IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12827

AUTOMATION IN CYBER FORENSICS: ACCELERATING DIGITAL ARTIFACT EXTRACTION

Dhanyashree Manjunath Hegde[1], Vidya S[2]

Student, MCA Department, Bangalore Institute of Technology, Bangalore, India[1]

Faculty, MCA Department, Bangalore Institute of Technology, Bangalore, India[2]

Abstract: The rise of digital technology has increased both the frequency and complexity of cybercrimes, demanding efficient methods for uncovering digital evidence. Digital Artifact Extractors address this need by automating the identification, extraction, and preservation of artifacts left by user activity, system operations, or applications. Unlike manual processes that are time-consuming and error-prone, automation improves accuracy, reduces investigation time, and allows investigators to focus on analysis and case resolution. This paper highlights the functionality, significance, and applications of digital artifact extractors in modern cyber forensics and cybersecurity.

Keywords: Digital Forensics, Cybercrime, Artifact Extraction, Automation, Cybersecurity

I. INTRODUCTION

Digital forensics has become a cornerstone of modern cybercrime investigations, enabling analysts to uncover and reconstruct user activity across computers, mobile devices, and cloud environments. A critical step in this process is **artifact extraction**, which involves identifying and analyzing traces of digital activity from storage media or memory. A forensic image—whether of a hard drive, SSD, or system RAM provides an exact, bit-by-bit copy of the original data source, including deleted files and unallocated areas, ensuring that no potential evidence is overlooked [15], [18].

Traditional forensic approaches have largely focused on disk-based analysis, such as recovering deleted files, parsing registry entries, and examining system logs. While effective in many cases, these methods are increasingly limited by the widespread use of encryption, cloud synchronization, and ephemeral storage. As a result, investigators are turning to memory forensics, which provides access to decrypted, in-use data and transient session information that may never be stored on persistent media, offering a critical window into real-time user activity and volatile artifacts.

Digital artifacts such as registry hives, event logs, browser histories, instant messaging conversations, and session tokens are essential for reconstructing timelines and detecting unauthorized activities. With the rise of cloud services and messaging platforms, volatile and encrypted artifacts have become especially valuable, as they often contain crucial evidence that traditional disk analysis alone cannot reveal [20].

However, as the volume and complexity of digital evidence grow, manual artifact extraction is becoming increasingly impractical. The process is time-consuming, repetitive, and prone to human error, often delaying investigations. To address these challenges, researchers and practitioners are focusing on automation in artifact extraction, which can significantly reduce processing time, improve accuracy, and enhance scalability in large-scale investigations [2].

II. RELATED WORK

Drawing on insights synthesized from over twenty empirical studies and forensic experiments, this work underscores the growing importance of memory forensics as a core investigative strategy.

Volatile Memory Forensics

Volatile memory forensics enables the extraction of runtime artifacts that are not preserved on persistent storage, including decrypted content, session data, and transient application states. Techniques leveraging tools such as Windows Memory Extractor and IM Artifact Finder have demonstrated the ability to recover user messages, contact lists, session timestamps, user identifiers, and even fragments of deleted content. These tools prove particularly effective in investigations involving IM platforms.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12827

Registry and System-Level Artifact Recovery

The forensic examination of system artifacts such as Windows Registry hives, shellbags, Most Recently Used (MRU) entries, and Volume Shadow [5] Copies enables detailed reconstruction of user activity and system timelines. Utilizing hybrid forensic workflows in alignment with the National Institute of Standards and Technology (NIST) guidelines, analysts can recover evidence from both logical and physical layers. Tools like MetroExtractor exemplify this approach by integrating registry analysis with memory data to enhance evidentiary completeness.

Artifacts

Artifacts are pieces of data—files, logs, records, metadata, memory remnants, or registry entries. Artifacts may be stored on disk, in memory (RAM), or in system metadata like registries or logs.

The categorization of digital artifacts in forensic analysis is primarily based on their source of origin (operating system, installed applications, user activities, or external services like networks and cloud platforms), storage location, functional purpose, platform type, and forensic relevance.

TABLE I CATEGORIES OF ARTIFACT

Category	Examples	Location
File system	MFT, Deleted files, Timestamp	NTFS / Disk
Application	Chat logs, Browser History, App cache	App data, Program folders
Registry	MRUs, Shellbags, Runkeys	Windows Registry
Memory(RAM)	Decrypted messages, Tokens, Processes	RAM dumps
Network	IP logs, DNS cache, PCAP tiles	Network stack, Log files
Logs	Event logs, Syslogs, Application logs	C:\Windows\System32\winevt
Timeline	File MAC times, Registry Timestamps	Disk, Registry
Credentials	Password, Tokens, Session keys	Memory, SAM, Credential Manager
Malware	Persistence, Payloads	Disk, Registry, Memory
System congiguration	Software list, Time zone, Services	Registry, System logs

Artifacts are generally categorized into three types:

System Artifacts: These include Windows Registry hives, Prefetch files [2] , Event Logs, SRUM (System Resource Usage Monitor), and Amcache, which offer insights into system behavior and resource usage.

Application Artifacts: Derived from user-installed software such as web browsers (e.g., Google Chrome, Microsoft Edge), messaging platforms (e.g., Telegram Desktop, WhatsApp Desktop), and email clients. These artifacts typically include browsing history, cookies, saved credentials, chat conversations, and attachment metadata.

User Artifacts: These consist of personal documents, downloaded files, recently accessed items[5], saved passwords, and communication logs, which collectively reveal user intent and activity patterns.

III. METHODOLOGY

Artifact Extraction Process

The extraction of digital artifacts from forensic images constitutes a fundamental process in cyber forensics, aimed at uncovering evidentiary data for investigative and legal purposes.

Mount the Forensic Image : Use tools like Autopsy, FTK Imager to open image files (.E01, .dd, .img) in read-only mode to ensure the data remains unchanged[1].

Parse the File System : Identify the file system (e.g., NTFS, FAT32, ext4) using tools like TSK or Autopsy [16] to view folders, deleted files, and metadata such as MFT and journaling.

Extract Digital Artifacts Locate:

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12827

- System artifacts (Registry, Prefetch, Event Logs, SRUM, Amcache)
- Application artifacts (Browsers, IM apps like Telegram, email clients)
- User artifacts (Documents, downloads, chat logs, credentials)

Use Artifact Parsers: Tools like RegRipper (Registry), Hindsight (Browser), PECmd, SrumECmd, AmcacheParser, MFTECmd, ShellBags Explorer are used to parse the artifacts.
Keyword and Hash Search: Use Autopsy or Bulk Extractor to find keywords and match file hashes with known sets like NSRL.
Export and Report: Extract key artifacts, validate with hashes, and document everything in a forensic report, ensuring chain of custody is maintained.

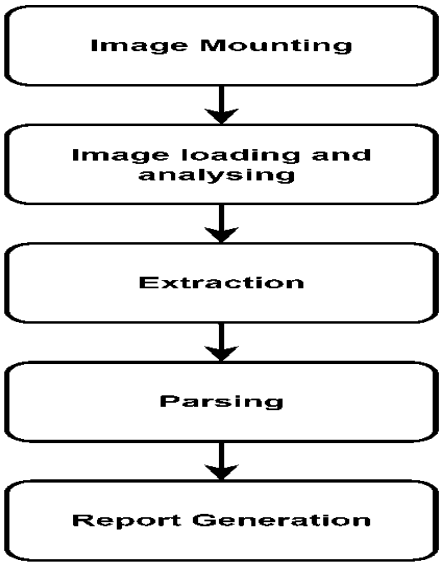

Fig 1. Architecture Diagram

The fig 1 describes the architecture of extraction process . The initial phase involves the mounting of the forensic image, which typically includes file formats such as .E01, .dd, or .img. Specialized forensic tools such as FTK Imager[18], Autopsy[16] Forensics are employed to load these images in read-only mode, thereby preserving the integrity of the original data source. This ensures that no alterations are made to the digital evidence during examination.

Subsequently, file system identification and parsing is conducted to understand the underlying structure of the imaged disk. File systems such as NTFS, FAT32, and ext4 are commonly encountered and can be analyzed using tool like Autopsy. This step allows forensic analysts to access directory hierarchies, recover deleted content, and interpret file system metadata, including timestamp information[5], journaling logs, and the Master File Table (MFT)[16].

The core phase is the extraction of digital artifacts, which involves isolating data remnants relevant to user activity, system operations, and application usage.

Once a scanned image is processed by the tool, the extracted artifact information is automatically stored in a structured database. This approach ensures that all relevant data is securely archived, easily searchable, and readily available for further analysis or reporting. By centralizing extracted artifacts in a database, the system facilitates efficient data management, traceability, and integration with other digital forensic workflows.

To automate and standardize the extraction process, a suite of specialized artifact parsers is utilized. Common tools include:
FTK Imager: A tool that creates a forensic image of computer data without affecting original evidence and hashes for file integrity. FTK imager creates a bit-by-bit image, including unallocated space and slack space [18].
RegRipper: A widely used Windows Registry parser that automates the extraction of forensic artifacts from hive files like NTUSER.DAT, SOFTWARE, and SYSTEM. Its plugin-based design helps investigators quickly identify evidence related to user activity, system settings, USB history, and application usage.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12827

Hindsight: A browser analysis tool for Google Chrome and Microsoft Edge. It parses artifacts stored in SQLite databases—such as history, cookies, downloads, autofill, credentials, and sessions—and presents them in a timeline view, making it easier to trace browsing activity and detect suspicious behavior [19].

MFTECmd: A command-line parser that analyzes the NTFS Master File Table ($MFT). It provides metadata on file creation, modification, access, and deletion, including details of renamed or deleted files, which helps reconstruct user file activity.

SrumECmd: Parses the System Resource Usage Monitor (SRUM) database (SRUDB.dat), revealing details of application execution, bandwidth use, user sessions, and power consumption. It offers valuable insights into long-term system usage patterns.

PECmd: A parser for Windows Prefetch files [2], which record application execution details such as last run time, frequency, and related DLLs. It helps determine which programs were launched and how often.

AmcacheParser: Extracts metadata from the Amcache.hve file, including file paths, hashes, timestamps, and program versions. It can confirm prior execution or installation of software, even if the program was later deleted or Prefetch disabled.

These tools transform raw binary data into structured, human-readable formats (e.g., CSV, XML, JSON), thereby facilitating further forensic examination.

Following artifact extraction, keyword and hash-based searches are performed to identify specific terms or known malicious files. Tools such as Autopsy and Bulk Extractor support the scanning of data for predefined keywords (e.g., user names, email addresses) and allow for comparison against known file hash sets, such as those provided by the National Software Reference Library (NSRL).

The final phase involves the validation and reporting of findings. All extracted artifacts are validated using cryptographic hash functions (e.g., MD5, SHA-1, SHA-256) to ensure data integrity. A comprehensive forensic report is then generated, detailing the methodologies employed, tools used, evidence recovered, and the analytical conclusions drawn. Furthermore, strict adherence to chain of custody procedures is maintained throughout the process to ensure the evidentiary admissibility of the digital artifacts in legal proceedings.

The report provides a clear, defensible record of the investigation, ensuring transparency and reliability. This guarantees the findings can withstand legal scrutiny and support court proceedings.

IV. RESULTS AND DISCUSSION

The automated artifact extraction tool demonstrated a significant improvement in forensic workflows, reducing processing time from 4–5 hours to under 30 minutes—an efficiency gain of nearly 80%. Beyond speed, the tool enhanced accuracy by consistently retrieving artifacts that manual methods often overlooked, while also reducing the likelihood of documentation errors.

It successfully extracted a wide range of evidence, including system artifacts (registry hives, event logs), application artifacts (browser histories, instant messaging data), and user artifacts (documents, downloads, saved credentials). Notably, memory forensics enabled the recovery of volatile data such as decrypted chat content and session tokens, which are often inaccessible through traditional disk-based methods.

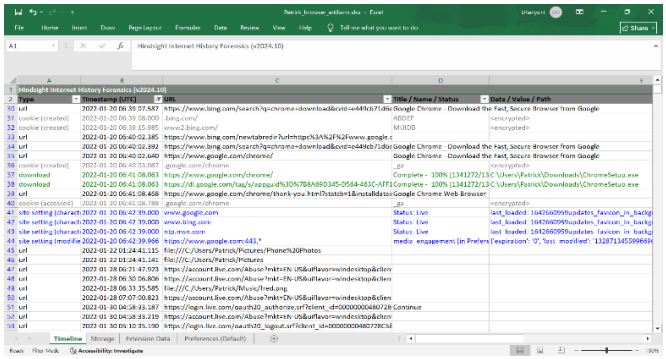

Fig 2. Browser History

The fig 2 shows the extracted browser artifact.

By automatically storing results in a structured database, the tool simplified evidence management, making search and retrieval faster and more reliable. Artifact classification further supported timeline reconstruction, helping investigators correlate user actions with system events. However, challenges remain in extending automation to handle encrypted platforms, cloud environments, and evolving obfuscation techniques, highlighting the need for continuous adaptation.

IARJSET ISSN (O) 2393-8021, ISSN (P) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Impact Factor 8.311 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 12, Issue 8, August 2025

DOI: 10.17148/IARJSET.2025.12827

Thus, advancing automation in digital forensics requires scalable solutions that can adapt to diverse and rapidly changing technological landscapes.

Future improvements may focus on integrating artificial intelligence for smarter artifact classification, supporting real-time monitoring for proactive forensics, and ensuring legal compliance to maintain evidentiary value. These enhancements would make automated tools faster, more reliable, and better equipped to address evolving cybercrime challenges.

V. FUTURE SCOPE

- **AI and Machine Learning Integration**: Future tools can go beyond extraction to automatically classify artifacts, detect anomalies, and highlight suspicious patterns.
- **Cloud, IoT, and Mobile Support**: Expanding capabilities to handle evidence from cloud services, smart devices, and mobile platforms will make investigations more complete.
- **Real-Time Forensics** – Automating live capture of volatile data (like chat sessions, tokens, and decrypted content) will help in cases where evidence disappears quickly.
- **Cross-Platform Compatibility**: Unified support for Windows, Linux, Android, and iOS would simplify investigations and reduce time.
- **Scalability for Large Cases**: Enhancing automation to handle massive datasets and batch processing will improve efficiency in enterprise-level investigations.

VI. CONCLUSION

As cybercrime continues to evolve in complexity and scale, digital artifact extraction has emerged as a cornerstone of modern forensic investigations. This paper has emphasized the critical role of memory forensics and artifact parsers in recovering volatile, encrypted, and application-specific data. By integrating disk and memory analysis, registry parsing, and application-level artifact recovery, investigators can reconstruct comprehensive timelines and uncover hidden traces of user activity.

The adoption of automated and open-source tools such as FTK Imager, RegRipper, Hindsight ensures scalability, reliability, and forensic soundness throughout the evidence extraction process. This multidisciplinary workflow not only enhances the productivity and efficiency of cyber investigations but also supports legal admissibility by preserving data integrity and maintaining a clear chain of custody. Ultimately, the unified framework presented in this study serves as a practical and adaptable methodology for forensic analysts confronting the challenges of encrypted communication, volatile data, and evolving digital ecosystems.

REFERENCES

[1]. J. Rongen and Z. Geradts, “Extraction and forensic analysis of artifacts on wearables,” *Int. J. Forensic Sci. Pathol.*, vol. 5, no. 1, pp. 312–318, Jan. 2017.

[2]. L. Garland, A. Neyaz, C. Varol, and N. K. Shashidhar, “Investigating digital forensic artifacts generated from 3D printing slicing software: Windows and Linux analysis,” *Electronics*, vol. 13, no. 14, p. 2864, Jul. 2024.

[3]. A. A. Adesina, A. A. Adebiyi, and C. K. Ayo, “Identification of forensic artifacts from the registry of Windows 10 device in relation to iDrive cloud storage usage,” *Bull. Electr. Eng. Inform.*, vol. 11, no. 1, pp. 521–529, Feb. 2022.

[4]. A. Dimitriadis, N. Ivezic, B. Kulvatunyou, and I. Mavridis, “D4I – Digital forensics framework for reviewing and investigating cyber attacks,” *Array*, vol. 5, p. 100015, Dec. 2019.

[5]. M. Soni and S. R. Pathak, “A review of forensic artifacts in a Windows 8 environment,” *Int. J. Comput. Appl.*, vol.
1, pp. 20–24, 2015.

[6]. [V. Vijayan, “Android forensic capability and evaluation of extraction tools,” M.S. thesis, Edinburgh Napier Univ., Edinburgh, U.K., 2012.

[7]. H. Carvey, “The Windows registry as a forensic resource,” *Digit. Investig.*, vol. 2, no. 3, pp. 201–205, Nov. 2005.

[8]. S. J. Yang *et al.*, “Live acquisition of main memory data from Android smartphones and smartwatches,” *Digit. Investig.*, vol. 23, pp. 50–62, Dec. 2017.

[9]. T. Roy and A. Jain, “Windows registry forensics: An imperative step in tracking data theft via USB devices,” *Int. J. Comput. Sci. Inf. Technol.*, vol. 3, no. 3, pp. 4427–4433, Jun. 2012.

DOI: 10.17148/IARJSET.2025.12827

[10]. S. Murtuza, R. Verma, J. Govindaraj, and G. Gupta, "A tool for extracting static and volatile forensic artifacts of Windows 8.x apps," in *Advances in Digital Forensics XI*, G. Peterson and S. Shenoi, Eds. Cham, Switzerland: Springer Int. Publishing, 2015, pp. 305–320.

[11]. A. Ramani and S. K. Dewangan, "Digital forensic identification, collection, examination and decoding of Windows registry keys for discovering user activities patterns," *Int. J. Comput. Trends Technol.* (IJCTT), vol. 17, no. 2, pp. 101–111, Nov. 2014, doi: 10.14445/22312803/IJCTT-V17P120.

[12]. A. M. Neil, M. Elmogy, and A. M. Riad, "A proposed framework for crime investigation based on Windows registry analysis," *Int. J. Comput. Sci. Inf. Secur.* (IJCSIS), vol. 14, no. 1, pp. 1–17, Jan. 2016.

[13]. L. Martignoni, A. Fattori, R. Paleari, and L. Cavallaro, "Live and trustworthy forensic analysis of commodity production systems," in *Recent Advances in Intrusion Detection*, vol. 6307, Springer, pp. 297–316, 2010.

[14]. B. Dolan-Gavitt, "Forensic analysis of the Windows registry in memory," *Digit. Investig.*, vol. 5, no. Supplement, pp. S26–S32, 2008, doi: 10.1016/j.diin.2008.05.003.

[15]. S. S. Nagamuthu Krishnan, "Digital evidence extraction and analysis of source images in disk forensics," *J. Emerg. Technol. Innov. Res.* (JETIR), vol. 10, no. 6, pp. 121–132, Jun. 2023. [Online]. Available: https://www.jetir.org/view?paper=JETIR2306215

[16]. M. A. Ashawa and I. O. Otache, "Forensic data extraction and analysis of left artifacts on emulated Android phones: A case study of instant messaging applications," *Circulation Comput. Sci.*, vol. 2, no. 11, pp. 8–16, Dec. 2017, doi: 10.22632/ccs-2017-252-67.

[17]. R. Sihwail, K. Omar, K. A. Zainol Ariffin, and S. Al Afghani, "Malware detection approach based on artifacts in memory image and dynamic analysis," *Appl. Sci.*, vol. 9, no. 18, p. 3680, Sep. 2019, doi: 10.3390/app9183680.

[18]. R. Dave, N. R. Mistry, and M. S. Dahiya, "Volatile memory based forensic artifacts & analysis," *Int. J. Res. Appl. Sci. Eng. Technol.* (IJRASET), vol. 2, no. 1, pp. 120–124, Jan. 2014.

[19]. K. Gupta, D. Oladimeji, C. Varol, A. Rasheed, and N. Shahshidhar, "A comprehensive survey on artifact recovery from social media platforms: Approaches and future research directions," *Information*, vol. 14, no. 12, p. 629, Nov. 2023, doi: 10.3390/info14120629.

[20]. P. Fernández-Álvarez and R. J. Rodríguez, "Extraction and analysis of retrievable memory artifacts from Windows Telegram Desktop application," *Forensic Sci. Int.: Digit. Investig.*, vol. 40, p. 301342, 2022, doi: 10.1016/j.fsidi.2022.301342.

[21]. V. L. L. Thing, K.-Y. Ng, and E.-C. Chang, "Live memory forensics of mobile phones," *Digit. Investig.*, vol. 7, pp. S74–S82, 2010.

[22]. H. Macht, "Live memory forensics on Android with Volatility," Diploma thesis, Dept. Comput. Sci., Friedrich-Alexander Univ. Erlangen-Nuremberg, Erlangen, Germany, Jan. 2013.

[23]. M. Parekh and S. Jani, "Memory forensic: Acquisition and analysis of memory and its tools comparison," *Int. J. Eng. Technol. Manag. Res.*, vol. 5, no. 2:SE, pp. 90–95, Feb. 2018, doi: 10.5281/zenodo.1198968.

[24]. S. Gaur and R. Chhikara, "Memory forensics: Tools and techniques," *Indian J. Sci. Technol.*, vol. 9, no. 48, Dec. 2016, doi: 10.17485/ijst/2016/v9i48/105851.

[25]. A. Case and G. G. Richard III, "Memory forensics: The path forward," *Digit. Investig.*, vol. 20, pp. 23–33, Jan. 2017, doi: 10.1016/j.diin.2016.12.005.

[26]. H. Nyholm *et al.*, "The evolution of volatile memory forensics," *J. Cybersecurity Privacy*, vol. 2, pp. 556–572, Jul. 2022, doi: 10.3390/jcp2030028.

[27]. Y. Cheng, X. Fu, X. Du, B. Luo, and M. Guizani, "A lightweight live memory forensic approach based on hardware virtualization," *Inf. Sci.*, vol. 379, pp. 23–41, Jul. 2017, doi: 10.1016/j.ins.2016.07.019.

[28]. Y. Cheng and X. Fu, "Investigating the hooking behavior: A page-level memory monitoring method for live forensics," in *Proc. Inf. Secur. Conf.* (ISC), 2014, pp. 255–272.

[29]. F. Block, "Windows memory forensics: Identification of (malicious) modifications in memory-mapped image files," *Forensic Sci. Int.: Digit. Investig.*, vol. 45, p. 301561, 2023, doi: 10.1016/j.fsidi.2023.301561.

[30]. T. Prem, V. Paul Selwin, and A. K. Mohan, "Disk memory forensics analysis of memory forensics frameworks flow," in *Proc. Int. Conf. Innovations Power Adv. Comput. Technol.* (i-PACT), Apr. 2017, doi: 10.1109/IPACT.2017.8244896.