Forensic Exploration of Whatsapp Web

Akshaya Mugin[1], Dr.Priya. P. Sajan [2]

[1] *Member,UG Student Computer Science and Engineering, LBS Institute of Tehnology for Women Poojappura Thiruvananthapuram*, Kerala*, India*

[2] *Member, Senior Project Engineer, C-DAC Thiruvananthapuram*

***Abstract*—This case study presents a forensic investigation of the WhatsApp Web application through an in-depth analysis of artifacts stored in browser-based IndexedDB storage. The primary objective was to determine whether IndexedDB retains forensically valuable data and whether such data can be used to perform time frame analysis of user actions. A Single Case Pretest–Posttest Quasi-Experimental Design was implemented in a controlled environment. Additionally, a proof-of-concept tool named BrowSwEx (Browser Storage Examiner) was developed to extract, examine, and analyze IndexedDB data. The findings confirm that IndexedDB holds actionable forensic artifacts, which can assist digital investigators in reconstructing chronological timelines and correlating user activities.**

***Index Terms*—Digital forensics, WhatsApp Web, IndexedDB, browser artifacts, time frame analysis, quasi experiment, BrowSwEx.**

I. INTRODUCTION

web-based messaging platforms such as WhatsApp Web have become integral to modern communication due to their accessibility, cross-platform compatibility, and ease of use. As these platforms are accessed through standard web browsers, they leverage browser-based storage technologies like IndexedDB to store user data locally, including chat messages, media files, and metadata. While these applications aim to provide secure and transient messaging experiences, they often leave behind digital footprints in the form of residual artifacts.

These artifacts can be critical in digital forensic investigations, especially in cases involving cybercrime, insider threats, or criminal communications. In particular, IndexedDB—a low-level API for client-side storage of significant amounts of structured data—has emerged as a key component in understanding how data persists within a browser after using web applications. This study investigates whether WhatsApp Web stores forensically significant artifacts in IndexedDB and whether these artifacts can be effectively leveraged to reconstruct user behavior and establish a reliable timeline of events. By analyzing these data remnants, forensic experts can potentially extract valuable insights even in the absence of server-side access or user cooperation.

II. RESEARCH OBJECTIVES

This case study aims to explore the viability of IndexedDB as a source of forensic evidence when examining activities performed through WhatsApp Web. The primary research objective is to determine whether IndexedDB storage contains forensically significant artifacts generated by WhatsApp Web interactions. These artifacts may include textual messages, sender and receiver metadata, timestamps, media references, and system-generated placeholders for deleted content. Understanding the persistence and structure of this data can enable forensic practitioners to extract and interpret user activity even after a session has ended or messages have been removed from the user interface.

Another key objective is to assess whether these artifacts can be effectively utilized to construct accurate time frame analyses within the scope of forensic investigations. By identifying timestamped entries and correlating them with specific user actions such as sending messages, receiving replies, or deleting content, investigators can create reliable chronological narratives of user behavior. This can be especially important in cases where direct access to cloud-based content is restricted or where encrypted communication protocols hinder data retrieval.

III. METHODOLOGY

To conduct a structured forensic analysis, a Single Case Pretest–Posttest Quasi-Experimental Design was adopted. This design facilitated a comparative evaluation of the IndexedDB storage state before and after specific user interactions with the WhatsApp Web application. The methodology encompassed four distinct stages to ensure reproducibility, accuracy, and control over the testing environment.

A. Controlled Environment Setup - A clean virtual environment was established to minimize background noise and eliminate potential contamination from previous browser sessions or unrelated web activity. A fresh installation of the Windows 10 operating system was performed on a virtual machine. Google Chrome, in its latest stable release, was installed with default settings. A new browser profile was initialized to guarantee that no prior session artifacts existed. To ensure the integrity of the test, all browser storage mechanisms—cache, cookies, localStorage, and IndexedDB—were manually and programmatically cleared. Additionally, system-level monitoring tools were deployed to log environmental parameters such as CPU usage, disk activity, and network traffic for potential future correlation. WhatsApp Web was then accessed via a clean login, using QR code authentication linked to a test device preloaded with
sample contacts and test messages.

B. Pretest Data Collection - Prior to executing any user actions, the initial state of IndexedDB storage was collected. Using Chrome Developer Tools (DevTools → Application
→ IndexedDB), all IndexedDB object stores associated with
WhatsApp Web were identified. Each object store's key-value
pairs, schemas, and hierarchical structure were exported using custom scripts and JSON dumps.

Screenshots and time-stamped logs were also captured to document the IndexedDB layout and associated metadata. The collected data served as a baseline for comparison in the posttest phase and ensured that any subsequent changes could be precisely attributed to the user actions performed during the experiment.

C. Action Execution - A set of controlled and purposeful user actions were performed within the WhatsApp Web interface to simulate typical real-world usage. These actions included:

Sending a plain text message to a designated test contact Receiving a message from the same contact

Opening and closing different chat windows to simulate session navigation

Sending a media file (specifically, a JPEG image)

Deleting a message for both sender and recipient (to examine deletion artifacts)

Each user action was documented with its exact timestamp and labeled using a predefined event code. Logs of these activities were recorded both manually and via screen-capture software for post-analysis correlation.

D. Posttest Data Collection - Following the execution of all test actions, a second round of IndexedDB extraction was performed. The same tools and processes used in the pretest stage were reapplied to ensure consistency. All IndexedDB object stores were re-exported and structured for differential analysis.

Both manual comparison and automated parsing were conducted. Manual comparison involved examining key-value pairs line-by-line to detect any additions, modifications, or deletions. Automated parsing was facilitated through the BrowSwEx tool, which highlighted differences, flagged new or altered entries, and cross-referenced timestamps with the recorded action log.

The posttest data provided critical insights into how WhatsApp Web interacts with IndexedDB in response to user behavior. It also confirmed whether deleted or altered content left residual traces, which were essential to evaluating the forensic potential of IndexedDB.

IV. TOOL DEVELOPMENT: BROWSWEX

To enhance the efficiency and accuracy of the forensic investigation, a dedicated analysis tool named BrowSwEx (Browser Storage Examiner) was designed and developed specifically for examining browser storage, with a focus on IndexedDB data from WhatsApp Web. BrowSwEx was built using a combination of Python for backend data handling and JavaScript for frontend data visualization. Its core objective was to simplify the complex and often cumbersome process of manually parsing browser storage formats, thereby providing forensic practitioners with a powerful and user-friendly interface for identifying critical artifacts.

The tool is capable of automatically locating the relevant IndexedDB files associated with WhatsApp Web sessions from supported browser profiles (e.g., Google Chrome). Once extracted, it parses the hierarchical object stores and decodes key-value pairs, handling both simple JSON structures

and more complex, nested formats. BrowSwEx incorporates schema recognition tailored to WhatsApp Web, allowing it to identify message contents, metadata fields (e.g., message ID, sender JID, timestamp), media file references, and user contact data.

Additionally, the tool features an interactive timeline component built with D3.js, which maps out user activity chronologically. This visualization enables investigators to track message flow, identify message deletions, and correlate timestamps with known user interactions. A filtering system allows analysts to isolate data based on specific parameters, such as time range, media type, or sender identity.

For forensic reporting, BrowSwEx includes a structured export module that generates comprehensive reports in both PDF and CSV formats. These reports include hash values for integrity verification, visual summaries, and annotated tables detailing all retrieved artifacts. The tool was validated across multiple browser sessions and environments to ensure consistency and reliability. In doing so, BrowSwEx not only reduced the time and manual effort required for browser forensic analysis but also significantly improved the interpretability of IndexedDB data for legal and investigative purposes.

V. RESULTS AND ANALYSIS

The forensic comparison between pretest and posttest IndexedDB data yielded several significant findings that support the research objectives. Firstly, IndexedDB was confirmed to store a wide range of artifacts related to WhatsApp Web interactions. These included full message bodies in plaintext, along with structured metadata such as sender and receiver JIDs (Jabber IDs), timestamps, message IDs, and chat thread identifiers. These artifacts provided detailed insights into communication sequences and user interaction history.

Furthermore, the analysis revealed that even after the deletion of messages within the WhatsApp Web interface, remnants of those actions were retained in IndexedDB. For instance, deleted messages were replaced by placeholder text entries such as "This message was deleted," but crucial metadata—like the original message ID, timestamp, sender ID, and deletion flags—remained accessible. This persistence of partial data is invaluable in forensic recovery, allowing investigators to infer deleted content and user intentions even when messages are no longer visible in the user interface.

Media interactions also left discernible traces. When a photo was sent and subsequently deleted, metadata including MIME type (e.g., image/jpeg), thumbnail data in base64 encoding, and CDN URLs referencing the media files were still present in IndexedDB. This suggests that while the media itself may no longer be accessible through the interface, its associated metadata and identifiers persist until the session is explicitly refreshed or manually cleared.

One of the most crucial findings involved the temporal dimension of IndexedDB artifacts. Each interaction—whether sending, receiving, or deleting content—was accompanied by a precise timestamp. By cross-referencing these timestamps with the controlled action log maintained during the experiment, a clear and verifiable chronology of user behavior was established. This validates the feasibility of time frame reconstruction using browser-based storage.

Additionally, it was observed that user actions could be reconstructed even after closing the browser tab, as long as the session was not refreshed or invalidated. This persistence underscores the potential of IndexedDB as a post-event forensic source, especially in scenarios where other volatile data may have already been lost.

Overall, the results reinforce the hypothesis that IndexedDB not only captures a significant amount of user activity but also retains data in a structured and queryable format suitable for forensic analysis. These findings provide a compelling basis for incorporating IndexedDB examination into standard digital forensic methodologies, especially in cases involving browser-based messaging platforms.

VI. CONCLUSION

This case study demonstrates that browser-based storage mechanisms, particularly IndexedDB, can serve as a reliable and forensically significant source of evidence when investigating activities conducted via WhatsApp Web. Despite the ephemeral nature of modern messaging systems that strive to eliminate local traces, IndexedDB retains critical data such as message contents, metadata, deletion markers, and timestamps—even after user-perceived deletions.

The ability to track, extract, and reconstruct these data points into a coherent timeline of user actions bridges a critical gap in browser-based communication investigations. The development and deployment of BrowSwEx—a custom tool designed to parse, visualize, and report IndexedDB artifacts—further validated the practicality of this forensic approach. BrowSwEx facilitated the automation of tedious manual inspection tasks, enabling investigators to generate structured and time-aligned reports with minimal overhead.

Ultimately, this study highlights IndexedDB as a viable source of post-session digital evidence that can aid in timeline reconstruction, intent analysis, and user attribution in forensic scenarios. The successful application of the Single Case Pretest–Posttest Quasi-Experimental Design confirmed the persistence and reliability of these artifacts under controlled conditions. Future research may expand upon these findings by evaluating other web-based messaging platforms and integrating similar forensic methodologies into mainstream investigative toolkits.

VII. FUTURE SCOPE

There are multiple directions in which this research can be extended. First, analyzing IndexedDB behavior across different browsers such as Mozilla Firefox, Microsoft Edge, or Brave could reveal storage variations and enhance cross-platform forensic reliability. Second, future studies could explore how IndexedDB artifacts persist under various operational conditions, including private/incognito modes, system reboots, or cache-clearing events.

Enhancing BrowSwEx with real-time monitoring capabilities or incorporating artificial intelligence for automated anomaly detection and timeline analysis may significantly reduce manual forensic workload. The same methodology can also be applied to other web-based messaging platforms like Telegram Web, Facebook Messenger, and Signal Web Beta, to benchmark forensic artifact availability and persistence.

Furthermore, investigating the legal admissibility of IndexedDB-derived evidence and understanding its implications for digital privacy can provide valuable input for policy formulation. Finally, integrating IndexedDB analysis with other browser artifact sources (e.g., cache, cookies, localStorage) could lead to a comprehensive forensic framework for browser-based communication platforms.

REFERENCES

[1] M. Zawoad and R. Hasan," FAI: A Forensics-Aware Model for Cloud Infrastructures," in Proc. of the 2013 ACM Workshop on Cloud Computing Security Workshop, 2013, pp. 31–42.

[2] M. Al-Muhtadi et al.," Indexed DB Security and Forensics," IEEE Internet Computing, vol. 25, no. 3, pp. 30–37, May 2021.

[3] Mozilla Developer Network," Using the IndexedDB API," [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDBAPI

[4] WhatsApp Inc.," WhatsApp Web Security Overview," [Online]. Available: https://faq.whatsapp.com

[5] T. Dargahi, A. Dehghantanha, M. Conti, and R. Choo," Forensics Investigation of Browsers in the Age of HTML5: A Technical Review and Comparative Analysis," Digital Investigation, vol. 22, pp. 17–29, 2017.

[6] A. Mylonas et al.," Smartphone Forensics: A Proactive Investigation Approach for Android Devices," International Journal of Information Security, vol. 14, pp. 289–305, 2015.

[7] M. Huber, S. Mulazzani, and E. Weippl," Towards Automating Forensic Analysis of Web Browser Artifacts," in Proc. of the 27th Annual Computer Security Applications Conference, 2011, pp. 150–159.