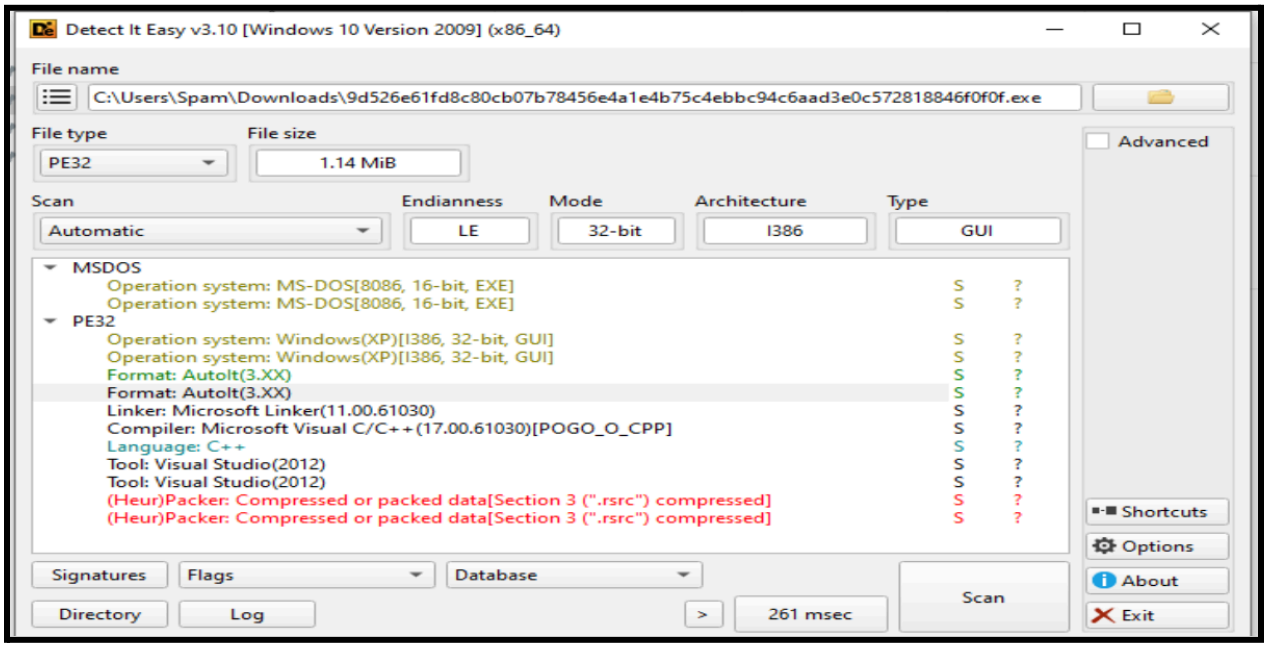



pestudio 9.61 - Malware Initial Assessment - www.winitor.com | c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0...
file settings about
c:\users\spam\downloads\9d526e61fd8c80cb07b
indicators (virustotal > score)
footprints (type > sha256)
virustotal (score > 29/72)
dos-header (size > 64 bytes)
dos-stub (size > 200 bytes)
rich-header (tooling > Visual Studio 2012)
file-header (executable > 32-bit)
optional-header (subsystem > GUI)
directories (count > 6)
sections (files > 2)
libraries (flag > 6)
imports (flag > 161)
exports (n/a)
thread-local-storage (n/a)
.NET (n/a)
resources (signature > AutoIt)
strings (count > 24715)
debug (debug > reserved10)
manifest (level > asInvoker)
version (items > 4)
certificate (n/a)
overlay (n/a)
indicator (28)
detail
file > name
c:\users\spam\downloads\9d526e61fd8c80cb07b78456e4a1e4b75c4ebb
file > signature
AutoIt compiled script (2.XX-3.XX) | Microsoft Linker 11.0 | Microsoft Vis
file > sha256
9D526E61FD8C80CB07B78456E4A1E4B75C4EBBC94C6AAD3E0C57281884
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 > flag
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
IPHLPAPI.DLL (IP Helper API)
imports > ordinal > count
49
imports > flag
AddAce | AdjustTokenPrivileges | AllocateAndInitializeSid | AttachThread
imphash > md5
7B58DE7A39F9282E413134A0FDB0C20D
exports
n/a
overlay
n/a
ha256 > 9D526E61FD8C80CB07B78456E4A1E4B75C4EBBC94C6AAD3E0C572818846F0F0F
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\9d526e61fd8c80cb07b
indicators (virustotal > score)
footprints (type > sha256)
virustotal (score > 29/72)
dos-header (size > 64 bytes)
dos-stub (size > 200 bytes)
rich-header (tooling > Visual Studio 2012)
file-header (executable > 32-bit)
optional-header (subsystem > GUI)
directories (count > 6)
sections (files > 2)
libraries (flag > 6)
imports (flag > 161)
exports (n/a)
thread-local-storage (n/a)
.NET (n/a)
resources (signature > AutoIt)
strings (count > 24715)
debug (debug > reserved10)
manifest (level > asInvoker)
version (items > 4)
certificate (n/a)
overlay (n/a)
imports (518)
flag (161)
type
ordinal
first-thunk (IAT)
first-thunk-origin
151 (__WSAFDIsSet)
16 (recv)
19 (send)
21 (setsockopt)
15 (ntohs)
17 (recvfrom)
18 (select)
115 (WSAStartup)
9 (htons)
1 (accept)
13 (listen)
2 (bind)
3 (closesocket)
4 (connect)
116 (WSACleanup)
10 (inet_addr)
20 (sendto)
111 (WSAGetLastError)
11 (inet_ntoa)
52 (gethostbyname)
57 (gethostname)
23 (socket)
GetFileVersionInfoW
VerQueryValueW
GetFileVersionInfoSizeW
timeGetTime
waveOutSetVolume
mciSendStringW
ImageList_Destroy
ImageList_Remove
ImageList_SetDragCursorIma...

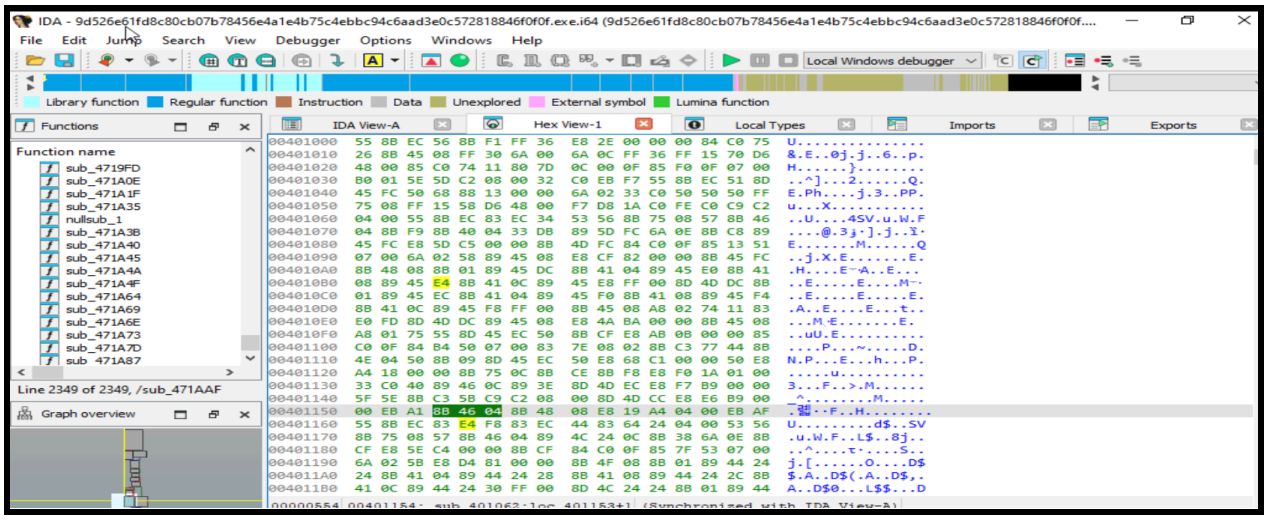
8.2 DIE Analysis

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"”

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.

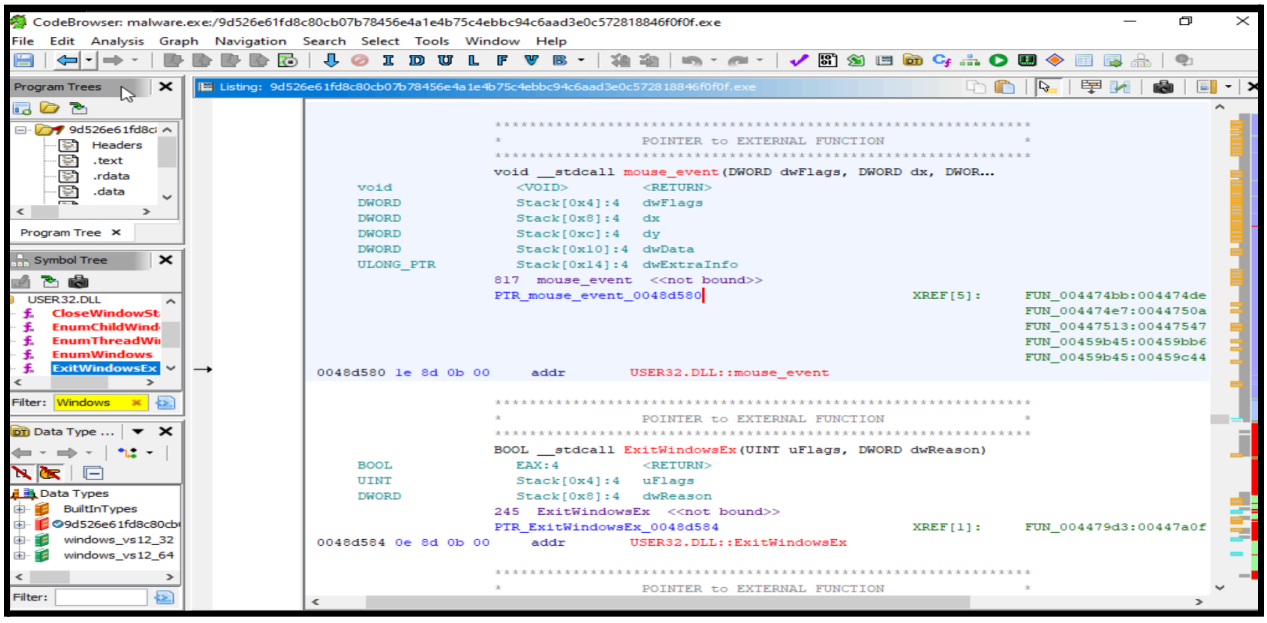
Sandbox results from CAPE, VMRay, and Yomi Hunter indicated suspicious behaviours, alongside observations of significant API calls, process creations, file modifications, and network connections. For instance, VMRay alone logged 48 API calls and 8 network connections, providing concrete evidence of the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

9. Reverse Engineering

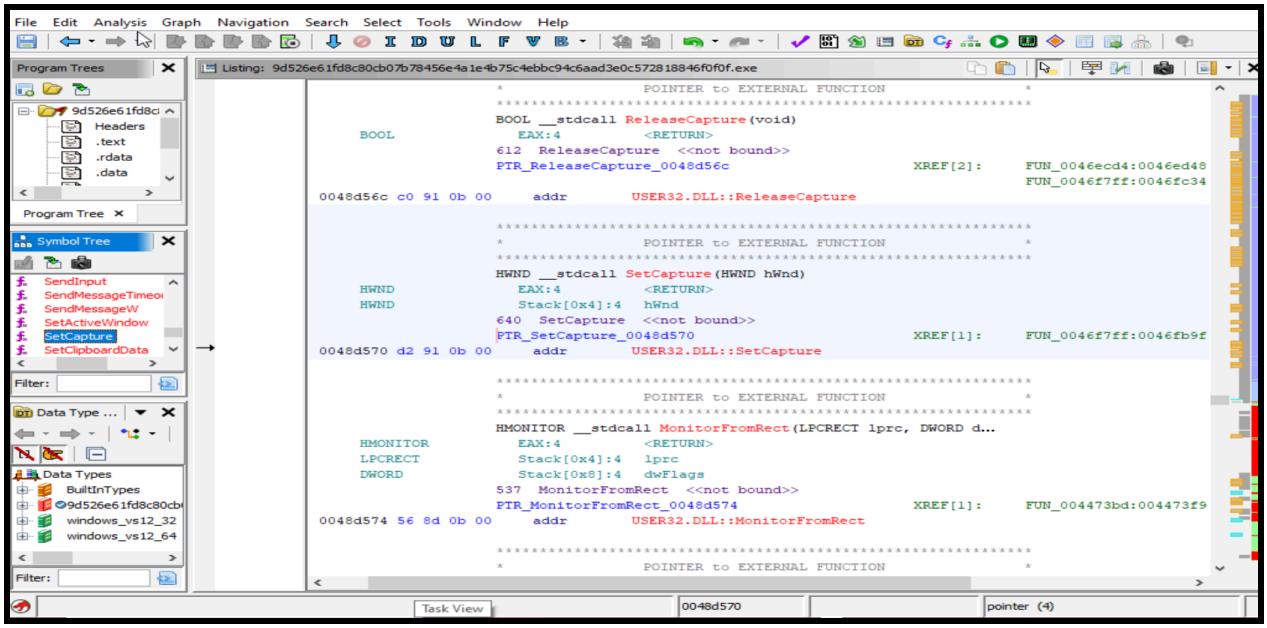
To start Ghidra, run the ghidraRun script and create a non-shared project. Import your binary file via *File → Import*, and let Ghidra detect the format. After import, select the file to open it in the CodeBrowser tool. When prompted, start analysis with default options to begin exploring the executable.

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.

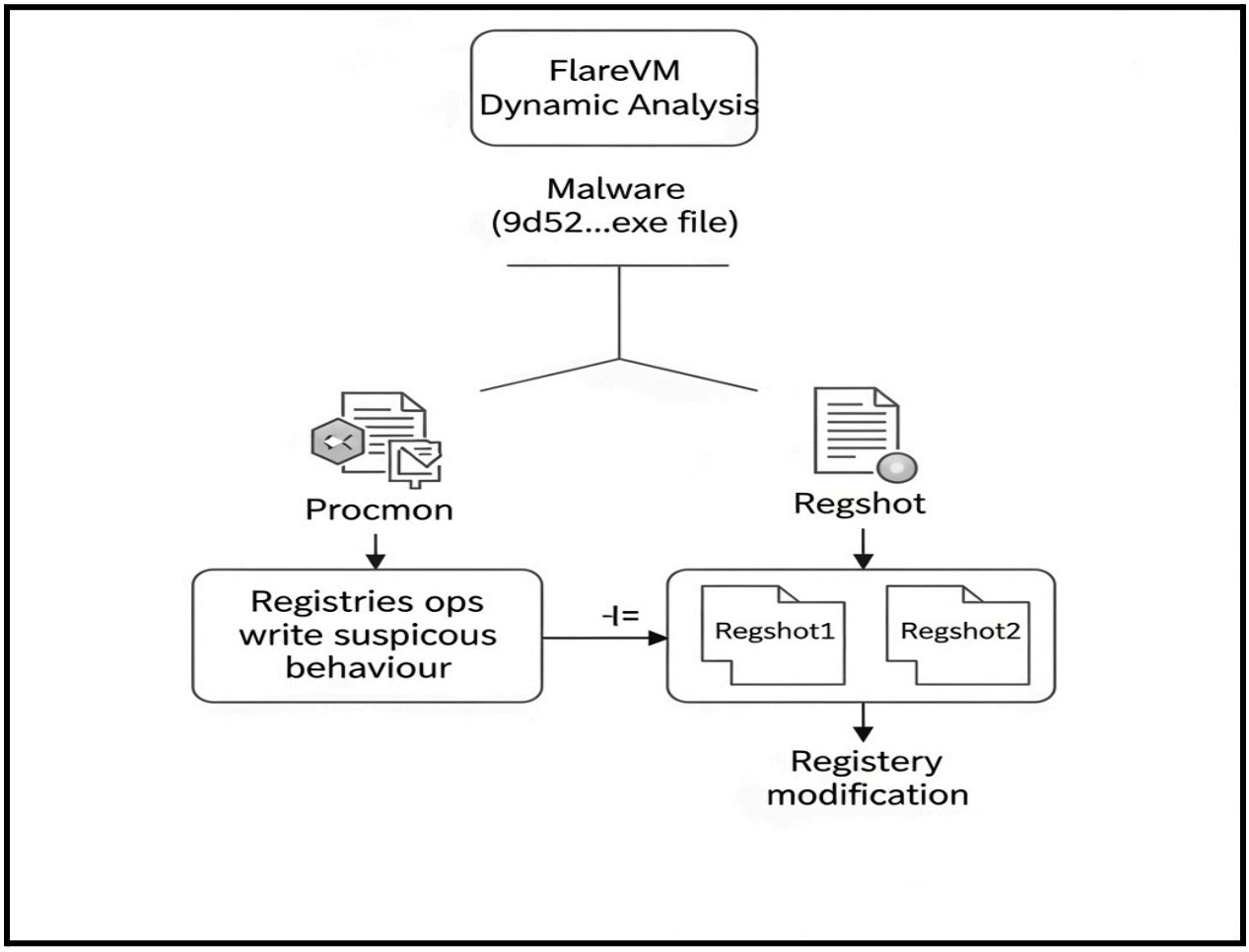


I conducted a static analysis of the sample 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe using Ghidra. The binary revealed several suspicious imports from USER32.DLL, including mouse_event, ExitWindowsEx, SetCapture, and MonitorFromRect. These functions suggest the malware may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


File Edit Analysis Graph Navigation Search Select Tools Window Help
Program Trees
Listing: 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe
Symbol Tree
SendInput
SendMessageTimeo
SendMessageW
SetActiveWindow
SetCapture
SetClipboardData
Data Type ...
Data Types
BuiltInTypes
windows_vs12_32
windows_vs12_64
POINTER to EXTERNAL FUNCTION
BOOL __stdcall ReleaseCapture(void)
BOOL EAX:4 <RETURN>
612 ReleaseCapture <<not bound>>
PTR_ReleaseCapture_0048d56c XREF[2]: FUN_0046ecd4:0046ed48
FUN_0046f7ff:0046fc34
0048d56c c0 91 0b 00 addr USER32.DLL::ReleaseCapture
HWND __stdcall SetCapture(HWND hWnd)
HWND EAX:4 <RETURN>
HWND Stack[0x4]:4 hWnd
640 SetCapture <<not bound>>
PTR_SetCapture_0048d570 XREF[1]: FUN_0046f7ff:0046fb9f
0048d570 d2 91 0b 00 addr USER32.DLL::SetCapture
HMONITOR __stdcall MonitorFromRect(LPCRECT lprc, DWORD d...
HMONITOR EAX:4 <RETURN>
LPCRECT Stack[0x4]:4 lprc
DWORD Stack[0x8]:4 dwFlags
537 MonitorFromRect <<not bound>>
PTR_MonitorFromRect_0048d574 XREF[1]: FUN_004473bd:004473f9
0048d574 56 8d 0b 00 addr USER32.DLL::MonitorFromRect
Task View
0048d570
pointer (4)

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.

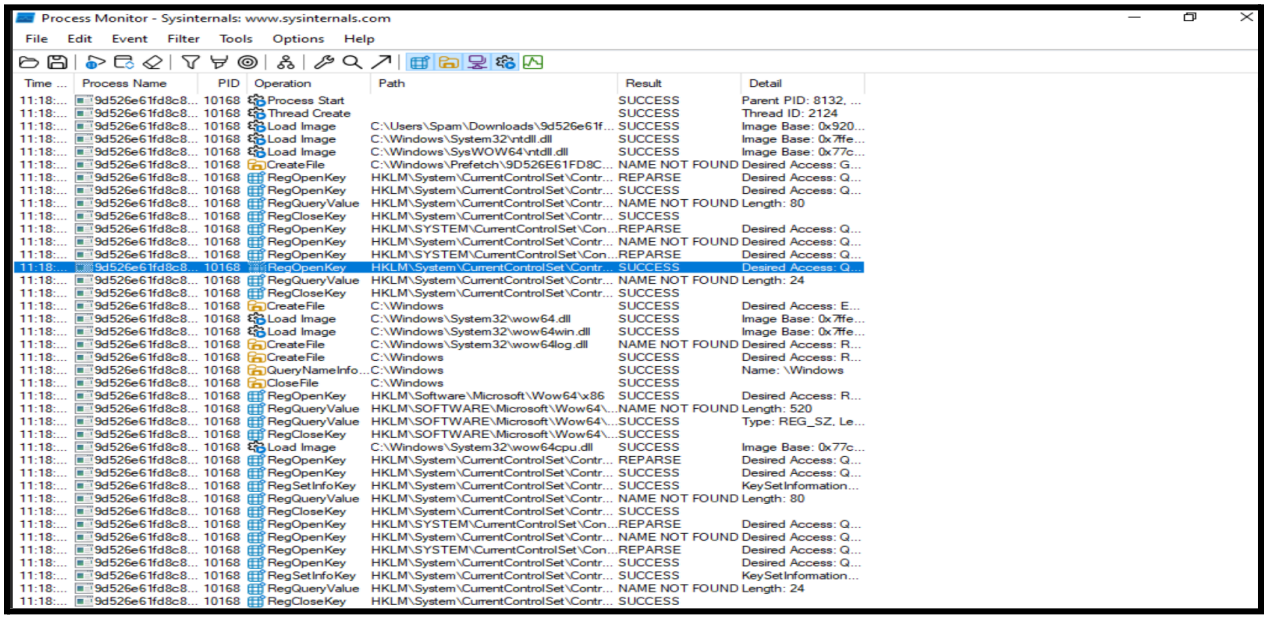


10.1 Process Monitor(Procon)

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.

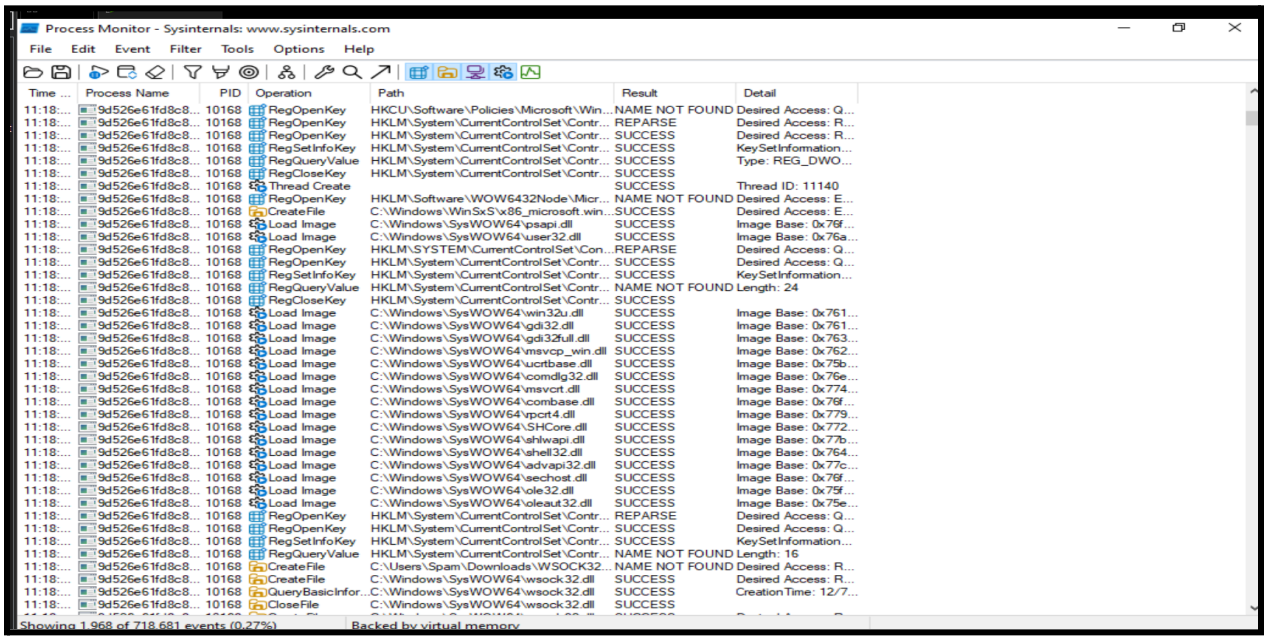
Steps to run Procon:

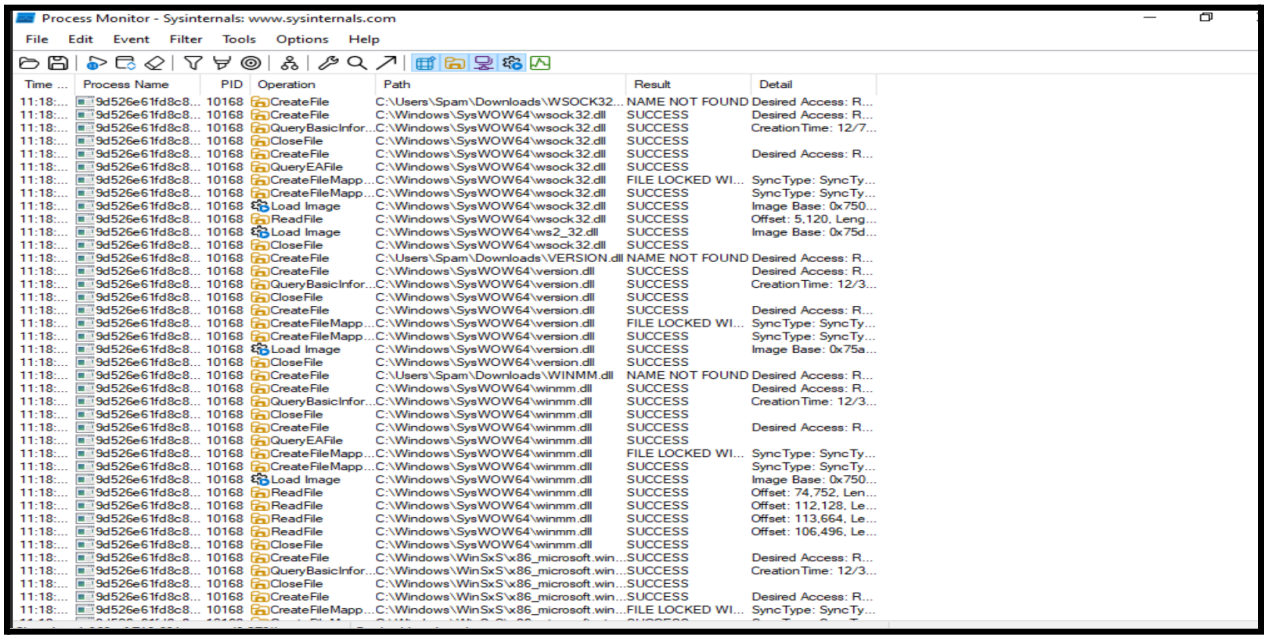
1. Run Procon as the administrator.
2. Set a filter: Go to **Filter → Filter...**
   Set: Process Name → is → 9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe(Your malware.exe file)
3. Click Add and Ok
4. We need to capture the procmon events.
5. Look for suspicious file writes, registry edits, process creation, or network activity.



Regarding the Procmon logs, the executable `9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe` extensively interacts with the Windows Registry, performing numerous `RegOpenKey` operations,

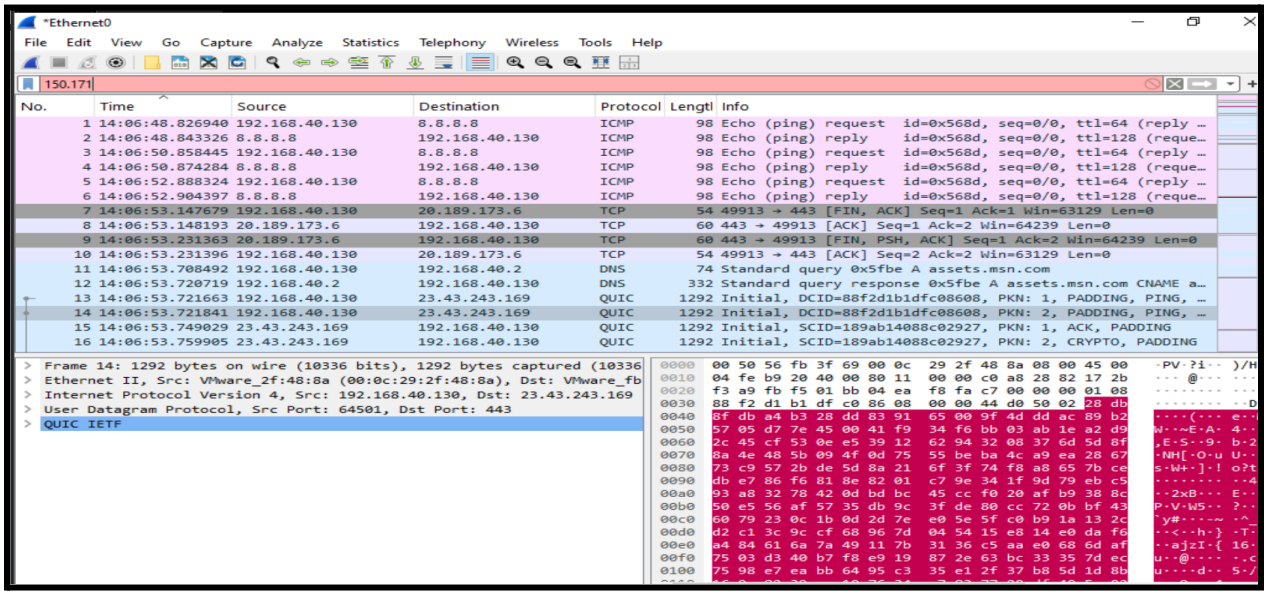
often probing for non-existent paths. The process also frequently uses `RegQueryValue` to read registry data and employs `RegSetValue` to modify or create registry entries. Furthermore, the presence of `RegSetInfoKey` alongside these other registry activities indicates attempts by the process to not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





10.2 Wireshark

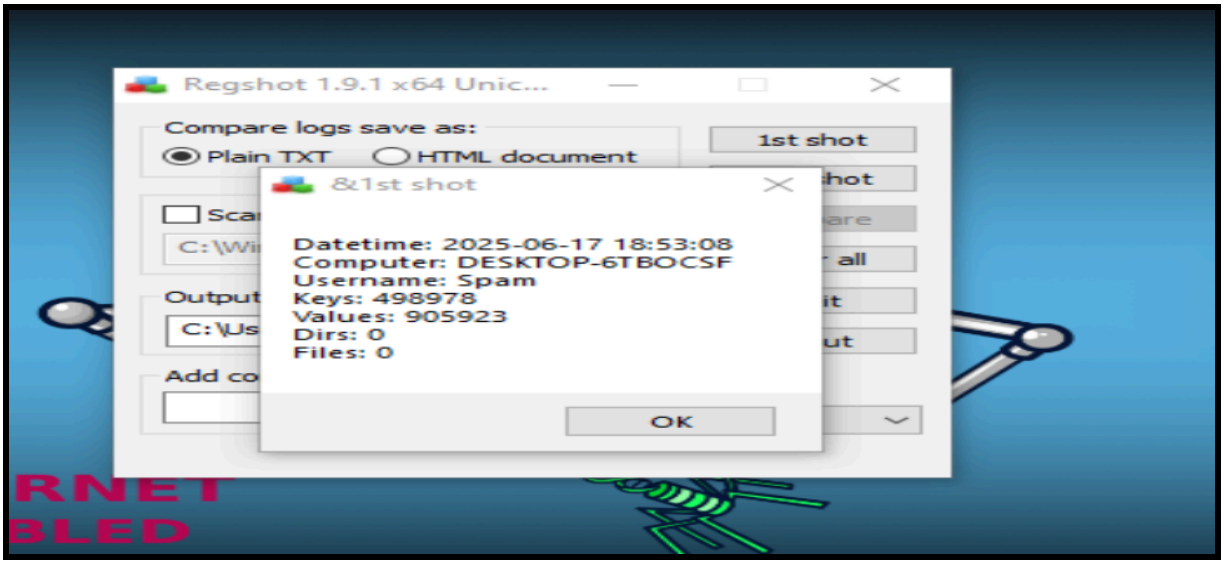
The malware from IP `192.168.40.130` was observed pinging Google's DNS (`8.8.8.8`) to check for internet access. It also initiated encrypted traffic using the QUIC protocol to `23.43.243.169`, a method often used to avoid detection. Another encrypted connection occurred with `150.171.28.11`, suggesting 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.



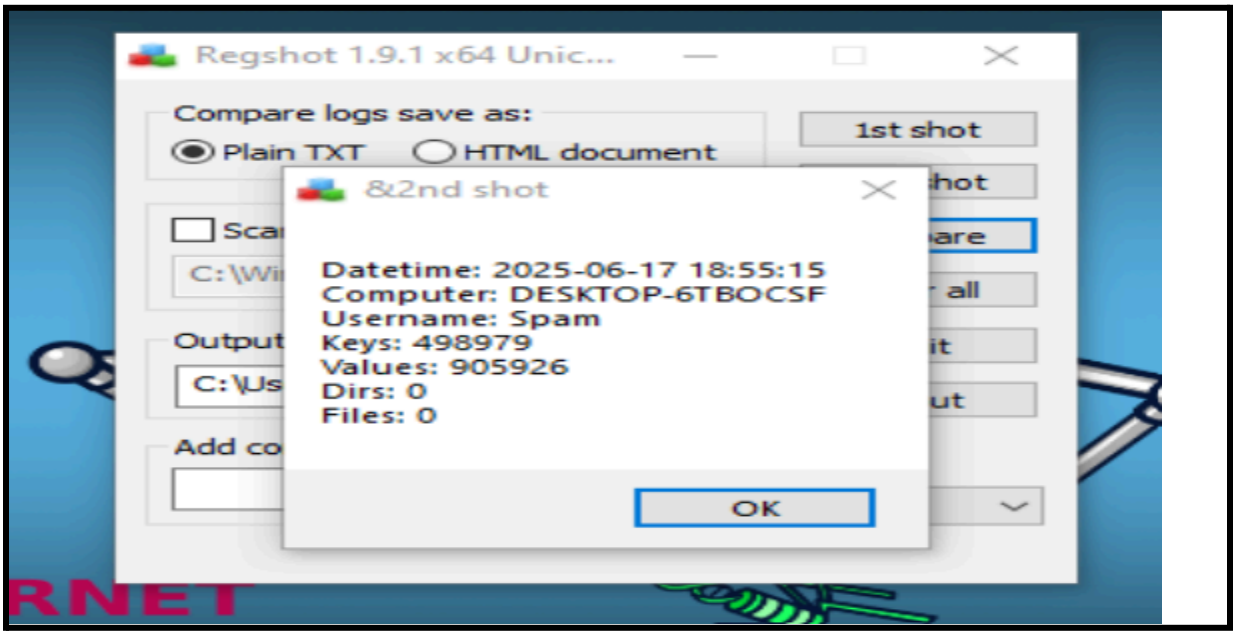
10.3 Regshot

To further investigate the registry modifications performed by malware_sim.exe, the Registry `9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe` Snapshot Tool (Regshot) was utilised. Two snapshots of the system registry were taken.

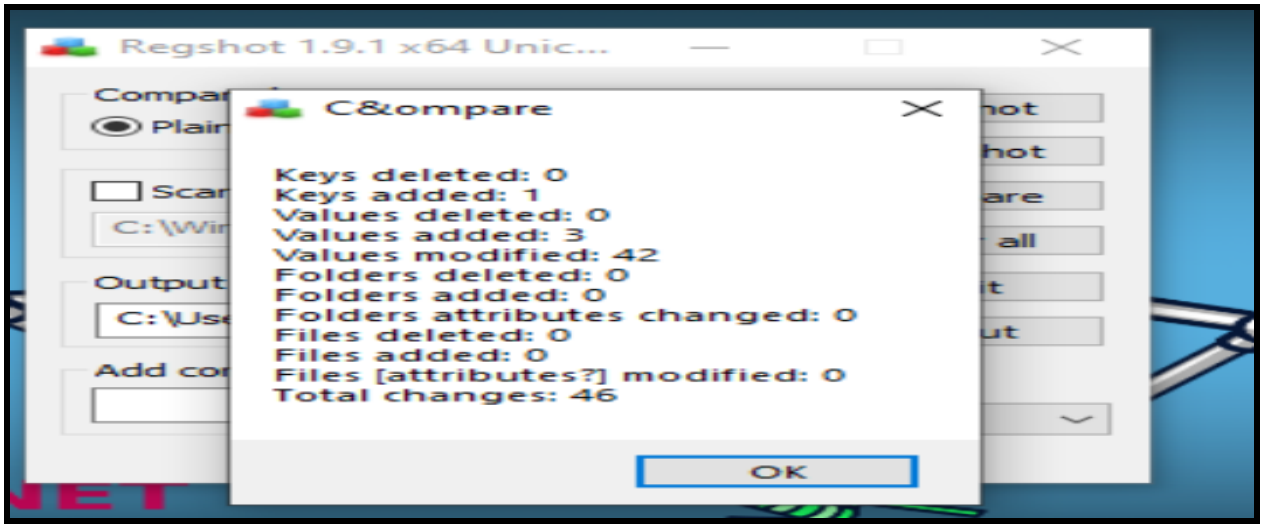
1st shot (Baseline): A baseline snapshot of the registry was captured before executing to record the initial state.



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`.


Regshot 1.9.1 x64 Unic...
C&ompare
Keys deleted: 0
Keys added: 1
Values deleted: 0
Values added: 3
Values modified: 42
Folders deleted: 0
Folders added: 0
Folders attributes changed: 0
Files deleted: 0
Files added: 0
Files [attributes?] modified: 0
Total changes: 46
OK

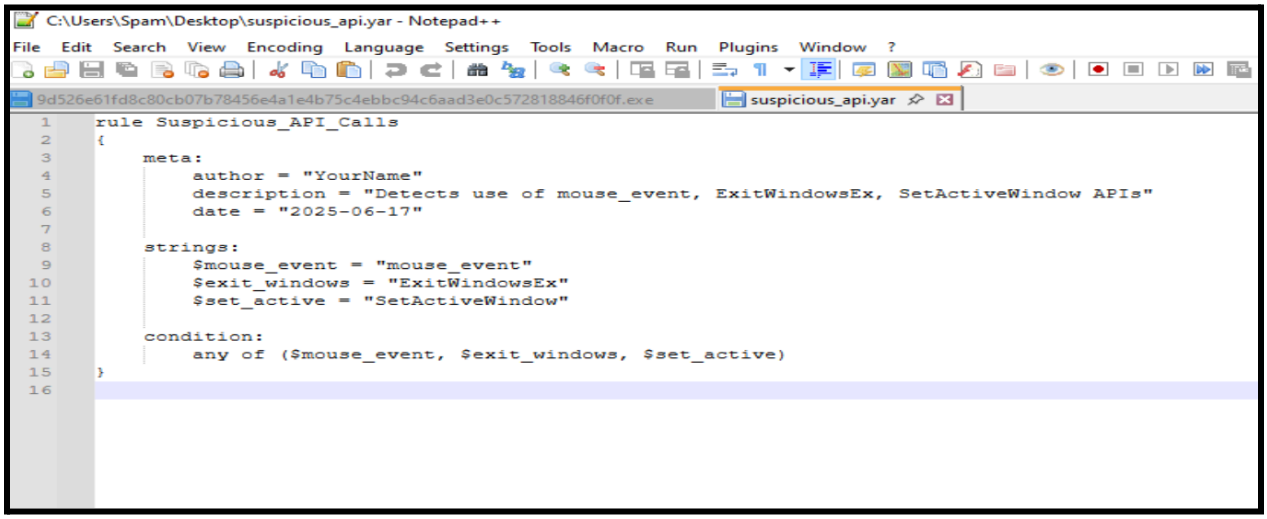

~res-x64.txt - Notepad
File Edit Format View Help
Regshot 1.9.1 x64 Unicode (beta r321)
Comments:
Datetime: 2025-06-17 18:53:08, 2025-06-17 18:55:15
Computer: DESKTOP-6TBOCSF, DESKTOP-6TBOCSF
Username: Spam, Spam
Keys added: 1
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\9060
Values added: 3
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\9060\Terminator: "HAM"
HKLM\SOFTWARE\Microsoft\Windows\Windows Error Reporting\TermReason\9060\Reason: 0x00000004
HKLM\SOFTWARE\Microsoft\Windows\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\SettingsRequests\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\Data\418A073AA3BC1C75: 86 0C 00 00 00 00 00 00 04 00 04 00 01

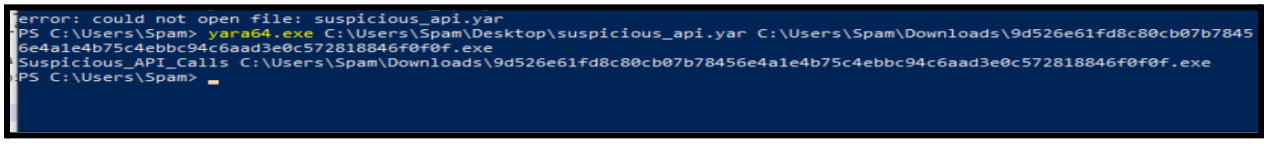
11. IOC Extractors

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

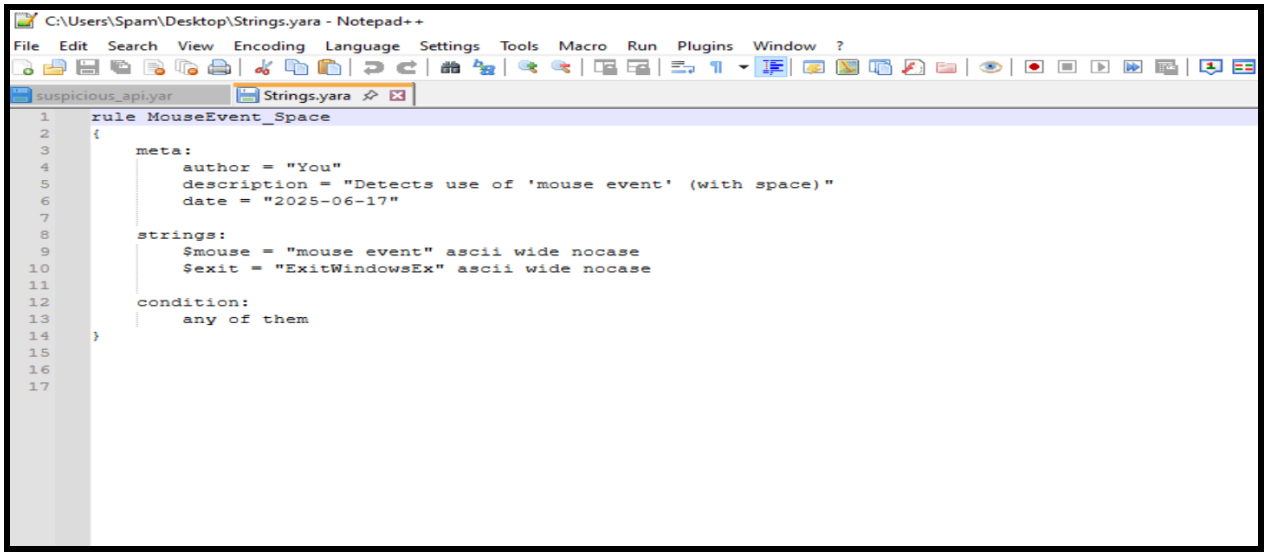
An initial Yara rule was created to detect specific characteristics within `9d526e61fd8c80cb07b78456e4a1e4b75c4ebbc94c6aad3e0c572818846f0f0f.exe`, such as defined strings. Scanning the malware with this rule.

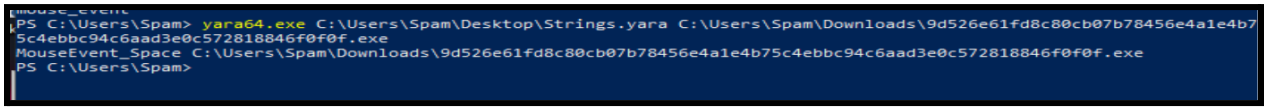




Yara rules for Strings.

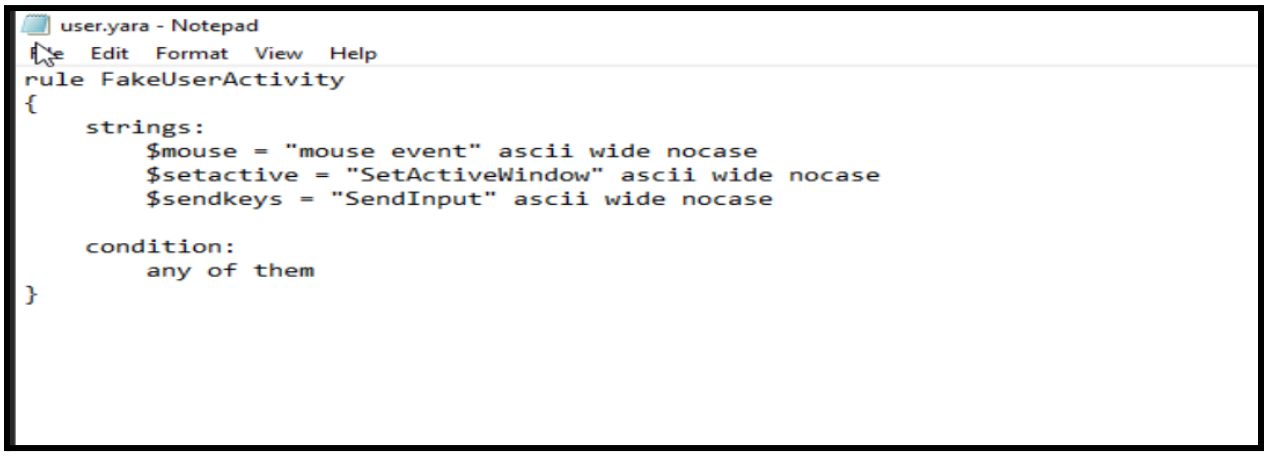
These strings represent Windows API calls used to simulate user activity (`mouse_event`), force system shutdown (`ExitWindowsEx`), and control window focus (`SetActiveWindow`). Malware often abuses them to evade detection, disrupt users, or automate interactions.

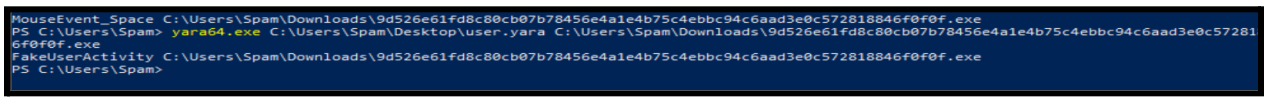




Suspicious User Behaviour

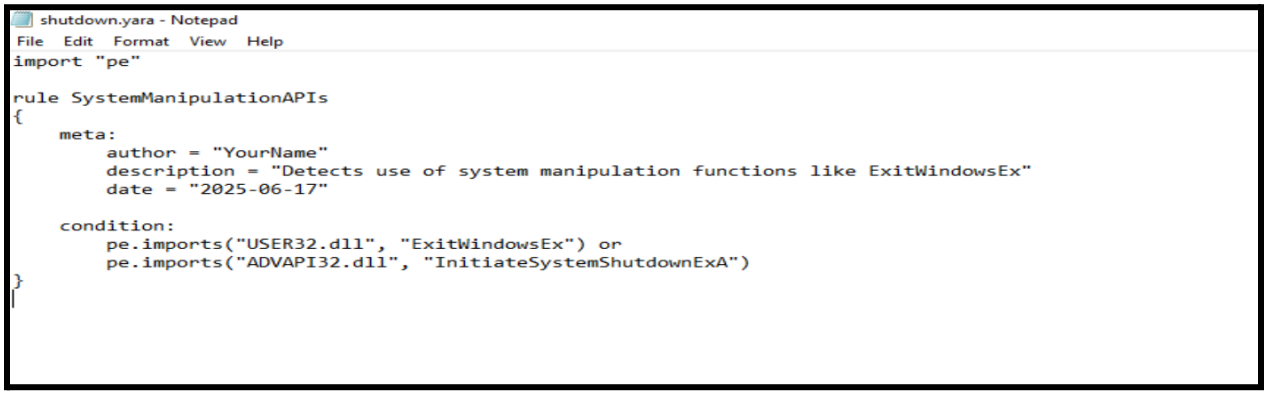
These represent Windows API calls used to simulate user activity (`mouse_event`), force system shutdown (`ExitWindowsEx`), and control window focus (`SetActiveWindow`). Malware often abuses them to evade detection, disrupt users, or automate interactions.

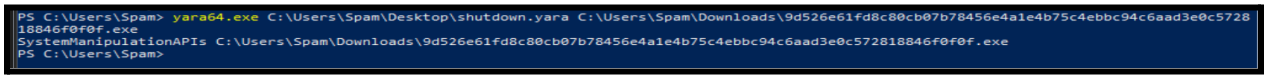




Detect Shutdown

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.





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.

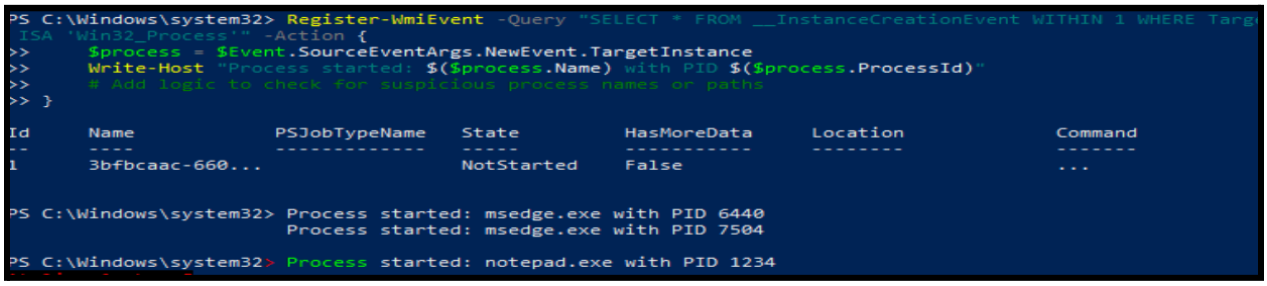
Behavioural Monitoring via Process Creation Event Tracking

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.

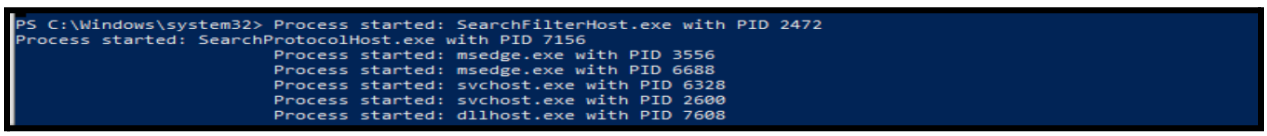
Steps:

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)"
}
```



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



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.