WhatsApp Analysis of WhatsApp Web: Session Persistence & Backup Security Risks

Prakriti Negi[1], Anand Kumar Mishra[2,*]
Computer Science and Engineering
NIIT University, Neemrana
[1]prakriti.negi22@st.niituniversity.in, [2]anand.mishra@niituniversity.in

***Abstract*—WhatsApp Web has become a widely used platform for communication, yet its forensic implications remain underexplored. The study investigates the browser-based forensic analysis of WhatsApp Web in the Firefox browser, highlighting session storage and backup security. The research examines whether session tokens stored in IndexedDB or local storage persist after logout and whether they can be reused for unauthorized access. In addition, the study investigates the security of WhatsApp backup data, identifying how different types of backup files are stored and whether they can be extracted or manipulated. The IP address of WhatsApp's back-end system is also captured to assess its forensic significance. Our experimental analysis highlights the persistence of certain forensic artifacts and examines the consequences of session storage mechanisms. These understandings contribute to digital forensic methodologies and help assist in recognizing possible security concerns in WhatsApp Web's architecture. The research offers a structured forensic approach to analyze WhatsApp Web, which can contribute to existing forensic research and aid in identifying potential security concerns.**

***Index Terms*—Forensic Analysis, WhatsApp Web, IndexedDB, Session Token, Browser Forensics, Digital Evidence, CyberSecurity, Firefox Browser.**

I. Introduction

WhatsApp is one of the most widely used instant messaging applications, allowing users to send and receive messages through a Web browser. With it's increasing adoption, forensic analysis of WhatsApp Web has become crucial for understanding how user data are stored and what security risks exist. This research focuses on analyzing WhatsApp Web's forensic artifacts in the Firefox browser, identifying stored data such as UUID (Universally Unique Identifier), SessionID, timestamps, and other artifacts. This research offers a detailed analysis of WhatsApp Web data storage and security, contributing to forensic methodologies.

A. Background

WhatsApp is one of the most widely used instant messaging applications, allowing billions of conversations per day. Due to it's vast user base, WhatsApp forensics has become a crucial area of digital investigations[1], helping law enforcement, cybersecurity researchers, and forensic analysts recover valuable digital evidence. The platform's end-to-end encryption (E2EE) protocol[2] provides robust mobile protection, but its browser implementation faces unique challenges. Studies show 681.5 million users access WhatsApp Web daily[3], creating forensic blind spots not covered by mobile-focused methods. Most forensic studies have focused primarily on the mobile application, analyzing chat databases, encryption mechanisms, and metadata storage. However, WhatsApp Web offers distinct forensic and security challenges, as it operates through a browser-based interface rather than a standalone application.

The difference between the mobile and web versions in terms of security and data storage is shown in table I:

TABLE I
WhatsApp Mobile vs. Web Forensic Comparison

Category	Mobile Version	Web Version
Storage	Encrypted RealmDB	IndexedDB + LocalStorage
Sessions	Device-bound E2EE	QR-code authenticated
Encryption	Message E2EE	Tokens unencrypted
Forensics	Requires root access	Browser artifacts available
Risks	Physical access needed	Session hijacking possible

Why WhatsApp Web Session Persistence Creates Forensic and Security Concerns: Unlike mobile applications that require an active device connection for decryption, WhatsApp Web maintains persistent sessions using stored tokens and IndexedDB artifacts, even after the user closes the browser. This persistence raised security concerns, as an attacker who gains access to the stored session information may be able to hijack the session without requiring QR code authentication. In addition, forensic investigators must analyze how long these artifacts persist, whether they remain after logout, and how they can be used to reconstruct user activity.

This research aims to bridge the gap between the forensic analysis of WhatsApp Web and security concerns by identifying key artifacts, evaluating security risks, and proposing forensic techniques to extract, analyze, and interpret stored data.

B. Related Work

The field of WhatsApp forensics began with mobile device analysis. Early work[4] cracked open Android's storage systems to recover chat databases, creating methods that became standard for police investigations. These techniques worked well for phones but missed the browser-based access methods that would later become common.

Building on this foundation, later studies[5] developed smarter ways to piece together deleted messages and verify exactly when conversations happened. Their timestamp analysis methods helped solve cases where timing was crucial evidence. Still, nobody had yet examined how WhatsApp Web browsers - a gap that became obvious as more people started using WhatsApp Web.

A major review paper[6] finally pointed out this blind spot in forensic research. After analyzing hundreds of cases, the authors showed how existing tools failed to properly recover evidence from browser sessions. This clear warning from experts motivated our decision to study Firefox's storage systems.

Meanwhile, other researchers[7] were uncovering privacy risks in WhatsApp's design. They proved how simple metadata - like who messaged whom and when - could reveal secrets even without reading actual messages. These findings made us realize browser-stored profile data might create similar risks.

The first real breakthrough came when a research team[8] discovered WhatsApp Web leaves behind session fragments in browser databases. Their discovery of these digital crumbs in IndexedDB showed us where to look, though they didn't test whether this leftover data could actually be used to break into accounts.

Real-world investigators[9] then created tools for grabbing evidence during active WhatsApp Web sessions. Their live forensic techniques work great when police raid an office and need to capture open chats. But they couldn't answer our burning question: how long does the evidence stick around after someone logs out?

Recent security nightmares[10] revealed flaws in how WhatsApp syncs sessions across devices. Their discovery that phones and computers don't always properly disconnect helped explain why we kept finding browser data that should have been deleted.

Network specialists[11] studying similar apps gave us smart ways to track WhatsApp Web's hidden internet traffic. While they focused on Signal messenger, their packet inspection tricks worked perfectly for our Wireshark experiments.

The scariest findings came from hacker[12] who demonstrated ways to hijack WhatsApp sessions. Their real-world attacks using leftover tokens directly inspired our security tests. Thankfully, other researchers[13] were already working on fixes that could protect against these vulnerabilities.

Most recently, a detailed comparison[14] of different WhatsApp versions proved browsers store information differently than phones. Their side-by-side tests confirmed we were right to focus specifically on Firefox's storage characteristics.

How is this Research Different?

This study significantly advances prior research by implementing a comprehensive forensic methodology that extends beyond basic artifact extraction to evaluate three critical sections: persistence characteristics, session hijacking, and backup security implications. The investigation specifically examines the potential exploitation of session tokens extracted from IndexedDB and local storage folders, testing whether these digital artifacts could enable unauthorized session restoration without legitimate authentication. The research introduces a combined analytical framework that integrates three unique investigative vectors. First, it conducts strict session persistence testing to determine whether cached authentication credentials maintain validity for potential attacker reuse. Second, systematic investigations of artifact persistence in order to determine accurate longevity measures for different forensic traces - including UUID, timestamp, and session tokens - across multiple user initiated actions such as logout procedure, cache clearance, and browser restart. Third, the study performs thorough backup data security assessments to identify recoverable backup fragments within browser storage systems while assessing the privacy risks involved.

C. Contribution

This study provides new insights into WhatsApp Web forensics that have not been thoroughly examined before. One major focus was identifying and analyzing critical digital traces, such as UUID, timestamp, profile picture, and SessionID. We tested how long these artifacts remain in the system after various actions - like logging out, clearing browser data, or deleting storage - which could be crucial for forensic investigation.

The research also explored serious security concerns, particularly around session hijacking. While other papers have noted that sessions can persist, we actually tested whether attackers could misuse stored session tokens from IndexedDB or local storage to take over active sessions. This reveals real-world risks that users and investigators should know about.

Another unique aspect was examining WhatsApp Web's backup system, something most studies overlook. We checked whether forensic tools could recover backup files and identified potential security flaws in how backups are handled. Our work also developed clear methods for tracking how long session data and other artifacts remain available after different user activities.

By combining artifact techniques with security testing, this research gives practical forensic methodologies. The findings help both in solving crimes (by showing that data can be recovered) and in preventing them (by revealing vulnerabilities that need fixing).

D. Outline of the Paper

This paper opens with the Introduction, which highlights the increasing use of WhatsApp Web and the need for browser-based forensic analysis to uncover digital traces eft during user sessions. The Related Work discusses previous studies conducted in the domain of WhatsApp forensics and emphasizes the lack of detailed research specifically targeting browser-based WhatsApp Web forensics. The Contribution section outlines the aim of this research, which is to identify and analyze forensic artifacts left behind when WhatsApp Web is accessed through a browser, focusing on session data, cache, cookies, and local storage. The Research Methodology explains the experimental setup, including the use of forensic

tools for artifact extraction. The Data Acquisition section describes the process of collecting browser artifacts and system-level traces generated during WhatsApp Web usage. The Data Analysis section presents the extracted data, including session identifiers, profile-related metadata, and other relevant information, and evaluates their forensic significance. The Results and Analysis section interprets the findings, discusses the persistence and recoverability of the artifacts, and explores the possibility of exploiting session information. Finally, the Conclusion and Future Work section summarizes the highlights of the study, highlights contributions to browser-based forensics, and suggests potential improvements such as analyzing other browsers and integrating network traffic analysis for a more comprehensive forensic approach.

II. Research Methodology

This research used standard digital forensic methods to study how WhatsApp Web stores information in Firefox and what security risks might exist. We followed careful procedures to collect, examine, and verify data while making sure that everything was done correctly to maintain accuracy. For our tests, we only used Mozilla Firefox because it handles stored data in ways that are good for forensic analysis. We worked with several forensic tools: MZcacheview[15] helped us look at cookies and cached files, FTK Imager[16] let us pull out and study browser storage information, DBbrowser for SQLite[17] helped us work with IndexedDB records, and Wireshark[18] tracked network activity. The main part of our work focused on finding and analyzing important pieces of data that WhatsApp Web leaves behind. We looked for things like UUID, profile picture, timestamp and SessionID in different storage areas of the browser.A particularly important question we tried to answer was whether this information stays in the browser even after user logs out or clears their cache - this tells us both how useful it might be for investigations and what security risks it could create. We also checked whether someone could hijack a WhatsApp session using leftover login tokens. This involved seeing if these tokens still worked after logout, which would be a serious security problem. Another area we examined was how WhatsApp handles backup data in the browser, checking if any files were saved that could be recovered. To make sure our findings were reliable, we repeated all tests several times and carefully recorded everything with screenshots and notes. This carefully crafted approach not only follows good forensic practices, but also creates a method that other researchers can use for similar studies of web-based messaging services.

A. Experimental Setup

The experimental setup for this forensic investigation was carefully designed to ensure accurate and reliable analysis of WhatsApp Web artifacts while maintaining controlled conditions throughout all testing procedures. We established our primary testing environment using a clean Windows 10[19] installation as the operating system, chosen for its widespread use and stable performance characteristics that would allow for consistent replication of results. For the browser component, Mozilla Firefox was specifically selected due to its transparent storage mechanisms, comprehensive developer tools[20], and excellent compatibility with WhatsApp Web's technical requirements. The research utilized the most current version of WhatsApp Web available during the testing period to ensure all findings would accurate reflect the platform's up-to-date storage behaviors and security implementations. Our forensic toolkit included several specialized applications, each serving distinct but complementary functions in the data examination process. FTK Imager served as our primary disk imaging and analysis tool, allowing us to create forensic copies of browser storage locations while maintaining evidentiary integrity through write-protected examination of local storage and cookie databases. For parsing and interpreting Firefox's cache structures, we employed MZcacheview, which enabled systematic recovery of temporary files that might contain fragments of profile pictures, media files, or other session artifacts. When working with the complex data structures within IndexedDB, we relied on DB Browser for SQLite, which provided essential capabilities to navigate and query database contents without modifying the original files. Network-level analysis was conducted using Wireshark, configured to monitor all traffic between the Firefox browser and WhatsApp's servers while filtering irrelevant background activity, with particular attention to capturing IP addresses and authentication handshakes. Our experimental data collection involved establishing real-time WhatsApp Web sessions using test accounts to generate authentic artifacts across various user activities, including message exchanges, profile modifications, and active session management. Browser storage data was collected at multiple stages - during active sessions, immediately after logout, and following various cleanup procedures like cache clearing, cookie deletion, and storage wiping - to thoroughly document persistence characteristics. Session-specific data points like SessionIDs were extracted and cataloged for both authentication analysis and persistence testing, while network traffic captures were synchronized with user actions to correlate storage artifacts with their corresponding network transactions. To ensure methodological rigor, each test scenario was repeated across fresh browser installations to control for potential contamination from previous sessions, with environmental variables standardized across all trials, including system time synchronization, network configuration, and background process management. Verification procedures included hash validation of interpretation methods, while the controlled environment was isolated from unnecessary internet services and potential interference sources to guarantee that all observed behaviors originated exclusively from the WhatsApp Web application under examination. This comprehensive experimental framework was deliberately structured to procedure forensically sound results that could withstand academic and legal scrutiny while providing actionable insights for both security professionals and forensic investigators, with its multi-layered approach combining storage analysis, session moni-

toring, and network inspection creating a robust foundation for examining WhatsApp Web's data management practices from multiple evidentiary perspectives. By maintaining strict controls throughout all testing phases while simulating realistic usage conditions, the methodology ensured that findings would be both scientifically valid and practically relevant to real-world forensic scenarios involving WhatsApp Web investigations.

III. Data Acquisition

This section describes the process of collecting relevant forensic artifacts and session data from WhatsApp Web accessed through the Firefox browser. The focus was on extracting data from browser storage mechanisms such as IndexedDB, local storage, cache, and cookies, as well as capturing the IP address.

The forensic investigation began by logging into WhatsApp Web on a controlled test system and performing regular activities like viewing profile picture, sending/receiving text messages, and initiating sessions. Tools such as FTK Imager, DB Browser for SQLite, Wireshark, and MZCacheView were used to examine browser storage and extract key artifacts, including UUID, SessionID, timestamp, and profile picture.

Artifacts were collected at multiple stages - immediately after login, post-logout, and after cache and cookie clearance, to evaluate their persistence. This helped assess how long these data points remain stored in the system and if they can be retrieved later for forensic purposes or if they pose any security risks.

In addition, attempts were made to inspect stored session credentials to evaluate whether a session hijack could be feasible using previously saved data. Although complete hijacking was not achieved in this phase, data was still collected to support theoretical feasibility and potential vulnerabilities.

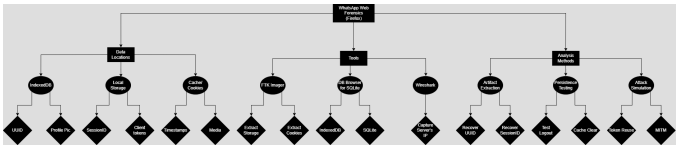

Fig. 1. Taxonomy

IV. Data Analysis

The forensic examination of WhatsApp Web artifacts involved a comprehensive analysis of data persistence patterns, security vulnerabilities, and potential forensic applications, with particular attention given to how different types of user data behaved under various conditions. The investigation began with a thorough assessment of artifact persistence across multiple browser storage mechanisms, where we carefully monitored the duration for which sections of evidence remained accessible after various user actions. When examining IndexedDB storage, the analysis revealed that UUID and profile picture demonstrated notable persistence, remaining retrievable even after user logged out of their sessions, a finding that carries significant implications for both forensic investigators and security professionals. SessionID, however, presented a different behavior pattern when stored in local storage, as they were consistently cleared upon logout or cache clearance, suggesting more robust session management for these particular token. Timestamp data, which provides crucial chronological context for digital investigations, showed intermediate persistence characteristics, remaining available after logout but being permanently erased when users cleared their browser cache. To validate these initial observations, we performed detailed testing in various user scenarios. This verification process included systematic trials of logging out and back into accounts, comprehensive cache and cookie clearance procedures, and complete browser restarts. The consistency of these findings across repeated tests strengthened the accuracy of the conclusions while also highlighting the complex interactions between user behavior and data persistence in web-based applications. Of particular forensic significance was the discovery that certain artifacts could survive typical user cleanup attempts, suggesting that WhatsApp Web sessions might leave behind more digital traces than users expect. The study then progressed to examining one of the most critical security considerations: the potential for session hijacking through token reuse. This phase of the analysis involved extracting active session credentials from both local storage and IndexedDB folders, then systematically testing whether these captured tokens maintained their authentication validity after normal session termination. While the research confirmed that session data sometimes persisted in storage systems beyond logout events, practical exploitation attempts generated mixed results, as most tokens were effectively invalidated despite their continued presence in browser storage. However, the very persistence of these tokens in certain scenarios suggests potential attack vectors that determined malicious attackers might exploit. Parallel to the session analysis, we conducted a detailed examination of WhatsApp Web's backup storage mechanisms. In contrast to the initial expectations, the investigation found no discrete backup files stored within browser-accessible locations, instead revealing a system architecture that primarily relies on live synchronization with connected mobile devices. This discovery has important impacts for both forensic investigators and security analysts, as it suggests that recovering complete message histories or media files through browser analysis alone may prove challenging. However, the research did identify potential security concerns in the synchronization process itself, particularly the possibility of man-in-the-middle attacks intercepting data during transmission between devices. Additionally, we noted that certain message fragments or metadata might be temporarily stored in cache in IndexedDB storage, creating windows of opportunity for data recovery that could prove valuable in time-sensitive investigations. The study examined several possible attack scenarios that combine the observed artifact behaviors with potential exploitation techniques, building on these core findings. The session hijacking analysis considered situations where attackers with physical or remote access to a target system might

leverage persistent tokens before their eventual expiration. The artifact tracking investigations demonstrated how long-term collection of UUID and timestamps could enable advanced user profiling and activity monitoring, even without direct access to message content. Perhaps most significantly, the backup system analysis revealed potential vulnerabilities in WhatsApp's distributed architecture, where temporary data caching might expose message fragments or metadata that users assume remain protected. A comprehensive risk classification structure was created from these security assessments, clearly highlighting the persistence characteristics and threat levels associated with each artifact type - from low-risk UUID that survive logout but offer limited exploitation potential, to high-risk session tokens that, while typically invalidated upon logout, could enable account compromise if recovered during brief windows of vulnerability. The findings particularly emphasize the importance of comprehensive cache clearance for users concerned about leaving behind recoverable traces, while also demonstrating the forensic value of persistent artifacts in legitimate investigations. This dual perspective - examining both security vulnerabilities and forensic opportunities - represents on of the study's most significant contributions to understanding WhatsApp Web's complex data management systems.

TABLE II
PERSISTENCE AND SECURITY RISKS OF EXTRACTED WHATSAPP WEB ARTIFACTS

Artifact	Persists after		Security Risk
	Logout?	Cache Clear?	Level
UUID	Yes	No	Low
Profile Picture	Yes	No	Medium
Timestamp	Yes	No	Low
SessionID	No	No	High
Backup Files	Yes	Yes	High

V. RESULT AND ANALYSIS

The forensic analysis of WhatsApp Web revealed a variety of digital artifacts that were recoverable using browser-based investigation techniques. The reporting phase focused on identifying, extracting, and analyzing these artifacts to evaluate their significance from both a forensic and a security perspective.

1. Universally Unique Identifier (UUID) - During the examination of IndexedDB, UUID associated with session and deice identification were discovered. These values serve as unique fingerprints that can be used to trace specific login instances or connected devices, providing valuable information for reconstructing user sessions.

2. Timestamps - extracted from the cookies folder with the help of FTK Imager. These help establish a detailed timeline of user interactions, such as when the session started or when the last activity was performed. Their presence strengthens the evidentiary value of browser artifacts in forensic investigations.

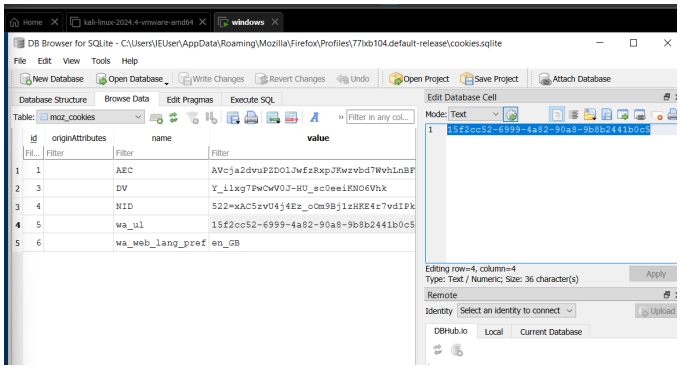

Fig. 2. UUID

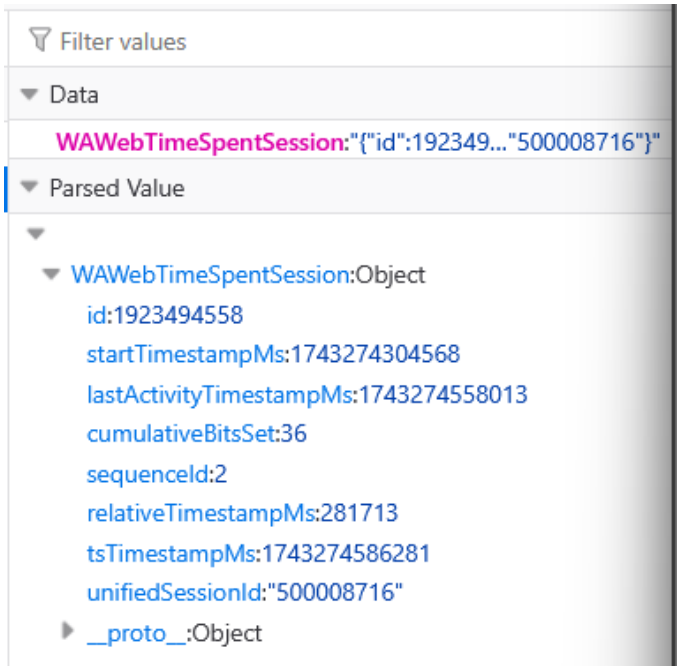

Fig. 3. Webtimestamp

3. Profile Picture - Cached versions of user profile pictures were located within the browser cache. This shows that certain user media are temporarily stored on the client side, even if they are not directly saved by the user. Such artifacts can be linked to specific accounts and provide visual identifiers during forensic investigation.

4. SessionID - Perhaps the most critical finding was the recovery of SessionID from active session tokens. This session identifier plays a central role in maintaining user authentication on WhatsApp Web. This highlights potential risks, including session hijacking, especially if attackers accessed those tokens.

5. WhatsApp Backend IP Address - Using Wireshark, we captured packets during an active WhatsApp Web session. By applying filters, we were able to identify the backend server IP address as 31.13.79.39, which is owned by WhatsApp. The screenshot below shows the QUIC (Quick UDP Internet Connections) protocol exchanged between the client and the

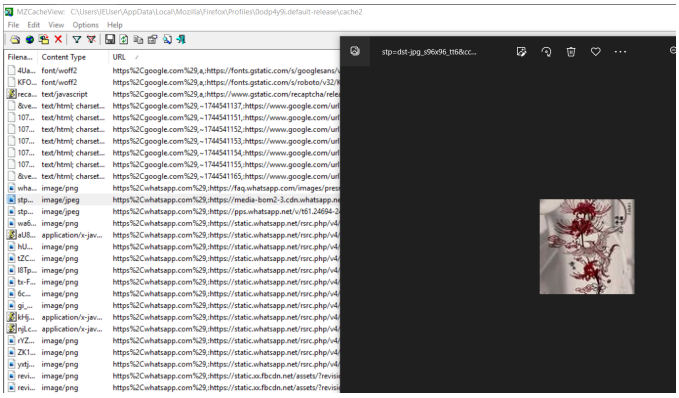

Fig. 4. Profile Picture

4	AbpropsWebRefreshId	1	1	0	0	4
5	BusinessProfileLidTableMigrationComplete	4	1	0	0	true
6	abprops-refresh-id	2	1	0	0	10
7	WALangPrefDidMismatchWithCookie	5	1	0	0	false
8	Session	20	1	0	0	bq7skd:1743274331281
9	WAWebTimeSpentSession	218	1	1	0	◆◆>{"id":1923494558,"startTimes

Fig. 5. SessionID

server.

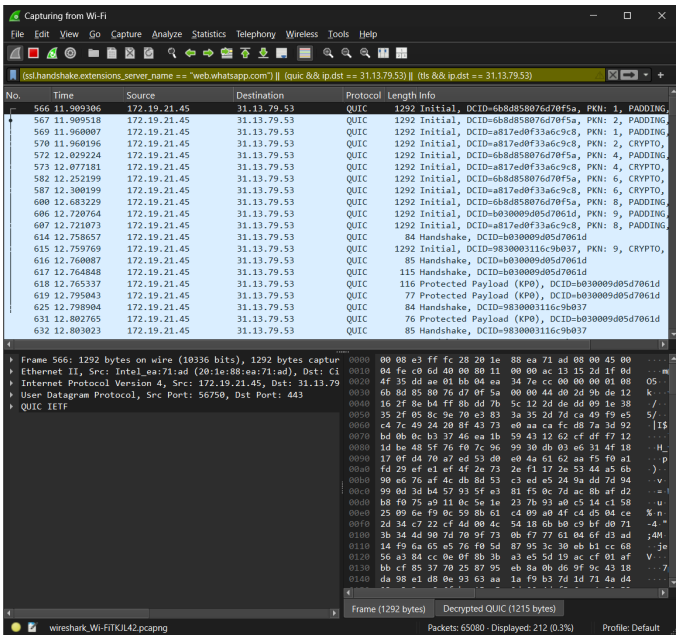

Fig. 6. Backend Server IP address

This structured analysis of WhatsApp Web confirms that browser-based messaging platforms leave behind a rich set of data that can be pivotal in forensic investigations.

VI. Conclusion and Future Work

This research demonstrates that WhatsApp Web preserves a number of important artifacts in the browser environment that can be recovered using forensic methods. In addition to providing useful evidence for forensic investigation, components such as UUID, SessionID, profile picture, and timestamp also draw attention to possible weaknesses in web-based applications.

Future research will examine active exploitation techniques, such as session hijacking, to more fully evaluate WhatsApp Web's security posture, even if the study's current phase concentrated on artifact recovery. These extra actions could provide more information on the real-world dangers that exposed session identifiers and help formulate recommendations for improved user privacy and application security.

The results highlight the need for strong session handling and cache management procedures, particularly in platforms that handle media and private communications. Making sure browser-based apps are resilient in terms of forensics and CyberSecurity is still a crucial research topic as their use continues to increase.

References

[1] Statista, "Whatsapp - statistics & facts," https://www.statista.com/topics/2018/whatsapp/, 2024, accessed: 2025-04-13.

[2] "Whatsapp backend architecture (pdf)," https://scontent.xx.fbcdn.net/v/t39.8562-6/455962147_1148247109601582_1673264986279156121_n.pdf, accessed: 14-April-2025.

[3] Cooby, "Top whatsapp web statistics," https://www.cooby.co/en/post/whatsapp-web-statistics, 2024, accessed: 2025-04-13.

[4] C. Anglano, "Forensic analysis of whatsapp messenger on android smartphones," *Digital Investigation*, vol. 11, no. 3, pp. 201–213, 2014.

[5] K. Kaushik and Y. Katara, "Forensic analysis of whatsapp chat data," in *2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO)*. IEEE, 2022, pp. 1–6.

[6] N. Soni, "Forensic analysis of whatsapp: A review of techniques challenges and future directions," *Journal of Forensic Science and Research*, 2024.

[7] N. Rastogi and J. Hendler, "Whatsapp security and role of metadata in preserving privacy," *arXiv Prepr. arXiv1701*, vol. 6817, pp. 269–275, 2017.

[8] F. Paligu and C. Varol, "Browser forensic investigations of whatsapp web utilizing indexeddb persistent storage," *Future Internet*, vol. 12, no. 11, p. 184, 2020.

[9] A. M. M. Soares, "Whatsapp web client live forensics technique." in *ICISSP*, 2022, pp. 629–636.

[10] A. Tal, "{WhatsApp} with privacy? privacy issues with {IM}{E2EE} in the multi-device setting," in *18th USENIX WOOT Conference on Offensive Technologies (WOOT 24)*, 2024, pp. 11–16.

[11] A. Afzal, M. Hussain, S. Saleem, M. K. Shahzad, A. T. Ho, and K.-H. Jung, "Encrypted network traffic analysis of secure instant messaging application: A case study of signal messenger app," *Applied Sciences*, vol. 11, no. 17, p. 7789, 2021.

[12] D. Wijnberg and N.-A. Le-Khac, "Identifying interception possibilities for whatsapp communication," *Forensic Science International: Digital Investigation*, vol. 38, p. 301132, 2021.

[13] L. Cherckesova, E. Revyakina, E. Roshchina, and V. Porksheyan, "The development of countermeasures against session hijacking," in *E3S Web of Conferences*, vol. 531. EDP Sciences, 2024, p. 03019.

[14] G. Kim, U. Hur, S. Kang, and J. Kim, "Analyzing the web and uwp versions of whatsapp for digital forensics," *Forensic Science International: Digital Investigation*, vol. 52, p. 301861, 2025.

[15] N. Sofer, "Mozilla cache viewer," https://www.nirsoft.net/utils/mozilla_cache_viewer.html, n.d., accessed: 2025-04-13.

[16] Exterro, "FTK Imager 4.7.3.81," 2025, accessed: 2025-04-04. [Online]. Available: https://www.exterro.com/ftk-product-downloads/ftk-imager-4-7-3-81

[17] T. D. B. for SQLite Team, "Db browser for sqlite," https://sqlitebrowser.org/, n.d., accessed: 2025-04-13.

[18] T. W. Team, "Wireshark," https://www.wireshark.org/download.html, n.d., accessed: 2025-04-13.

[19] Microsoft, “Windows 10 download,” https://www.microsoft.com/en-in/software-download/windows10?msockid=2cde223b4faa61842ce937974e5860b5, n.d., accessed: 2025-04-13.
[20] Mozilla, “Firefox storage inspector documentation,” https://firefox-source-docs.mozilla.org/devtools-user/storage_inspector/index.html, n.d., accessed: 2025-04-13.