Windows User Activity Analysis

INTRODUCTION

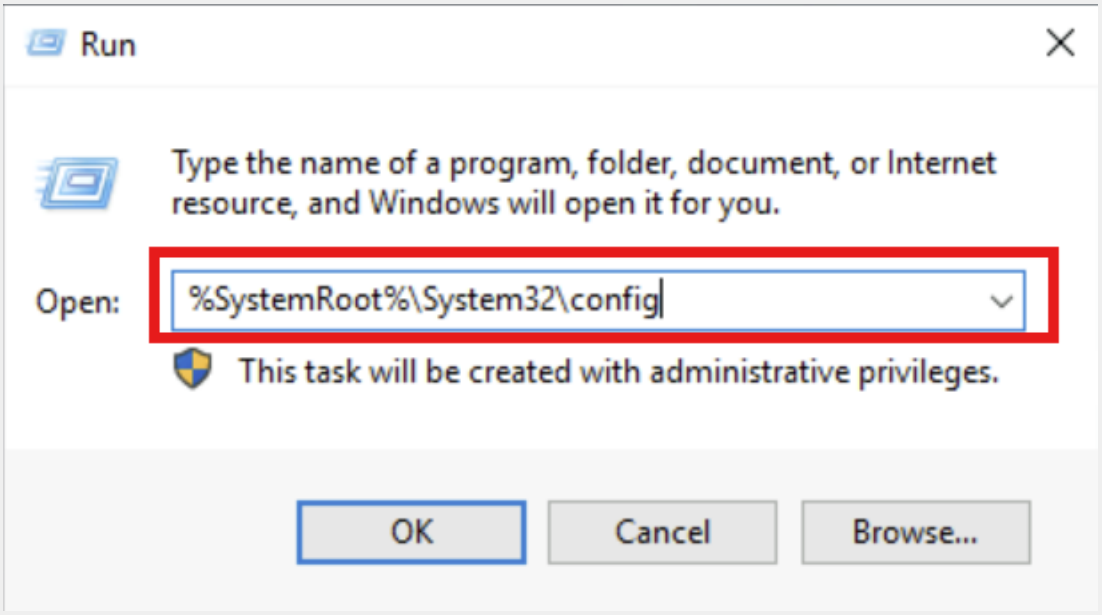
Windows User Activity Analysis is a cornerstone of Digital Forensics and Incident Response (DFIR) investigations. This process involves examining artifacts generated by user interactions with the Windows operating system, applications, and network resources. These artifacts, often dispersed across event logs, registry entries, file system metadata, and application-specific logs, provide a chronological narrative of user actions. By analyzing this data, investigators can reconstruct events leading up to and following a security incident, identify the individuals involved, and ascertain the methods used by malicious actors. The importance of Windows User Activity Analysis cannot be overstated in the context of modern cybersecurity challenges. As a widely used operating system in enterprise environments, Windows is often a primary target for cyberattacks and insider threats. Investigating user activity on a compromised system enables organizations to understand the scope and impact of an incident. For example, such analysis can reveal unauthorized access, data exfiltration attempts, or deliberate misuse of privileges. Additionally, it is instrumental in identifying gaps in security controls and establishing patterns indicative of emerging threats. From a DFIR perspective, the ability to analyze Windows user activity is critical for several reasons. First, it aids in the preservation and interpretation of digital evidence, ensuring that findings are admissible in legal proceedings if required. Second, it enhances threat intelligence efforts by uncovering tactics, techniques, and procedures (TTPs) used by attackers. Third, this analysis provides actionable insights that enable organizations to improve their security posture and resilience against future incidents.

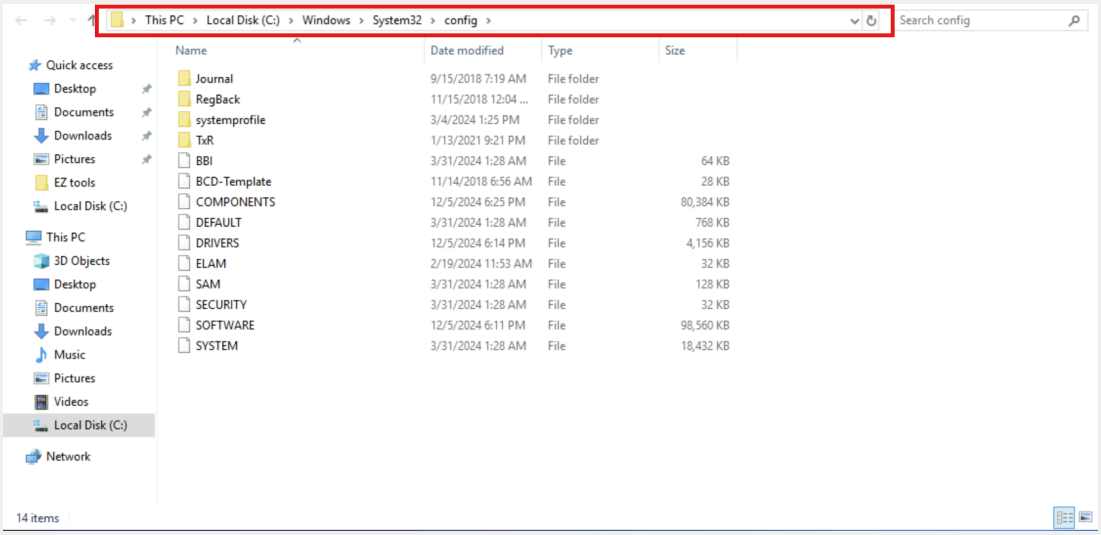
Baris Dincer / Cyber Threat Intelligence Investigator & CIO / Lex Program

Importance of Windows Registry in User Activity Analysis

The Windows Registry serves as a centralized hierarchical database that stores configuration settings and options for the operating system, applications, hardware, and user profiles. It is a critical artifact in Windows User Activity Analysis as it provides a wealth of information about system and user behaviors. By examining specific registry keys and values, investigators can uncover details such as user login activity, installed programs, recently accessed files, connected USB devices, and application usage. The registry's importance in DFIR stems from its role in preserving traces of user interactions and system configurations. These traces are often leveraged to establish a timeline of events, identify suspicious activities, or correlate evidence from other sources. For example, the registry can reveal evidence of persistence mechanisms employed by malware, unauthorized software installations, or deleted user profiles that might otherwise be missed in file system analysis. Moreover, specific registry hives are crucial for forensic investigations as they store information at both the system and user levels.

The Windows Registry is structured as a collection of hierarchical databases known as "**hives**", each of which serves a specific purpose in managing system and user configurations. These hives are stored as files in the **%SystemRoot%\System32\config** directory, making them accessible for analysis during forensic investigations.





The **SAM** (**Security Account Manager**) hive contains security-related information about user accounts and security policies. This includes details such as user and group account data, password hashes, and login attempts. In a live system, the SAM hive is loaded under the key **HKLM\Local_Machine\SAM**. Forensic analysis of this hive is crucial for identifying account-based activities, unauthorized access attempts, and potential privilege escalations.

The **SECURITY** hive holds configuration data related to system security, including user authentication mechanisms, permissions, and local security policies. It provides insight into how access control is enforced on the system. Investigating this hive is vital for understanding how attackers may have exploited weak security configurations or bypassed authentication controls.

The **SYSTEM** hive contains essential configuration data about the operating system's hardware, device drivers, and startup settings. This includes details about services, connected devices, and boot configurations. It plays a significant role in determining how the system was configured and whether malicious changes were made to facilitate persistence or disable security features.

The **SOFTWARE** hive stores configuration information for installed applications and system-wide settings. This includes details about application versions, installation dates, and sometimes usage statistics. Analysis of this hive can reveal evidence of unauthorized or malicious software installations and provide a timeline of software-related activities.

The **DEFAULT** hive acts as a template for creating new user profiles. It provides default settings for user-specific configurations, including basic preferences and system behavior. While less commonly analyzed in detail, this hive can be relevant when investigating the setup of new accounts, particularly those that may have been created for malicious purposes.

These registry hives are integral to Windows' operation, storing critical information about the system's state, user activities, and security configurations. By examining them during DFIR investigations, forensic experts can uncover significant evidence about how a system was used or compromised.

In addition to the core system-level registry hives such as SECURITY and DEFAULT, Windows also maintains user-specific hives that store configurations and preferences unique to individual user profiles. These hives, **NTUSER.DAT** and **USRCLASS.DAT**, are pivotal for understanding user activity and personalized system usage.

The **NTUSER.DAT** file contains user-specific settings and configurations for each user profile on the system. It stores information about user preferences, application settings, recently accessed files, and other personalized data. This hive plays an important role in analyzing user behavior and identifying activities such as file usage, system customizations, and software interactions. Each user's **NTUSER.DAT** file is located within their profile directory under the **%USERPROFILE%** path. In a live system, the **NTUSER.DAT** hive is mapped to **HKEY_CURRENT_USER** (**HKCU**) in the Windows Registry. This mapping allows the operating system and applications to access user-specific settings dynamically during operation. Examining **NTUSER.DAT** is essential for reconstructing a timeline of user activity. It can reveal recently accessed files, application usage patterns, and potential evidence of malicious actions or unauthorized access attempts.

```
PS C:\Users\Administrator> Get-ChildItem -Path C:\Users\ -Recurse -Force -Filter NTUSER.dat 2>$null


    Directory: C:\Users\Administrator

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-h--        3/31/2024   1:28 AM        1048576 NTUSER.DAT

    Directory: C:\Users\Default

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-h--       10/13/2022  10:55 PM         524288 NTUSER.DAT
```

The **USRCLASS.DAT** file stores user-specific class and interface settings, including data related to Windows Explorer's settings and user interactions with the desktop environment. This hive provides insights into user interaction with the Windows shell, such as desktop configurations, folder view settings, and recently used directories. The USRCLASS.DAT file resides within the user's local application data directory. It is related to **%USERPROFILE%\AppData\Local\Microsoft\Windows\UsrClass.dat** directory. In the live registry, **USRCLASS.DAT** is mapped under **HKEY_CURRENT_USER\Software\Classes**. This key holds information about file associations, shell extensions, and other interaction-related settings for the current user. Analyzing **USRCLASS.DAT** can help investigators identify recent user interactions with files, folders, and system settings. It also aids in detecting anomalies, such as modifications to file associations or suspicious shell extensions introduced by malware.

```
PS C:\Users\Administrator> Get-ChildItem -Path C:\Users\ -Recurse -Force -Filter UsrClass.dat 2>$null


    Directory: C:\Users\Administrator\AppData\Local\Microsoft\Windows


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-h--        3/31/2024   1:28 AM        1572864 UsrClass.dat


    Directory: C:\Users\Default\AppData\Local\Microsoft\Windows

Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a-h--        3/17/2021   2:57 PM          65536 UsrClass.dat
```

The hives act as top-level keys within the registry and are essential for managing both system-wide and user-specific configurations. Below is a detailed explanation of the primary registry hives:

HKEY_CLASSES_ROOT (HKCR):
This hive contains information about file associations and Object Linking and Embedding (OLE) object classes. It determines how files are opened and which application is used to handle specific file types.

Purpose:

- Defines file extensions and their associated programs.
- Stores COM (Component Object Model) object registration data.
- Plays a key role in the integration between applications.

Analyzing **HKCR** can reveal malicious alterations to file associations, such as redirecting file types to malware or tampered executable paths.

HKEY_CURRENT_USER (HKCU):
This hive contains configuration settings for the currently logged-in user. It includes user-specific preferences, such as desktop settings, application configurations, and user environment variables.

Purpose:

- Manages user-specific appearance and personalization settings.
- Controls preferences for applications specific to the active user.

Examination of **HKCU** can uncover user activity, such as changes to software settings, evidence of user interactions, or potential traces of user-specific malware.

HKEY_LOCAL_MACHINE (HKLM):
This hive holds system-wide settings and configurations applicable to all users on the computer. It contains essential information about hardware, software, and security.

Subkeys of Interest:

- SAM: Maps to HKLM\Local_Machine\SAM, containing user account security data.
- SYSTEM: Stores information about system hardware and drivers.
- SOFTWARE: Provides details about installed applications and system-wide configurations.

HKLM is critical for analyzing system integrity, identifying installed software, and detecting unauthorized modifications to security policies or drivers.

HKEY_USERS (HKU):
This hive contains settings for all user profiles on the system. Each user has a dedicated subkey identified by their Security Identifier (SID).

Purpose:

- Maintains global configurations for multiple users.
- Provides access to specific user hives, including NTUSER.DAT.

Investigators can analyze inactive user profiles for evidence of activity or malware targeting specific accounts.

HKEY_CURRENT_CONFIG (HKCC):
This hive provides information about the current hardware profile being used by the system. It acts as a dynamic view of hardware-related settings, such as device configurations.

Purpose:

- Reflects the current system configuration during runtime.
- Resolves conflicts between multiple hardware profiles.

Investigating **HKCC** can reveal changes to hardware configurations, potential tampering with connected devices, or evidence of rogue peripherals.

The registry is an indispensable artifact in forensic investigations. Each hive offers unique insights into the state of the system and user activities. As a short itinerary, you can check the order below:

- **HKCR:** Tracks file associations and application integration, aiding in malware detection.
- **HKCU:** Reveals user-specific activities and preferences.
- **HKLM:** Provides a comprehensive view of the system's software and hardware configuration.
- **HKU:** Facilitates the investigation of inactive or less frequently used user profiles.
- **HKCC:** Helps understand hardware-related changes and configurations.

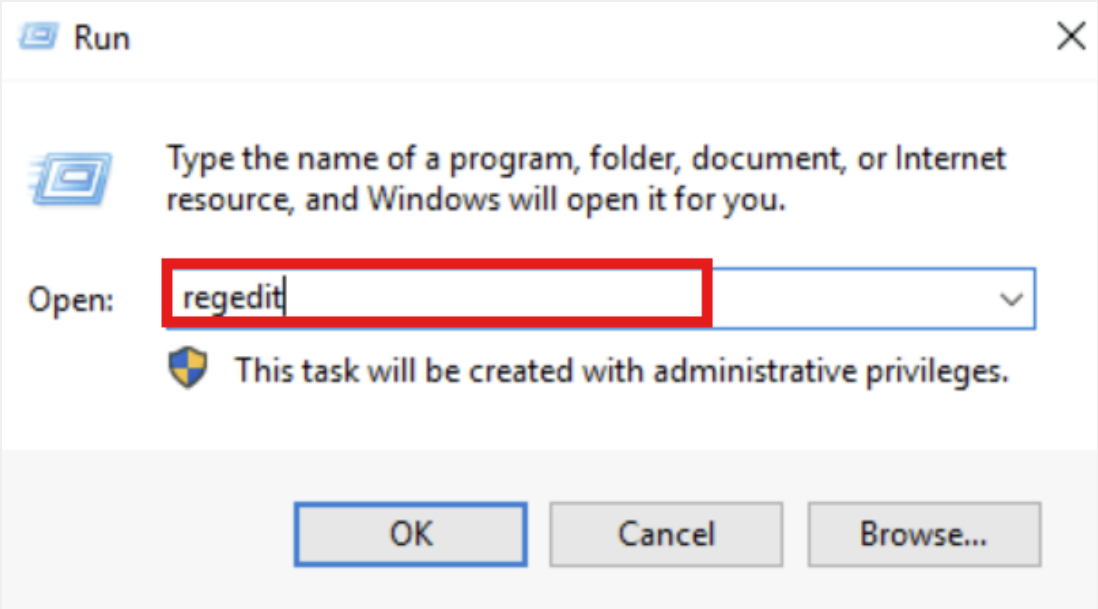
By systematically analyzing these hives, investigators can uncover critical evidence of compromise, misconfigurations, or malicious activity, forming a detailed narrative of an incident.

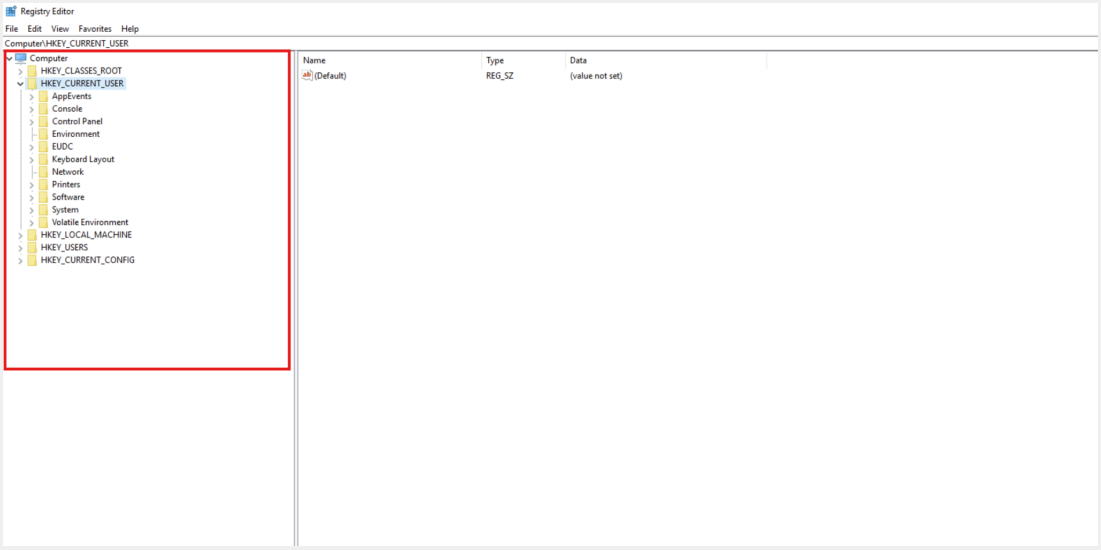
Transaction logs and dirty hives are critical components of Windows registry management and hold significant importance in forensic investigations. They provide insights into system activity, configuration changes, and potential security issues. Transaction logs are used to ensure the integrity and consistency of the Windows Registry. They record changes made to the registry hives over time and allow the operating system to handle failures gracefully. Transaction logs are stored in the same directory as the main hive files, typically **%SystemRoot%\System32\config**. They are named after their corresponding hive files with extensions such as **.LOG1** and **.LOG2**. In databases, these logs track changes (inserts, updates, deletes) to maintain data integrity and ensure consistency in case of failures. Transaction logs allow forensic analysts to reconstruct registry activity over time. They are invaluable for identifying recent changes to the system, determining when configuration modifications occurred, and correlating these changes with other artifacts to build a timeline of events.

]\

```
C:\Windows\System32\config>dir /a
 Volume in drive C has no label.
 Volume Serial Number is A8A4-C362

 Directory of C:\Windows\System32\config

12/05/2024  06:27 PM    <DIR>          .
12/05/2024  06:27 PM    <DIR>          ..
03/31/2024  01:28 AM            65,536 BBI
09/15/2018  06:09 AM            40,960 BBI.LOG1
09/15/2018  06:09 AM            81,920 BBI.LOG2
11/14/2018  06:56 AM            28,672 BCD-Template
11/14/2018  06:56 AM            28,672 BCD-Template.LOG
11/14/2018  06:56 AM                 0 BCD-Template.LOG1
11/14/2018  06:56 AM                 0 BCD-Template.LOG2
12/05/2024  06:25 PM        82,313,216 COMPONENTS
09/15/2018  06:09 AM             8,192 COMPONENTS.LOG1
09/15/2018  06:09 AM        15,727,614 COMPONENTS.LOG2
03/31/2024  01:28 AM           786,432 DEFAULT
09/15/2018  06:09 AM           323,584 DEFAULT.LOG1
09/15/2018  06:09 AM            65,536 DEFAULT.LOG2

[REDACTED] - MORE
```

The **Windows Registry Editor**, commonly referred to as **Regedit**, is a built-in graphical tool for accessing, viewing, and modifying the Windows Registry. It provides users and administrators with a direct interface to interact with registry hives and their associated keys and values. While it is a powerful tool, it must be used cautiously, as incorrect modifications can lead to system instability or malfunction. **Regedit** allows users to navigate through the hierarchical structure of the registry, organized into **Hives**, **Keys**, **Subkeys**, and **Values**. Users can **add**, **modify**, or **delete** registry keys and values. **Regedit** includes a search feature to locate specific keys or values within the registry. Users can export parts of the registry to a **.reg** file for backup or sharing. **Regedit** provides the ability to set permissions for keys, ensuring proper security and access control.

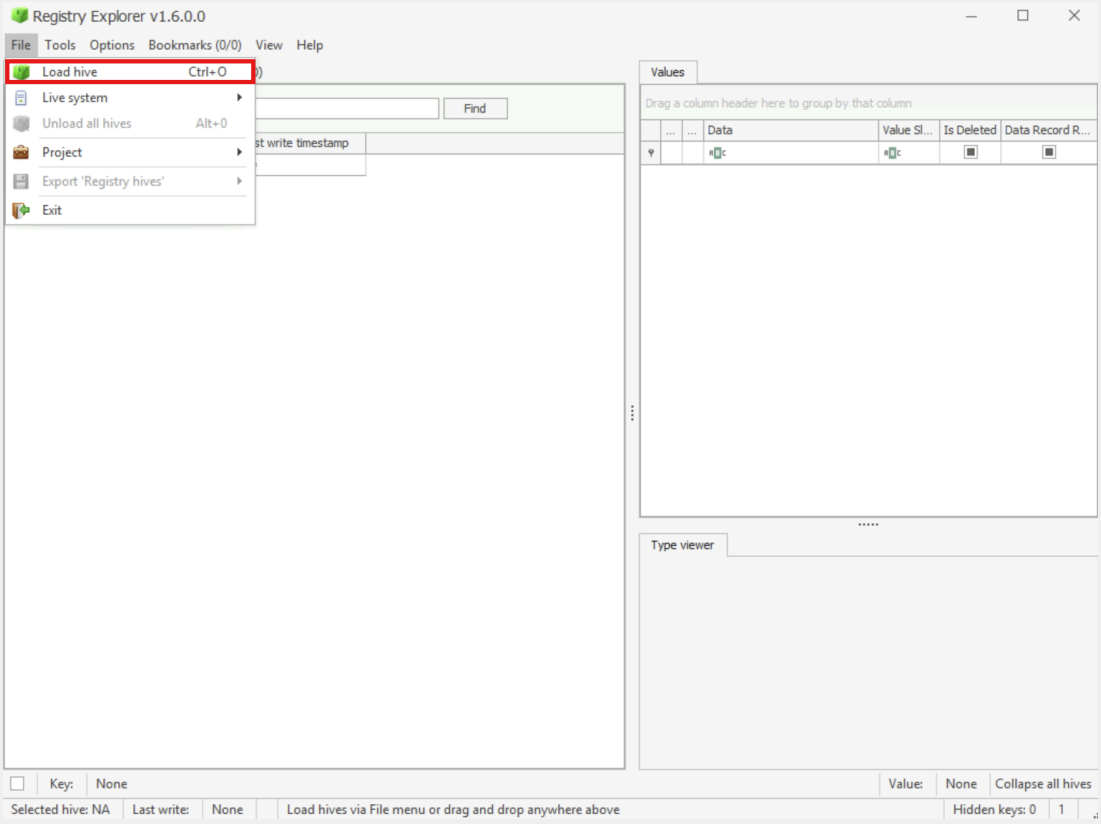


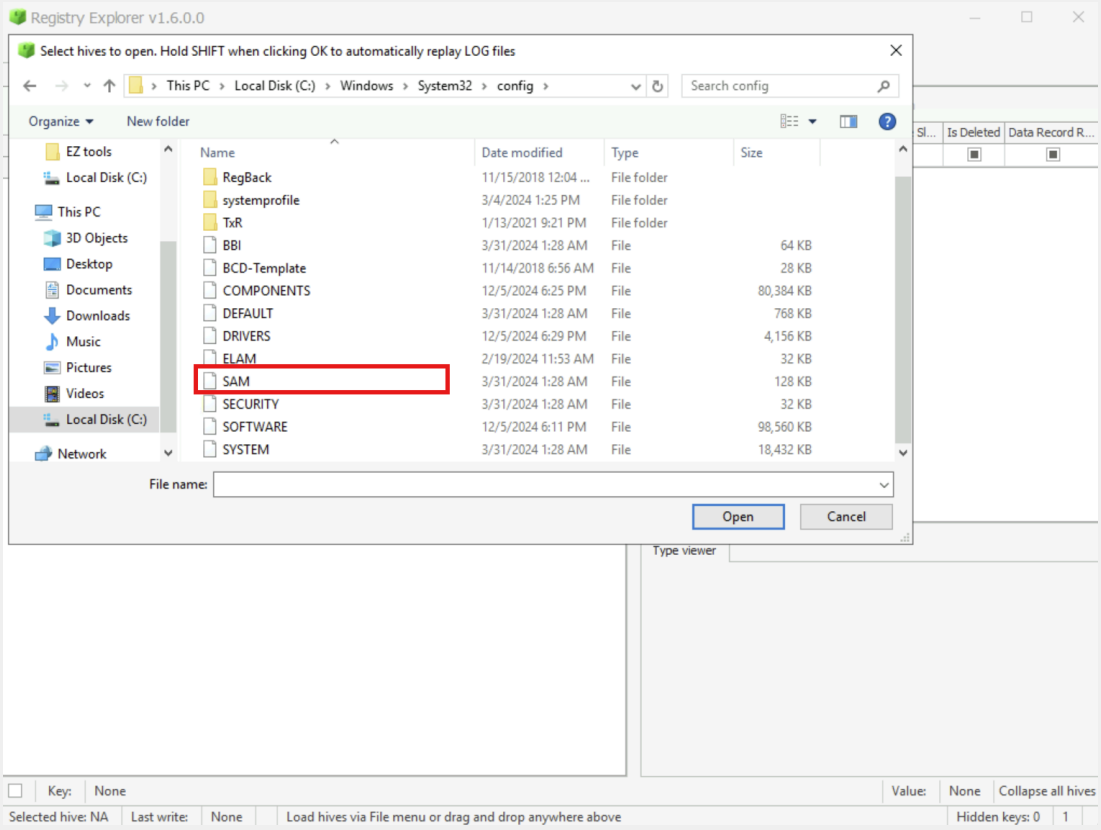


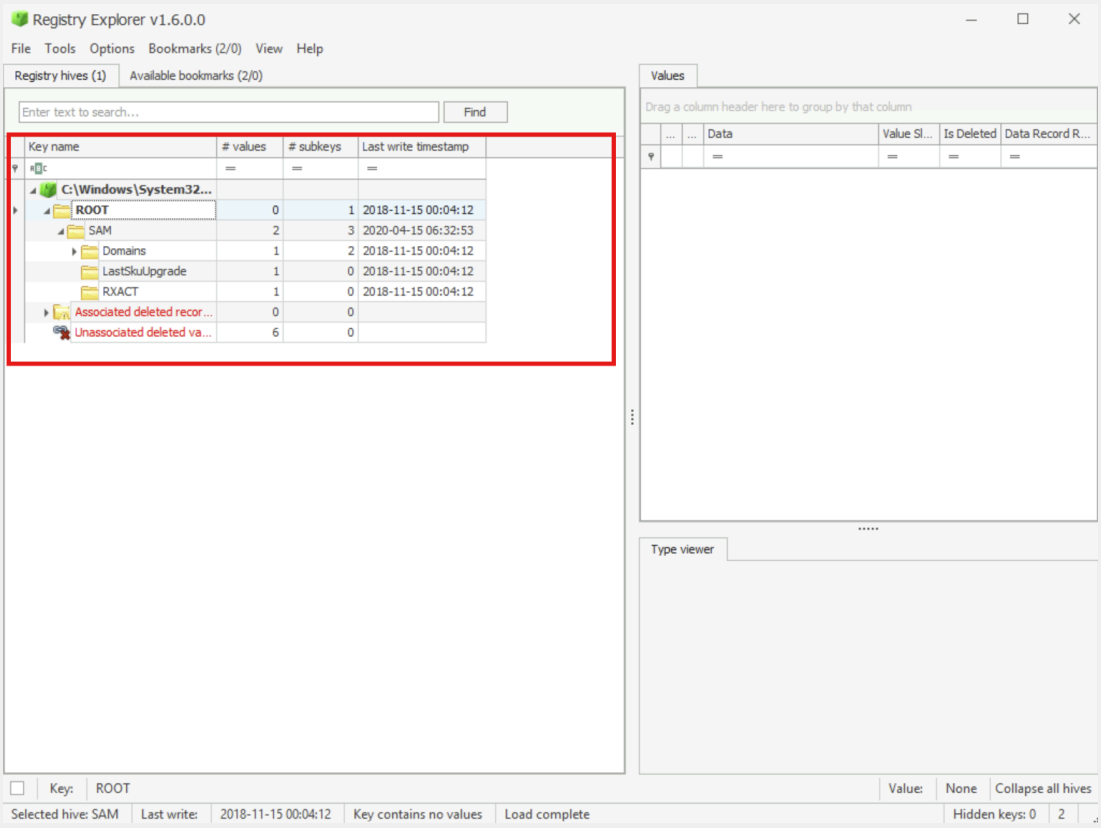
Registry Explorer is an advanced forensic tool designed specifically for analyzing the Windows Registry. Developed by Eric Zimmerman, it provides a robust and user-friendly interface for registry analysis, going beyond the capabilities of the built-in Regedit utility. The tool is widely used in Digital Forensics and Incident Response (DFIR) for detailed and comprehensive investigations. It offers a highly organized and efficient view of the registry's hierarchical structure, and displays hidden and unallocated registry keys that are not accessible via Regedit. This tool allows exporting registry data into

]\

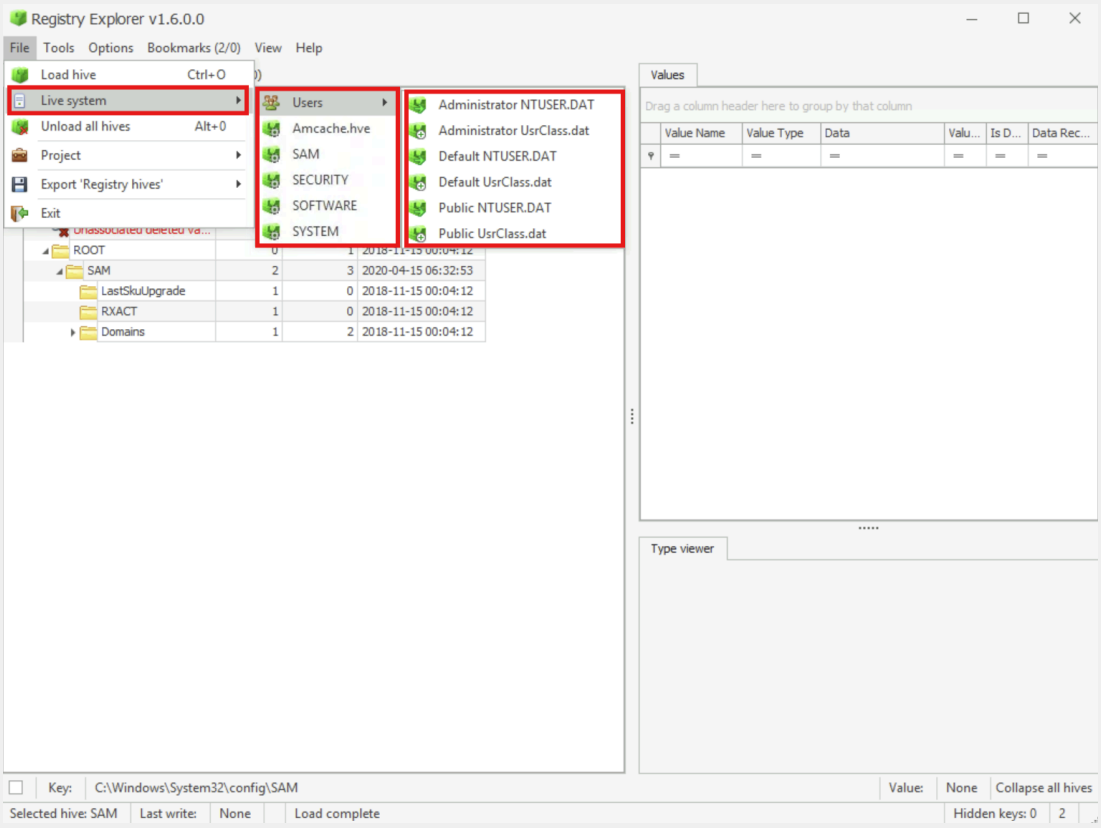
various formats, such as CSV, JSON, or HTML, for reporting or further analysis. Unlike Regedit, Registry Explorer can parse offline hive files, making it ideal for forensic investigations. You need to run the **Registry Explorer** as an **administrator** to analyze the Live Hives.

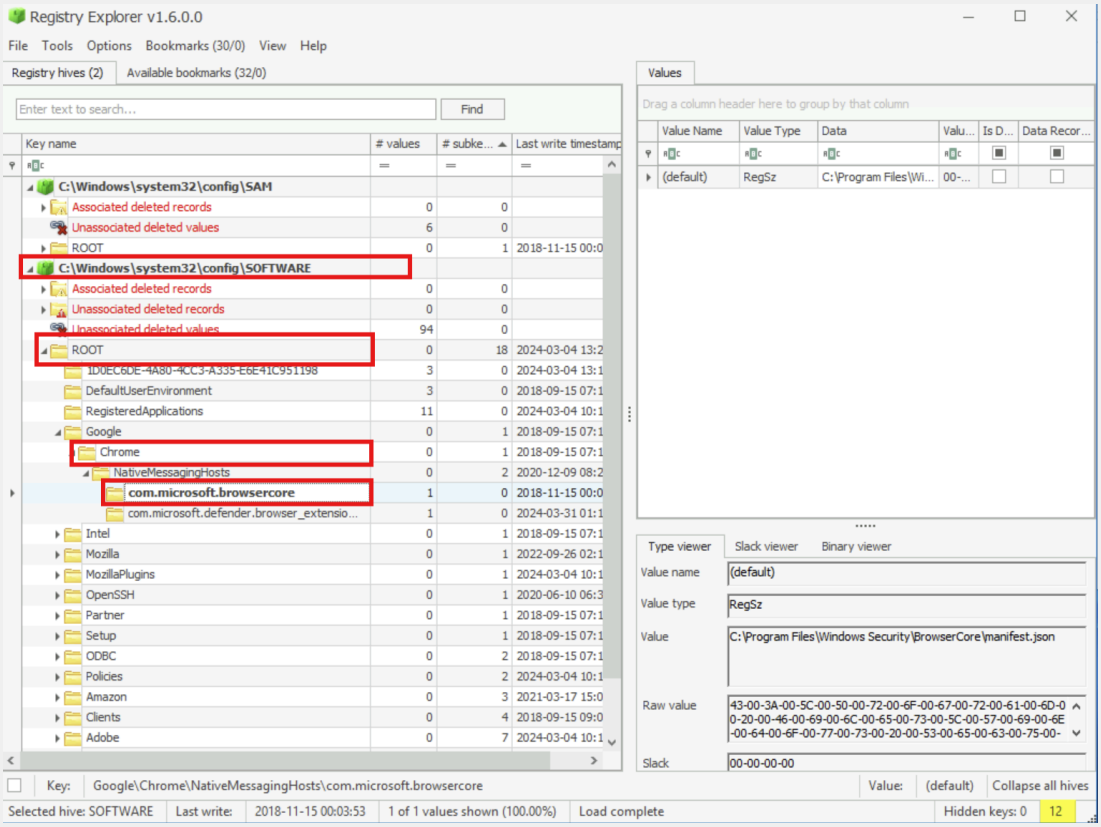



Registry Explorer v1.6.0.0
Select hives to open. Hold SHIFT when clicking OK to automatically replay LOG files
This PC › Local Disk (C:) › Windows › System32 › config ›
Search config
Organize
New folder
EZ tools
Local Disk (C:)
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network
Name
Date modified
Type
Size
RegBack 11/15/2018 12:04 ... File folder
systemprofile 3/4/2024 1:25 PM File folder
TxR 1/13/2021 9:21 PM File folder
BBI 3/31/2024 1:28 AM File 64 KB
BCD-Template 11/14/2018 6:56 AM File 28 KB
COMPONENTS 12/5/2024 6:25 PM File 80,384 KB
DEFAULT 3/31/2024 1:28 AM File 768 KB
DRIVERS 12/5/2024 6:29 PM File 4,156 KB
ELAM 2/19/2024 11:53 AM File 32 KB
SAM 3/31/2024 1:28 AM File 128 KB
SECURITY 3/31/2024 1:28 AM File 32 KB
SOFTWARE 12/5/2024 6:11 PM File 98,560 KB
SYSTEM 3/31/2024 1:28 AM File 18,432 KB
File name:
Open
Cancel
Is Deleted
Data Record R...
Type viewer
Key: None
Value: None
Collapse all hives
Selected hive: NA
Last write: None
Load hives via File menu or drag and drop anywhere above
Hidden keys: 0



We can use **Live Mode** too.


Registry Explorer v1.6.0.0
File Tools Options Bookmarks (2/0) View Help
Load hive Ctrl+O
Live system
Unload all hives Alt+0
Project
Export 'Registry hives'
Exit
Users
Amcache.hve
SAM
SECURITY
SOFTWARE
SYSTEM
Administrator NTUSER.DAT
Administrator UsrClass.dat
Default NTUSER.DAT
Default UsrClass.dat
Public NTUSER.DAT
Public UsrClass.dat
Values
Drag a column header here to group by that column
Value Name
Value Type
Data
Type viewer
ROOT
SAM
LastSkuUpgrade
RXACT
Domains
Key: C:\Windows\System32\config\SAM
Value: None
Collapse all hives
Selected hive: SAM
Last write: None
Load complete
Hidden keys: 0



Analyzing TypedPaths

The **TypedPaths** registry key, located within the **NTUSER.dat** hive, is an essential artifact for forensic analysis of user and adversary behavior on a Windows system. It stores the paths entered into the address bar of **File Explorer** or the **Run** dialog, reflecting the directories or files the user or adversary accessed or searched for during their activities. The live registry location for the **TypedPaths** key is **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\TypedPaths**.

This registry key is dynamically updated as paths are typed, providing a historical record of user interactions with the system's file structure. The **TypedPaths** key contains values representing paths the user has manually entered. Each path is stored as a separate value under the key, typically named in sequence (e.g., **url1**, **url2**, etc.), reflecting the order in which they were accessed. This provides a chronological view of user or adversary activity.

During an investigation, the TypedPaths key can be examined to identify potential signs of suspicious activity. The **TypedPaths** key has significant forensic relevance because it helps reconstruct:

User Workflow:

- Identifies the files, directories, or network locations accessed by the user.
- Provides insight into routine activities, such as accessing shared drives or frequently used directories.

Adversary Activities:

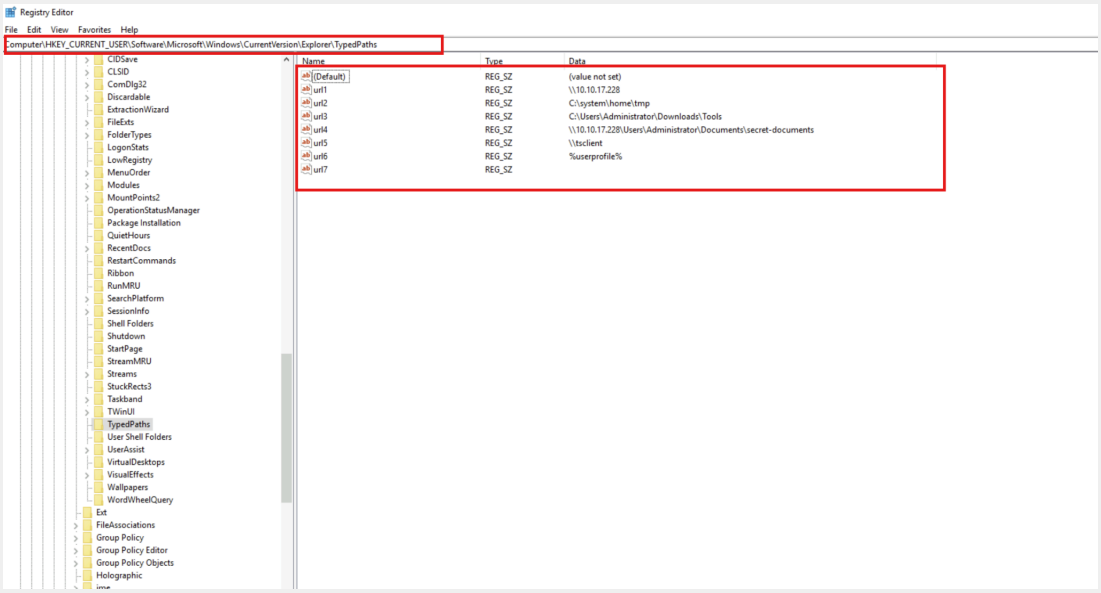
- If the system is compromised, the key can reveal the adversary's exploration of the victim's file system.
- Highlights specific directories or files targeted during an intrusion.

Context for Investigations:

- Shows intent by revealing what the user or adversary was searching for or accessing.
- Correlates with other artifacts, such as recently opened documents or logs, to create a detailed timeline.

Evidence Recovery:

- Provides leads for recovering deleted files or uncovering the presence of suspicious directories.

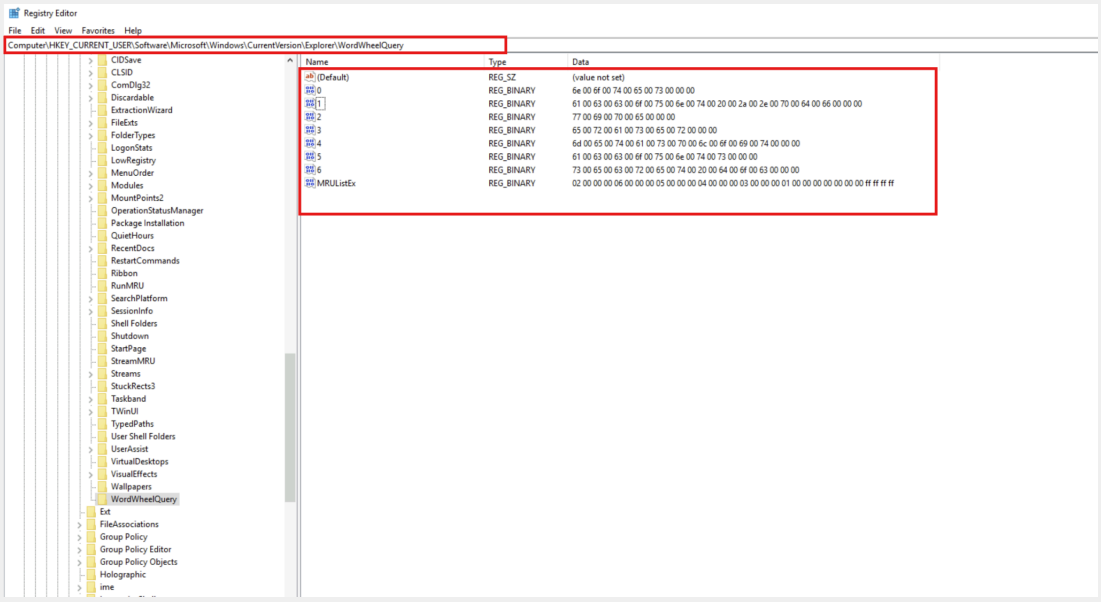


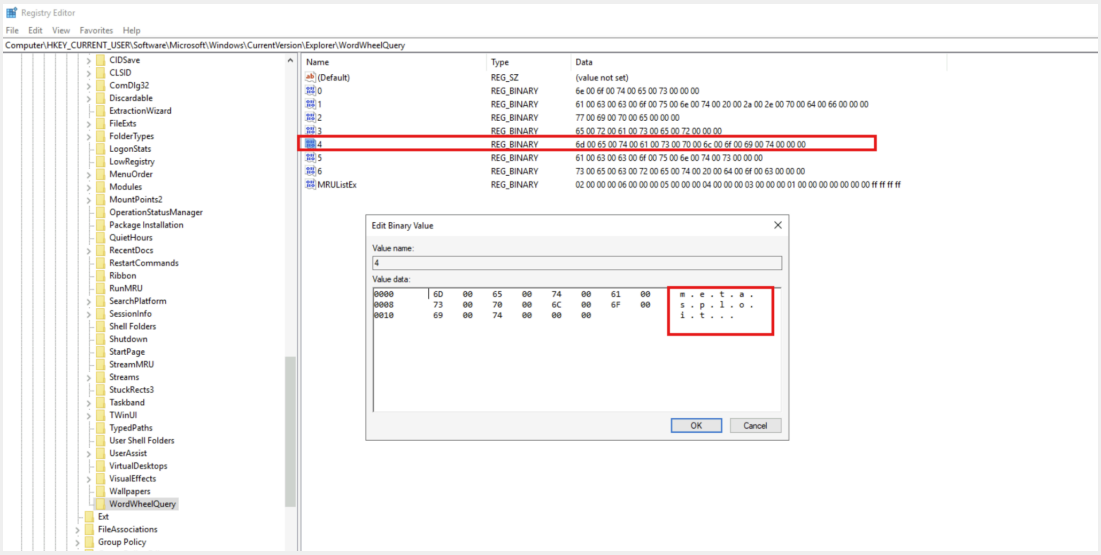
Analyzing WordWheel Query & Search History

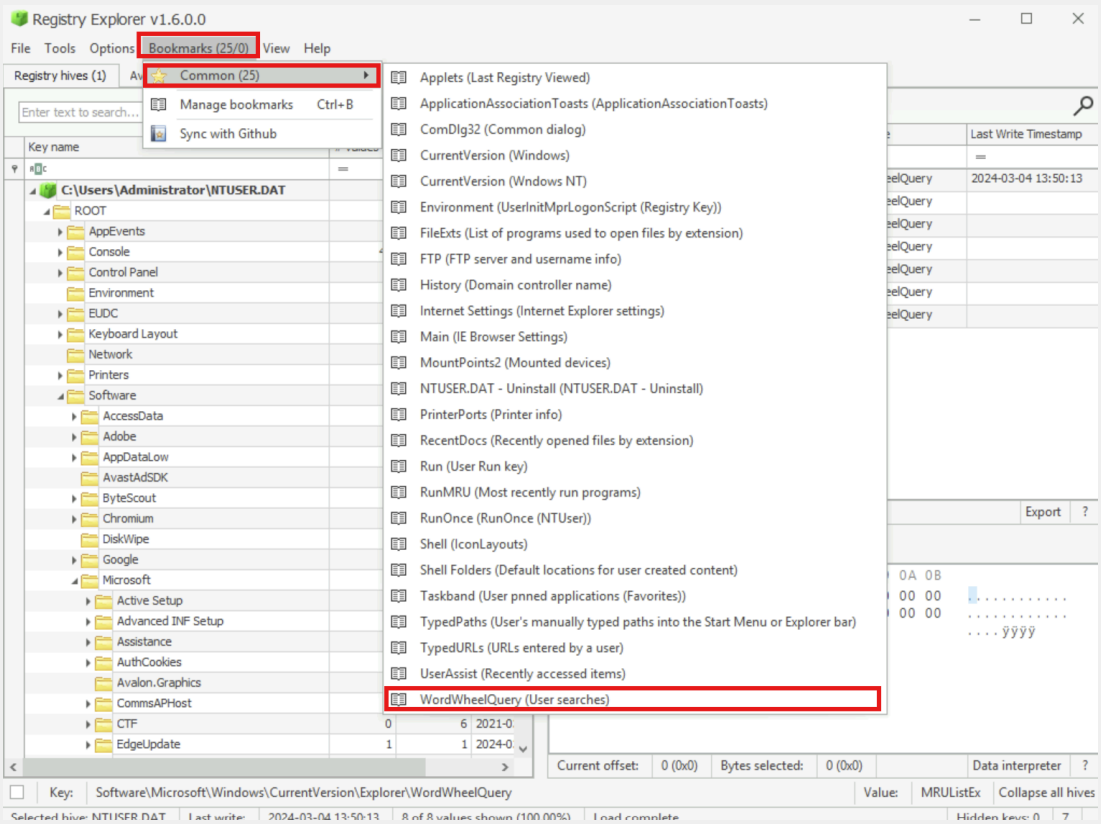
The **WordWheel Query** registry key is a significant artifact for reconstructing user search behavior on Windows systems. It records terms entered into the **Windows Search Bar** and other search fields, offering a detailed view of what the user searched for within the operating system. This information can provide critical context in forensic investigations, especially for understanding user intent or adversary actions. The WordWheel Query key resides in the following live registry location **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery**.

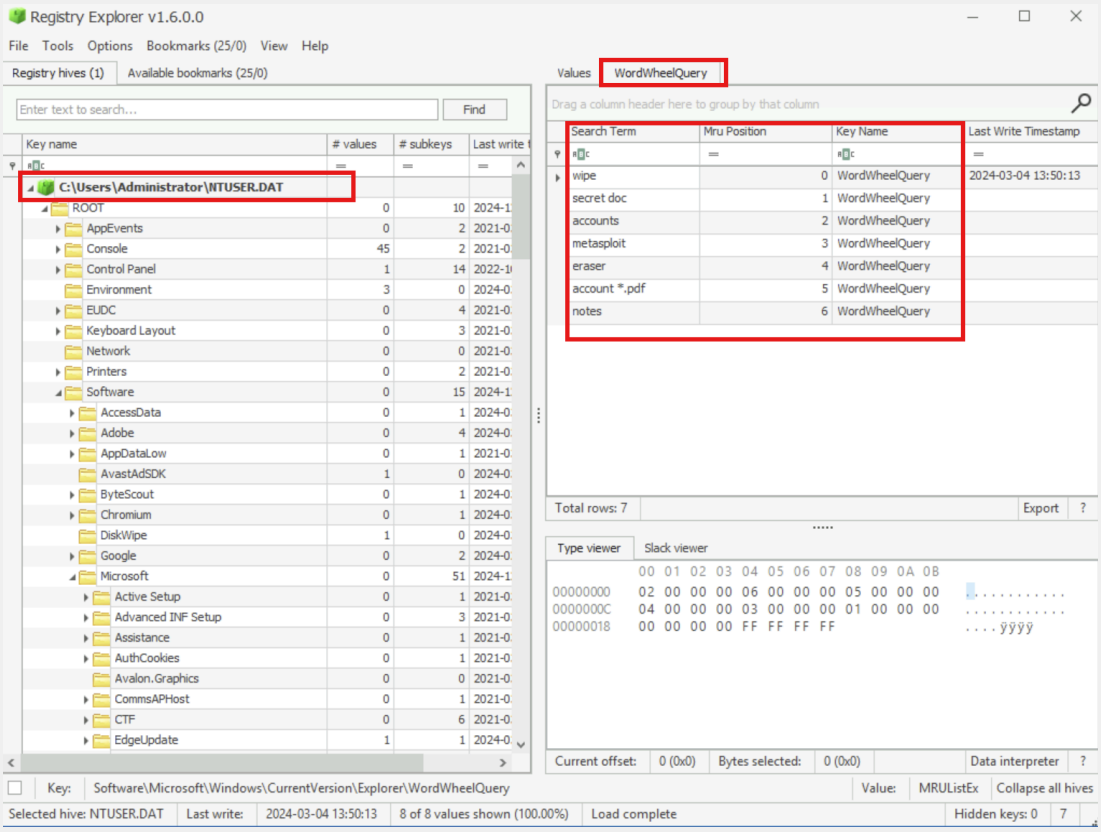
This key is updated dynamically whenever a user performs a search, making it a reliable source for tracking recent activity. Each search term is saved as a value under this key, typically assigned sequential names such as **0**, **1**, **2**, and so on.

This history provides insight into user behavior and preferences, often revealing the focus of their recent activities.




Registry Editor
File Edit View Favorites Help
Computer\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery
Name
Type
Data
(Default)
REG_SZ
(value not set)
0
REG_BINARY
6e 00 6f 00 74 00 65 00 73 00 00 00
1
REG_BINARY
61 00 63 00 63 00 6f 00 75 00 6e 00 74 00 20 00 2a 00 2e 00 70 00 64 00 66 00 00 00
2
REG_BINARY
77 00 69 00 70 00 65 00 00 00
3
REG_BINARY
65 00 72 00 61 00 73 00 65 00 72 00 00 00
4
REG_BINARY
6d 00 65 00 74 00 61 00 73 00 70 00 6c 00 6f 00 69 00 74 00 00 00
5
REG_BINARY
61 00 63 00 63 00 6f 00 75 00 6e 00 74 00 73 00 00 00
6
REG_BINARY
73 00 65 00 63 00 72 00 65 00 74 00 20 00 64 00 6f 00 63 00 00 00
MRUListEx
REG_BINARY
02 00 00 00 06 00 00 00 05 00 00 00 04 00 00 00 03 00 00 00 01 00 00 00 00 00 00 00 ff ff ff ff
Edit Binary Value
Value name:
4
Value data:
0000 6D 00 65 00 74 00 61 00
0008 73 00 70 00 6C 00 6F 00
0010 69 00 74 00 00 00
m . e . t . a .
s . p . l . o .
i . t . . .
OK
Cancel


Registry Explorer v1.6.0.0
File Tools Options Bookmarks (25/0) View Help
Registry hives (1)
Common (25)
Manage bookmarks Ctrl+B
Sync with Github
Applets (Last Registry Viewed)
ApplicationAssociationToasts (ApplicationAssociationToasts)
ComDlg32 (Common dialog)
CurrentVersion (Windows)
CurrentVersion (Windows NT)
Environment (UserInitMprLogonScript (Registry Key))
FileExts (List of programs used to open files by extension)
FTP (FTP server and username info)
History (Domain controller name)
Internet Settings (Internet Explorer settings)
Main (IE Browser Settings)
MountPoints2 (Mounted devices)
NTUSER.DAT - Uninstall (NTUSER.DAT - Uninstall)
PrinterPorts (Printer info)
RecentDocs (Recently opened files by extension)
Run (User Run key)
RunMRU (Most recently run programs)
RunOnce (RunOnce (NTUser))
Shell (IconLayouts)
Shell Folders (Default locations for user created content)
Taskband (User pnned applications (Favorites))
TypedPaths (User's manually typed paths into the Start Menu or Explorer bar)
TypedURLs (URLs entered by a user)
UserAssist (Recently accessed items)
WordWheelQuery (User searches)
Enter text to search...
Key name
Last Write Timestamp
C:\Users\Administrator\NTUSER.DAT
ROOT
AppEvents
Console
Control Panel
Environment
EUDC
Keyboard Layout
Network
Printers
Software
AccessData
Adobe
AppDataLow
AvastAdSDK
ByteScout
Chromium
DiskWipe
Google
Microsoft
Active Setup
Advanced INF Setup
Assistance
AuthCookies
Avalon.Graphics
CommsAPHost
CTF
EdgeUpdate
2024-03-04 13:50:13
Export
Current offset: 0 (0x0)
Bytes selected: 0 (0x0)
Data interpreter
Key: Software\Microsoft\Windows\CurrentVersion\Explorer\WordWheelQuery
Value: MRUListEx
Collapse all hives



The **WordWheel Query** key is a powerful resource for forensic analysis, offering a window into the user's search activities on the system.

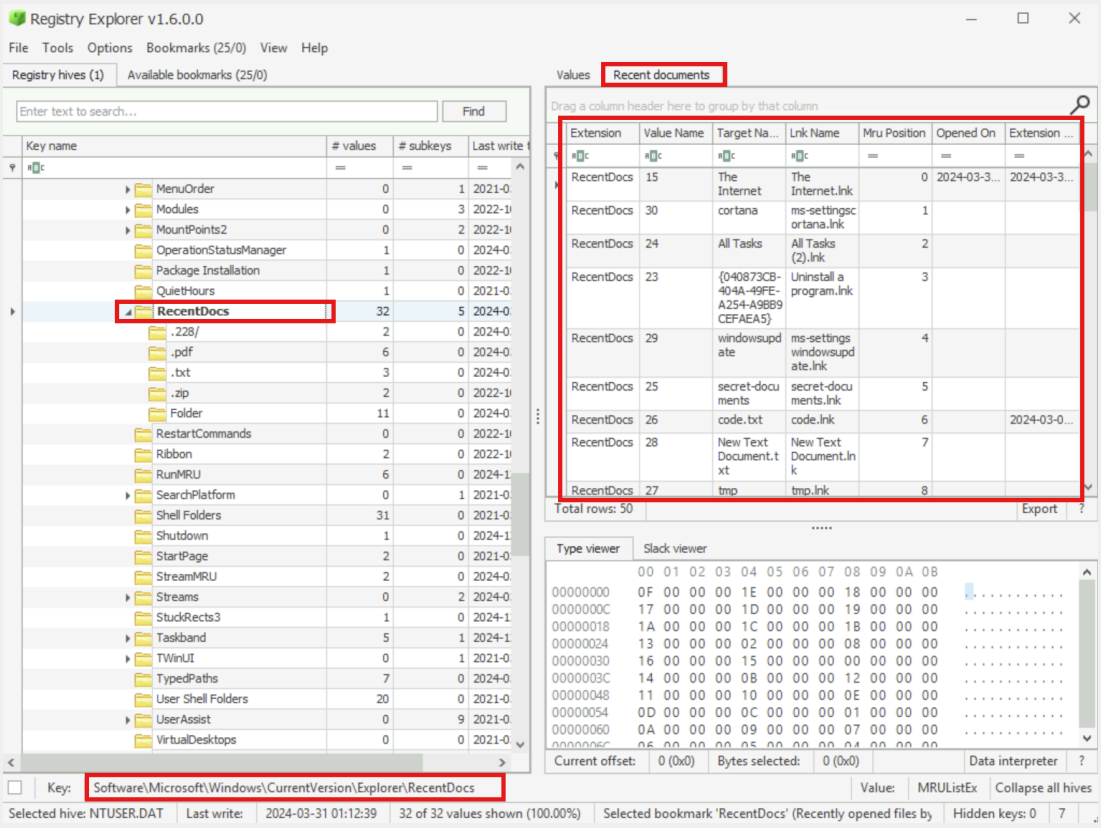
Analyzing RecentDocs & Document Access Tracking

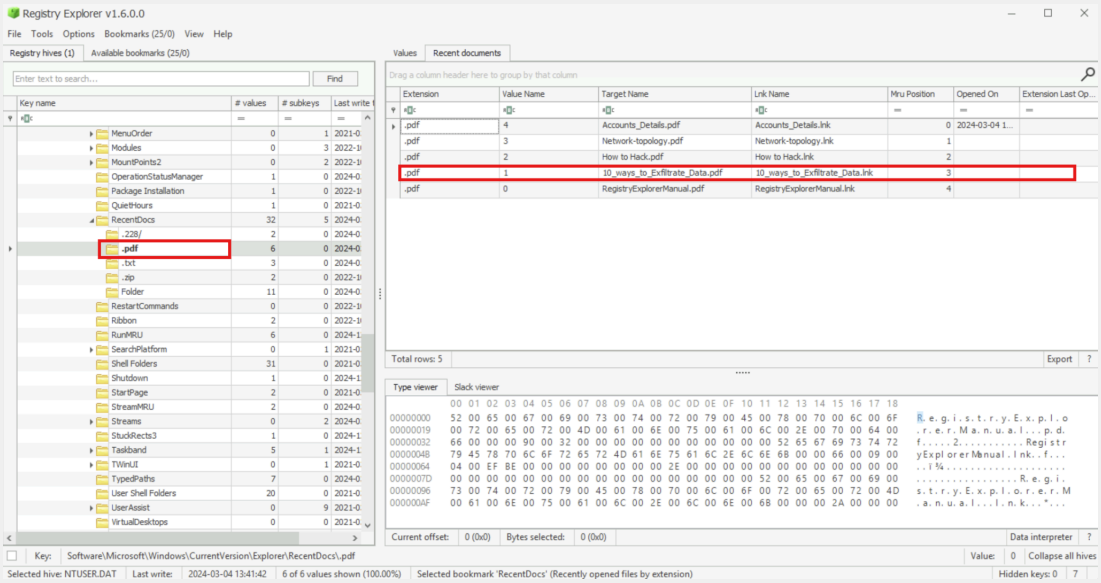
The **RecentDocs** registry key is a vital artifact in Windows forensic investigations. It records information about recently accessed documents and files, providing insight into user activity, workflow, and potential adversary actions. By analyzing the **RecentDocs** key, forensic investigators can identify which files were opened, the sequence of access, and the types of files that were most frequently used. The RecentDocs key is stored in the following registry location **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs**.

This key organizes entries based on file extensions, enabling a categorized view of document access. The RecentDocs key contains subkeys for different file types based on their extensions (e.g., **.docx**, **.pdf**, **.jpg**). It also

includes a general subkey for all accessed files, irrespective of their type. Each entry in the subkeys stores the names of files accessed recently. Binary values may indicate the path of the files accessed, their order of access, and timestamps.

Entries are updated dynamically when a user opens a file. The system records the most recent files in the order of their access.






```
PS C:\Users\Administrator> Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs" | Format-List

MRUListEx      : {32, 0, 0, 0...}
3              : {69, 0, 90, 0...}
4              : {82, 0, 101, 0...}
5              : {84, 0, 111, 0...}
6              : {68, 0, 111, 0...}
7              : {104, 0, 97, 0...}
9              : {83, 0, 65, 0...}
10             : {65, 0, 114, 0...}
1              : {68, 0, 111, 0...}
12             : {82, 0, 101, 0...}
13             : {82, 0, 101, 0...}
14             : {49, 0, 48, 0...}
16             : {49, 0, 48, 0...}
17             : {72, 0, 111, 0...}
18             : {78, 0, 101, 0...}
11             : {105, 0, 109, 0...}
20             : {123, 0, 54, 0...}
0              : {99, 0, 111, 0...}
21             : {78, 0, 101, 0...}
22             : {76, 0, 111, 0...}
8              : {67, 0, 58, 0...}
2              : {84, 0, 104, 0...}
19             : {65, 0, 99, 0...}
27             : {116, 0, 109, 0...}
28             : {78, 0, 101, 0...}
26             : {99, 0, 111, 0...}
25             : {115, 0, 101, 0...}
```

]\

```
29           : {119, 0, 105, 0...}
23           : {123, 0, 48, 0...}
24           : {65, 0, 108, 0...}
30           : {99, 0, 111, 0...}
15           : {84, 0, 104, 0...}
31           : {78, 0, 84, 0...}
32           : {65, 0, 100, 0...}
PSPath       :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\Recent
Docs
PSParentPath :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
PSChildName  : RecentDocs
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```

```
PS C:\Users\Administrator> $recentDocsPath =
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\RecentDocs"
PS C:\Users\Administrator> $recentDocs = Get-ItemProperty -Path $recentDocsPath
PS C:\Users\Administrator> foreach ($property in $recentDocs.PSObject.Properties) {
>>     # Skip the MRUListEx property, process only numbered entries
>>     if ($property.Name -match '^\d+$') {
>>         # Decode the binary data into readable text
>>         $decodedData = [System.Text.Encoding]::Unicode.GetString($property.Value)
>>
>>         # Output the decoded data
>>         Write-Output "Entry $($property.Name): $decodedData"
>>     } elseif ($property.Name -eq "MRUListEx") {
>>         Write-Output "MRUListEx (Order of Access): $($property.Value -join ', ')"
>>     }
>> }

MRUListEx (Order of Access): 32, 0, 0, 0, 31, 0, 0, 0, 15, 0, 0, 0, 30, 0, 0, 0, 24, 0, 0, 0, 23, 0, 0, 0, 29,
0, 0, 0, 25, 0, 0, 0, 26, 0, 0, 0, 28, 0, 0, 0, 27, 0, 0, 0, 19, 0, 0, 0, 2, 0, 0, 0, 8, 0, 0, 0, 22, 0, 0, 0,
21, 0, 0, 0, 0, 0, 0, 0, 20, 0, 0, 0, 11, 0, 0, 0, 18, 0, 0, 0, 17, 0, 0, 0, 16, 0, 0, 0, 14, 0, 0, 0, 13, 0,
0, 0, 12, 0, 0, 0, 1, 0, 0, 0, 10, 0, 0, 0, 9, 0, 0, 0, 7, 0, 0, 0, 6, 0, 0, 0, 5, 0, 0, 0, 4, 0, 0, 0, 3, 0,
0, 0, 255, 255, 255, 255
Entry 3: EZ tools f2     婅琠潯獬氮歮 J 쀀     .                EZ tools.lnk
Entry 4: RegRipper3.0-master.zip †2     慒剧灩数da· ○涯猲整⍰槃k` 쀀     .                RegRipper3.0-master.lnk &
Entry 5: Tools \2     潔汯⍰槃kD 쀀     .                Tools.lnk
Entry 6: Downloads h2     潫濤潭慨⍰槃kL 쀀     .                Downloads.lnk
Entry 7: hacking-tools t2     慨正潼⍰獡汯⍰槃kT  쀀     .                hacking-tools.lnk
Entry 9: SAM V2     䅓⍰槃k@     쀀     .                SAM.lnk
Entry 10: Artifacts h2     牁楴慳瑯⍰槃kL       쀀     .                Artifacts.lnk
Entry 1: Documents h2     潫畣敭瑮⍰槃kL 쀀     .                Documents.lnk
Entry 12: RegistryExplorerManual.pdf  2     慒楧瑳祲硅汰牯牥慍畮污氮歮 f        쀀     .
RegistryExplorerManual.lnk *
Entry 13: RegistryExplorer ~2     慒楧瑳祲硅汰牯牥氮歮 Z        쀀     .                RegistryExplorer.lnk $
Entry 14: 10.10.17.228/ †2     瑨灴⍰〈⍰〱ㄠ⸷㈲8k`     쀀     .                http--10.10.17.228-.lnk &
Entry 16: 10_ways_to_Exfiltrate_Data.pdf œ2     〱睟祡彳潴䕟晸汩牴瑡彥慄慴氮歮 n        쀀     .
```

]\

```
10_ways_to_Exfiltrate_Data.lnk .
Entry 17: How to Hack.pdf n2     潔ᵓ豬覺捡⍰棃kP 螋     .              How to Hack.lnk
Entry 18: Network-topology.pdf ~2     敎暉牯⍰豬潰湋祧氮歮 Z     螋     .              Network-topology.lnk $
Entry 11: important_documents †2     浩潰瑲湡彴潤畣敭瑮⍰棃k`     螋     .              important_documents.lnk &
Entry 20: {63A7F0F7-6ACD-4D19-92FE-FB4BD9D35BA6} †2     桃湡敧慍捯湵⁴祴数⍰棃k`  螋     .              Change
account type.lnk &
Entry 0: cortana „2     獭猭瑥楴杮潣瑲湡氮歮 ^          螋     .              ms-settingscortana.lnk &
Entry 21: New folder 12     敎ᵓ潦摬牥氮歮 N     螋     .              New folder.lnk
Entry 22: Local Disk (C:) „2     潌慣楄歳⠠⤩⸲氮歮 ^     螋     .              Local Disk (C) (2).lnk &
Entry 8: C:\ x2     潌慣楄歳⠠⤩氮歮 V     螋     .              Local Disk (C).lnk "
Entry 2: This PC b2     桔獩倠⍰棃kH     螋     .              This PC.lnk
Entry 19: Accounts_Details.pdf ~2     捁潣湵獴䑟瑥楡獬氮歮 Z     螋     .              Accounts_Details.lnk $
Entry 27: tmp V2     浴⍰棃k@     螋     .              tmp.lnk
Entry 28: New Text Document.txt  2     敎ᵓ敔瑸䐠捯浵湥⍰棃k\     螋     .              New Text Document.lnk $
Entry 26: code.txt Z2     潣敤氮歮 B     螋     .              code.lnk
Entry 25: secret-documents ~2     敳牣瑥搭捯浵湥獴氮歮 Z     螋     .              secret-documents.lnk $
Entry 29: windowsupdate -2     獭猭瑥楴杮睳湩潤獷灵慤整氮歮 j     螋     .              ms-settingswindowsupdate.lnk
,
Entry 23: {040873CB-404A-49FE-A254-A9BB9CEFAEA5} †2     湕湩瑳污⁡牰杯慲⍰棃k`     螋     .              Uninstall a
program.lnk &
Entry 24: All Tasks t2     汁慔歳⍰⠲⤩棃kT     螋     .              All Tasks (2).lnk
Entry 30: cortana „2     獭猭瑥楴杮潣瑲湡氮歮 ^     螋     .              ms-settingscortana.lnk &
Entry 15: The Internet r2     桔湉整湲瑥氮歮 R 螋     .              The Internet.lnk
Entry 31: NTUSER.DAT 12     呎单刭䅄呔氮歮 N     螋     .              NTUSER.DAT.lnk
Entry 32: Administrator t2     摁業楮瑳慲潴⍰棃kT     螋     .              Administrator.lnk
```

Analyzing Common Dialog Box & Activity

The **Comdlg32** registry key is a critical artifact in Windows forensics, as it records user interactions with common dialog boxes, such as **File Open** or **File Save As** dialogs. This artifact can provide valuable insights into files accessed, saved, or interacted with by the user or an adversary. The Comdlg32 key is found in the following registry location, **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32**.

This key typically contains two important subkeys:

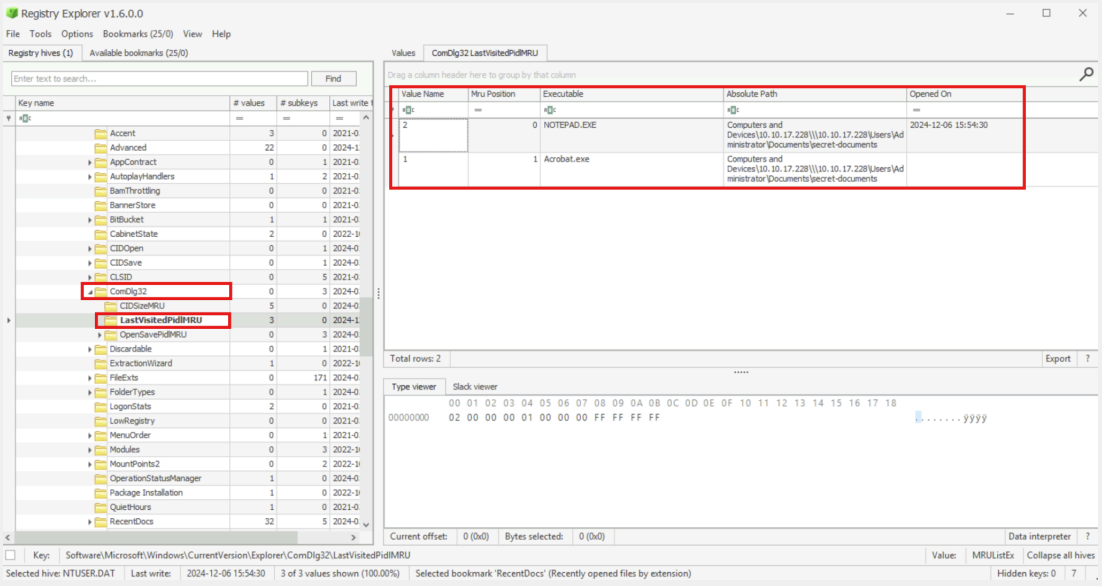
LastVisitedPidlMRU:

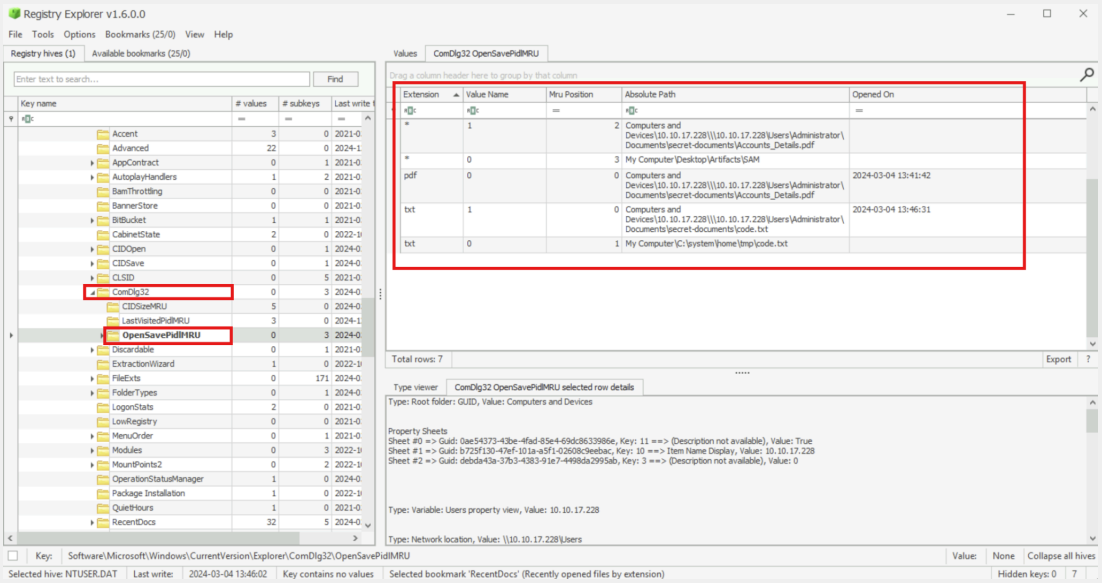
- Stores the paths of directories that were recently accessed via common dialog boxes.
- Tracks directories the user navigated to while opening or saving files.
- Useful for identifying file system exploration patterns.

OpenSavePidlMRU:

- Maintains a record of recently accessed files, grouped by file extensions (e.g., .docx, .pdf, .jpg).

- Entries include metadata about the files accessed or saved through dialog boxes.
- Helps pinpoint specific files of interest to the user or adversary.






```
PS C:\Users\Administrator> Get-ItemProperty -Path
"HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg32\LastVisitedPidlMRU" | Format-List

MRUListEx     : {0, 0, 0, 0...}
1             : {65, 0, 99, 0...}
```

```
2           : {78, 0, 79, 0...}
0           : {82, 0, 101, 0...}
PSPath      :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg
32\LastVisitedPidlMRU
PSParentPath :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\ComDlg
32
PSChildName  : LastVisitedPidlMRU
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Analysing User Assist & Tracking Program Execution

The **UserAssist** registry key is a powerful forensic artifact in Windows that logs information about applications executed by a user. It is designed to enhance the user experience by tracking frequently used programs, but for forensic investigators, it provides invaluable insights into user activity, including application launches, frequency, and timestamps. The UserAssist key is located in the following paths in the Windows Registry **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist**.

Within this key, there are unique **GUID** subkeys that contain encoded information about application usage. Each **GUID** subkey corresponds to a category of activity, such as desktop programs or shortcuts. For instance, **{75048700-EF1F-11D0-9888-006097DEACF9}** tracks programs executed through Windows Explorer. Entries within the **GUID** subkeys are encoded using the **ROT13** cipher (a basic letter substitution). The entries include application paths, execution counts, and the last run timestamp. The timestamp is stored in Windows **FILETIME** format, which requires decoding for interpretation.

Below are some commonly encountered **GUID**s along with their meanings:

{75048700-EF1F-11D0-9888-006097DEACF9}

- Description: Tracks programs and shortcuts executed through the Start Menu or Windows Explorer.
- Significance: This GUID records application usage and provides valuable insight into a user's program execution habits.

{CEBFF5CD-ACE2-4F4F-9178-9926F41749EA}

- Description: Tracks user interactions with shortcuts or files pinned to the Taskbar or Start Menu.

- Significance: Useful for determining frequently accessed files or applications.

{F4E57C4B-2036-11D1-9953-00C04FD919C1}
- Description: Logs details about applications executed via Run Dialog or shortcuts in Windows Explorer.
- Significance: Indicates ad-hoc program executions or direct application launches.

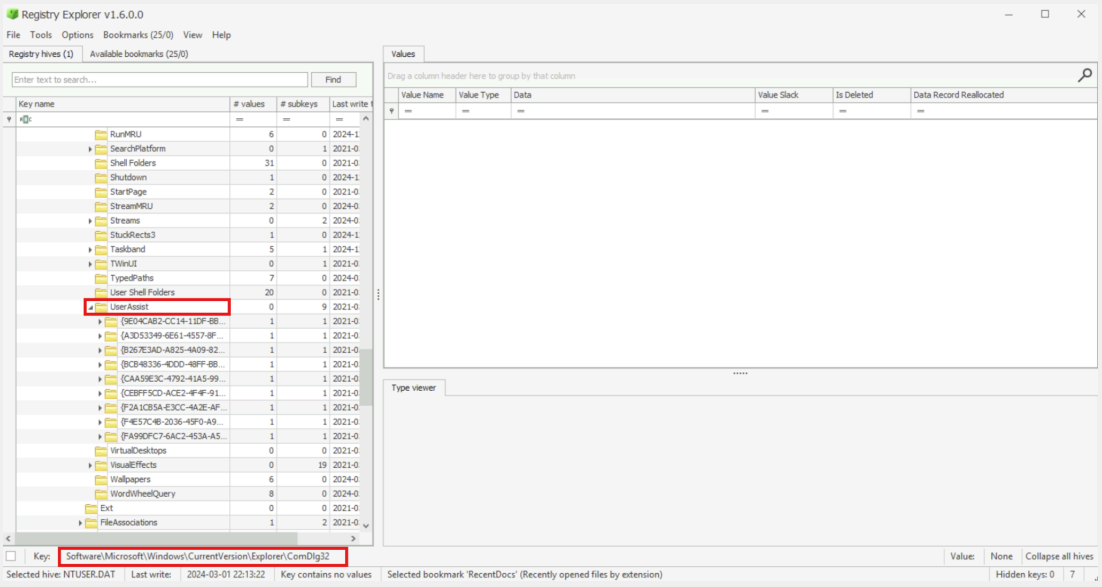
{5E6AB780-7743-11CF-A12B-00AA004AE837}
- Description: Tracks Internet Explorer history, including URLs and web applications opened.
- Significance: Key for understanding web activity in legacy systems.

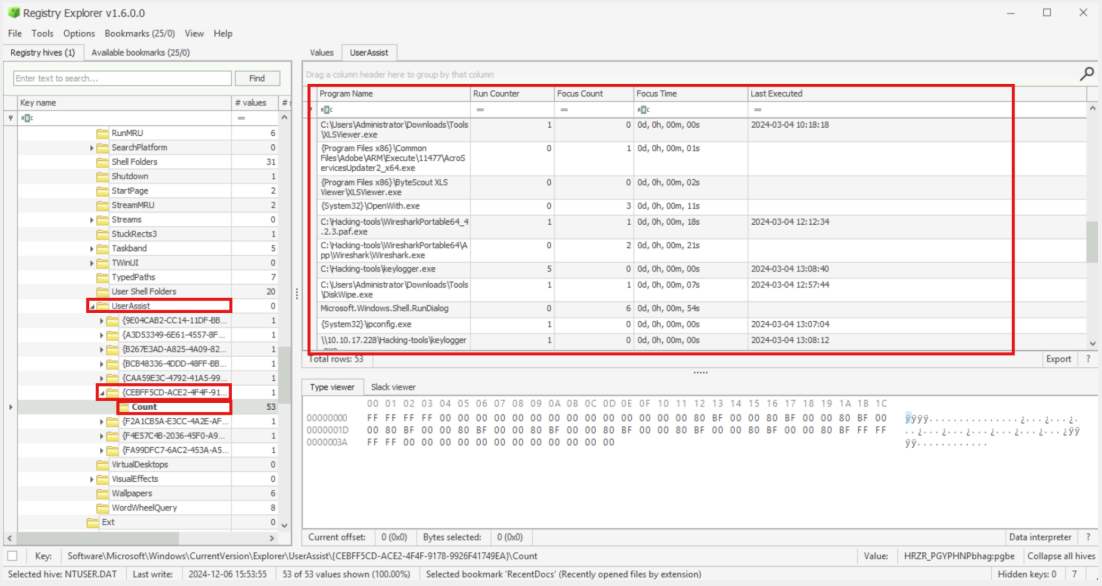
{9E04CAB2-CC14-11DF-BB8C-A2F1DED72085}
- Description: Tracks recently opened folders in the File Explorer.
- Significance: Indicates user navigation patterns, revealing directories of interest.

{1B4B7C2A-0003-4F53-91F5-065F8404D01C}
- Description: Tracks games played on the system (in earlier versions of Windows).
- Significance: Highlights recreational or suspicious activity tied to gaming applications.

{8983036C-27C0-404B-8F08-102D10DCFD74}
- Description: Tracks UWP (Universal Windows Platform) applications executed on Windows 8 and later.
- Significance: Provides insight into modern application usage.

{BCB48336-4DDD-48FF-BB0B-D3190DACB3E2}
- Description: Tracks control panel items accessed by the user.
- Significance: Useful for identifying administrative actions or configuration changes.





Analyzing RunMRU & Run Dialog Box

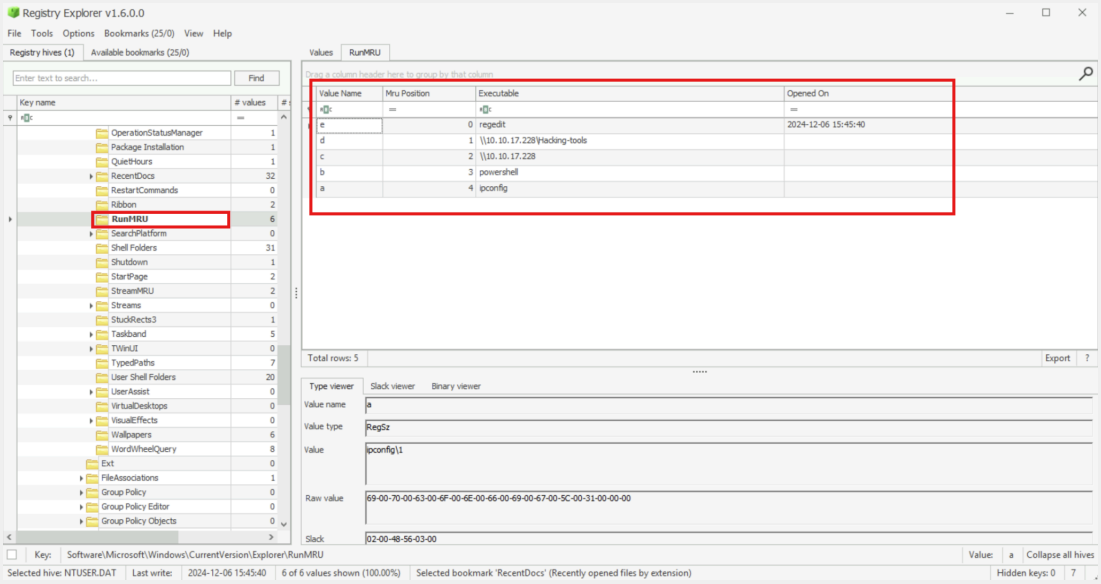
The **RunMRU** registry key stores entries typed into the Run dialog box, accessible via the **Win+R** shortcut or the **Start** menu's search bar. This key can provide valuable forensic insights into the commands and paths executed by a user, revealing their interaction with the system. The **RunMRU** key is designed to store a list of commands, file paths, or application names

entered into the **Run** dialog box. It acts as a "**most recently used**" list, aiding users by auto-suggesting previously entered commands. The entries are stored as individual values under the **RunMRU** key. Each value is named with an alphabetical label (e.g., **a**, **b**, **c**), and its data contains the command or path entered. A special value named **MRUList** maintains the order in which these entries were made, providing chronological context.

Commands stored in **RunMRU** can reveal:

- File paths accessed.
- Applications executed.
- Administrative commands used, such as **cmd**, **regedit**, or **msconfig**.

Entries in **RunMRU** can validate activities found in other logs or registry keys, such as **UserAssist** or **TypedPaths**. It is in **HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU**.




```
PS C:\Users\Administrator> Get-ItemProperty -Path
'HKCU:\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU'

a              : ipconfig\1
MRUList        : edcba
b              : powershell\1
c              : \\10.10.17.228\1
d              : \\10.10.17.228\Hacking-tools\1
```

```
e            : regedit\1
PSPath       :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer\RunMRU
PSParentPath :
Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Explorer
PSChildName  : RunMRU
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
```

Analyzing Shell Bag Artifacts

From a forensic perspective, shell bag artifacts serve as critical pieces of evidence, acting as silent witnesses to a user's interactions with the file system. These artifacts provide valuable insights into folder activities, even if files or folders have been deleted or attempts have been made to obscure activity. Shell bag artifacts can reveal not only which folders were accessed but also the methods of access, timestamps, and even folder-specific configurations such as view settings. This makes them an indispensable resource in digital investigations, aiding in reconstructing user behavior and identifying potential malicious activities.

The following table summarizes the key registry locations and files where shell bag artifacts are stored, along with a brief description of their forensic significance:

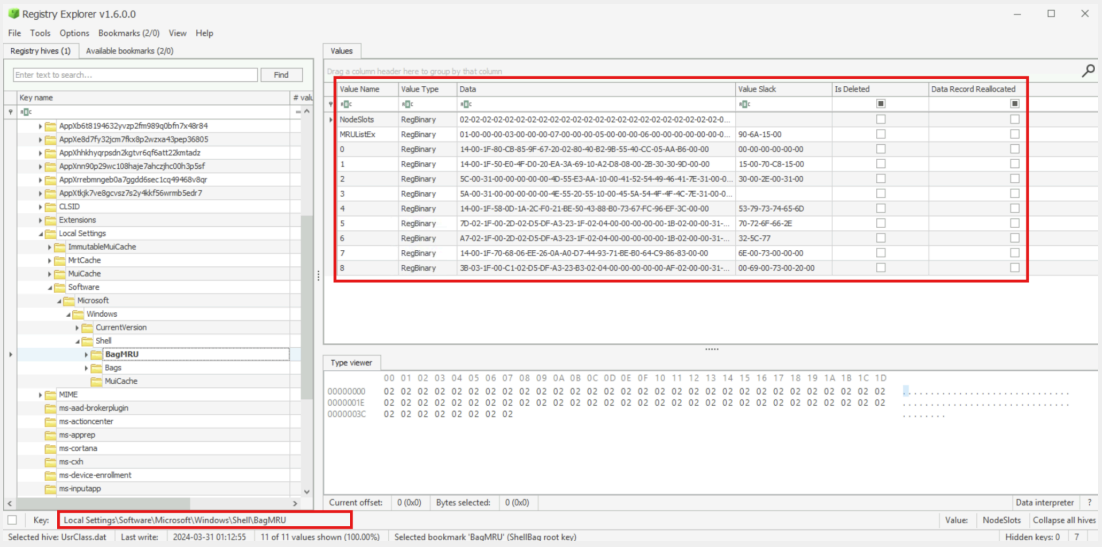
NTUSER.DAT: HKCU\Software\Microsoft\Windows\Shell\BagMRU
Stores hierarchical information about accessed folders.
Helps reconstruct folder structures traversed by the user.

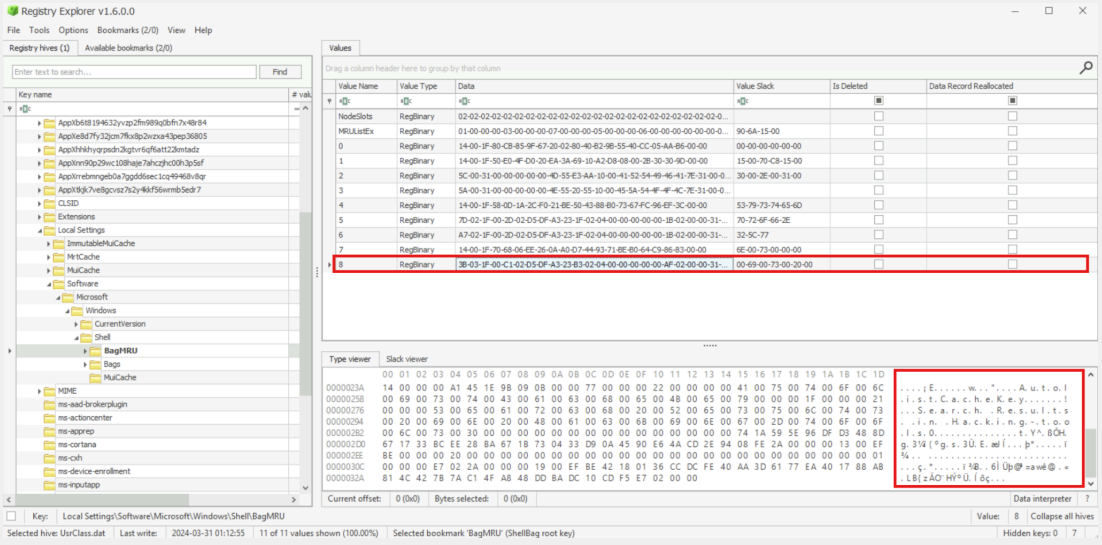
NTUSER.DAT: HKCU\Software\Microsoft\Windows\Shell\Bags
Contains metadata related to folder-specific settings, such as view modes, icon sizes, and sort orders.

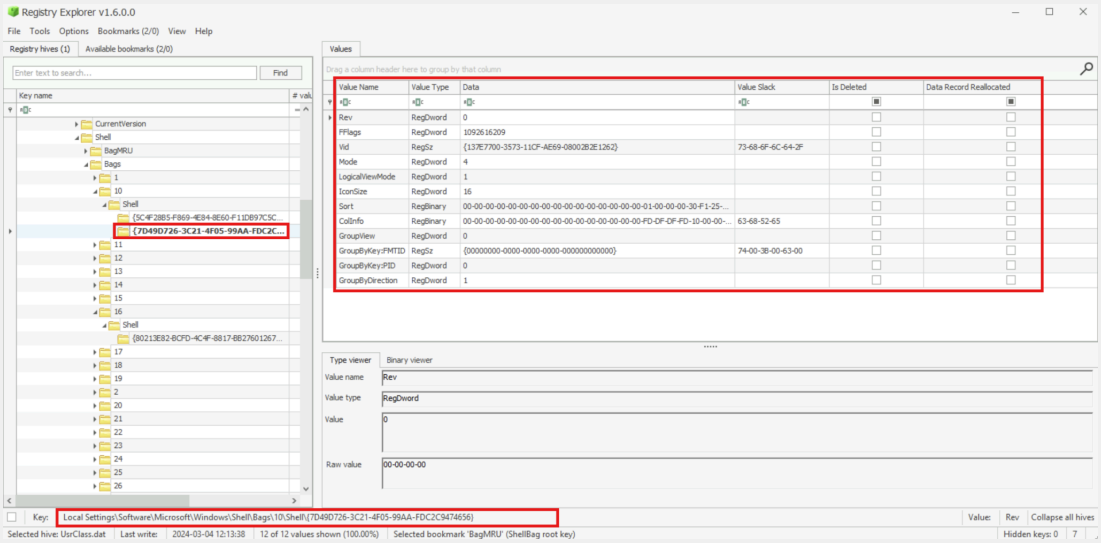
USRCLASS.DAT: HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\BagMRU
Similar to the BagMRU key in NTUSER.DAT but tracks folders accessed through certain applications.

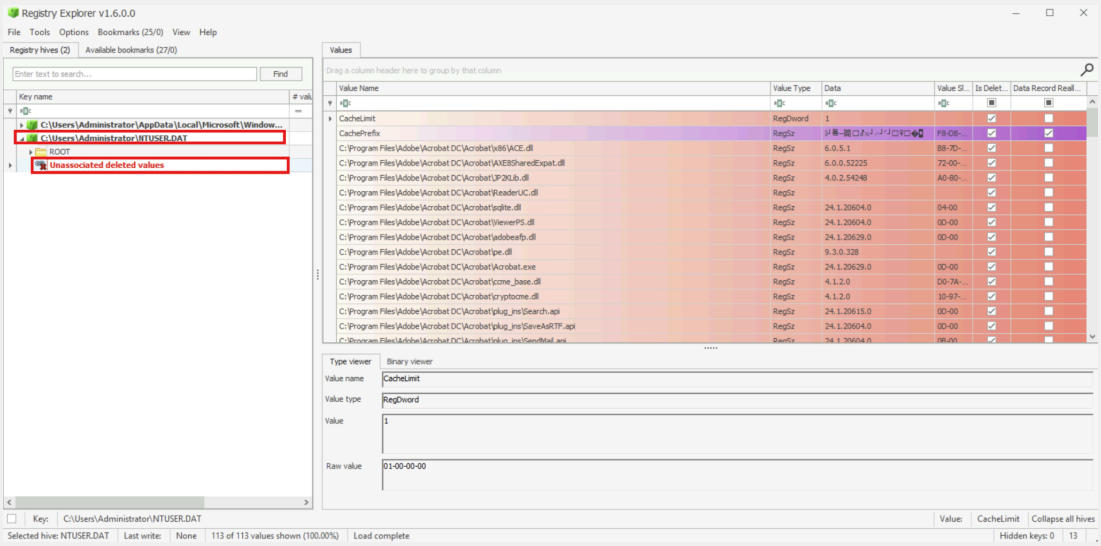
USRCLASS.DAT: HKCU\Software\Classes\Local Settings\Software\Microsoft\Windows\Shell\Bags
Complements the BagMRU key by maintaining additional metadata for folders accessed via applications.

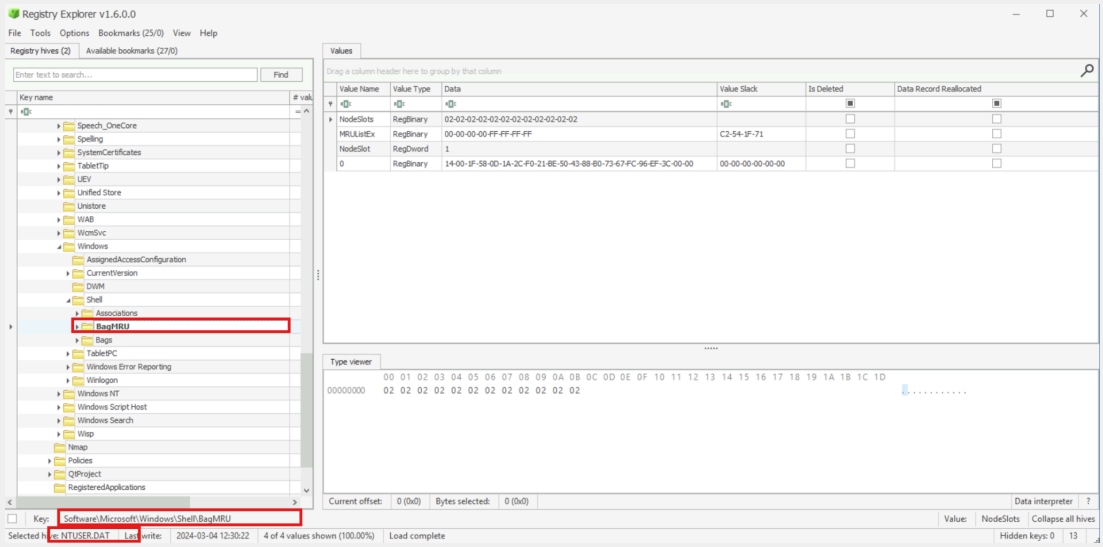
Even if a folder is deleted, its metadata often remains in the shell bag records, enabling investigators to uncover previously accessed or hidden directories. Timestamp data associated with shell bags can help establish a sequence of events, critical in identifying when certain activities occurred. Folder view settings and navigation patterns can provide context about a user's intentions and habits. When combined with other artifacts, such as RecentDocs or UserAssist, shell bags can substantiate findings or reveal inconsistencies in user claims.



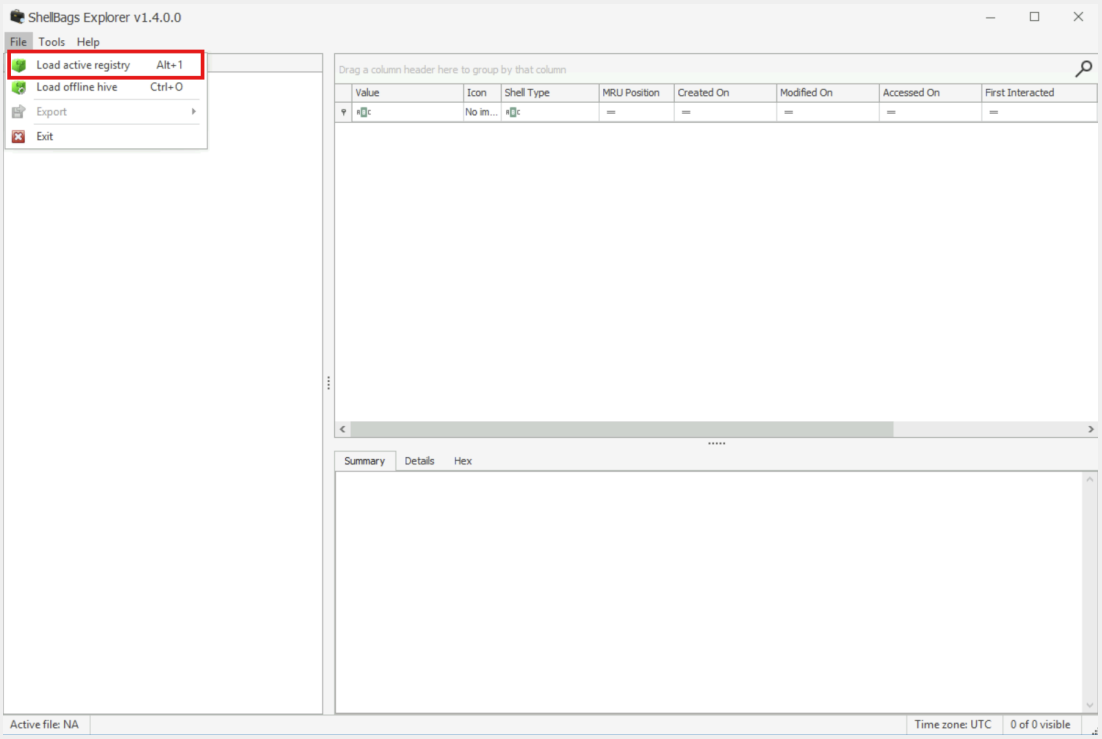


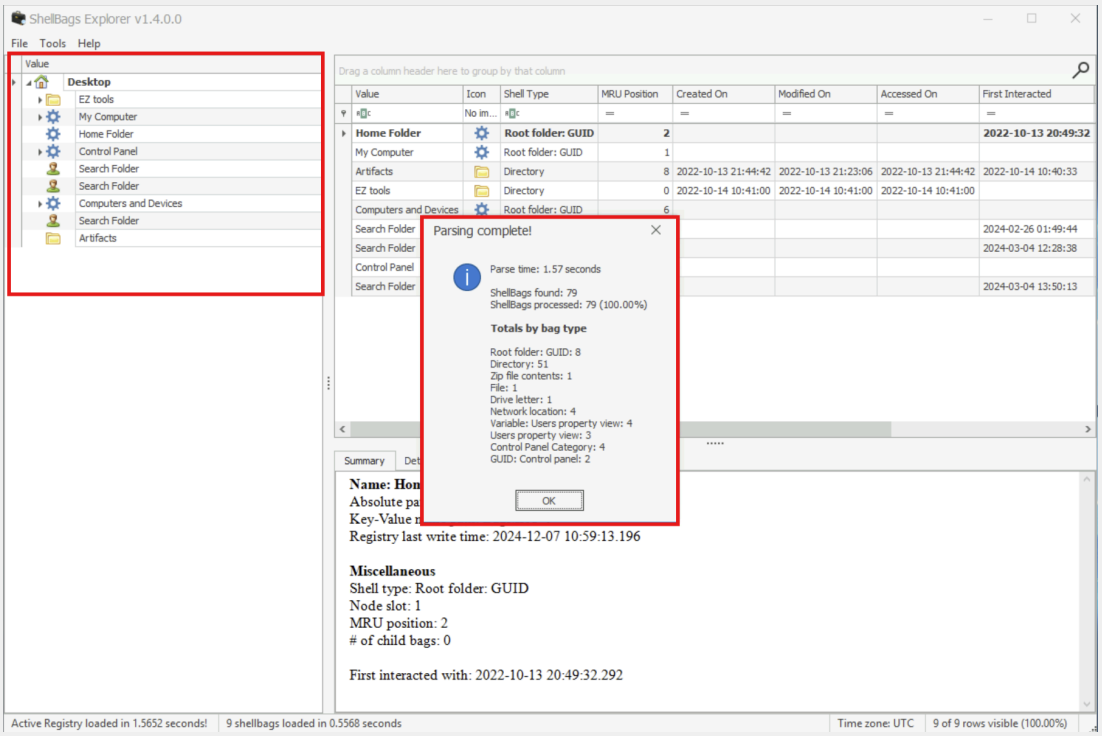

Registry Explorer v1.6.0.0
File Tools Options Bookmarks (2/0) View Help
Registry hives (1) Available bookmarks (2/0)
Values
Value Name Value Type Data Value Slack Is Deleted Data Record Reallocated
Rev RegDword 0
FFlags RegDword 1092616209
Vid RegSz {137E7700-3573-11CF-AE69-08002B2E1262} 73-68-6F-6C-64-2F
Mode RegDword 4
LogicalViewMode RegDword 1
IconSize RegDword 16
Sort RegBinary 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-01-00-00-00-30-F1-25-...
ColInfo RegBinary 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-FD-DF-DF-FD-10-00-00-... 63-68-52-65
GroupView RegDword 0
GroupByKey:FMTID RegSz {00000000-0000-0000-0000-000000000000} 74-00-3B-00-63-00
GroupByKey:PID RegDword 0
GroupByDirection RegDword 1
Type viewer Binary viewer
Value name Rev
Value type RegDword
Value 0
Raw value 00-00-00-00
Key: Local Settings\Software\Microsoft\Windows\Shell\Bags\10\Shell\{7D49D726-3C21-4F05-99AA-FDC2C9474656}
Value: Rev Collapse all hives
Selected hive: UsrClass.dat Last write: 2024-03-04 12:13:38 12 of 12 values shown (100.00%) Selected bookmark 'BagMRU' (ShellBag root key)
Hidden keys: 0 7

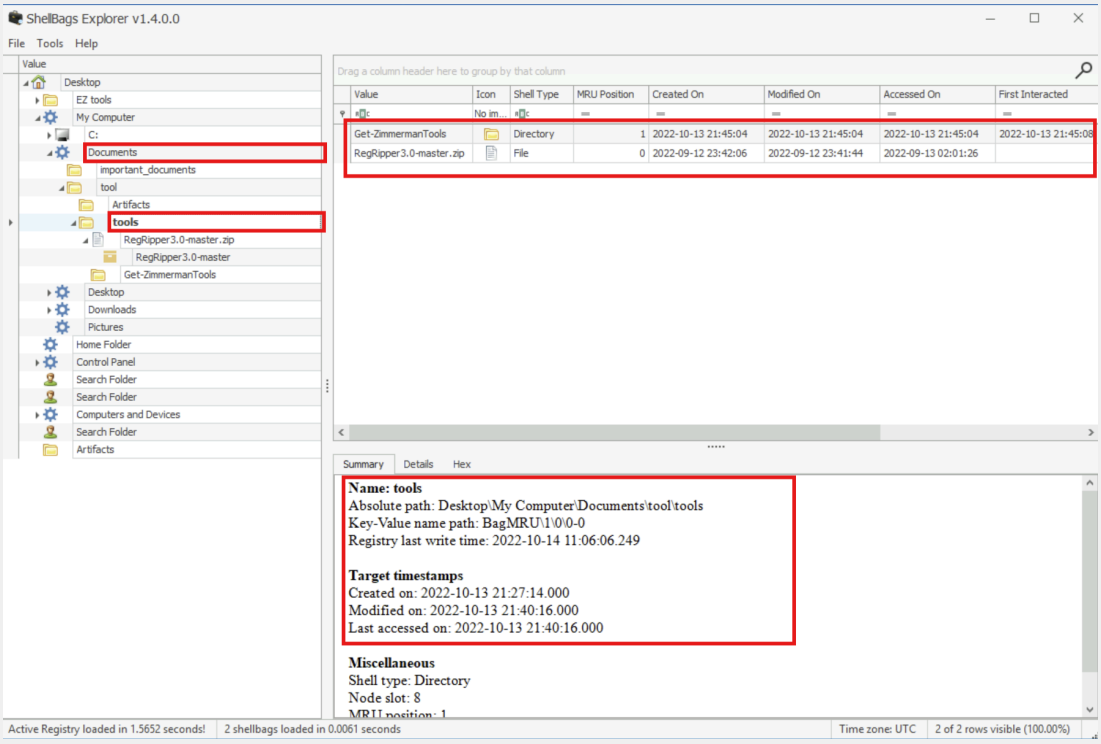

Registry Explorer v1.6.0.0
File Tools Options Bookmarks (25/0) View Help
Registry hives (2) Available bookmarks (27/0)
C:\Users\Administrator\AppData\Local\Microsoft\Window...
C:\Users\Administrator\NTUSER.DAT
ROOT
Unassociated deleted values
Values
Value Name Value Type Data
CacheLimit RegDword 1
CachePrefix RegSz
Type viewer Binary viewer
Value name CacheLimit
Value type RegDword
Value 1
Raw value 01-00-00-00
Key: C:\Users\Administrator\NTUSER.DAT
Value: CacheLimit Collapse all hives
Selected hive: NTUSER.DAT Last write: None 113 of 113 values shown (100.00%) Load complete
Hidden keys: 0 13

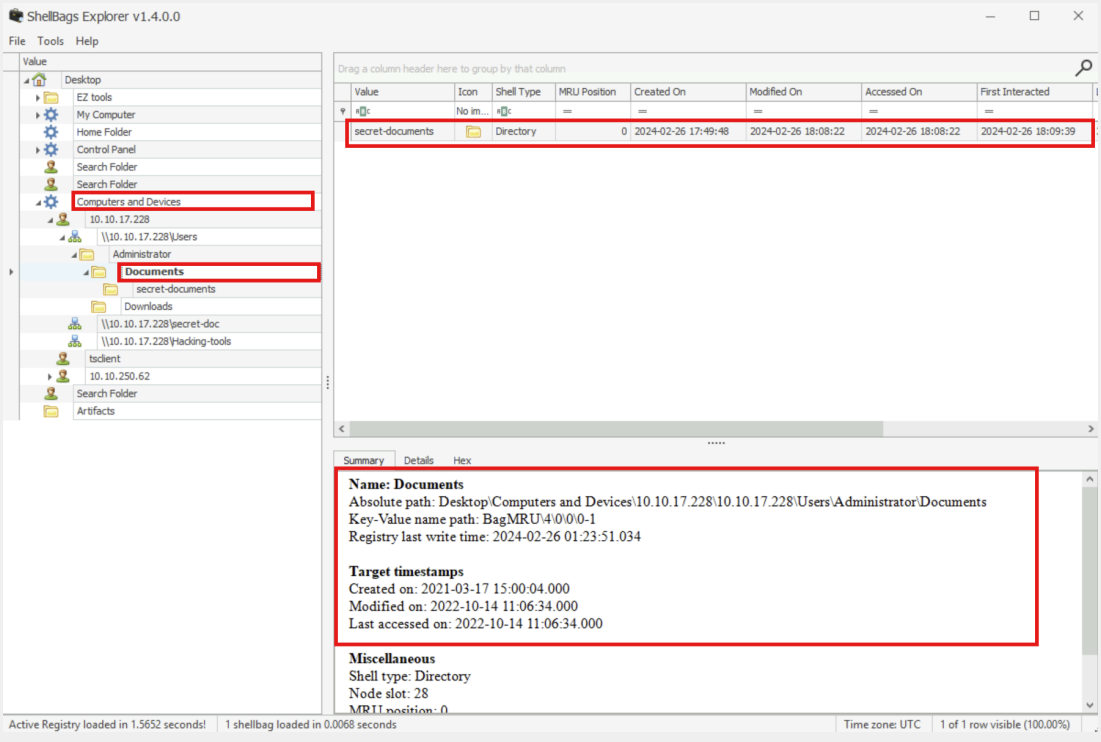


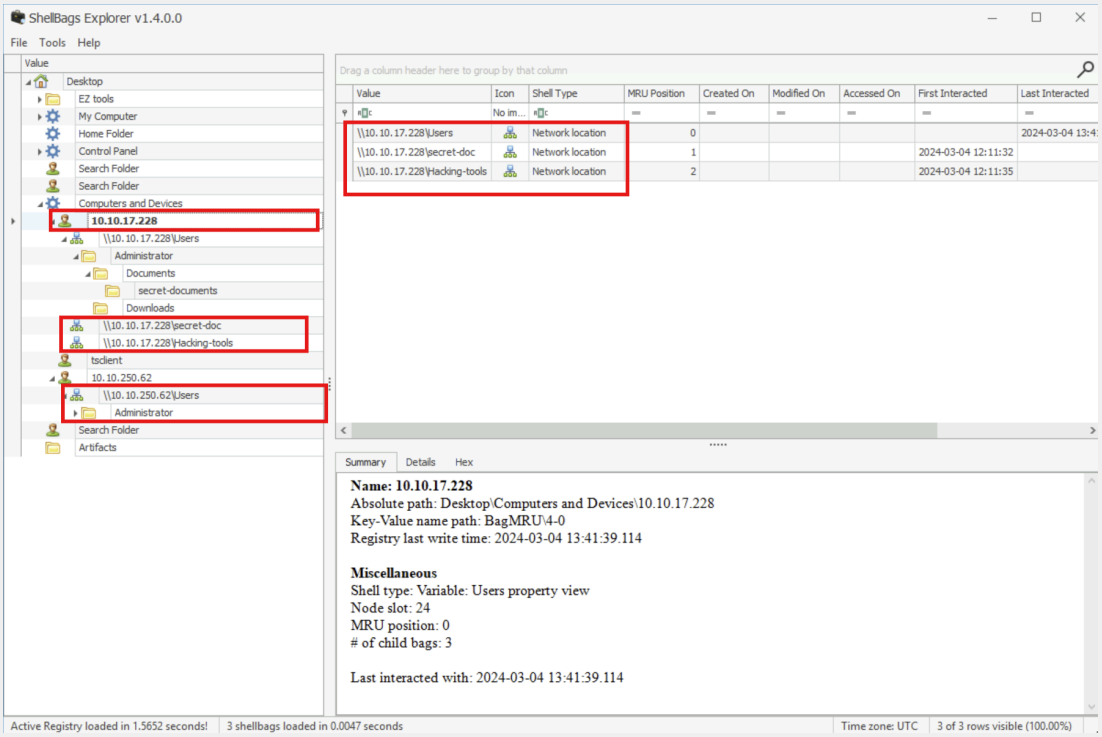
These artifacts are particularly resilient and remain a vital source of evidence, even in cases where an adversary has taken steps to erase traces of their actions. We can also use the **ShellBag Explorer** application to analyze it. We can examine each directory, setting, network share, etc, accessed by the user. **ShellBag Explorer** simplifies the process of extracting, visualizing, and analyzing ShellBag data. It offers a user-friendly interface to parse complex registry structures, providing investigators with clear insights into folder interactions on a system. This is particularly useful when examining user behavior or detecting unauthorized activity.


ShellBags Explorer v1.4.0.0
File Tools Help
Load active registry Alt+1
Load offline hive Ctrl+O
Export
Exit
Drag a column header here to group by that column
Value Icon Shell Type MRU Position Created On Modified On Accessed On First Interacted
Summary Details Hex
Active file: NA
Time zone: UTC
0 of 0 visible


ShellBags Explorer v1.4.0.0
File Tools Help
Value
Desktop
EZ tools
My Computer
Home Folder
Control Panel
Search Folder
Search Folder
Computers and Devices
Search Folder
Artifacts
Drag a column header here to group by that column
Value Icon Shell Type MRU Position Created On Modified On Accessed On First Interacted
Home Folder Root folder: GUID 2 2022-10-13 20:49:32
My Computer Root folder: GUID 1
Artifacts Directory 8 2022-10-13 21:44:42 2022-10-13 21:23:06 2022-10-13 21:44:42 2022-10-14 10:40:33
EZ tools Directory 0 2022-10-14 10:41:00 2022-10-14 10:41:00 2022-10-14 10:41:00
Computers and Devices Root folder: GUID 6
Search Folder 2024-02-26 01:49:44
Search Folder 2024-03-04 12:28:38
Control Panel
Search Folder 2024-03-04 13:50:13
Parsing complete!
Parse time: 1.57 seconds
ShellBags found: 79
ShellBags processed: 79 (100.00%)
Totals by bag type
Root folder: GUID: 8
Directory: 51
Zip file contents: 1
File: 1
Drive letter: 1
Network location: 4
Variable: Users property view: 4
Users property view: 3
Control Panel Category: 4
GUID: Control panel: 2
OK
Summary
Registry last write time: 2024-12-07 10:59:13.196
Miscellaneous
Shell type: Root folder: GUID
Node slot: 1
MRU position: 2
# of child bags: 0
First interacted with: 2022-10-13 20:49:32.292
Active Registry loaded in 1.5652 seconds!
9 shellbags loaded in 0.5568 seconds
Time zone: UTC
9 of 9 rows visible (100.00%)


ShellBags Explorer v1.4.0.0
File Tools Help
Value
Desktop
EZ tools
My Computer
C:
Documents
important_documents
tool
Artifacts
tools
RegRipper3.0-master.zip
RegRipper3.0-master
Get-ZimmermanTools
Desktop
Downloads
Pictures
Home Folder
Control Panel
Search Folder
Search Folder
Computers and Devices
Search Folder
Artifacts
Drag a column header here to group by that column
Value Icon Shell Type MRU Position Created On Modified On Accessed On First Interacted
Get-ZimmermanTools Directory 1 2022-10-13 21:45:04 2022-10-13 21:45:04 2022-10-13 21:45:04 2022-10-13 21:45:04
RegRipper3.0-master.zip File 0 2022-09-12 23:42:06 2022-09-12 23:41:44 2022-09-13 02:01:26
Summary Details Hex
Name: tools
Absolute path: Desktop\My Computer\Documents\tool\tools
Key-Value name path: BagMRU\1\0\0-0
Registry last write time: 2022-10-14 11:06:06.249
Target timestamps
Created on: 2022-10-13 21:27:14.000
Modified on: 2022-10-13 21:40:16.000
Last accessed on: 2022-10-13 21:40:16.000
Miscellaneous
Shell type: Directory
Node slot: 8
Active Registry loaded in 1.5652 seconds!
2 shellbags loaded in 0.0061 seconds
Time zone: UTC
2 of 2 rows visible (100.00%)


ShellBags Explorer v1.4.0.0
File Tools Help
Value
Desktop
EZ tools
My Computer
Home Folder
Control Panel
Search Folder
Search Folder
Computers and Devices
10.10.17.228
\\10.10.17.228\Users
Administrator
Documents
secret-documents
Downloads
\\10.10.17.228\secret-doc
\\10.10.17.228\Hacking-tools
tsclient
10.10.250.62
Search Folder
Artifacts
Drag a column header here to group by that column
Value Icon Shell Type MRU Position Created On Modified On Accessed On First Interacted
secret-documents Directory 0 2024-02-26 17:49:48 2024-02-26 18:08:22 2024-02-26 18:08:22 2024-02-26 18:09:39
Summary Details Hex
Name: Documents
Absolute path: Desktop\Computers and Devices\10.10.17.228\10.10.17.228\Users\Administrator\Documents
Key-Value name path: BagMRU\4\0\0\0-1
Registry last write time: 2024-02-26 01:23:51.034
Target timestamps
Created on: 2021-03-17 15:00:04.000
Modified on: 2022-10-14 11:06:34.000
Last accessed on: 2022-10-14 11:06:34.000
Miscellaneous
Shell type: Directory
Node slot: 28
Active Registry loaded in 1.5652 seconds!
1 shellbag loaded in 0.0068 seconds
Time zone: UTC
1 of 1 row visible (100.00%)



Below is a summary of the key points about the valuable information **ShellBags** contain:

Folder View Settings: ShellBags record how a user views specific folders, including view modes such as list, icons, or details. This insight can help reconstruct the user's preferences and behaviors when interacting with the file system.

Folder Paths: ShellBags track the paths of directories accessed by the user, whether on the local machine, external devices, or network shares. This data is crucial for tracing the user's navigation patterns and identifying sensitive or suspicious activity.

Timestamps: ShellBags store various timestamps, such as the first time a folder was created, last accessed, or potentially modified. These timestamps are invaluable for building a timeline of activities and determining the sequence of events.

User Preferences: Detailed information, such as the positioning of icons, window sizes, and folder sort orders, is also captured. These preferences provide a deeper understanding of how a user interacted with the system.

Deleted Folders: One of the most compelling forensic advantages is ShellBags' ability to retain data about folders that have been deleted. This capability allows investigators to uncover evidence even after attempts to erase it.

Network and External Drive Access: ShellBags log folders accessed on external drives or network shares. This information sheds light on the use of external or remote storage, offering insights into data movement or exfiltration.

Windows Version Specifics: The structure and data within ShellBags vary depending on the version of Windows being analyzed. Understanding these nuances ensures that investigators extract the maximum amount of relevant data.

Analyzing LNK Files

LNK files, also known as shortcut files, are small binary files in Windows that serve as references to other files, folders, or system objects. These files are automatically created by the Windows operating system when a user interacts with certain items, such as opening a document, launching an application, or creating a shortcut. Forensic analysis of **LNK** files can reveal a wealth of information about user activity and system interactions. **LNK** files provide a timeline of accessed files and folders, helping investigators understand a user's workflow and behavior. Even if the target file or folder is deleted or moved, the **LNK** file retains its metadata, offering evidence of past interactions. **LNK** files referencing external drives or USB devices can help identify the use of removable media and data transfer activity. Shortcut files pointing to network locations can indicate external resource usage or remote interactions. **LNK** files are sometimes used in phishing or malware campaigns. Analyzing them can reveal malicious payload execution paths or attacker tools.

Key informations contained in **LNK** files are below:

Target Path
- The full path to the file or folder the shortcut points to.
- This helps identify which files or folders the user accessed.

File Metadata
- Timestamps: Creation, last modification, and last access times of the target file or folder.
- File size: The size of the referenced file at the time the LNK file was created.

Volume Information
- Includes the drive serial number, volume label, and the drive letter where the target file resides.
- This can be critical when tracking files on removable drives or external media.

Network Information
- For shortcuts pointing to network resources, LNK files may include UNC paths or IP addresses of the network share.

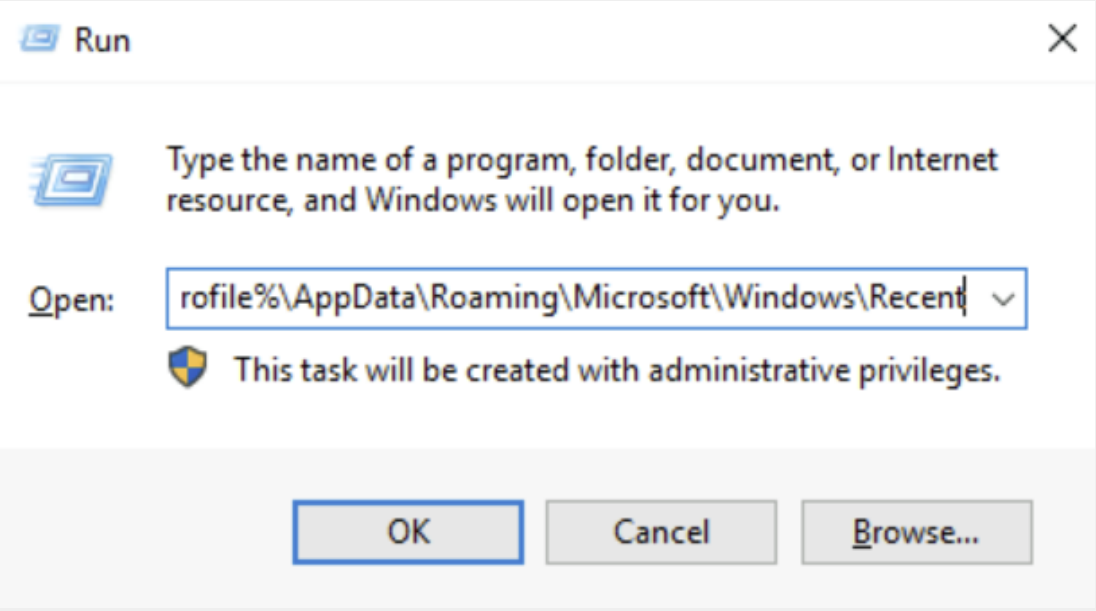
- Useful for identifying connections to shared drives or remote servers.

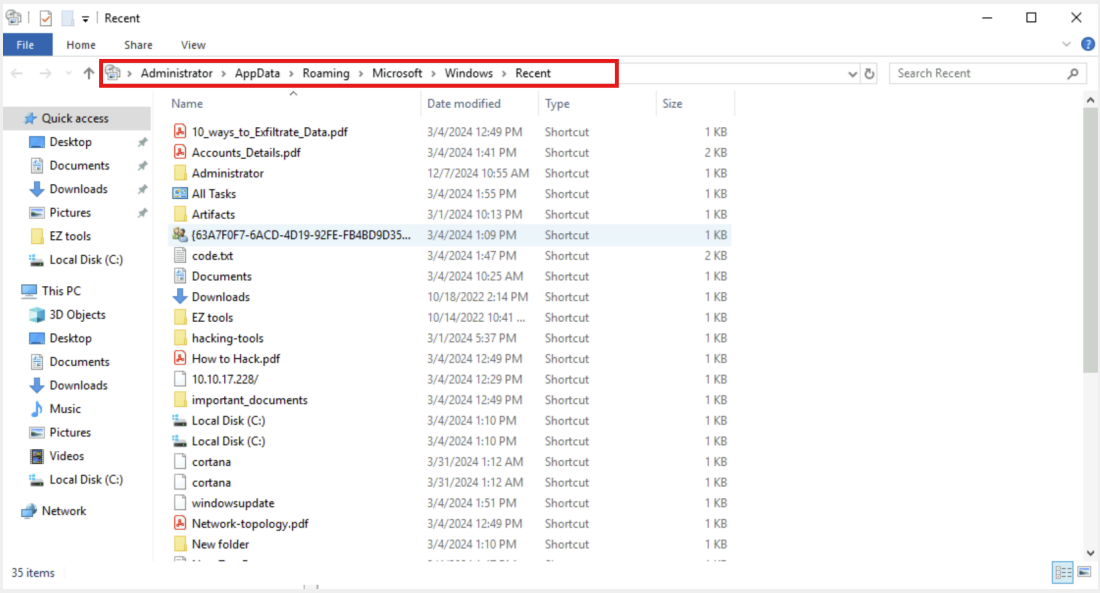
Execution Details
- Stores details about how the program or file was opened, such as the working directory and command-line arguments.
- Reveals how specific applications were used.

Icon and Metadata
- Contains information about the icon used for the shortcut, sometimes pointing to additional files or resources.

LNK files can be found in various locations, including:

- **Desktop:** User-created shortcuts.
- **Recent Items:** C:\Users\<username>\AppData\Roaming\Microsoft\Windows\Recent or %userprofile%\AppData\Roaming\Microsoft\Windows\Recent or %userprofile%\recent
- **Start Menu:** Shortcuts to applications or utilities.
- **Custom Paths:** As determined by user activity or application-specific behavior.






```
PS C:\Users\Administrator> Get-ChildItem -Path
"C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent" -Filter "*.lnk"


    Directory: C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         3/4/2024  12:49 PM            998 10_ways_to_Exfiltrate_Data.lnk
-a----         3/4/2024   1:41 PM           1851 Accounts_Details.lnk
-a----        12/7/2024  10:55 AM            646 Administrator.lnk
-a----         3/4/2024   1:55 PM            104 All Tasks (2).lnk
-a----         3/4/2024   1:09 PM            104 All Tasks.lnk
-a----         3/1/2024  10:13 PM            605 Artifacts.lnk
-a----         3/4/2024   1:09 PM            464 Change account type.lnk
-a----         3/4/2024   1:47 PM           1927 code.lnk
-a----         3/4/2024  10:25 AM            489 Documents.lnk
-a----       10/18/2022   2:14 PM            489 Downloads.lnk
-a----       10/14/2022  10:41 AM            522 EZ tools.lnk
-a----         3/1/2024   5:37 PM            527 hacking-tools.lnk
-a----         3/4/2024  12:49 PM            923 How to Hack.lnk
-a----         3/4/2024  12:29 PM            156 http--10.10.17.228-.lnk
-a----         3/4/2024  12:49 PM            661 important_documents.lnk
-a----         3/4/2024   1:10 PM            386 Local Disk (C) (2).lnk
-a----         3/4/2024   1:10 PM            386 Local Disk (C).lnk
-a----        3/31/2024   1:12 AM            154 ms-settingscortana.lnk
-a----         3/4/2024   1:51 PM            166 ms-settingswindowsupdate.lnk
-a----         3/4/2024  12:49 PM            948 Network-topology.lnk
-a----         3/4/2024   1:10 PM            512 New folder.lnk
```

```
-a----         3/4/2024   1:47 PM            653 New Text Document.lnk
-a----        12/7/2024  10:55 AM            821 NTUSER.DAT.lnk
-a----         3/4/2024  10:26 AM            719 RegistryExplorer.lnk
-a----         3/4/2024  10:26 AM           1183 RegistryExplorerManual.lnk
-a----       10/14/2022  10:44 AM            876 RegRipper3.0-master.lnk
-a----         3/1/2024  10:13 PM            773 SAM.lnk
-a----         3/4/2024   1:47 PM           1676 secret-documents.lnk
-a----        3/31/2024   1:12 AM            104 The Internet.lnk
-a----         3/4/2024   1:10 PM            104 This PC.lnk
-a----         3/4/2024   1:45 PM            669 tmp.lnk
-a----       10/18/2022   2:14 PM            587 Tools.lnk
-a----         3/4/2024   1:55 PM            448 Uninstall a program.lnk
-a----        12/7/2024  10:49 AM           1315 UsrClass.dat.lnk
-a----        12/7/2024  10:49 AM           1064 Windows.lnk
```

To examine the files accessed by the suspect, we will examine the **LNK** files created as a result using a tool called **LECmd.exe**.

```
PS C:\Users\Administrator\Desktop\EZ tools> .\LECmd.exe -f
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\10_ways_to_Exfiltrate_Data.lnk

LECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/LECmd

Command line: -f C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\10_ways_to_Exfiltrate_Data.lnk

Processing C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\10_ways_to_Exfiltrate_Data.lnk

Source file: C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\10_ways_to_Exfiltrate_Data.lnk
  Source created:  2024-03-04 12:49:18
  Source modified: 2024-03-04 12:49:18
  Source accessed: 2024-03-04 12:49:18

--- Header ---
  Target created:  2024-03-04 12:28:26
  Target modified: 2024-03-01 18:20:11
  Target accessed: 2024-03-04 12:28:26

  File size: 511,447
  Flags: HasTargetIdList, HasLinkInfo, HasRelativePath, HasWorkingDir, IsUnicode, DisableKnownFolderTracking
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwNormal (Activates and displays the window. The window is restored to its original size and
position if the window is minimized or maximized.)

Relative Path: ..\..\..\..\..\Documents\important_documents\10_ways_to_Exfiltrate_Data.pdf
Working Directory: C:\Users\Administrator\Documents\important_documents
```

```
--- Link information ---
Flags: VolumeIdAndLocalBasePath, CommonNetworkRelativeLinkAndPathSuffix

>> Volume information
  Drive type: Fixed storage media (Hard drive)
  Serial number: A8A4C362
  Label: (No label)

  Network share information
    Share name: \\4N6\Users
    Provider type: WnncNetLanman
    Share flags: ValidNetType

  Local path: C:\Users\
  Common path: Administrator\Documents\important_documents\10_ways_to_Exfiltrate_Data.pdf

--- Target ID information (Format: Type ==> Value) ---

  Absolute path: My Computer\Documents\\

  -Root folder: GUID ==> My Computer

  -Root folder: GUID ==> Documents

  -Directory ==> (None)
    Short name: IMPORT~1
    Modified:    2024-03-04 12:28:28
    Extension block count: 1

    --------- Block 0 (Beef0004) ---------
    Long name:
    Created:     2024-03-04 10:25:14
    Last access: 2024-03-04 12:28:28
    MFT entry/sequence #: 730/34 (0x2DA/0x22)

  -File ==> (None)
    Short name: 10_WAY~1.PDF
    Modified:    2024-03-01 18:20:12
    Extension block count: 1

    --------- Block 0 (Beef0004) ---------
    Long name:
    Created:     2024-03-04 12:28:28
    Last access: 2024-03-04 12:28:28
    MFT entry/sequence #: 92781/27 (0x16A6D/0x1B)

--- End Target ID information ---

--- Extra blocks information ---
```

```
>> Tracker database block
   Machine ID:  4n6
   MAC Address: 02:aa:2f:47:a0:ab
   MAC Vendor:  (Unknown vendor)
   Creation:    2024-03-04 12:28:25

   Volume Droid:       f6953da0-d6bb-4c14-8dd4-1d39a7683054
   Volume Droid Birth: f6953da0-d6bb-4c14-8dd4-1d39a7683054
   File Droid:         b29225ee-da22-11ee-82dd-02aa2f47a0ab
   File Droid birth:   b29225ee-da22-11ee-82dd-02aa2f47a0ab

>> Property store data block (Format: GUID\ID Description ==> Value)
   446d16b1-8dad-4870-a748-402ea43d788c\104    Volume Id                          ==> Unmapped GUID:
19127295-0000-0000-0000-100000000000


---------- Processed
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\10_ways_to_Exfiltrate_Data.lnk in 0.23644580
seconds ----------
```

```
PS C:\Users\Administrator\Desktop\EZ tools> .\LECmd.exe -f
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk

LECmd version 1.5.0.0

Author: Eric Zimmerman (saericzimmerman@gmail.com)
https://github.com/EricZimmerman/LECmd

Command line: -f C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk

Processing C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk

Source file: C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk
  Source created:  2024-03-04 13:45:51
  Source modified: 2024-03-04 13:47:41
  Source accessed: 2024-03-04 13:47:41

--- Header ---
  Target created:  2024-03-04 13:46:31
  Target modified: 2024-03-04 13:46:31
  Target accessed: 2024-03-04 13:46:31

  File size: 21
  Flags: HasLinkInfo, HasWorkingDir, IsUnicode, HasExpString, DisableKnownFolderTracking
  File attributes: FileAttributeArchive
  Icon index: 0
  Show window: SwNormal (Activates and displays the window. The window is restored to its original size and
position if the window is minimized or maximized.)

Working Directory: \\10.10.17.228\Users\Administrator\Documents\secret-documents
```

```
--- Link information ---
Flags: CommonNetworkRelativeLinkAndPathSuffix

  Network share information
    Share name: \\10.10.17.228\USERS
    Provider type: WnncNetLanman
    Share flags: ValidNetType

  Common path: Administrator\Documents\secret-documents\code.txt

--- Extra blocks information ---

>> Vista and above ID List data block
   Root folder: GUID ==> Computers and Devices

>> Environment variable data block
   Environment variables: \\10.10.17.228\Users\Administrator\Documents\secret-documents\code.txt

>> Tracker database block
   Machine ID:
   MAC Address: 02:19:93:4f:de:8d
   MAC Vendor:  (Unknown vendor)
   Creation:    2024-02-28 21:58:34

   Volume Droid:       f6953da0-d6bb-4c14-8dd4-1d39a7683054
   Volume Droid Birth: f6953da0-d6bb-4c14-8dd4-1d39a7683054
   File Droid:         84c0033a-d684-11ee-82db-0219934fde8d
   File Droid birth:   84c0033a-d684-11ee-82db-0219934fde8d


---------- Processed C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk in 0.10768020
seconds ----------
```

We can use powershell functions to analyze it too.

```
PS C:\Users\Administrator\Desktop\EZ tools> $lnkFile =
"C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\code.lnk"
PS C:\Users\Administrator\Desktop\EZ tools> $shell = New-Object -ComObject WScript.Shell
PS C:\Users\Administrator\Desktop\EZ tools> $shortcut = $shell.CreateShortcut($lnkFile)

PS C:\Users\Administrator\Desktop\EZ tools> Write-Host "Target Path: $($shortcut.TargetPath)"
Target Path: \\10.10.17.228\Users\Administrator\Documents\secret-documents\code.txt
PS C:\Users\Administrator\Desktop\EZ tools> Write-Host "Arguments: $($shortcut.Arguments)"
Arguments:
PS C:\Users\Administrator\Desktop\EZ tools> Write-Host "Working Directory: $($shortcut.WorkingDirectory)"
Working Directory: \\10.10.17.228\Users\Administrator\Documents\secret-documents
PS C:\Users\Administrator\Desktop\EZ tools> Write-Host "Icon Location: $($shortcut.IconLocation)"
Icon Location: ,0
```

Analyzing Jump Lists

Jump Lists are a feature introduced in Windows 7 to improve user experience by providing quick access to recently or frequently accessed items for applications pinned to the **Taskbar** or **Start Menu**. However, from a forensic standpoint, **Jump Lists** are a goldmine for investigators, offering insights into a user's file access history, application usage, and even external device interactions. **Jump Lists** are essentially metadata files that track recently or frequently accessed documents, links, or other items associated with an application.

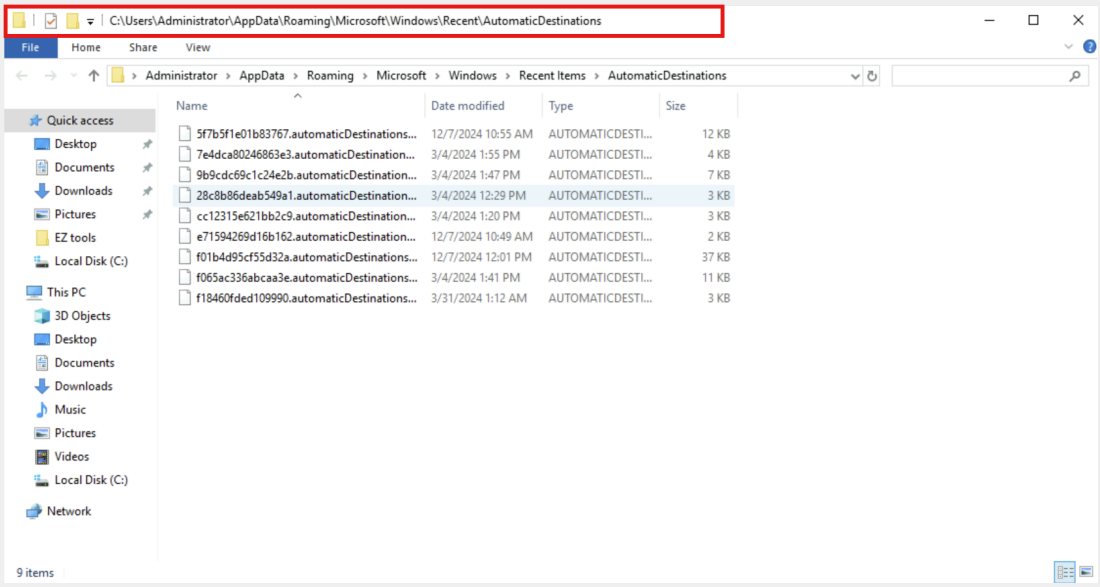
They are categorized into two types:

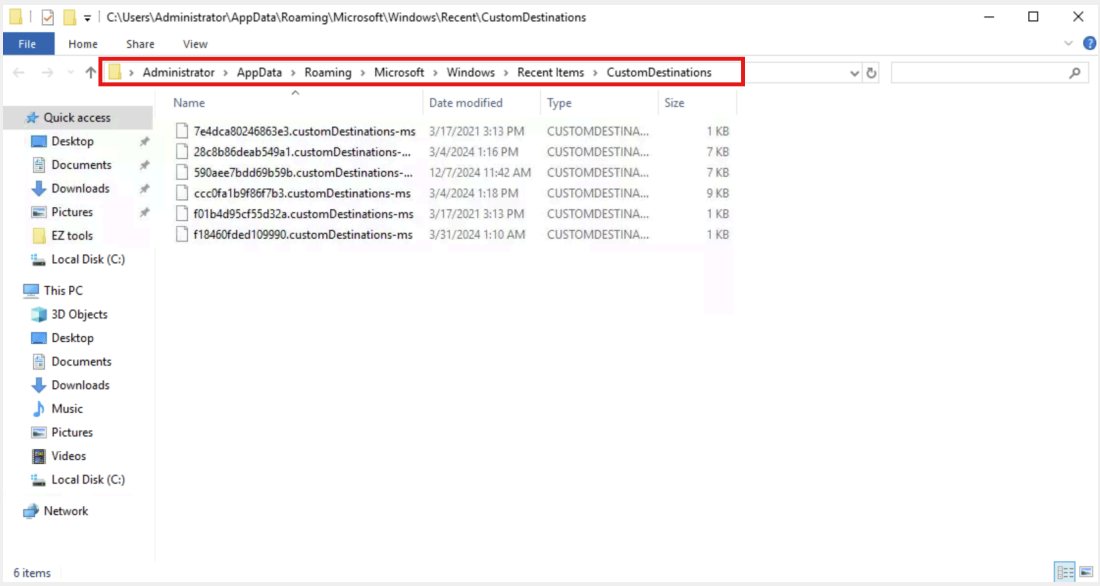
Automatic Destinations (.automaticDestinations-ms): Automatically generated Jump Lists that track recent activity for supported applications, stores system-generated lists with filenames structured as unique hashes.

Custom Destinations (.customDestinations-ms): Application-specific Jump Lists created by developers to define frequently accessed files or locations, contains application-defined lists, less common in practice.

Jump List files are stored in the user profile directory:

- **%APPDATA%\Microsoft\Windows\Recent\AutomaticDestinations**
- **%APPDATA%\Microsoft\Windows\Recent\CustomDestinations**

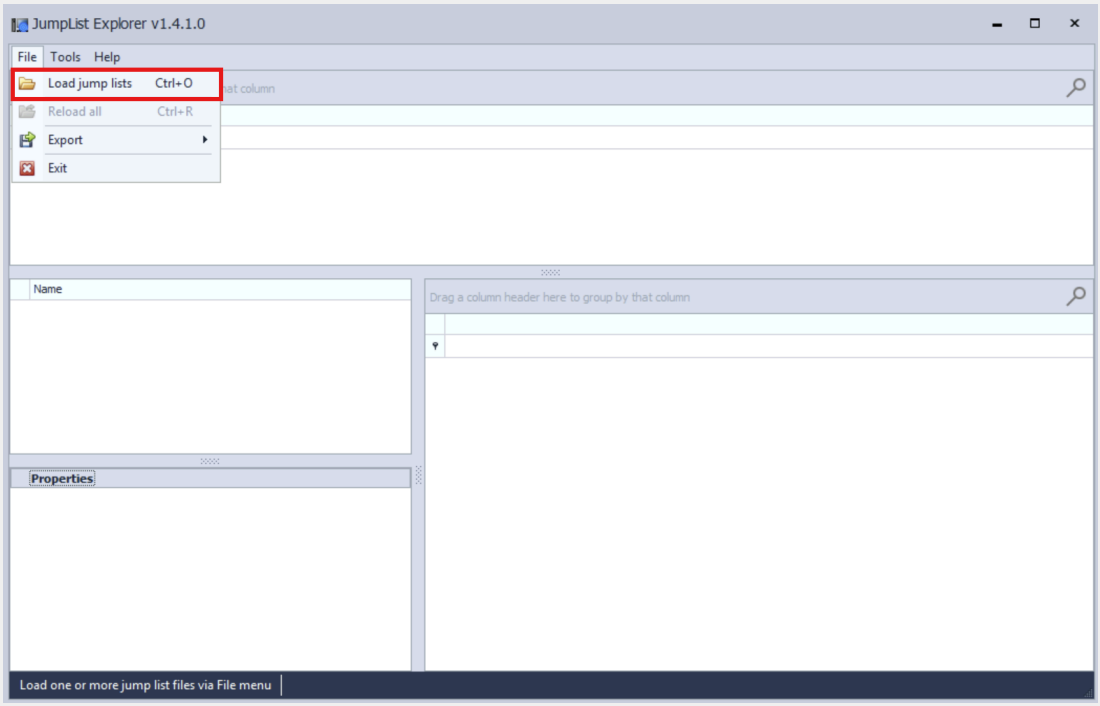


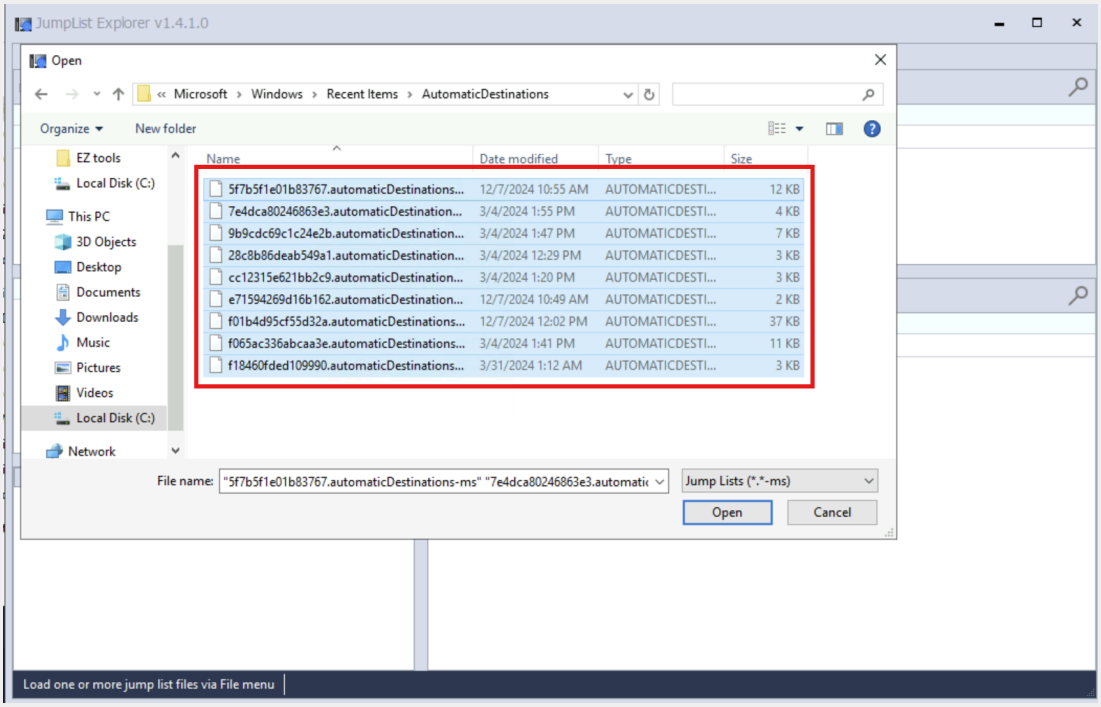


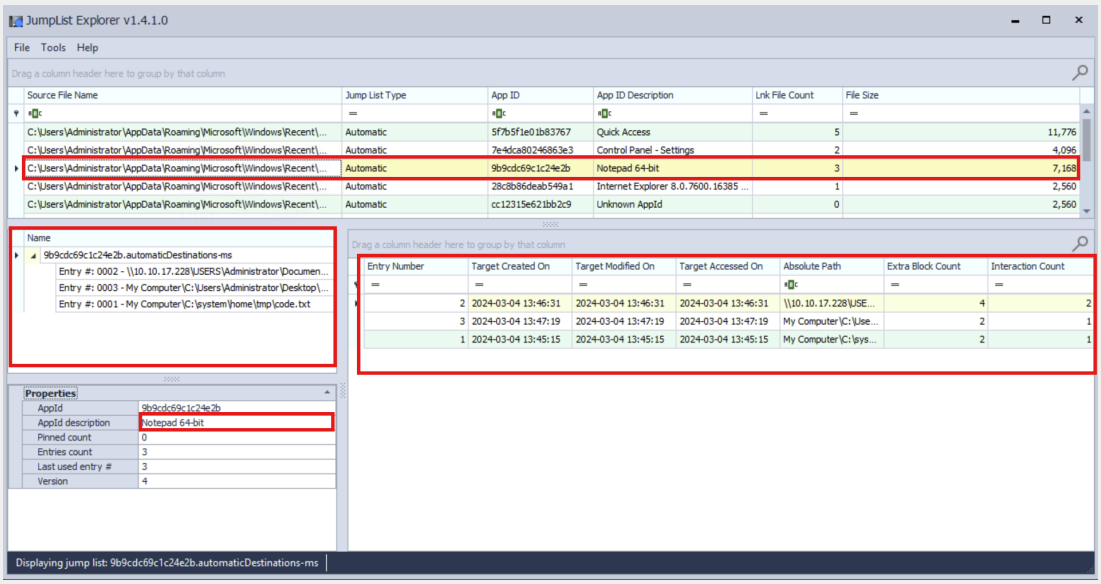

```
PS C:\Users\Administrator\Desktop\EZ tools> Get-ChildItem -Path
"$env:APPDATA\Microsoft\Windows\Recent\AutomaticDestinations" -Filter "*ms"

    Directory: C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
-a----         3/4/2024  12:29 PM           2560 28c8b86deab549a1.automaticDestinations-ms
```

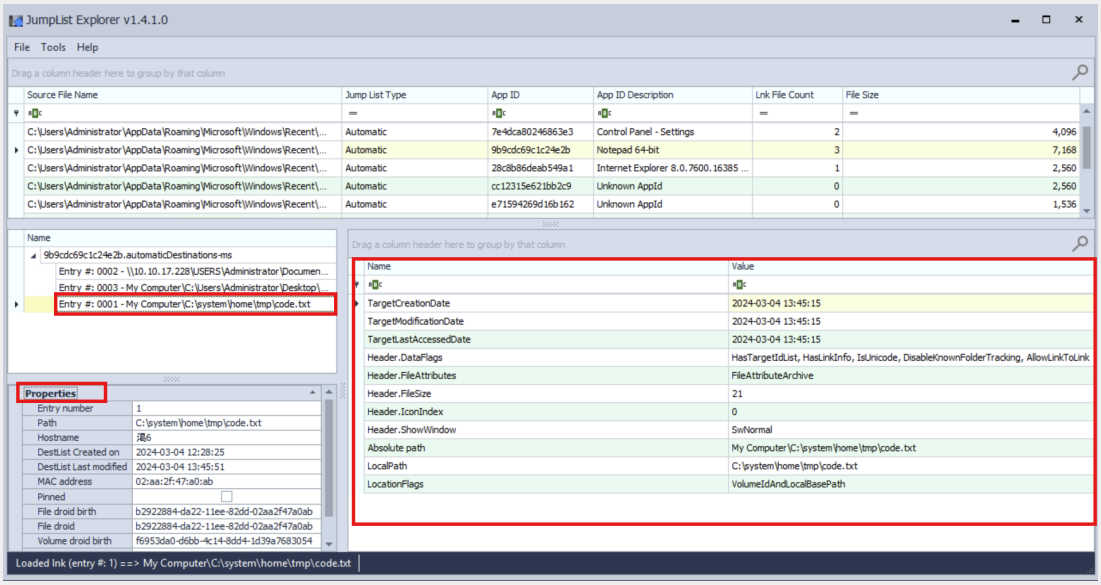
```
-a----        12/7/2024  10:55 AM          11776 5f7b5f1e01b83767.automaticDestinations-ms
-a----         3/4/2024   1:55 PM           4096 7e4dca80246863e3.automaticDestinations-ms
-a----         3/4/2024   1:47 PM           7168 9b9cdc69c1c24e2b.automaticDestinations-ms
-a----         3/4/2024   1:20 PM           2560 cc12315e621bb2c9.automaticDestinations-ms
-a----        12/7/2024  10:49 AM           1536 e71594269d16b162.automaticDestinations-ms
-a----        12/7/2024  12:02 PM          37376 f01b4d95cf55d32a.automaticDestinations-ms
-a----         3/4/2024   1:41 PM          10752 f065ac336abcaa3e.automaticDestinations-ms
-a----        3/31/2024   1:12 AM           3072 f18460fded109990.automaticDestinations-ms
```

We can also use the **JumpListExplorer** tool.




JumpList Explorer v1.4.1.0
Open
« Microsoft › Windows › Recent Items › AutomaticDestinations
Organize
New folder
EZ tools
Local Disk (C:)
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Network
Name
Date modified
Type
Size
5f7b5f1e01b83767.automaticDestinations... 12/7/2024 10:55 AM AUTOMATICDESTI... 12 KB
7e4dca80246863e3.automaticDestination... 3/4/2024 1:55 PM AUTOMATICDESTI... 4 KB
9b9cdc69c1c24e2b.automaticDestination... 3/4/2024 1:47 PM AUTOMATICDESTI... 7 KB
28c8b86deab549a1.automaticDestination... 3/4/2024 12:29 PM AUTOMATICDESTI... 3 KB
cc12315e621bb2c9.automaticDestination... 3/4/2024 1:20 PM AUTOMATICDESTI... 3 KB
e71594269d16b162.automaticDestination... 12/7/2024 10:49 AM AUTOMATICDESTI... 2 KB
f01b4d95cf55d32a.automaticDestinations... 12/7/2024 12:02 PM AUTOMATICDESTI... 37 KB
f065ac336abcaa3e.automaticDestinations... 3/4/2024 1:41 PM AUTOMATICDESTI... 11 KB
f18460fded109990.automaticDestinations... 3/31/2024 1:12 AM AUTOMATICDESTI... 3 KB
File name: "5f7b5f1e01b83767.automaticDestinations-ms" "7e4dca80246863e3.automatic
Jump Lists (*.*-ms)
Open
Cancel
Load one or more jump list files via File menu


JumpList Explorer v1.4.1.0
File Tools Help
Drag a column header here to group by that column
Source File Name
Jump List Type
App ID
App ID Description
Lnk File Count
File Size
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\... Automatic 5f7b5f1e01b83767 Quick Access 5 11,776
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\... Automatic 7e4dca80246863e3 Control Panel - Settings 2 4,096
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\... Automatic 9b9cdc69c1c24e2b Notepad 64-bit 3 7,168
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\... Automatic 28c8b86deab549a1 Internet Explorer 8.0.7600.16385 ... 1 2,560
C:\Users\Administrator\AppData\Roaming\Microsoft\Windows\Recent\... Automatic cc12315e621bb2c9 Unknown AppId 0 2,560
Name
9b9cdc69c1c24e2b.automaticDestinations-ms
Entry #: 0002 - \\10.10.17.228\USERS\Administrator\Documen...
Entry #: 0003 - My Computer\C:\Users\Administrator\Desktop\...
Entry #: 0001 - My Computer\C:\system\home\tmp\code.txt
Drag a column header here to group by that column
Entry Number
Target Created On
Target Modified On
Target Accessed On
Absolute Path
Extra Block Count
Interaction Count
2 2024-03-04 13:46:31 2024-03-04 13:46:31 2024-03-04 13:46:31 \\10.10.17.228\USE... 4 2
3 2024-03-04 13:47:19 2024-03-04 13:47:19 2024-03-04 13:47:19 My Computer\C:\Use... 2 1
1 2024-03-04 13:45:15 2024-03-04 13:45:15 2024-03-04 13:45:15 My Computer\C:\sys... 2 1
Properties
AppId 9b9cdc69c1c24e2b
AppId description Notepad 64-bit
Pinned count 0
Entries count 3
Last used entry # 3
Version 4
Displaying jump list: 9b9cdc69c1c24e2b.automaticDestinations-ms



Analyzing Prefetch Files

Prefetch files are created by the **Windows Task Scheduler** when an application is executed. They are stored in the **Prefetch** folder and contain metadata about the executed program. These files have:

- **File Name:** The name of the executed application.
- **Last Executed Time:** Timestamp of the most recent execution.
- **Execution Count:** Number of times the application has been executed.
- **Referenced Files:** List of files accessed during the application's execution.

Prefetch files are located in **C:\Windows\Prefetch**. Each file has a **.pf** extension and includes the application's name, followed by a hash value.

Windows does not natively provide tools to parse **prefetch** files, so third-party tools and scripts are often used.

We will use **PECmd.exe**, for example.

```
PS C:\Users\Administrator\Desktop\EZ tools> .\PECmd.exe -d C:\Windows\Prefetch -o C:\Desktop\Output

PECmd version 1.5.0.0
Processing Prefetch files from: C:\Windows\Prefetch
Output directory: C:\Desktop\Output

----------------------------------------------------------
```

```
Prefetch File: NOTEPAD.EXE-3D6F57C9.pf
------------------------------------------------------------
Executable Name: NOTEPAD.EXE
Hash Value: 3D6F57C9
File Size: 28 KB
Prefetch File Version: 30 (Windows 10)
Last Executed: 2024-12-05 08:30:15 UTC
Run Count: 12

Accessed Files:
  1. C:\Windows\System32\kernel32.dll
  2. C:\Windows\Fonts\segoeui.ttf
  3. C:\Users\Admin\Documents\example.txt

Timestamps:
  First Execution Time: 2024-12-01 14:22:00 UTC
  Last Execution Time: 2024-12-05 08:30:15 UTC

Additional Information:
  - Volume Serial Number: 0x1234ABCD
  - Volume Path: C:\
  - Volume Creation Time: 2023-01-15 11:00:00 UTC

------------------------------------------------------------
Prefetch File: WINWORD.EXE-5A7B8E3C.pf
------------------------------------------------------------
Executable Name: WINWORD.EXE
Hash Value: 5A7B8E3C
File Size: 35 KB
Prefetch File Version: 30 (Windows 10)
Last Executed: 2024-12-05 10:45:00 UTC
Run Count: 34

Accessed Files:
  1. C:\Windows\System32\msvcrt.dll
  2. C:\Program Files\Microsoft Office\Root\Office16\WINWORD.EXE
  3. C:\Users\Admin\Desktop\Report.docx

Timestamps:
  First Execution Time: 2024-11-28 09:15:00 UTC
  Last Execution Time: 2024-12-05 10:45:00 UTC

Additional Information:
  - Volume Serial Number: 0x1234ABCD
  - Volume Path: C:\
  - Volume Creation Time: 2023-01-15 11:00:00 UTC
```

Key informations are:

- **Executable Name:** Name of the program associated with the prefetch file.

- **Hash Value:** Unique hash generated by Windows for the application.
- **Run Count:** Number of times the application was executed.
- **Timestamps:**
  - First Execution Time: The first recorded execution of the application.
  - Last Execution Time: The most recent execution of the application.
- **Accessed Files:** List of files and DLLs accessed by the application during execution.
- **Volume Information:**
  - Serial number of the disk.
  - Volume creation date and path.
- **Prefetch File Version:** Indicates the Windows version (e.g., Windows 10 uses version 30).

Closing Remarks

The ability to analyze these various sources, whether they be recent document records, run commands, typed paths, or shellbags, allows a forensic analyst to gain a comprehensive view of the user's interactions with the system. Prefetch files, for example, offer a detailed history of program execution, while registry hives like NTUSER.dat and USERAssist can reveal much about the user's preferences, behaviors, and even their attempts to cover their tracks.

As we've seen, forensics tools like PECmd, Registry Explorer, and ShellBag Explorer facilitate the extraction and parsing of these artifacts. These tools enable investigators to identify and interpret critical data in a systematic way, leading to accurate and reliable findings. When combined with manual techniques and the knowledge of artifact locations and formats, these tools form a robust framework for analyzing user activity on Windows systems.

In an ever-evolving digital landscape, where adversaries and users alike attempt to hide their footprints, understanding and leveraging these artifacts remains an indispensable skill for forensic analysts. Whether you're responding to an incident, conducting an internal audit, or gathering evidence for legal purposes, the comprehensive analysis of user activity is essential to uncovering the truth hidden within the system. The combination of technical expertise, analytical tools, and a deep understanding of Windows' internal structures is what ultimately empowers forensic investigators to uncover and understand the actions of users within the digital domain.

Through continued research and practice, investigators can better refine their processes and tools, ensuring they remain prepared to handle the complexities of modern-day digital forensics.