



**International Journal of Electronic Security and Digital Forensics**

ISSN online: 1751-9128 - ISSN print: 1751-911X  
<https://www.inderscience.com/ijesdf>

---

## **Forensic investigation and analysis of malware in Windows OS**

Frank Fiadufe, Krishna Modi, Kapil Shukla, Felix O. Etyang

**DOI:** [10.1504/IJESDF.2025.10062010](https://doi.org/10.1504/IJESDF.2025.10062010)

**Article History:**

|                   |                   |
|-------------------|-------------------|
| Received:         | 15 June 2023      |
| Last revised:     | 24 August 2023    |
| Accepted:         | 21 September 2023 |
| Published online: | 23 December 2024  |

---

## Forensic investigation and analysis of malware in Windows OS

---

Frank Fiadufe\*, Krishna Modi and  
Kapil Shukla

Department of Forensic Sciences,  
National Forensic Sciences University,  
Gandhinagar, India  
Email: frankkingdom300@gmail.com  
Email: krishna.modi@nfsu.ac.in  
Email: kapil.shukla@nfsu.ac.in  
\*Corresponding author

Felix O. Etyang

Department of Digital Forensics and Information Security,  
National Forensic Sciences University,  
Gandhinagar, India  
Email: felixetyang54@gmail.com

**Abstract:** Malware has become a pervasive concern for malware analysts and digital forensics. This research investigates malware forensics to detect, investigate, and analyse malicious software. The research examines the application of digital forensic science to dissect threat vectors, specifically malware, shedding light on their behaviour on computer hard disks and memory. Using various digital forensic tools, memory forensics, and hard disk forensics are performed on an infected Windows 7 OS, followed by static and dynamic analysis of malicious software. Memory samples are analysed using volatility for memory forensics, while disk images are analysed using autopsy for hard disk forensics. The malware's functionality is fully comprehended through meticulous extraction and analysis. A robust framework for malware forensic investigation emerges, facilitating detection, analysis, and understanding of malware behaviour. This research underscores the significance of integrating digital forensics tools and techniques to combat evolving malware threats effectively.

**Keywords:** digital forensics; memory forensics; hard disk forensics; static and dynamic analysis.

**Reference** to this paper should be made as follows: Fiadufe, F., Modi, K., Shukla, K. and Etyang, F.O. (2025) 'Forensic investigation and analysis of malware in Windows OS', *Int. J. Electronic Security and Digital Forensics*, Vol. 17, Nos. 1/2, pp.169–182.

**Biographical notes:** Frank Fiadufe received his Master's in Forensic Science with specialisation in Cyber Forensics from the National Forensic Sciences University, India. He attained his first degree in Forensic Science from the University of Cape Coast, Ghana and an associate degree from the University of Cordoba, Spain. He served as a teaching and research assistant at the

University of Cape Coast, Ghana. His current active research includes forensic investigation of malware in Windows 10 OS and its comparison with Windows 7 OS.

Krishna Modi is an academician with 5+ years of teaching experience. She received her Bachelor's and Master's in Computer Engineering from Gujarat Technological University. She is currently pursuing her PhD from Indus University. Her area of interest is machine learning, cyber security and digital forensics.

Kapil Shukla is serving as a teaching and research assistant at School of Forensic Science, National Forensics University, Gandhinagar, India. He is having 14 years of experience in academics. He is currently pursuing his PhD in field of Machine Learning. He has cleared GSET examination for Lectureship in 2012. He is life member of CSI and ISTE.

Felix O. Etyang is a Master holder in Digital Forensics and Information Security from National Forensic Sciences University, India. He attained his Bachelor's in Information and Communications Technology from Laikipia University, Kenya. He is a well-versed in all aspects of cyber security, including incident response and management, IT risk and management, cyber laws, ethical hacking, digital forensics, and information security. He is currently a PhD candidate.

---

## 1 Introduction

Malware forensics entails systematically examining a computer system to identify and assess malicious software's presence and ascertain its impact on the system. The primary objective of malware forensics is to ascertain the *modus operandi* employed by the malware to gain unauthorised access to the system, comprehensively understand its activities during its presence, and evaluate the extent of the resultant harm inflicted. They may be sent to the system by hackers through phishing attacks and may also be downloaded mistakenly from the internet. Malware, when used by attackers, may perform several functions in the victim's machine (Tekiner et al., 2021). It may be used to get unauthorised access to victim machines, steal data, demand ransom, and several others.

Malware is malicious software that compromises the system's functionality when present in a computer system. It affects the confidentiality, integrity, and availability of data and the entire system. Confidentiality, integrity, and availability, known simply as the CIA triad, are the core concepts guarding the Information security system (Pawar and Palivela, 2022). The effect of malware activities may lead to severe damage or loss of critical data. It is, therefore, very important to quickly detect and analyse malware present in a system. In dealing with malware, time is essential to prevent data loss or severe damage. Malware forensic analysts analyse infected systems or machines and examine the activities of files and programs using Forensic tools to detect malware. Additionally, the researchers evaluate the malware's activities within the system to ascertain the malicious software's intended objectives.

Hard disk forensics is a critical field in computer science that examines and analyses hard disk drives. Hard disks are integral components of computers, storing many

applications, operating systems, and data files. Recently, it has emerged as a crucial resource for acquiring information and evidence pertinent to forensic investigations. Information or evidence can be retrieved from the log files, registry entries, malicious files, manipulations from date-time tampering, web-browser history, and even installed programs and software. Memory forensics is very important in malware forensics as it analyses the RAM dump of the system. In memory forensics, various kernel data structures available inside the physical memory dump are analysed to determine running processes and their associated artefacts (Dija et al., 2020).

## **2 Related work**

Computer Forensics is a burgeoning and rapidly evolving discipline, with the Bureau of Labor Statistics projecting a substantial 17% growth in the computer forensics industry from 2016 to 2026 (Tsado and Kim, 2022). For this reason, cybercriminals utilise the rapid development in cyberspace for their interest. They utilise malware as a threat vector to launch attacks against their victims. Malware can be categorised into two different types such as fileless malware and host-dependent malware (Kara, 2022). Two common methods for analysing malware are static analysis and dynamic analysis. Static malware analysis is done without actually running the malware. The executable file has different sections and compactness (Ijaz et al., 2019), while in dynamic analysis, the behaviour of malware is observed while running the malware in a controlled environment (Ahirao, 2021). Malware changes the registry key maliciously and takes the privileged mode of the operating system when the malware is executed during dynamic analysis (Ijaz et al., 2019). Rodrigo et al. (2021) combined the static and dynamic malware approaches for malware detection; they used dynamic traces and static feature occurrences for malware detection. Agrawal et al. (2020) proposed a manual method for dynamic malware analysis so that security researchers or malware analysts can easily understand the behaviour of malware and implement a better solution for reducing the risk of malware attacks in the future. They focused their experiment on running the malware in a Windows 7 OS hosted in a virtual environment. This experiment used Volatility, FTK Imager Lite Forensics toolkit, and other digital forensic tools in a virtual and safe environment.

In the forensic investigation of malware, memory forensics is very important to understand and show forensic artefacts necessary to fully understand the attack or incident. The well-known memory forensics tools, Volatility, and Autopsy, have different memory forensics capabilities when used in separate malware behaviour situations. Also, each device used at various periods of operation for memory analysis consumes different measurements of CPU and memory. Volatility is known to have the most accurate memory analysis, while Autopsy requires more memory to examine a memory image file (Daghmehchi Firoozjaei et al., 2022). Memory forensic investigation involves processes that are broadly categorised into three main distinguished processes:

- memory procurement
- data analytics
- evidence recuperating.

Each of these processes requires different digital forensic tools like FTK, Belkasoft, Autopsy, Volatility, DumpIt, and wxHexEditor, among many others, with some of these tools existing as a command-line framework (Sweeney, 2002). The computer hard disk is a main source of evidence against cyberattacks as it maintains its digital information. By performing disk forensics which involves the investigation of file systems on the retrieved hard disk, several very important artefacts or information could be generated, which may greatly help in the forensic investigation. The hard disk contains free or slack space where sensitive or malicious information may be hidden (Meshram and Patil, 2018).

Volatility is a command-line memory analysis tool that is run on Windows Powershell. It is an open-source tool used for parsing and analysing memory dump files. This tool uses many plugins to identify malicious software (Chayal et al., 2022).

Disk forensics is another very important technique in digital forensics that deals with collecting, analysing, and preserving electronic data stored on computer hard drives and other storage devices that help investigate and analyse digital evidence. Drive creation is the most common way of separating data from hard drives, memory cards, pen drives, and others (Mandela et al., 2023). Autopsy is a forensic tool capable of analysing disk images to determine possible. Autopsy is a GUI-based program used to explore computers' hard drives. Autopsy is easy to use, flexible, fast, and cost-effective (Sachdeva et al., 2020).

### **3 Problem statement**

Detecting and analysing malware in today's digital environments, especially windows-based ones, requires a more holistic and coordinated strategy. Current approaches generally fail to account for the complex interplay between malware's actions in memory and their effects on-disk structures, which results in limited understanding and ineffective defences against ever-evolving threats. Given the current state of knowledge, it is clear that a more comprehensive framework is required to accurately uncover, investigate, and analyse malicious activities. This would strengthen computer systems' resistance to sophisticated malware attacks and improve cyber defence.

### **4 Research methodology**

This section details the extensive research methods used to achieve the study's goals, which included the systematic and rigorous detection, examination, and analysis of potentially harmful software using a forensic approach.

#### *4.1 Research design*

The research combined analytical and empirical approaches, combining memory forensics and hard drive forensics to paint a complete picture of malware's functioning. The study made use of a compromised virtual machine running Windows 7.

## 4.2 Data collection

- 1 *Memory forensics*: Taking RAM dumps from the infected Windows 7 VM was crucial to the memory analysis. Volatility, a command-line memory analysis program, was used to investigate kernel data structures, track down active processes, and locate possible remnants of the malware.
- 2 *Hard disk forensics*: Disk images of the infected Windows 7 virtual machine were acquired using the FTK Imager. The captured disk images were subsequently analysed using Autopsy, a GUI-based forensic tool. Various file system components, such as log files, registry entries, malicious files, and user-specific data, were examined for evidence of the malware's presence and activities.

## 4.3 Data analysis

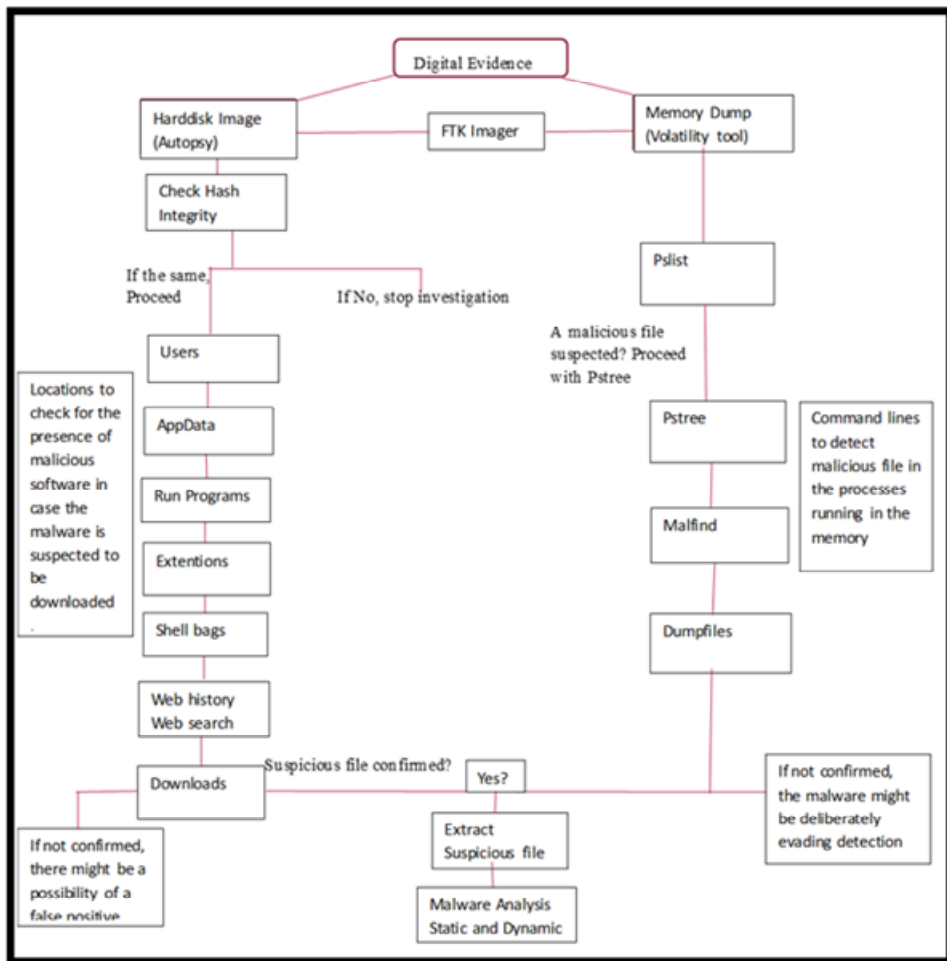
- 1 *Memory forensics analysis*: The Volatility tool was central in analysing the captured memory dumps. Commands such as 'windows.pslist', 'windows.pstree', and 'windows.malfind' were utilised to identify running processes, parent-child relationships, and potential indications of malicious behaviour. The goal was to pinpoint suspicious processes masquerading as legitimate ones.
- 2 *Hard disk forensics analysis*: Autopsy facilitated the analysis of the acquired disk images. Focusing on key areas like AppData, run programs, extensions, shell bags, web history, web search, and downloads, the investigation sought to reveal traces of the malware's actions, potential sources of infection, and user involvement.

## 4.4 Malware analysis

- 1 *Static malware analysis*: The suspicious executable, identified as 'tool.exe', underwent static analysis to determine its malicious nature. The file was scanned using VirusTotal to ascertain its detection status across various antivirus engines. Additionally, the executable was examined for obfuscation techniques, revealing its use of UPX1 packing.
- 2 *Dynamic malware analysis*: Procmon was employed to perform dynamic analysis by monitoring the behaviour of 'tool.exe' in a controlled environment. This allowed for observing the malware's changes to the system, including file modifications, network communications, and potential attempts to evade analysis.

# 5 Proposed framework

In light of the experiment conducted, a detailed plan for a malware forensic investigation was developed in a proposed framework. This framework explains the investigation plan from hash integrity check and gives the locations malware forensic investigators could look to during malware forensic investigation and also gives malware analysis techniques.

**Figure 1** Proposed framework (see online version for colours)

The framework also answers the research objectives and questions discussed below:

- The first part of the framework explains the evidence and data acquisition, which is a critical part of digital forensic investigation procedures. In this research, with the digital evidence provided (Windows 7 OS), an FTK imager was used to create a disk image for hard disk forensics using Autopsy forensic tool and to capture the memory for memory forensics analysis using the Volatility tool.
- The next step is to check the evidence integrity by comparing the hashes obtained from FTK imaging and the hashes obtained from using Autopsy. This is to ensure that the evidence was not compromised before the investigation. According to the framework, the investigation can proceed if the hashes are identical. Still, the investigation should halt if there is a mismatch since it suggests the evidence is compromised.

- After establishing integrity, the framework recommends some locations using the Autopsy tool where evidence relating to the malware could be found. A very important part of the investigation is to determine which user of the machine the attack came from. This simplifies the investigation to a specific user and also narrows down the investigation. After the user is confirmed, locations like the AppData, run programs, extensions, shell bags, web history, and web search and downloads could reveal very important information regarding the malware.
- Using the Volatility tool, the provided commands in the framework could reveal all the running processes, their PID and PPIDs, and their child processes. Also, the malfind command could list the suspicious processes.
- When the malware is detected, it is extracted from both hard disk and memory analysis into a safe folder and ready for malware analysis by static and dynamic analysis to better understand its functionalities.

**Figure 2** windows.pstree (see online version for colours)

win10 [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Select Windows PowerShell

```
PS C:\Users\Bismark\Desktop>volatility3-1.0.0\volatility3-1.0.0\python vol.py -f Z:\analysis\adware\memdump.mem windows.pstree
```

Volatility 3 Framework 1.0.0

Progress: 100.00 PDB scanning finished

| PID      | PPID | ImageFileName     | Offset(V) | Threads | Handles | SessionId | Wow64 | CreateTime                 | ExitTime |
|----------|------|-------------------|-----------|---------|---------|-----------|-------|----------------------------|----------|
| 4        | 0    | System            | 0x0       | 556     | N/A     | False     |       | 2023-04-04 19:00:55.000000 | N/A      |
| * 272    | 4    | smss.exe          | 0x0       | 2       | N/A     | False     |       | 2023-04-04 19:00:55.000000 | N/A      |
| 352      | 344  | csrss.exe         | 0x0       | 1       | 443     | 0         | False | 2023-04-04 19:00:56.000000 | N/A      |
| 404      | 344  | wininit.exe       | 0x0       | 3       | 76      | 0         | False | 2023-04-04 19:00:56.000000 | N/A      |
| * 532    | 404  | lsass.exe         | 0x0       | 154     | 0       | False     |       | 2023-04-04 19:00:57.000000 | N/A      |
| * 508    | 404  | services.exe      | 0x0       | 9       | 205     | 0         | False | 2023-04-04 19:00:56.000000 | N/A      |
| ** 2572  | 508  | mscorsvw.exe      | 0x0       | 5       | 77      | 0         | False | 2023-04-04 19:03:06.000000 | N/A      |
| ** 916   | 508  | svchost.exe       | 0x0       | 48      | 573     | 0         | False | 2023-04-04 19:00:58.000000 | N/A      |
| ** 2716  | 508  | spoolsv.exe       | 0x0       | 4       | 147     | 0         | False | 2023-04-04 19:03:06.000000 | N/A      |
| ** 3756  | 508  | taskhost.exe      | 0x0       | 4       | 132     | 1         | False | 2023-04-04 19:16:23.000000 | N/A      |
| ** 692   | 508  | VBoxService.exe   | 0x0       | 12      | 131     | 0         | False | 2023-04-04 19:00:57.000000 | N/A      |
| ** 1080  | 508  | svchost.exe       | 0x0       | 15      | 460     | 0         | False | 2023-04-04 19:00:59.000000 | N/A      |
| ** 1340  | 508  | svchost.exe       | 0x0       | 10      | 188     | 0         | False | 2023-04-04 19:01:00.000000 | N/A      |
| ** 956   | 508  | svchost.exe       | 0x0       | 33      | 987     | 0         | False | 2023-04-04 19:00:58.000000 | N/A      |
| ** 2752  | 508  | svchost.exe       | 0x0       | 11      | 352     | 0         | False | 2023-04-04 19:03:06.000000 | N/A      |
| ** 1220  | 508  | spoolsv.exe       | 0x0       | 13      | 282     | 0         | False | 2023-04-04 19:00:59.000000 | N/A      |
| ** 840   | 508  | svchost.exe       | 0x0       | 19      | 464     | 0         | False | 2023-04-04 19:00:58.000000 | N/A      |
| *** 1260 | 840  | audiiodg.exe      | 0x0       | 4       | 123     | 0         | False | 2023-04-04 19:34:48.000000 | N/A      |
| ** 1992  | 508  | SearchIndexer.exe | 0x0       | 14      | 667     | 0         | False | 2023-04-04 19:01:38.000000 | N/A      |
| ** 2012  | 508  | taskhost.exe      | 0x0       | 9       | 254     | 1         | False | 2023-04-04 19:01:31.000000 | N/A      |
| ** 1248  | 508  | svchost.exe       | 0x0       | 19      | 387     | 0         | False | 2023-04-04 19:00:59.000000 | N/A      |
| ** 748   | 508  | svchost.exe       | 0x0       | 10      | 265     | 0         | False | 2023-04-04 19:00:58.000000 | N/A      |
| ** 1388  | 508  | svchost.exe       | 0x0       | 25      | 305     | 0         | False | 2023-04-04 19:01:00.000000 | N/A      |
| ** 2540  | 508  | mscorsvw.exe      | 0x0       | 5       | 84      | 0         | True  | 2023-04-04 19:03:06.000000 | N/A      |
| ** 632   | 508  | svchost.exe       | 0x0       | 10      | 356     | 0         | False | 2023-04-04 19:00:57.000000 | N/A      |
| ** 888   | 508  | svchost.exe       | 0x0       | 20      | 450     | 0         | False | 2023-04-04 19:00:58.000000 | N/A      |
| *** 1052 | 888  | dwm.exe           | 0x0       | 3       | 95      | 1         | False | 2023-04-04 19:01:31.000000 | N/A      |
| * 524    | 404  | lsass.exe         | 0x0       | 7       | 568     | 0         | False | 2023-04-04 19:00:57.000000 | N/A      |
| 416      | 396  | csrss.exe         | 0x0       | 11      | 495     | 1         | False | 2023-04-04 19:00:56.000000 | N/A      |
| 452      | 396  | winlogon.exe      | 0x0       | 3       | 111     | 1         | False | 2023-04-04 19:00:56.000000 | N/A      |
| 1116     | 1048 | explorer.exe      | 0x0       | 28      | 902     | 1         | False | 2023-04-04 19:01:31.000000 | N/A      |
| * 3136   | 1116 | tool.exe          | 0x0       | 6       | 161     | 1         | True  | 2023-04-04 19:35:50.000000 | N/A      |
| * 3168   | 1116 | MSASCui.exe       | 0x0       | 7       | 171     | 1         | False | 2023-04-04 19:16:23.000000 | N/A      |
| * 2812   | 1116 | FTK Imager.exe    | 0x0       | 28      | 488     | 1         | False | 2023-04-04 19:37:57.000000 | N/A      |
| * 2000   | 1116 | VBoxTray.exe      | 0x0       | 13      | 139     | 1         | False | 2023-04-04 19:01:32.000000 | N/A      |
| * 668    | 1116 | chrome.exe        | 0x0       | 30      | 1220    | 1         | False | 2023-04-04 19:19:12.000000 | N/A      |
| ** 2336  | 668  | chrome.exe        | 0x0       | 14      | 195     | 1         | False | 2023-04-04 19:19:18.000000 | N/A      |
| ** 2340  | 668  | chrome.exe        | 0x0       | 8       | 119     | 1         | False | 2023-04-04 19:19:12.000000 | N/A      |
| ** 3480  | 668  | chrome.exe        | 0x0       | 7       | 143     | 1         | False | 2023-04-04 19:19:12.000000 | N/A      |
| ** 2424  | 668  | chrome.exe        | 0x0       | 15      | 243     | 1         | False | 2023-04-04 19:20:52.000000 | N/A      |
| ** 3900  | 668  | chrome.exe        | 0x0       | 18      | 328     | 1         | False | 2023-04-04 19:21:34.000000 | N/A      |
| ** 3176  | 668  | chrome.exe        | 0x0       | 11      | 212     | 1         | False | 2023-04-04 19:19:14.000000 | N/A      |
| ** 1940  | 668  | chrome.exe        | 0x0       | 15      | 320     | 1         | False | 2023-04-04 19:20:53.000000 | N/A      |
| ** 1516  | 668  | chrome.exe        | 0x0       | 11      | 183     | 1         | False | 2023-04-04 19:19:22.000000 | N/A      |
| ** 1676  | 668  | chrome.exe        | 0x0       | 15      | 301     | 1         | False | 2023-04-04 19:20:24.000000 | N/A      |
| ** 1752  | 668  | chrome.exe        | 0x0       | 14      | 230     | 1         | False | 2023-04-04 19:20:26.000000 | N/A      |
| ** 3832  | 668  | chrome.exe        | 0x0       | 18      | 345     | 1         | False | 2023-04-04 19:19:14.000000 | N/A      |
| ** 2864  | 668  | chrome.exe        | 0x0       | 15      | 256     | 1         | False | 2023-04-04 19:20:26.000000 | N/A      |
| ** 3032  | 668  | chrome.exe        | 0x0       | 13      | 207     | 1         | False | 2023-04-04 19:19:12.000000 | N/A      |
| ** 4092  | 668  | chrome.exe        | 0x0       | 15      | 331     | 1         | False | 2023-04-04 19:20:34.000000 | N/A      |
| 996      | 1956 | GoogleCrashHan    | 0x0       | 5       | 92      | 0         | True  | 2023-04-04 19:01:34.000000 | N/A      |
| 584      | 1956 | GoogleCrashHan    | 0x0       | 5       | 85      | 0         | False | 2023-04-04 19:01:34.000000 | N/A      |

## 6 Experiment and result

### 6.1 Memory forensics

Digital forensics encompasses memory forensics as a branch that deals mainly with analysing memory dump or RAM information to determine occurrences in a system's memory during a forensics investigation.

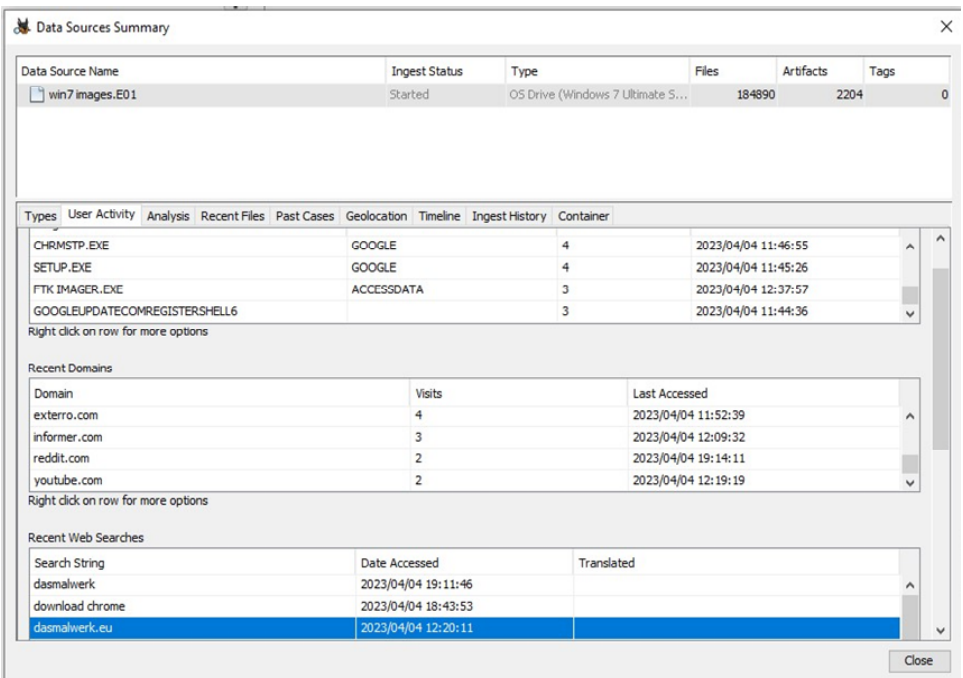
It is highly useful in investigating malware; running processes can be investigated using the Volatility tool to determine genuine processes spawned or masqueraded by the malicious software. 'python vol.py -f 'file location' windows.ps/tree command list all malicious processes masquerading as legitimate Windows processes.

This image shows that explorer.exe has spawned tool.exe with other processes which also spawned other chrome.exe processes. tool.exe with a PID of 3136 and a PPID of 1,116 represents the malicious file identified as a running process in the memory.

### 6.2 Hard disk forensics

Hard disk forensics is a very important branch of digital forensics that involves retrieving and analysing data collected from a computer's hard disk in the case of digital forensics investigation. With hard disk forensics, deleted or hidden data can be retrieved and analysed using specialised digital forensic tools.

**Figure 3** results of the disk (see online version for colours)



### 6.2.1 Locations to find forensic evidence users

It is important to investigate the users of the infected system to determine the user from which the attack started. We identified users as *Felix*, *default*, *public*, and *default user*. Felix had more logins than the others, so Felix was duly investigated along with other users.

### 6.2.2 AppData

AppData is a folder in the Windows OS that contains application specific-data and settings for each user account. It contains three subfolders, namely, Local, LocalLow, and Roaming.

Users/Felix/AppData/Local/Google/Chrome/UserData/Default/ServiceWorker/Cache Storage.

Information in the AppData could reveal files that have been dropped or downloaded. The IP traffic was investigated to determine the IP addresses the malware might communicate with. Some suspected IP addresses were discovered.

### 6.2.3 Run programs

Going through the Run Programs, another 'TOOL.EXE' file was opened from the same folder downloaded. The malware also accessed a file in the Windows registry, the *UsrClass.dat* –which stores user profile information.

### 6.2.4 Extensions

The file was investigated to determine the type. It was found to be an executable file. It was found among the executables in the Extensions.

### 6.2.5 Shell bags

This artefact in the Windows registry discloses information about the files, folders, and applications and stores the user's preferences for how the desktop and folders should be shown. It is an extremely important source of information during Digital Forensic investigations.

### 6.2.6 Web history and web search

The web history and web search were analysed to understand the malicious software's source better. It was discovered that the site 'dasmalwerk.eu', which is a malicious site for downloading malware, was searched severally through the user 'Felix's' login. This could prove that either someone got unauthorised access through Felix's login and downloaded the malware to infect the system or Felix did it himself knowingly or by mistake. The web search recorded several malicious site searches, proving intent.

### 6.2.7 Downloads

In the downloads folder of 'Felix-user,' a file was identified which was suspected to be malicious. This folder was opened, and the malicious file 'tool.exe' was identified. This

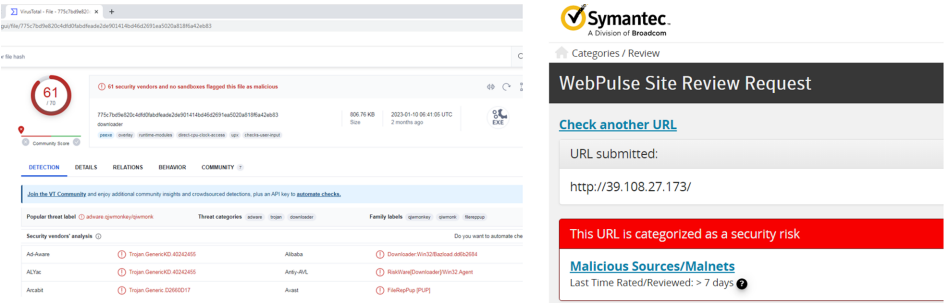
may prove that the source of the attack came from Felix downloading the malware from the internet.

6.3 Malware analysis

6.3.1 Static malware analysis

The extracted suspicious software is then scanned using Virustotal to determine whether anti-virus engines have flagged it as malicious. The IP address was also checked with *Symantec SiteReview*. The IP address was categorised as having security risk by threat actors.

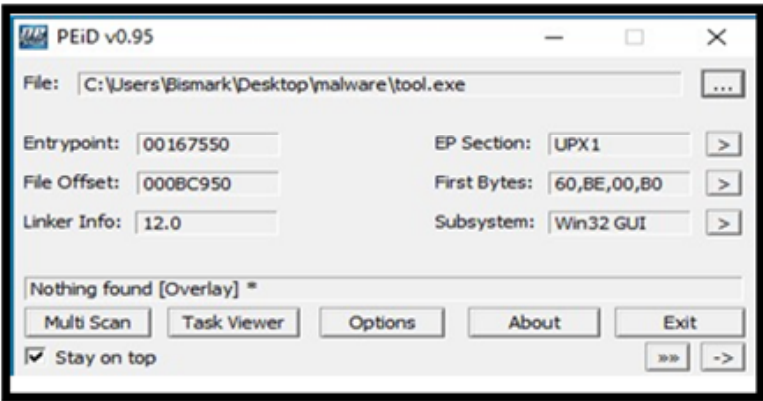
Figure 4 Analysis with virustotal.com (see online version for colours)



6.3.2 Obfuscation

The executable tool.exe was loaded into Peid to check whether the file was packed or unpacked. It was identified that the file was packed with UPX1, the entry point of the file was identified as 00167550, and the File Offset was identified as 000BC950.

Figure 5 Checking obfuscation using PEiD (see online version for colours)



### 6.3.3 Dynamic malware analysis Procmon

Procmon was duly utilised to monitor and observe the changes after running the malware by double-clicking on it.

**Figure 6** Analysis with Procmon (see online version for colours)

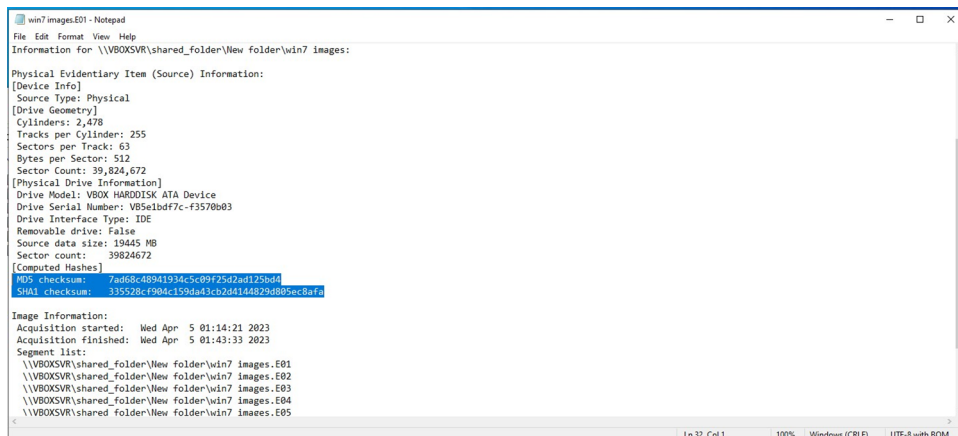
| Time ...  | Process Name | PID  | Operation        | Path                                    | Result         | Detail                 |
|-----------|--------------|------|------------------|-----------------------------------------|----------------|------------------------|
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\System32\wow64win.dll        | SUCCESS        | Offset: 323,584, Le... |
| 2:31:4... | notool.exe   | 3392 | CreateFile       | C:\Windows\System32\wow64log.dll        | NAME NOT FOUND | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\System32\wow64.dll           | SUCCESS        | Offset: 337,408, Le... |
| 2:31:4... | notool.exe   | 3392 | CreateFile       | C:\Windows                              | SUCCESS        | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | QueryNameInfo... | C:\Windows                              | SUCCESS        | Name: \Windows         |
| 2:31:4... | notool.exe   | 3392 | CloseFile        | C:\Windows                              | SUCCESS        |                        |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\Software\Microsoft\Wow64\86        | SUCCESS        | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\SOFTWARE\Microsoft\Wow64\...       | NAME NOT FOUND | Length: 520            |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\SOFTWARE\Microsoft\Wow64\...       | SUCCESS        | Type: REG_SZ, Le...    |
| 2:31:4... | notool.exe   | 3392 | RegCloseKey      | HKLM\SOFTWARE\Microsoft\Wow64\...       | SUCCESS        |                        |
| 2:31:4... | notool.exe   | 3392 | Load Image       | C:\Windows\System32\wow64cpu.dll        | SUCCESS        | Image Base: 0x772...   |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\System32\wow64cpu.dll        | SUCCESS        | Offset: 6,144, Leng... |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\System32\wow64cpu.dll        | SUCCESS        | Offset: 9,216, Leng... |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | REPARSE        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegSetInfoKey    | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        | KeySetInformation...   |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Length: 80             |
| 2:31:4... | notool.exe   | 3392 | RegCloseKey      | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        |                        |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\SYSTEM\CurrentControlSet\Con...    | REPARSE        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\SYSTEM\CurrentControlSet\Con...    | REPARSE        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegSetInfoKey    | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        | KeySetInformation...   |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Length: 24             |
| 2:31:4... | notool.exe   | 3392 | RegCloseKey      | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        |                        |
| 2:31:4... | notool.exe   | 3392 | CreateFile       | C:\Users\Bismark\Desktop\malware        | SUCCESS        | Desired Access: E...   |
| 2:31:4... | notool.exe   | 3392 | Load Image       | C:\Windows\SysWOW64\kernel32.dll        | SUCCESS        | Image Base: 0x756...   |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\SysWOW64\kernel32.dll        | SUCCESS        | Offset: 425,984, Le... |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\SysWOW64\kernel32.dll        | SUCCESS        | Offset: 602,112, Le... |
| 2:31:4... | notool.exe   | 3392 | Load Image       | C:\Windows\SysWOW64\KernelBase.dll      | SUCCESS        | Image Base: 0x769...   |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\SysWOW64\KernelBase.dll      | SUCCESS        | Offset: 1,959,936, ... |
| 2:31:4... | notool.exe   | 3392 | ReadFile         | C:\Windows\SysWOW64\KernelBase.dll      | SUCCESS        | Offset: 1,982,464, ... |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Length: 528            |
| 2:31:4... | notool.exe   | 3392 | QueryNameInfo... | C:\Windows\SysWOW64\KernelBase.dll      | SUCCESS        | Name: \Windows\...     |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Length: 528            |
| 2:31:4... | notool.exe   | 3392 | QueryNameInfo... | C:\Windows\SysWOW64\KernelBase.dll      | SUCCESS        | Name: \Windows\...     |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Length: 528            |
| 2:31:4... | notool.exe   | 3392 | QueryNameInfo... | C:\Windows\SysWOW64\kernel32.dll        | SUCCESS        | Name: \Windows\...     |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | REPARSE        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | REPARSE        | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | NAME NOT FOUND | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\Software\WOW6432Node\Polic...      | REPARSE        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\SOFTWARE\Policies\Microsoft\...    | SUCCESS        | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegSetInfoKey    | HKLM\SOFTWARE\Policies\Microsoft\...    | SUCCESS        | KeySetInformation...   |
| 2:31:4... | notool.exe   | 3392 | RegQueryValue    | HKLM\SOFTWARE\Policies\Microsoft\...    | NAME NOT FOUND | Length: 80             |
| 2:31:4... | notool.exe   | 3392 | RegCloseKey      | HKLM\SOFTWARE\Policies\Microsoft\...    | SUCCESS        |                        |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKCU\Software\Policies\Microsoft\Win... | NAME NOT FOUND | Desired Access: Q...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | REPARSE        | Desired Access: R...   |
| 2:31:4... | notool.exe   | 3392 | RegOpenKey       | HKLM\System\CurrentControlSet\Contr...  | SUCCESS        | Desired Access: R...   |

Showing 107,796 of 270,666 events (39%)      Backed by virtual memory

## 7 Discussion

- How do we determine the integrity of evidence during malware forensic investigation?

In this research, the integrity of the evidence was determined by comparing the hashes obtained from the disk image in the FTK imager to the hash generated through the Autopsy tool.

**Figure 7** Comparing hashes (see online version for colours)

- Where or which areas should investigators look to during malware forensic investigation?

Knowing the areas to look to during malware forensic investigation is an important arsenal for solving the case. In the case of hard disk forensics, it is important to look in areas like the: shell bags, AppData, run programs, extensions, web search, web history, and downloads. In the case of memory forensics using the Volatility tool, the command lines like windows.pslist, windows.pstree, windows.malfind could help locate the malicious file.

- Which tools can be employed and utilised in forensic malware investigation?

This investigation used tools like the Volatility command line tool for memory forensics analysis and the Autopsy tool for hard disk forensics analysis. These tools proved to be very efficient in malware forensic investigation. FTK imager was used to capture the disk image for hard disk forensics and the memory for memory forensics analysis.

- How does malware behave in host OS and virtual systems?

This research observed that modern malware behaves differently in physical host OS and virtual environments. This is to help them evade detection and analysis by malware analysts. The malware used for this research was observed to evade detection during memory analysis in volatility using Windows.malfind command.

- How do investigators determine the functionality of malware?

To determine the malware's functionality in this research, the detected malware was extracted, and malware analysis was performed to understand the malware better. This research performed static and dynamic malware analysis on the malware.

## 8 Research gap and contribution

The research addresses a significant gap in the malware forensics field by comprehensively combining memory and hard disk techniques to detect, investigate, and

analyse malicious software. The study's most important contribution is the development of a systematic framework that combines empirical and analytical methods to provide a complete picture of malware's activity. The research provides insights into the relationship between malware and the host system and the operation of the infection by conducting in-depth analysis with tools like Volatility and Autopsy. The study also underlines the need of employing realistic virtual environments to successfully tackle modern malware's sophisticated evasion strategies. This study broadens our understanding by providing a methodical method for specialists to detect and analyse malware, improving the efficiency of digital forensics in the face of an ever-changing cyber threat.

## 9 Conclusions

This study provides a novel and comprehensive malware forensics methodology that combines memory and hard disk examination methods. By successfully identifying and examining malicious software inside of a controlled environment, the research provides essential insights into understanding the behaviour and functionality of such software. The suggested methodology provides digital forensics practitioners with a methodical strategy to detect and evaluate malware activity, thereby augmenting the field's ability to tackle emerging cyber threats.

## 10 Future works

While the findings of this study are helpful, further investigation into the use of machine learning and artificial intelligence to improve malware detection and attribution is an essential area of study. Using deep learning and behavioural analysis, two examples of cutting-edge algorithms could facilitate the development of predictive models that can spot new malware trends and attribute attacks to specific threat actors. Future work can better address the ever-changing spectrum of cyber threats by including these technologies in the suggested malware identification and response framework.

## References

- Agrawal, A.K., Sah, S. and Khatri, P. (2020) 'Forensic analysis of a ransomware', *International Journal of Innovative Technology and Exploring Engineering*, Vol. 9, No. 3, pp.3618–3622.
- Ahirao, P. (2021) 'Proactive technique for securing smart cities against malware attacks using static and dynamic analysis', *International Research Journal of Innovations in Engineering and Technology*, Vol. 5, No. 2, p.10.
- Chayal, N.M., Saxena, A. and Khan, R. (2022) 'A review on spreading and forensics analysis of windows-based ransomware', *Annals of Data Science*, Vol. 3, pp.1–22.
- Daghmechi Firoozjaei, M., Habibi Lashkari, A. and Ghorbani, A.A. (2022) 'Memory forensics tools: a comparative analysis', *Journal of Cyber Security Technology*, Vol. 6, No. 3, pp.149–173.
- Dija, S., Ajana, J., Indu, V. and Sabarinath, M. (2020) 'Cyber forensics: discovering traces of malware on windows systems', in *2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, IEEE, December, pp.141–146.

- Ijaz, M., Durad, M.H. and Ismail, M. (2019) 'Static and dynamic malware analysis using machine learning', in *2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, IEEE, January, pp.687–691.
- Kara, I. (2022) 'Fileless malware threats: recent advances, analysis approach through memory forensics and research challenges', *Expert Systems with Applications*, Vol. 214, p.119133.
- Mandela, N., Mahmoud, A.A.S. and Agrawal, A.K. (2023) 'A forensic analysis of the tor network in tails operating system', in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, IEEE, March, pp.546–551.
- Meshram, B.B. and Patil, D.N. (2018) 'Digital forensic analysis of hard disk for evidence collection', *International Journal of Cyber-Security and Digital Forensics*, Vol. 7, No. 2, pp.100–111.
- Pawar, S. and Palivela, H. (2022) 'LCCI: a framework for least cybersecurity controls to be implemented for small and medium enterprises (SMEs)', *International Journal of Information Management Data Insights*, Vol. 2, No. 1, p.100080.
- Rodrigo, C., Pierre, S., Beaubrun, R. and El Khoury, F. (2021) 'BrainShield: a hybrid machine learning-based malware detection model for Android devices', *Electronics*, Vol. 10, No. 23, p.2948.
- Sachdeva, S., Raina, B.L. and Sharma, A. (2020) 'Analysis of digital forensic tools', *Journal of Computational and Theoretical Nanoscience*, Vol. 17, No. 6, pp.2459–2467.
- Sweeney, L. (2002) 'Achieving k-anonymity privacy protection using generalization and suppression', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, pp.571–588.
- Tekiner, E., Acar, A., Uluagac, A.S., Kirda, E. and Selcuk, A.A. (2021) 'SoK: cryptojacking malware', in *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, IEEE, September, pp.120–139.
- Tsado, L. and Kim, J.S. (2022) 'Assessing the practical cybersecurity skills gained through criminal justice academic programs to benefit security operations centers (SOCs)', *Journal of Cybersecurity Education, Research and Practice*, Vol. 2022, No. 1, p.2.