Icon Cache Reconstruction: Non-Boot Recovery Strategies in Windows Forensics

First B Gayatthri[1], Second Dr.Priya. P. Sajan[2]
[1]*Member, UG student Computer Science and Engineering (Cyber Security), Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology Avadi, Chennai, Tamil Nadu, India*
[2]*Member, Senior Project Engineer, C-DAC Thiruvananthapuram*

***Abstract*—The iconcache. db is an important Windows operating system forensic investigation entity for artifact recovery, timeline correlation and file presence proof by improving the operating system performance. This research article attempts to explain how to inspect the icon cache, examining the analysis possibilities, and restores vital information like original file path and associated program names. To boot up, this research paper also describes the interpretation and correlation of icon cache analysis results with other system artifacts for examining further forensic artifacts and antiforensics approaches.**

***Index Terms*—Icon cache, Windows forensics, Anti-forensics**

I. INTRODUCTION

THE icon cache is a crucial component in Microsoft Windows operating systems designed to optimize the display of icons. Instead of having to extract icon images from every executable (.exe) and dynamic-link library (.dll) file each time they need to be displayed, Windows stores these icons in a dedicated cache. This significantly speeds up the rendering of icons in File Explorer, on the desktop, and within various applications.

History of Icon Cache in Windows

- Early Windows Versions (Windows 95/98/NT4/2000): In these older versions, the icon cache file was typically named ShellIconCache and located directly in the Windows directory (e.g., C:or C:). The size of this cache was often limited, and corruption was a fairly common issue for users.
- Windows XP: With Windows XP, the icon cache file was renamed to Iconcache.db and was moved to a user- specific location: C:and Settings User Name Settings Data.db. This change reflected the multi-user nature of Windows NT- based systems and allowed each user to have their own icon cache.
- Windows Vista and later (Windows 7, 8, 10, 11): From Windows Vista onwards, the Iconcache.db file (or sometimes multiple iconcache*.db files) is generally found in %USER- PROFILE%.db or %localappdata% These newer versions also saw improvements in how the cache is managed and rebuilt, though manual intervention is still sometimes necessary. These versions had icon caches in distributed forms rather than in a single structure as Windows XP

In complex digital forensic situations, file and system artifacts can be used to reconstruct events and understand user activity. Among these is the Windowsicon cache, a collection of hidden database files (old versions of iconcache.db and the latest window db), as a clear repository of visual metadata. Its main function is to improve system performance frequently using frequently used icons. This will repeatedly extract the operating system from available sources or dynamic link libraries. In contrast to more explicit execution protocols, the icon cache can maintain traces of activity even after source files are deleted or other forensic artifacts are intentionally deleted. However, its own binary form makes direct human interpretation impossible, requiring specialized tools and methodological approaches. These difficulties explain how forensic tools can analyze complex structures and find and obtain mechanisms that outline the most important data points that can be extracted. Additionally, we examine how these results are interpreted, correlate with other system artifacts for verification, and understand the inherent limitations of this unique source of digital evidence. By mastering icon cache analysis, investigators can gain deeper insight into system usage patterns, review event templates, and uncover important evidence that could

otherwise be hidden.

II. LITERATURE SURVEY

The discipline of digital forensics is continuously changing with the advent of new features in operating systems, storage paradigms, and anti-forensic methods. In this ever- changing scenario, the Windows icon cache has continued to remain a useful, though sometimes underappreciated, source of investigatory intelligence. This literature review examines major research and publications that emphasize the forensic relevance, structural development, and analytical techniques related to the Windows icon cache.

One of the earliest works to discern the forensic potential of the Icon Cache is Collie (2013). In this work, the evidential potential of the IconCache.db file was investigated in a systematic manner, focusing specifically on monitoring activity from USB connectable devices on Windows systems. Collie showed that the IconCache.db contains artifacts pertaining to executable files even when they only existed on, not necessarily run from, external media. This revealed the cache's value in circumstances when direct execution logs may not be present or falsified. The article noted that these artifacts are formed when the system is booted up and again populated if host- based executables are invoked or if executables are installed or executed from other media, such as DVDs and USB drives. Collie's work was followed by Lee and Lee (2014), who further developed the IconCache.db file's structure. Their work detailed more of the internal structure of the file, important for building good parsing tools. By explaining the file format, they greatly aided the capacity of forensic analysts to pro- grammatically parse out useful information from the cache. They also touched on how they can apply their research in examining anti-forensic activity, such as the indication that the absence or presence of a certain entry or alteration of file timestamps might reflect an effort to hide activity. This study highlighted that IconCache.db logs paths of executed, opened, stored, installed, or copied applications, thus extend- ing its investigative reach beyond executables on the external drives. Follow-up studies and papers, usually incorporated into comprehensive digital forensic textbooks and manuals (for instance, those of Harlan Carvey), invariably uphold the significance of the icon cache. These materials reaffirm that the icon cache may hold file paths to executed programs and procedures on fixed drives and mounted drives, and is thus helpful when dealing with unauthorized system usage, data export, or malware examination.

Evolution of Windows operating systems has also had an impact on the location and organization of the icon cache. IconCache.db was a single, dominant file in Windows XP, Vista, and 7, but Windows 8, 10, and 11 brought with them a more dispersed organization. In these newer systems, the icon cache is split across several iconcache*.db files. This shift in architecture, triggered by optimizations across different dis- play resolutions and performance, results in forensic software having to be adjusted for reading multiple files and assembling the overall image. Contemporary forensic suites and dedi- cated software are constantly revisited to accommodate such structural shifts, testifying to the digital forensic community's persistent dedication to deriving useful intelligence from this artifact. It's also worth mentioning the interaction and dis- tinction of the icon cache from other similar artifacts, like the thumbnail cache (Thumbs.db). Although both have to do with visual caching for usability for the user, studies (e.g., by researchers into Windows 7 thumbnail cache behavior) suggest they have different uses and cache different kinds of data.

Thumbnail caches are more interested in visual previews for particular file types (such as images and documents), while the icon cache is interested in program and file type icons. They can, nonetheless, deliver complementary information about user behavior. One common thread through the literature is the caveats about timestamps. While the icon cache is good evidence of what was available or opened, it tends to not keep exact execution timestamps. Its modification times in the file system represent when the cache itself was updated or rebuilt, not necessarily when a single icon was initially loaded or a program started. As such, the icon cache is best utilized in combination with other timestamp-dense artifacts such as Prefetch files, Amcache, ShimCache, Jump Lists, and Event Logs to create a well-rounded timeline of events. Lastly, the literature is indirectly referring to the recovery factor when it talks about how to flush or rebuild the icon cache (e.g., with batch scripts or Disk Cleanup tools), usually as a system administrator troubleshooting measure. Though these measures modify the forensic data by replacing an existing,

possibly faulty cache with a fresh one, the fact that a recent rebuild of the cache might have been performed is, in itself, an investigative lead indicating possible anti-forensics behavior.

III. EXISTING SYSTEM/METHODOLOGY

1. Icon Cache Recovery/Rebuilding Solutions

- Manual Deletion through File Explorer: This is the basic approach, which consists of accessing the hidden icon cache files and manually delete themselves, need to turn on ”Show hidden files, folders, and drives” in order to do this. A later system restart or explorer.exe restart will cause Windows to rebuild the cache.
- Command Prompt/Batch Scripts: This is a more powerful and automated type of manual deletion. It most often entails: Killing explorer.exe with taskkill /f/im explorer.exe. Moving to the cache directory Removing the files with del iconcache*.db /a. Relaunching explorer.exe with start ex- plorer.exe. Numerous IT workers and users make basic .bat scripts to make these actions automated.
- Disk Cleanup Utility (cleanmgr.exe): The native Disk Cleanup in Windows provides an option to remove ”Thumb- nails” that typically entails or initiates a rebuild of the corresponding icon cache. This is a friendly GUI method.
- ie4uinit -show Command: This command, run from the Run dialog or Command Prompt, is a less aggressive method intended to refresh the icon cache. While it doesn’t always perform a full rebuild, it can resolve minor icon display glitches without killing explorer.exe or requiring a full restart.
- Third-Party Utilities/Tweakers: Several third-party system utilities and ”tweaker” programs (e.g., Winaero Tweaker) commonly have a specific feature to ”Reset Icon Cache” or ”Rebuild Icon Cache.” These applications usually encapsulate the command-line or manual removal process within a user- friendly graphical interface.

1. Effectiveness for Recovery: All of these remedies tend to be effective in correcting icon display problems by compelling Windows to create a new, uncorrupted cache. The method of choice often hinges on user preference with command-line versus graphical tools, as well as the extent of corruption.

2. Solutions for Forensic Analysis of Icon Cache: Forensic examination of the icon cache is carried out using expert digital forensics software since the cache files are stored in a proprietary binary database format, for which special parsers are needed.

3. Commercial Digital Forensic Suites: These are the most developed solutions, providing integrated platforms for acquiring, parsing, and examining large varieties of digital artifacts, such as the icon cache.

i. Magnet AXIOM: Extremely popular, AXIOM is strong at parsing artifacts and offers a very rich graphical view of extracted icon paths, images, and correlating them to other evidence. AccessData

ii. Forensic Toolkit (FTK): A very solid platform capable of parsing icon cache files as part of its general artifact collection and analysis.

iii. EnCase: Another top-of-the-line forensic suite with very strong carving and parsing capabilities for many different Windows artifacts, including icon cache.

iv. X-Ways Forensics: Distinguished by its efficiency and in-depth capabilities, it is well capable of parsing icon cache data.

v. Open-Source/Free Forensic Tools: With sometimes needing more technical knowledge or being specialized for certain artifacts, these tools have affordable alternatives avail- able.

vi. Autopsy/The Sleuth Kit (TSK): Autopsy is a GUI frontend based on TSK. While TSK offers the underlying libraries for file system analysis, Autopsy will frequently have or support plugins for parsing Windows artifacts such as the icon cache.

vii. Thumbcache Viewer: Although mainly used for thumbnail caches, certain versions or analogous tools could possess features for extracting data from iconcache*.db files.

viii. Eric Zimmerman’s Tools (such as PECmd, JLECmd

- although not icon cache specific, they complement its analysis): Although Eric Zimmerman’s tools are famous for parsing other important artifacts (Prefetch, Jump Lists), stan- dalone tools to parse the icon cache files directly from his suite are not as frequently emphasized as his other tools. Still, they are crucial to correlate data present in the icon cache with evidence of execution.

ix. Custom Python Scripts: Due to the dynamic nature of Windows artifacts, most researchers and experienced prac- titioners write custom Python scripts to analyze a particular version of icon cache files or obtain distinctive data points that are not yet optimized in commercial tools. A sample, ”First-

Sound" for Windows 10 IconCache.db analysis, illustrates this open-source community activity.

x. Effectiveness in Analysis: The tools are effective in that they know how to interpret the intricate internal organization of the icon cache files, making it possible for them to:

- Extract the original file paths that were embedded.
- Display the cached icon pictures for a visual inspection.
- Render the information in an organized, searchable form (e.g., tables, CSV, JSON) that can be incorporated into a comprehensive forensic investigation.

IV. CONCLUSION

The Windows icon cache, a seemingly ordinary system component intended for performance enhancement, is a key and oft-overlooked asset in digital forensic investigations. This tutorial has outlined the process of examining these otherwise invisible binary files, from their identification and collection through the essential use of advanced parsing tools. We have seen that the icon cache is a strong source of evidence about the existence of certain programs, the presence of external storage devices, and other user activity details.

In fact, the Windows cache, including both the icon and thumbnail caches, is invaluable in digital forensics, presenting specific insights into file use. By carefully following the artifact locations of accessed, modified, and new files, these caches are direct evidence of system usage. Though recovery is possible by way of system restarts or forceful Explorer restarts, the inherent timestamps in both the icon and thumbnail caches are integral to locating digital footprints and reconstructing an accurate timeline of user activity. Although the icon cache does not provide exact execution timestamps, its special capability to hold remnants of artifacts even after files are removed makes it a precious corroborating resource. Its capacity to execute non-reboot recovery, through adept handling of the 'explorer.exe' process, also adds to its usefulness by enabling system administrators to troubleshoot display problems with- out interfering with ongoing critical operations, as well as offering forensic examiners a method of rebuilding the cache after acquisition if the corrupted state is no longer considered useful for the examination. However, the alteration of the cache's metadata during a rebuild underscores the importance of meticulous documentation in any forensic endeavor. Ulti- mately, understanding and leveraging the icon cache's data is a crucial skill for any modern digital forensic professional.

V. FUTURE SCOPE

The future scope of Iconcache.db is vibrant in nature because of emerging malware behaviours, explosion of anti-forensics techniques and other system performance optimizations.Some of these entities are pointed as follows.

1. Non-Reboot Icon/Thumbnail Cache Management

i. Controlled Cache Rebuilding: Implement a method to safely and reliably erase and rebuild Windows icon and thumbnail caches without the need for a system reboot. This will be a matter of exerting fine-grained control over the explorer.exe process (and associated components) and main- taining accurate determinations of existing cache file locations for different Windows platforms. This will fulfill an essential requirement of system administrators and incident responders to mend display problems or "clean" the cache in a controlled environment without system downtime.

ii. Pre-Rebuild Forensic Snapshot: Most importantly, prior to any cache delete and rebuild, the utility will be able to take a fast, forensically valid snapshot of the current icon and thumbnail cache files. This will include:

- Taking Volume Shadow Copies (VSCs) of the affected volumes if it is possible.
- Copying the locked cache files directly from the live system using methods that avoid file locks (e.g., raw disk access, handle manipulation).
- Saving these original cache files forensically securely (e.g., with cryptographic hashes) to maintain their "be- fore" state for subsequent analysis. This is crucial to ensure the chain of custody and that the original digital evidence of what had been cached beforehand is not deleted during recovery.

2. Recovery of Timestamps from Deleted Icon Cache Entries

i. Deep Analysis of Restored Cache Files: For any recovered icon cache files (either from the pre-rebuild snapshot or from unallocated space), the utility will have sophisticated parsing abilities to retrieve all

internal timestamps linked with individual icon entries. While the icon cache itself does not usually store direct ”last accessed” timestamps for the source files, it could have internal creation or modification timestamps of the cache entries themselves (which can be obtained from its internal database structure). The tool will carefully pull these out.

ii. Artifact Correlation: The application will enable correlation of these icon cache entries recovered from the system with the other system artifacts (e.g., Prefetch, Amcache, Jump Lists, UserAssist, MFT entries, USN Journal) in order to make more accurate timelines and place the ”digital footprints” from accessed, modified, or created files whose icons were cached in context.

3. Advanced Deleted File Metadata and Timestamp Recovery

i. File Carving and Metadata Reconstruction: The soft- ware will include sophisticated file carving methods to restore the contents of files deleted recently (whose icons may have been cached), even though their file system records are no longer available.

ii. Forensic File System Analysis: It will examine the underlying file system (e.g., NTFS Master File Table - MFT, $LogFile, USN Journal) to recover as much metadata as possi- ble for deleted files, such as their MACE (Modified, Accessed, Created, Entry Modified) timestamps, even though the file content itself may be partially overwritten or unavailable. This is an extremely challenging task since Windows rapidly reuses space.

iii. Recycle Bin Artifacts: In particular, the software will extensively analyze Recyle Bin metadata (for example, *I and* R files) to restore original file paths, times of deletion, and other related timestamps for deleted items sent to the Recycle Bin.

4. Future-Proofing and Adaptability

i. Dynamic Windows Version Support: The application will be implemented with a modular design that enables quick adaptation to upcoming Microsoft Windows updates. This involves a continuous effort of reverse-engineering new icon cache structures and mapping changes in Windows’ file system and shell behavior. It is desirable that the application will still be effective ”within the next update,” showing an active development cycle instead of fixing patches.

ii. Community Contribution/Open Source (Optional but worthwhile): Consider an open-source model or community- based update mechanism to tap into combined forensic knowl- edge in interpreting new Windows artifacts.

REFERENCE

[1] N. Collie, ”Forensic Analysis of the Windows Icon- Cache.db file,” presented at the Digital Forensic Research Workshop (DFRWS) Europe, 2013.

[2] J. Lee and S. Lee, ”A Study on the File Format of the Windows IconCache.db and its Application in Digital Forensics,” J. Digital Forensics, Security and Law, vol. 9, no. 1, 2014.

[3] H. Carvey, Windows Forensic Analysis Toolkit: Ad- vanced Analysis Techniques for Windows 7, Windows 8, and Windows 10, Syngress/Elsevier, various editions.

[4] M. Mohan and A. S. Rao, ”Digital Forensic Analysis of Windows 10 Thumbcache and Iconcache Database Files,” Int.

[5] J. Computer Science and Mobile Computing, vol. 5, no. 11, pp. 329-335, Nov. 2016.

[6] Magnet Forensics, AXIOM User Guide. [Online]. Avail- able: Magnet Forensics documentation (specific URL can be added if available and stable). Accessed: July 1, 2025.

[7] AccessData, Forensic Toolkit (FTK) User Manual. [On- line]. Available: AccessData documentation (specific URL can be added if available and stable). Accessed: July 1, 2025.

[8] Guidance Software, EnCase Forensic User Guide. [On- line]. Available: Guidance Software documentation (specific URL can be added if available and stable). Accessed: July 1, 2025.

[9] The Autopsy Project, Autopsy Documentation. [Online]. Available: https://www.autopsy.com/documentation/. Accessed: July 1, 2025.

[10] E. Zimmerman, Tools for Digital Forensics. [Online]. Available: https://github.com/EricZimmerman/SecurityTools. Accessed: July 1, 2025.