CENTRE FOR DEVELOPMENT OF ADVANCED COMPUTING

(C-DAC), THIRUVANANTHAPURAM, KERALA

A PROJECT REPORT

ON

"Windows Registry Analysis: Investigating User Activity & System Evidence"

SUBMITTED TOWARDS THE

PG-DCSF August 2025

By

Group Number – 05

Avinash	**PRN: 250860940005**
Ayush Kumar	**PRN: 250860940006**
Govind Vyas	**PRN: 250860940013**
Hinge Shreeniket Ashok	**PRN: 250860940015**
Sunil Kumar	**PRN: 250860940039**

Under The Guidance Of

Mr. Jayaram P.

Centre Co- Ordinator & Project Guide

Table of Contents

1. Abstract

This project presents a structured forensic investigation focused on analyzing Windows Registry artifacts to reconstruct user and system activity. The objective was to demonstrate how persistent registry evidence can be used to identify behavioral patterns, document access, executed programs, authentication events, and external device interaction. Rather than relying on volatile memory or network data, the study emphasizes registry-based artifacts as a reliable historical record of system usage.

The investigation examined multiple categories of registry evidence, including executed application traces, user command history, startup persistence entries, file and folder access records, USB device connection history, system configuration metadata, and user account activity. Each artifact was interpreted individually and then correlated with related evidence to establish context. This artifact-driven approach allowed the project to move beyond isolated findings and build a coherent narrative of system behavior.

By correlating timestamps across independent registry sources, a defensible timeline of events was reconstructed. The resulting timeline demonstrated intentional user interaction, document handling, authentication behavior, and removable media usage within a defined period. The project highlights how registry artifacts can support attribution and behavioral reconstruction even when other forensic sources are unavailable.

Overall, the study confirms the investigative value of registry forensics in digital investigations. It provides a practical framework for documenting artifact analysis, correlating evidence, and presenting findings in a professional forensic report format. The project serves as a foundation for more advanced multi-source investigations that combine registry analysis with memory, disk, and network forensics.

2. Introduction

Digital forensic investigations rely on the ability to reconstruct past system behavior from persistent evidence. In Windows environments, one of the most valuable sources of such evidence is the Windows Registry. The registry acts as a centralized hierarchical database that stores configuration settings, user preferences, system metadata, and traces of software interaction. Because it is continuously updated as the operating system and users interact with the system, it becomes a long-term historical record of activity. From a forensic perspective, the registry is not merely a configuration store — it is an evidentiary timeline embedded within the operating system.

The importance of the Windows Registry in digital forensics lies in its persistence and breadth. Many user actions leave behind registry artifacts even when files are deleted or logs are cleared. Program execution traces, device connections, startup behavior, account activity, and recently accessed files are all reflected in specific registry keys. These artifacts provide investigators with insight into intent, behavior, and sequence of events. Unlike volatile memory, registry data survives system reboots, making it especially valuable when investigating historical activity. Proper interpretation of registry artifacts allows investigators to move beyond speculation and rely on system-generated evidence.

Registry artifacts are particularly powerful for investigating user activity because they capture interaction at multiple layers. They record not only what applications were installed, but what programs were executed, what files were accessed, and what commands were issued by a user. This distinction is critical in forensic analysis: installation does not imply usage, but execution artifacts strongly indicate intentional action. Similarly, USB device history can reveal external storage interaction, and startup entries can indicate persistence mechanisms. When correlated, these artifacts form a behavioral profile that helps investigators understand how a system was used.

In real-world investigations, registry analysis plays a key role in identifying insider threats, policy violations, and suspicious behavior. For example, registry evidence can demonstrate whether confidential documents were accessed, whether unauthorized software was executed, or whether removable media was used in proximity to sensitive activity. Organizations frequently rely on such evidence to enforce acceptable-use policies, investigate data leakage, or validate compliance. Registry artifacts are also valuable in incident response scenarios, where rapid reconstruction of user actions can determine the scope and impact of an event.Another reason registry forensics is essential is its role in timeline reconstruction. Individual artifacts provide isolated facts, but when timestamps are correlated across registry keys, they reveal patterns and sequences..This project focuses specifically on registry-based investigation to demonstrate how meaningful conclusions can be drawn from persistent artifacts alone. By isolating the registry as the primary evidence source, the project emphasizes methodological discipline: careful extraction, artifact interpretation, and cross-correlation. The goal is not to replace full-spectrum digital forensics, but to show that registry analysis itself is a powerful investigative technique. Through structured documentation and timeline reconstruction, this project illustrates how registry artifacts can provide a defensible foundation for understanding user and system activity.

3. Problem Statement

Suspicious user activity was suspected on a Windows-based system, raising concerns about potential unauthorized actions and improper handling of data. There was a requirement to examine persistent system evidence to determine what activities had occurred and whether user behavior aligned with expected system usage.

The investigation focused on analyzing Windows Registry artifacts as a primary source of evidence. Registry data was selected because it provides reliable historical traces of executed programs, file access, authentication behavior, device connections, and system configuration changes. The challenge was to interpret these artifacts in a structured manner and correlate them to reveal meaningful patterns.

The core problem addressed in this project was the need to reconstruct a defensible timeline of user and system events using registry-derived evidence alone. By systematically analyzing and correlating artifacts, the project aimed to transform isolated registry entries into a coherent narrative of activity.

4. Objectives of the Project

- To analyze Windows Registry artifacts that reflect user interaction and system activity.
- To extract evidence related to **USB device usage, executed programs, file and folder access, and startup persistence mechanisms.**
- To **correlate registry timestamps** in order to reconstruct a chronological sequence of events.
- To **interpret registry artifacts in a structured forensic framework** rather than as isolated entries.
- To how registry evidence can support attribution and behavioral analysis.
- To understand the investigative value and limitations of registry-based digital forensics.
- To document findings in a professional forensic report format suitable for real-world DFIR workflows.

5. Scope of the Project

This project is limited to the forensic analysis of **Windows Registry–based artifacts**. The investigation focuses exclusively on persistent registry evidence that reflects user interaction and system behavior. No live system acquisition, memory capture, or active incident response actions were performed. The objective is artifact interpretation rather than real-time monitoring.

The scope includes both **user-level** and **system-level** registry evidence. User-level artifacts were analyzed to understand executed programs, file access patterns, command history, and user account activity. System-level artifacts were examined to establish environment configuration, startup behavior, device connections, and operating system metadata. Together, these layers provide contextual understanding of how the system was used.

This project does **not** include network traffic analysis, volatile memory forensics, disk carving, or malware reverse engineering. These areas are outside the defined boundary to maintain a focused methodology centered on registry forensics. The constrained scope ensures that conclusions are drawn strictly from registry artifacts and prevents overextension beyond the available evidence.

By clearly limiting the investigation to registry-based analysis, the project maintains methodological credibility while demonstrating the standalone value of registry artifacts in reconstructing user and system activity.

---

6. Background / Theoretical Overview

6.1 Windows Registry Overview

The Windows Registry is a hierarchical database used by the Microsoft Windows operating system to store configuration settings and operational metadata. It contains information about hardware configuration, installed software, user preferences, system policies, and application behavior. Unlike simple configuration files, the registry functions as a structured database where keys and values represent organized system knowledge. Every interaction with the operating system — from launching an application to connecting a device — may generate or update registry entries. This continuous updating makes the registry a long-term record of system activity.

At a structural level, the registry is divided into logical containers known as **hives**. Each hive corresponds to a physical file stored on disk and represents a category of system or user information. The most important hives in forensic investigations include:

- **SYSTEM** – Stores hardware configuration, control sets, device information, and system-level settings
- **SOFTWARE** – Contains installed application data, OS version details, and system-wide software configuration
- **SAM** – Holds user account and authentication metadata
- **SECURITY** – Stores security policies and permission-related data
- **NTUSER.DAT** – Contains per-user configuration, user activity traces, and interaction history
- **UsrClass.dat** – Records shell-level user interaction such as file browsing and Explorer behavior

These hives are loaded into memory during system operation but persist on disk even after shutdown. Because of this persistence, registry artifacts survive reboots and remain available for post-incident examination.

A key conceptual distinction in registry analysis is the difference between **user hives** and **system hives**. System hives (SYSTEM, SOFTWARE, SAM, SECURITY) describe the environment of the machine itself — hardware, OS configuration, and account structure. User hives (NTUSER.DAT and UsrClass.dat) describe how a specific user interacted with the system. This separation allows investigators to attribute actions to identities rather than treating the system as a single anonymous entity.

Another important concept is the presence of **control sets** within the SYSTEM hive. Windows maintains multiple control sets to preserve configuration states. The active configuration is referenced through CurrentControlSet, which points to the control set used during the last successful boot. Understanding control sets ensures accurate interpretation of system configuration artifacts.

The registry's hierarchical design, timestamped keys, and persistent storage make it uniquely suited for forensic reconstruction. It acts as a bridge between configuration and behavior, allowing

investigators to see not only what the system is configured to do, but what has actually happened over time.

6.2 Registry in Digital Forensics

In digital forensics, the Windows Registry is considered a high-value artifact source because it preserves traces of activity that are difficult to eliminate completely. Many user actions automatically generate registry updates that remain long after files are deleted or applications are uninstalled. This persistence makes registry artifacts especially useful in investigations where intentional cleanup or anti-forensic behavior may have occurred.

Registry artifacts are primarily **non-volatile**, meaning they remain stored on disk after power loss or shutdown. This distinguishes them from volatile artifacts such as memory contents, running processes, or live network sessions. Volatile evidence provides insight into current system state, but non-volatile evidence provides historical continuity. Registry analysis therefore complements memory forensics by supplying a durable activity record.

The registry persists evidence in several ways:

- Execution traces are recorded when applications are launched
- File access artifacts appear when documents are opened or saved
- Device connection history is stored when USB media is inserted
- Authentication metadata updates when users log in
- Startup entries record persistence behavior

These artifacts are generated automatically by the operating system, not by investigative tools. Their system-generated nature strengthens their evidentiary reliability.

One of the most powerful uses of registry artifacts is **timeline reconstruction**. Each registry key contains a **Last Written timestamp** that reflects when the key was modified. By correlating timestamps across multiple keys, investigators can determine sequences of actions and relationships between events. This temporal correlation transforms isolated artifacts into a narrative of behavior.

Timeline reconstruction using registry data allows investigators to answer critical forensic questions:

- When did a user access specific files?
- What programs were executed before or after that access?
- Was removable media connected during the same period?
- Did authentication changes occur near suspicious activity?

Because registry timestamps are generated by the system clock, they provide consistent chronological anchors. When normalized to the correct timezone, they form a defensible timeline that supports investigative reasoning.

Registry forensics also plays a major role in attribution. By linking artifacts from user hives with account metadata from system hives, investigators can associate actions with specific identities. This identity linkage is essential in cases involving insider threats, policy violations, or unauthorized data handling.

From a theoretical perspective, the registry represents a hybrid evidence layer: it captures both configuration intent and behavioral residue. This dual role makes it uniquely valuable. While it cannot replace memory or disk forensics, it often serves as the backbone of activity reconstruction when other evidence sources are incomplete.

In summary, the Windows Registry is not just a configuration database — it is a persistent behavioral archive. Its structure, timestamps, and automatic update mechanisms make it one of the most reliable sources of forensic evidence in Windows investigations. Understanding its theoretical foundations is essential for interpreting artifacts accurately and constructing defensible investigative conclusions.

7. Tools and Environment

7.1 Operating Environment

The forensic analysis was performed in a controlled laboratory environment using a Windows-based workstation configured specifically for offline evidence examination. The purpose of using a controlled environment was to ensure that registry hive files could be analyzed without altering original evidence. All registry artifacts were examined in a read-only context, following standard forensic handling practices to preserve evidentiary integrity.

The analysis environment simulated a post-acquisition forensic workstation rather than a live response system. No changes were made to the original registry data during examination. This approach aligns with accepted digital forensic principles, where evidence is analyzed in isolation to prevent contamination and maintain repeatability.

7.2 Forensic Tools Used

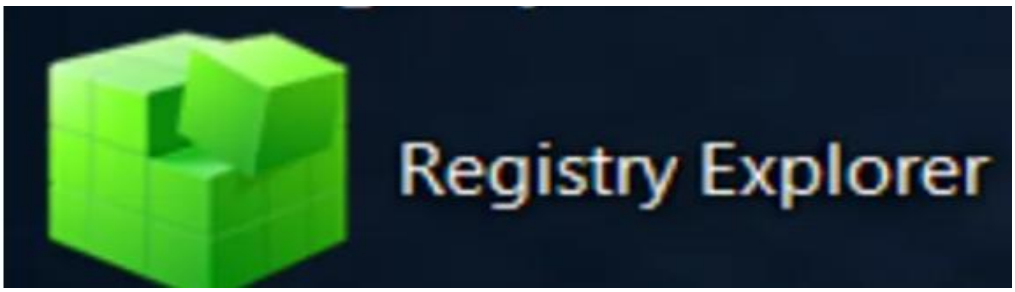
The project relied on specialized registry forensic tools designed for structured artifact parsing and visualization:

CyberCheck (Fran Edition)

A forensic artifact analysis tool used to extract and summarize registry-based evidence. It provides categorized views of user activity, startup entries, USB history, and system metadata. The tool was used to generate artifact summaries and cross-validate registry findings.

Registry Explorer (Eric Zimmerman Tools)

A registry hive viewer used to manually inspect keys, decode values, and analyze timestamp metadata. It allows investigators to navigate registry hierarchies at a granular level and verify



artifact interpretation. This tool was primarily used for detailed examination and validation of extracted evidence.

Both tools were used in combination to ensure consistency. Automated artifact summaries were verified through manual registry inspection to reduce interpretation errors.

Exterro FTK Imager

Exterro FTK Imager is a forensic acquisition tool used to collect and preview digital evidence without modifying the original data. It enables investigators to extract registry hive files and other artifacts in a read-only manner while generating cryptographic hash values to verify evidence integrity.

The tool follows the principle of forensic soundness by ensuring that acquired evidence remains unchanged and verifiable. In this project, FTK Imager was used to obtain registry hives safely before analysis, supporting proper evidence preservation and maintaining credibility of findings.

7.3 Reason for Tool Selection

The selected tools were chosen based on reliability, transparency of artifact parsing, and compatibility with Windows registry hives. The goal was to use tools that support both automated extraction and manual validation, rather than relying on black-box outputs.

CyberCheck was selected for its ability to present registry artifacts in structured investigative categories, which supports rapid identification of relevant evidence. **Registry Explorer** was chosen for its precision and low-level access to registry data, allowing investigators to confirm timestamps and values directly from hive structures.

Using complementary tools ensured cross-verification of findings, improved analytical accuracy, and strengthened the credibility of conclusions. The emphasis remained on artifact interpretation rather than tool dependency, reinforcing the methodological focus of the project.

8. Methodology

This investigation followed a structured digital forensic workflow focused on registry-based artifact analysis. The methodology emphasized evidence preservation, systematic artifact extraction, and cross-correlation of findings. Each phase was conducted in a controlled environment to maintain analytical integrity and reproducibility.

8.1 Evidence Identification

The first step involved identifying relevant evidence sources within the Windows operating system. The Windows Registry was selected as the primary investigative target because it stores persistent records of user interaction and system behavior. Key registry hives were identified based on their forensic relevance, including SYSTEM, SOFTWARE, SAM, NTUSER.DAT, and UsrClass.dat. These hives collectively contain artifacts related to user activity, system configuration, authentication behavior, and device interaction.

The objective of this stage was to define the evidence scope and determine which artifact categories could contribute to activity reconstruction.

8.2 Registry Hive Acquisition

Registry hive files were obtained in a manner that preserved their original structure and timestamps. Acquisition focused on offline registry data rather than live system interrogation. Working with offline hives ensured that no additional writes occurred during analysis, preventing contamination of evidence.

The acquired hive files were stored in a controlled forensic workspace. Integrity preservation principles were followed, meaning the original evidence remained unchanged while copies were used for analysis. This approach mirrors standard forensic practice where original artifacts are protected and all examination occurs on working copies.

8.3 Artifact Extraction Using Forensic Tools

Once registry hives were acquired, they were loaded into forensic analysis tools capable of parsing registry structures. Relevant artifact categories were extracted, including:

- Executed programs
- USB device history
- File and folder access traces
- Startup persistence entries
- User account and authentication data
- System configuration metadata

Automated artifact extraction provided structured summaries, while manual registry navigation was used to verify values and timestamps. This dual approach reduced the risk of interpretation errors and ensured that extracted artifacts accurately reflected the underlying registry data.

8.4 Analysis and Interpretation

Extracted artifacts were analyzed individually to determine their forensic significance. Each artifact was interpreted in context, focusing on what user or system behavior it represented. Analysis emphasized distinguishing between passive system behavior and intentional user actions.

Artifacts were documented using a standardized template that recorded registry paths, timestamps, observed values, and forensic interpretation. This structured documentation ensured consistency across artifact categories and supported later correlation.

The goal of this phase was not merely to list artifacts, but to translate technical registry entries into meaningful behavioral evidence.

8.5 Timeline Correlation

The final methodological step involved correlating timestamps from multiple artifacts to reconstruct a chronological narrative. Registry Last Written timestamps were normalized and compared across artifact categories. Events were ordered sequentially to identify patterns, clusters of activity, and relationships between actions.

Timeline reconstruction transformed isolated registry entries into a cohesive sequence of user and system events. This step provided the foundation for investigative conclusions by linking authentication activity, file access, executed programs, and device usage into a unified behavioral timeline.

This methodology demonstrates a disciplined forensic workflow: controlled acquisition, structured extraction, contextual interpretation, and chronological correlation. By following a repeatable process, the investigation ensured that conclusions were based on verifiable registry evidence rather than assumption.

9. Registry Artifact Analysis (CORE SECTION)

9.1 USB Device History (USBSTOR Entries)

Registry Location:

The USB device connection history was obtained from the following Windows Registry locations:

- `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USBSTOR`
- `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Enum\USB`

These registry paths store detailed information about USB mass storage devices that have been connected to the system, including device identifiers, serial numbers, and timestamps.

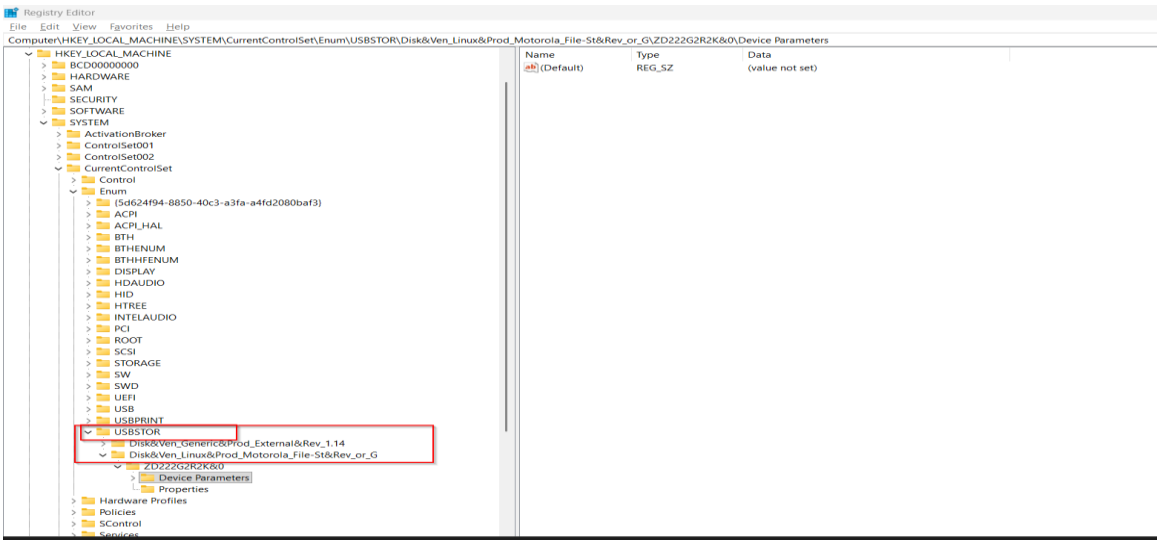
Description of the Artifact:

The **USB Device History artifact** represents records of **external USB storage devices** that have been connected to the Windows system. When a USB mass storage device such as a pen drive, external hard disk, or mobile storage device is connected, Windows creates persistent registry entries under the USBSTOR key.

This artifact records **system-level activity** related to:

- Detection and enumeration of USB storage devices
- Identification of device type, manufacturer, and serial number
- Timestamp indicating when the device was last connected or accessed

USB device artifacts are crucial in digital forensic investigations as they help determine whether **external storage media** was used on the system, which may indicate **data transfer, data exfiltration, or unauthorized access**.



[Logical representation of USB device enumeration process]

Extraction Method

The USB device history was extracted using **multiple forensic tools** to ensure accuracy and validation of findings.

Tools Used:

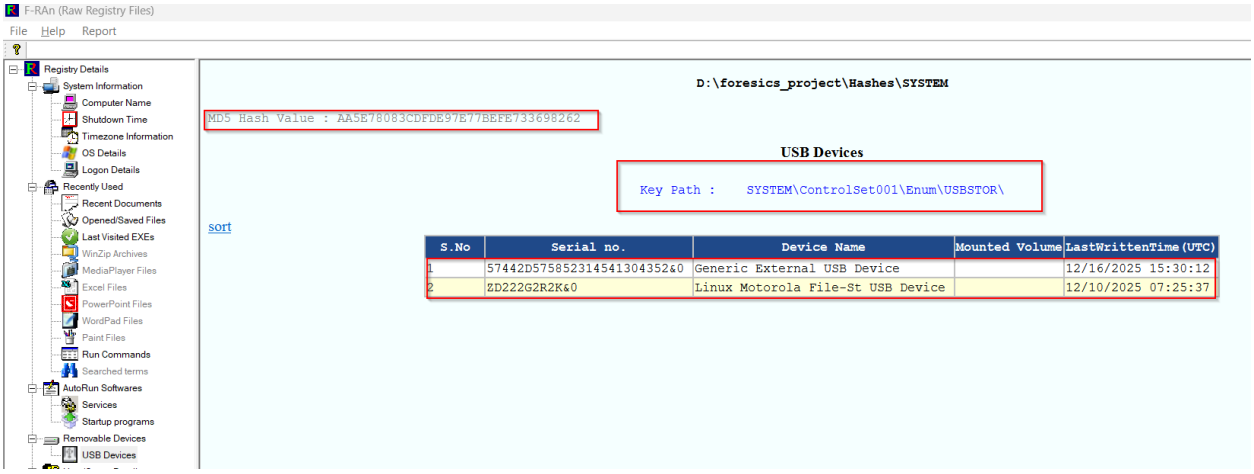
- CyberCheck (Fran Edition)
- **Registry Explorer**

Registry Hive Involved:

- `SYSTEM` hive

Extraction Procedure:

1. The SYSTEM registry hive was loaded into **Registry Explorer**.
2. The path `CurrentControlSet\Enum\USBSTOR` was navigated to identify connected USB devices.
3. Device-specific subkeys were examined to extract:
    - Device names
    - Serial numbers
    - Mounted volume identifiers
    - LastWrittenTime values
4. The same artifact was analyzed using **CyberCheck Fran** to cross-verify device information and timestamps.
5. Screenshots of registry entries and tool outputs were captured for documentation and evidentiary support.

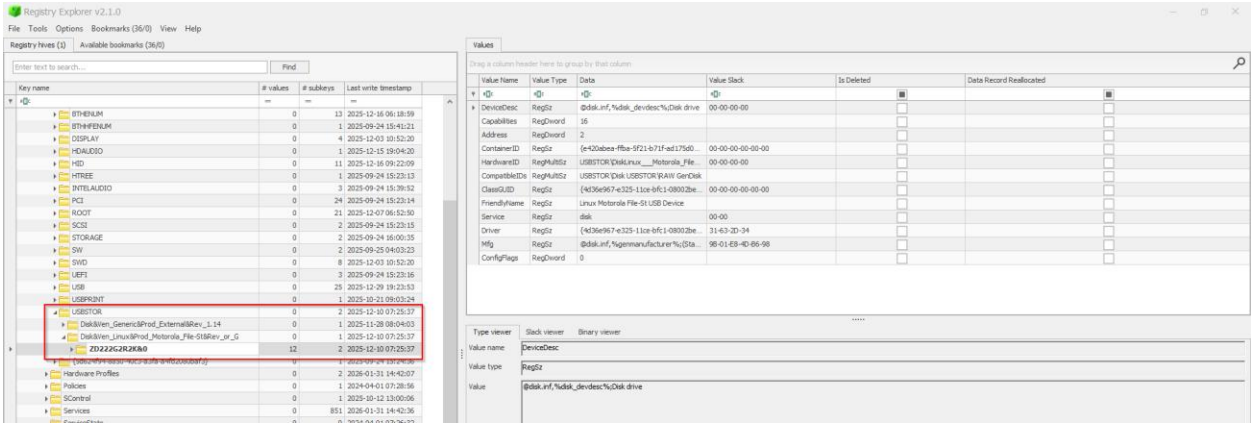


The following USB storage devices were identified as having been connected to the system:

S. No.	Serial Number	Device Name	Mounted Volume	LastWrittenTime (UTC)
1	57442D575852314541304352&0	Generic External USB Device	Present	16-12-2025 15:30:12
2	ZD222G2R2K&0	Linux Motorola File-St USB Device	Present	10-12-2025 07:25:37

Summary of Findings:

- Two distinct USB storage devices were detected.
- Unique serial numbers confirm that **different physical devices** were used.
- The LastWrittenTime values indicate **specific dates and times** when the devices were last connected or accessed.
- Presence of a **mobile device storage interface** suggests possible file transfer between the system and a mobile device.



[Detailed registry entry with serial number and timestamp]

Forensic Interpretation

The presence of USBSTOR registry entries conclusively indicates that **external USB storage devices were connected to the system** during the observed time period. The recorded serial numbers uniquely identify the physical devices, eliminating ambiguity regarding device reuse or duplication.

This evidence is **forensically significant** because:

- It proves **external device usage**, which may involve data transfer operations.
- It establishes a **timeline of device interaction** using reliable system-generated timestamps.
- It supports investigation scenarios involving **data leakage, unauthorized copying, or introduction of malicious files**.

By correlating USB device history with other artifacts such as **file access logs, recent documents, and user activity artifacts**, investigators can reconstruct user behavior and determine whether sensitive data may have been accessed or transferred using removable media.

9.2 Executed Programs (Application Execution Evidence)

Registry Location

The execution evidence for programs was extracted from the following registry hives and paths:

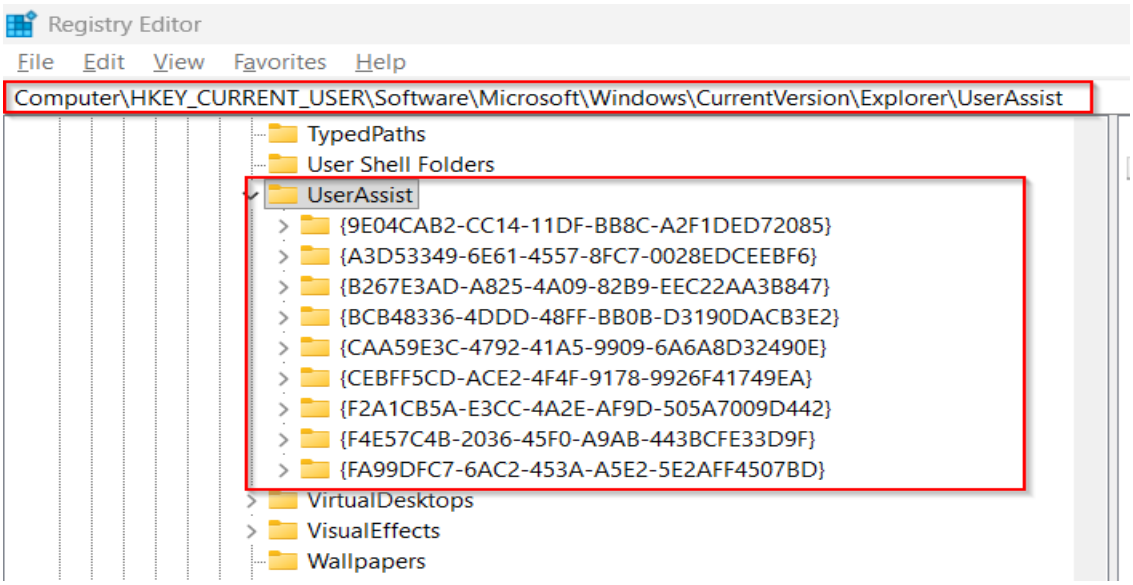
- NTUSER.DAT
- Amcache.hve

Relevant locations examined include:

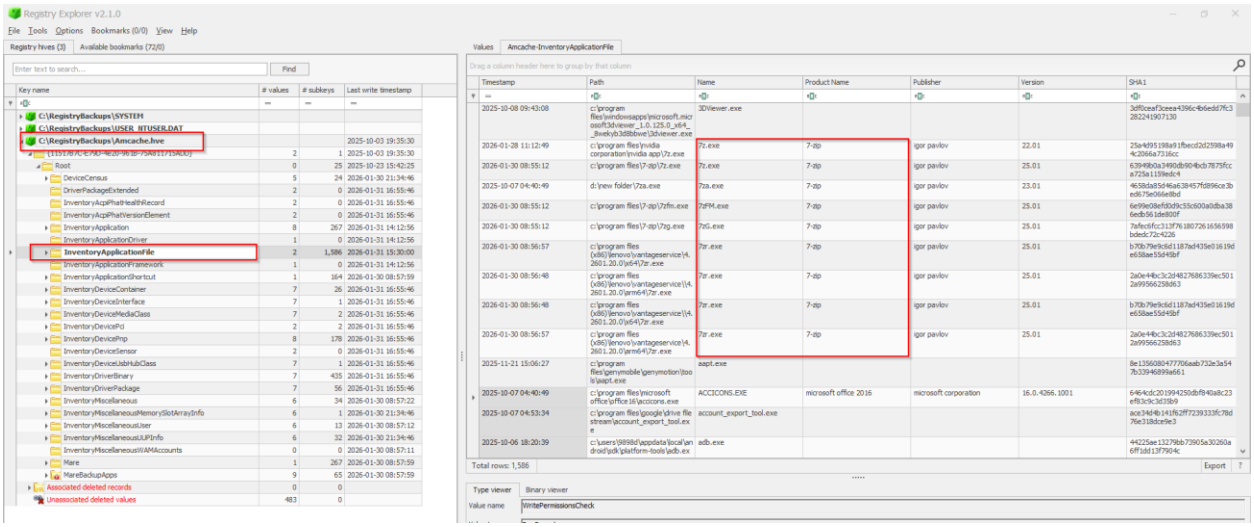
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist`

- `Amcache.hve\Root\File\`

These locations store records related to application execution, usage counts, and executable metadata.



[Registry path showing UserAssist entries]



[Amcache.hve loaded in Registry Explorer]

Description of the Artifact

The **Executed Programs artifact** represents evidence of **applications that were run on the system**, either by direct user interaction or system-assisted execution. Windows records such activity in multiple locations to support usability, compatibility tracking, and application management.

This artifact records:

- Names of executed applications
- Full execution paths of executables
- Indirect execution timestamps (via LastWrittenTime updates)

The presence of an application in these artifacts strongly indicates that the executable was **launched at least once** on the system. Unlike installed software lists, this artifact specifically focuses on **actual execution**, making it highly valuable in forensic investigations.

Extraction Method

Tools Used:

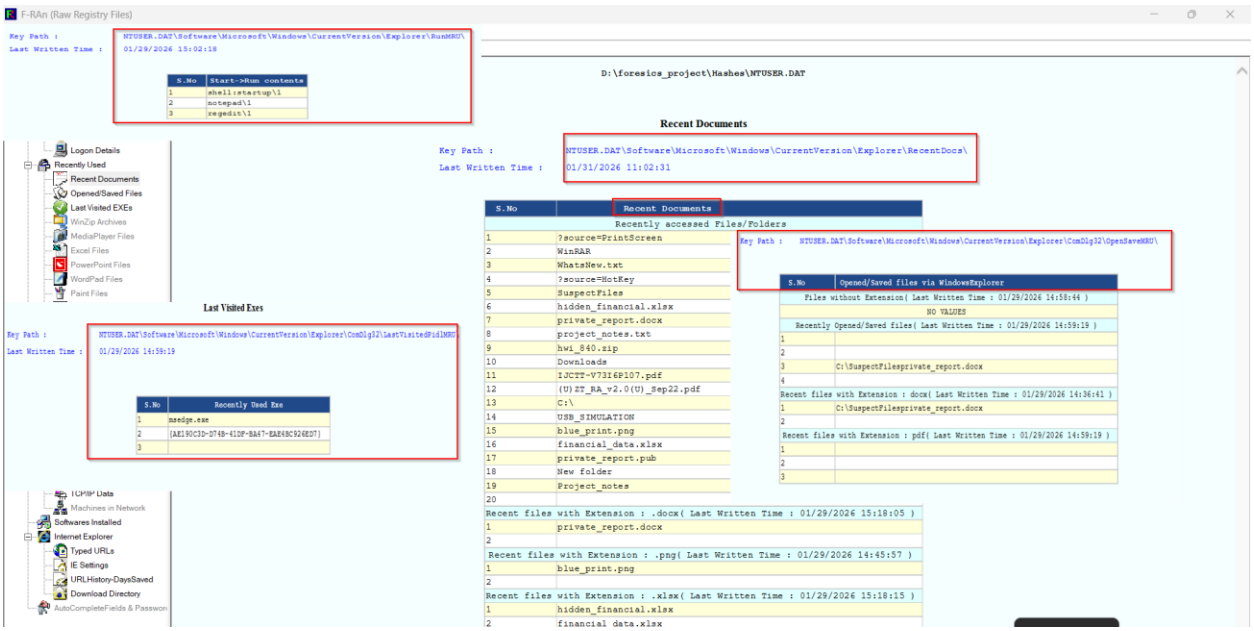
- CyberCheck
- Registry Explorer

Registry Hives Involved:

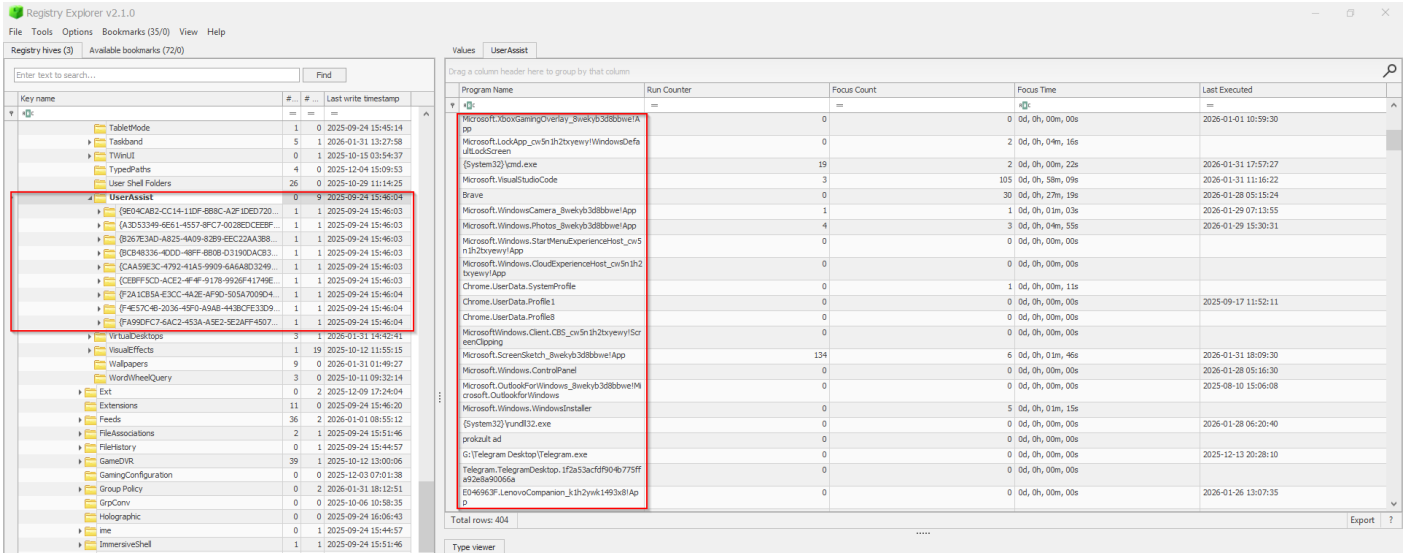
- `NTUSER.DAT`
- `Amcache.hve`

Extraction Procedure:

1. The `NTUSER.DAT` hive was loaded into **Registry Explorer**.
2. The `UserAssist` registry keys were parsed to identify executed applications linked to user activity.
3. The `Amcache.hve` hive was analyzed to extract executable metadata such as application names and execution paths.
4. CyberCheck Fran was used to validate findings and present correlated execution data.
5. Screenshots of registry keys, decoded values, and tool outputs were captured for evidentiary documentation.



[CyberCheck Fran executed programs view]



[Registry Explorer decoded UserAssist values]

Observed Evidence

Analysis of the registry artifacts revealed multiple executed applications, including both system utilities and user-launched programs. Notable examples include:

S. No.	Executed Application	Execution Path
1	Microsoft Word	`\Microsoft Office\Office16\WINWORD.EXE`
2	Microsoft Excel	`\Microsoft Office\Office16\EXCEL.EXE`
3	Command Prompt	`\cmd.exe`
4	Registry Editor	`\regedit.exe`
5	Google Chrome	`Chrome`
6	Microsoft Edge	`MSEdge`
7	7-Zip File Manager	`\7-Zip\7zFM.exe`
8	HWiNFO64	`\HWiNFO64\HWiNFO64.EXE`
9	WinPrefetchView	`D:\Downloads\winprefetchview\WinPrefetchView.exe`
10	Downloaded Executables	`C:\Users\suspect_user\Downloads\hwi64_840.exe`, `7z2501-x64.exe`

Summary of Evidence:

- Both **system tools** and **third-party utilities** were executed.
- Presence of executables launched from the **Downloads directory** indicates deliberate user execution of downloaded files.
- Execution of forensic and system inspection tools suggests active interaction beyond routine system use.

Forensic Interpretation

The analyzed registry artifacts confirm that **multiple programs were actively executed by the user** on the system. Execution paths and application names demonstrate intentional interaction rather than passive system background activity.

This evidence is forensically relevant because:

- It establishes **user presence and interaction** with the system.
- It differentiates between **installed software** and **actually executed applications**.
- Execution of utilities from user-controlled directories (such as *Downloads*) indicates **intentional user actions**, which may be correlated with file access or data handling events.

When correlated with other artifacts such as **Recent Documents, USB device history, and RunMRU entries**, this evidence contributes to reconstructing a **clear activity timeline**, supporting conclusions related to user behavior and possible data access or manipulation.

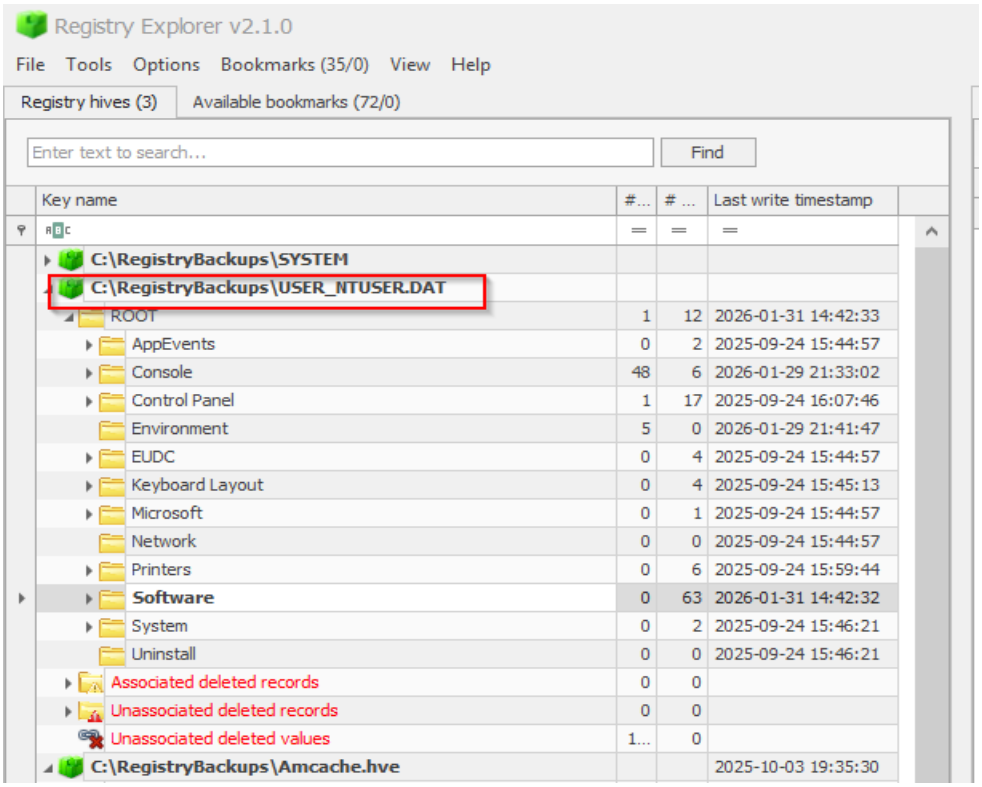
9.3 User Activity – Run Commands and Application Interaction (RunMRU & UserAssist)

Registry Location

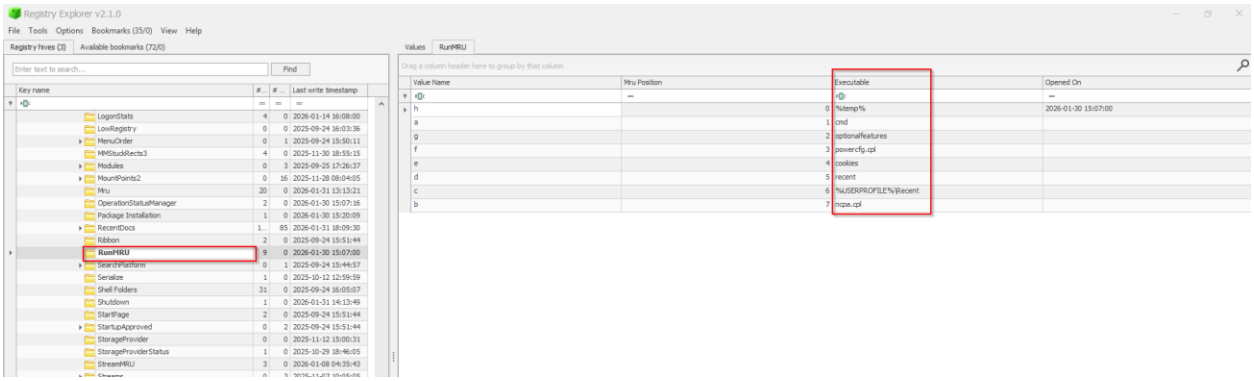
User-initiated activity was extracted from the following registry locations within the **NTUSER.DAT** hive:

- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU`
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count`

These locations record commands executed by the user and applications launched through interactive Windows components.



[NTUSER.DAT loaded in Registry Explorer]



[RunMRU registry path with decoded values]

Description of the Artifact

The **User Activity artifact** represents evidence of **explicit user interaction with the operating system**, rather than background or automated system behavior. It is one of the most critical artifacts for establishing **user intent** in a forensic investigation.

This artifact records:

- Commands manually entered by the user via **Start → Run dialog**
- Applications launched through **Explorer, Start Menu, or shortcuts**
- Frequency and sequence of application usage

The **RunMRU** key captures textual commands executed by the user, while **UserAssist** records applications launched via graphical interfaces. Together, they provide strong proof of **intentional actions performed by the logged-in user**.

Extraction Method

Tools Used:

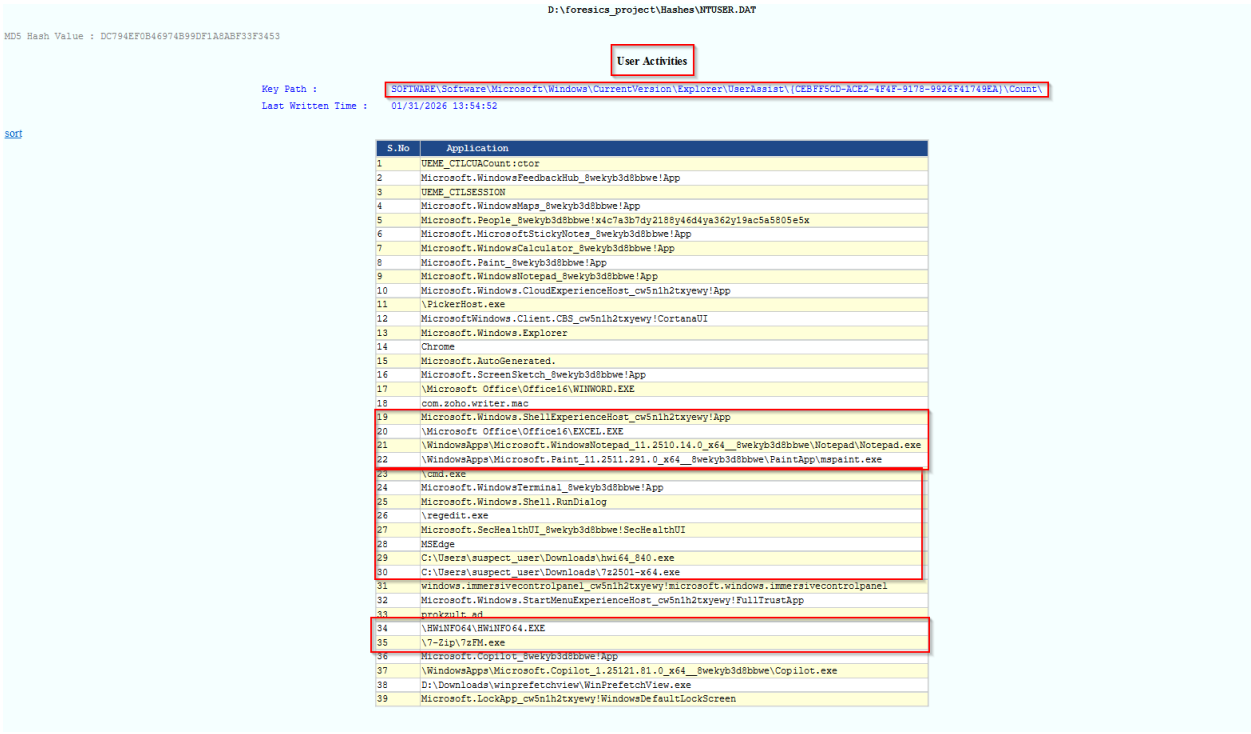
- CyberCheck (Fran Edition)
- Registry Explorer

Registry Hive Involved:

- `NTUSER.DAT`

Extraction Procedure:

1. The `NTUSER.DAT` hive was loaded into **Registry Explorer**.
2. The `RunMRU` key was examined to identify commands typed by the user into the Run dialog.
3. The `UserAssist` key was decoded to reveal executed applications and utilities.
4. CyberCheck Fran was used to parse and present decoded UserAssist entries in a readable format.
5. Screenshots were captured for registry paths, decoded values, and tool outputs.



D:\foresics_project\Hashes\NTUSER.DAT

MD5 Hash Value : DC794EF0B46974B99DF1A8ABF33F3453

User Activities

Key Path : SOFTWARE\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}\Count\

Last Written Time : 01/31/2026 13:54:52

sort

S.No	Application
1	UEME_CTLCUACount:ctor
2	Microsoft.WindowsFeedbackHub_8wekyb3d8bbwe!App
3	UEME_CTLSESSION
4	Microsoft.WindowsMaps_8wekyb3d8bbwe!App
5	Microsoft.People_8wekyb3d8bbwe!x4c7a3b7dy2188y46d4ya362y19ac5a5805e5x
6	Microsoft.MicrosoftStickyNotes_8wekyb3d8bbwe!App
7	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App
8	Microsoft.Paint_8wekyb3d8bbwe!App
9	Microsoft.WindowsNotepad_8wekyb3d8bbwe!App
10	Microsoft.Windows.CloudExperienceHost_cw5n1h2txyewy!App
11	\PickerHost.exe
12	MicrosoftWindows.Client.CBS_cw5n1h2txyewy!CortanaUI
13	Microsoft.Windows.Explorer
14	Chrome
15	Microsoft.AutoGenerated.
16	Microsoft.ScreenSketch_8wekyb3d8bbwe!App
17	\Microsoft Office\Office16\WINWORD.EXE
18	com.zoho.writer.mac
19	Microsoft.Windows.ShellExperienceHost_cw5n1h2txyewy!App
20	\Microsoft Office\Office16\EXCEL.EXE
21	\WindowsApps\Microsoft.WindowsNotepad_11.2510.14.0_x64__8wekyb3d8bbwe\Notepad\Notepad.exe
22	\WindowsApps\Microsoft.Paint_11.2511.291.0_x64__8wekyb3d8bbwe\PaintApp\mspaint.exe
23	\cmd.exe
24	Microsoft.WindowsTerminal_8wekyb3d8bbwe!App
25	Microsoft.Windows.Shell.RunDialog
26	\regedit.exe
27	Microsoft.SecHealthUI_8wekyb3d8bbwe!SecHealthUI
28	MSEdge
29	C:\Users\suspect_user\Downloads\hw164_840.exe
30	C:\Users\suspect_user\Downloads\7z2501-x64.exe
31	windows.immersivecontrolpanel_cw5n1h2txyewy!microsoft.windows.immersivecontrolpanel
32	Microsoft.Windows.StartMenuExperienceHost_cw5n1h2txyewy!FullTrustApp
33	prokzult.ad
34	\HWiNFO64\HWiNFO64.EXE
35	\7-Zip\7zFM.exe
36	Microsoft.Copilot_8wekyb3d8bbwe!App
37	\WindowsApps\Microsoft.Copilot_1.25121.81.0_x64__8wekyb3d8bbwe\Copilot.exe
38	D:\Downloads\winprefetchview\WinPrefetchView.exe
39	Microsoft.LockApp_cw5n1h2txyewy!WindowsDefaultLockScreen

[CyberCheck Fran User Activity module output]

Observed Evidence

A. RunMRU (Start → Run Commands)

S. No.	Command Executed	Interpretation
1	`shell:startup`	User intentionally accessed startup folder
2	`notepad`	User launched Notepad manually
3	`regedit`	User opened Registry Editor

RunMRU LastWrittenTime: `01/29/2026 15:02:18`

This confirms **manual command execution** rather than automated system actions.

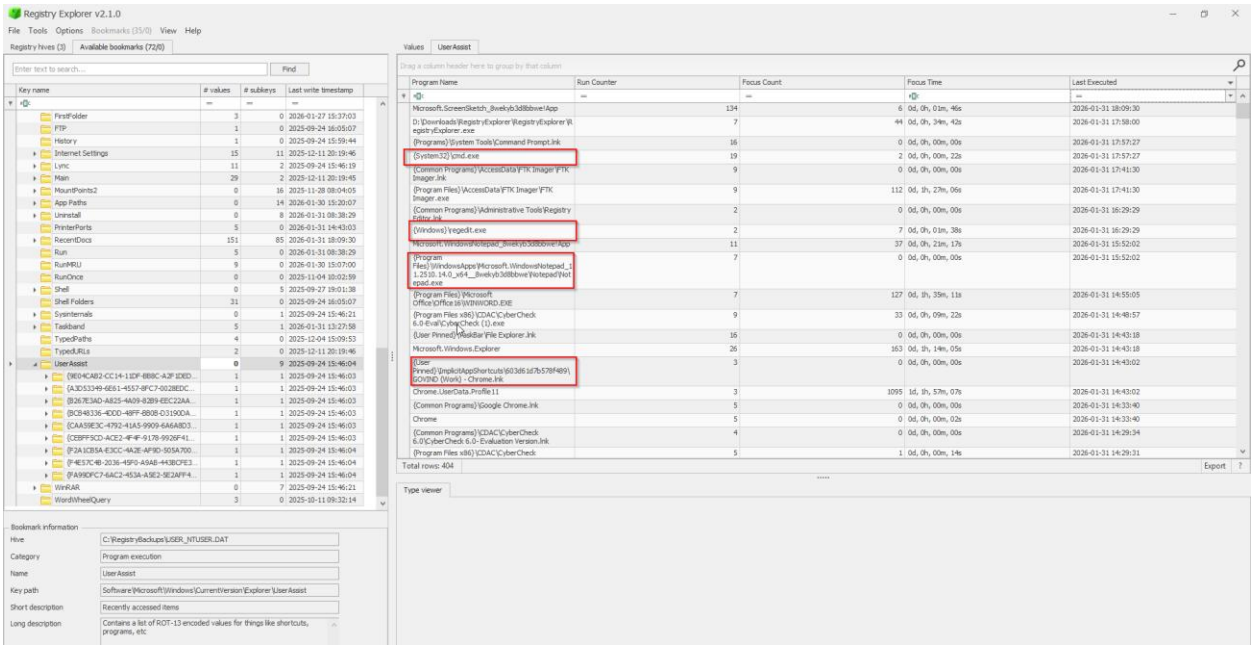
B. UserAssist (Application Interaction)

The following applications and executables were identified as launched through user interaction:

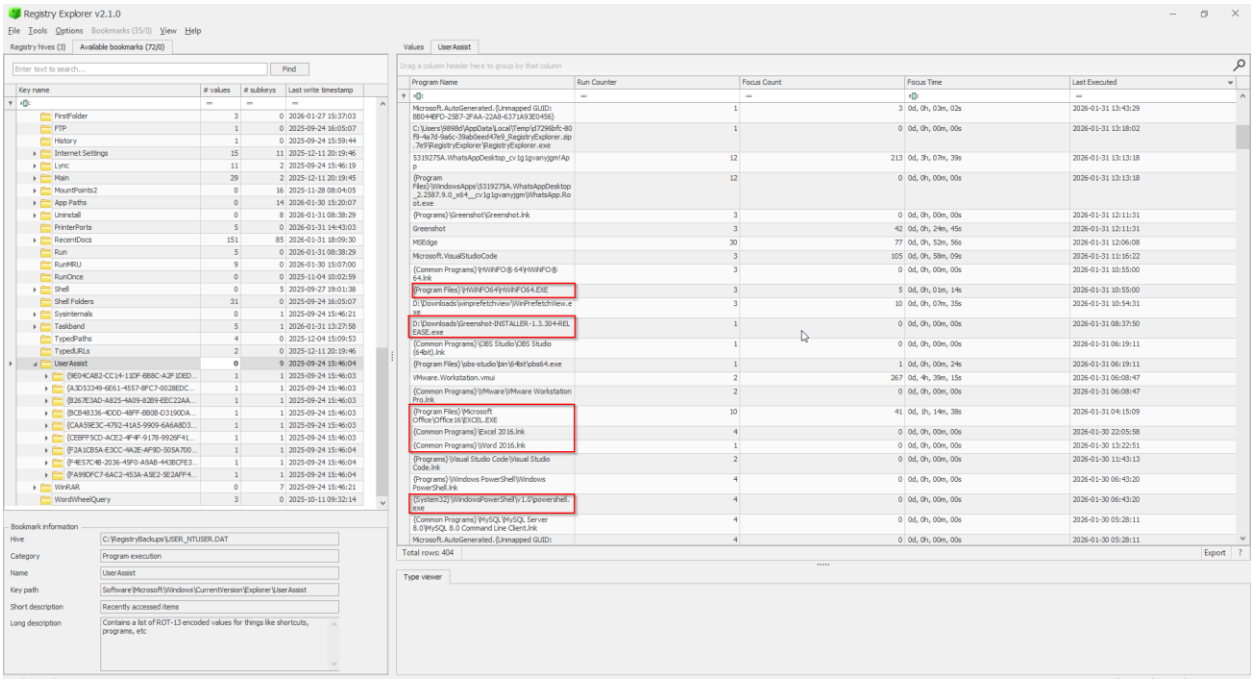
S. No.	Application / Executable
1	Microsoft Windows Explorer
2	Google Chrome
3	Microsoft Edge
4	Command Prompt (`cmd.exe`)
5	Registry Editor (`regedit.exe`)
6	Microsoft Word (`WINWORD.EXE`)
7	Microsoft Excel (`EXCEL.EXE`)
8	7-Zip File Manager
9	HWiNFO64
10	Microsoft Paint
11	Microsoft Notepad
12	Microsoft Windows Terminal
13	Downloaded executables from `C:\Users\suspect_user\Downloads\`
14	WinPrefetchView
15	Microsoft Copilot

UserAssist LastWrittenTime: `01/31/2026 13:54:52`

This demonstrates **sustained and diverse user interaction** with system tools, productivity software, and downloaded utilities.



[Decoded UserAssist entries highlighting executables]



[Evidence of executables launched from Downloads directory]

Forensic Interpretation

The RunMRU and UserAssist artifacts conclusively establish **active, intentional user behavior** on the system. Commands such as `regedit` and `shell:startup` indicate that the user was not merely performing routine tasks but was **exploring system configuration and startup behavior**.

This evidence is highly relevant because:

- It directly proves **user intent**, not just system activity
- It confirms **manual execution of administrative and system-level tools**
- It supports correlation with other artifacts such as **startup persistence, file access, and executed programs**

When combined with executed program artifacts and startup entries, this data strengthens the timeline reconstruction by showing **what the user chose to do**, **how**, and **when**.

9.4 Startup / Persistence Mechanisms

Registry Location

Startup and persistence evidence was extracted from the following registry locations across multiple hives:

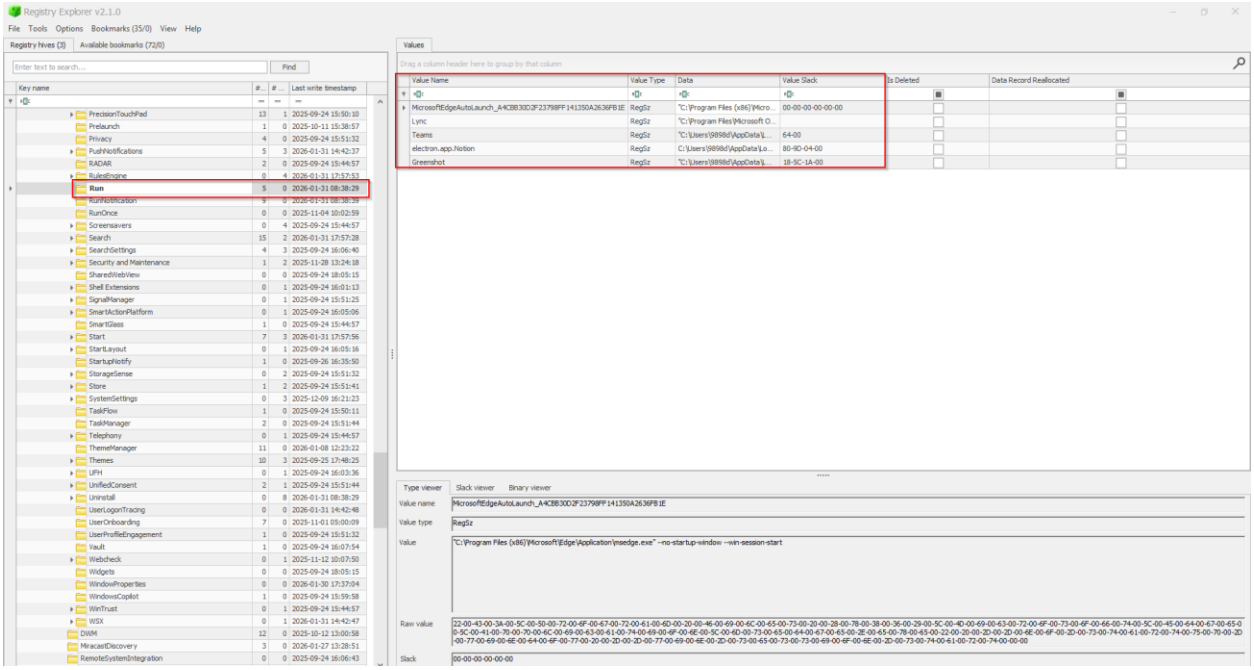
From NTUSER.DAT (User-level persistence):

- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run`
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\RunOnce`

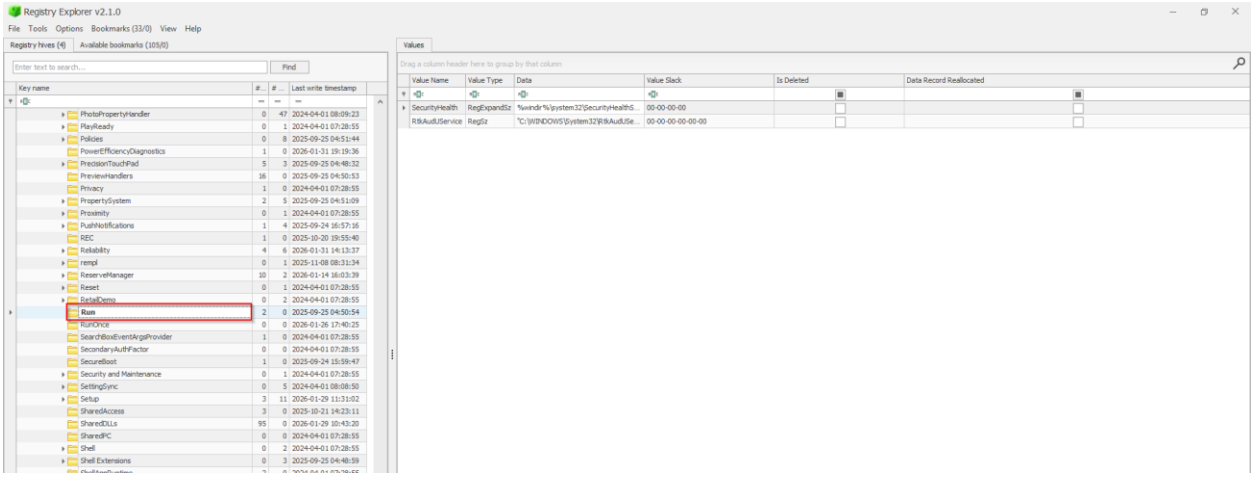
From SOFTWARE hive (System-wide persistence):

- `SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
- `SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce`

These registry keys define applications configured to **automatically execute during system startup or user logon**.



[NTUSER.DAT Run key loaded in Registry Explorer]



[SOFTWARE hive Run key showing startup programs]

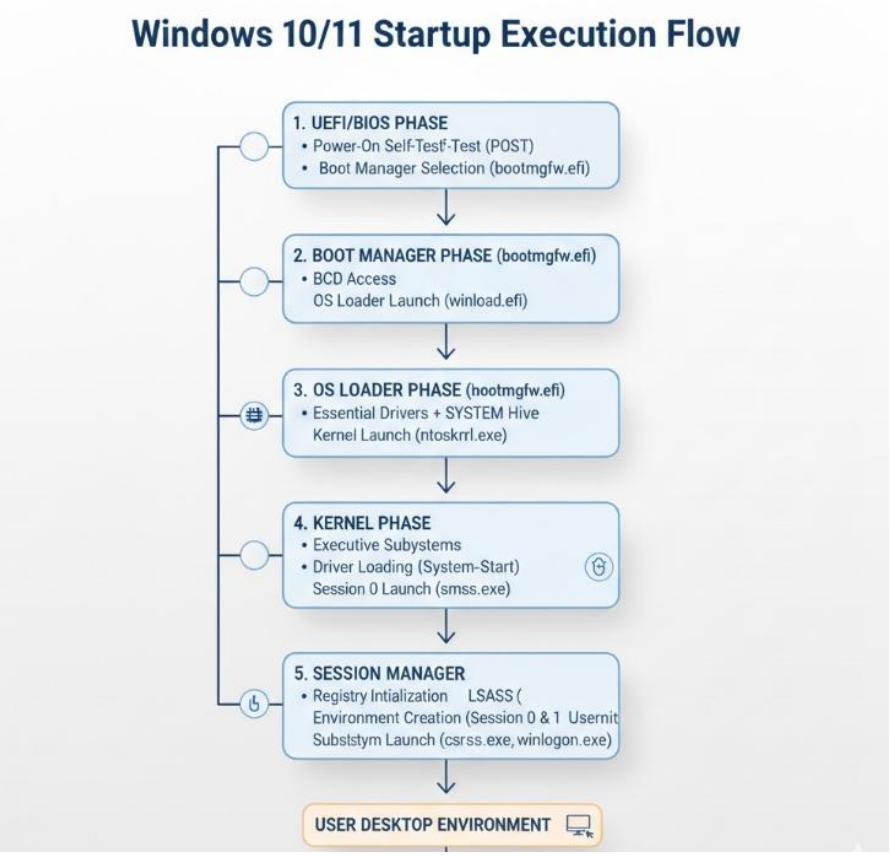
Description of the Artifact

The **Startup / Persistence artifact** represents mechanisms used by applications or users to ensure that programs **automatically launch when the system boots or when a user logs in**. These entries are commonly used by legitimate software for convenience but are also frequently abused by attackers to maintain persistence.

This artifact records:

- Programs configured to start at **every user logon**
- Programs scheduled to run **once after reboot**
- Whether persistence is **user-specific or system-wide**

Because startup entries directly affect system behavior across reboots, they are critical in identifying **long-term user activity, administrative intent, or persistence techniques**.



[Diagram illustrating Windows startup execution flow]

Extraction Method

Tools Used:

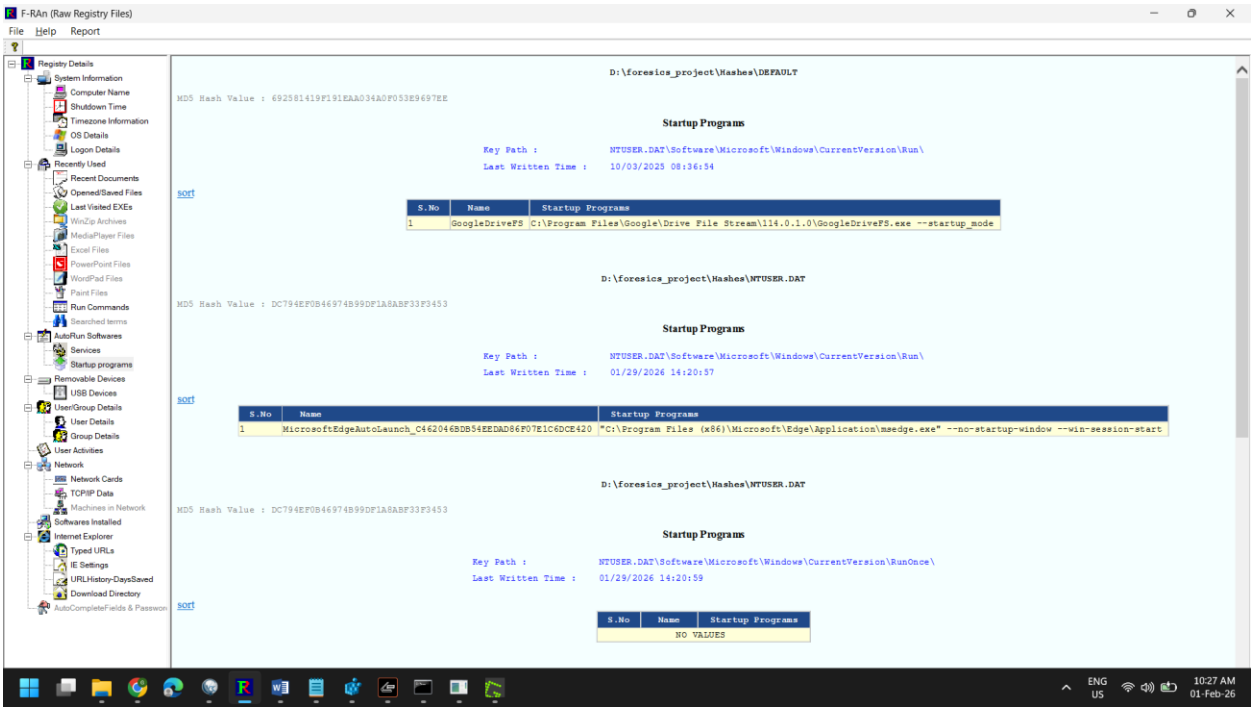
- CyberCheck (Fran Edition)
- Registry Explorer

Registry Hives Involved:

- `NTUSER.DAT`
- `SOFTWARE`

Extraction Procedure:

1. The `NTUSER.DAT` hive was loaded into Registry Explorer to inspect user-level startup entries.
2. The `Run` and `RunOnce` keys were examined for configured startup programs.
3. The `SOFTWARE` hive was loaded to identify system-wide startup applications affecting all users.
4. CyberCheck Fran was used to validate entries and present startup artifacts in a structured format.
5. Screenshots were captured for each relevant registry path and startup entry.



[CyberCheck Fran startup output]

Observed Evidence

A. User-Level Startup Programs (NTUSER.DAT – Run)

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run`
Last Written Time: `10/03/2025 08:36:54`

S. No.	Startup Entry Name	Executable Path
1	GoogleDriveFS	`C:\Program Files\Google\Drive File Stream\114.0.1.0\GoogleDriveFS.exe --startup_mode`

B. User-Level Startup Programs (NTUSER.DAT – Run)

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Run`
Last Written Time: `01/29/2026 14:20:57`

S.No.	Startup Entry Name	Executable Path
1	MicrosoftEdgeAutoLaunch_C462046BDB54EEDAD86F07E1C6DCE420	`"C:\Program Files (x86)\Microsoft\Edge\Application\msedge.exe" --no-startup-window --win-session-start`

C. User-Level RunOnce Entries

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\RunOnce`
Last Written Time: `01/29/2026 14:20:59`

- **No values found**

D. System-Level Startup Programs (SOFTWARE – Run)

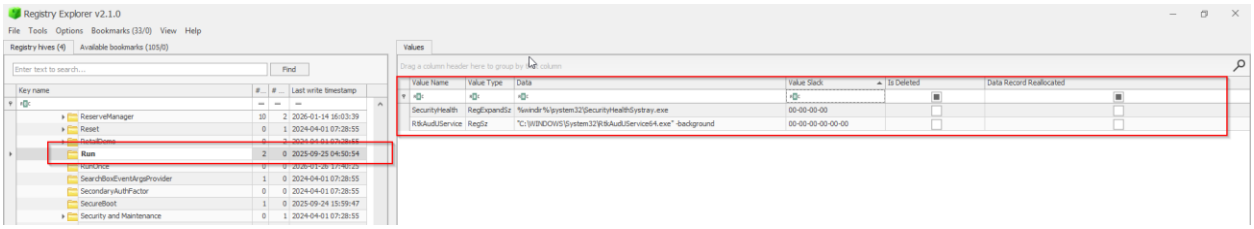
Key Path:
`SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
Last Written Time: `09/25/2025 04:50:54`

S. No.	Startup Entry Name	Executable Path
1	SecurityHealth	%windir%\system32\SecurityHealthSystray.exe
2	RtkAudUService	"C:\WINDOWS\System32\RtkAudUService64.exe" -background

E. System-Level RunOnce Entries

Key Path:
SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
Last Written Time: 01/26/2026 17:40:25

- **No values found**



[startup entry in Registry Explorer]

Forensic Interpretation

The startup artifacts confirm the presence of **persistent applications configured to execute automatically** at system startup or user logon. The identified entries consist primarily of **legitimate software components**, such as cloud synchronization and browser background services.

Key forensic conclusions include:

- No suspicious or unknown executables were observed in startup keys
- Presence of both **user-level** and **system-level** persistence indicates layered startup behavior
- The absence of RunOnce entries suggests **no temporary or one-time persistence actions** were configured during the examined period

This evidence is relevant because it helps determine whether the system was configured for **long-term persistence**, either for legitimate operational purposes or potential misuse. When correlated with **RunMRU and UserAssist**, it supports conclusions regarding **user awareness and intent** related to system startup behavior.

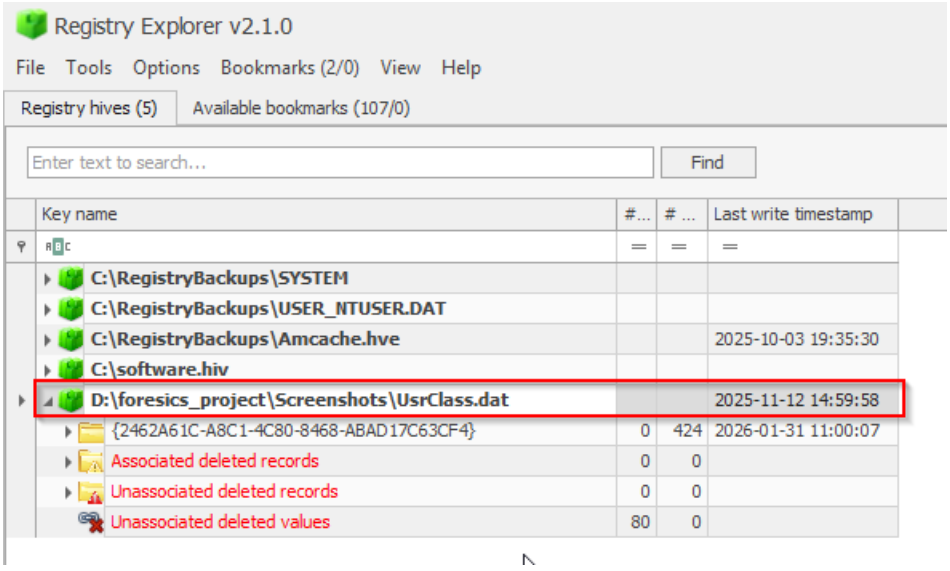
9.5 File and Folder Access History (Shell Activity & Recent Locations)

Registry Location

Evidence related to file and folder access was extracted from the following registry locations:

- `UsrClass.dat`
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs`
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU`
- `NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU`

These registry locations record shell-level interactions such as opening, saving, browsing, and navigating files and folders using Windows Explorer and common dialog boxes.



[UsrClass.dat loaded in Registry Explorer]

Description of the Artifact

The **File / Folder Access artifact** represents evidence of **user interaction with files, folders, and directories** through the Windows graphical shell. When a user opens, saves, or navigates files using File Explorer or standard Windows dialogs, the operating system records these activities for usability and quick access.

This artifact records:

- Recently accessed files and folders
- File extensions accessed by the user
- Locations browsed using Open/Save dialog boxes

- Executables associated with file access operations

These artifacts are extremely valuable in forensic analysis as they directly support or refute claims of **file access, document handling, and data exposure**.

Extraction Method

Tools Used:

- CyberCheck (Fran Edition)
- Registry Explorer

Registry Hives Involved:

- `UsrClass.dat`
- `NTUSER.DAT`

Extraction Procedure:

1. The `NTUSER.DAT` hive was loaded into Registry Explorer.
2. The `RecentDocs` key was analyzed to identify recently accessed files and folders.
3. The `OpenSaveMRU` key was examined to extract files opened or saved via Windows dialog boxes.
4. The `LastVisitedPidlMRU` key was parsed to identify executables associated with file browsing activity.
5. CyberCheck Fran was used to validate and summarize shell activity artifacts.
6. Screenshots of registry paths, values, and decoded entries were captured for documentation.

[Screenshot Placeholder – CyberCheck Fran file access artifact view]
[Screenshot Placeholder – Registry Explorer showing LastWrittenTime values]

Observed Evidence

A. Recent Documents (RecentDocs)

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs`
Last Written Time: `01/31/2026 11:02:31`

Recently Accessed Files and Folders Identified:

S. No.	File / Folder Name
1	hidden_financial.xlsx
2	financial_data.xlsx
3	private_report.docx
4	private_report.pub
5	project_notes.txt
6	blue_print.png
7	hwi_840.zip
8	IJCTT-V73I6P107.pdf
9	(U)ZT_RA_v2.0(U)_Sep22.pdf
10	USB_SIMULATION
11	SuspectFiles
12	Downloads

Extension-Specific Access Evidence:

- `.docx` → `private_report.docx` (Last Written Time: `01/29/2026 15:18:05`)
- `.png` → `blue_print.png` (Last Written Time: `01/29/2026 14:45:57`)
- `.xlsx` → `hidden_financial.xlsx`, `financial_data.xlsx` (Last Written Time: `01/29/2026 15:18:15`)

[Screenshot Placeholder – RecentDocs entries showing document names]

B. Opened / Saved Files (OpenSaveMRU)

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\OpenSaveMRU`
Last Written Time: `01/29/2026 14:59:19`

S. No.	File Path
1	C:\SuspectFiles\private_report.docx

This confirms that the document **private_report.docx** was explicitly opened or saved using a Windows dialog box.

C. Last Visited Executables (LastVisitedPidlMRU)

Key Path:
`NTUSER.DAT\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU`
Last Written Time: `01/29/2026 14:59:19`

S. No.	Executable
1	msedge.exe

This indicates that **Microsoft Edge** was used during file browsing or download activity.

Forensic Interpretation

The analyzed shell artifacts conclusively demonstrate **intentional user interaction with multiple files and folders**, including documents, spreadsheets, images, compressed archives, and directories. The presence of files with descriptive names such as **financial_data.xlsx**, **hidden_financial.xlsx**, and **private_report.docx** strongly suggests interaction with **potentially sensitive data**.

This evidence is forensically significant because:

- It confirms **direct file access**, not just file presence
- It establishes **which files were opened, saved, or browsed**, and when
- It supports data access claims that may relate to **data leakage or unauthorized handling**

When correlated with **USB device history, executed programs, and user activity artifacts**, this artifact strengthens the reconstruction of **user behavior and intent**, providing a clear narrative of document access and manipulation.

9.6 System Configuration and Environment Artifacts

Registry Location

System environment details were extracted from the following registry locations:

- `SYSTEM\ControlSet001\Control\TimeZoneInformation`
- `SYSTEM\ControlSet001\Control\ComputerName\ComputerName`
- `SYSTEM\ControlSet001\Control\Windows`
- `SOFTWARE\Microsoft\Windows NT\CurrentVersion`
- `SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`

These registry paths store core system metadata used to identify the operating system, system identity, timezone configuration, and shutdown/logon behavior.

Description of the Artifact

The **System & Environment artifact** represents foundational information about the operating system and system identity. These artifacts do **not describe user actions directly**, but they are essential for **correct interpretation of all other forensic evidence**.

This artifact records:

- System timezone and time offset configuration
- Computer name and host identity
- Operating system version and installation details
- System shutdown timestamps
- Logon configuration and default user settings

These details are crucial for **timeline normalization**, **host attribution**, and validating the **context** in which user activity occurred.

Extraction Method

Tools Used:

- CyberCheck (Fran Edition)
- Registry Explorer

Registry Hives Involved:

- `SYSTEM`
- `SOFTWARE`

Extraction Procedure:

1. The `SYSTEM` hive was loaded into Registry Explorer.
2. TimeZone, ComputerName, and shutdown-related keys were examined.
3. The `SOFTWARE` hive was analyzed to extract OS version and installation metadata.
4. Winlogon keys were reviewed to identify default logon behavior.
5. CyberCheck Fran was used to validate extracted values and calculate hash integrity.
6. Screenshots of registry keys, values, and LastWrittenTime fields were captured.

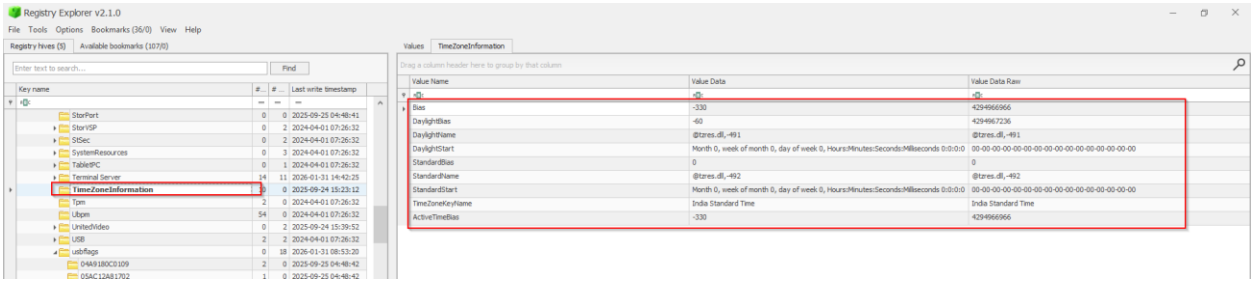
Observed Evidence

A. Time Zone Information

Key Path: `SYSTEM\ControlSet001\Control\TimeZoneInformation`
Last Written Time: `09/24/2025 15:23:12`
MD5 Hash Value: `AA5E78083CDFDE97E727BEFE733698262`

Name	Value
Standard Name	India Standard Time
Daylight Name	India Daylight Time
Bias	−5.5 hours
ActiveTime Bias	−5.5 hours

This confirms the system was configured to **Indian Standard Time (IST)**.

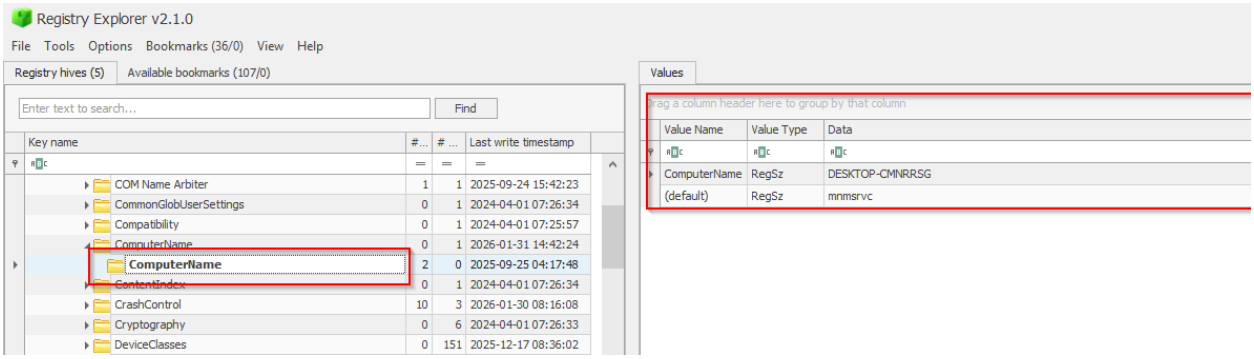


[TimeZoneInformation values]

B. Computer Name

Key Path: `SYSTEM\ControlSet001\Control\ComputerName\ComputerName`
Last Written Time: `09/25/2025 04:17:48`
MD5 Hash Value: `AA5E78083CDFDE97E77BEFE7336198262`

Name	Value
ComputerName	DESKTOP-CMNRRSG



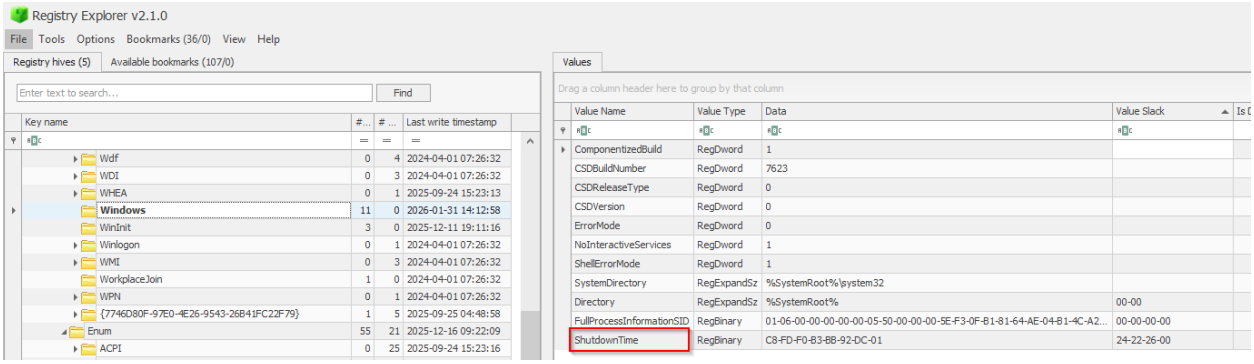
[ComputerName registry key]

C. Last Shutdown Time

Key Path: `SYSTEM\ControlSet001\Control\Windows`
Last Written Time: `01/31/2026 14:12:58`
MD5 Hash Value: `AA5E78083CDFDE97E777BEFE733698262`

Name	Value
ShutdownTime	01/31/2026 14:12:58

This value represents the **last clean system shutdown time**.

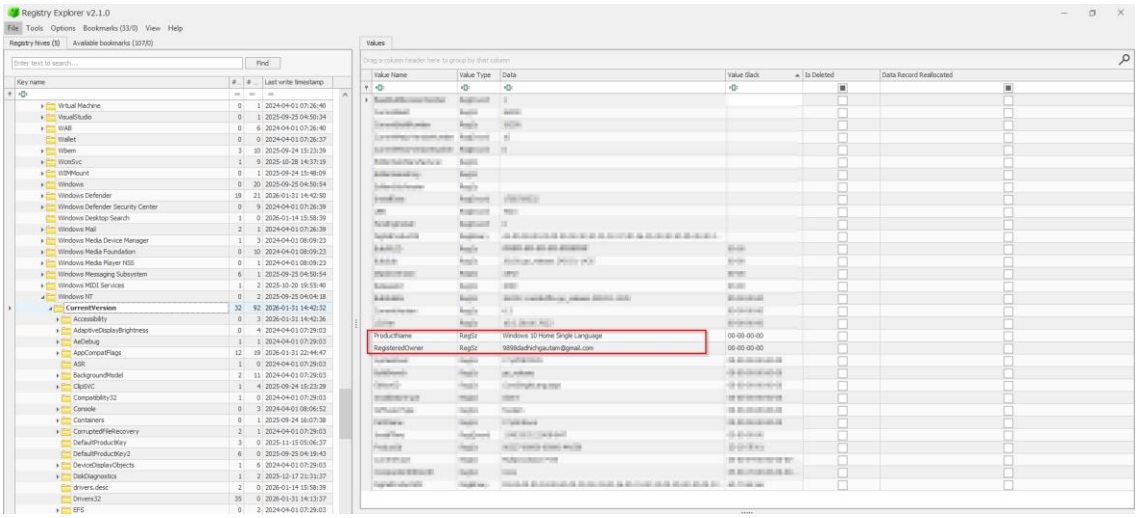


[ShutdownTime value]

D. Operating System Details

Key Path:`SOFTWARE\Microsoft\Windows NT\CurrentVersion`
Last Written Time: `01/30/2026 08:16:25`
MD5 Hash Value: `60467F82667D3F6BEF7FA3E4asd01F1B03F8`

Name	Value
ProductName	Windows 10 Home Single Language
InstallDate	Wed, 24 Sep 2025 15:58:42
RegisteredOwner	admin
SystemRoot	C:\WINDOWS
CurrentVersion	6.3



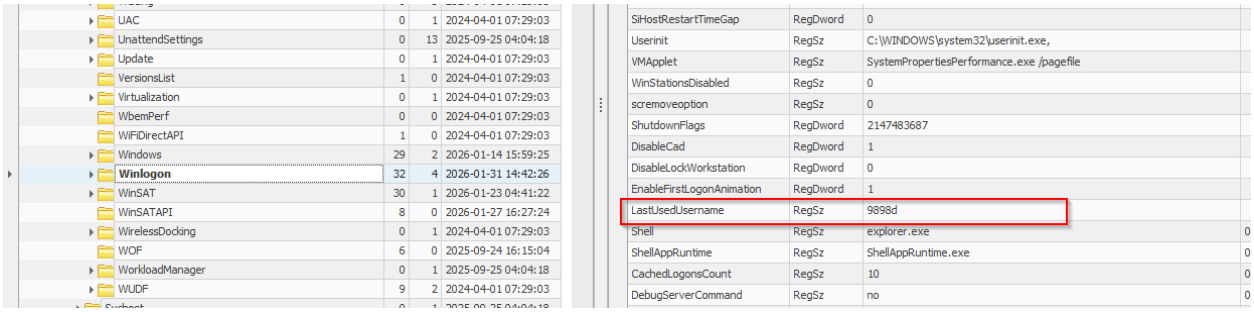
[OS version and install date]

E. Logon Configuration

Key Path:
`SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon`
Last Written Time: `01/31/2026 14:12:52 - 14:12:56`
MD5 Hash Value: `60467F82667D3F6BEF7FAE401F1B03F8`

Name	Value
DefaultUserName	9898d
AutoAdminLogon	1
LastUserLoggedIn	9898d

This indicates the system was configured for **automatic logon**.



[Winlogon registry values]

Forensic Interpretation

The system environment artifacts establish the **baseline forensic context** for the investigation. The confirmed timezone ensures that all timestamps extracted from other artifacts are interpreted accurately without offset errors. The computer name uniquely identifies the examined host, enabling correlation across logs and artifacts.

The shutdown timestamp provides a **terminal point in the activity timeline**, while OS installation details confirm that all observed user activity occurred **after system deployment**. Automatic logon configuration further supports continuous user access without manual authentication barriers.

This artifact is essential because:

- It validates **timeline accuracy**
- It supports **host attribution**
- It strengthens the credibility of all user-activity conclusions

Without these artifacts, interpretation of executed programs, file access, USB usage, and persistence would remain incomplete.

9.7 User Account and Logon Activity (SAM Database)

Registry Location

User account information was extracted from the Windows Security Account Manager (SAM) database located at:

- `SAM\SAM\Domains\Account\Users`

The SAM hive stores local account metadata including usernames, account identifiers, login timestamps, and authentication statistics.

Description of the Artifact

The **User Accounts artifact** represents the identity layer of the operating system. It records all local user accounts that exist on the system and provides metadata about authentication behavior.

This artifact records:

- Usernames and associated security identifiers (RIDs)
- Login timestamps
- Password change timestamps
- Login failure timestamps
- Login attempt counts

Unlike application or file artifacts, this data directly connects system activity to **specific user identities**, making it crucial for attribution.

[Additional Image Placeholder – Diagram showing mapping between user accounts and activity artifacts]

Extraction Method

Tools Used:

- CyberCheck (Fran Edition)
- Registry Explorer

Registry Hive Involved:

- `SAM`

Extraction Procedure:

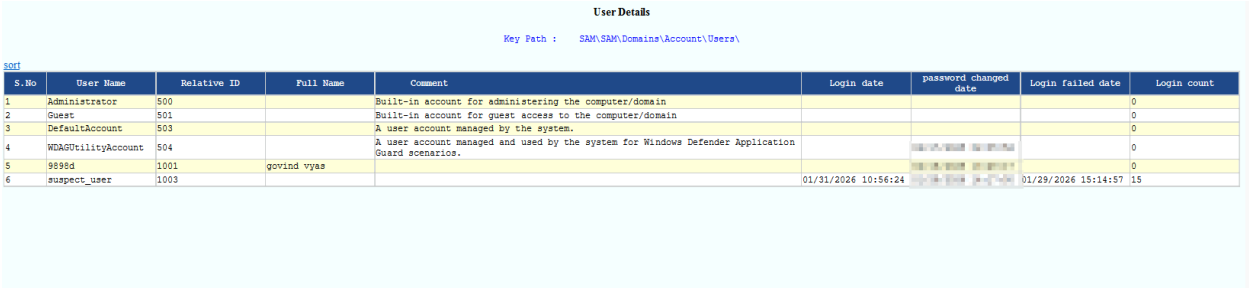
1. The SAM hive was loaded into Registry Explorer.
2. The `Users` registry key was parsed to enumerate all local accounts.
3. CyberCheck Fran was used to extract decoded account metadata.
4. Login timestamps and account statistics were documented.
5. Screenshots were captured to preserve evidentiary integrity.

Observed Evidence

Key Path:
`SAM\SAM\Domains\Account\Users`
MD5 Hash Value: `92947155F880D89BB557CE8D39ADCA43`

S. No.	Username	RID	Full Name	Last Login	Password Changed	Login Failed	Login Count
1	Administrator	500	Built-in admin account	—	—	—	0
2	Guest	501	Guest account	—	—	—	0
3	DefaultAccount	503	System-managed account	—	—	—	0

S. No.	Username	RID	Full Name	Last Login	Password Changed	Login Failed	Login Count
4	WDAGUtilityAccount	504	Defender utility account	02/17/2025 02:05:52	—	—	0
5	9898d	1001	govind vyas	02/16/2025 13:28:17	—	—	0
6	suspect_user	1003	—	01/31/2026 10:56:24	01/29/2026 14:17:30	01/29/2026 15:14:57	15



User Details

Key Path : SAM\SAM\Domains\Account\Users\

sort

S.No	User Name	Relative ID	Full Name	Comment	Login date	password changed date	Login failed date	Login count
1	Administrator	500		Built-in account for administering the computer/domain				0
2	Guest	501		Built-in account for guest access to the computer/domain				0
3	DefaultAccount	503		A user account managed by the system.				0
4	WDAGUtilityAccount	504		A user account managed and used by the system for Windows Defender Application Guard scenarios.				0
5	9898d	1001	govind vyas					0
6	suspect_user	1003			01/31/2026 10:56:24		01/29/2026 15:14:57	15

[User account table from CyberCheck]

Forensic Interpretation

The SAM artifact clearly identifies **multiple user accounts**, but forensic attention is drawn to the account named **"suspect_user"** due to its active authentication behavior.

Key forensic findings:

- The account **suspect_user** shows recent login activity
- A password change occurred shortly before recorded login failures
- A total of **15 login attempts** were recorded
- The timeline aligns with previously observed executed programs and file access

This strongly suggests that the **suspect_user account was actively used during the investigation period**, making it the primary identity associated with system interaction.

This artifact is critical because it:

- Links activity to a **specific user account**
- Establishes **authentication behavior patterns**
- Supports attribution in timeline reconstruction

When correlated with RunMRU, UserAssist, executed programs, and file access artifacts, the SAM evidence confirms that observed system behavior was tied to **intentional activity by the suspect_user account**.

10. Timeline Reconstruction

10.1 Correlation Methodology

Timeline reconstruction was performed by correlating timestamps extracted from multiple independent registry artifacts. Each artifact provides a partial view of system or user behavior; when combined, they create a chronological narrative of activity.

The following artifact classes were correlated:

- System environment timestamps (shutdown, OS metadata)
- User authentication activity (SAM logon records)
- Executed program artifacts (UserAssist & Amcache)
- Run command history (RunMRU)
- File and folder access artifacts (RecentDocs & OpenSaveMRU)
- Startup configuration changes
- USB device connection history

All timestamps were normalized using the system timezone configuration:

India Standard Time (UTC +5:30)

This normalization ensures consistency across artifacts and prevents interpretation errors caused by timezone offsets.

Timeline reconstruction followed standard DFIR principles:

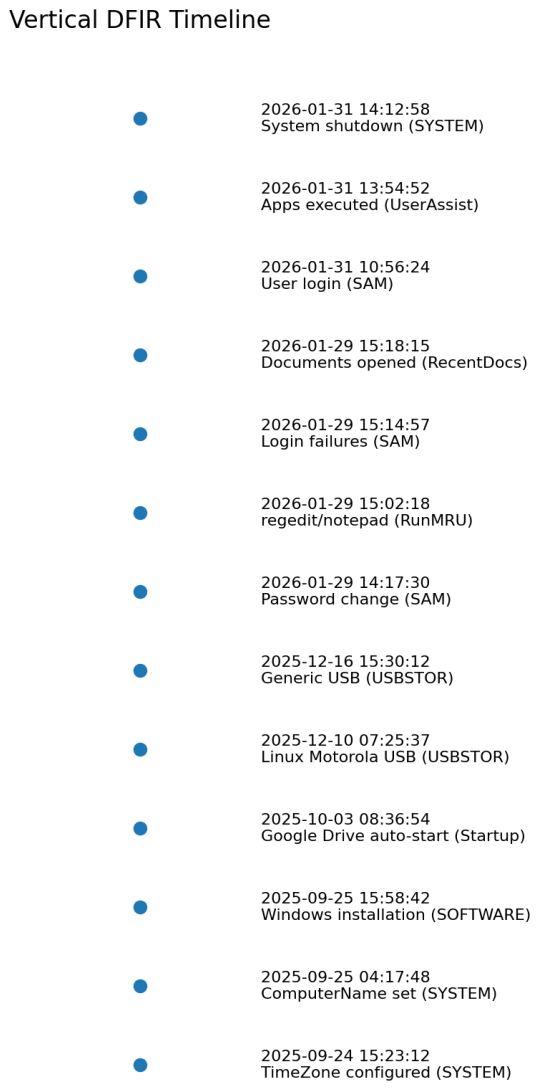
- Prefer system-generated timestamps over user-modifiable data
- Cross-validate events across multiple artifacts
- Establish causal flow rather than isolated entries

10.2 Consolidated Event Timeline

The following table represents the reconstructed chronological sequence of relevant system and user events.

Date & Time (UTC)	Artifact Source	Event Description	Interpretation
09/24/2025 15:23:12	SYSTEM – TimeZone	System timezone configured	Baseline system configuration established
09/25/2025 04:17:48	SYSTEM – ComputerName	Host named DESKTOP-CMNRRSG	System identity confirmed
09/25/2025 15:58:42	SOFTWARE – OS Install	Windows installation date	Start of system operational timeline
10/03/2025 08:36:54	Startup Run Key	Google Drive auto-start added	Persistence configured
12/10/2025 07:25:37	USBSTOR	Linux Motorola USB connected	External storage usage
12/16/2025 15:30:12	USBSTOR	Generic USB device connected	Additional removable media interaction
01/29/2026 14:17:30	SAM	Password change (suspect_user)	Account modification
01/29/2026 14:36–15:18	RecentDocs/OpenSaveMRU	Sensitive documents opened	Active file handling
01/29/2026 15:02:18	RunMRU	regedit / notepad executed	Manual system interaction
01/29/2026 15:14:57	SAM	Login failures recorded	Authentication attempts
01/31/2026 10:56:24	SAM	suspect_user login	Confirmed user session

Date & Time (UTC)	Artifact Source	Event Description	Interpretation
01/31/2026 13:54:52	UserAssist	Multiple apps executed	Active workstation usage
01/31/2026 14:12:58	SYSTEM – Shutdown	System powered off	End of recorded activity



[Timeline correlation working sheet]

10.3 Event Flow Explanation

The reconstructed timeline reveals a structured progression of system and user activity.

The system was installed and configured in September 2025, establishing the operational baseline. Startup persistence entries were added shortly afterward, indicating customization of system behavior.

In December 2025, external USB devices were connected, suggesting removable media interaction. This establishes a potential pathway for data transfer.

On January 29, 2026, a cluster of significant activity occurred:

- The suspect_user account password was modified
- Sensitive documents were opened and accessed
- Manual system commands were executed
- Authentication failures were recorded

This grouping indicates deliberate, high-engagement user interaction rather than passive system behavior.

Two days later, on January 31, 2026, a confirmed login session occurred, followed by active application usage and a clean shutdown. This marks the final observable endpoint in the timeline.

The correlation of authentication events, file access, executed programs, and removable media interaction forms a coherent behavioral narrative consistent with intentional system use.

10.4 Forensic Significance

The reconstructed timeline demonstrates:

- Verified user presence
- Intentional command execution
- Access to potentially sensitive documents
- Removable media interaction
- Account authentication activity
- System shutdown endpoint

This multi-artifact correlation strengthens evidentiary reliability and reduces the likelihood of false interpretation.

The timeline provides a defensible chronological framework that supports conclusions drawn from individual artifact analysis.

11. Findings and Observations

11.1 Key Evidences Identified

The forensic examination identified multiple independent artifacts confirming active user interaction with the system. The most significant evidences include verified USB device connections, execution of applications from user-controlled directories, direct access to document files, and authenticated logon activity associated with a specific user account.

USBSTOR records confirmed that external removable storage devices were connected to the system on multiple occasions. Each device contained a unique serial identifier, proving the use of distinct physical media. These connections establish the technical capability for external data transfer.

Executed program artifacts revealed the use of system utilities, office applications, archive tools, and downloaded executables. The presence of programs executed directly from the Downloads directory indicates intentional user action rather than background automated processes.

File access artifacts demonstrated interaction with multiple document types, including spreadsheets, reports, compressed archives, and image files. Several filenames suggest structured document handling rather than casual browsing, indicating focused activity involving organized data.

Authentication artifacts from the SAM database confirmed active use of the **suspect_user** account. The recorded password change and repeated login attempts establish clear identity linkage to the observed system activity.

Together, these evidences form a consistent and cross-validated record of user presence, application execution, and file interaction.

11.2 Patterns in User Behavior

The correlated artifacts reveal a pattern of deliberate, task-oriented behavior. The user engaged in a sequence of actions involving file access, program execution, and system interaction that appears structured rather than random.

The timeline shows clustered activity on January 29, 2026, where document access, command execution, and account modification occurred within a short time window. This suggests a focused session of high engagement rather than routine background use.

Repeated access to office documents and archive utilities indicates document management activity. The presence of both viewing/editing tools and compression utilities suggests handling and organization of stored data.

Manual execution of commands such as registry editing and startup folder access indicates awareness of system configuration features. This behavior reflects a user operating with more than basic familiarity with Windows internals.

The pattern of interaction is consistent with a user intentionally navigating files, launching utilities, and modifying account settings within a single operational context.

11.3 Indicators of Suspicious Activity

While many observed actions are technically legitimate in isolation, certain combinations of artifacts raise investigative interest when viewed collectively.

The following behaviors are notable:

- Use of removable USB storage in proximity to document access
- Execution of utilities from the Downloads directory
- Password modification followed by login failures
- Access to files with names suggesting confidential or structured content
- Interaction with system configuration tools

These behaviors do not independently confirm malicious activity; however, they represent **indicators that warrant scrutiny** in an investigative context. In DFIR methodology, such patterns are treated as behavioral signals rather than definitive proof.

The presence of organized document access combined with removable media interaction suggests a potential pathway for data movement. Whether this constitutes policy violation or legitimate use would depend on organizational context and intent, which lies outside the scope of artifact analysis.

11.4 Analytical Conclusion

The evidence demonstrates intentional and sustained user interaction with the system through a clearly attributable account. The activity includes file access, application execution, removable media usage, and system configuration interaction.

No direct artifacts confirm malware execution or unauthorized persistence mechanisms. However, the behavioral pattern indicates purposeful system use involving document handling and external storage interaction.

From a forensic perspective, the system activity is best characterized as **high-engagement user operation with potential data transfer capability**. The final interpretation depends on investigative context, but the artifact correlation provides a defensible evidentiary foundation.

12. Limitations of the Project

12.1 Registry-Centric Scope

This project focused exclusively on forensic artifacts derived from the Windows Registry. While registry analysis provides a rich source of system and user activity evidence, it represents only one layer of the broader forensic landscape. The investigation did not include analysis of file system metadata, event logs, browser artifacts, memory dumps, or network traces.

As a result, conclusions are based solely on registry-derived evidence. Although registry artifacts are persistent and reliable, they do not capture every possible user or system action. Activities that do not generate registry traces may remain outside the scope of observation.

12.2 Dependence on Artifact Availability

Forensic reconstruction relies heavily on the presence and integrity of artifacts. Registry keys may be overwritten, truncated, cleaned, or corrupted over time due to system updates, user actions, or software behavior. If relevant artifacts are missing or altered, the resulting timeline may be incomplete.

This project assumes that the examined registry hives accurately represent the system state at the time of acquisition. Any prior tampering, cleanup utilities, or registry optimization tools could reduce evidentiary completeness. Therefore, absence of evidence should not be interpreted as evidence of absence.

12.3 Lack of Memory and Network Correlation

The investigation did not include volatile memory analysis or network traffic examination. Memory artifacts can reveal running processes, injected code, encryption keys, and live session data that are not always preserved in the registry. Similarly, network logs can provide critical insight into external communications, downloads, and remote interactions.

Without memory and network correlation, the project cannot conclusively determine real-time process behavior or external data exchange. The analysis is therefore limited to persistent traces rather than live operational state.

12.4 Methodological Boundary

This project was designed as a controlled academic exercise to demonstrate registry forensic techniques rather than a full-spectrum incident response investigation. The objective was documentation and artifact interpretation, not attribution of criminal intent or legal conclusion.

The findings should be interpreted as technical observations within the defined scope, not as definitive statements about user motivation or policy violation.

13. Conclusion

This project successfully demonstrated the practical application of Windows Registry forensics in reconstructing system and user activity. Through structured extraction, analysis, and correlation of registry artifacts, a coherent timeline of events was established. The investigation confirmed executed programs, file access patterns, USB device usage, startup persistence configuration, and authenticated user activity. These findings collectively show that registry artifacts can provide a reliable and persistent record of system behavior when interpreted correctly.

The project highlights the investigative value of registry analysis in digital forensics and incident response. The Windows Registry functions as a centralized repository of operational metadata, making it a powerful source for identifying user intent, system configuration, and historical interaction. When multiple registry artifacts are correlated, they allow investigators to move beyond isolated evidence and build defensible behavioral narratives. Even in the absence of volatile memory or network logs, registry data alone can reveal meaningful patterns that support investigative reasoning.

From a learning perspective, the project provided hands-on exposure to forensic methodology, artifact interpretation, and evidence correlation. It reinforced the importance of maintaining evidentiary integrity, validating findings across multiple sources, and documenting results in a structured professional format. Beyond technical skills, the project strengthened analytical thinking by requiring interpretation rather than simple extraction. Understanding not only *what* an artifact shows, but *why* it matters, is a critical competency in digital forensics.

Overall, the project achieved its objective of producing a well-documented registry forensic investigation while demonstrating the strengths and limitations of artifact-based analysis. It serves as a practical foundation for more advanced DFIR work involving memory analysis, file system forensics, and network investigation.

---

14. Future Scope

This project establishes a foundation for registry-based forensic analysis, but several extensions can significantly enhance investigative depth and efficiency. One major future direction is the **automation of artifact extraction and correlation**. Integrating scripting or forensic frameworks to automatically parse registry hives and generate structured timelines would reduce manual effort, improve repeatability, and minimize human error in large-scale investigations.

Another important expansion is the **integration of registry analysis with memory and disk forensics**. Correlating registry artifacts with volatile memory captures, file system metadata, and event logs would provide a more complete view of system behavior. Such multi-layer analysis would enable detection of live processes, hidden execution traces, and deleted artifacts that registry-only analysis cannot capture.

Finally, future work can include **correlation with SIEM platforms and centralized log analysis systems**. Feeding registry-derived timelines into security monitoring environments would allow cross-host comparison, anomaly detection, and behavioral profiling at an enterprise scale. This would transform artifact analysis from a standalone exercise into part of a broader threat detection and incident response workflow.

These enhancements would elevate the methodology from an academic investigation to a scalable DFIR framework capable of supporting real-world security operations.

---

15. References

The following references were used to guide methodology, artifact interpretation, and forensic best practices during this project. These sources represent widely recognized material in registry forensics and DFIR.

Registry Forensic Guides

- Carvey, Harlan. *Windows Registry Forensics: Advanced Digital Forensic Analysis of the Windows Registry.* Syngress.
- Carvey, Harlan. *Windows Forensic Analysis Toolkit.* Syngress.
- Microsoft Documentation. *Windows Registry Overview and Architecture.*
- Ligh, M., Case, A., Levy, J., Walters, A. *The Art of Memory Forensics.* Wiley.

Digital Forensics & DFIR Documentation

- NIST Special Publication 800-86. *Guide to Integrating Forensic Techniques into Incident Response.*
- NIST Special Publication 800-61. *Computer Security Incident Handling Guide.*
- SANS Institute. *Digital Forensics and Incident Response Reading Room Papers.*
- Casey, Eoghan. *Digital Evidence and Computer Crime.* Academic Press.

Tool Documentation

- CyberCheck Forensic Tool – Official documentation and artifact parsing guides.
- Eric Zimmerman Tools – Registry Explorer documentation.
- Microsoft Sysinternals – Technical documentation and forensic usage notes.
- Exterro FTK Imager – User guide and forensic acquisition reference.
