INVESTIGATION OF INDEXEDDB PERSISTENT STORAGE FOR DIGITAL FORENSICS

_____

Dissertation

Presented to

Faculty of the Department of Computer Science

Sam Houston State University

_____

In Partial Fulfillment

of the Requirements for the Degree of

Doctor of Philosophy

_____

by

Furkan Paligu

August 2022

INVESTIGATION OF INDEXEDDB PERSISTENT STORAGE FOR DIGITAL FORENSICS

by

Furkan Paligu

_____

APPROVED:

Cihan Varol, PhD
Committee Director

Hyuk Cho, PhD
Committee Member

Narasimha Shashidhar, PhD
Committee Member

Amar Rasheed, PhD
Committee Member

Hamadou Saliah-Hassane, PhD
Committee Member

John B. Pascarella, PhD
Dean, College of Science and Engineering Technology

DEDICATION

This dissertation is dedicated to my father Ret. Col. Ziya Paligu and the rest of my family who always motivated me throughout my graduate studies.

ABSTRACT

Paligu, Furkan., *Investigation of IndexedDB persistent storage for digital forensics*. Doctor of Philosophy (Digital and Cyber Forensic Science), August 2022, Sam Houston State University, Huntsville, Texas.

The dependency on electronic services is increasing at a rapid rate in every aspect of our daily lives. While the Covid-19 virus remolded how we conduct business through remote collaboration applications, social media is rooting its grasp more in our day-in and day-out activities. Every day, a substantial amount of data is left in both desktop and web-based applications. As the size and the sophistication of stored data increases, so does the complexity of the technology that handles it. Consequently, forensic investigators are facing challenges in constantly adapting to emerging technologies. Hence, these technologies constitute the base for handling the vast size and volume of data in the modern era of information technology. In the scope of this dissertation the efficacy of emerging client-side technology, namely IndexedDB, is scrutinized for forensic value, practices of extraction, processing, presentation, and verification. Accordingly, a series of single case pretest-posttest quasi experiments are conducted to populate artifacts in the underlying storage technologies of IndexedDB. Subsequently, the populated artifacts are extracted and processed based on signature patterns and evaluated for their significance. Additionally, the artifacts are characterized, verified, and presented with the help of cornerstone tools that are implemented in this scope. Furthermore, time-frame analysis is constructed where it is possible to display ordered sequences of events for investigators in a suitable format.

KEY WORDS: Browser forensics; Persistent storage; Digital forensics; Artifact verification

ACKNOWLEDGEMENTS

I would first like to thank my dissertation supervisor Dr. Cihan Varol, who always pushed me with insightful feedback and motivation. His expertise helped shape my research into this dissertation and the publications I made throughout its journey.

I would like to thank all faculty of the Computer Science Department at SHSU who provided me with valuable knowledge throughout my courses. It is a gift I will carry with me for the rest of my academic career.

In addition, I would like to thank my family, particularly Yasemin, Ziya, and Yessica for their support and sympathetic ear. I am thankful for their stimulating discussions and happy distractions that made this journey much less stressful.

PREFACE

Chapter 3 partial texts referring to the exploratory first-level experiments, along with Table 8 are modified from Paligu et al. (2019) of which I am an author. The rest of the content is either my original work, or modified from Paligu and Varol (2020), and Paligu and Varol (2022) which were constructed to be placed in this dissertation with my dissertation supervisor Dr. Cihan Varol and published ahead of time.

All of the tools utilized for extraction, processing, verification, and presentation are my original work designed and implemented from scratch.

TABLE OF CONTENTS

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER I

Introduction

The dependency on digital evidence is increasing to come up with a verdict on critical civil and criminal cases (Antwi-Boasiako & Venter, 2017). The craigslist killer being identified from the emails of the victims (Braga & Pierce, 2011), and the change of the court's decision on the murder case of Casey Anthony (Arshad et al., 2018) are the proof of a future where the influence of digital evidence will be more substantial. Unfortunately, the sophistication of digital evidence today is a more pressing issue than it was in the past (Casey, 2019). As a result of rapid technological advancements, the volume and complexity of digital data are growing at an accelerated rate. On average, an individual living in the U.S. spends approximately two hours on social media sites every day while leaving a substantial amount of digital artifacts on various sites (Chew & Gunasekeran, 2021). The situation is not any different offshore. A striking example is the case of Max Schrems, an Austrian privacy activist, who made a request to see his personal information stored on Facebook servers and received 1,222 pages of documents (Mann, 2018). As the amount of data usage increases, so does the need to store it in innovative ways. This phenomenon creates a heavy burden on all individuals and forensic investigators as the ways we communicate, and store information is subject to constant change.

The technological advancements in the volume and the complexity of the data also enable the suspects to craft sophisticated manipulations on the digital evidence. Therefore, law enforcement can no longer solely rely on hard evidence being present on the suspect's computer. Furthermore, it is difficult to establish a connection between the

suspect and the evidence as the digital artifacts are fragile and can be affected by various parameters outside the suspect's reach (Kao et al., 2018). Therefore, renovation and diversification of the technologies subject to digital forensics is a constant obligation.

The impact of the increased complexity of data reflects heavily on web browser artifacts along with the techniques we utilize to extract, verify, and present them. In fact, this issue was addressed by several research papers over the course of the last decade (Mendoza et al., 2015). However, the number of research articles addressing the modern web browser storage technologies is still trivial compared to the number of articles further scrutinizing the traditional storage fields (Oh et al., 2011; Patil & Meshram, 2019). From this perspective, it is apparent that the emerging technologies are not being investigated to their full potential. As our capabilities stagnate on former techniques, significant opportunities for value generation are lost constantly.

By developing and applying innovative digital forensic techniques for the new generation of data storage, forensic investigators can be assisted in their work to find evidence and create a connection to the suspects much more efficiently. Not only these technologies can be used as a direct source of evidence, but it is also possible to utilize them for the verification of existing evidence. It is in view of the fact that when digital evidence is altered, a chain of inconsistencies occurs in the temporal data in both former and emerging storage. The coherence or the inconsistencies among data sources can point the investigators to the tempering of the evidence. Alternatively, it can prove legitimacy.

In this dissertation, the focus is turned to IndexedDB, which is an under-researched technology utilized by modern web browsers to provide client-side persistent storage (*IndexedDB API*, n.d.). It can provide over 50 MB of storage for each origin as

distinct protocol, domain, host, and URL to access a source on the web (*Browser Storage Limits and Eviction Criteria*, n.d.). This is over 45 MB of increased data storage capacity compared to previous Web Storage technology, Web Storage (Mendoza et al., 2015). Due to advantageous storage capability and fast access to persistent data with the utilization of JavaScript and JSON (JavaScript Object Notation) objects (Kimak & Ellman, 2015), IndexedDB is the potential primary storage technology in modern web browsers.

Scope and Research Question

In the scope of this dissertation, the efficacy of IndexedDB for digital forensics is inspected by scrutinizing the forensic value and the best practices of extraction, processing, and presentation of artifacts. Furthermore, new techniques are developed where IndexedDB is utilized not only as a source of evidence but also as a source of verification for the evidence that is collected from preceding sources of digital forensics. In this perspective four research questions will be addressed:

- Is it possible that the IndexedDB storage data contain a significant value for digital forensic investigations?
- Are IndexedDB storage artifacts convenient for the construction of time frame analysis in digital forensic investigations?
- Can we utilize IndexedDB artifacts to validate evidence obtained from traditional web browser technologies?
- Can we obtain a unified method to assess the level and accuracy of verification provided by IndexedDB artifacts?

In order to test the raised research questions, a series of single case pretest-posttest quasi experiments are constructed for population and evaluation of the artifacts in

IndexedDB storage of web browsers. Additionally, cornerstone tools are implemented to extract data from underlying storage technologies of IndexedDB. These tools can be potentially turned to complete investigations suites in the foreseeable future, benefiting both industry and academia. Based on the effectiveness of the metrics defined to measure the degree of the verification, a base for future evidence verification research is intended to be established. The following activities are performed for testing the research questions:

- Usage of IndexedDB is determined in the most popular fifteen websites listed by Alexa (*Top Sites in United States*, n.d.).
- A series of single case pretest-posttest quasi experiments are constructed to populate IndexedDB artifacts in prevalently used websites.
- The methods of extraction, processing, and presentation of IndexedDB artifacts are constructed for prevalent web browsers and desktop applications.
- Artifacts obtained from case-specific sources such as most frequently utilized websites, business, and social media applications are evaluated for their value to digital forensic investigations
- Methods of time-frame construction and suspect connection analysis are scrutinized on potential artifacts
- Proof-of-concept tools are designed and implemented for demonstration of extraction, processing, and presentation of data with included time-frame construction techniques
- Case-specific mechanisms for storage and management of artifacts are designed to aid proof-of-concept tools in their operations

- Techniques to investigate preceding sources of digital evidence are constructed and integrated with IndexedDB forensics to enable additional verification of the obtained evidence. These techniques are constructed and tested over the artifacts of the most popular websites listed by Alexa (*Top Sites in United States*, n.d.).

Contribution

The contributions of the dissertation can be listed as follows:

- Scrutinization of a new source of artifacts for digital forensic investigations
- Improvement of the continuity of adequate forensic investigations by incorporating new technologies with traditional techniques in terms of evidence verification
- New techniques of data extraction, processing, and presentation specific to persistent storage artifacts
- Reusability of the proof-of-concept solutions in future projects
- New methods of time-frame construction for improved activity analysis
- Detecting possible security flaws of IndexedDB storage files that would allow investigations despite browser same-origin policy restrictions

Background

Even though legacy client-side storage technologies have been present in web browsers for twenty-five years (Bortz et al., 2012), their structure has undergone a major change with the introduction of persistent storage. As it is discussed in the Literature Review Section, the coverage of newer and persistent technologies in academic work is less than ideal. In this section, a background is provided for the emphasized technologies and their relevant extensions to provide a better understanding.

Client-Side and Server Side-Storage

One of the most frequently used web service models in computer networks is the client and server model. It constitutes a large portion of the services provided on the world wide web including website hosting, and application data storage (Dixon et al., 1997). In client-server architectures, data can be stored on both server and client sides. Both options carry advantages and disadvantages (Al-Shaikh & Sleit, 2017). Predominantly, the security of the client-side information is inevitably dependent on the user and the web browser since the control of the stored data resides predominantly with the user of the client device (Youn et al., 2018). Nevertheless, keeping a large data storage on the server-side generates inconvenient levels of network traffic for the service providers which eventually affects the performance and scalability. Furthermore, the scalability of the applications is dependent on the choice of storage techniques. When a substantial amount of data is stored on the server, the performance is directly proportional to the changes in the number of active users (Woods et al., 1999). The most suitable option depends on the specifications of the application. A commonly accepted practice is to utilize server-side storage for sensitive information when less vulnerable information is stored on the client-side (Walker & Chapra, 2014).

The client-side storage is associated with the performance of the application. Therefore, the need to create scalable applications where less sensitive data can be utilized fast and efficiently resulted in the rapid advancement of client-side storage technologies over the last twenty-five years (Wyse & Subramanian, 2013). These technologies can be listed as Cookies, Flash Storage, Web Storage, IndexedDB, and

Cache API in chronological order. Table 1 gives a summary of the client-side storage technologies with their capacities.

Table 1

Client-Side Technologies and Storage Capacity

Technology	Storage Size	Notes
Cookies	4 KB	Legacy client-side storage technology
Web Storage	5 MB	Predecessor of IndexedDB which is still actively in use
Session Storage	5 MB	Non-persistent storage
IndexedDB	>50 MB	Bridge between Web Storage and Cache API
Cache API	>500 MB	Currently does not have considerable content in Alexa top 20 web sites

The early client-side storage technologies such as cookies were smaller in size compared to subsequently emerging technologies (Millett et al., 2001). Cookies, which are still in use in modern web browsers, can store up to four kilobytes of data while storage with Cache API is associated with up to fifty percent of available disk space in the client device (*Browser Storage Limits and Eviction Criteria*, n.d.).

As the size of the storage differs, the content that is associated with it also divaricates (Nalawade et al., 2016). Technologies that are similar in their size and structure such as Web Storage and IndexedDB display similarities in their content. However, the technologies that are far apart in terms of size such as cookies and Cache API store very different data. Despite the differences in their size, content, and performance capabilities, client-site storage technologies have features in common. For instance, they are persistent, and the handled data is saved in the file system of the client device.

Table 2 gives a summary of the commonly available data in client-side storage technologies.

Table 2

Comparison of Client-Side Storage Contents

Technology	Storage Content
Cookies	Advertisement trackers, user configurations, user preferences
Web Storage	Client related data such as identifiers and saved configurations, time stamps, logs, limited application data such as usernames
Session Storage	Temporary content related to single use of the application such as selections from previous pages
IndexedDB	Client related data such as identifiers and saved configurations, application data such as instant messages and usernames, time stamps, logs
Cache API	Application related data that requires large storage such as high-resolution game data

IndexedDB

IndexedDB is a persistent NoSQL transactional database technology that takes advantage of client-side storage for web applications. It is fast and highly efficient since B-trees are heavily utilized in its structure. B-trees enable fast manipulation of data on databases of considerable size (Ferragina & Grossi, 1999). With IndexedDB, entire databases can be employed for each web origin in a consistent structure even within different platforms. This consistency is ensured by the specifications shared by W3C (World Wide Web Consortium). Currently, W3C published three versions of IndexedDB with the latest one; IndexedDB API 3.0 being released in 2021 (*Indexed Database API 3.0*, n.d.) Moreover, it was highly adopted by major web browsers in a relatively short amount of time. Table 3 presents the dates major web browsers started their support for IndexedDB technology. These browsers constitute over ninety percent of the desktop-

based web browser market (*Browser Market Share,* n.d.). The implementation technologies of the browsers are also listed in Table 3.

Table 3

IndexedDB Support and Technology in Web Browsers

Browser	**Support**	**Underlying Technology**
Google Chrome	2012	LevelDB
Mozilla Firefox	2011	SQLite
Microsoft Edge	2015	LevelDB
Opera	2013	LevelDB
Internet Explorer	Only partial support	.dat file format

SQLite is prevalently seen in the preponderance of storage technologies including the ones providing fundamental functionality to the browsers such as history and bookmarks. Similarly, it is the underlying technology for IndexedDB in Firefox browsers. Contrarily, Google Chrome constructed IndexedDB over LevelDB technology (Lin, 2015; Liu et al., 2020). In various benchmark and experimental research, LevelDB proved more efficient and secure compared to SQLite in terms of fast operations on key-value pairs and batch updates (Luo et al., 2016). LevelDB keeps storage files with two key extensions, namely ldb and log, in this location. The rest of the files are subsidiary and aid ldb and log extension files in keeping versions and exchanging information. The .log files keep the most recent information. When .log files reach a size limit, the data is passed to .ldb files and a new version .log file appears. Google Chrome's implementation of LevelDB deletes the .log files after the browser is terminated (*Hang on! That's Not SQLite! Chrome, Electron and LevelDB*, 2020). However, several methods will be

discussed in this dissertation where this security mechanism shows insufficiency in implementation. As it is seen in Table 3, the pioneering implementation of Google Chrome has been adapted through most of the modern browsers for IndexedDB, making SQLite the second most popular implementation. The API standards of IndexedDB published by W3C provide standard management for the developers through JavaScript code. In other words, even though the underlying technology of the IndexedDB might differ in browsers, its control is identical.

There is a level hierarchy in the IndexedDB API. The highest level in this hierarchy is a database. All databases are associated with a version that helps the server keep the storage up to date with updates. If developers introduce updates to the database, a function called onupgradeneeded is called to regenerate the database and its lower-level structures. Subsequently, databases contain object stores that are analogous to tables in traditional databases (*Using IndexedDB*, n.d).

The initiation process of IndexedDB includes three steps including establishing a database, creating an object store, and populating it with data. A data retrieval is similarly easy with the utilization of the read function over transactions. The function access is handled with onsuccess and onerror functions. The function access is handled with onsuccess and onerror functions. These functions define what operations will be performed on the data when it is accessed successfully and what actions will be performed when an error is encountered.

Table 4 demonstrates the initiation process through JavaScript code according to the API 3.0 specifications, which is the same process for all web browsers (*Indexed Database API 3.0*, n.d.).

Table 4

Initiation of Database

Step	Code
1	*// Establishing a database* const firstRequest = indexedDB.open("IndexedDBDemonstration"); let DemonstrationDatabase; *// Onupgradeneeded is called when database does not exist or an upgrade is needed.* firstRequest.onupgradeneeded = function() { const DemonstrationDatabase = firstRequest.result;
2	*// Creating an object store. keyPath defines how the data will be indexed* const firstStore = DemonstrationDatabase.createObjectStore("users", {keyPath: "id"}); *// Adding additional index* const nameIndex = firstStore.createIndex("by_name", "name", {unique: true});
3	*// Adding data* firstStore.put({name: "John Doe", id: 120134}); firstStore.put({name: "Jane Doe", id: 120135}); };

Similarly, Table 5 demonstrates the process of data retrieval in three steps including establishing a transaction, creating a request, and handling the return.

Table 5

Data Retrieval

Step	Code
1	*// Establishing a database* const firstTransaction = DemonstrationDatabase.transaction("users", "readonly"); const secondStore = firstTransaction.objectStore("users");
2	*// Creating a request* const secondRequest = secondIndex.get("John Doe");
3	*// Handling the returned data or error* *// Without error* secondRequest.onsuccess = function() { const ourResults = secondRequest.result; }; *// With error* secondRequest.onerror = function(event) { *// Handling the error* };

Structure and Presentation

The structure of this dissertation aims to answer the research questions on three levels. First, the general characteristics of IndexedDB storage and its artifacts are evaluated on the most prevalent websites of US-listed by Alexa. Second, case-specific studies are conducted where the extraction, processing, and presentation of the artifacts are covered for both implementations of LevelDB and SQLite. LevelDB is covered with Google Chrome, and additionally with a desktop application. Table 6 shows level two studies and the technologies they scrutinize.

Table 6

Studies in Second Level of Dissertation

Case Study	Implementation	Browser/Application	Tool Implementation
WhatsApp Web	LevelDB	Google Chrome	Yes
Microsoft Teams	LevelDB	Desktop Application	No
Instagram	SQLite	Mozilla Firefox	Yes

The case studies of WhatsApp Web, Microsoft Teams, and Instagram was mainly determined based on the preliminary results obtained from the experiments of the first level. Additionally, the key terms ".ldb", ".sqlite" were searched on the Windows 10 computer, utilized for the first experiment, to discover additional applications that utilize IndexedDB storage. Microsoft Teams desktop application was determined as the only application besides web browsers to utilize IndexedDB storage. Therefore, it was added as a case study. In the third level, a method is suggested for the verification of traditional web browser artifacts with IndexedDB storage. Accordingly, experiments aim at implementation and the verification of the suggested techniques.

The hypotheses which are created to answer the research questions are given in the methodology of their levels in accordance with the aim of their experiments. The rest of the sections are divided into the literature review, methodology, results, discussion, and conclusion.

CHAPTER II

Literature Review

This chapter gives prior work on six sub-sections. In the first sub-section, the work on persistent storage and IndexedDB are presented. In the second, third, and fourth sub-sections, case-specific studies on WhatsApp, Microsoft Teams, and Instagram forensics are put forth. In the fifth and final sub-section, the work on verification of web browser forensic artifacts is provided. Finally, a conclusion on the adequacy and future needs of the academic literature are discussed.

Review of Prior Work on Persistent Storage and IndexedDB

The major source of information on the IndexedDB implementation is the specification shared by W3C. A few studies shared information on the implementation of the technology and gave limited insight into its forensic value (Bhattacharya & Gwizdka, 2021). However, to the best of our knowledge, no single study comprehensively discusses the implementation details of IndexedDB for different browser vendors across various operating system platforms and the utilization of the technology for web forensic analysis. Even though several studies addressed the performance of IndexedDB (Al-Shaikh & Sleit, 2017), most of the existing studies on the IndexedDB focused on the security aspects of the technology (Kimak et al., 2014).

Bhattacharya and Gwizdka (2021) developed a tool called YASBIL (Yet Another Search Behavior Interaction Logger). The tool took advantage of Mozilla Firefox browser extensions and the WordPress plugin to collect information from IndexedDB storage. As browser extension was available for users, the WordPress plugin was required to be installed in a central data repository. The tool was beneficial for the user to obtain data

from its own browser. However, this is not the case for forensic investigators who are required to bypass the SOP (Same Origin Policy) of web browsers (*Same Origin Policy - Web Security*, n.d.).

Al-Shaikh and Sleit (2017) made a comparison of IndexedDB performance in Google Chrome, Mozilla Firefox, Internet Explorer, and Opera web browsers. The comparison was based on a set of operations including the execution of commands that created databases on top of reading, writing, and deleting data in the IndexedDB storage. The study determined Internet Explorer as the top performance browser followed by Mozilla Firefox, Opera, and Google Chrome. The work was beneficial as it conducted actual operations on the web browsers. However, the forensic aspect of the storage was not covered.

Other researchers (Kimak, 2016; Kimak et al., 2014; Kimak & Ellman, 2015) provided valuable information on IndexedDB; however, information on the implementation details is limited. Specifically, the locations and file types used by Firefox and Chrome browsers were discussed, while the file structures, naming schemas, and the details of different operating systems were not studied. Additionally, as IndexedDB 2.0 specifications were released by W3C in January 2018, the specifications in the studies were already outdated. For example, the locations of the storage files given for the two browsers in the previous studies do not remain the same as before.

Singh and Singh (2017) performed a specific study on the forensics of the Windows 10 Cortana. The study focused only on the user interaction with Windows Cortana, which was recorded in the Windows Edge's IndexedDB storage file,

IndexedDB.edb. Therefore, this work did not provide the specifics of the IndexedDB technology.

As discussed in the limited cases, the existing implementations of web investigation tools are utilized to provide insights into the IndexedDB contents of the websites. However, these implementations are insufficient for detailed IndexedDB analysis as the target investigation functions are only implemented as supplementary extensions to different properties of the targeted applications. Also, with the release of the Indexed Database (commonly referred to as IndexedDB) API 2.0, a recent change in the implementation specifications has left many applications outdated and incomplete, although some applications partially provide insights into the IndexedDB contents of the websites exist (Basques, 2019).

As the IndexedDB technology has not been yet implemented to its fullest for most of the popular websites, the application and implementation details that the existing studies discussed are limited. Furthermore, no tool that specifically targets the artifacts of the technology is publicly available yet. Although the widely used forensic tools such as FTK, Encase, and Autopsy can read the random bits of the IndexedDB SQLite files, they do not provide a specific protocol to decode the contents in IndexedDB files. The same argument can be made for SQLite tools as well. Therefore, it is necessary to employ further updated forensic analysis functionalities that can help comprehensively extract latent information from the latest web storage technology.

Review of Prior Work on Microsoft Teams Forensic Investigations

A limited number of studies examined Microsoft Teams in terms of digital forensic investigations and application artifact value (Kim & Kwon, 2021; Nicoletti & Bernaschi, 2021). Only a limited subset of these studies touched persistent storage artifacts without full coverage (Kim & Kwon, 2021). At the same time, its security and privacy concerns were covered in several papers with regards to malicious user content (Singh & Awasthi, 2020), security policy sufficiency (John, 2020), and network security (Gauthier & Husain, 2021).

Nicoletti and Bernaschi (2021) presented different utilization scenarios of Microsoft Teams that digital forensic investigators might encounter during their examinations. These scenarios included situations where the user makes phone calls from Microsoft Teams to a mobile phone. The forensic investigation approach suggested by the authors involved the utilization of Microsoft Teams Admin Center, and Microsoft 365 security compliance functionalities to obtain analytic reports of the application's utilization. Additionally, information from a Wireshark and Syslog server was added to the data for a better understanding of the artifacts. The paper concluded that forensic analysis can be efficiently performed on Microsoft Teams with the utilization of services provided by Microsoft 365. The methodology presented by the authors involves access to an administrative unit that needs to be set up beforehand to investigate artifacts particularly related to VOIP utilization of the application. This approach highly differs the paper from this dissertation in terms of technologies and artifacts under investigation.

Kim and Kwon (2021) inspected Windows and Android artifacts of Microsoft Teams. The study created a complimentary analysis with configured timeline structure

for the Microsoft Team Desktop Application. However, it is not available for non-Korean readers. Additionally, the classification of behavior given for Microsoft Teams users did not appear to include setting scheduled meetings which is a big indicator of collective behavior in cooperation. Even though snippets of material from persistent storage technologies can be seen in figures and tables, no indication of voice mail or account configurations appeared to be shared in detail. Paper constituted as a good source of material for a comprehensive look into the investigations of Microsoft Teams. However, it does not cover the IndexedDB artifacts in-depth and structure which distinguishes it from this dissertation.

Singh and Awasthi (2020) gave a general look into the security of video conferencing platforms including Microsoft Teams. It presented overall security mechanisms such as two-factor authentication, single-sign-on, and the protection provided by the active directory. Another aspect that was put under consideration was the security threats that could emerge from the malicious user content sent and uploaded by the users. It was pointed out that Microsoft Advanced Threat Protection (ATP) safe links were not provided in the applications even though a technology adaption program was reviewing the case for improvement.

John (2022) considered the privacy issues of Microsoft Teams. Specifically, the privacy of users in all major video conference applications, including Microsoft Teams, was questioned. It is mentioned that video conference applications utilize the security policies of their providers, which in this case is Microsoft for Microsoft Teams. Several resources such as the Digital Lab of Consumer Reports were addressed to point out that the security of Microsoft Teams lacked important elements. A set of criteria published in

the pointed resources for the privacy of users were shared. These criteria include the basics of handling user information such as its collection, protection, deletion, and access. The paper concludes with a set of strategies suggested to users to better protect their information on major video conference applications.

Gauthier and Husain (2021) provided work on dynamic analysis of Microsoft Teams along with Zoom and Google Meet. VOIP calls were simulated in a virtual environment and network packets were captured for analysis. The paper concluded that the encryption standards of the applications were satisfactory to protect data from interrupting parties. However, privacy concerns were raised as the encryptions were not end-to-end and vendors could easily reach user data.

Predominantly, the research on digital forensics of IndexedDB is limited and exclusively focused on web browser artifacts. Whereas the papers targeting Microsoft Teams focus primarily on its privacy and overall application security. Therefore, the examination of IndexedDB technology as a digital forensic data source for applications other than web browsers is a significant requirement for academia.

Review of Prior Work on Instagram Forensic Investigations

Instagram web browser artifacts left on personal computers were covered in a notable number of studies (Chang & Yen, 2019; Ghafarian & Keskin, 2020; Jadoon et al., 2019). Additionally, Instagram forensics was the subject of several research papers with a broader scope of social media forensics. These papers covered various aspects such as network traffic analysis (Walnycky et al., 2015), mobile application investigations (Mushcab & Gladyshev, 2015; Pambayun & Riadi, 2020), user behavioral analysis (Seo

et al., 2018), digital image/video analysis (Douglas, 2018), and criminal impersonation (Zarei et al., 2019).

Chang and Yen (2019) conducted a study that analyzes the Instagram artifacts on Android and Windows 10 platforms. It further diversified the artifacts with the examination of different web browsers. A broad range of artifacts from pictures to user activities such as liking, commenting, and tagging was analyzed. The study pointed out that the artifacts available in different browsers show significant differences. This is attributed to the variant privacy mechanisms of web browsers. Despite covering valuable sources such as log files, history records, and temporary network file artifacts, the study does not provide considerable information on persistent web storage technologies such as IndexedDB.

Ghafarian and Keskin (2020) had a particular forensic focus on the artifacts of Facebook and Instagram obtained from Windows hibernation files. A detailed discussion is provided for social media forensics in the cases where it is not possible to obtain information from the suspect's hard drive. The paper shares detailed techniques for creating a connection between the suspect and the extracted artifacts. However, IndexedDB and similar persistent storage technologies are not covered in the study.

Jadoon et al. (2019) scrutinized the forensic resistance of tor browser in a Windows virtual environment. The experiment utilized a list of activities to populate data that included Instagram user activities such as following, visiting accounts, and liking pictures. The artifacts of visited links were successfully recovered while artifacts from comments and liked pictures could not be recovered. The examined technologies are

listed as the registry and main memory of the virtual drive which did not include persistent storage technologies.

Walnycky et al. (2015) acquired and analyzed artifacts from network packets and android end devices utilizing content from twenty different applications including the Instagram android application. The research aimed at a full or partial reconstruction of data artifacts. The images sent from the Instagram application were successfully intercepted in the network traffic. This paper covers only the android application of Instagram and does not include information about the persistent storage analysis.

Mushcab and Gladyshev (2015) investigated the efficient techniques to eliminate challenges of forensic analysis of Instagram and Path social media applications on iPhone devices. The process included a complete process of data population, image acquisition, and automatic examination. It is concluded that the internal memory of the iPhone 5s device contained artifacts from the social media applications even though they are not installed in the internal memory.

Pambayun and Raidi (2020) particularized the mobile investigations of Instagram on android devices. The research applied the investigation stages of the Digital Forensics Research Workshop (DFRWS) with the utilization of the OXYGEN mobile forensics tool and JSON Viewer. The paper concludes on the sufficiency of the forensic tools to obtain artifacts from chat sessions which are extracted from both internal and external memory. The paper presented a systematic look into the android device investigations of Instagram. However, the shared methodology was slightly short on detail to provide an extension to other platforms.

Seo et al. (2018) called attention to the information exposure on Instagram pages with potential threats such as impersonation, fraud, and copyright violation. A reverse engineering method is applied to obtain personal information from Instagram pages and data sources such as cache and SQLite files. Several tools including ADB (Android Debug Bridge) and DB Browser for SQLite was utilized during the examination. It was concluded that the user behavior on Instagram can be analyzed by arbitrary users with little effort. The applied method is also suggested as a digital forensic examination technique.

Douglas (2018) conducted a detailed investigation of the images handled by Instagram. Various best practices such as the selection of the image quality during extraction are shared to provide a framework for obtaining reliable data. A cross-platform experiment, questioning how an image from an android device would look when uploaded to an Apple device, was made available as supplementary data. As the primary outcome, the essentials of a set of techniques to identify whether an image is original and authentic or downloaded from Instagram were laid out. Additionally, the characteristics of Instagram processed images were presented as a framework for future research.

Zarei et al. (2019) analyzed the impersonators targeting Instagram users to construct a technique of impersonation identification. A large data is utilized with both fake and genuine accounts where users were divided into clusters based on their profile characteristics. The paper presented findings that indicated a considerable size of political interest in bot-like clusters. It is also discussed that the interest of bot-like clusters in news agencies and sports stars were not significantly different.

The overall focus of research on the Instagram web and mobile application forensics is diversified in miscellaneous aspects including both user interaction and data artifacts. However, to the best of our knowledge, there is not any work conducted on IndexedDB and Instagram which can be a practical guide for cases under investigation. Therefore, systematic research on the persistent storage artifacts of Instagram and their reflected value as digital evidence based on web browser IndexedDB technology is indispensable.

Review of Prior Work on WhatsApp Forensic Investigations

While several research studies scrutinized the applicability of forensic techniques and tools on the WhatsApp mobile application (Anglano, 2014; Shortall & Azhar, 2015) and its network investigations (Karpisek et al., 2015; Umar et al., 2017), fewer studies focused on WhatsApp Web forensics (Karpisek et al., 2015). Additionally, several of the research studies with a broader scope of instant messaging applications touched upon forensic analysis of WhatsApp (Actoriano & Riadi, 2018; Mahajan et al., 2013; Sgaras et al., 2015).

Anglano (2014) investigated the artifacts left on Android devices by the WhatsApp Messenger application. The author considered all the artifacts left by WhatsApp Messenger and showed a correlation of information merged from various artifacts. The acquired data reflected the activities of adding/deleting a contact, sending messages, and the feedback on the delivery of the messages. The scope of this research was limited to WhatsApp Messenger on Android devices, while WhatsApp Web and iOS systems were not addressed.

Thakur (2013) provided an outline of a methodology for forensic investigations of WhatsApp on Android phones. The research covered volatile memory acquisition as an addition to static external storage acquisition. The research utilized the WhatsAppXtract tool on the SQLite database in the backed-up folder without rooting the android device for non-volatile memory acquisition. Whereas, a memory dump utility, Memfetch, was used for the extraction of information from the volatile memory. A tool called whatsappRamXtract was introduced for the analysis and presentation of the extracted data. It was argued that the data analyzed was critical and can be forensically significant.

Shortall and Azhar (2015) inspected WhatsApp mobile application for forensic investigations on iOS 8.3, Android Lollipop 5.0.1, and Windows Phone 8.1 utilizing the forensic investigation tools of EnCase, UFED, and Oxygen Forensic Suite. The authors explicated that contact information was recovered from iOS and Android with UFED and Oxygen Forensic Suite. However, nothing was recovered from Windows Phone 8.1. It is discussed in the paper that the security mechanisms employed by Windows Phone 8.1 made it very difficult to perform a traditional forensic acquisition. As a result, the paper suggested live data forensic investigations to recover WhatsApp artifacts from Windows Phone 8.1 operating systems that were not accessible with traditional techniques.

Karpisek et al. (2015) conducted research on the decryption of the information transferred during WhatsApp calls. The procedure included decryption of network traffic, obtaining artifacts of the call, and the development of a Python tool to convert obtained Wireshark dump files to HTML format. The research stated the artifacts that can be obtained from the WhatsApp calls as phone numbers, duration of the call, IP addresses of the server, and the time of call termination.

Umar et al. (2017) used NIST measurement parameters to evaluate the forensic investigation tools specializing in Android phones to compare their performances on WhatsApp forensics. Nine core assertions, seven optional assertions along with six-core requirements, and optional feature requirements of NIST measurement parameters were utilized in the scope of the work. The research identified Belkasoft Evidence as the tool that scored the best among other tools, whereas WhatsApp Key/DB Extractor came out as the most cost-efficient, and Oxygen Forensic as the best at obtaining the artifacts.

Actoriano and Riadi (2018) gave a forensic investigation framework for WhatsApp mobile and WhatsApp Web browser applications. Mobile information extraction technique used for the framework involves extracting the encrypted message storage file of WhatsApp conversations. Similarly, web application information extraction involves using FTK Imager to obtain Google Chrome SQLite Database files for history, cache, and web session information. The persistent storage information including IndexedDB storage is not included in the research.

Mahajan et al. (2013) utilized internal memories of Android phones for the forensic investigations of WhatsApp and Viber. UFED (Universal Forensic Extraction Device) was used for file extraction and the Sqlite database browser was used for the investigation of the data. The research was conducted on five android phones with three different operating systems installed; Froyo 2.2, GingerBread 2.3, and Ice-Cream Sandwich 4.0. The authors discussed in the paper that based on the logs and history files investigated, forensically significant data was detected for WhatsApp and Viber in all different operating systems of android phones.

Sgaras et al. (2015) presented evidence collection methods for forensic investigations on WhatsApp, Viper, Skype, and Tango instant messaging applications. The research was performed on both iOS and Android operating systems. The definition of types of artifacts that can be found in the four instant messaging applications was listed along with the methodologies of extractions.

Overall, research on WhatsApp security and forensics is mostly focused on mobile devices and the IndexedDB only has a handful of research conducted but not WhatsApp specifically as it is still an emerging technology. Therefore, research on WhatsApp Web and its forensic investigation based on web browser IndexedDB technology is a dire need.

Review of Prior Work on Verification of Web Browser Artifacts

Various authors have conducted experiments on web browsers and their digital artifacts for the last several decades (Nalawade et al., 2016; Oh et al., 2011; Patil & Meshram, 2019; Rathod, 2017). In this scope, a broad area of topics has been covered. Including web browser investigation tools (Mahaju & Atkison, 2017) and specialized issues such as portable browser investigations (Marrington et al., 2012). Whereas persistent storage technologies were inspected in specific only by a limited number of research papers since their usage became prevalent (Mendoza et al., 2015). Furthermore, research on the verification of web browser artifacts is very limited (Graeme, 2020).

Oh et al. (2011) have listed the existent web browser forensic tools and suggested that they were not sufficient for efficient web browser forensics. A list of criteria was given and demonstrated with a suggested methodology that integrates artifacts from different browsers. Internet Explorer, Firefox, Chrome, Safari, and Opera web browsers

were employed in the demonstration of the suggested techniques. Paper outlined its methodology around the construction of timeline analysis, search history, and URL encoding of visited websites. These artifacts were merged with user activity analysis that incorporated the content of the visited sites with the frequency of their visits. A tool called WEFA (Web Browser Forensic Analyzer) was utilized in the demonstration process. The paper lacks incorporation of persistent storage technologies which was not prevalently used in its time.

Nalawade et al. (2016) made a comparison of the web browser forensic tools in form of a short survey study. Private browsing and anti-forensic techniques were taken into account and their analysis was pointed out as a critical function. Additionally, the importance of keyword and regular expression search in digital forensics was emphasized in terms of investigation efficacy. The paper suggests that the WEFA tool was able to deliver more artifacts even when the browsers were operated in their private browsing modes.

Rathod (2017) conducts a study focused on Google Chrome browsers in Windows, Linux, and MAC operating systems. The analysis of history, downloads, visited URLs, cookies, login information along prefetch files were included in the paper. Additionally, information about how to recover deleted data regarding the files of the storage units was discussed. The recovery technique relies on the carving of the deleted files and reconstruction from prefetch locations. Even though cookies were included in the paper, advanced persistent storage technologies such as Web Storage, and IndexedDB were not covered.

Mahaju and Atkinson (2017) made an evaluation of the digital forensic tools for the Mozilla Firefox browser. Tools were checked against criteria such as memory utilization, CPU utilization, and artifact processing time. The paper gave detailed information about the web browser forensic process with storage file locations and artifact placement. The tools were compared in a broad section of features including user visit count of websites, bookmarks, login information, and time-frame information. The listed sources and features did not include advanced persistent storage technologies of Web Storage and IndexedDB.

Marrington et al. (2012) covered the forensic investigation of portable web browsers in both cases of initiation from USB drives and installation in computer hard drives. A set of browsing activities were performed on Google Chrome and the disks were imaged with FTK imager after the experiment. The experiment included a short but common set of user activities including watching videos on YouTube, making image searches on Google, and browsing items on the eBay online sales platform. It also incorporated a comparison of the artifacts created by the portable operation of Google Chrome to the artifacts extracted from its regular operation. It was discussed based on the obtained results that there are artifacts left in portable forms of web browsers. Therefore, the privacy of the users is not high tent with the usage of portable browsers. The paper only covered only Google Chrome browser and did not provide verification of the artifacts with alternate web browser technologies.

Mendoza et al. (2015) scrutinized the forensic investigations and remnants of persistent storage technology specifically for Web Storage technology of web browsers. The paper investigated the usage of Web Storage technology in prevalently used websites

and major web browsers such as Google Chrome, Internet Explorer, Opera, Mozilla Firefox, and Safari. It provided a proof-of-concept tool to demonstrate the extraction and processing of the Web Storage artifacts in Windows operating systems. Even though a comprehensive analysis of Web Storage is given in the paper, it lacks the content of the IndexedDB. Additionally, the artifacts are not utilized for verification of the traditional digital forensic artifacts.

Graeme (2020) took attention to the lack of quality assurance systems in digital forensic investigations. The concept of fast-changing technology and the potential of missing digital artifacts were stressed. A verification framework for digital evidence (VODE) based on the offense type was introduced. The paper based its framework on a concept called new knowledge scenario which deals with digital artifacts for the very first time. The system included the verification of new knowledge, determination of relevant digital traces, and reconstruction of events that occurred in the system. Even though the framework is a unique work on its target which is the verification of the digital artifacts with a systematic quality assurance system, it did not target persistent storage particularly. Therefore, the potential of persistent storage artifacts for verification was not laid out in this paper.

The research on the value of traditional digital artifacts and their utilization is predominant in literature. The specifics of these artifacts are prevalently available to digital examiners with tools covering extraction, processing, and limited verification techniques. However, the research on persistent storage artifacts and their connection to traditional web browser artifacts for verification is scarce.

Summary of Prior Work on Verification of Web Browser Artifacts

The previous sections laid out the previous work on IndexedDB artifact value and utilization. Additionally, the studies on the forensic investigations of the particular origins, where IndexedDB can be employed, were presented. It can be observed that the persistent storage technologies and their combination with regular browser artifacts are missing pieces of the literature. Table 7 summarizes the coverage of previous papers on IndexedDB and how this dissertation distinguishes its target field.

Table 7

Summary of Previous Studies and Their Coverage of IndexedDB

Academic Study/Paper	**IndexedDB Security**	**IndexedDB Performance**	**IndexedDB Forensic Value**	**Artifact Verification Through IndexedDB**
Bhattacharya and Gwizdka (36)	Not Covered	Not Covered	Partially Covered	Not Covered
Kim and Kwon (45)	Not Covered	Not Covered	Partially Covered	Not Covered
Singh and Singh (42)	Partially Covered	Not Covered	Partially Covered	Not Covered
Kimak (41)	Covered	Covered	Not Covered	Not Covered
Kimak et al (37)	Covered	Covered	Not Covered	Not Covered
Kimak S, Ellman J. (16)	Covered	Covered	Not Covered	Not Covered
Al-Shaikh and Sleit (20)	Not Covered	Covered	Not Covered	Not Covered
This Dissertation	Covered	Covered	Covered	Covered

CHAPTER III

Methodology

This chapter provides the methodology of experiments that are conducted for population, extraction, evaluation, presentation, and utilization of data for preliminary verification of IndexedDB artifacts. First, the evaluation of IndexedDB popularity among major websites and the level of its utilization are targeted for assessment. The type of information kept in the IndexedDB storage is also included as a target in this initial scrutinization. In this scope, basic operations in various settings were designed for top websites ranked by Alexa. Subsequently, single case pretest-posttest quasi experiments were designed to populate and extract data in a more detailed and targeted fashion. These experiments target popular sites of WhatsApp Web, Instagram, and the desktop application of Microsoft Teams. Finally, techniques are suggested for the utilization of IndexedDB as a verification source. The methods of verification of the suggested methodologies are presented at the end of the chapter.

Methods for Determination of Level and Characteristics of IndexedDB Artifacts in most Popular Websites

Prior to the investigation of cases in-depth, the targets of the experiments are determined by assessing the level of use and characteristics of IndexedDB artifacts in the fifteen most popular websites listed by Alexa (*Top Sites in United States*, n.d.). The list provided by Alexa is constantly updated with changing web user behavior. However, most changes apply to the ranking of the websites within the top fifteen. Therefore, only a few changes are introduced to the entire list of websites making it to the top fifteen.

In order to obtain the populated IndexedDB contents of the websites, each website is subjected to a set of operations, which include access, registration, authentication, communication, and specialized operations. Access operation is basic access to a website through a browser. Registration is the creation of a specialized record on a website, such as a user and account entries. Authentication is private access to a website with the information provided in the registration operation. Communication is any kind of interaction from private messaging to public comments. Specialized operations are website-specific actions such as registering items to chart, moving to checkout, and opening a streaming media. The operations are performed where they are available. For example, communication is not an option for Netflix whereas it dominates most actions on social network websites. Table 8 presents the operations performed on the most popular fifteen websites.

Table 8

Preliminary Operation Categories Performed on Top 15 US Websites

Rank	Website	Operations	Specialized Operations
1	Google.com	Access	Searching Keywords
2	YouTube.com	Access	Streaming Videos, Leaving Comments
3	Facebook.com	Access, Registration, Authentication, Communication	Displaying Media, Streaming Videos, Posting Media
4	Amazon.com	Access, Registration, Authentication	Searching Items, Adding Items to Shopping Chart
5	Reddit.com	Access, Registration, Authentication	Searching Keywords, Displaying Media, Streaming Videos
6	Yahoo.com	Access, Registration, Authentication	Searching Keywords
7	Wikipedia.org	Access	Searching Keywords

(continued)

Rank	Website	Operations	Specialized Operations
8	Twitter.com	Access, Registration, Authentication, Communication	Displaying Media, Posting Media
9	Instagram.com	Access, Registration, Authentication, Communication	Displaying Media, Posting Media
10	LinkedIn.com	Access, Registration, Authentication, Communication	
11	eBay.com	Access, Registration, Authentication	Searching Items, Adding Items to Shopping Chart
12	Netflix.com	Access, Registration, Authentication	Streaming Videos
13	ESPN.com	Access	Streaming Videos
14	Twitch.tv	Access	Streaming Videos
15	Microsoftonline.com	Access, Registration, Authentication	Searching Keywords

The purpose of the operations in this section is to populate the IndexedDB storage of websites to inspect the level of utilization. More detailed experiments with detailed operations are performed on sources with a promising level of utilization.

For the assessment of the consistency of artifacts in different web browsers, all listed operations are performed in Google Chrome, Mozilla Firefox, Opera, Internet Explorer, and Microsoft Edge browsers. All browsers keep their storage files in different locations with several underlying technologies as listed in Table 3. LevelDB technology utilized by Google Chrome, Opera, and Microsoft Edge browsers gives user-readable information blended with binary data in .log files. These files persist till the storage capacity is reached. Subsequently, their contents are transferred to .ldb files. All the operations of the experiments performed in this stage are present in .log files. SQLite

artifacts of Mozilla Firefox browser blended with binary data are viewed with SQLite Browser (*DB Browser for SQLite*, n.d.). More effective methods of extraction of each technology are given in later sections of the dissertation.

Methods and Experiments for Microsoft Teams IndexedDB Investigations

Experiments in this section are focused on LevelDB implementation of the Microsoft Teams desktop application and the artifacts it contains about the collaborative actions of corporate users. Accordingly, the techniques were constructed around two major hypotheses.

Hypothesis 1 (H1). Microsoft Teams artifacts populated in IndexedDB storage can be efficiently deployed for digital forensic investigations.

Hypothesis 2 (H2). Microsoft Teams artifacts extracted from IndexedDB storage can be utilized to construct time-frame analysis.

Experimental Design

The accuracy of the hypotheses is checked with a Single Case Pretest–Posttest Quasi Experiment formed with the principles described by Cook and Campbell (Cook & Campbell, 1979). Correspondingly, a set of measurements are applied before and after the application of a treatment that is designed to populate the IndexedDB storage of Microsoft Teams. Based on the differences observed from the outcomes of the measurements, the artifacts that are exclusively populated by the treatment are identified. The following stages of operations are applied within the scope of the experiment.

Pretest: Data in the IndexedDB storage is extracted and evaluated with the intention to obtain the artifacts inherently present in the storage location.

Treatment: A set of Microsoft Teams use cases, created from the observed user behavior, are carried out to populate artifacts in the IndexedDB storage.

Post-Hoc Test: Data is extracted one more time after the application of the treatment and compared to the artifacts of the pretest to isolate the artifacts resulting from the treatment.

The subject of this experiment is the Microsoft Teams v1.4.00.29469 (64-bit) desktop application. The experimental environment incorporates two HP Pavilion 15 laptops installed with Windows 10 Home v20H1-2004 b10.0.19041.1348 Operating System (PC1 and PC2) and one Galaxy 20 FE installed with Android OS v11 (Phone1). The experimental environment is calibrated with the following steps.

- Three different Microsoft 365 accounts, Account1, Account2, and Account3, are created from the Microsoft 365 management center with a corresponding name and last names of “account one”, “account two”, and “account three”.
- Windows 10 Home operating system of PC1 and PC2 is formatted and logged in with the credentials of Account1 and Account2.
- Microsoft Teams desktop application is installed on PC1 and PC2.
- Microsoft Teams android application is installed in Phone1

Pretest

The artifacts that are present in the Microsoft Teams IndexedDB storage of PC1 are tested before the treatment is carried out. The pretest procedure is performed with the following procedure.

- In PC1 Microsoft Teams desktop application is initiated without log-in
- The application is left idle without any user interactions for five minutes.

- The artifacts are extracted from the storage of the application utilizing the techniques described in the extraction and measurements section.

Treatment

A set of use cases were designed as the treatment to populate the artifacts in the IndexedDB storage. These use cases were determined based on the expected user interactions with the application. The user behavior of three volunteers was observed during two days of their regular work period to construct the expected behavior. Whenever the volunteers utilized the application, they recorded their usage with TechSmith screen recording application (*TechSmith Capture*, n.d.). These recordings were utilized to identify the most prevalent usage of the application. Activities listed in Table 9 constitute the majority of the overall activities performed with Microsoft Teams.

Table 9

Prevalent Activities Performed by Users in Microsoft Teams

No	Activity
1	Creating a team
2	Adding members to the team
3	Sending messages in private chat
4	Using emojis, and hyperlinks during a private chat
5	Sending documents, and media such as pictures and videos from private chat
6	Video calling a private member
7	Muting and unmuting microphone
8	Turning the video on and off
9	Changing background of the user during a video meeting
10	Sending private chat messages during a video meeting
11	Ending the video call

(continued)

No	Activity
12	Creating whiteboard for a team
13	Sharing documents to the whiteboard of a team
14	Creating a scheduled meeting for a team
15	Joining a team scheduled meeting with logged in computer
16	Sending private messages to a specific user in a scheduled meeting
17	Recording a meeting
18	Leaving a team meeting
19	Joining a team meeting as a guest user
20	Joining a team meeting from a mobile device
21	Sharing screen with members during a team meeting
22	Kicking users from a team meeting
23	Muting and unmuting users in a team meeting
24	Making calls to mobile phones
25	Receiving calls from mobile phones

The remaining actives were either user specific or associated with search and navigation within the application. The treatment procedure was created based on the given activities. Table 10 lists the treatment procedure steps for all accounts.

Table 10

Microsoft Teams Treatment Procedure Steps

Steps	Account1	Account2	Account3
1	Account 1 is logged in from PC1		
2	A new team is created with the name “ExperimentTeam1”, description “Team for Experiments”, and Privacy		

(continued)

Steps	Account1	Account2	Account3
	option: "Private - Only team owners can add members"		
3	Account2 and Account3 are added to ExperimentTeam1		
4	A private message with the content “This is message1” is sent to the ExperimentTeam1		
5	A private message with the content “Hyperlink1: https://www.microsoft.com/en-us/microsoft-teams/group-chat-software and Emoji1: <a smiley face>“ is sent to Account2		
6	Pic1 is sent to Account2 in a private message		
7	Pic1 is added to the files of the ExperimentTeam1		
8	WhiteBoard1 is added to the ExperimentTeam1 as a new tab		
9	A new note with content "WhiteBoard Message1"is added to WhiteBoard1		
10	A scheduled meeting with the title “Meeting1” and description “This is the description for Meeting1” is created with the weekly schedule of every Wednesday 2.30pm for ExperimentTeam1 by Account1 (content can be seen in Figure 2)		
11		Account2 is logged in from PC2	
12		Private messages sent from Account1 are displayed	
13		Private message "This is reply1" is sent to Account1	
14	A video call is requested to Account2	Video call is accepted by Account2	

(continued)

Steps	Account1	Account2	Account3
15	Account1 is muted and then unmuted by Account1	Account2 is muted and then unmuted by Account2	
16	The video camera is turned off and then turned on by Account1	The video camera is turned off and then turned on by Account2	
17	Chat message with the content “Message3-1: during video call” is sent to Account2	Message 3-1 is liked by Account2	
18	Video call is ended by Account1		
19			Account3 is logged in from Phone1
20	Meeting1 is started by Account1	Account2 is joined to Meeting1	Account3 is joined to Meeting1
21	Chat message with the content “Message4: during scheduled meeting” is sent to the attendees	Message4 is displayed by Account2	Message4 is displayed by Account3
22	The recording for Meeting1 is started by Account1		
23	The recording for Meeting1 is stopped by Account1		
24	Chat message with the content “Message4: during scheduled meeting” is sent privately to Account3		Meeting1 is left by Account3
25			Account3 will rejoin to Meeting1
26	Account2 is kicked from Meeting1 by Account1		
27	Meeting1 is ended by Account1		
28			Meeting1 is left by Account3
29			Account3 rejoined to Meeting1
30	Account2 is kicked from Meeting1 by Account1		

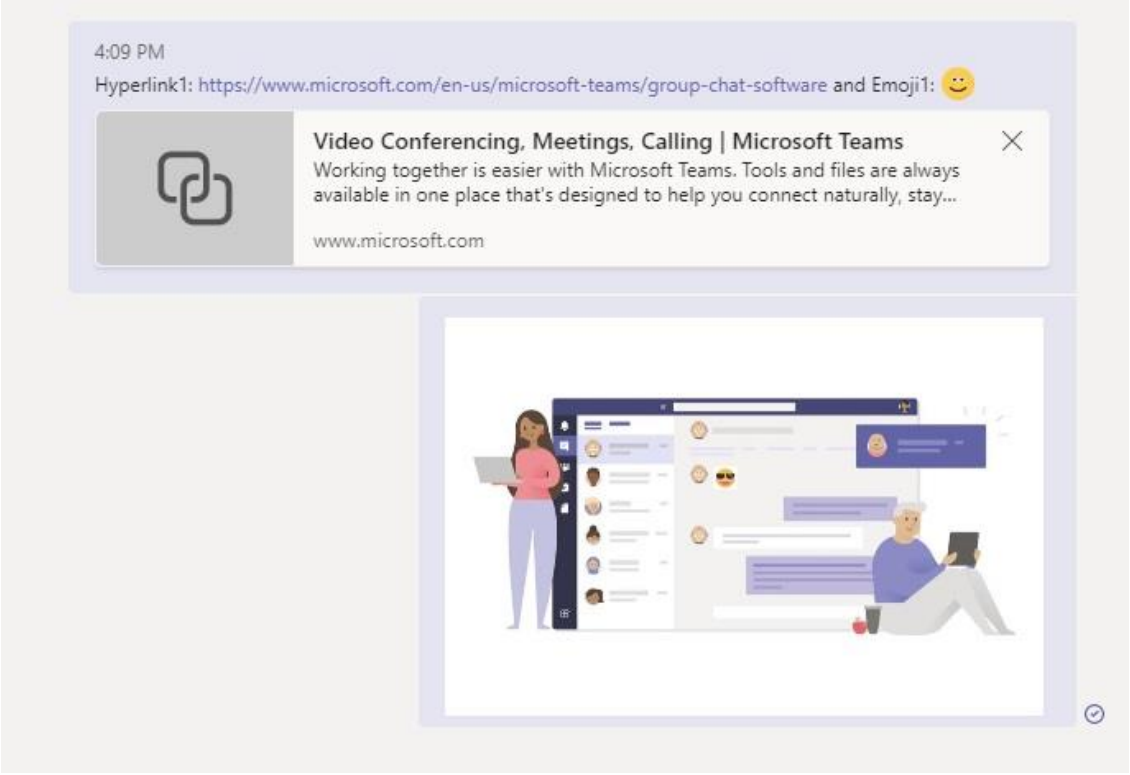
(continued)

Steps	Account1	Account2	Account3
31	Meeting1 is ended by Account1		
32	Account2 is added as a contact by Account1		
33	Voicemail is activated by Account1		
34		A call is initiated from Account2 to Account1 and voice message with content: "This is message one for Account1" is left in the voice mail	
35	Voice mail from Account2 is displayed as text by Account1		

Figure 1 displays the demonstration of the treatment steps five and six in detail.

Figure 1

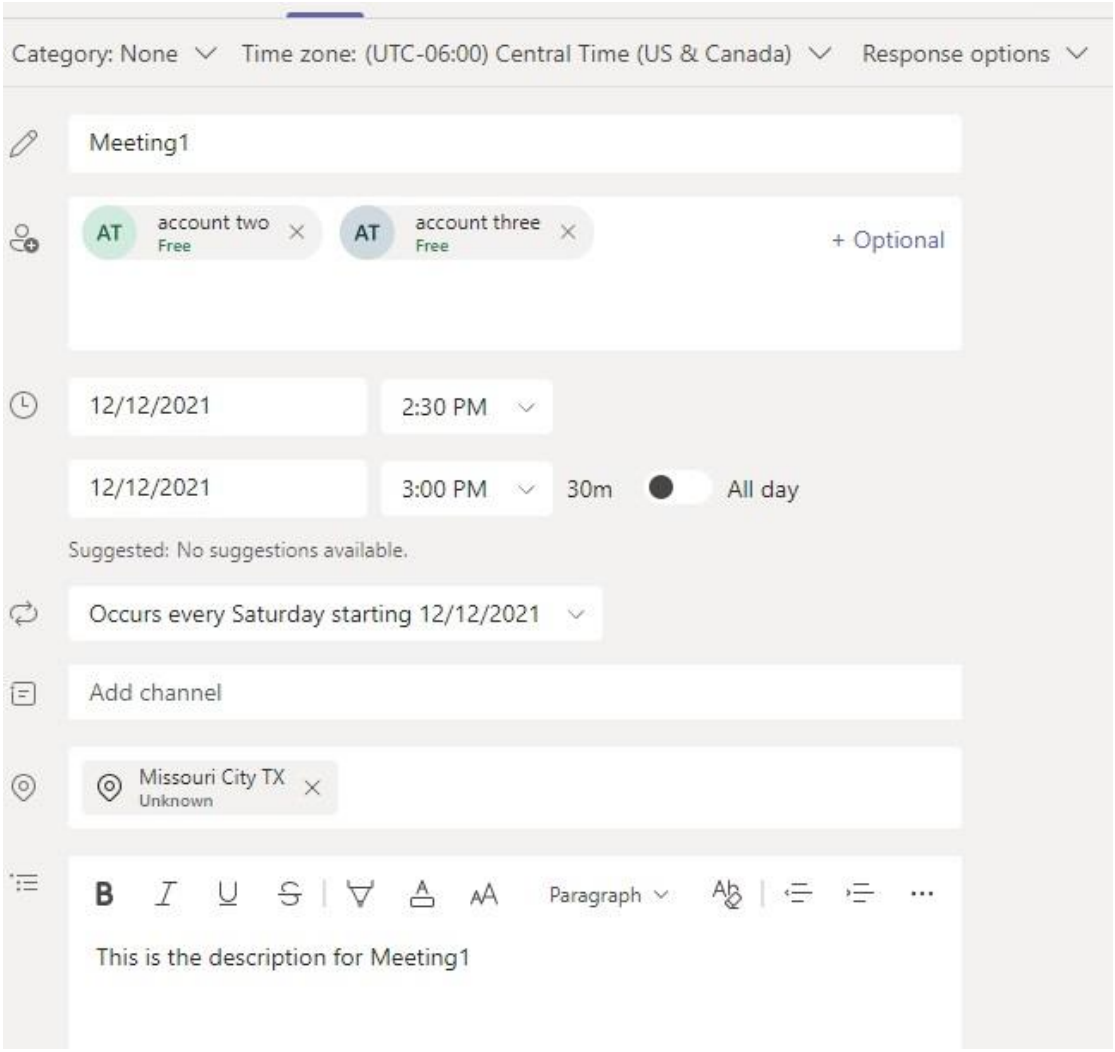
Message with Media Attachment



The private chat message, which is sent to Account2 includes a hyperlink and a smiling face emoji. Under the private message mentioned in step five of the treatment, the image referred as Pic1 in step six can be observed. Figure 2 displays the demonstration of the treatment step ten in detail.

Figure 2

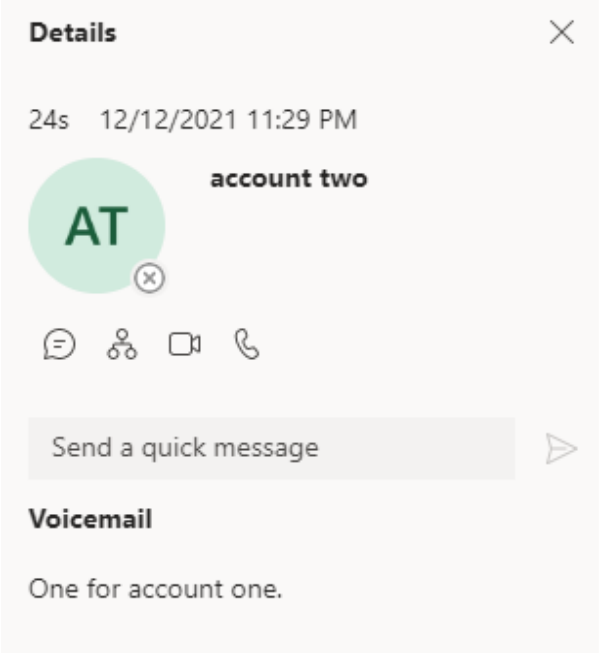
Meeting1 Schedule Details



Finally, Figure 3 displays the demonstration of the treatment step thirty-five in detail, including meeting name, the added accounts, meeting timing, location, and meeting description.

Figure 3

Voice Mail in Text Format



Post-Hoc Test

In the scope of Post-Hoc tests, artifacts created as the result of the treatment are extracted from PC1 with the techniques described in the extraction and measurements section. The data obtained from the Post-Hoc tests are also compared to the data obtained from Pretests. Therefore, the artifacts that are created only by the treatment are isolated. The isolated artifacts of the treatment are listed in the Results section.

Extraction and Measurements

The artifacts of the Microsoft IndexedDB storage are extracted from the .logs files. These files are located in the "C:\Users\<username>\AppData\Roaming\Microsoft\Teams\IndexedDB\https_teams.microsoft.com_0.indexeddb.leveldb" folder where LevelDB implementation of IndexedDB keeps its storage files. There are storage files with two key extensions, namely .ldb and

.log, in this location. The rest of the files are subsidiary and aid .ldb and .log extension files in keeping versions and exchanging information. The .log files keep the most recent information. When .log files reach a size limit, the data is passed to .ldb files and a new version .log file appears. Figure 4 displays a sample .log file in HEX-Editor extension of Notepad++ (P. Jones, 2019; *What Is Notepad++*, n.d.).

Figure 4

LevelDB log file displayed in HEX Editor

Address	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f	Dump
000001c0	75	6e	74	32	40	6e	61	2e	65	64	75	22	09	67	69	76	unt2@na.edu".giv
000001d0	65	6e	4e	61	6d	65	22	07	61	63	63	6f	75	6e	74	22	enName".account"
000001e0	07	73	75	72	6e	61	6d	65	22	03	74	77	6f	22	05	65	.surname".two".e
000001f0	6d	61	69	6c	22	0f	61	63	63	6f	75	6e	74	32	40	6e	mail".account2@n
00000200	61	2e	65	64	75	22	08	75	73	65	72	54	79	70	65	22	a.edu".userType"
00000210	06	4d	65	6d	62	65	72	22	0b	64	69	73	70	6c	61	79	.Member".display
00000220	4e	61	6d	65	22	0b	61	63	63	6f	75	6e	74	20	74	77	Name".account tw
00000230	6f	22	04	74	79	70	65	22	06	70	65	72	73	6f	6e	22	o".type".person"
00000240	03	6d	72	69	22	2c	38	3a	6f	72	67	69	64	3a	37	33	.mri",8:orgid:73
00000250	35	31	31	64	38	34	2d	62	37	32	31	2d	34	33	64	66	511d84-b721-43df
00000260	2d	61	32	65	35	2d	61	32	35	38	62	61	64	36	65	37	-a2e5-a258bad6e7
00000270	36	61	22	08	6f	62	6a	65	63	74	49	64	22	24	37	33	6a".objectId"$73
00000280	35	31	31	64	38	34	2d	62	37	32	31	2d	34	33	64	66	511d84-b721-43df
00000290	2d	61	32	65	35	2d	61	32	35	38	62	61	64	36	65	37	-a2e5-a258bad6e7
000002a0	36	61	22	1b	24	24	61	74	4d	65	6e	74	69	6f	6e	73	6a".$$atMentions
000002b0	5f	4c	61	73	74	41	63	63	65	73	73	65	64	44	54	4e	_LastAccessedDTN
000002c0	20	c3	80	e2	80	93	52	72	c3	85	77	42	22	1b	24	24	Ã€â€"RrÃ…wB".$$
000002d0	61	75	64	69	6f	56	69	64	65	6f	5f	4c	61	73	74	41	audioVideo_LastA
000002e0	63	63	65	73	73	65	64	44	54	4e	20	c3	80	e2	80	93	ccessedDTN Ã€â€"
000002f0	52	72	c3	85	77	42	22	14	24	24	70	32	70	5f	4c	61	RrÃ…wB".$$p2p_La
00000300	73	74	41	63	63	65	73	73	65	64	44	54	4e	20	c3	80	stAccessedDTN Ã€
00000310	e2	80	93	52	72	c3	85	77	42	22	0e	24	24	72	65	71	â€"RrÃ…wB".$$req
00000320	75	65	73	74	43	6f	75	6e	74	49	02	22	15	24	24	66	uestCountI.".$$f
00000330	69	72	73	74	6e	61	6d	65	5f	6c	6f	77	65	72	63	61	irstname_lowerca
00000340	73	65	22	07	61	63	63	6f	75	6e	74	22	14	24	24	6c	se".account".$$l
00000350	61	73	74	6e	61	6d	65	5f	6c	6f	77	65	72	63	61	73	astname_lowercas
00000360	65	22	03	74	77	6f	22	14	24	24	66	75	6c	6c	6e	61	e".two".$$fullna
00000370	6d	65	5f	6c	6f	77	65	72	63	61	73	65	22	0b	61	63	me_lowercase".ac
00000380	63	6f	75	6e	74	20	74	77	6f	22	14	67	75	65	73	74	count two".guest
00000390	6c	65	73	73	44	69	73	70	6c	61	79	4e	61	6d	65	22	lessDisplayName"
000003a0	0b	61	63	63	6f	75	6e	74	20	74	77	6f	22	0b	64	65	.account two".de
000003b0	73	63	72	69	70	74	69	6f	6e	22	08	41	43	43	4f	55	scription".ACCOU
000003c0	4e	54	32	22	09	24	24	73	75	62	54	79	70	65	22	06	NT2".$$subType".
000003d0	41	44	55	73	65	72	22										ADUser"

Google Chrome's implementation of LevelDB deletes the .log files after the browser is terminated (*Hang on! That's Not SQLite! Chrome, Electron and LevelDB*,

2020). However, it was observed during the preliminary work of this experiment that Microsoft Teams .log files are not deleted after the termination of the application. They survive a reboot of the host machine and stay intact for multiple days. After new versions of the .log files are created, only a small number of records are passed to .ldb files. Therefore, in this experiment, the artifacts are mostly obtained from the text based .log files where clear text and binary information have meshed.

The applicable data is collected from the .log files by separating the text from binary data and parsing the meaningful fields in a structured order with the utilization of the software that is developed in PHP (*PHP: Hypertext Preprocessor*, n.d.). Meaningful fields are determined by signature characters that surround the text of records. For instance, it can be observed that the private chat message records are always encapsulated with the signature UTF-8 characters "ÿÿ" in the beginning and "" at the end. These signatures are "0x16 0x12 0x14" and "0xc3 0xbf 0x13 0xc3 0xbf" in HEX values. Initially, full conversion to HEX values was intended. This approach is later changed to obtaining patterns from text signatures that are cleared from blob characters. The full list of signatures that are used during the experiments can be found in Appendix. It is noteworthy to mention that the application is clearing blob data before it starts looking for patterns. This process includes replacing special characters with the dot character. The dot character is chosen because of its light appearance in the text. Consequently, the signatures formed by the application are expected to include dot characters that do not exist in the raw file. Even though these signatures are not native to the files, they are intended to give a basis for where to look for patterns. After clearing the files from the out-of-range characters of the blob, the text in between the signatures is extracted.

Subsequently, the data is processed back to a readable format to extract the required fields. The extraction is processed in the following order:

- Detection of the usernames, identifiers, and their display names
- Detection of the suspect user among extracted users
- Gathering information that belongs to previously detected users. For instance, their SMTP (Simple Mail Transfer Protocol) and email addresses.
- Collecting scheduled meeting data
- Collecting tab extension data (the tab extension data is an added Whiteboard in this experiment)
- Collecting message chat data
- Collecting voice mail data
- Collecting and associating date and time information to previously extracted data

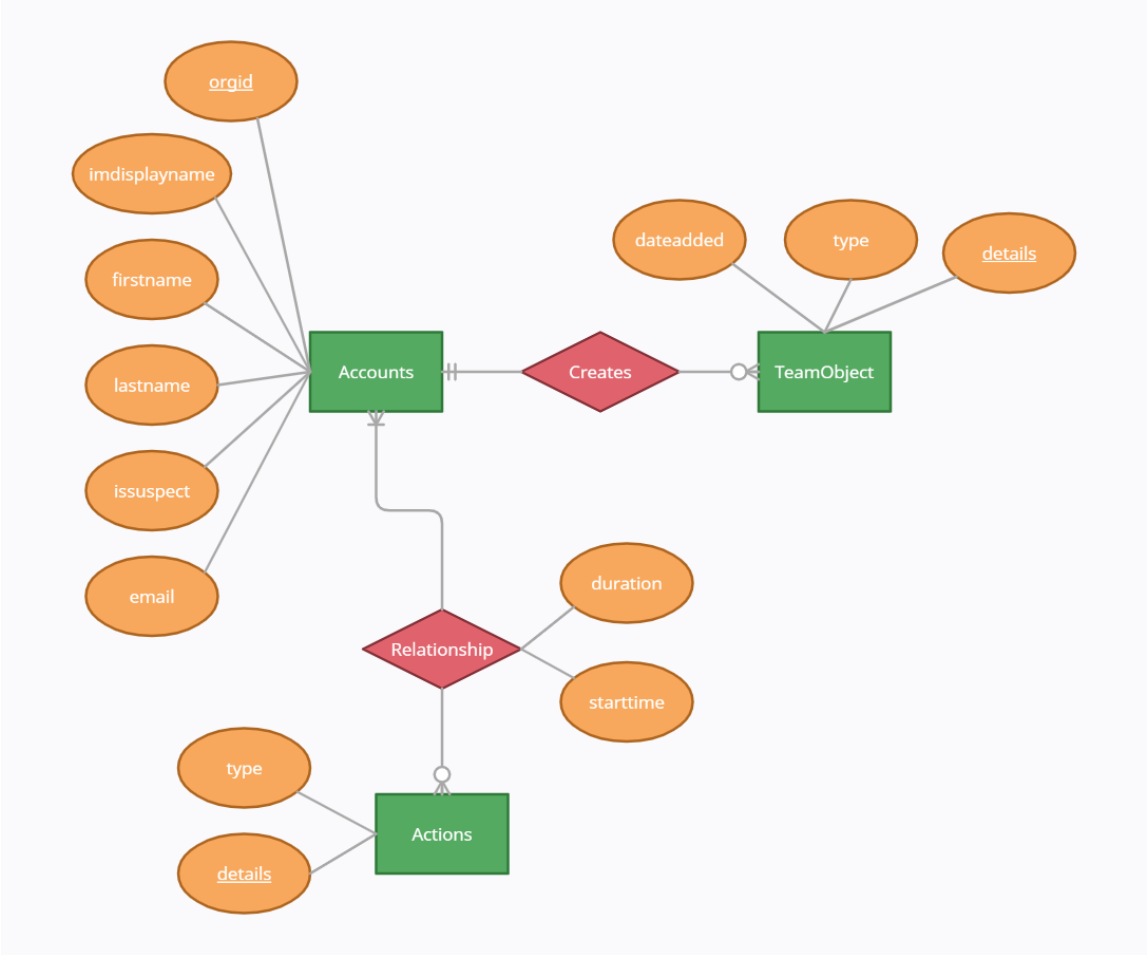
Construction of Time-Frame Analysis

Time frame analysis is performed by processing, isolating, and sorting collected information based on the time indicated in their record fields. Composition times, event starting times, and time stamps that are obtained from their corresponding records are added to a MySQL database and sorted with SQL queries based on their time related fields. It is noteworthy to mention that the database is only intended to support the experiments with a demonstration of time-frame analysis and not with the intention of establishing a forensic tool. Therefore, it does not support multiple cases. After the implementation of the database, it is populated with extracted artifacts. The information from the extracted records is queried from the database in a time ordered format. The

design of the MySQL database, where the extracted strings are stored for access in their desired format and order, can be seen in Figure 5.

Figure 5

Database Design for Time-Frame Analysis



The extracted artifacts include multiple forms of time information. For instance, composition times and client arrival times are determined distinctly. For simplification, only selected time fields of the artifact records are utilized as the time and duration fields in the MySQL database. For event call actions, the time field is based on the “startTime” and duration is based on the Epoch time difference of the “endTime” and “startTime”.

Construction of Time-Frame Analysis

The data that is extracted during the experiments is compared manually to the data available in the .log file in the BLOB format. The first examination is performed on the missing data. The .log file is inspected for any information that is not present in the results extracted by the PHP code. Additionally, every record listed by the code is confirmed to be existent in the .log file.

Methods and Experiments Applied for Instagram IndexedDB Investigations

Experiments in this section are focused on SQLite implementation of Mozilla Firefox and artifacts of the Instagram website. Accordingly, the techniques were constructed around two major hypotheses.

Hypothesis 1 (H1). Data generated by Instagram in IndexedDB storage contain a significant value for digital forensic investigations.

Hypothesis 2 (H2). Instagram data artifacts in IndexedDB storage are suitable for utilization in time frame analysis in digital forensic investigations.

Experimental Design

A Single Case Pretest–Posttest Quasi Experiment is carried out to test the hypotheses. The methodology adopted in the section resembles the experimental design put forward by Cook and Campbell (1979). A single subject is measured before and after being put through an adapted treatment. A comparison of the measurements is provided to disclose the change obtained by the treatment. Artifacts inherently present in the IndexedDB storage locations are discovered with the pretest experiment. The artifacts generated by the application of the treatment are identified with the comparison of the results obtained after the pretest and the treatment. In simpler terms, the difference

between the measurements is the proof that the discovered artifacts are the outcome of the treatment.

The subject provided in this experiment is the Instagram web application. The experimental environment is an HP Laptop that operates a Windows 10 Home Operating System with Mozilla Firefox 89.0.2 (64-bit) and Google Chrome v91.0.4472.124 (64-bit) browsers installed. Additionally, a Samsung s20 FE android device is utilized to create the Instagram accounts as part of the environmental setup procedure. The experiment is applied twice, once on Mozilla Firefox and once on Google Chrome for the verification of the data obtained from Mozilla Firefox SQLite storage. In order to set the experimental environment prior to the quasi-experiment, the following steps are carried out.

- Three different Instagram Accounts are created with Samsung Android Device (Phone1). Table 11 displays the personal information details of the accounts which are used to determine their availability in the IndexedDB storage.

Table 11

User Accounts Utilized in Instagram Experiments

Personal Information	**Account1**	**Account2**	**Account3**
Name	Forensic Researcher 1	Forensics Researcher 2	Forensic Researcher 3
Username	forensicreasearchaccount1	forensicresearchaccount2	forensicresearchaccount3
Website	x	http://forensicresearcher2.com (Dummy info)	http://forensicresearcher3.com (Dummy info)
Bio	x	Bio of Forensic Researcher 2	Bio of Forensic Researcher 3
Email	forensicresearchaccount1@protonmail.com	forensicresearchaccount2@protonmail.com	forensicresearchaccount3@protonmail.com
Email Confirmation	Not Confirmed	Confirmed	Confirmed

(continued)

Personal Information	**Account1**	**Account2**	**Account3**
Phone Number	x	+33 4 64 03 67 89 (Randomly generated)	x
Phone Number Confirmation	x	Not Confirmed	x
Gender	x	Male	Female

- Account1 and Account2 are added as followed connections through the android application.
- Account2 and Account3 are added as followed connections through the android application. (No connection is created between Account1 and Account3)
- In Account1 and Account2, a public account (Account4) is added as a followed connection to increase the scope and diversity of the available data. The idea is that the experimental accounts can overlook some data that exists in an operational account. Account4: awesome.photographers (*Awesome Photographers: Instagram Photos and Videos*, n.d.).
- Windows 10 computer (PC1) is formatted and installed with Mozilla Firefox (Browser1) and Google Chrome (Browser2) browsers.

Pretest

The artifacts inherently present in Mozilla Firefox and Google Chrome browsers are tested with the following procedures.

- Instagram Web Application is accessed through Browser1 and Browser2 without logging into the accounts.
- The connection is left idle for a time of fifteen minutes.

- The artifacts are collected from IndexedDB storage locations of Browser1 and Browser2 in PC1

Treatment

A set of activities were designed according to the observation of common user behavior on Instagram. These activities constitute the treatment of the research. The observations were made over the stored data of activity from five Instagram accounts of volunteers. Three of the volunteers were students in the computer science department whereas two were faculty again in the computer science department of a university in Stafford, Texas. The listed activities constitute over ninety-eight percent of all activities of the volunteers in the subsequent nine days of the collection.

- Instant private messaging
- Sending messages with video and pictures contents
- Displaying messages with videos and pictures received from other users
- Adding stories
- Displaying stories
- Visiting profiles
- Displaying recently added posts of followed connections on the home page
- Commenting on posts of followed connections
- Liking posts of followed connections
- Discovering new accounts through the explore page
- Searching an account with its name
- Adding a post with graphic content to a personal account

Proceeding from the observation-based user activities, the treatment procedure of the experiment is established with the following steps:

- Account2 is logged in from Phone1
- A random picture of a carpet is added to Account2 with Phone1. Including the description “Forensic Researcher 2 – Post 1”
- A random picture of a ceiling is added to Account2 as a story.
- A message with the content: “Message1 -Account2 to Account1” is sent from Account2 to Account1 from the messages page through Phone1
- Account2 is logged out from Phone1
- Account3 is logged in from Phone1
- Post1 of Account2 is liked with Account3 on Phone1
- Post1 of Account2 is commented on with the following content: “Account3 comment for Account2 - Post1” by Account3 on Phone1
- Account3 is logged out of Phone1
- Account1 is logged in from Browser1 in PC1
- Account1 home page is displayed while scrolling down to display the posts and comments made by Account2 and Account3 on Browser1
- Story1 of Account2 is displayed on the home page
- An emoji reaction is added to Story1 of Account2
- The messages page is accessed and the message from Account2 is displayed with Account1 on Browser1
- A message with the content “Message2 – Account1 to Account2 with emoji content: ☹⛀㊙” is sent as a reply to Message1

- Account1 home page is accessed and a recent story from Account4 is displayed through Account1 on Browser1
- The explore page is accessed through Account1 on Browser1
- The words “Forensics Researcher 2” are entered on the search-box of Account1 on Browser1
- The profile page of Account2 is accessed through Account1 on Browser1
- The procedure followed with Account1 on Browser1 is repeated on Browser2
- The procedure, where Account1, Account2, and Account3 are set as public accounts, is repeated with Account3 as a private account

Additionally, observations from the preliminary work of the experiment showed that more steps in the treatment are needed to obtain additional artifacts. Firstly, it was determined that there were changes in the stored information from the accounts when their comment on the posts was displayed by hovering the mouse, and when the profile pages were visited. Therefore, two additional extractions are performed. In total, three cases where extraction is performed are as follows:

- When only the comment of Account3 is displayed on the home page but no interactions with the Account3 profile are taken
- When the brief profile information of Account3 is displayed by hovering the mouse over the profile section of the comment.
- When the Account3 profile page is visited

Figure 6 displays the comment of Account3 over the post of Account2 from the view of Account1, where no action was taken on the comment and the profile page of Account3.

Figure 6

Comment is Displayed without Profile Interactions

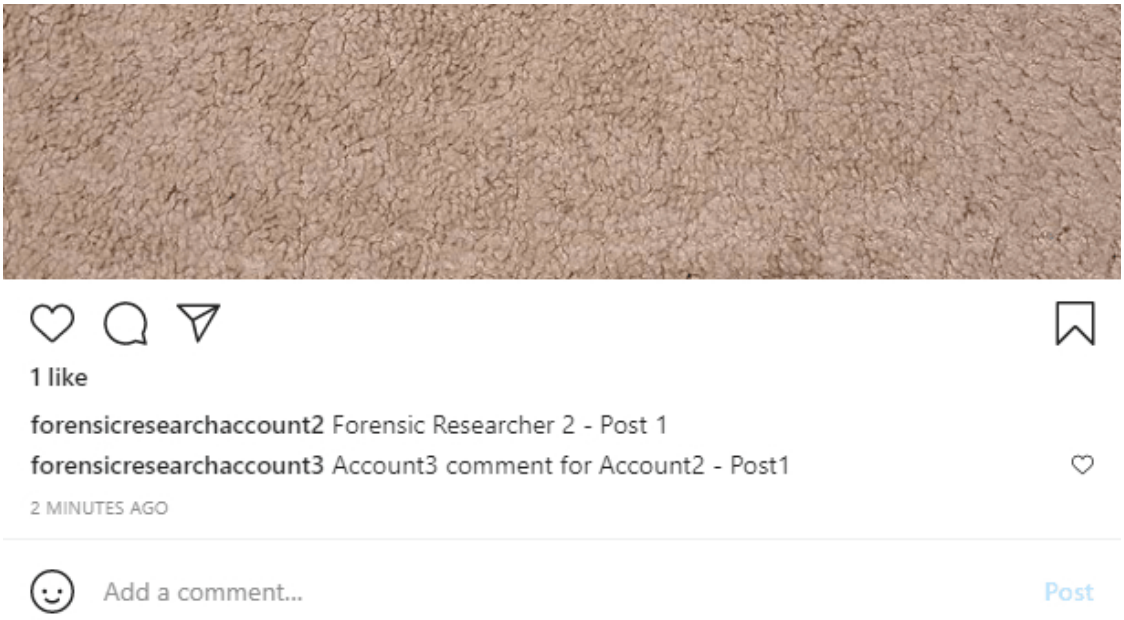


Figure 7 displays the same comment with the display of the Account3's profile.

Figure 7

Comment is Displayed with Profile Displayed Through Mouse Hovering

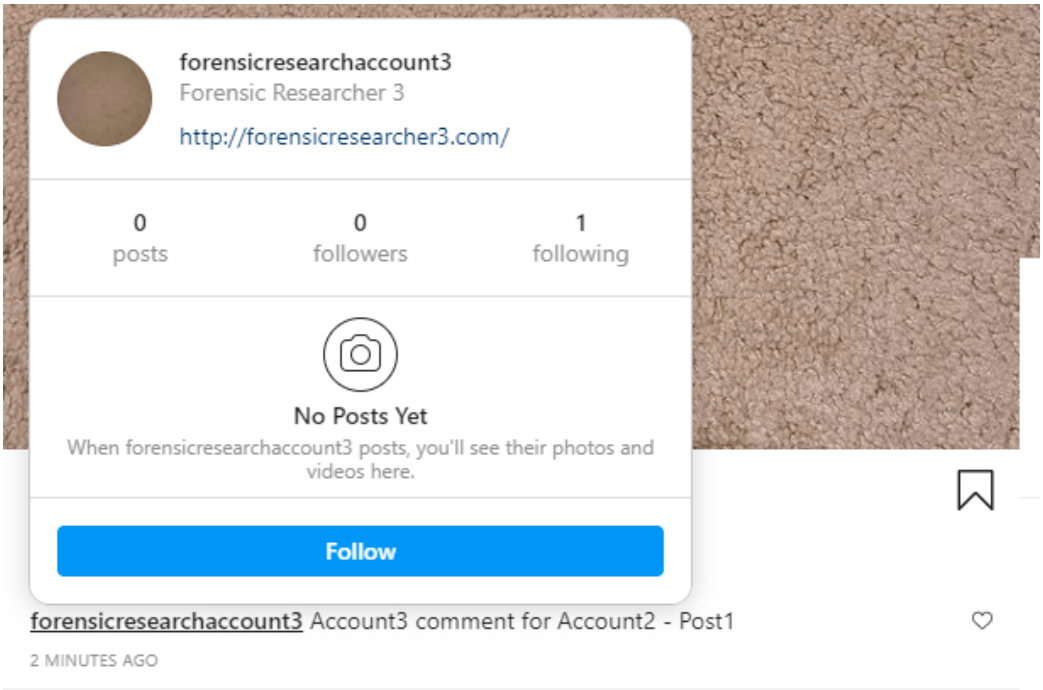
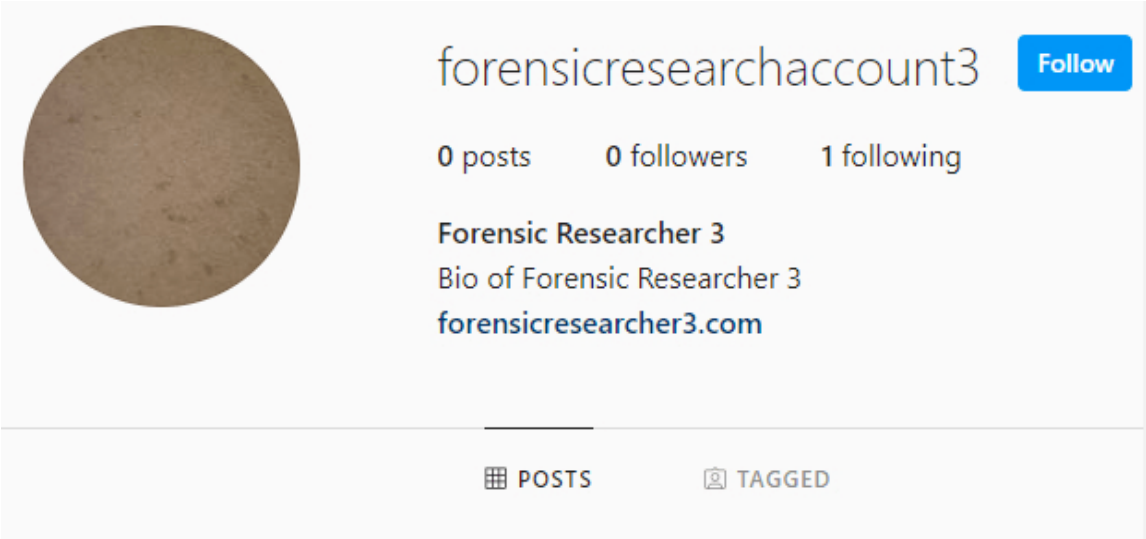


Figure 8 displays the profile of Account3, when it was visited by Account1 from its comment.

Figure 8

Commenting Profile is Visited



Secondly, the first observations on the record of the relationships showed an attribute called stable. In order to discover the actions that would result in changes to this attribute, the following steps were added to the treatment with an extra step of extraction to test the changes in the values at each step. These additional steps can be listed as follows:

- Account3 is blocked and Post1 of Account2 is displayed on the home page of Account1
- Account3 is unblocked and Post1 of Account2 is displayed on the home page of Account1
- Account3 is restricted and Post1 of Account2 is displayed on the home page of Account1

- Account3 is unrestricted and Post1 of Account2 is displayed on the home page of Account1

Post-Hoc Test

In accordance with the procedures of Pretest–Posttest Quasi Experiments, an independent test is conducted without the treatment following the experiment's environmental set-up. Consequently, the artifacts created in PC1 for IndexedDB are tested after the treatment is applied. The comparison of the results from before and after the treatment allows the authors to isolate artifacts that can be tied exclusively to the treatment. Therefore, the results section presents the observation of the artifacts created by the treatment and only by the treatment in the IndexedDB storage of PC1.

Methods and Experiments Applied for WhatsApp Web IndexedDB Investigations

Experiments in this section are focused on LevelDB implementation of Google Chrome and artifacts of the WhatsApp Web application. Accordingly, the techniques were constructed for the given two hypotheses.

Hypothesis 1 (H1). IndexedDB storage carries forensically significant artifacts for WhatsApp Web Application.

Hypothesis 2 (H2). WhatsApp Web Application artifacts in IndexedDB can be used to create time frame analysis in forensic investigations.

Experimental Design

In this section, a Single Case Pretest-Posttest Quasi Experiment is performed to populate IndexedDB storage (Cook & Campbell, 1979). The subject of the experiment is the WhatsApp Web Application. The experimental environment is a Sony Vaio Laptop that operates a Windows 10 Single Language Operating System with Google Chrome

v83.0.4103.97 (Official Build) (64-bit) browser installed. It is complemented by two Samsung S8+ phones that are installed with OS 10 - Q running the WhatsApp Messenger v2.20.172.

The pretest experiment indicates the artifacts are inherently present at the storage location. The comparison of the results from the pretest and treatment shows what artifacts are created by the treatment. In other words, it is proof that the artifacts discovered are the result of the treatment applied.

The following steps are performed to set the experimental environment before the quasi-experiment is performed.

- PC1 - Windows 10 Computer is formatted and installed with the Google Chrome browser.
- Phone1 and Phone2 - Already functional 10-Q Android Phones are installed with WhatsApp Messenger.
- Phone1 and Phone2 are added as contacts to each other in the built-in phonebook.

Pretest

The artifacts that are inherently present in the IndexedDB storage are tested with the following steps:

- WhatsApp Messenger is initiated in both Phone1 and Phone2.
- web.whatsapp.com is reached through PC1
- Connection barcode at the PC browser is shown to the screening of the Phone1 Messenger
- The connection between Phone1 and Phone2 is left idle
- The artifacts are collected from the Chrome IndexedDB storage location in PC1

Treatment

The treatment of this research includes the actions taken with WhatsApp Web Application to create artifacts on the IndexedDB storage. The activities were created according to the observation of common user behavior with web browsers and messenger communication applications. When stored information of activity was inspected on three volunteers' WhatsApp Messenger and Web Application, the following activities were observed to constitute over ninety-five percent of all activities in two consecutive weeks.

- Text Messaging
- Sending media messages including video and pictures; pictures including jpeg files and gif files, and displaying transferred files
- Voice calls
- Video calls
- Blocking and unblocking contacts
- Displaying contact user photo

Additionally, in the preliminary and exploratory investigations performed to discover the potential of the research, some records of user presence were observed. Therefore, walking away from the computer with the phone was added to the given activity list. Based on the observed behavior, the following steps were constructed as the treatment:

- Phone1 WhatsApp Messenger is connected to PC1 WhatsApp Web Application via QR code
- The message "This is message 1" is sent from PC1 (Phone1 connected) to Phone2
- The message "This is reply 1" is sent from Phone2 to PC1

- The link of the sample video (*Wildlife Windows 7 Sample Video - YouTube*, 2012) is sent from PC1 to Phone2
- The link of the sample video “Wildlife Windows 7 Sample Video” is sent from Phone2 to PC1
- The video “Wildlife Windows 7 Sample Video” is played in Phone1
- The video “Wildlife Windows 7 Sample Video” is played in PC1
- The video “Wildlife Windows 7 Sample Video” is played in Phone2
- A video call request is sent from Phone2 to Phone1. The call is not answered
- A video call request is sent from Phone2 to Phone1. The call is answered and kept active for over 5 seconds
- A video call request is sent from Phone1 to Phone2. The call is answered and kept active for over 10 seconds
- Phone1 is carried around twenty meters (estimated roughly with twenty steps) away from PC1
- Phone1 is carried back next to PC1
- Phone1 is disconnected from PC1 and reconnected after 5 seconds
- Phone2 account is blocked and unblocked from Phone1

Post-hoc Test

As the procedure for the quasi-experiments, an independent test is conducted without the manipulations performed at the treatment. Employment of all procedures takes place after the set-up of the experimental environment. The observation of the artifacts created in the IndexedDB storage files of PC1 is presented in the Results section.

Methods and Experiments Applied for Evaluation of IndexedDB as a Verification Source

Techniques suggested in this section, along with the experiments they were built on, were designed for the evaluation of IndexedDB as a verification source for traditional web browser artifacts. Three hypotheses were constructed in the scope of this section.

Hypothesis 1 (H1). Persistent storage technologies can be utilized for the verification of artifacts obtained in browser forensic investigations.

Hypothesis 2 (H2). A score of artifact inconsistency can be constructed utilizing persistent storage artifacts.

Hypothesis 3 (H3). The level of accuracy of the verification derived from persistent storage technologies can be determined and measured.

Experimental Design

Single Case Pretest-Posttest Quasi Experiment of Cook and Campbell (Cook & Campbell, 1979) utilized in the previous sections were chosen in this section for the operations intending to populate IndexedDB artifacts. Distinctively, this time the artifacts are not only intended to be extracted from IndexedDB storage but from various web browser storage technologies. Subsequent to the design of the experiment, the measurements for the level of inconsistency and level of reliability were methodized.

The subjects in this experiment are the top twenty websites in the US listed by Alexa (*Top Sites in United States*, n.d.). The experimental environment is a Sony VAIO Laptop that operates a Windows 10 Single Language Operating System installed with Google Chrome v102.0.5005.63, Mozilla Firefox v100.0.2, and Microsoft Edge

v101.0.1210.53 web browsers. These three browsers constitute over eighty-four percent of the overall browser usage (*Browser Market Share*, n.d.).

The measurement process was constructed with a series of activities including acquisition by FTK Imager (*FTK Imager Version 4.5*, 2020), extraction of the browser history, typed URLs, and persistent storage artifacts by digital forensic tool Autopsy (*Autopsy - Digital Forensics*, n.d.), and calculating verification reliability and inconsistency scores. The pretest experiment is intended to determine the artifacts that are inherently present for the web browsers and verify that no inherent condition affects the scores. Whereas the post-test experiment aims at isolating the artifacts and their final scores which are produced as a result of the treatment. The experimental environment is set with the following steps

- The hard disk of the Windows 10 computer (PC1) is partitioned into two spaces to ease the imaging and extraction process. The first partition (Partition1) is given seventy gigabytes of space and the second partition (Partition2) is given four-hundred-six gigabytes which is the rest of the hard disk space.
- Partition1 is formatted with Windows 10 and installed with Google Chrome, Mozilla Firefox, and Microsoft Edge browsers.
- An external hard disk (Disk1) is wiped seven times as suggested by NIJ recommendations (National Institute of Justice, 2001) and reserved for storage of the disk images.

Pretest

The artifacts that are inherently present in the IndexedDB storage of the subject websites are determined with the following steps:

- An image of the Partition1 is acquired with FTK Imager and saved to Disk1 with the live acquisition
- Artifacts of browsing history, typed URLs, and IndexedDB storage files are extracted by Autopsy to an examination computer (PC2).
- A calculation of verification and integrity scores is performed

Treatment

The treatment of this research includes the user activities in the top twenty websites of the US listed by Alexa (*Top Sites in United States*, n.d.) to create artifacts on three web browsers. At this level of the experiments, the top twenty websites of Alexa were preferred to the top fifteen since verification of the artifacts requires more data for validity than exploratory experiments. The actions were selected based on the observations of user behavior obtained in the experiments in the previous sections. Table 12 lists steps that were constructed as treatment which are performed three times: once in Google Chrome, once in Mozilla Firefox, and once in Microsoft Edge web browsers.

Table 12

IndexedDB Verification Experiment Treatment Steps

Alexa Ranking	Source	Treatment
1	Google.com	1. An account is created with First Name “Forensic” Last Name “Researcher” and username forensicresearch1@gmail.com 2. The options “Save my web & app activity in my Google account” and “Show me personalized ads”. Additionally,” show my YouTube history in my Google account” are selected to increase the number of potential artifacts. 3. The search term “Search term1” is entered in the search box 4. Did you mean “search term 1” link under the search box is clicked. 5. In the page that displays the results, “Search term2” is entered in the top search box

(continued)

Alexa Ranking	Source	Treatment
2	Youtube.com	6. Youtube.com is accessed with the google account created for the source Google.com 7. The term “search term1” is entered in the search box 8. The video W(*Keywords vs Search Terms - What Is the Difference?*, 2019) is watched for 2 minutes. 9. The video (*What Are Search Terms with Examples*, 2019) is accessed through the suggestion panel and watched for two minutes
4	Yahoo.com	10. An account is created with First Name “Forensic” Last Name “Researcher” and username forensicresearch1@yahoo.com 11. The search term “Search term1” is entered in the search box 12. The first listed result (Chris, 2021) is clicked on the search results 13. Tab will be switched back to the results list and “Search term2” will be entered in the top search box
5	Facebook.com	14. Two accounts are created with credentials: Account1: First Name “Forensic” Last Name “Researcher” Account2: First Name “Forensic” Last Name “ResearcherTwo” 15. Account2 is added as friend by Account1 16. The friendship request is accepted by Account2 17. “Post1 of Account2” is added to the wall of Account2 18. A picture is added as account picture by Account2 19. Account picture and wall post of Account2 is displayed by Account1
8	Reddit.com	20. reddit.com is accessed and an account (forensicresearch1) is created based on the google account created for the source Google.com. 21. The search term “funny videos” is entered in the search box 22. The video (*Parents Make a Funny Video : MadeMeSmile*, 2022) is accessed. 23. The picture (*Funny Picture Lol : ChargeYourPhone*, 2022) is displayed.
9	Bing.com	24. Bing.com is logged in with fxp017@shsu.edu Microsoft 365 account linked to Sam Houston State University. 25. The search term “Search term1” is entered in the search box 26. The first listed result (Chris, 2021) is clicked on the search results 27. Tab is switched back to the results list and “Search term2” is entered in the top search box
10	Office.com	28. Office.com is logged in with fxp017@shsu.edu Microsoft 365 account linked to Sam Houston State University. 29. The install office selection box at the right top corner is clicked and “Office 365 apps” is chosen.

(continued)

Alexa Ranking	Source	Treatment
11	Wikipedia.com	30. An account is created with username forensicresearch1@yahoo.com and linked to the email provided in the source Google.com 31. The search term “Forensic science” is entered in the search box. 32. “American Academy of Forensic Sciences” is clicked from the “See also” section
12	Myshopify.com	33. A free trial is started for “forensicresearch1@gmail.com” 34. A store is created with the name “forensicstore1” 35. A product draft with the name “Search Product1” and description “Description1” is created on products page 36. The phrase “Search term1” is entered in the search box and search is initiated for “Apps” category
13	Ebay.com	37. The phrase “Search term1” is entered in the search box for all categories 38. The first listed result is clicked on the search results 39. Tab is switched back to the results list and “Search term2 is entered in the top search box for all categories
14	Chase.com	40. The continue button is clicked at the section with title “For new Chase checking customers” 41. A new account* is created and logged in
15	Microsoft.com	42. Microsoft.com is logged in with fxp017@shsu.edu Microsoft 365 account linked to Sam Houston State University. 43. “Search term1” is entered at the top search box 44. The first listed result (Chris, 2021) is clicked on the search results 45. Tab is switched back to the results list and “Search term2” is entered in the top search box
17	Netflix.com	46. An account is created with the email “forensicresearch1@gmail.com” and name “Forensic Researcher” 47. Subscription is created at the “Basic” option* 48. The first listed item with the title “Red Notice” is clicked and watched for two minutes. 49. Tab is navigated back to main screen
19	Instagram.com	50. An Instagram account “forensicreasearchaccount2” that is followed by the account “forensicresearchaccount1” is logged in 51. A picture is added, and a story is shared by “forensicresearchaccount2”. 52. “forensicresearchaccount1” is logged in and shared material of “forensicresearchaccount2” is displayed on the home page.

(*) The details of financial credentials and account information was not shared for privacy and security regulations.
(**) The sources that did not contain noteworthy artifacts in persistent storage technologies were not included in this table

Only fourteen websites out of twenty listed by Alexa are given in Table 12. The list of fourteen was constructed with a preliminary work that checked if the persistent storage was utilized by the sites that were listed in the top twenty list of Alexa. The sites that contained any identifying information such as time, location, and IP address in their persistent storage were added to the treatment. For five websites, there were no significant data in IndexedDB storage. Additionally, the content of one website was pornographic in nature. Therefore, it was not added to the list.

Post-hoc Test

As the procedure for the quasi-experiments, a test with identical steps listed in the pretest section is conducted after the operations and manipulations are performed according to the treatment of the experiment. The observation of the artifacts populated in the experimental environment and their verification reliability score (VRS) and artifact inconsistency score (AIS) scores are presented in the Results section.

Measurement of VRS and IAS

In the underlying principle of the verification through persistent storage, there is an assumption of a correlation between the different sources pointing to the same attributes of action. Therefore, the persistent storage artifacts are queried for the verification of the information obtained from web browsing artifacts. IAS determines the level of inconsistency and VRS indicates the reliability of the IAS score based on the completeness of the data that is utilized in the analysis.

VRS is determined by the variety of the data through which the verification will be achieved. Therefore, it is a multi-layer score that incorporates the number of sources, actions, and attributes. For instance, one website is a source that can verify several

actions such as visits with the date and time information it stores on the activities of the users. Additionally, a limited number of sources keep information on the type and the version of the web browser, IP address, and even the account usernames of the user. Detailed information on discovered artifacts is given in detail in the results section. For the calculation of VRS, first, a priority factor for each verification item is constructed. The priority factors can be manually assigned by the investigators based on the examination they are performing. In this experiment, the number of existing resources was determined as the priority factor determination criteria. The priority factor is calculated according to Equation 1. Second, the reliability factor is calculated for each item based on the variety of sources it can verify. If a source can verify all possible variety of artifacts (APVA), its reliability factor is determined as one. The APVA is further discussed in the results section where all the matching artifacts of persistent storage are listed. The reliability factor is calculated with Equation 2.

$$PF_{\mathrm{j}} = \sum(\mathrm{V_i}) \,/\, \sum\left(V_{i\mathrm{j}}\right) \qquad (1)$$

$$\mathrm{R}F_{\mathrm{i}} = \sum(\mathrm{AV_k}) \,/\, \mathrm{APVA} \qquad (2)$$

For Equation 1 and Equation 2; i is the verification item numerator, j is the source numerator, PF_j is the measured priority factor, V_i is the verification items for a single source whereas V_{ij} is the verification items for all sources. In Equation 2, $\mathrm{R}F_{\mathrm{i}}$ represents the reliability factor of a source. Correspondingly, k is the variety numerator and $\mathrm{AV_k}$ yields one when the given category of artifacts is present in the source.

VRS is determined by the incorporation of priority and reliability factors. It should be noted that the sum of priority factors of all the sources adds to one. The

reliability factor of each source takes a value from zero to one. Therefore, the reliability factor is divided by the total number of sources obtained from Alexa which is fourteen. Within this scope, in a complete verification where all the artifacts are matched, the VRS value takes the value one. This value is converted to its percentage representation for a better sense of comparison. Equation 3 shows the calculation of VRS.

$$\mathrm{VRS} = \sum \left(\mathrm{R}F_{\mathrm{j}} * PF_{\mathrm{j}} \right) * \mathbf{100} \qquad (3)$$

The IAS is calculated with the number of mismatches between the existing persistent storage artifacts and the non-persistent storage artifacts. It takes the total number of verification items as the basis and divides the existing matches of verification items by this total. However, the contribution of each item is also taken into consideration by weighting the total with the priority factor obtained from the VRS score. IAS is calculated according to Equation 4.

$$\mathrm{IAS} = \sum \left(\mathrm{MV}_{\mathrm{ij}} * PF_{\mathrm{ij}} \right) / \sum \left(V_{ij} \right) \qquad (4)$$

Cross Consistency and Communality Analysis Among IndexedDB Storage Content Through Term and Inverse Document Frequency

As the VRS and IAS scores determine the level of consistency between the IndexedDB storage and traditional browser storage technologies, the commonalities between the IndexedDB storage artifacts will be examined through term frequency and inverse document frequency analysis (tf-idf) (Jones, 1972). tf-idf focuses on the identification of meaningful keywords occurring in the documents by quantity cross examination. In this context the following terms can be defined:

- t: keyword
- d: document
- tf(t, d): term frequency. Defined as the number of times t is encountered in d
- df(t): document frequency. Defined as the number of documents that contain t
- C: Corpus
- N: Total number of documents
- idf(t, d): inverse document frequency

The measurement of tf-idf in this dissertation resembles the calculations utilized in Harvlant and Kreinovich (2017). Therefore, the commonality of artifacts in IndexedDB storage of the websites listed in Table 12 is evaluated for tf(t, d) as the raw count of t in d. The idf(t, d) are evaluated according to Equation 5. Idf value without the use of ln results in high values. Therefore, using ln limits these numbers. ln is log for base e instead of ten, which gives a better range of results that does not dominate the overall analysis.

$$\text{idf}(\text{t}, \text{d}) = \ln [N \,/\, tf(\text{t}, \text{d})] \qquad (5)$$

CHAPTER IV

Results

This chapter presents the results obtained from the experiments defined in Chapter III. Initially, storage locations of IndexedDB artifacts for different web browsers are laid out. Based on the exploratory observations of the artifacts in these locations, the popularity of IndexedDB among major websites is designated. Subsequently, categorization of the IndexedDB artifacts of Microsoft Teams, Instagram, and WhatsApp Web is provided. Finally, the evaluation of IndexedDB along with its predecessor LocalStorage as a verification source of traditional browsing artifacts was presented.

Usage of IndexedDB in Major Websites of US

For each operating system, the location that is used to keep the IndexedDB data is identified and summarized in Table 13 for every web browser.

Table 13

Locations of IndexedDB Client-Side Storage

Operating System	Web Browser	Location of IndexedDB Storage
Windows 10	Chrome	%USERPROFILE%\AppData\Local\Google\Chrome\User Data\Default\IndexedDB
Windows 10	Explorer	%USERPROFILE%\AppData\Local\Microsoft\Internet Explorer\Indexed DB
Windows 10	Firefox	%USERPROFILE%\AppData\Roaming\Mozilla\Firefox\Profiles\m8n0p71x.default(profile id)\storage\default\
Windows 10	Opera	%USERPROFILE%\AppData\Roaming\Opera Software\Opera Stable\IndexedDB
Windows 10	Safari	IndexedDB Not Supported
MacOS 10.13.6	Chrome	%USERPROFILE%/Library/Application Support/Google/Chrome/Default/IndexedDB/
MacOS 10.13.6	Explorer	NA

(continued)

Operating System	Web Browser	Location of IndexedDB Storage
MacOS 10.13.6	Firefox	%USERPROFILE%/Library/Application Support/Firefox/Profiles/rjx9p40s.default(profile id)/storage/default/
MacOS 10.13.6	Opera	%USERPROFILE%/Library/Application Support/com.operasoftware.Opera/IndexedDB/
MacOS 10.13.6	Safari	%USERPROFILE%/Library/Safari/Databases/__IndexedDB/
Ubuntu 18.04	Chrome	/home/[USERNAME]/.config/google-chrome/Default/IndexedDB/
Ubuntu 18.04	Explorer	NA
Ubuntu 18.04	Firefox	/home/[USERNAME]/.mozilla/firefox/*.default/storage/persistent/
Ubuntu 18.04	Opera	/home/[USERNAME]/.config/opera/IndexedDB/

It was observed that the implementation details do not change for different operating systems. However, Internet Explorer is not supported on MacOS and Ubuntu operating systems. Also, Safari 5.1.7, which is the latest version of the Safari browsers that can run on Windows operating systems, does not support IndexedDB. Also, Safari is not available on Ubuntu operating systems. As the implementation details do not differ among the considered operating systems, for brevity, discussion on IndexedDB is limited to Windows systems for the rest of the paper. As the same principles apply, the discussion can be applicable to the other operating systems without loss of generality. Additionally, when the storage files exist, different web browsers did not appear to make any difference in the contents of the information kept in the IndexedDB with the exception of Google as it keeps in the Chrome browser particular information such as doodles, images, and scripts using the IndexedDB storage.

Table 14 displays observed utilization levels of IndexedDB storage for the most popular websites.

Table 14

Usage of IndexedDB for Top 15 US Websites

Rank	Website	IndexedDB Usage
1	google.com	Light
2	youtube.com	Heavy
3	facebook.com	Light
4	amazon.com	Heavy
5	reddit.com	Little-to-non
6	yahoo.com	Little-to-non
7	wikipedia.org	Little-to-non
8	twitter.com	Light
9	instagram.com	Heavy
10	linkedin.com	Little-to-non
11	ebay.com	Little-to-non
12	netflix.com	Light
13	espn.com	Light
14	twitch.tv	Little-to-non
15	microsoftonline.com	Little-to-non

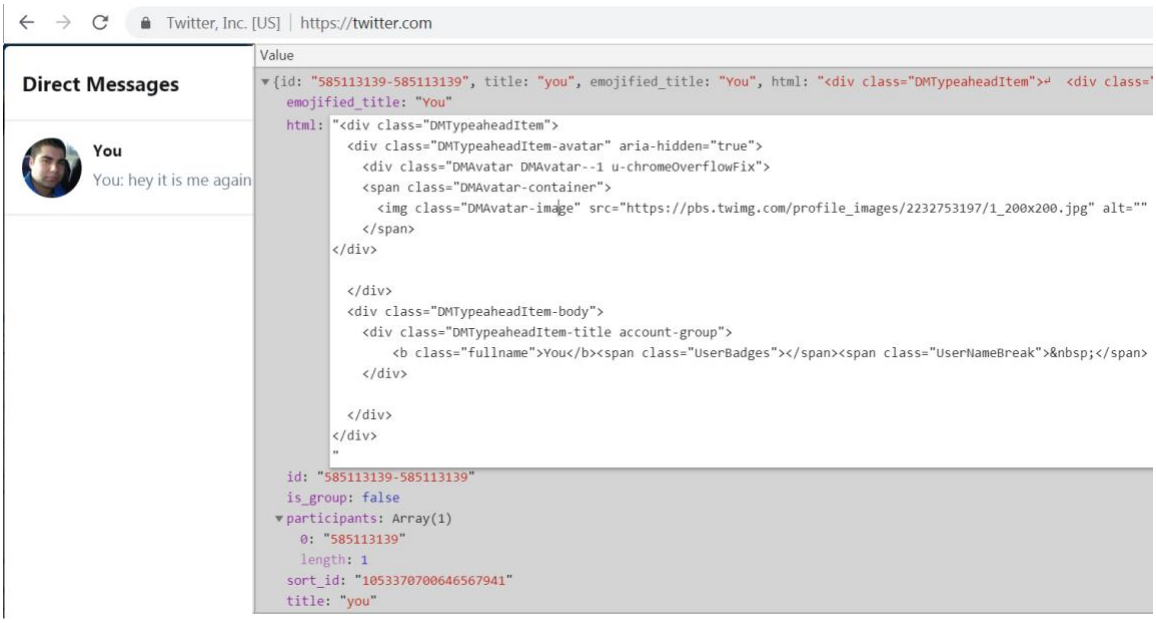
(*) The categorization was created according to the extent of the usage with the labels heavy, light, and little-to-non employment, where heavy means that the website uses at least 10 parameters in the database; light means 2 to 10 parameters are kept in the IndexedDB, and little-to-non refers to the website holding less than 2 parameters in IndexedDB or not using the technology at all.

IndexedDB storage of the top 15 websites was determined to contain information that can be significant in the identification of the suspects and their activities. Such as user screen names, IDs, subjects of conversations, permissions, and the image locations.

Figure 9 shows the information observed in Twitter IndexedDB storage as an illustration of the identifying information that can be obtained with IndexedDB forensic investigations.

Figure 9

Twitter IndexedDB Content on Chrome Developer Tools



Among the top 15 websites, Instagram generated the heaviest level of artifacts in the IndexedDB storage. Including complete contents of the posts and comments. Additionally, the information about the relationship between the suspect and the owner of the posts and comments was recorded. YouTube was another website that displayed significant usage of the technology, which included the number of high priority notifications, id tokens, and authentication keys. The website did not require any login information to create an IndexedDB database for the user. However, more information was added after the login. Facebook and Twitter were determined to keep the user's id in IndexedDB storage after the login. Facebook did not appear to store additional information, while Twitter kept track of the user and its conversations with other users. It

also stored some scripts and HTML codes for the user page. Some websites such as Amazon showed heavy reliance on the IndexedDB technology to keep scripts and codes to be displayed, benefiting from a lowered bandwidth since the information transfer is reduced heavily with IndexedDB implementation.

It was also observed that some websites such as Netflix keep information in IndexedDB when there is a registration in progress or when the user is already registered and logged in.

IndexedDB Artifacts of Microsoft Teams

The artifacts created in the IndexedDB storage as a result of the treatment interactions displayed information of users with the content, time, and configuration setting information. An overview of the extracted artifacts is given in Table 15 with the treatment steps that created them.

Table 15

Artifacts Created in IndexedDB Storage of Microsoft Teams by Treatment

Steps	Activity	Artifacts
1	Account login	Account configuration settings
2	Team creation	Team configuration information, shared space access links, content and time information of the last sent message, role of the suspect in the team
3	Adding accounts to a team	No artifacts encountered
4	Sending chat messages	Conversation identifiers, message content and type, sending account name, time information, message account interactions
5	Sending chat messages with hyperlinks and emojis	Hyperlinks, hyperlink properties, hyperlink content description, emoji description, emoji code in the application
6	Sending chat messages with media content	Media attachment characteristics, access links

(continued)

Steps	Activity	Artifacts
7	Adding media to the shared files of a team	No artifacts encountered
8	Adding extension tabs (white board)	Name, type, access links, and settings of the extension
9	Adding content to extension tabs	No artifacts encountered
10	Creating scheduled meetings for a team	Name, type, schedule time details, location, reoccurrence periods, settings of scheduled meeting
11	Initiating video calls	Conversation identifiers, session type information, timing information, call type and direction, originating and target account identifiers, acceptance or rejection of the call, conversation access links
12	Muting the account during video calls	No artifacts encountered
13	Turning the camera on and off during video calls	No artifacts encountered
14	Giving reactions to send chat messages	No artifacts encountered
15	Terminating video calls	End time of the event call records
16	Starting scheduled meetings	Meeting activity type, participant count and list, participant attendance duration (stored incorrectly)
17	Joining and leaving scheduled meetings	Addition to participant list of scheduled meeting records
18	Recording scheduled meetings	Recording access links, recording status, reason of termination for recording, meeting specific information, recording time information
19	Kicking accounts from scheduled meetings	No artifacts encountered
20	Terminating scheduled meetings	No artifacts encountered
21	Adding accounts as contacts	Contact account names (in various formats), identifiers, electronic addresses, and description
22	Activating voice mail	User account configuration for voice mails being set to true
23	Leaving and displaying voice mails	Message content in text form, time information, message identifiers, source account identifiers, access links

Several useful pieces of the artifacts such as the timestamps of the hyperlink and emoji chat messages were missing in the records. However, the fields that were supposed to contain these pieces were present. The artifacts were predominantly direct records of the user interactions in the application such as private messaging and team meetings. However, the records were accompanied by summary information displaying information about every user that has been part of the interactions. Additionally, information that can be utilized only by the application such as the “syncToken”, “continuationToken”, “latestSequenceId” were present in the storage. The records are resulted from the treatment and therefore can be attributed to the actions of the users are presented in this dissertation.

The application specific records are not included in the results. The detailed information about the listed artifacts is presented in the following subsections. Additionally, the significance of these artifacts for forensic investigations is scrutinized in the Discussions section.

Private Chat Messages

The entire content of the messages was successfully extracted from the IndexedDB storage. Moreover, information about the accounts composing the message and the time-date information was encountered. Time-date fields affiliated with the messages included details such as compose time, server arrival time, and client arrival time. Additional fields observed in the records indicated the status and identification of the messages. For instance, a "clientMessageId" field contained a unique number associated with each message. Additionally, fields such as "isRead",

"isSentByCurrentUser", and "isLastMessageFromMe" appeared to record interactions of the accounts with the chat box and the messages.

The messages that contain special content such as hyperlinks, emojis, and media attachments resulted in the creation of extra fields in the records while missing some of the existing fields from the clear text message records. The records generated with Message3-1 and Reply1 of the treatment are displayed in Table 16.

Table 16

Artifacts Extracted from Private Chat Message Records

Field	**Value at First Level**	**Value at Second Level**
conversationId	“19:73511d84-b721-43df-a2e5-a258bad6e76a_c1a1f0b8-d4eb-4c4e-8624-61121af94cb6@unq.gbl.spaces”	
clientMessageId	“3514591942426958357”	
contentType	“text”	
messageType	"RichText/Html"	
threadType	“NULL”	
imdisplayname	“account one”	
properties	“files”	{“type":"http://schema.skype.com/File" “baseUrl":"https://nau3203-my.sharepoint.com/personal/account1_na_edu/", "type":"jpg", "title":"microsoftteams.jpg", "state":"active", "objectUrl":"https://nau3203-my.sharepoint.com/personal/account1_na_edu/Documents/Microsoft Teams Chat Files/microsoftteams.jpg}
	“preview”	"previewUrl": https://us-api.asm.skype.com/v1/objects/0-cus-d15-6a6096d784a449cc28b604098fabc4ed/views/imgo”,

(continued)

Field	Value at First Level	Value at Second Level
		"previewHeight":933, "previewWidth":1400}
composetime	“2021-12-11T22:11:02.215Z”	
originalarrivaltime	“2021-12-11T22:11:02.215Z”	
clientArrivalTime	“2021-12-11T22:11:03.424Z”	
isDisabled	false	
isRead	T	

The records created for the messages that include hyperlinks and emojis contained binary data in between each word of the content field. When the binary data was filtered, the hyperlink and emoji contents were obtained completely. It is noteworthy to mention that a description from the source of the hyperlinks were also stored in the record. The type and a link to the source of emojis were acquired in HTML format. The time fields were set to NULL for these records. However, this might be a result of data being corrupted during extraction. Table 17 displays the record of the message containing Hyperlink1 and Emoji1.

Table 17

Artifacts Extracted from Private Chat Records with Hyperlink and Emoji Content

Field	Value at First Level	Value at Second Level
conversationId	“19:73511d84-b721-43df-a2e5-a258bad6e76a_c1a1f0b8-d4eb-4c4e-8624-61121af94cb6@unq.gbl.spaces”	
clientMessageId	“92172487201831629”	
contentType	“text”	
messageType	"RichText/Html"	

(continued)

Field	Value at First Level	Value at Second Level
threadType	“NULL”	
imdisplayname	“account one”	
content	“ Hyperlink1: https://www.microsoft.com/en-us/microsoft-teams/group-chat-software and Emoji1: ”	
properties	“links”	{“type":"http://schema.skype.com/HyperLink", "itemid":"https://www.microsoft.com/en-us/microsoft-teams/group-chat-software", "url":"https://www.microsoft.com/en-us/microsoft-teams/group-chat-software", "previewenabled":true,
	“preview”	{"previewurl":"", "title":"Video Conferencing, Meetings, Calling | Microsoft Teams", "description":"Microsoft Teams is the hub for team collaboration in Microsoft 365 that integrates the people, content, and tools your team needs to be more engaged and effective."}
composetime	“NULL”	

(continued)

Field	Value at First Level	Value at Second Level
originalarrivaltime	“NULL”	
clientArrivalTime	“NULL”	
isSentByCurrentUser	T	

A structured data storage was observed for records with media attachments. The first level enclosed data that is very similar to the data observed in clear text message records. A distinction between the records in this level was the "threatType" field which appeared to be set to a NULL value for the messages with media attachments. The second level provided detailed information about the attached media including name, type, and direct access links. Further scrutinization of the records and their value for digital forensic investigations is given in the Discussion section.

Team Creation

The record containing artifacts from the creation of ExperimentTeam1 had complete information on the name, description, and privacy status. Moreover, a field called “teamStatus” indicated the extent to which the additional applications were integrated into the team, including the status of usage for team notebooks, SharePoint, and exchange teams. If the applications were not utilized, their value was indicated as 1. If the applications were integrated, the value appeared as 2. Additional information such as the creation date, and whether more users had permission to join the team. The team records also incorporated information for the interactions of the members. A field called "memberProperties" specified the role of the suspect in the team. Whereas the "lastjoinat" displayed the epoch time when the last member joined it and the "lastMessage" delivered detailed information about the last message sent to the team.

Table 18 displays the artifacts extracted from the record containing information for ExperimentTeam1.

Table 18

Artifacts Extracted from Team Creation Records

Field	**Value**
teamAlias	“ExperimentTeam1”
description	“Team for Experiments”
teamStatus	“{exchangeTeamCreationStatus":1," sharepointSiteCreationStatus":2," teamNotebookCreationStatus":1}"
lastSyncTime	1633549595208
visibility	private
sharepointSiteUrl	“https://nau3203.sharepoint.com/sites/ExperimentTeam1557”
Shared Documents	"channelDocsFolder"
channelDocsFolderRelativeUrl	“2/sites/ExperimentTeam1557/Shared Documents/General”
teamSmtpAddress	“ExperimentTeam1557@nau3203.onmicrosoft.com”
createdat	1639260480976
joiningenabled	false
lastjoinat	1639260482517
memberProperties	Role: “Admin”
lastMessage	content: “ This is message1 ” messagetype: "RichText/Html" contenttype: “text” imdisplayname: "account one" composetime: “2021-12-11T22:09:01.998Z ” originalarrivaltime: “2021-12-11T22:09:01.998Z ” clientArrivalTime: “2021-12-11T22:13:27.264Z”
lastPrunedClearHistoryTime	X

(*) fields that were not related to users or their actions and fields that were not deemed significant for digital forensic investigations were not included in this table. (**) The value ‘x’ represents data not being available in the record

"lastPrunedClearHistoryTime", which is a field indicating the latest time the history of the meeting was cleared, displays "X" in Table 18. This is because the history of ExperimentTeam1 was not deleted in the treatment. It can also be observed in Table 18 that several links were available in this record such as the link to the team shared documents, email address, and SharePoint site URL.

Scheduled Meetings

Except for the particulars of the added team members, the information provided during the creation of a scheduled meeting was extracted completely from scheduled meeting records. The obtained information included the details of the time and location for which the meeting was scheduled, meeting type, privacy settings, join URLs, and suspect affiliation with the meeting. The artifacts observed in the scheduled meeting records can be seen in Table 19.

Table 19

Artifacts Extracted from Scheduled Meeting Creation Records

Field	**Value**
typeName	“CalendarEvent”
startTime	“2021-12-12T06:00:00+00:00”
endTime	“2021-12-18T06:00:00+00:00”
eventTimeZone	PacificSt
eventType	Occurrence
subject	“Meeting1”
location	"Missouri City TX"
private	“true”
skypeTeamsMeetingUrl	“https://teams.microsoft.com/l/meetup-join/19%3ameeting_YjU3Y2FkZGUtMzM5OS00NTdhLTg4MzItOTVhZmUyODQ4YjBl%40thread.v2/0?context=%7b%22Ti

(continued)

Field	Value
	d%22%3a%224001e375-31b4-4f03-b7d2-cb95ac6f5ff7%22%2c%22Oid%22%3a%22c1a1f0b8-d4eb-4c4e-8624-61121af94cb6%22%7d”
isOnlineMeeting	T
myResponseType	“Organizer”
organizerName	“account one”
isCancelled	0
isResponseRequested	T
isReminderSet	T
private	F
isHostless	true
content	“ This is the description for Meeting1 ”
eventRecurrenceRange	{"startDate":"2021-10-06T00:00:00-07:00", "endDate":null}
eventRecurrencePattern	{"patternType":"Weekly", "weekly":{"daysOfTheWeek":["Wednesday"],"interval":1}}
eventType	"RecurringMaster "

As it can be observed in Table 19, the repeating schedule patterns of scheduled meetings were reached through the “eventRecurrencePatern” field. For the weekly repeating meeting that was set in the treatment of the experiment, a subfield called “daysOfTheWeek” displayed a list data structure. This structure contained the days of the week for the meeting in clear text format. Additionally, a field called “location” was set to the value “Missouri City TX” as the location of the Meeting1 was set in the treatment of the experiment. A notable characteristic of the record was observed in the implementation variance of the balloon data types. "isCancelled" value was set to 0 while

the "isResponseRequested" value was set to "T", and the “isHostless” value was set to "true".

While these records contained fields indicating general information about a series of scheduled meetings, team meeting records appeared to contain event specific data for a single instance of scheduled meetings. Team meeting record content observed for Meeting1 is given in Table 20.

Table 20

Artifacts Extracted from Scheduled Meeting Event Records

Field	Value at First Level	Value at Second Level
contentType	“application/user+xml”	
activityType	“Event/Call”	
partlist	count=”4” <part> <displayName></displayName> <duration>1720</duration> </part> <part> <displayName>account three</displayName> <duration>1720</duration> </part> <part> <displayName>account one</displayName> <duration>1720</duration> </part> <displayName>account two</displayName> <duration>1720</duration> </part> </partlist>	
Recording	type	“Video.2/CallRecording.1”
	url_thumbnail	"https://us-prod.asyncgw.teams.microsoft.co

(continued)

Field	Value at First Level	Value at Second Level
		m/v1/objects/0-wus-d6-c3afdc4e5c6ea64cba189ed9e9cca71c/views/thumbnail"
	RecordingStatus	status="Success" code="200"
	SessionEndReason	value="CallEnded"
	<Title>Meeting1</Title>	
	Play	
	OriginalName	v="Meeting1-20211213_052035-Meeting Recording.mp4"
	RecordingContent	timestamp="2021-12-13T05:19:55.3021074Z" duration="0:00:32.14 " canVideoExpire="False"
	item	"https://us-prod.asyncgw.teams.microsoft.com/v1/objects/0-wus-d6-c3afdc4e5c6ea64cba189ed9e9cca71c/views/video"

Similar to private chat records with special content, meeting event records displayed in Table 20 were structured in two levels. The start and end times of the events were not present under a specified field whereas the second level fields associated with the recording of the event displayed the starting time and the duration of the recording correctly. Even though a participant list was displayed under the field "partlist", the duration data displayed the same number for every participant as "648". The value "648" did not reflect the correct participation duration of the accounts. In the treatment of the experiment, Account3 had left the meeting only to join back immediately after.

Therefore, the duration sub-field was expected to be different for Account3. The incident of Account3 leaving and rejoining the meeting was observed in the record as an extra participant without a set "diplayName".

Event Calls

The one-on-one video call requested by Account1 and accepted by Account2 resulted in the generation of a record that contained significantly different fields compared to the record from Meeting1. These fields can be observed in Table 21.

Table 21

Artifacts Extracted from Event Call Records

Field	Value at First Level	Value at Second Level
conversation	“48”	
participantList	null	
sessionType	”default”	
targetLink	“https://amer.ng.msg.teams.microsoft.com/v1/threads/48:calllogs"	
properties	startTime	“2021-12-13T05:08:48.4158327Z”
	connectTime	"2021-12-13T05:09:03.3848728Z"
	endTime	“2021-12-13T05:11:35.21407Z
	callDirection	“outgoing”
	callType	“twoParty”
	callState	“accepted”
	targetParticipantId	"73511d84-b721-43df-a2e5-a258bad6e76a"
originator	type	“default”
	displayName	"account one"

The distinct fields enclosed information about the direction, type, timing, state, and joining parties of the call event. Despite the Meeting1 record not containing explicit information on the timing of the meeting, the event call record emerged with definite timing fields such as "startTime", "connectTime", and "endTime". Despite the fact of two major fields "originator" and "targetParticipant" addressing both sides of the event call, the displayName sub-field of the "targetParticipant" was read as "null". However, another identifier called "targetParticipantId" was set to a thirty-six-character string value. This value can be observed to be assigned to Account2 in Table 23. Another noteworthy field in event call records is "targetLink" which indicates that call logs of the two-party event calls are recorded by the application.

White Board Contents and SharePoint Files

The creation record of WhiteBoard1 in ExperimentTeam1 was available in the IndexedDB storage. However, the contents added to it could not be found. It was observed that whiteboards were classified as tab extensions and recorded accordingly in the storage. The specific record displayed information about WhiteBoard1 with its name, type, access URL, and the time of its addition to the team space. This information can be seen in Table 22.

Table 22

Artifacts Extracted from White Board Records

Field	Value
type	“tab”
name	"WhiteBoard1"
directive	"extension-tab"
settings	{"subtype":"extension",

(continued)

Field	Value
	"url”:https://app.whiteboard.microsoft.com/me/whiteboards/cc6cd41c-22a8-4d7e-b58e-4f3f1da7e52b?embed=1&token=6f80cf64e70842e9b276f500d0e4de83_4001e375-31b4-4f03-b7d2-cb95ac6f5ff7_cc6cd41c-22a8-4d7e-b58e-4f3f1da7e52b&isOpenInTab=1”, "removeUrl":null, "websiteUrl":"https://app.whiteboard.microsoft.com/me/whiteboards/cc6cd41c-22a8-4d7e-b58e-4f3f1da7e52b", "dateAdded":"2021-12-11T22:13:26.725Z"}

Added Contacts

In the treatment of the experiment, Account2 was added as a contact. The artifacts from this action were encountered in a specific record that incorporated fields with repeating information. The fields such as "displayName", "givenName", "surname", "firstname_lowercase", "lastname_lowercase" appeared to enclose very similar information in slightly different formats. The record also included an identifier string that was utilized in the event call record between Account1 and Account2. It can be noted that details about the contact, e.g., it being a member and a person along with the information about the relation to the suspect, e.g., existing only for Microsoft Teams were stored in this record. Table 23 displays artifacts obtained from the added contact records.

Table 23

Artifacts Extracted from Added Contact Records

Field	Value
accountEnabled	T
alias	“account2”
mail	“account2@na.edu”
objectType	User

(continued)

Field	Value
isPrivateChatEnabled	T
coExistenceMode	“TeamsOnly”
smtpAddresses	"account2@nau3203.onmicrosoft.com"
isShortProfile	F
givenName	“account”
surname	“two”
email	“account2@na.edu”
displayName	“account two”
type	“person”
orgid	“73511d84-b721-43df-a2e5-a258bad6e76a”
firstname_lowercase	“account”
lastname_lowercase	“two”
fullname_lowercase	“account two”
guestlessDisplayName	“account two”
description	“ACCOUNT2”

Voice Mail

The voice mail left for Account1 by Account2 with the content "This is message1 for Account1" was converted to text by the application. The records about this text being delivered to Account1 were stored in a voice mail record. The voice message was converted to text imperfectly as "Hi, this is message 14, count 1". The converted text can be seen in Figure 3. The entire content of this converted message was successfully extracted from the storage. As similar to the rest of the records, timing information and

links pointing to the artifact content were also extracted from the voice mail record. The "orgid" field was used as an identifier for the account that left the voice mail. The value displayed for this field matches Account2 in Table 23. A distinct field called "activityProcessingLatency" was added to this record indicating the processing time the application takes to convert the audio to text format. The contents of the voice mail record can be observed in Table 24.

Table 24

Artifacts Extracted from Voice Mail Records

Field	Value
activityType	call
activitySubtype	“voicemail”
activityTimestamp	“2021-12-13T05:29:24.656Z”
orgid	“73511d84-b721-43df-a2e5-a258bad6e76a”
sourceUserImDisplayName	“account two”
messagePreview	“One for account one.”
activityProcessingLatency	102.1209
composetime	“2021-12-13T05:27:23.742Z”
originalarrivaltime	“2021-12-13T05:27:23.742Z”
clientArrivalTime	“2021-12-13T05:29:22.415Z”
conversationLink	“conversation/48:notifications”
messageid	1639373243742
messages	“https://amer.ng.msg.teams.microsoft.com/v1/users/ME/conversations/48:notifications/messages”

Suspect Account Configurations

Beyond the results of the experiments, a record containing user account configuration settings was discovered during the extraction process. This record contained information about the type of the account, resource settings, and the policies adopted for calls and messages. The artifacts discovered in the account configuration records are displayed in Table 25.

Table 25

Artifacts Extracted from Suspect Account Configuration Records

Field	Value
licenseType	"Student"
userType	"member"
accessType	"UnrestrictedAccess_TeamsApp"
mailboxStatus	"Discoverable"
autoAnswerEnabledType	"account two"
callingPolicy	allowPrivateCalling:T allowGroupCalling:T allowCallForwarding:T allowVoicemail:T
userResourcesSettings	isOrganizationTabEnabled:T isSkypeBusinessInteropEnabled:T isVideoEnabled:T isScreenSharingEnabled:T isSharePointEnabled:T isExchangeEnabled:T isOfficeEnabled:T isOneDriveForBusinessEnabled:T isProjectWorkManagementEnabled:T isMailboxDiscoverable:T
messagingPolicy	allowUserChat:T allowGiphyDisplay:T

(continued)

Field	Value
	allowPasteInternetImage:T allowMemes:T allowStickers:T allowUserTranslation:T allowUrlPreviews:T allowPasteInternetImage:T allowUserEditMessage:T

Time Frames

The PHP software utilized to order the records with database queries yielded a list of actions. These actions can be seen in Table 26.

Table 26

Time Frame Ordered List of the Extracted Action Records

Time	Duration	Action	Participants
2021-12-11 22:08:02	Instant	Team Creation	Account1
2021-12-11 22:09:01	Instant	Chat Message	Account1
2021-12-11 22:11:02	Instant	Chat Message	Account1
2021-12-11 22:13:26	Instant	Extension Tab Creation	Account1
2021-12-11 22:21:28	Instant	Meeting Creation	Account1
2021-12-13 05:08:08	Instant	Chat Message	Account1, Account2
2021-12-13 05:08:48	167	Video Call	Account1, Account2
2021-12-13 05:10:11	Instant	Chat Message	Account1, Account2
2021-12-13 05:19:18	Instant	Chat Message	Account1, Account3
2021-12-13 05:19:55	32	Meeting Recording	Account1
2021-12-13 05:29:24	Instant	Voice Mail	Account1, Account2

(*) Duration field is in seconds

The order is based on the time the action was taken. The details of the fields selected for this construction are discussed in the methodology section. It appeared that the application system saved time information in a different time zone. A seven-hour difference between the display time of the experimental environment and the saved record was observed. The listed actions include records where time information was available. The time information fields in the records of chat messages with hyperlink and emoji content were encountered as NULL. Therefore, they were not listed in Table 26. Similarly, some of the participants are missing from their corresponding actions due to incomplete data in their records.

Discussion

The artifacts created in the IndexedDB storage as a result of the treatment contain information about the interactions of users with the content, time, and configuration setting information. Several useful pieces of the artifacts such as the timestamps of the hyperlink and emoji chat messages from the treatments were missing. However, the fields that were supposed to contain these pieces were present. The intent of the implementation appeared to store these artifacts completely as well. As a result, enough artifacts were gathered to display most of the interactions to the digital forensic examiners along with sufficient time information to create time-frame analysis. Additional information was attached to the records, i.e., the last sent message of the teams were present with timestamps in the team creation records.

Even though the treatment of the experiment was designed to isolate the user account interactions in the application, additional information about account configurations was observed. These configurations were the side artifacts of the

treatment. For instance, when the user allowed the application to accept voice mails, corresponding records were also populated with unrelated account configuration information as well.

Artifacts observed in this work included information on shared activities. Interactions such as chat messages and attendance at scheduled meetings are shared activities with the rest of the team. Therefore, the involvement of the accounts other than the suspect account is an additional opportunity for investigators to conduct investigations that would otherwise require additional warrants.

The artifacts displayed potential for further forensic analysis that is outside the scope of this work. With scrutinization of the IndexedDB storage for Microsoft Teams applications, it is possible to establish connections between the user accounts. Their frequency of interactions, contents of the private messages shared between them, and the time periods in which they interact can be created from driving pattern analysis of the records. These analyses are often utilized for the investigation of collective criminal activity such as organized crime coordination, drug transactions, and collective bullying (Brunty & Helenek, 2014; Pyrooz & Moule, 2019).

Another potential investigation opportunity provided with Microsoft Teams IndexedDB storage is the emoji content extracted from the chat messages. Emoji content and usage frequency is a concept used by researchers to the evaluate mental states of the users (An et al., 2018; Marengo et al., 2017). The emotional responses of corporate employees can be analyzed when the point of interest activities was shared with them, potentially indicating willingness and eagerness to participate in criminal activities.

Various fields of the artifacts contained time information about the interactions of the accounts. These fields gave a detailed look into when actions were taken and when they were reflected in the application. For instance, message composing times and message arriving times were indicated separately within the records. The time stored in the application was not the displayed time of the user. For this experiment, it was seven hours ahead of the experimental environment's displayed time. However, the time zone of the records was indicated in the "eventTimeZone" field of the meeting creation records. This provides the forensic investigators the means of adjusting time artifacts before starting to analyze the time artifacts.

IndexedDB Artifacts of Instagram

The treatment of the experiment produced identical artifacts in the IndexedDB storage of Mozilla Firefox and Google Chrome browsers. The preponderance of artifacts obtained from the experiments were the results of posts or comments of users being displayed on the home page. Consequently, they were scrutinized for their potential value to the forensic examinations. The encountered artifacts almost exclusively belonged to users and their interactions on the application. A summary of the created artifacts is seen in Table 27 with their corresponding treatment action.

Table 27

Artifacts Created in IndexedDB Storage of Instagram by Preliminary Activities

Steps	Activity	Artifacts
1	A post of a followed account was displayed on the home page	• Posting time of the post • Owner identification artifacts • Artifacts indicating the relationship and permissions between the posting and viewing account

(continued)

Steps	Activity	Artifacts
2	A comment over the post of a shared connection account was displayed on the home page	• Entire content of the comment • Posting time of the comment • Statistic of the interactions of the comment such as like count • Owner identification artifacts • Artifacts indicating the relationship and permissions between the posting and viewing account • Direct link to displayed profile picture
3	An account profile was displayed by hovering the mouse over its comment or post	As additions to the previous field: • Full Name, Biography, and other detailed information from of the posting account, e.g., its saved website • Status information of the posting account, e.g., account being new or private
4	An account page was visited after displaying a post or comment from the home page	As additions to the previous field: • Direct link to high-definition profile picture • Artifacts indicating whether it is possible to display account contacts • Account categories and classification
5	An account page was visited without displaying a post or comment from the home page	No artifacts encountered
6	An account was blocked and a post with a comment from this account was displayed on the home page	Artifacts indicating the block status between the posting and displaying accounts
7	An account was restricted and a post with a comment from this account was displayed on the home page	Artifacts indicating the restriction status between the posting and displaying accounts
8	A direct message was sent to a followed account with emoji content	Emojis and the number of times they were used
9	A direct message was displayed form a followed account	No artifacts encountered
10	A story of a followed account was displayed from the home page	• Time artifacts for the story such as expiry date, post date, last post date • Owner identification artifacts • Artifacts indicating the relationship and permissions between the posting and viewing account

(continued)

Steps	Activity	Artifacts
		• Artifacts carrying technical information about the story, e.g., whether it is muted
11	An emoji reaction was added to the story of a followed account	No artifacts encountered

Treatments that have not created meaningful artifacts are not listed in Table 27. The obtained artifacts are presented in the following subsections. The value and peculiar characteristics of the artifacts are discussed in detail in the Discussions section.

Users.users

When the home page is accessed, users.users record is populated with artifacts from the connections that have recent posts on display. These artifacts contain profile information of the connections and their mutual followers with the suspect account. Surprisingly, the artifacts are not limited to the followed connections, but to all the accounts that made a comment on the displayed posts. At initial display, generated artifacts are limited to several fields. However, as the suspect interacts with the posting account, the number of collected artifacts increases drastically. Additionally, when a post or comment is not displayed on the home page, visiting the posting account's profile page does not populate any records in IndexedDB storage.

Utilizing users.users records, a digital forensic investigator can obtain information about the connections of the suspect. Including their mutual followers, profile pictures, and account characteristics. More on how these fields can be useful are given in the Discussion section.

Table 28 gives details on the artifacts contained in users.users for three cases of suspect interactions with the posting accounts.

Table 28

Artifacts Found in users.users Record with User Profile Interactions

Attribute	Displaying posts/comments without interaction	Displaying posts/comments and hovering over the posting account with mouse	Displaying posts/comments and visiting the posting account's profile page
bio:	x	"Bio of Forensic Researcher 3"	"Bio of Forensic Researcher 3"
followedBy:	x	0	0
follows:	x	1	1
fbid:	x	x	"17841448515262719"
fullName:	x	"Forensic Researcher 3"	"Forensic Researcher 3"
id:	"48581753175"	"48581753175"	"48581753175"
isNew:	x	FALSE	TRUE
isPrivate:	x	FALSE	FALSE
mutualfollowers:	x	An empty list	An empty list
profilePicUrl:	A (lengthy) link to profile picture is obtained	A (lengthy) link to profile picture is obtained	A (lengthy) link to profile picture is obtained
username:	"forensicresearchaccount 3"	"forensicresearchaccount 3"	"forensicresearchaccount 3"
website:	x	"http://forensicresearcher 3.com/"	"http://forensicresearcher 3.com/"

The 'x' entry is inserted for artifacts that were not present for their corresponding case. The users.users record contains more fields. However, most of the fields were not containing any meaningful information for this experiment. Therefore, they were omitted from Table 28.

Relationships

The relationships record contains information about the following and the blocking status between the accounts. For Account 1 and Account3, there was no connection according to the treatment. Account3 was blocked and restricted by Account1 as described in the Materials and Methods section. Artifacts obtained from two different cases are displayed in Table 29.

Table 29

Artifacts Found in relationships Record with User Profile Interactions

Attribute	Account1 and Account3	Account1 and Account2
blockedByViewer:	"BLOCK_STATUS_UNBLOCKED"	"BLOCK_STATUS_BLOCKED"
followedByViewer:	"FOLLOW_STATUS_NOT_FOLLOWING"	"FOLLOW_STATUS_NOT_FOLLOWING"
followsViewer:	"FOLLOW_STATUS_NOT_FOLLOWING"	"FOLLOW_STATUS_NOT_FOLLOWING"
hasBlockedViewer:	null	null
restrictedByViewer:	"RESTRICT_STATUS_UNRESTRICTED"	"RESTRICT_STATUS_RESTRICTED"

It can be seen in Table 29 that the information of the suspect following, blocking, and restricting another account can be obtained from the relationships record. However, the blocking status of the accounts for the suspect account was obtained as null.

Comments.byId & comments.byPostId

When comments are displayed on the home page of an account, comments.byId records are created. As users.users records contain information about the owners of the comments, and supplemental information specific to the comments is stored in comments.byId records. The entire text content of comments can be found in this record.

Additionally, important time information of when the comment is posted can be obtained in epoch time with the postedAt field of the record. An instance of this record can be seen in Table 30 for the comment of Account3 to the post of Account2.

Table 30

Artifacts Found in comments.byId Record

Attribute	Value
deleted:	FALSE
didReportAsSpam:	FALSE
id:	"17884757831263207"
isAuthorVerified:	FALSE
isRestrictedPending:	FALSE
likeCount:	0
likedByViewer:	FALSE
postedAt:	1625222296
text:	"Account3 comment for Account2 - Post1"
userId:	"48581753175"

comments.byPostId record targets not a specific comment, but a summary of all the comments made for a post. It displays the count and display information, e.g., how many of the comments are visible on the post. In addition to an overall look into the comments, an array list of all the account ids is also stored in this record.

Posts.byId

Similar to comments, information about every post displayed on the home page is also recorded in IndexedDB storage of Instagram. posts.byId record contains detailed information for posts including their location and posting time. Another noteworthy

information for connecting suspects and the evidence is in the viewerInPhotoOfYou and ownerfullName fields. The owner of the post and whether the suspect is involved in the post can be obtained from these fields. Table 31 shows the posts.byId record for Post1 of Account2.

Table 31

Artifacts Found in posts.byId Record

Attribute	**Value**
accessibilityCaption:	"Photo by Forensics Researcher 2 in Missouri City, Texas."
caption:	"Forensic Researcher 2 - Post 1"
commentsDisabled:	FALSE
followHashtagInfo:	null
hasAudio:	TRUE
isVideo:	FALSE
likedByViewer:	FALSE
likers:	An empty list
location->id:	"228672033"
lat:	undefined
lng:	undefined
location->name:	"Missouri City, Texas"
slug:	"missouri-city-texas"
numComments:	1
numPreviewLikes:	1
owner->fullName:	"Forensics Researcher 2"
owner->id:	"16009265888"
isNew:	FALSE

(continued)

Attribute	Value
isPrivate:	FALSE
username:	"forensicresearchaccount2"
postedAt:	1625222164
previewCommentIds:	0 -> "17884757831263207"
savedByViewer:	FALSE
usertags:	An empty list
viewerCanReshare:	TRUE
viewerInPhotoOfYou:	FALSE

Users.usernameToId & users.viewerId

The stored data, which is associated with posts and comments of Instagram users, are linked to their owner accounts through user ids. users.usernameToId record provides the connection between the usernames and the user ids. Table 32 displays information from users.usernameToId record for the usernames and user ids of the accounts utilized in the experiments.

Table 32

Artifacts Found in users.usernameToId Record

Attribute	Value
awesome.photographers	"1077125"
forensicreasearchaccount1	"46912168943"
forensicresearchaccount2	"16009265888"
forensicresearchaccount3	"48581753175"

users.viewerId record is a single field record that indicates the id of the user for whose account the IndexedDB storage is populated. This can be matched to the username of the suspect in the users.usersnameToId record. For Account1, users.vieverId was recorded as "46912168943”.

Direct.emojis

direct.emojis record stores the emojis used by the user whose account is under investigation. The actions that populate the direct.emojis record includes direct messages, comments, and posts. The record stores the emoji and the number of times it is used. However, the direct message or account for which the emoji is used is not specified. This record is not populated when an emoji is used, sent, or displayed from another account’s entries or direct messages. It is also noteworthy to remark that when a reaction is given to a story or a direct message with emojis, no data is inserted to direct.emojis record. Table 33 gives the contents of this record from the experiments.

Table 33

Artifacts Found in direct.emojis Record

Emoji	Number of Times It is Used
☹	1
⛀	1
㊙	1

Stories.feedTray & stories.reels

The story items displayed on the home page yield artifacts in stories.feedTray and stories.reels records. The stories.feedTray lists the user ids of all the accounts for which a story is displayed on the home page. stories.reels contains details of the stories.

Considerable fields in the stories.reels record emphasizes times of interactions. The seen attribute gives the epoch time of when the user displayed the story for the first time. latestReelMedia takes it one step further and gives the time of post for the latest story of the account. expiringAt attribute represents the time when the story will be out of the display. Furthermore, at what position and what order the story was seen is recorded in seenRankedPosition attribute. stories.reels record also has attributes similar to attributes of post records including location information and the abilities of the viewing account on the story. Table 34 displays the stories.feedTray record for Account1 where there is only one story from Account2.

Table 34

Artifacts Found in stories.feedTray Record

Attribute	**Value**
id:	"16009265888"
length:	1

Table 35 displays stories.reels record for Story1 posted by Account2.

Table 35

Artifacts Found in stories.reels Record

Attribute	**Value**
canReply:	TRUE
canReshare:	TRUE
expiringAt:	1625683666
id:	"16009265888"
isCloseFriends:	FALSE

(continued)

Attribute	Value
latestReelMedia:	1625597266
locationId:	undefined
muted:	FALSE
rankedPosition:	1
seen:	1625597266
seenRankedPosition:	1
tagName:	undefined
title:	undefined
userId:	"16009265888"

Discussion

The artifacts encountered in the IndexedDB storage of Instagram Web are primarily created from the interactions on the home page. Posts, comments, and stories displayed on the home page populate the IndexedDB storage with data such as owner identification, account relationships, permissions, and direct link to post resources, e.g., direct links to profile pictures. The entire contents of comments and post descriptions, post locations, and user tags were able to be extracted from the storage. The number of artifacts belonging to an account also increases when more interactions with their profile page are provided. The additional artifacts that can be obtained with supplemental profile interactions exhibit the full name, biography, saved website, status, and category of the accounts. It is also possible to obtain statistical information from the account profiles such as the number of followers and the number of accounts followed. Furthermore, significant information pertinent to the accounts, e.g., whether they are new, private,

professional, unpublished, or verified accounts can be obtained through the profile page interactions in the case their posts or comments are displayed on the home page.

With scrutinization of IndexedDB storage for the Instagram Application, it is possible to create connections between the account owners. A dedicated record called relationships contains valuable information that can be utilized, along with the information from the users.users record, for the construction of maps that can indicate the place of suspects in their social networks. Furthermore, the level of the relationships can be estimated based on attributes such as viewerInPhotoOfYou attribute from posts.byId records and is-CloseFriends attribute from stories.reels records. These connections and their strength are valuable to forensic investigations as social media applications are often utilized for criminal activity of drug transactions, organized crime coordination, and cyber bullying (Pyrooz & Moule, 2019). It creates opportunities to detect the accomplices of a crime and to collect information on the posts, and comments of the accounts that are private to the public would require additional warrants.

Artifacts created by the usage of user stories contained an additional value for the establishment of user connections. Mainly, they contain an attribute called isCloseFriends. Additionally, it is possible to obtain information on when a story was displayed and in what order it was seen. Utilizing the artifacts extracted from users.users, users.usernameToId, stories.reels, and relationships records, we were able to create a connection analysis between the accounts employed in the experiments. The techniques for establishing connections between Instagram accounts are discussed in the proof-of-concept tool section.

By examining the IndexedDB artifacts of Instagram, examiners can also detect suspect account's focus and interests on social networks. If the contents of a post or comment on the home page of the account contain an indication of a criminal activity or tendency to criminal behavior, likedByViewver, savedByViewer, savedByViewerToCollection attributes from the posts.byId record can be utilized to detect any interest in these posts or comments.

Another potential subject of interest for the behavioral analysis is the direct.emojis records. Researchers have been utilizing emoji analysis to detect the personality and mental states of users since it gained popularity (Marengo et al., 2017). It is particularly important for criminal behavior analysis. The direct.emojis records provide the emojis used by the suspect account and the frequency of usage for each emoji. This frequency can be practical, e.g., in cases, a sad face is used extensively, or in cases of a religious symbol is used with a similar frequency to emojis that indicate negative feelings.

Artifacts obtained from postedAt attribute of posts.byId, expiringAt and seen attributes of stories.reels, and postedAt attribute of comments.byId display time data in UNIX epoch time format. It is a frequently encountered format since it eliminates the need for time conversions from different time zones. Additionally, the post captions contain information on the share time of the posts. These attributes can be utilized in the time frame analysis of forensic investigations.

IndexedDB Artifacts of WhatsApp Web

The treatment applied to the subject has resulted in the creation of artifacts in the IndexedDB storage file of the Chrome browser. It is observed that a great number of artifacts are created with the treatment. The application keeps records of technical operations, such as sync and async of the device, acknowledgment from the server about the contact availability, etc. Due to a large number of artifacts, the results were narrowed down to the records that can serve as important evidence for the investigations. These records deemed significant are listed in Table 36 with their triggering treatment activities.

Table 36

Artifacts Created in IndexedDB Storage of WhatsApp Web by Treatment

Steps	Activity	Artifacts
1	Sending text messages from Phone1 to Phone2 and vice versa	• Send Action Message Chat Record • Recv Action Message Relay Chat Record
2	Sending media messages including video and pictures; pictures including jpeg files and gif files and displaying transferred files	• Media Load on Loaded Data Record
3	Starting a video/voice call from Phone1 to Phone2 and vice versa	• Recv: s<Number> [Call, ...]
4	Walking away with Phone1 from PC1 and returning to the connection distance	• Action Presence Unavailable Record • Action Presence Available Record
5	Disconnecting with Phone1 from PC1 and reconnecting	• Stream:rememberMe: true Record
6	Initiating the application and navigating to its tab on browser	• Network Status Online Record

Every significant record obtained from the storage is explained in the following subsections.

Network Status Online

When the user's tab on the browser containing WhatsApp Web Application is active, the Network Status Online record is created in the IndexedDB storage. This record includes repeated information of record labels and the time stamp on its body. An instance on this record can be found in Figure 10.

Figure 10

Network Status Online

{line: 28, log: "_T#= 2020-02-14 13:56:31.326:NetworkStatus online", timestamp: 1581710192170.54}
line: 28
log: "_T#= 2020-02-14 13:56:31.326:NetworkStatus online"
timestamp: 1581710192170.54

NetworkStatus online carries a forensically significant value as it is an indicator of the time user is interacting with the application. In an investigation, the charges against a suspect often rely on the timestamps of the evidence (Schatz et al., 2006). Matching time settings between the evidence timestamps and user interaction with the computer can serve as a strong indicator of a suspect is responsible for the evidence. As can be seen in Figure 10, there are both epoch and date string time stamps on the record.

Stream:rememberMe:true

Stream:rememberMe record is created, when a user establishes the connection between a computer browser embodying WhatsApp Web Application and a phone with WhatsApp Messenger. The record includes information on the record label and a time stamp on its body. Stream:rememberMe is used when the suspect wants WhatsApp

Messenger to remember the computer. It is a sign that the computer is used frequently by the suspect. Digital forensic investigation reports include information on the behavior of the suspect. This information is presented to the court to make better sense of the suspect's motives. A suspect's frequent interaction with the evidence computer is important as it gives an inside into the behavioral characteristics of the suspect. Figure 11 shows an instance of this record.

Figure 11

Stream:rememberMe Record

{line: 29, log: "_T#= 2020-02-14 13:56:31.387:Stream:rememberMe: true", timestamp: 1581710192170.58}
line: 29
log: "_T#= 2020-02-14 13:56:31.387:Stream:rememberMe: true"
timestamp: 1581710192170.58

Media Load on Loaded Data

MediaLoad:video.onloadeddata record is created, when a media file such as a video file is opened on either messenger or web application. Timestamps indicating the opening time of the media are placed in the record with the record label. Figure 12 shows an instance of media load records.

Figure 12

Media Load on Loaded Data Record

{line: 1008, log: "XoL+ 2019-04-29 22:42:10.331:MediaLoad:video.onloadeddata #3", timestamp: 1556595730878.635}
line: 1008
log: "XoL+ 2019-04-29 22:42:10.331:MediaLoad:video.onloadeddata #3"
timestamp: 1556595730878.635

Many cases involving digital evidence are related to illegal images and videos of minors (Pollitt, 2010). There are also cases of privacy issues between couples involving the distribution and exposition of private media. In these cases, the information on when the media is accessed and how frequently it is accessed is crucial to the investigation (Kao, 2016).

Recv: s<Number> [Call, ...]

Recv: s<Number> [Call, ...] record is created when a video or voice call is active using either messenger or the web application. Timestamp information of when the calls are active is included in the data with the record label. The application puts this log to the file multiple times during the call is active. It is also observed that a missed call can produce the Recv: s<Number> [Call, ...] record. An unanswered voice call that rings for 1 minute is seen to generate over 9 instances of the record. Figure 13 demonstrates this record type with an instance.

Figure 13

Recv: s<Number> Record

{line: 244, log: "_S+X 2020-06-20 06:19:26.946: recv: s67 [Call, ...]", timestamp: 1592651969452.665}
line: 244
log: "_S+X 2020-06-20 06:19:26.946: recv: s67 [Call, ...]"
timestamp: 1592651969452.665

The time frame of a video or voice call being active can provide information on when important calls are made. It can be further useful particularly in the cases where the investigation is taking place with both parties of the call.

Action Presence Available - Unavailable

action,presence,unavailable record is created when the user walks away from the computer with the phone connected through WhatsApp Messenger. The beginning of the time user steps away from the computer is recorded with the record label. It should be noted that when the connection between the phone and the computer is problematic, a presence unavailable record is occasionally added. During the treatments, one such record has been observed. It is also notable that when the duration of the user's absence is long, multiple records are added. Similarly, user available record is occasionally added without the user stepping away from the computer. Figure 14 shows an instance of this record type.

Figure 14

Action Presence Unavailable Record

{line: 1137, log: "_T#= 2020-02-14 14:38:35.941:sending: 1581712715.--144, action,presence,unavailable", timestamp: 1581712716354.625}
line: 1137
log: "_T#= 2020-02-14 14:38:35.941:sending: 1581712715.--144, action,presence,unavailable"
timestamp: 1581712716354.625

In cases involving digital evidence, one of the very common defenses is the claim of the suspect not being the person using the computer at the time of the offense. Personal devices such as mobile phones are more likely to be used by a single individual. Although computers can be used by more than one user, this is more of a case in business settings with printing, database, and common purpose computers. A record displaying when the suspect walks away from the computer is potentially significant information to support or be against the defense of not being present during the time offense takes place.

Send Action Message Chat and Recv Action Message Relay Chat

send<code>,action,message,chat and recv<code> action,msg,relay,chat records are created when a text message is sent from one account to another. The application keeps more detailed information on the handshake of the device to send the message and further details on its delivery. However, these messages vary in means of network delays and server states. Plain Send - Receive record is observed to be sufficient to pinpoint a messaging activity for the time frame analysis. Figure 15 shows an instance of this record type.

Figure 15

Send Action Message Record

{line: 697, log: "^XX 2020-06-07 07:52:45.803: send: 3EB0A2F3697…6B3365, action,message,chat,,3EB0A2F36976646B3365", timestamp: 1591534369731.81}*
line: 697
log: "^XX 2020-06-07 07:52:45.803: send: 3EB0A2F36976646B3365, action,message,chat,,3EB0A2F36976646B3365"*
timestamp: 1591534369731.81

The Send - Receive records contain timestamps and record labels. Additionally, an identifier code such as 3EB0A2F3697…6B3365 is present in the record. This code seems to contain information on the account the message is sent to, or the account the message originated from. Therefore, it can be used for separating the conversations into different accounts. However, there is no identifiable way to determine what account the code belongs to.

In digital investigations where the examiner does not have access to the suspect's WhatsApp credentials, the information of when the suspect sent a message and received a message can be useful for determining his actions at the given time.

Discussion

The artifacts created in the IndexedDB storage by WhatsApp Web Application appear to provide extensive information on the actions of a user. This information is recorded in a format including the time of the action in the UNIX epoch time format in addition to the human-readable format. The records are divided by numbering which

makes them easier for parsing. It can be observed in Figures 8 to 13 that the information repeats in different formats. The first set of records represents the information in brackets whereas the subsequent record separates the time, label, and line number into different lines. This is indicating a design that intends to support different methods of information collection.

It is observed that the actions taken with the WhatsApp Messenger Application on the phone are recorded in WhatsApp Web IndexedDB storage during an active connection. If a user answers a video call or watches a video through the application from the phone, its information record will be found on the computer.

As the actions taken by a user in WhatsApp Messenger and Web Applications are stored in a LevelDB file that can easily be parsed and manipulated by the suspect, a concern for privacy can be raised. Even though there is no conversation saved directly into this file, the timeline of a person viewing media files and the information of the times they are spending with their computers can easily be calculated.

Evaluation of IndexedDB as a Verification Source

The artifacts extracted from both traditional and persistent storage were compared for the identification of verification items. The extracted artifacts were verified to be the products of the experimental treatment. Table 37 shows the traditional browsing artifacts of the treatment with their verification items from IndexedDB storage artifacts. Traditional browsing artifacts are obtained from the forensic investigation tool Autopsy by exporting “Operating System Information”, “Web Accounts”, “Web History”, “Web Search”, and “Web Form Autofill” tables. All related information from these tables is incorporated into Table 37.

Table 37

IndexedDB Verification Items of Traditional Browser Artifacts

Browser	Domain	Verification Item	Traditional	Persistent
Chrome	Bing	Search Term	Search term1	query..search.term1..inputsa.
Chrome	Bing	Identifier Email	fxp017@shsu.edu	userPrincipalName.. fxp017@shsu.edu
Chrome	Bing	Identifier Name	Furkan Paligu	displayName..Paligu..Furkan
Chrome	Youtube	OS	Windows	..osName..Windows..
Chrome	Youtube	OS Version	10	osVersion..10.0.
Chrome	Youtube	Browser	Google Chrome	..browserName..Chrome.
Chrome	Youtube	Browser Version	102.0.5005.63	.browserVersion..102.0.5005.63
Chrome	Youtube	Search Term	search term1	search_query.3Dsearch.2Bterm1.
Chrome	Youtube	History	2022-05-30 20:50:59	requestTimeMs..1653961858708
Chrome	Reddit	OS	Windows	..osName.:.windows
Chrome	Reddit	Browser	Google Chrome	browserName.:.chrome
Chrome	Reddit	Browser Version	102.0.5005.63	browserVersion.:.102.0.5005.63
Chrome	Reddit	History	2022-05-30 21:10:33	createdTimestamp.:1653963065843..
Chrome	Instagram	Identifier Username	forensicresearch account2	o..forensicresearchaccount2..
Edge	Youtube	OS	Windows	osName..Windows
Edge	Youtube	OS Version	10	osVersion..10.0
Edge	Youtube	Browser	Microsoft Edge	browserName..Edge.Chromium
Edge	Youtube	Browser Version	101.0.1210.53	browserVersion..101.0.1210.53
Edge	Youtube	Search Term	search term1	search_query.3Dsearch.2Bterm1.
Edge	Youtube	History	2022-05-30 21:57:08	requestTimeMs..1653965820228..

(continued)

Browser	Domain	Verification Item	Traditional	Persistent
Edge	Instagram	Identifier Username	forensicresearchaccount2	o..forensicresearchaccount2..
Edge	Netflix	History	2022-05-30 22:28:55	0sessionId1653967735
Firefox	Bing	Identifier Email	fxp017@shsu.edu	PrincipalName.O@fxp017@shsu.edu
Firefox	Bing	Identifier Name	Furkan Paligu	display.2@Paligu..Furkan
Firefox	Instagram	Identifier Username	forensicresearchaccount2	o..forensicresearchaccount2..
Firefox	Youtube	OS	Windows	&cos=Windows
Firefox	Youtube	OS Version	10	&cosver=10
Firefox	Youtube	Browser	Mozilla Firefox	cbr=Firefox&cbrver=100.0
Firefox	Youtube	Browser Version	100.0.2	cbr=Firefox& cbrver=100.0
Firefox	Youtube	History	2022-05-30 23:04:40	SAPISIDHASH.1653969882
Firefox	Reddit	OS	Windows	osNRWindows
Firefox	Reddit	OS Version	10	"osVersion":"10"
Firefox	Reddit	Browser	Mozilla Firefox	cbr=Firefox
Firefox	Reddit	Browser Version	100.0.2	&cbrver=100.0
Firefox	Netflix	History	2022-05-30 23:00:29	STANDARD.1653969641574

Since the LocalStorage artifacts are similar for all three web browsers, only Google Chrome verification items are given in this section. These verification items are listed in Table 38.

Table 38

LocalStorage Verification Items of Traditional Browser Artifacts

Browser	Domain	Verification Item	Traditional	Persistent
Chrome	Bing	Browser	Google Chrome	browserNam.Chrome
Chrome	Bing	Browser Version	102.0.5005.63	:0102.0.5005.63.
Chrome	Bing	Identifier Email	fxp017@shsu.edu	userPrincipalName.:.fxp017@shsu.edu
Chrome	Bing	Identifier Name	Furkan Paligu	displayName.:.Paligu..Furkan
Chrome	Bing	Visited Search Result	What are search terms?	path.:.what.are.search.terms
Chrome	Bing	Identifier IP	98.197.202.192	._Hip.:.98.197.202.192
Chrome	Bing	Identifier Location	US..TX Missouri City 77489	nae..United.St. subdivis._..Texas z.k.77489.cit.Missouri.C
Chrome	Youtube	History	2022-05-30 20:50:59	Mon.May.30.2022.21:58:24.GMT.0700..Pacific.Daylight.Time
Chrome	Google	Search Term	Search term 1	search.term.1..35..362.39
Chrome	Yahoo	History	2022-05-30 20:57:34	https://search.yahoo.com/ 1653962236
Chrome	Yahoo	Visited Search Result	What are search terms?	path.:.what.are.search.terms
Chrome	Wikipedia	Search Term	Forensic science	.q:forensic.scienc
Chrome	Wikipedia	History	2022-05-30 21:20:45	D.1654050045377
Chrome	Reddit	Search Term	funny videos	Quer4 funny.videos
Chrome	Reddit	Visited Search Result	/r/MadeMeSmile/	url.:./r/MadeMeSmile/
Chrome	Office	History	2022-05-30 21:18:31	1653963512..
Chrome	Office	Identifier Email	fxp017@shsu.edu	upn.:.fxp017@shsu.edu
Chrome	Myshopify	Identifier Username	forensicresearch1	META: forensicresearch1.myshopify.com

(continued)

Browser	Domain	Verification Item	Traditional	Persistent
Chrome	Myshopify	History	2022-05-30 21:24:51	Wed.30.May.2022.21:24:48.
Chrome	Ebay	Search Term	Search term1	m570.l1313._nkw.Search.term1._sacat.0
Chrome	Chase	History	2022-05-30 21:33:54	currentTimestamp.:1653964435199
Chrome	Chase	Browser	Google Chrome	browserId.:.gc
Chrome	Chase	Browser Version	102.0.5005.63	102.0.5005.63
Chrome	Microsoft	History	2022-05-30 21:34:51	Mon.May.30.2022.19:34:51.GMT.0700.Pacific.Daylight.Time
Chrome	Netflix	History	2022-05-30 21:40:16	lastDeviceInfoTime.:1653964831376
Chrome	Instagram	History	2022-05-30 21:41:46	cu_sessions.1653964907026
Chrome	Facebook	History	2022-05-30 21:09:38	xfa0vl:1653963047428

Table 39 displays how Mozilla Firefox and Microsoft Edge verification items differ from Google Chrome.

Table 39

LocalStorage Browser Verification Items Compared to Google Chrome

Browser	Missing Items	Additional Items	Notes
Firefox	Yahoo - Visited Search Result Myshopify - All	Reddit – History Bing - History	
Edge	Office - History	Google – History	Bing exists as MSN in Edge

The differences regarding the content in the persistent storage technologies of different browsers is due to both differences in storage and data not being identified

properly. Artifacts are mashed with binary data storage files. Consequently, as artifacts are moved from .log files to .ldb files in LevelDB, the level of accessible content changes. Additionally, some items are placed differently in SQLite files compared to LevelDB in terms of binary and clear text data.

It was observed, in terms of storage content dissimilarities, that Microsoft Edge keeps the bing domain artifacts in the MSN domain. Even though bing has IndexedDB artifacts in Google Chrome and Mozilla Firefox browsers, MSN IndexedDB content is close to nonexistent in the Microsoft Edge browser. Another missing IndexedDB storage for Microsoft Edge is Reddit. Even though there are LocalStorage artifacts of Reddit, no IndexedDB storage was observed in the Microsoft Edge browser.

The history time match between the traditional and persistent storage artifacts was mostly obtained from time strings to epoch time comparisons. Epoch time artifacts were converted to time strings and checked with their corresponding browser and domains. It was observed that a difference in terms of milliseconds exists between the timestamps of traditional and persistent storage artifacts. For instance, the time of visit is obtained as 2022-05-30 23:04:40 for YouTube in Firefox from VisitedURLs. This information is matched to the epoch time stamp of 1653969882 in IndexedDB storage. The exact conversion of the IndexedDB time stamp to time string is 2022-05-30 2022 23:04:42.

It appears that YouTube and Instagram use IndexedDB more than LocalStorage. Google uses LocalStorage more than IndexedDB. Reddit uses both technologies. The number of times an item can perform verification through a combination of IndexedDB and LocalStorage is given in Table 40 for all three web browsers.

Table 40

Number of Browser Verification Items Identified in the Experiments

Verification Item	Microsoft Edge	Mozilla Firefox	Google Chrome
OS	1	2	2
OS Version	1	2	1
Browser	3	4	4
Browser Version	3	4	4
Identifier Email	2	2	2
Identifier Name	1	1	1
Identifier Username	1	2	3
Identifier IP	1	1	1
Identifier Location	1	1	1
Visited Search Result	3	2	3
Search Term	4	4	6
History	10	10	10
Total	31	35	38

This information is necessary for the calculation of the Priority Factor during inconsistency and verification score analysis. Verification and inconsistency scores are calculated with the combination of both IndexedDB and LocalStorage verification items.

VRS and IAS Calculation

In this section, a scenario is given over the experimental results to demonstrate the calculations. Accordingly, it is assumed that the browser information of the suspect could not be detected from the IndexedDB storage due to unexpected file storage issues. Additionally, the suspect is assumed to change the visit timestamp of YouTube from the

history storage file of the browser. Table 41 gives the number of verifications that can be performed on the YouTube activities of the suspect in the Google Chrome browser.

Table 41

Verification Items for YouTube in Google Chrome According to Given Scenario

Verification Item	Is Verification be Expected?	Is Verification Achieved?
OS	Yes	Yes
OS Version	Yes	Yes
Browser	Yes	No
Browser Version	Yes	Yes
Search Term	Yes	Yes
History	Yes	Yes (as mismatch)
Total	6	5

"Is verification expected field" of Table 41 is correct and from the results of the experiments. The browser information of the "is verification achieved" field is changed from yes to no for the demonstration.

The number of verification items for Google Chrome is thirty-eight according to Table 40. YouTube can verify six of these. The priority factor of YouTube is calculated as 6 / 38 = 0.1579. The reliability factor for YouTube is calculated as 5 / 6 = 0.8334 because in the scenario only five out of six items could be used for verification. VRS (The level we can rely on the IAS score for YouTube) is calculated with respect to Equation 3: (0.1579 x 0.8334) x 100 = 13.16. The VRS score for a single domain range from a minimum of zero to maximum of the domain's priority factor in percentage representation. In other words, if verification for all items of YouTube was achieved, its VRS score would be its priority factor multiplied by 100, which is 15.79. When one of

the items cannot provide verification, VRS score for YouTube is calculated as 13.16 out of 15.79. This number, in a sense, gives how much the inconsistency score for YouTube can be trusted. However, VRS is intended for an overall picture including all domains. Therefore, VRS score of a single domain is not very descriptive on its own.

The summation of VRS for all domains yield the total number which determines the reliability of IAS for all the sources. Table 42 gives the VRS calculations for all domains of the experiment.

Table 42

Priority and Reliability Factors of All Domains of the Experiments in Google Chrome

Domain	**Number of Items it can Verify**	**Priority Factor**	**Reliability Factor**	**VRS (PF x RF)**
Bing	8	8/38 = 0.2105	8/8 = 1	21.05
YouTube	6	6/38 = 0.1579	5/6 = 0.8834	13.16
Google	1	1/38 = 0.0263	1/1 = 1	2.63
Yahoo	2	2/38 = 0.0526	2/2 = 1	5.26
Wikipedia	2	2/38 = 0.0526	2/2 = 1	5.26
Reddit	6	6/38 = 0.1579	6/6 = 1	15.79
Office	2	2/38 = 0.0526	2/2 = 1	5.26
Myshopify	2	2/38 = 0.0526	2/2 = 1	5.26
Ebay	1	1/38 = 0.0263	1/1 = 1	2.63
Chase	3	3/38 = 0.0789	3/3 = 1	7.89
Microsoft	1	1/38 = 0.0263	1/1 = 1	2.63
Netflix	1	1/38 = 0.0263	1/1 = 1	2.63
Instagram	2	2/38 = 0.0526	2/2 = 1	5.26
Facebook	1	1/38 = 0.0263	1/1 = 1	2.63

(continued)

Domain	Number of Items it can Verify	Priority Factor	Reliability Factor	VRS (PF x RF)
Total	38	38/38 = 1		97.34

According to Table 42, the total VRS is 97.34, which determines the level at which the IAS score can be trusted with the missing verification field given in the scenario. Similarly, Equation 4 gives the number of inconsistencies multiplied by the priority factor. If one of these items shows inconsistency with traditional artifacts as given in the scenario, the IAS for YouTube will be calculated as (1 x 0.1579) = 0.1579. Since no inconsistencies are given for the rest of the domains, the total IAS is determined as 0.1579. If all the items were inconsistent for all domains, the IAS score would be 4.4736. Therefore, inconsistency score in a percentage representation is 3.4624%. In other words, there is a 3.4624% inconsistency between the traditional browsing artifacts and IndexedDB storage artifacts.

When put in perspective, in a digital investigation, with one inconsistent artifact of YouTube between traditional browsing artifacts and IndexedDB storage artifacts. In case of one artifact not being obtained properly, the inconsistency is at 3.4624% with a trust level of 97.34%.

Term Frequency and Inverse Time Frequency

A dictionary of the words appearing in the IndexedDB storage is created for utilization in term frequency and inverse time frequency calculations. Table 43 displays the top five keywords for each domain in the Google Chrome browser.

Table 43

Top Five Term Frequency and Inverse Term Frequency Scores of Major Web Sites with IndexedDB storage in Google Chrome Browser

Domain	Word	df	idf
YouTube	eventtime	4131	0.6931
YouTube	eventtimemsny	4130	2.3025
YouTube	eventtype	1823	0.6931
YouTube	type	559	0.5108
YouTube	counter	420	1.6094
Bing	indices	2653	2.3025
Bing	keywords	177	1.6094
Bing	identifiers	105	2.3025
Bing	engine	72	1.6094
Bing	search	72	1.2039
Yahoo	convertnotificationurl	5	2.3025
Yahoo	offlinebeacon	5	2.3025
Yahoo	precache	5	2.3025
Yahoo	data	3	1.2039
Yahoo	meta	3	2.3025
Reddit	push_notification	9	1.6094
Reddit	browsername	6	2.3025
Reddit	notification	6	1.2039
Reddit	chrome	3	1.6094
Reddit	default	3	2.3025
Office	itemb	85	2.3025
Office	0y4g	2	2.3025
Office	1oq4	2	2.3025

(continued)

Domain	Word	df	idf
eBay	createtime	1	1.6094
eBay	x0v0	1	2.3025
eBay	publickey	1	2.3025
eBay	privatekey	1	1.6094
eBay	type	1	0.6931
Chase	value	3	0.9162
Netflix	1454650993858250	1	2.3025
Instagram	fullname	12	2.3025
Instagram	text_color	11	2.3025
Instagram	overallcategoryname	10	2.3025
Instagram	icon_type	9	1.6094
Instagram	240163565	8	2.3025

It is worth mentioning that most information in IndexedDB storage has meshed with binary blobs. Therefore, the words can be picked out while being checked against a given keyword during a search. However, in a word extraction absent of a base keyword for comparison, it is difficult to pick them out. In order to prevent the binary data from being attached to the words and making them appear more unique throughout the document, the words extending twenty-five letters were extracted from the list. Therefore, some domains that have meaningful IndexedDB storage have less than five words listed for them. Additionally, as the verification was performed through both IndexedDB and LocalStorage, verification was achieved through only LocalStorage for some of the domains. These domains either have no data for IndexedDB or they have very insignificant binary data. These domains were not listed in Table 43.

It can be seen from the listed words of the domains in Table 43 that domains such as YouTube and Bing are focused on storing the actions of users whereas domains such as Yahoo and eBay are focused on the storage of web elements. Instagram appears to contain information for both user actions and web elements. Therefore, it can be seen that the focus of data in the IndexedDB storage of a domain can be hinted at by term frequency analysis.

CHAPTER V

Conclusions

In this dissertation, the structure, utilization, and characteristics of IndexedDB storage were studied on three levels. First, the prevalence of its usage on major websites was investigated with an analysis of significance for forensic investigations. Subsequently, case studies of different domains, web browsers, and applications were put under scrutinization. The storage of the targeted domains and technologies were populated with four single case pretest posttest quasi experiments, The methods of extraction, processing, time-frame analysis, and presentation were laid out in a structural format. Finally, the efficiency of IndexedDB storage and its predecessor LocalStorage was evaluated for verification of traditional browsing artifacts. In total nine hypotheses were created for examination. The obtained artifacts and their extraction techniques did not show any difference in cases where virtual machines and network proxies were utilized. Therefore, the results of the dissertation are valid on a broad perspective.

The experiments resulted in the extraction of a large number of artifacts that indicated substantial value for digital forensic investigations. Additionally, the substantial amount of time related information made it possible to construct an efficient time-frame analysis with only minor deficient records. The potential for further investigation techniques was indicated in terms of location, social network, and emoji analysis. Therefore, the hypotheses created in the second level of the dissertation are validated. In fact, the IndexedDB storage of Microsoft Teams, Instagram, and WhatsApp Web contain artifacts that can benefit the digital forensic investigations to a great degree.

In the third level, it was demonstrated that persistent storage technologies including IndexedDB can be utilized to verify the traditional browsing artifacts. Furthermore, inconsistency of the artifacts and the level of verification accuracy were calculated over a given scenario. Therefore, the hypotheses created in the third level of the dissertation are also validated.

The growing usage of social media, online gaming, and the rest of the online platforms on top of the limited capabilities to transfer large amounts of data over the internet makes the usage of IndexedDB and its possible successors potentially critical for the future. It appears that major domains will store more information in web browsers and desktop applications, potentially creating even a greater source of digital forensic investigations. It is crucial for law enforcement and forensic researchers to keep up with the development of given client-side technologies for an effective justice system.

REFERENCES

Actoriano, B., & Riadi, I. (2018). *Forensic Investigation on Whatsapp Web Using Framework Integrated Digital Forensic Investigation Framework Version 2*. *7*, 410–419.

Al-Shaikh, A., & Sleit, A. (2017). Evaluating IndexedDB performance on web browsers. *2017 8th International Conference on Information Technology (ICIT)*, 488–494. https://doi.org/10.1109/ICITECH.2017.8080047

An, J., Li, T., Teng, Y., & Zhang, P. (2018). Factors Influencing Emoji Usage in Smartphone Mediated Communications. In *Transforming Digital Worlds* (Vol. 10766, pp. 423–428). Springer Cham. https://doi.org/10.1007/978-3-319-78105-1_46

Anglano, C. (2014). Forensic analysis of WhatsApp Messenger on Android smartphones. *Digital Investigation*, *11*(3), 201–213. https://doi.org/https://doi.org/10.1016/j.diin.2014.04.003

Antwi-Boasiako, A., & Venter, H. (2017). *Advances in Digital Forensics XIII* (G. Peterson & S. Shenoi, Eds.; Vol. 511). Springer International Publishing. https://doi.org/10.1007/978-3-319-67208-3

Arshad, H., Jantan, A. bin, & Abiodun, O. I. (2018). Digital forensics: Review of issues in scientific validation of digital evidence. *Journal of Information Processing Systems*, *14*(2). https://doi.org/10.3745/JIPS.03.0095

Autopsy - Digital Forensics. (n.d.). Retrieved July 12, 2022, from https://www.autopsy.com/

Awesome Photographers • Instagram photos and videos. (n.d.). Retrieved July 12, 2022, from https://www.instagram.com/awesome.photographers/

Basques, K. (2019). View and edit local storage. *Chrome Developers*. https://developer.chrome.com/docs/devtools/storage/localstorage/

Bhattacharya, N., & Gwizdka, J. (2021). YASBIL: Yet Another Search Behaviour (and) Interaction Logger. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2585–2589. https://doi.org/10.1145/3404835.3462800

Bortz, A., Barth, A., & Czeskis, A. (2012). Origin cookies: Session integrity for web applications. *ACM Transactions on Internet Technology (TOIT)*, *2*.

Braga, A. A., & Pierce, G. L. (2011). Reconsidering the ballistic imaging of crime bullets in gun law enforcement operations. *Forensic Science Policy & Management: An International Journal*, *2*(3), 105–117. https://doi.org/10.1080/19409044.2011.613444

Browser market share. (n.d.). Net Marketshare. Retrieved July 12, 2022, from https://netmarketshare.com/browser-market-share.aspx

Browser storage limits and eviction criteria. (n.d.). MDN Web Docs. Retrieved July 12, 2022, from https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Browser_storage_limits_and_eviction_criteria

Brunty, J., & Helenek, K. (2014). Social media investigation for law enforcement. In *Social Media Investigation for Law Enforcement*. https://doi.org/10.4324/9781315721323

Casey, E. (2019). The chequered past and risky future of digital forensics. *Australian Journal of Forensic Sciences*, *51*(6), 649–664. https://doi.org/10.1080/00450618.2018.1554090

Chang, M. S., & Yen, C. P. (2019). Forensic analysis of social networks based on Instagram. *International Journal of Network Security*, *21*(5), 850–860. https://doi.org/10.6633/IJNS.201909_21(5).18

Chew, A. M. K., & Gunasekeran, D. V. (2021). Social media big data: The good, the bad, and the ugly (un)truths. *Frontiers in Big Data*, *4*, 6. https://doi.org/10.3389/fdata.2021.623794

Chris, A. (2021). What are search terms? (With examples). *Reliablesoft.* https://www.reliablesoft.net/what-are-search-terms/

Cook, T. D., & Campbell, D. T. (1979). The design and conduct of true experiments and quasi-experiments in field settings. In R. M. Steers & R. T. Mowday (Eds.), *Reproduced in part in Research in Organizations: Issues and Controversies*. Goodyear Publishing Company. https://www.scholars.northwestern.edu/en/publications/the-design-and-conduct-of-true-experiments-and-quasi-experiments--3

DB Browser for SQLite. (n.d.). Sqlitebrowser. Retrieved July 12, 2022, from https://sqlitebrowser.org/

Dixon, M. W., McGill, T. J., & Karlsson, J. M. (1997). Using a network simulation package to teach the client-server model. *Proceedings of the 2nd Conference on Integrating Technology into Computer Science Education - ITiCSE '97*, 71–73. https://doi.org/10.1145/268819.268842

Douglas, Z. (2018). *Digital image recompression analysis of Instagram* [Thesis]. University of Colorado—Denver.

Ferragina, P., & Grossi, R. (1999). The string B-tree: A new data structure for string search in external memory and its applications. *Journal of the ACM*, *46*(2), 236–280. https://doi.org/10.1145/301970.301973

FTK Imager Version 4.5. (2020). AccessData. https://accessdata.com/product-download/ftk-imager-version-4-5

Funny picture lol: ChargeYourPhone [Online forum post]. (2022). Reddit. https://www.reddit.com/r/ChargeYourPhone/comments/u4o8c2/funny_picture_lol/

Gauthier, N. H., & Husain, M. I. (2021). Dynamic security analysis of Zoom, Google Meet and Microsoft Teams. *Communications in Computer and Information Science*, *1383 CCIS*. https://doi.org/10.1007/978-3-030-72725-3_1

Ghafarian, A., & Keskin, D. (2020). Windows 10 Hibernation file forensics. In S. and B. R. Arai Kohei and Kapoor (Eds.), *Intelligent Computing* (pp. 431–445). Springer International Publishing.

Graeme, H. (2020). Part 1:- quality assurance mechanisms for digital forensic investigations: Introducing the Verification of Digital Evidence (VODE) framework. *Forensic Science International: Reports*, *2*, 100038. https://doi.org/10.1016/j.fsir.2019.100038

Hang on! That's not SQLite! Chrome, Electron and LevelDB. (2020). *Ccl Solutions*. https://www.cclsolutionsgroup.com/post/hang-on-thats-not-sqlite-chrome-electron-and-leveldb

Havrlant, L., & Kreinovich, V. (2017). A simple probabilistic explanation of term frequency-inverse document frequency (tf-idf) heuristic (and variations motivated by this explanation). *International Journal of General Systems*, *46*(1). https://doi.org/10.1080/03081079.2017.1291635

Indexed Database API 3.0. (n.d.). *W3C*. Retrieved July 12, 2022, from https://www.w3.org/TR/IndexedDB-3/

IndexedDB API. (n.d.). *MDN Web Docs*. Retrieved July 12, 2022, from https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API

Jadoon, A. K., Iqbal, W., Amjad, M. F., Afzal, H., & Bangash, Y. A. (2019). Forensic analysis of Tor Browser: A case study for privacy and anonymity on the web. *Forensic Science International*, *299*, 59–73. https://doi.org/https://doi.org/10.1016/j.forsciint.2019.03.030

John, A. S. (2020). Google Meet, Microsoft Teams, Webex Privacy Issues - Consumer reports. *Consumer Reports*. https://www.consumerreports.org/video-conferencing-services/videoconferencing-privacy-issues-google-microsoft-webex-a7383469308/

Jones, K. (1972). A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, *28*(1). https://doi.org/10.1108/eb026526

Jones, P. (2019). NPP_HexEdit. *GitHub*. https://github.com/chcg/NPP_HexEdit/releases

Kao, D.-Y. (2016). Cybercrime investigation countermeasure using created-accessed-modified model in cloud computing environments. *The Journal of Supercomputing*, *72*(1), 141–160. https://doi.org/10.1007/s11227-015-1516-7

Kao, D.-Y., Chao, Y.-T., Tsai, F., & Huang, C.-Y. (2018). Digital evidence analytics applied in cybercrime investigations. *2018 IEEE Conference on Application, Information and Network Security (AINS)*, 111–116. https://doi.org/10.1109/AINS.2018.8631403

Karegowda, A., Manjunath, A., & M.A, J. (2010). Comparative study of attribute selection using gain ratio and correlation-based feature selection. *Int J Inf Technol Knowl Manage, 2*.

Karpisek, F., Baggili, I., & Breitinger, F. (2015). WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages. *Digital Investigation, 15*, 110–118. https://doi.org/10.1016/j.diin.2015.09.002

The SEM Academy. (2019, April 17). *Keywords vs Search Terms - What is the difference?* [Video]. YouTube. https://www.youtube.com/watch?v=rv-5OAYIeWg

Kim, Y.-H., & Kwon, T.-K. (2021). On artifact analysis for user behaviors in collaboration tools - Using differential forensics for distinct operating environments. *Journal of the Korea Institute of Information Security & Cryptology*, *31*(3), 353–363. https://doi.org/10.13089/JKIISC.2021.31.3.353

Kimak, S. (2016). *An Investigation into possible attacks on HTML5 IndexedDB and their prevention* [Dissertation]. Northumbria University.

Kimak, S., & Ellman, J. (2015). The role of HTML5 IndexedDB, the past, present and future. *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*, 379–383. https://doi.org/10.1109/ICITST.2015.7412126

Kimak, S., Ellman, J., & Laing, C. (2014, October 15-16). *Some potential issues with the security of HTML5 IndexedDB* [Paper presentation]. 9th IET International Conference on System Safety and Cyber Security, Manchester, United Kingdom. https://doi.org/10.1049/cp.2014.0971

Lin, J. (2015). Building a self-contained search engine in the browser. *Proceedings of the 2015 International Conference on The Theory of Information Retrieval*, 309–312. https://doi.org/10.1145/2808194.2809478

Liu, X., Yu, X., Ma, X., & Kuang, H. (2020). A method to improve the fresh data query efficiency of blockchain. *2020 12th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 823–827. https://doi.org/10.1109/ICMTMA50254.2020.00179

Luo, H., Jiang, H., Zhichao Yan, & Yaodong Yang. (2016). Fast transaction logging for smartphones. *2016 32nd Symposium on Mass Storage Systems and Technologies (MSST)*, 1–5. https://doi.org/10.1109/MSST.2016.7897094

Mahajan, A., Dahiya, M. S., & Sanghvi, H. P. (2013). *Forensic analysis of instant messenger applications on Android devices*. https://doi.org/10.5120/11602-6965

Mahaju, S., & Atkison, T. (2017). Evaluation of Firefox Browser forensics tools. *Proceedings of the SouthEast Conference*, 5–12. https://doi.org/10.1145/3077286.3077310

Mann, M. (2018). The Max Schrems Litigation: A Personal Account. In E. Fahey (Ed.), *Institutionalisation beyond the Nation State: Transatlantic relations: Data, privacy and trade law* (pp. 75–89). Springer International Publishing. https://doi.org/10.1007/978-3-319-50221-2_5

Marengo, D., Giannotta, F., & Settanni, M. (2017). Assessing personality using emoji: An exploratory study. *Personality and Individual Differences*, *112*, 74–78. https://doi.org/10.1016/j.paid.2017.02.037

Marrington, A., Baggili, I., Ismail, T. al, & Kaf, A. al. (2012). Portable web browser forensics: A forensic examination of the privacy benefits of portable web browsers. *2012 International Conference on Computer Systems and Industrial Informatics*, 1–6. https://doi.org/10.1109/ICCSII.2012.6454516

Mendoza, A., Kumar, A., Midcap, D., Cho, H., & Varol, C. (2015). BrowStEx: A tool to aggregate browser storage artifacts for forensic analysis. *Digital Investigation*, *14*, 63–75. https://doi.org/https://doi.org/10.1016/j.diin.2015.08.001

Millett, L. I., Friedman, B., & Felten, E. (2001). Cookies and Web browser design. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '01*, 46–52. https://doi.org/10.1145/365024.365034

Mushcab, R. al, & Gladyshev, P. (2015). Forensic analysis of Instagram and path on an iPhone 5s mobile device. *2015 IEEE Symposium on Computers and Communication (ISCC)*, 146–151. https://doi.org/10.1109/ISCC.2015.7405508

Nalawade, A., Bharne, S., & Mane, V. (2016). Forensic analysis and evidence collection for web browser activity. *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, 518–522. https://doi.org/10.1109/ICACDOT.2016.7877639

National Institute of Justice. (2001). Electronic crime scene investigation guide: A guide for first responders. *United States Department of Justice Office of Justice*.

Nicolescu, Christina. (2019, July 14). *What are Search Terms with Examples* [Video]. YouTube. https://www.youtube.com/watch?v=3qLZpqDZlY8

Nicoletti, M., & Bernaschi, M. (2021). *Forensics for Microsoft Teams*. https://doi.org/10.48550/arxiv.2109.06097

Oh, J., Lee, S., & Lee, S. (2011). Advanced evidence collection and analysis of web browser activity. *Digital Investigation*, *8*, S62–S70. https://doi.org/https://doi.org/10.1016/j.diin.2011.05.008

Paligu, F., Kumar, A., Cho, H., & Varol, C. (2019). BrowStExPlus: A tool to aggregate IndexedDB artifacts for forensic analysis. *Journal of Forensic Sciences*, *64*(5). https://doi.org/10.1111/1556-4029.14043

Paligu, F., & Varol, C. (2020). Browser forensic investigations of WhatsApp web utilizing IndexedDB persistent storage. *Future Internet*, *12*(11), 184. https://doi.org/10.3390/fi12110184

Paligu, F., & Varol, C. (2022). Microsoft Teams desktop application forensic investigations utilizing IndexedDB storage. *Journal of Forensic Sciences*. https://doi.org/10.1111/1556-4029.15014

Pambayun, S., & Riadi, I. (2020). Investigation on Instagram Android-based using digital forensics research workshop framework. *International Journal of Computer Applications*, *175*(35), 15–21. https://doi.org/10.5120/ijca2020920904

Parents make a funny video: MadeMeSmile [Online forum post]. (2022, May 18). Reddit. https://www.reddit.com/r/MadeMeSmile/comments/usqvsz/parents_make_a_funny_video/

Patil, D. N., & Meshram, B. B. (2019). Web browser analysis for detecting user activities. In S. and H. I. K. and S. M. N. Sa Pankaj Kumar and Bakshi (Eds.), *Recent findings in intelligent computing techniques* (pp. 279–291). Springer Singapore.

PHP: Hypertext Preprocessor. (n.d.). Retrieved July 12, 2022, from https://www.php.net/

Pollitt, M. (2010). A History of Digital Forensics. In *IFIP Advances in Information and Communication Technology* (Vol. 337, pp. 3–15). Springer. https://doi.org/10.1007/978-3-642-15506-2_1

Pyrooz, D. C., & Moule, Jr. , R. K. (2019). Gangs and social media. In *Oxford Research Encyclopedia of Criminology and Criminal Justice*. Oxford University Press. https://doi.org/10.1093/acrefore/9780190264079.013.439

Rathod, D. (2017). Web browser forensics: Google Chrome. *International Journal of Advanced Research in Computer Science*, *8*(7), 896–899. https://doi.org/10.26483/ijarcs.v8i7.4433

Same origin policy - Web security. (n.d.). *W3C*. Retrieved July 12, 2022, from https://www.w3.org/Security/wiki/Same_Origin_Policy

Schatz, B., Mohay, G., & Clark, A. (2006). A correlation method for establishing provenance of timestamps in digital evidence. *Digital Investigation*, *3*, 98–107. https://doi.org/10.1016/j.diin.2006.06.009

Seo, S., Kim, Y., & Lee, C. (2018). Instagram users behavior analysis in a digital forensic perspective. *Journal of the Korea Institute of Information Security & Cryptology*, *28*(2), 407–416.

Sgaras, C., Kechadi, M.-T., & Le-Khac, N.-A. (2015). Forensics acquisition and analysis of Instant Messaging and VoIP applications. In F. Garain Utpal and Shafait (Ed.), *Computational Forensics* (pp. 188–199). Springer International Publishing.

Shortall, A., & Azhar, M. A. B. H. (2015). Forensic acquisitions of WhatsApp Data on popular mobile platforms. *2015 Sixth International Conference on Emerging Security Technologies (EST)*, 13–17. https://doi.org/10.1109/EST.2015.16

Singh, B., & Singh, U. (2017). A forensic insight into Windows 10 Cortana search. *Computers & Security*, *66*, 142–154. https://doi.org/https://doi.org/10.1016/j.cose.2017.01.007

Singh, R., & Awasthi, S. (2020). Updated comparative analysis on video conferencing platforms- Zoom, Google Meet, Microsoft Teams, WebEx Teams and GoToMeetings. *Easy Chair: The World for Scientist*.

TechSmith Capture. (n.d.). *TechSmith*. Retrieved July 12, 2022, from https://www.techsmith.com/jing-tool.html

Thakur, N. (2013). *Forensic analysis of WhatsApp on Android smartphones* [Master's thesis, University of New Orleans]. University of New Orleans Theses and Dissertations. https://scholarworks.uno.edu/td/1706

Top sites in United States. (n.d.). *Alexa.* Retrieved January 3, 2022, from https://www.alexa.com/topsites/countries/US

Umar, R., Riadi, I., & Maulana, G. (2017). A comparative study of forensic tools for WhatsApp analysis using NIST measurements. *International Journal of Advanced Computer Science and Applications*, *8*(12), 69–75. https://doi.org/10.14569/IJACSA.2017.081210

Using IndexedDB. (n.d.). *MDN Web Docs*. Retrieved July 12, 2022, from https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB

Walker, J. D., & Chapra, S. C. (2014). A client-side web application for interactive environmental simulation modeling. *Environmental Modelling & Software*, *55*, 49–60. https://doi.org/10.1016/j.envsoft.2014.01.023

Walnycky, D., Baggili, I., Marrington, A., Moore, J., & Breitinger, F. (2015). Network and device forensic analysis of Android social-messaging applications. *Digital Investigation*, *14*, S77–S84. https://doi.org/https://doi.org/10.1016/j.diin.2015.05.009

What is Notepad++. (n.d.). *Notepad++*. Retrieved July 12, 2022, from https://notepad-plus-plus.org/

Joso, Timotius. (2012, October 12). *Wildlife Windows 7 Sample Video* [Video]. YouTube. https://www.youtube.com/watch?v=a3ICNMQW7Ok

Woods, D., Snee, T., & Pekowsky, K. (1999). *Developer's guide to the Java Web Server: Building effective and scalable server-side applications with Cdrom* (1st ed.). Addison-Wesley Longman Publishing Co., Inc.

Wyse, L., & Subramanian, S. (2013). The viability of the web browser as a computer music platform. *Computer Music Journal*, *37*(4), 10–23. https://doi.org/10.1162/COMJ_a_00213

Youn, T.-Y., Chang, K.-Y., Rhee, K.-H., & Shin, S. U. (2018). Efficient client-side deduplication of encrypted data with public auditing in cloud storage. *IEEE Access*, *6*, 26578–26587. https://doi.org/10.1109/ACCESS.2018.2836328

Zarei, K., Farahbakhsh, R., & Crespi, N. (2019). Typification of impersonated accounts on Instagram. *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC)*, 1–6. https://doi.org/10.1109/IPCCC47392.2019.8958763

APPENDIX

Record Patterns Utilized in The Experiments

Description	Identifier Patterns	Disqualifying Patterns
Record Start Patterns	"t.a.c.v.2", "s.p.a.c.e.s", "p.r.i.v._.a.g.g.r.e.g.a.t.e", "p.r.i.v._.p.r.e.f.s", "a.c.t.i.o.n.M.e.t.a.d.a.t.a", "p.r.i.v._.d.a.t.a.b.a.s.e", "n.o.t.i.f.i.c.a.t.i.o.n.s", "C.o.n.v.e.r.s.a.t.i.o.n.L.i.s.t.S.y.n.c.S.t.a.t.e", "s.u.b.s.t.r.a.t.e._.d.a.t.a._.u.p.d.a.t.e.d", "l.a.s.t._.s.e.l.e.c.t.e.d._.v.i.e.w", "M.o.s.t.R.e.c.e.n.t.C.o.n.v.e.r.s.a.t.i.o.n"	
Account Record Identifying Patterns	"priv_prefs", "orgid", "displayName", "orgid", "givenName", "surname", "firstname_lowercase"	
Event/Call Record Identifying Patterns	"startTime", "endTime", "callDirection", "twoParty"	
Meeting Creation Identifying Patterns	"CalendarEvent", "eventType", "organizerName"	

Description	Identifier Patterns	Disqualifying Patterns
Instant Message Identifying Patterns	"content", "div", "composetime", "previewHeight", "HyperLink"	“hasCustomPostToChannelMessage”
	targetParticipantId	
Scheduled Event/Call Identifying Patterns	"meeting", "application", "creator", "partlist	
Whiteboard Identifying Patterns	"whiteboard","directive","extension",”dateAdded"	
Voice Mail Identifying Patterns	"activityType", "call", "voicemail"	
Team Creation Identifying Patterns	"teamStatus", "teamAlias", "createdat"	

VITA

Furkan Paligu is a lecturer of computer science in North American University and a PhD candidate of digital and cyber forensics in Sam Houston State University. He has a bachelor's degree in Computer Engineering from Marmara University and master's degree in Cybersecurity Engineering from Istanbul Sehir University. He has taken part in various information technology and cybersecurity projects as a software developer and software security engineer in Turkish Scientific and Technological Research Association (Equivalent of National Science Foundation in USA) from 2012 to 2018. An extensive part of his work has been focused on large scale projects in complex domains including data leak prevention systems, voice translation engines, customer application and management systems, static code analyzers, malware scanning systems operating on cloud architecture, electronic signature systems, and electronic cash registers. Starting January 2020, he has been teaching in North American University Computer Science Department.