TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Felix Waschke 223591IVCM

FORENSIC ANALYSIS OF THE SLACK WEB APPLICATION

Master's Thesis

Supervisor: Pavel Tšikul
M.Sc.

Tallinn 2024

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Felix Waschke 223591IVCM

SLACKI VEEBIRAKENDUSE KOHTUEKSPERTIISI ANALÜÜS

Magistritöö

Juhendaja: Pavel Tšikul
M.Sc.

Tallinn 2024

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Felix Waschke

Date: 12.05.2024

Abstract

Collaboration tools and platforms are an integral part of most modern organisations. Slack, as one of the most popular collaboration platforms, helps users to interact with each other. The tight integration inside an organisation makes Slack a valuable information source during internal investigations. When Slack is used without authorisation, obtaining the data directly becomes difficult. This thesis investigates the digital artefacts left by the Slack web application on the hard drive, employing a pre-test/post-test quasi-experimental design across eight distinct experimental setups. The results demonstrated that crucial parts of Slack usage, such as messages, user information, and files, can be recovered. Notably, the main source of information is the IndexedDB, a client-side storage for web applications. With the focus on the IndexedDB, further research was conducted in the IndexedDB. In the research, the potential to recover data after the deletion of the IndexedDB or all browser data is demonstrated. The results of this work allow investigators to potentially gather more insight in the behaviour of a user on web applications by inspecting the hard drive.

The thesis is written in English and is 90 pages long, including 7 chapters, 26 figures and 11 tables.

Annotatsioon

Slacki veebirakenduse kohtuekspertiisi analüüs

Koostöövahendid ja -platvormid on enamiku kaasaegsete organisatsioonide lahutamatu osa. Slack, olles üks populaarsemaid koostööplatvorme, aitab kasutajatel omavahel suhelda. Tihe integreerimine organisatsiooni sees muudab Slacki väärtuslikuks informatsioonial-likaks sisemiste uurimuste ajal. Kui Slacki kasutatakse ilma loata, muutub andmete otse hankimine keeruliseks. See väitekiri uurib digitaalseid jälgi, mida Slacki veebirakendus jätab kasutajate kõvaketastele, kasutades pre-test/post-test kvasieksperimentaalset kujundust kaheksas erinevas eksperimentaalses seadistuses. Tulemused näitasid, et Slacki kasutamise olulisi osi, nagu sõnumid, kasutaja teave ja failid, on võimalik taastada. Eelkõige on peamine informatsiooniallikas IndexedDB, kliendipoolne salvestusruum veebirakendustele. Fokuseerides IndexedDB-le, viidi läbi edasisi uuringuid IndexedDB-s, mis näitasid andmete taastamise potentsiaali pärast IndexedDB või kogu brauseri andmete kustutamist. Selle töö tulemused võimaldavad uurijatel koguda potentsiaalselt rohkem teavet veebirakenduste kasutaja käitumise kohta, uurides kõvaketast.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 90 leheküljel, 7 peatükki, 26 joonist, 11 tabelit.

List of Abbreviations and Terms

API	Application Programming Interface
BEC	Belkasoft Evidence Center
BLOB	Binary Large OBject
JSON	JavaScript Object Notation
MFA	Multifactor Authentication
OS	Operation System
PC	Personal Computer
PDF	Portable Document Format
PNG	Portable Network Graphics
RAM	Random Access Memory
SQL	Structured Query Language
VM	Virtual Machine
WHATWG	Web Hypertext Application Technology Working Group

Table of Contents

List of Figures

List of Tables

1. Introduction

With changing demands in the work environment, remote work has become an essential part of many organisations. This trend has increased the usage of collaboration tools in the business environment. Collaboration tools allow the user to exchange information with others in various ways, such as text, voice, and video. According to [1], Slack and its biggest competitor, Microsoft Teams, make up the largest market share of collaboration tools. In addition to their abilities as communication tools, collaboration tools act as repositories of vast amounts of information in many organisations.

Due to these capabilities, investigating and analysing collaboration tools have become increasingly critical. Internal or criminal investigations often need to gather and analyse data from these tools to understand the actions and interactions of individuals or groups. The data available on such platforms is ideal for uncovering evidence and obtaining insight into potential wrongdoings, conflicts, or patterns of collaboration. Furthermore, cloud-hosted collaboration tools can be an especially attractive target for attacks and unauthorised access. This is primarily because of the location outside of the organisation's network and the importance of the tool inside the organisation. The 2022 attack on Uber demonstrates the possibility of attacking an organisation through such a collaboration tool, as the infiltrator accessed the company's Slack workspace. The attacker was able to buy the account details on the dark web and trick the affected employee into accepting an MFA and accessing the internal network. With access to the internal network, the attacker was able to exfiltrate sensitive data and disrupt the organisation's operations. After successfully exfiltrating the data, the attacker announced the data breach on Slack[2]. However, the attack did not focus on Slack. The attacker only used it to announce them. Yet, the attack shows that collaboration tools can be used to advance an attack.

This attack underscores the need for a forensic analysis of these tools to understand the development of certain attacks. Because these tools are often trusted external entities, they also allow an attacker to exfiltrate potentially undetected data. Investigations of such incidents can be difficult. The artefacts are stored within the infrastructure of the collaboration tool provider, which may limit access to the data or comprehensive forensic capabilities. This situation can be extremely challenging for an investigator who does not have administrative access to the tool.

Specifically, "Shadow IT" can be a problem in this context. Shadow IT describes the use

of software and cloud services inside an organisation without explicit approval. A survey conducted by CISCO revealed that 83% of employees in the IT sector used some form of unauthorised application at work[3]. In the context of the rapid change to remote work due to COVID-19, the use of shadow IT has become even more prevalent [4]. Due to the rapid transition to remote work, employees have sought and adopted collaboration tools that best suit their needs. This could often happen without the knowledge or approval of the IT departments. Additionally, many of those quickly set-up environments may still be in use. They are convenient and efficient tools for employees, even if a replacement is later provided by the organisation. Moreover, especially in the case of misconduct, participants are likely to avoid official company tools to prevent monitoring. Both reasons can cause problems during investigations of security incidents or internal investigations. Therefore, developing capabilities to investigate collaboration tools and their artefacts is crucial, especially when the investigator does not have administrative access to the tool or its infrastructure. This thesis focuses on the analysis of artefacts from the Slack web application, one of the most widely used collaboration tools. The decision to focus on Slack was mainly based on the non-existing research in this area and its widespread usage in various organisations.

1.1 Research objectives

This thesis seeks to explore the digital artefacts left after the use of the Slack web application. This leads to the following hypothesis.

H0: It is possible to recover useful artefacts from the web browser after using the Slack web application.

To validate the hypothesis, three research questions were formulated. Answering these research questions will contribute to our understanding of collaboration tools and their artefacts. The research questions are partly interdependent; therefore, they will be investigated sequentially:

RQ1: What artefacts remain on the device after using the Slack web application?

This question focuses on the identification and description of artefacts left on a device after using the Slack web application. The goal is to create a comprehensive list of artefacts and establish the usefulness of each artefact in an investigation.

RQ2: Are there differences in artefacts between different browsers and operating systems?

Recognising that users could access the Slack web application through various browsers and operating systems, this research seeks to compare artefacts generated across these platforms. Differences in artefact types, storage mechanisms, and persistence should be identified to understand how platform-specific attributes influence the artefacts left by the application.

RQ3: Can artefacts be extracted in an automated way?

As a last step in the research, the feasibility of automating the extraction of these artefacts needs to be evaluated. This aspect is crucial for streamlined investigations. Answering the question helps, to develop a tool capable of extracting relevant data and demonstrating a way to approach similar applications.

By addressing these research questions, the thesis aims to contribute to the field of digital forensics, particularly in understanding and managing the artefacts left by web applications.

1.2 Scope

This thesis aims to comprehensively analyse the artefacts left by the Slack web application. The analysis focuses on their identification and differences across various browsers and operating systems, and it explores the potential for automatically extracting these artefacts. The emphasis is on Windows and Linux operating systems and using Google Chrome and Mozilla Firefox browsers for data collection. The thesis focuses on nonvolatile memory artefacts. Therefore, volatile memory, the SWAP partition, and hibernation files are beyond its scope.

Creating a dataset for this study is resource-intensive. Therefore, the analysis is based on a small sample size. However, this thesis, unlike similar studies, incorporates a wide range of diverse test cases. As such, the findings are expected to contribute to advancing related fields.

The research is exclusively conducted on the most recent version (data-version-ts= "1709479051", 24.02.2024) of the Slack web application, as older or newer versions may possess variations in artefact structures and behaviour. Furthermore, the research focuses on the basics of the Slack web application and excludes advanced features or functionalities. In this thesis, functionalities like Canvas (online collaboration) and Huddle (voice communication via Slack), as well as integrations with third-party applications, are not investigated.

1.3 Novelty

The novelty of this research lies in its focus on the Slack web application. This platform has not been studied in terms of digital artefacts. Furthermore, the research also explores the differences between artefacts in different browsers and operating systems to broaden research in the field of web browser forensics. Some research on the discovery of artefacts using web applications has been conducted. However, the differences between Windows and Linux systems have not been explored extensively. Furthermore, the differences between web browsers, such as Google Chrome and Mozilla Firefox, are analysed.

To fill the above-mentioned gap, this thesis aims to research how the stored artefacts are influenced by different scenarios. Unlike previous research, where the focus was on available artefacts on different collaboration tools, this thesis elucidates the artefacts left by the Slack web application. Moreover, their variations across different browsers and operating systems are investigated.

Furthermore, the thesis provides deep insights into multiple scenarios that could occur during an investigation. Here, the aim is to expand the research in the field of forensic analysis of the IndexedDB.

1.4 Thesis structure

This thesis is structured as follows.

Chapter 1 Introduction: Chapter 1 provides an introduction to the research topic, including the problem statement and the novelty of the research.

Chapter 2 Background: Chapter 2 introduces some technical background as a foundation for the research, and the focus is on the field of browser forensics and its key areas.

Chapter 3 Literature Review: Chapter 3 presents the current state of the research and focuses on the forensic analysis of digital collaboration web applications.

Chapter 4 Methodology: Chapter 4 presents the choice of methodology and explains the methodological design of the research.

Chapter 5 Experimental Results: Chapter 5 discusses the different experiments and describes the results.

Chapter 6 Discussion: Chapter 6 evaluates the results of the previous chapter. It identifies limitations, answers the initial research questions, and recommends a process to handle investigations involving the Slack application.

Chapter 7 Conclusion: Chapter 7 summarises the work and presents the outlook of the study.

2. Background

This chapter presents the important terminology and concepts on which the thesis is based. The purpose of this chapter is to provide a background and foundation for the research project.

2.1 (Web) Browser Forensic

(Web) browser forensics is a branch of digital forensics that focuses on the investigation and analysis of web browser artefacts. Therefore, the focus is on analysing the artefacts left by the user's web activities. Depending on the aim of the investigation, the analysis of browser artefacts can provide valuable information in various scenarios, such as corporate espionage[5] or online fraud[6]. Key areas in browser forensics include [7]:

1. Web Browser History
2. Cookies
3. Cache
4. Client-side Storage

Due to the increase in web usage and the reliance on web browsers for online activities, browser forensics has become an essential area of digital investigation [8] [9]. This field requires expertise to understand how different operating systems store browser records and the specific data that can be recovered or analysed from web browsers. Furthermore, the value of the evidence also depends on the websites visited. Depending on the different technologies used on these sites, experience is needed to understand the user's behaviour on the website [10]. Due to the importance of web activities, most holistic forensic tools, such as Autopsy, Belkasoft Evidence Center, and EnCase, have built-in features and plugins. Those features and plugins are specifically designed to analyse web activity. [11].

2.2 Cache

The cache is a temporary storage location on a computer that stores parts of web pages and media files. This storage allows for faster data retrieval, as the computer can access the cached files instead of downloading them again from the Internet. The structure of the cache depends on the browser used [7]. However, by default, all major web browsing applications cache the visited website content on the local disk to improve browser efficiency. Cache

content can be valuable in digital forensic investigations, as it can provide evidence of a user's browsing history and the content viewed on websites [12]. Furthermore, the cache can be crucial to understanding what the user is seeing on the website. While the URL indicates which website the user visited, the cache can reveal the content that was accessed within that website. Because the content of a website can change quickly, analysing the cache can be essential. The reason is mainly that it can provide a snapshot of the website as it appeared at the time of browsing [12].

In his research Horsman described the cache as being frequently used as evidence in cases involving child exploitation material. It is often assumed that the cache contains a record of content viewed by a defendant via their browser. First, this is done because the images stored in the cache can be considered as being in the possession of the defendant. Second, the cache can be used to recreate the visited website offline [12].

Nevertheless, it is important to note that the cached content may not always accurately reflect the entirety of a user's browsing activity. While the cache can indicate, which content was viewed by the user, the user may not have intentionally accessed or viewed the content. The Horsman sought to examine whether it is possible to determine what was seen by the user based on the browser cache. His results revealed that an investigator cannot obtain an accurate representation of what a user saw based on the cached content [12].

Secondary to the content of a website, the cache can also be used to determine the time that a website was visited. According to [13], the cache can provide information about the browser's history. Should a user attempt to cover their tracks by deleting their browsing history, the cache can still retain evidence of the websites visited and the content accessed. This analysis can be more complicated and time-consuming than simply examining the browser history, but it can provide valuable information in forensic investigations. During such an analysis of the cache, distinguishing between visited websites and websites that provide additional information (e.g. advertisements or embedded content) is difficult, as the cache can store data from both types of websites. Furthermore, the timestamp of the visit can vary slightly between the real visit and the record in the cache. In their work, they identified that approximately 10 of the entries in the cache were written more than 30 seconds late in the Chrome cache.

Common tools to analyse the browser cache include Nirsoft Web Browser Tools, using MZCacheView and ChromeCacheView, as well as ChromagnonCache. In the relevant literature, Nirsoft Web Browser Tools has frequently been used to analyse browser caches. In general, MZCacheView and ChromeCacheView access the cache folder from the

browser. The following information can be retrieved: "URL, content type, file size, last accessed time, expiration time, server name, and server response".

2.3 Cookies

Cookies are small text files that store information from a browser locally. They are a vital part of today's web interactions. Cookies are used for various purposes, like tracking, session management, and personalisation. After a website gives a cookie, the browser will resend the cookie on the next visit. This allows for a more individualised experience on the website. In general, there are two common types of cookies, session cookies and persistent cookies. They differ in the duration, the cookie is stored in the browser. Due to the widespread use of HTTPS encryption, cookies are commonly encrypted and cannot be accessed or modified by unauthorised parties[7].

Cookies can be a valuable source of information during an investigation. They can provide insights into a user's browsing habits and activity on specific websites and potentially identify the actions performed on a website. Therefore, the value of cookies depends on the website investigated and the specific information stored in the cookies [13]. The analysis of cookies can be challenging due to the variety of information that they can store and the encryption used to protect them [7].

2.4 Browser History

The browser history is a record of all websites visited by a user. Analysing browser history can provide investigators with valuable information about user browsing activities [7]. This information includes the URLs of the visited websites, the date and time of visits, as well as any search queries made by the user.

During an investigation, the browser history can provide information on user activities. The value of the browser history depends on the specific case investigated. Although it can reveal when and what URL was visited, there are some limitations to consider. Because most websites are dynamic, the content that the user sees can often not be determined on the basis of the browser history alone. Nevertheless, it stores the browser history metadata, including the URL, time, and title of the webpage, which can give investigators more insights into the user [14].

2.5 Persistent Storage

Client-side storage in web browsers is widely used. It allows websites to store data on a user's device, which can be accessed and retrieved whenever the user visits the website again. The choice of storage depends on the intended use. Some of the most prominent technologies are cookies, web storage, session storage, indexed databases, and cache APIs. As early as the 1990s, cookies were being used for storing small amounts of data on a user's device. The need to store a greater amount of data resulted in the introduction of further technologies like web and session storage. Both technologies allow for storage of up to 5 MB. While web storage is heavily used to store data, session storage introduces non-persistent storage.

IndexedDB was based on Web Storage to further enhance the capabilities of client-side storage. IndexedDB is a NoSQL transactional database that stores all data as pairs of key values [15]. With the introduction of the IndexedDB, the storage size was increased to 50 MB.

The use cases of the different technologies vary depending on the type of technology. Furthermore, the way data is stored at the client side differs across those technologies. This variety introduces the need for specialised research in forensic science. The main technologies used are LevelDB for Google Chrome and other browsers using Chromium and SQLite for Mozilla Firefox. When comparing both approaches, the implementation of the database in LevelDB provides better performance, based on various benchmarks[16].

For LevelDB, the database consists of two file types, and the extensions are .ldb and .log. In general, log files store more recent data. The log file is divided into 32 KB blocks, with each block containing multiple records. Every record consists of a header, key, key length, and potentially, a value and value length [15]. As soon as the .log file reaches the size limit, the data is transformed into the .ldb file, and a new .log file is created [17].

Furkan Paligu demonstrated that the use of IndexedDB technology by many large websites provides an interesting data source for a forensic investigator [14]. Due to the widespread use of the technology on the largest websites, IndexedDB storage can be a valuable source of evidence for forensic investigations.

Currently, tools that can effectively extract and analyse data from IndexedDB are lacking. The main implementation available is CCL chromium IndexedDB, an implementation to decode Chrome IndexedDB in Python [18]. A similar option for Mozilla Firefox is available on GitLab [19]. Both tools are still in development, which may affect their output.

Another option, to gain the content of the IndexedDB, is to access the IndexedDB with the developer tools inside the web browser.

2.6 Notifications

Notifications are part of the browser's user interface that provide alerts or updates to the user. Websites can use them to notify users of new messages, events, or updates. In general, the user must enable notifications for specific websites. Therefore, this feature is mainly enabled for websites that a user frequently visits or considers important to receive updates from. The standard for the notification API is maintained by the Web Hypertext Application Technology Working Group (WHATWG), a collaboration of companies, including Apple, Google, Mozilla, and Microsoft [20]. According to the standard, each notification has multiple attributes associated with it. The most important attributes are title, body, tag, timestamp, and origin. The title and body contain the information displayed in the notification. The tag allows one to reverence the notification and alter or delete it. The timestamp contains time in the epoch format, and the origin specifies the website that created the website [20].

Furthermore, notifications are categorised as persistent and non-persistent notifications. These categories affect how they are processed. The main difference between the two is the storage in the list of notifications. Each "user agent must keep a list of notifications, which is a list of zero or more notifications"[20]. Non-persistent notifications should only be stored in this list for a few seconds. Persistent notifications should be stored in the list until they are removed.

In general, a web browser provides two different APIs to handle the notifications from a website: the Web Notification API and the Push API. Both APIs handle slightly different use cases and can be used in combination with each other. The Web Notification API mainly creates and displays notifications on the device after being triggered by a website [20]. The Push API allows websites to trigger notifications even if the web applications or the web browser are currently not active. This is archived by utilising service workers who constantly listen for updates in the background [21].

During a forensic investigation, analysing notifications can provide valuable information about the websites and applications that a user regularly interacts with. Although research on forensic analysis of notifications is still limited, it has the potential to reveal valuable insights about a user's online activities.

Initial inquiries suggest that tools to help the investigator parse notification data from

browsers directly are missing. However, manually parsing previous notifications is relatively easy. Similar to IndexedDB, the technique used to store the data differs across browsers. Google Chrome and similar Chromium-based browsers use the same storing technique, which is also used for IndexedDB data [15]. In contrast, Mozilla Firefox uses a JSON file called notificationstore.json, which contains the notification data.

2.7 Slack

Slack is an online collaboration platform that teams and organisations use to communicate and work together. The application was developed by Slack Technologies and was first launched in 2013. The main concept of the application is that it allows users to exchange messages directly with other people or within groups. Currently, Slack is available on all major operating systems, such as Windows, MacOS, Linux, IOS, and Android and as a Web application [22]. The applications utilise Electron, a cross-platform framework developed by GitHub. The Electron framework uses Chromium as the underlying technology [22].

The main design of the application provides the user with the opportunity to join workspaces. A workspace aims to provide a centralised hub for a team or organisation to collaborate on various projects and tasks. Each workspace consists of different channels that are dedicated to specific topics or discussions. The channels within Slack can be further categorised into public channels, where anyone in the workspace can join and participate, and private channels, which are invitation-only [22].

Every message written to a channel creates a thread in which other users can reply or react with a variety of emojis. Furthermore, with the application, the user can exchange various files in a channel or via a direct message [22].

Further actions in Slack include creating a canvas, somewhat like a blackboard, and Huddles. Huddles allow users to interact in voice chats with others and share the screen, with more options for collaboration [22].

Due to the widespread use of Slack in many organisations or remote working groups, it is an interesting source of information during forensic investigations. By analysing the Slack workspace of users, the investigator can obtain valuable information about communication patterns, collaboration activities, file exchanges, and more. Using Slack for online collaboration has become increasingly popular among teams and organisations.

3. Literature Review

The literature review aims to establish the current research level for forensic analysis in web collaboration tools. While the focus is on Slack, research on similar applications is also introduced.

3.1 Initial Search

For the initial search, a database search was performed with the following keywords:

- Web Browser Forensic
- (Slack *or* Microsoft Teams *or* Cisco Webex *or* instant messaging) Web Application forensic
- (Slack *or* Microsoft Teams *or* Cisco Webex *or* instant messaging) browser forensic

The keywords were chosen to include similar applications to Slack because research specific to Slack was deemed scarce. The initial literature review included only articles related in some way to the forensic analysis of web applications. To ensure accuracy, articles earlier than 2016 were excluded. The grey literature was excluded during this stage of the search. Furthermore, only literature in English was included. The literature review was conducted in IEEE Explorer[2], Elsevier[3], and Google Scholar[4].

This initial search produced the following articles:

- "Digital forensic analysis of discord on google chrome" by Gupta, Varol, and Zhou[23]
- "Browser Forensic Investigations of WhatsApp Web Utilizing IndexedDB Persistent Storage" by Paligu and Varol[24]
- "Forensic investigation of Cisco WebEx desktop client, web, and Android smartphone applications" by Khalid, Iqbal, Kamoun, *et al.*[25]

The listed articles laid the foundation for the next step of the search process.

[2]https://ieeexplore.ieee.org/
[3]https://www.sciencedirect.com/
[4]https://scholar.google.com/

3.2 Snowballing

Based on the as-relevant identified papers from the initial search, the snowballing approach was used to obtain more relevant information.

3.2.1 Forward snowballing

The three identified articles were used as a base for forward snowballing; the results are depicted in Table 1. For the forward search, Google Scholar was used.

Title	**Found citations**	**Relevant citations**
"Digital forensic analysis of discord on google chrome"	5	0
"Browser Forensic Investigations of WhatsApp Web Utilizing IndexedDB Persistent Storage"	12	3
"Forensic investigation of Cisco WebEx desktop client, web, and Android smartphone applications"	0	0

Table 1. Results forward snowballing

The limited number of citations could be due to the recent publication of the articles.

3.2.2 Backwards snowballing

Backward snowballing was applied iteratively to all identified papers to ensure that all relevant sources were found.

3.3 Results

Due to the previous steps, the following number of articles was identified:

Search method	**Found articles**
Initial search	3
Forward snowballing	3
Backwards snowballing	5

Table 2. Results of the literature search

3.4 Analysis

The following provides an overview of the literature based on the identified papers.

3.4.1 Forensic Analysis for web browser

The forensic analysis of different types of web browsers can be a crucial part of an investigation. Changing technologies increasingly pose a challenge for forensic analysis. In 2011, Oh, Lee, and Lee provided an overview of the methods to collect and analyse data from the web browser [26]. In the paper, the analysis focused on history, cookies, caches, and downloads. Furthermore, they stated that a tool for analysing a web browser should possess the following features [26]: support for all common browsers, timeline analysis, search history analysis, URL encoding analysis, user activity analysis, and recovery of deleted information.

Based on the paper "Forensic analysis and evidence collection for web browser activity" by Nalawade, Bharne, and Mane, the author identified additional aspects that need consideration during a forensic analysis of the usage of the Web browser. The main concerns are private browsing and the privacy capabilities of the different browsers. Private browsing can present some challenges during a forensic investigation. During private browsing, it disables browsing history and web cache, which provide key information about user activity in the browser [27]. The impact of privacy modes on modern web browsers was further investigated by Horsman, Findlay, Edwick, *et al.* [28]. They confirmed that most modern browsers increase local privacy when privacy mode is enabled.

Although both [26] and [27] introduced tools for a more extensive investigation, they were mainly used to cover general usage. In Chapter 3.4.4, more specific tools for analysing communication tools are discussed.

3.4.2 Artefacts of the Slack application

Research on the Slack application area is relatively scarce. Davis, McInnes, and Ahmed published an article on artefacts found in the Linux operating system after using Discord and Slack [29]. However, they only considered installed applications and focused on volatile memory. Although the study is not directly applicable to the present research, the results of the paper are helpful for determining the artefacts that could be recovered.
According to Davis, McInnes, and Ahmed [29], the following artefacts can be recovered from the Slack application:

- E-mail of the user
- Emojis
- Images
- Messages
- Status
- Slack Workspace
- Usernames

Other ways to gather the data, for example during an e-discovery process, were presented, by Joe Pochron [30]. These include exporting messages as JSON files, as well as using the Slack Plus compliance function, third-party products, and Slack's discovery API [30]. However, the standard export and the discovery API are the only ones that do not need preparation. Therefore, relying on the methods mentioned above can limit the investigation [30]. Specific to Slack is the software AXIOM Cloud by Magnet Forensic, which allows an investigator to analyse Slack via the cloud data, which also relies on the exports.

Another consideration is the security of the Slack application. In recent years, some security incidents related to the Slack application have occurred. The most notable was the storage of plain text credentials in the Android application in 2021 [31], [32]. Other web security flaws included the ability to exploit a feature of Slack to harass people and potentially craft phishing messages using a legitimate source [33]. Important for security considerations is also the fact that Slack does not support end-to-end encryption [34]. However, according to Slack, the data is encrypted in transit and at rest [35].

Focusing on third-party integration, Chen, Gao, Ceccio, *et al.* explored potential attacks against the Slack application [36]. In their work, they presented three proof-of-concept attacks that allowed them to intercept user messages, launch fake video calls, and automatically merge code in a repository. These attacks are mainly possible due to the wide permissions given to third-party apps [36].

3.4.3 Forensic investigation in similar applications

As shown in Chapter 3.4.2, research on Slack applications is scare. However, similar analyses were performed with applications such as Discord [23], [29], Cisco Webex [25], WhatsApp Web [24], [37], Instagram [38], TikTok [39], Google Meet [40] and Microsoft Teams [41], [42]. Other research works on online collaboration tools or instant messaging tools focused on installed applications.

Pandela and Riadi investigated the data left on the device after uploading a video to TikTok using Google Chrome. The approach was mainly based on forensic tools such as FTK Imager, Browser History Capturer, and Video Cache Viewer. With this approach, they were able to recover the username, profile photo, video thumbnail, and video link in the browser cache [39]. Gupta, Varol, and Zhou used the browser cache and the Google Chrome browser history similarly, and their focus was on the Discord web application [23].
Khalid, Iqbal, Kamoun, *et al.*, in their article "Forensic investigation of Cisco WebEx desktop client, web, and Android smartphone applications," presented a potential use case where the options discussed in Chapter 3.4.2 were not helpful during an investigation. Y, a long-term senior employee of the ABC food manufacturing company, was in a confidentiality agreement with the company about a trade secret. Y was offered a large sum of money by a rival company to reveal the secret. To avoid using his company-licensed Cisco WebEx account, which was logged into his WebEx desktop client application, Y created a personal WebEx account with his personal Gmail address. He then opted to use the Cisco WebEx Web application instead of his laptop PC to avoid leaving any trace of the communication. The meeting was held, and the deal was finalised. A few weeks later, ABC requested a forensic investigation of their employees' workstations. The hard drive of Y's laptop was imaged using the FTK Imager and examined for any pertinent information [25]. The authors showed that the IndexedDB files from the Google Chrome browser could be used to obtain data on the meeting.
Other authors, such as Paligu and Varol or Bowling, demonstrated the usefulness of analysing the IndexedDB data file for web applications. In their articles, both were able to recover messages and profile information [38], [41]. Gupta, Varol, and Zhou used a combination of different sources, such as browser history, cache data, form values, downloaded files, local storage, and IndexedDB in the case of Discord [23]. In most of the articles analysed, the most relevant data was stored in the browser cache and the IndexedDB, including messages, contacts, status information, and profile settings. In [40] IndexedDB was not considered, mainly because the research mainly utilised Autopsy to gather browser artefacts, without using potential plugins. Furkan Paligu argued in his work for the relevance of IndexedDB in forensic science. Based on his previous work [24], [38], [42], [43], he discussed the widespread use of the technology and large amounts of relevant data stored by many web applications in IndexedDB.

Different studies [23], [24], [38], [42], [43] have demonstrated that the same basic information could be extracted from the application. The above authors were able to obtain user information. Other artefacts, such as messages, were also available, as well as more application-related information.

None of the previously mentioned papers explains the artefacts that are recoverable after

the usage of the application in great detail. Instead, all works appear to be merely an analysis of the ideal scenario. For example, scenarios where the user has not attempted to cover the tracks or simply logged out of an application. This is a reasonable assumption in the case where the application is installed locally. However, in the case of web applications, this is not always the case.

Notably, [23], [24], [29], [38], [41] used the same methodology. Each of these papers conducted a pre-test/post-test quasi-experiment. Other works, such as [25] which use different wording, also used a similar method. In which they identified what artefacts are on the computer before and after the use of the application. Although the pre-test/post-test quasi-experiment seems suitable, other mythologies should be considered as well.

Due to the complicated investigation process, there are some ways to avoid the collection of artefacts, after an incident. To prepare for forensic investigations inside a text-based collaboration tool, Cahyani, Pratama, and Rahman proposed the use of bots. Those bots allow information to be collected almost in real time [44]. This functionality already exists in Slack as part of the compliance package [35].

3.4.4 Automatic Tools to Analyse the IndexedDB

Although research has been conducted on the use of IndexedDB in forensic investigations, as discussed in Chapter 3.4.3, the tools available are inadequate. Paligu, Kumar, Cho, *et al.* developed a prototype with a focus on WhatsApp [24], [43], based on the work of Mendoza, Kumar, Midcap, *et al.*[45]. However no other tools specifically designed to analyse web applications have been identified. Although BrowStEx is capable of providing information on stored data on websites [45].This data cannot be easily analysed because every application stores the information differently. For applications that potentially store a large amount of interesting data, these structures need to be investigated further.

3.5 Research Gap

The literature review in Chapter 3.4 has revealed a research gap in the analysis of Slack, particularly its web application. Although the research has already covered most similar applications, a closer look at Slack could be relevant.

Most of the research has centred on the Google Chrome browser. Chromium browsers hold the largest market share; however, other browsers such as Mozilla Firefox and Safari should also be considered. In particular, Firefox has some discrepancies in its IndexedDB

compared to Chromium, which should be taken into consideration during an investigation. In addition, most of the research has been performed on one single operating system. Although no differences across operating systems are expected, some research is still necessary.

Furthermore, research for IndexedDB is relatively scarce. Specifically, the recovery of the artefacts stored in IndexedDB has not yet been covered. Therefore, currently, the research does not cover the implications of deleting IndexedDB or utilising the private browsing mode.

4. Methodology

To gather insights into the various artefacts left by the usage of the Slack web application, a pretest/post-test quasi-experiment was conducted. Specifically, the aim was to compare an image before and after the usage of the Slack web application and analyse any differences caused by a performed action.

4.1 Justification

The pre-test/post-test quasi-experiment allows for a direct comparison of the image before and after using the Slack web application. During the initial design of the thesis, two research methods were considered to identify the artefacts created by the use of Slack. These methods were an experimental approach and a design-science approach. Notably, both options appeared to be viable for achieving the objective.

For the design science approach, the focus would be on creating a tool to automate the analysis of Slack artefacts. This approach emphasises the development and evaluation of artefacts to solve real-world problems. Due to its practicality, this approach is commonly used to develop new software solutions. "The goal of design science research is utility." [46], and this ensures that the research is practicable. Following the design science approach can ensure a usable product that can be utilised to solve the initial problem. Compared to the experimental approach, it can solve a relevant problem in the digital forensic field. However, the practical focus on the product could lead to a less extensive analysis of edge cases. Although this can be resolved with an extensive evaluation of the product, awareness of this circumstance is required. According to [47], the use of design science is not widely spread in research in the field of digital forensics. As claimed by the authors, a wider use of this research method could be beneficial for the standardisation of research in the field. Nevertheless, the authors seem to advocate for the design science approach, especially to create frameworks. The development of tools does not seem to be a huge factor. Another advantage of the design science approach is its vast flexibility in utilising the research approach, which allows us to focus the research as needed. This means that the research approach can be potentially more efficient, as it focuses on the part that offers the most promise in terms of reaching the result. In comparison, an experimental approach could be more bound to a specific hypothesis. The focus of an experimental approach is more suitable for studying the effects and changes in artefacts caused by the usage of the Slack web application. For an experimental approach, a controlled setting

would be necessary to compare the before and after images. Therefore, the pre-test/post-test quasi-experiment was considered. As shown in the literature review, the research approach has been widely used to perform pre-test/post-test quasi-experiments. Compared to the design science approach, it has the advantages of the experimental approach. The first advantage is similarity to related research. The similarity in the research methods can help the reader compare different applications and ideas more easily. This could provide a way to standardise research methods, as deemed necessary by [47].In the present work, the experimental approach can also be more transparent in analysing the findings. This theory is mainly based on the idea that, with priority given to tool development, the analysis of the artefacts could get less attention. Whether this could lead to a problematic situation depends on the complexity of the artefacts.

Although both research methods were suitable to approach the topic, the experimental approach was chosen. It was chosen because it may accommodate a framework to determine the different factors influencing the artefacts. Moreover, the experiment methodology may provide a more systematic approach for studying and analysing the impact of using the Slack web application on the creation of artefacts. Therefore, it can help in identifying potentially missed scenarios that could affect the interpretation of the results. Furthermore, this approach is more easily replicable. This can be helpful if the environment is changed or similar web applications need to be studied in the future. A quasi-experiment rather than an experiment was conducted because of the use of a web application. Potential changes are not possible to control from an external perspective. Specifically, the pre-test/post-test method was chosen because it helps to easily identify every change that occurs in the participants' behaviour. The results can be compared before and after using the Slack web application. Similar studies [23], [24], [29], [38], [41] have also conducted quasi-pretest/post-test experiments to collect information. Therefore, the method has been tested extensively.

4.2 Pre-test/post-test quasi-experiment

The quasi-experiment design was introduced by Cook Campbell in 1979 [48]. It resembles an experiment but lacks the random assignment of participants to groups. In the context of this study, the quasi-experiment design was appropriate because it allows for a pre-test and post-test design. The design allows us to compare the changes to the system and outcomes before and after the use of the Slack web application. The setup of the pre-test/post-test quasi-experiment in the context of computer science differs slightly from the originally introduced research design. In general, the principles described by Cook and Campbell apply. The research target is subjected to some tests and measurements before and after some form of treatment is performed. Based on the observed changes in the two measurements, the effect of the treatment can be determined [48].

4.3 Ethical considerations

To avoid ethical concerns, all of the collected data was created synthetically and without involving actual users of the Slack web application. This methodology was chosen to ensure the accuracy and validity of the collected data while considering ethical implications. Should a potential vulnerability be found during the experiment, this will be disclosed to the developer.

4.4 Experimental Design

Based on the chosen methodology, the pre-test/post-test quasi-experiment design of the experiment can be formulated in five different stages. During the research, multiple experiments with the same design but slightly different focus areas were conducted to explore various aspects of the Slack web application. To explore the different aspects, the treatments were varied slightly in each experiment, and the actions taken are explained in Chapter 5. To explore the effects of different variables on the use of the Slack web application, the experiment was designed to include variations in operating systems and browsers. For the research, Linux Debian 12 and Windows 11 with Google Chrome and Mozilla Firefox were selected.

Preparation

During the preparation stage, the necessary environment for the experiment needs to be set up. To set up the environment, a new Slack workspace needs to be created by accessing the website, https://www.slack.com, and following the instructions for creating a new workspace. To create the most realistic possible environment, different users were invited

to join the workspace. As a next step, a virtual machine was set up using VirtualBox and the chosen operating system and browser combination. After installing the required software, the preparation for the experiment was complete.

Pre-test

After the preparation, the quasi-experiment pre-test stage was conducted. The chosen measurements for comparison after the treatments require creating a complete image of the (virtual) hard drive. Using the advantages of the chosen virtual machine setup, the virtual hard drive was copied, and FTK Imager was used to create an image in the E01 format. The collected data aims to establish a baseline that clearly shows which files were created or changed by the usage of the Slack web application.

Treatment

During the treatment phase, different actions were performed by the user on the virtual machine as well as by remote users. The actions performed in this phase depended on the focus of each experiment. For example, to identify possible artefacts, users perform a variety of different tasks. Possible tasks can include sending and deleting messages or creating new channels to determine what can be found later on.

The specific actions performed are discussed under each experiment in the next chapter. The different actions taken and their purpose in each experiment, as well as the specific focus of each treatment, are discussed in detail.

Post-test

In the post-test phase, the effects of the treatments on the usage of the Slack web application were evaluated. Therefore, data collection was conducted in the same order as during the pre-test. After data acquisition, the post-test phase involves analysing the collected data and comparing it to the baseline established during the pretest. This analysis allows for a comparison between the files created or changed before and after the treatment phase. Therefore, it provides information on the effects of using the Slack web application.

Validation

The final step in the research process involved validating the results. The results can be validated by comparing the results with the actions performed during the treatment phase. This process helps to ensure the accuracy and reliability of the findings obtained from the experiment.

4.5 Utilised Software

For the experiments, software of the versions mentioned in Table 3 was used. All of the used software was the newest version, as of 24.02.2024.

Software	Version
Oracle VM Virtualbox	Version 7.0.4 r15460
Windows	11 Enterprise Evaluation, 10.0.22631 Build 22631
Debian	12 (Bookworm) kernel 6.1.0-18-amd64
Google Chrome (Linux)	122.0.6261.57
Google Chrome (Windows)	121.0.6167.185
Mozilla Firefox (Linux)	115.7.0esr
Mozilla Firefox (Windows)	123.0
Slack web application	data-version-ts=1709479051

Table 3. Software versions

5. Experimental Results

In this chapter, the experiments performed and the results are outlined. This chapter describes eight different experiments on collecting insights on the artefacts under different circumstances.

5.1 Analysing with Tools

Before performing a manual analysis, it is essential to use tools to assist in the process. To obtain an overview of the capabilities of current tools, the following tools were used to analyse the acquired images. The subsequent tools were considered because they are available for free or offer a free trial version:

- Autopsy
- Belkasoft X

Each software was initially tested with an image of the Windows 11 operating system using Google Chrome. This variation was used because it seems to have the highest statistical probability of being encountered in real-life cases.

5.1.1 Autopsy

Autopsy is an open-source digital forensic platform used for image analysis [49]. The platform provides a set of features for analysing various digital images. Autopsy allows for an extensible platform that allows one to extract web artefacts, perform keyword searches, and create timelines.

Autopsy allows an investigator to analyse images using a variety of built-in tools and external plugins. Therefore, the software can be used for a variety of tasks. To determine the usefulness of analysing Slack usage, we tried to identify potentially helpful plugins. Based on an initial search, no helpful plugin was identified. However, for similar applications such as Microsoft Teams (https://github.com/lxndrblz/forensicsim) and WhatsApp (https://github.com/sathwikv143/AutopsyWhatsapp-Plugin/tree/master), plugins specifically designed for analysing artefacts and data from these platforms exist. Here, it is important to note that both plugins focused on the application for the desktop application rather than websites. However, both desktop applications use the same underlying tech-

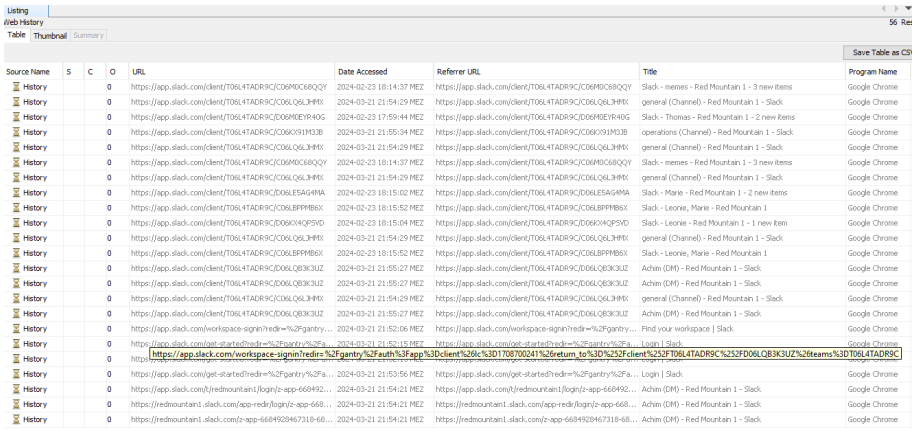


Figure 1. Autopsy - Web History

nology (IndexedDB) as the web version. Therefore, those plugins could be modified to support web application analysis.

During the test, no third-party plugins were used. Instead, we relied on Autopsy's built-in features to analyse the image of the Windows 11 operating system with Google Chrome. The analysis revealed information about the following:

- Web History
- Cookies
- Auto-Fill (used E-mail)
- E-mails of the users

By analysing the image with Autopsy, various artefacts, related to the usage of Slack on Google Chrome, can be extracted and analysed. As listed above, Autopsy allowes to examine the web history, cookies, and auto-fill data. As shown in Figure 1, the web history can provide the investigator with a timeline. The timeline could include events such as when the user opened Slack and when they accessed specific channels or conversations. This information can be crucial in understanding the user's activities and interactions on the Slack platform. Through the website title, it can further establish with whom the user communicates via direct messages. Although it is not possible to determine whether the user has sent messages, this information could still be interesting, depending on the context of the investigation. Another interesting finding from the autopsy is presented in Figure 2. Using the built-in 'keyword search' ingest module, it was possible to find the e-mail address of every participant in the Slack workspace. After further investigation, it was possible to identify that those e-mail addresses were stored in the IndexedDB file. The email address was identified by running a regular expression search against the whole image. However, due to the high number of email addresses found (4867), these options may cause some oversight of that information during an investigation. Furthermore, Autopsy was able

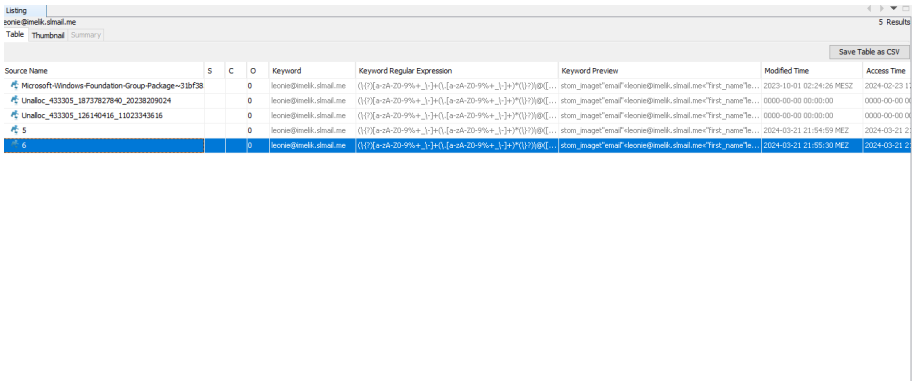


Figure 2. Autopsy - Contacts (E-mail)

to recover the email used to register on Slack by analysing the Auto-Fill database and displaying the cookies created by the use of Slack.

Important to note is also that while Autopsy shows some findings from the browser cache, there is nothing from the cache of the Google Chrome browser related to Slack. However, by analysing the same image with ChromeCacheView, it is possible to retrieve cached files related to Slack, including images and documents.

In conclusion, analysing the image with Autopsy offers valuable insight into the usage of Slack on Google Chrome. Nevertheless, Autopsy shows only some artefacts, which shows that Slack was used. However, to obtain a better understanding of the user's actions, further investigation using additional tools and techniques may be necessary.

5.1.2 Belkasoft Evidence Center X

Belkasoft is a company founded in 2002 that develops a variety of digital forensic software tools. The Belkasoft Evidence Center X (BEC) is the main product of the company and is widely used in forensic investigations. Forensic investigators use it to gather and analyse digital evidence from various devices, including PCs, mobile phones, and drones. While Autopsy relies on plugins to enhance its capabilities, Belkasoft Evidence Center X offers a variety of built-in analysis tools. As shown in the images 3 and 4, Belkasoft offers the option to analyse artefacts from social networks, including Facebook or Twitter. Furthermore, Belkasoft can be used to analyse instant messaging platforms, such as Slack, Skype, and WhatsApp. Of interest for the thesis is the ability to analyse Slack.

To analyse images, Belkasoft Evidence Centre X v.2.4.15170 trial version was used. In general, the software was capable of confirming most of the artefacts found with the analysis performed using Autopsy. The analysis revealed the following artefacts associated

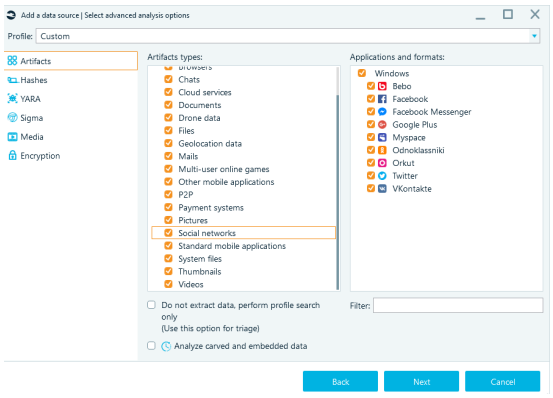


Figure 3. Belkasoft - Social Networks

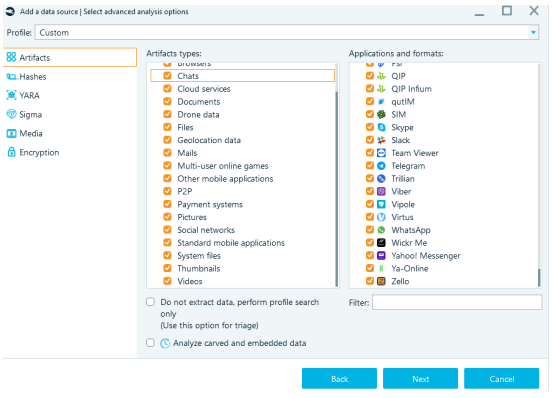


Figure 4. Belkasoft - Chats

with Slack usage:

- Web History
- Cookies
- Auto-Fill (used E-mail)
- Partially recovered messages
- Partially information about the user in the workspace

Especially interesting are the messages and contacts found from Slack usage. The artefacts, as shown in Figure 5, provide some insight into the communication and activities conducted through the Slack platform. Analysing the source of the artefacts reveals that the artefacts were found in the IndexedDB file. Although the artefacts are consistent with Slack usage, some of the information is missing and could mislead the investigation if not considered. For example, the result shows that there is no clear distinction in which channel a message was posted. This could lead to confusion when trying to establish the context of a conversation. Furthermore, there are mentions of people and channels not included in the displayed chat window, as well as messages that included only a file.

Due to the success of retrieving artefacts from the Chrome browser, a similar experiment was conducted with Firefox. Here, none of the messages could be retrieved by the tool. This behaviour suggests that the Belkasoft Evidence Centre was designed to analyse the desktop application Slack, which shares similar technology as Chrome. Further testing also suggested that it is not possible to retrieve those artefacts with BEC if the user logs out of the Slack platform.

Similarly to Autopsy, BEC was unable to retrieve files from the cache. However, some artefacts created by the application in the cache could be obtained, as shown in Figure 6. Overall, this gives the BEC a good starting point for potentially deeper investigations.

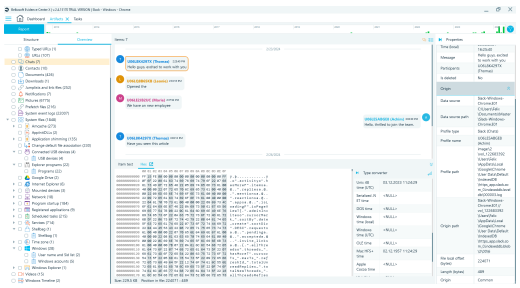


Figure 5. Belkasoft - Messages

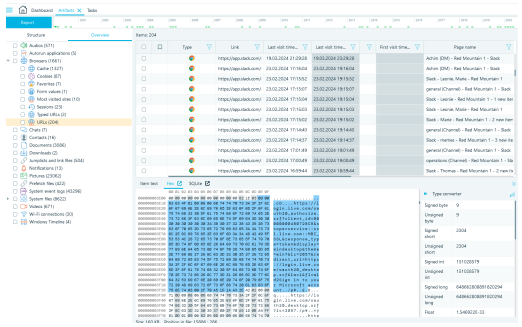


Figure 6. Belkasoft - Web cache

5.1.3 Overview of current tools

Based on the experimentation, current tools demonstrate different abilities to identify artefacts from Slack usage. The image used to gauge the capabilities of the tools was treated, as listed in Table 5. Actions were performed to group potentially available artefacts and compared with the results of the tools, as presented in Table 4. The table reveals that the tested tools were only able to find parts of the actions performed by the user. Furthermore, much information was only obtained by analysing the web history and interpreting the website titles, as demonstrated in Figure 1.

Artefacts	**Autopsy**	**BEC**
Workspace - Name	yes	yes
Workspace - URL	yes	yes
User - Name	partly	partly
User - profile picture	no	no
User - profile data	partly / not explicit	no
Channel - names	partly	partly
Channel - additional information	no	no
Channel - members	no	no
Messages - Contents	no	partly
Messages - DM	no	no
Messages - Reactions	no	no
Messages - Replies	no	no
Messages - modified	no	no
Messages - deleted	no	no
Messages - scheduled	no	no
Messages - Pinned	no	no
Files - file	no	no
Files - Metadata	no	no
Notifications - content	no	no
Notifications - metadata	no	no

Table 4. Comparison of the results generated by selected forensic tools

5.2 Identified artefacts

To identify which artefacts can be recovered, a short exchange between different users was recreated in a freshly created Slack workspace. This leads to the creation of four different workspaces. Each workspace was used by four different users. Three of those users were the same for every workspace, whereas the fourth user was varied. Each workspace was used in another combination of the operating system and web browser, resulting in the following combinations:

1. Windows 11 + Google Chrome
2. Windows 11 + Mozilla Firefox
3. Linux Debian 12 + Google Chrome
4. Linux Debian 12 + Mozilla Firefox

Pre- and post-tests were performed, as described in the methodology section. The treatment between those tests was performed to simulate realistic usage scenarios. Therefore, the steps and actions documented in Table 5 were performed within the Slack workspace.

Action	**User**	**Comment**	**Channel**
Create the Workspace			
Create Channel "Memes"	User 1		
Invite User 2	User 1		
Invite User 3	User 1		
Add profile Picture	User 1		
Add profile picture	User 2		
Add title "Sales"	User 2		
Add tile "Finance"	User 3		
Msg: "Hello guys, excited to work with you"	User 1		general
Reaction to Message from User 1	User 3	ThumbsUp	general
Answer to User 1 and Reaction	User 2	"Yeeeesss", Thank You Reaction	general
Creates Channel "Finance"	User 2		
Write "Opend the Finace Channel @User 3"	User 2		general
Reply "Great!"	User 3		general
Joins Finance	User 3		finance
Add link www.google.com	User 1		general

Write "Finacal Meeting is tomorrow at 1 PM"	User 3		finance
Post a picture	User 1		memes
Fun Emoji	User 2		memes
Msg: We have a new employee	User 3		general
Rply: Great sounds good	User 2		general
Invite User 4	User 1		
Message: Does someone know the password for Google?	User 3		general
Delete Message	User 3		general
Msg: Hello, great to be here	User 4		general
DM: Hey User 4, the password for the Network share is 1df456F_-GHD	User 1		User 4
DM: Great thank you, I will start immediately with reading these things	User 4		User 1
Create channel "Operations"	User 4		
Write "Have you seen the documentation?"	User 4		Operations
Modify Message "Hello, great to be here" to "Hello, thrilled to join the team!"	User 4		general
Message:: Have you seen this article www.newyorktimes.com	User 1		general
Pin Message "Have you seen the article ..."	User 3		finance
Set Channel Topic "Have fun here!"	User 2		memes
Schedule Message "Reminder: Weekly team sync at 10 AM every Monday"	User 4		general
Create User Group "Finance Team"	User 2		
Add Members to User Group "Finance Team"	User 2		
star channel	User 1		general

Share a File "Documentation.docx"	User 3		general
React to "Q1 Documentation.docx" with Eyes Emoji	User 2		finance
Set Reminder "Read Q1 Financial Report" for Tomorrow	User 4		

Table 5. Treatment - general experiment

The manual analysis focused on artefacts that can be found in cookies, browser history, notifications, browser caches, and client-side storage. Those fields were chosen based on the findings in Chapter 2.

5.2.1 Cookies

As shown in Figure 7, cookies do not carry much value for further investigations. Although Slack usage creates approximately 25 different cookies, they do not seem to store information that provides additional value. The default storage location is listed in Table 6.

Source Name	S	C	O	URL	Date Accessed	Name	Value	Program Name	Domain	Data Source
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	tz		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	OptanonConsent		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_cs_c		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_cs_id		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	_cs_mk_ga		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_cs_s		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_ga		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_lc2_fpi		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_lc2_fpi_meta		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_li_ss		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_li_ss_meta		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	_rdt_uuid		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:41 MEZ	cjConsent		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	_gcl_au		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	_ga_QTJQMESMSD		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	b		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	d		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	ec		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:27 MEZ	lc		Google Chrome	slack.com	Slack-Windows
Cookies			0	.slack.com	2024-03-21 21:55:30 MEZ	x		Google Chrome	slack.com	Slack-Windows
Cookies			0	.spiceworks.com	2024-03-21 21:52:15 MEZ	_swauth		Google Chrome	spiceworks.com	Slack-Windows
Cookies			0	.spiceworks.com	2024-03-21 21:52:15 MEZ	_swrid		Google Chrome	spiceworks.com	Slack-Windows
Cookies			0	.t.co	2024-03-21 21:52:07 MEZ	muc_ads		Google Chrome	t.co	Slack-Windows

Figure 7. Analysis of cookies

Web Browser	OS	File path
Chrome	Debian	~/.cache/google-chrome/Default/Network/Cookies
Chrome	Windows	%LOCALAPPDATA%\Google\Chrome\User Data\default\Network\Cookies
Firefox	Debian	~/.cache/mozilla/firefox/[random chars].default/cookies.sqlite
Firefox	Windows	%LOCALAPPDATA%\Mozilla \Firefox\Profiles\[random chars].default-esr\cookies.sqlite\

Table 6. Cookies Location

5.2.2 Browser History

As discussed in Chapter 5.1, the browser history offers some insight into user behaviour. The entries in the browser history consist mainly of URL, access timestamp, referrer URL, and title, as, for example, shown in Figure 1. The data can be found in the locations listed in Table 7. The analysis of the different collected data, reveals no significant difference between the different browsers and operating systems. However, the time and title accessed can provide useful insights into the behaviour of the user. To obtain information from the title, the following patterns were observed:

[Username](DM) - [Workspacename] - Slack (- [amount of unread messages] new items)

The pattern occurs when a user accesses the direct message (DM) with the username mentioned in the title. After the username, the workspace is specified. If the user has unread messages or notifications, the number of these items will be displayed with the caption X new items at the end of the title.

[Channnel](Channel) - [Workspacename] - Slack (- [amount of unread messages] new items)

Similarly to direct messages, if a user accesses a channel, the channel name is specified within the title. The other two potential attributes are the same, as with direct messages.

Based on those titles, with whom and in what channel the user is interacting can be

identified. Furthermore, where the unread messages were received and at approximately what time they were read can be determined.

Web Browser	OS	File path
Chrome	Debian	~/.cache/google-chrome/Default/History
Chrome	Windows	%LOCALAPPDATA%\Google\Chrome\User Data\default\History
Firefox	Debian	~/.cache/mozilla/firefox/[random chars].default/places.sqlite
Firefox	Windows	%LOCALAPPDATA%\Mozilla \Firefox\Profiles\[random chars].default-esr\places.sqlite\

Table 7. Browser history location

5.2.3 Notifications

As Slack creates notifications for new messages, if the user enables this function in the browser, notifications can help in recovering messages. Google Chrome and Mozilla Firefox use slightly different methods to store notifications. While Firefox stores the notifications in JSON form in a text file, Google Chrome stores the data in LevelDB. The location where notifications are stored is listed in 8.

Web Browser	OS	Technology	File path
Chrome	Debian	LevelDB	~/.config/google-chrome/Default/Platform Notifications
Chrome	Windows	LevelDB	%LOCALAPPDATA%\Google \Chrome\User Data\default\Platform Notifications\
Firefox	Debian	JSON	~/.mozilla/firefox/[random chars].default-esr/notificationstore.json
Firefox	Windows	JSON	%LOCALAPPDATA%\Mozilla \Firefox\Profiles\[random chars].default-esr\notificationstore.json\

Table 8. Web notification storage

The notifications from Mozilla Firefox can provide valuable information about the user's interactions on Slack. Namely, the notifications can reveal the content of messages as well as at what time they were received. As established above, the technology used by the

browser differs slightly. Nevertheless, the difference between Linux and Windows seems to be marginal. Interestingly, even if Vincent Lo demonstrated the potential to analyse notifications in Google Chrome [15], no traces were observed during manual inspection.

Because academic research on browser notifications does not provide answers to forensic questions, including, "Can browser notifications help to determine if a user was active on the website during the time of the notification?" To answer it, it is necessary to further investigate the impact of notifications. Therefore, further experiments were conducted and described later in the chapter.

5.2.4 Browser Cache

The browser cache plays a crucial role in forensic investigations, as it can provide valuable insights into a user's browsing activities. In the case of Slack, multiple files were stored during the user's interaction with the application. Analysing the browser cache with ChromeCacheViewer and MZCacheViewer revealed no difference between different combinations of the browser and operating system in terms of artefacts stored in the cache. A smaller difference was observed only in the file path where the cache files were stored, which is shown in Table 9.

Web Browser	**OS**	**File path**
Chrome	Debian	~/.cache/google-chrome/Default/Cache/
Chrome	Windows	%LOCALAPPDATA%\Google\Chrome\User Data\default\cache\
Firefox	Debian	~/.cache/mozilla/firefox/[random chars].default
Firefox	Windows	%LOCALAPPDATA%\Mozilla \Firefox\Profiles\[random chars].default-esr\cache2\

Table 9. Web cache location

Although there was no indication that messages exchanged over Slack could be recovered from the browser cache, other information related to the user's activity on Slack was visible. As stated in the treatment of the experiment, two files were sent by users on Slack: an image and a docx file. Both files were found in the browser cache. As shown in Figure 8, it was possible to recover the content of both files. Although the image was not altered, the docx file was converted to a PDF and a PNG thumbnail picture.

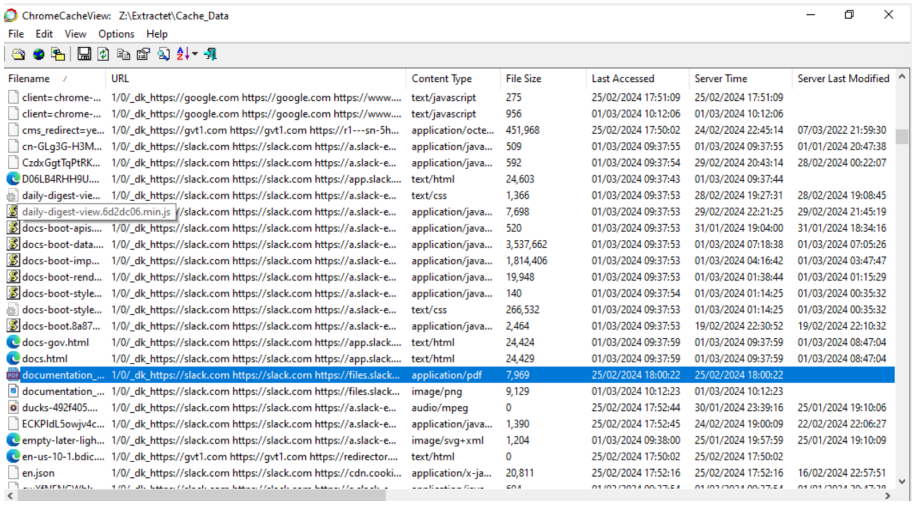


Figure 8. Analysis of the browser cache

Furthermore, it was possible to recover all profile pictures of the users from the browser cache. These findings highlight the potential forensic value of the browser cache in the case of a potential investigation. Based on the analysis, the browser cache is the only reliable source to retrieve files from the Slack workspace if the users are not downloading them. However, the recovered files are not the same as the original ones used during the experiment. A comparison of the hash values suggests that the files were altered after the upload to Slack.

5.2.5 Client-Side Storage

Client-side storage proved to be the most valuable source of digital evidence in analysing messages. By examining client-side storage, specifically IndexedDB, a wealth of information related to user activity and message exchanges on Slack was uncovered. Thus, the content was again similar across each browser and operating system variation. The main difference between Firefox and Chrome is, as described in the previous Chapter 2.5, the different technologies used for IndexedDB. Therefore, different parsing methods to read the content are necessary. The location of the IndexedDB is listed in Table 10.

Web Browser	OS	Technology	File path
Chrome	Debian	LevelDB	~/.config/google-chrome/Default/IndexedDB/
Chrome	Windows	LevelDB	%LOCALAPPDATA%\Google\Chrome\User Data\default\IndexedDB\
Firefox	Debian	SQLite	~/.mozilla/firefox/[random chars].default-esr/storage/default/
Firefox	Windows	SQLite	%LOCALAPPDATA%\Mozilla\Firefox\Profiles\[random chars].default-esr\storage\default\

Table 10. IndexedDB storage

Depending on the underlying technology, different approaches are needed to convert the files to a JSON file. The main tools, written in python are [18] for Google Chrome and [19] for Mozilla Firefox.

Google Chrome

In Google Chrome, the following file structure is used to store the IndexedDB file. As discussed in Chapter 2.5, LevelDB uses different files to store information.

```
|-- https_app.slack.com_0.indexeddb.blob
   |-- 2
      |-- 00
         |-- 2
|-- https_app.slack.com_0.indexeddb.leveldb
   |-- 000003.log
   |-- 000005.ldb
   |-- CURRENT
   |-- LOCK
   |-- LOG
   |-- LOG.old
   |-- MANIFEST-000001
```

The directory https_app.slack.com_0.indexeddb.leveldb stores most of the LevelDB database. The *.log file (i.e. 000003.log) stores up to 4 MB of data. Here are the most recent updates stored. If the size limit is reached, the updates are moved to a sorted table, and a new log file will be created [50]. The current log is also stored in memory. Analysing the context of the 000003.log file and 000005.ldb points to an external object, as shown in Figure 9.

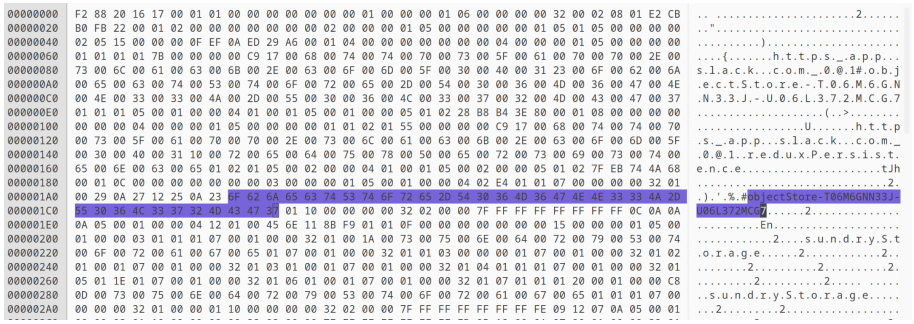


Figure 9. Chrome - Analysis 000003.log

The external objects were stored in the directory https_app.slack.com_0.indexeddb.blob and analysed later in the chapter. Other files in the directory https_app.slack.com_-0.indexeddb.leveldb were CURRENT, LOCK, LOG, LOG.old, and MANIFEST-000001. The manifest file stores metadata and will be recreated with an incrementing number when the database is reopened. The current file is a basic text file that specifies which manifest file is the most recent. The LOG and LOG.old files store informational log messages of the LevelDB, such as the creation of the database. An example of the content is shown in Figure 10. Furthermore, the directory can include other files like LOCK or *.dbtmp, which are used internally in the LevelDB or for miscellaneous purposes [50].

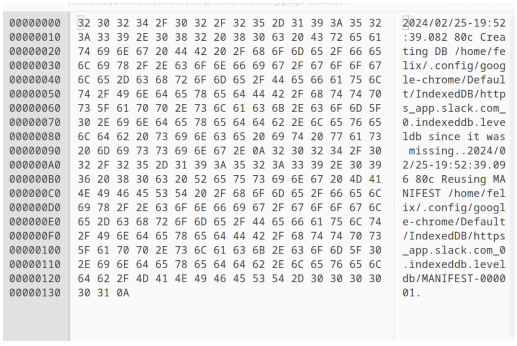


Figure 10. Chrome - Example LOG

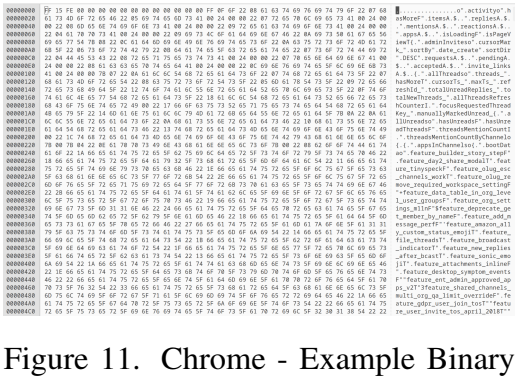


Figure 11. Chrome - Example Binary Blob

LevelDB stores the external objects as binary blobs [50]. In the analysed case, Slack heavily utilises external objects to store information on the client side. Inside the binary blob, shown in Figure 11, data is stored in a JSON file.

Mozilla Firefox

Unlike Chrome, Firefox uses SQLite to handle IndexedDB. The IndexedDB is mainly stored in the file *reednuextPseirs.sqlite. The SQLite database includes seven different tables, as shown in Figure 12.

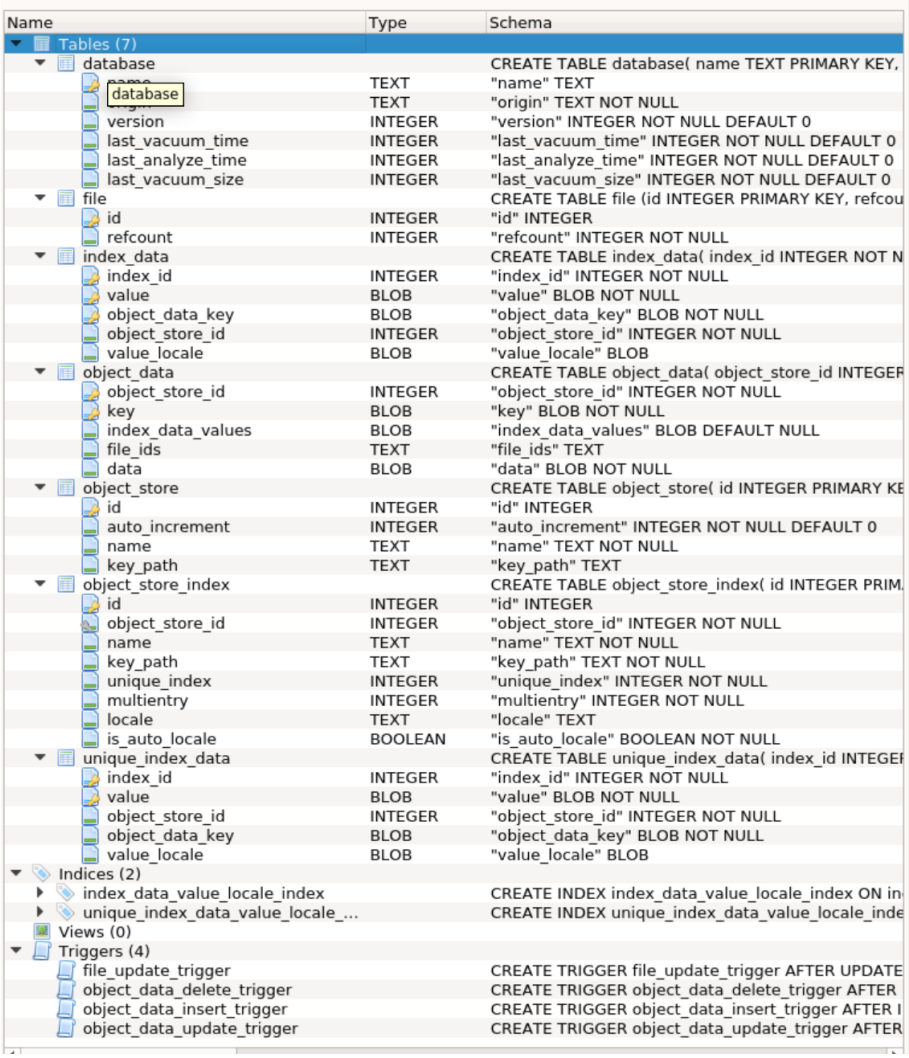


Figure 12. Firefox - Databases Indexed DB

Similar to the usage on Google Chrome, Slack uses external objects to store the information of the Slack workspace on the client side. To obtain the data, the following SQL statement can be used:

```
SELECT data
FROM object_data
ORDER BY object_store_id
```

The query returns a binary blob, as shown in Figure 11. The binary blob also contains the JSON file with the Slack user data.

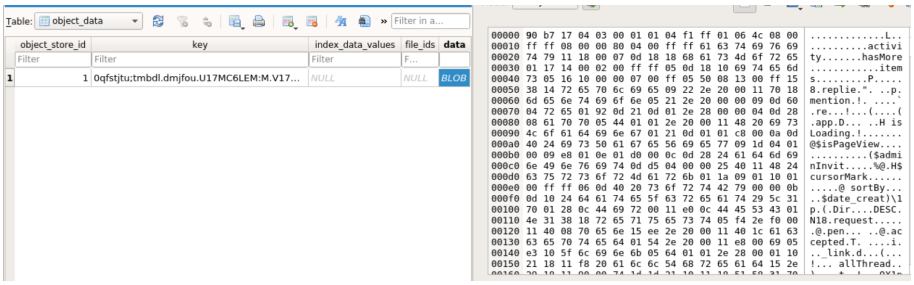


Figure 13. Firefox - External Object

JSON

Analysing the received JSON gives further insights into the user's activity on Slack. Based on the findings, three different databases were identified: channels, users, and messages. Figure 14 shows a general overview of the content and the relationship between these databases. The diagram was cut to include only the most relevant attributes. An example of each entry with all attributes is attached in Appendix 2.

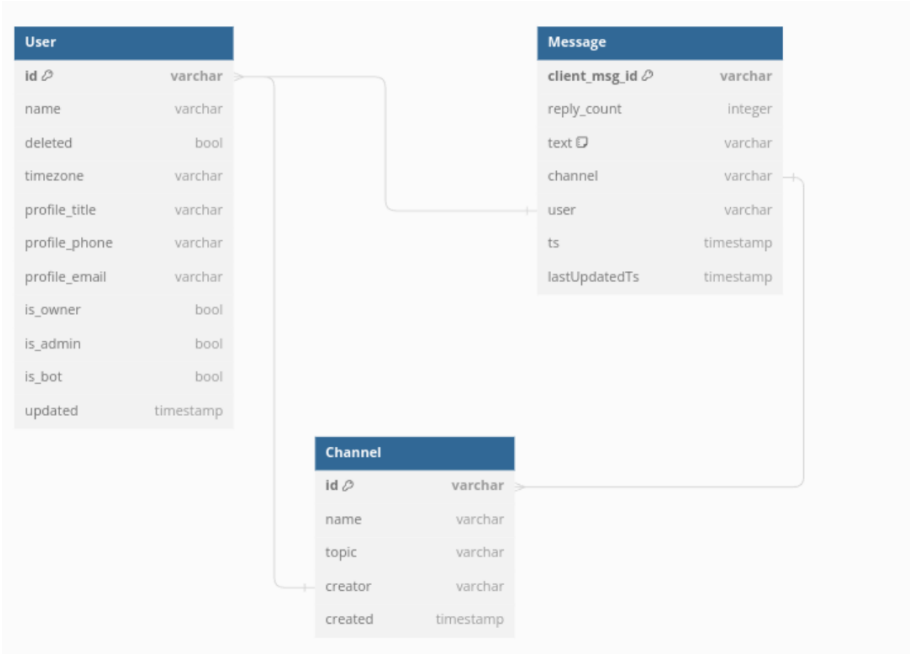


Figure 14. IndexedDB - stored information

5.2.6 Conclusion - found artefacts

With manual analysis of the different images, the number of artefacts found could be increased. As mentioned in this chapter, a variety of artefacts could be recovered. The artefacts contain different messages, responses, and direct messages, and some channels were not found during a manual inspection. Table 11 provides a summary of the results,

where particular actions were found (C - Web Cache, H - Web History, PS - Persistent Storage, N - Notifications).

Artefacts	**Recovered**
Workspace - Name	yes (H,PS)
Workspace - URL	yes (H)
User - Name	yes (PS)
User - profile picture	yes (C)
User - profile data	yes (PS)
Channel - names	yes (H,PS)
Channel - additional information	yes (PS)
Channel - members	yes (PS)
Messages - Content	yes (PS,N)
Messages - DM	yes (PS,N)
Messages - Reactions	no
Messages - Replies	yes (PS,N)
Messages - modified	no
Messages - deleted	no
Messages - scheduled	yes (PS)
Messages - Pinned	yes (PS)
Files - file	yes (C)
Files - Metadata	yes (PS)
Notifications - content	partly (N)
Notifications - metadata	partly (N)

Table 11. Summary of manual findings

5.3 Amount of stored Messages

Analysing the general availability of artefacts stored on the client side suggested that some messages sent were not stored in the IndexedDB. To better understand how many messages were stored, the following experiment was performed. For the experiment, a six-user workspace was created; four users would listen passively to the messages, whereas the other two would send messages into the workspace. After all the messages were exchanged, the stored data from the four passive users, each of whom uses a different setup, were extracted and further analysed.

In the experiment, the plan was to send 706 messages divided into the following:

- General Channel - 350 messages
- Other Channel - 300 messages
- No Visit Channel - 50 messages
- Small Channel - 5 messages

- New Channel - 1 message

The division of the different channels was based on the following assumptions:

1. There is a maximum amount stored per Channel, because older messages can be fetched from the server if needed.
2. The messages will be stored in the IndexedDB, even if the channel is not visited.
3. Messages will not be stored in the channel, if the amount is small enough to fetch them quickly from the server.

Therefore, the channels are based on those three assumptions. The General Channel and Other Channels were used to determine how many messages could be stored. Where more than 90% of the messages could be recovered, the experiment was repeated with more messages. The “No Visit” Channel sought to test the second hypothesis: while the passive user will read all the other messages, they will not access this channel. The last two channels, “Small” and “New”, focused on the third assumption, namely establishing whether a threshold exists.

To create the messages, two Sherlock Holmes short stories were used[1]. Each message sent in the Slack workspace consisted of one line from the short stories and an Identifier, as shown in the following examples:

```
"Excellent!" said Holmes, who was recovering his good-humour as his A1yEJ5jxYsSw
attention became more engrossed by the case. "Fortune has been your A0MG2tIQQnjW
friend." AOCWOpkTvpIi
```

The identifier at the end of each line was used to map the messages during the analysis, thus ensuring a unique message. The messages were written using a Python script to ensure the right number of messages and log those. The script can be found on GitLab[2]. After sending the messages, the JSON file was extracted from each browser on four different occasions to determine the impact of the listed actions on the stored data. The four actions are listed below.

1. Directly after the treatment
2. After reloading the website
3. After 2. and restarting the browser
4. After 3. and scrolling through the general and other channel

[1] https://sherlock-holm.es/stories/plain-text/
[2] https://gitlab.cs.ttu.ee/fewasc/slackwebappforensic/-/tree/main/Slack%20Messenger

The results of the experiment can be found in Figures 15 and 16. The data shows, that, in most cases all messages could be found in the IndexedDB. The exception is the extraction of the IndexedDB from Firefox (on Windows) directly after the treatment. In this case, the IndexedDB only contained 477 out of the 706 messages. This was true in the specific case that the newest messengers were not added to the IndexedDB, as indicated by the distribution per channel in Figure 17.

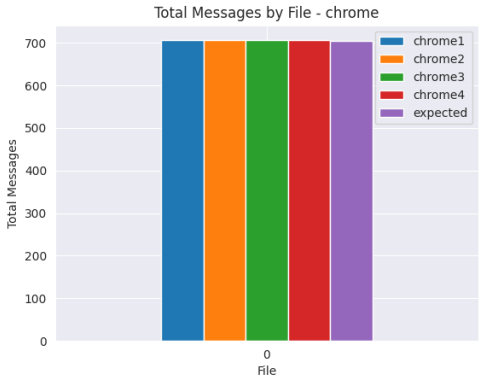


Figure 15. Chrome - Windows - stored Messages

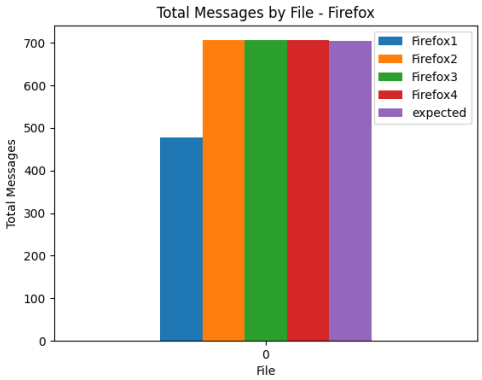


Figure 16. Firefox - Windows - stored Messages

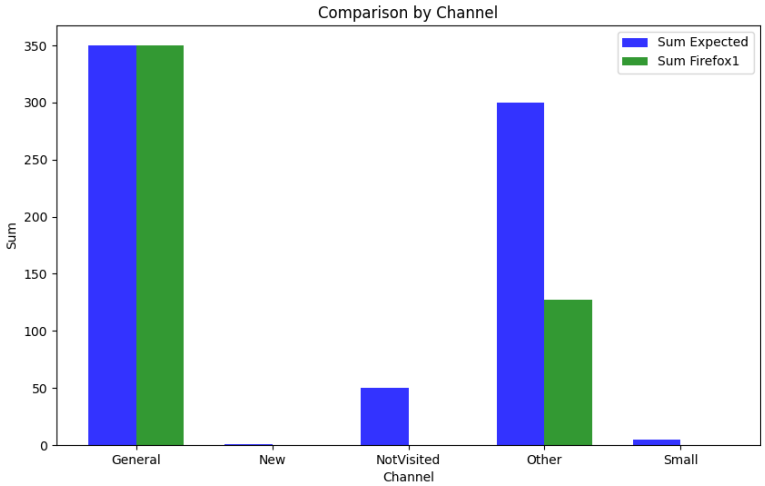


Figure 17. Firefox - Messages per Channel

In a second experiment, 4500 messages were sent by a single user to a single channel over a period of five hours. The result was comparable to the first experiment. This time, before refreshing the page, 4173 messengers were stored in the IndexedDB of Google Chrome; after the refresh, all 4500 messengers could be found. Firefox contained, at both times, all 4500 messages. Due to the discrepancy in the stored messages, a thorough experiment with 150 messages was conducted. Unlike the previous experiments, the site was left idle

for 10 min. In these cases, both web browsers stored all the messages without the need to refresh the page. In the case of Google Chrome, the IndexedDB had a size of 8.2MB after the 4500 messages were stored inside. Considering that the IndexedDB can store up to 50 MB per site, recovering more than 20000 messages from Slack, depending on the average length of each message, seems reasonable.

5.3.1 Conclusion - Amount of stored messages

Based on the experiments conducted, some insights into the storage behaviour were gained. The first insight is that there is seemingly no limitation to the number of messages stored per channel. The experiment was carried out with the assumption that only the newest messages would be stored locally. This assumption seems to be wrong, even 4500 messages in one channel were still stored locally. The assumption that a minimum amount of messages is required in a channel was also incorrect, since even a single message in a channel was stored locally. Another important insight from the previous experiments is that messages will not always be stored directly on the hard drive. As shown in Figure 16 and in a small experiment, it can take some time before all the previous messages are stored in IndexedDB. Refreshing the site can speed up this process.

5.4 Notifications

As established in Chapter 5.2, the analysis of notifications can offer some insight into the user's activity. There is currently not much research on the forensic value of browser notifications. Therefore, further experimentation is required to understand whether and how notifications can be used as a source of digital evidence in forensic investigations. To examine how the storage of notifications behaves and what information can be extracted, a new workspace was created with three users. The goal of the experiment was to gather notifications in different scenarios in Google Chrome and Mozilla Firefox. For simplicity of the experiment, only Linux was used as the operating system. During the treatment phase, the following scenarios were examined:

- Arriving message, when the user is in the same channel
- Arriving message, when the user is in a different channel
- Arriving message, as DM (direct message)
- Arriving message, when the Browser is minimised
- Arriving message, when the PC is locked
- Arriving message, when the PC is locked and the PC is shut down before reading the message

After the treatment phase, the data collected from the experiment was analysed to determine the behaviour of notification storage and what information could be extracted. Before examining the data, it is important to note that a notification was not generated in every scenario during the experiment. In cases where a message was received in the channel and the user is currently active, the notification was not displayed. There was also no notification created when the user was active in the channel and then locked the computer. When the browser was minimised, the notification was created.

5.4.1 Firefox

As established, in Chapter 2.6, the notifications created by Firefox were stored in the notificationstorage.json file. This file contained, after the experiment, three different entries. Each entry in the file follows the following schematic:


```
{"id":"{efdd5025-15f0-4c83-a245-dacce6d3e4ea}",
"title":"New␣message␣in␣#general","
dir":"auto","lang":"",
"body":"Noha:␣Now␣the␣Browser␣is␣minimized",
"tag":"tag_1711390872.577849",
"icon":"",
"alertName":"https://app.slack.com#tag:tag_1711390872.577849",
"timestamp":1711390872722,"
origin":"https://app.slack.com",
"data":"",
"mozbehavior":"{\"noclear\":false,\"noscreen\":false,
\"showOnlyOnce\":false,\"soundFile\":\"\"}",
"serviceWorkerRegistrationScope":""}
```


Considering that more than three notifications were created, the title of each stored notification differs. Therefore, only the last message received in a channel is stored, at least during the usage of Slack. A notification appears only when the user is not active in the receiving channel. As shown in the displayed entry, the stored notification consisted of different attributes, such as title, body, tag, icon, origin, and timestamp. For most forensic investigations of Slack usage, the attributes title, body, and timestamp could create the most value. The title can offer information on whether the message is sent as a channel or as a direct message. Additionally, the body contains the content of the message as well as the name of the sender. The timestamp is, in the case of Slack, similar to the time the message was sent. Notably, the timestamp can be freely chosen by the application [51]. Therefore, extensive testing is required if notifications from other applications are analysed. Based on the collected data, it does not seem possible to determine if a user has seen the

notification or was interacting with it (i.e. clicking on it). Moreover, it seems impossible to receive more information about the time the notification was created.

5.4.2 Google Chrome

As already shown in Chapter 5.2, during the initial inspection, previous notifications could not be inspected. This behaviour was confirmed during the second experiment. Although Vincent Lo showed that recent notifications were stored in the file 000003.log, these files remained empty during and after the experiment [15]. An extended search through other changes in files after receiving the notification also failed to reveal any notification content. Further analysis suggested that the storage of these notifications occurred only if the PushAPI was used.

5.4.3 Conclusion - Notifications

Based on the closer inspection of notifications, the following insights were obtained. First of all, it was only possible to find artefacts left by the notifications when Firefox was used. This behaviour contradicts the research by [15]. The difference from the experiment from Vincent Lo is the usage of PushAPI, which is not used by Slack. Firefox stored the previous notification in the notificationstore.json file. This file contained the last notification, which refers to a message in each channel. In general a notification will be created, when the user has a tab with Slack open. However, if the user is currently in the channel, where a new message was sent, no notification will be created.

5.5 Logging out

After identifying potentially useful artefacts and determining the amount of data available for analysis, the ability to collect data under less ideal circumstances needs further consideration. As previously established, the most valuable data, the messages, are stored in an IndexedDB structure. Based on a code analysis, the IndexedDB will be deleted during logout in Slack.

```
// function called during logout from https://www.slack.com
 function F(q, blocked: X=)

                    const he = indexedDB.deleteDatabase(q);
                    return X  he.addEventListener("blocked", ()=>X())
                    N(he).then(()=>
                    )
```

The deletion of the IndexedDB file could prevent the recovery of exchanged messages and user information from the user's hard drive. The problem is evident during experiments with the BEC in Chapter 5.1.2. As soon as an image in which the user logged out of Slack was used, the tool was no longer able to display any of the exchanged messages. Although this scenario is not covered by studies in Chapter 3.4.3, this problem seems likely to arise in practice. To determine whether and how a forensic investigator could recover artefacts after logout, another experiment was conducted. For this experiment, preparation included the actions in Table 5. After performing the actions, the first image, of the hard drive was taken. Subsequently, in the treatment phase, the user was logged out, and another image of the hard drive was taken. After verifying that the first image contained previously identified artefacts, the second image was inspected.

The inspection of the image after the logout confirmed that the IndexedDB file will be deleted, as shown in Figure 18 (before the logout) and Figure 19 (after the logout). Other artefacts, like cache, web history, and notifications (in Firefox), do not seem to be affected by the user logging out.

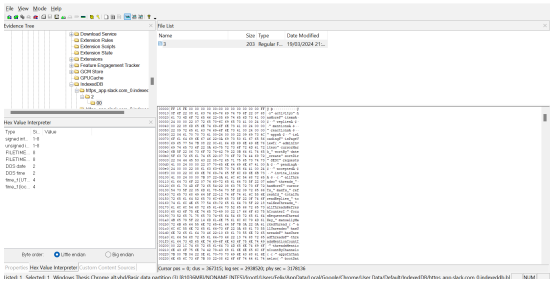


Figure 18. Chrome IndexedDB - before logout

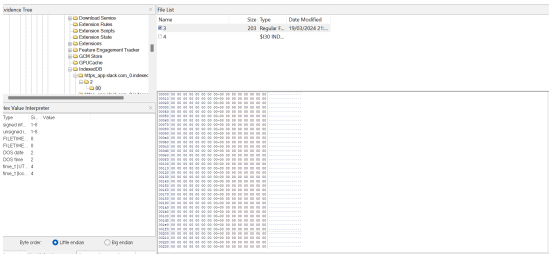


Figure 19. Chrome IndexedDB - after logout

5.5.1 Recovery of IndexedDB files in Chrome

After it was established that the IndexedDB files were deleted during logout, it seemed necessary to explore the possibility of recovering the data. Anuradha, Kumar, and Sobhana demonstrated the possibility of recovering deleted browser artefacts on a Linux operating system [52]. Yet, the IndexedDB was not covered by the research. Because Chrome uses different files to store the IndexedDB, the files may have only been deleted at the file system level and not immediately overridden. This assumption implies that the file can be located on the hard drive. To verify whether this assumption is valid, the pre-logout image was used to locate the file on the hard drive, as shown in Figure 20. The exact location allowed the same file to be identified on the post-logout image. Parsing the found file revealed that the content was equal to the pre-logout file. Because in most scenarios, the

location of the file is not known to the investigator, multiple Slack IndexedDB files were analysed. During the analysis, a search for the following bytes improved the ability to find the start of the file with minimal false positives.

Listing 5.1. Startbytes IndexedDB

```
FF 15 FE 00 00 00 00 00 00 00 00 00 00 00 FF 0F 6F
```

Reliable identification of the end of the file proved more difficult because the last entry in the file seemed to be a hash value, which means it was changing. To reliably extract the complete file, it was necessary to manually check all entries containing the following.

Listing 5.2. Endbytes IndexedDB

```
70 65 72 73 69 73 74 65 6E 63 65 48 61 73 68 65 73
```

After finding the specified bytes, checking was needed to determine whether they were the last for this IndexedDB blob. The analysis of different images suggested that they occurred mostly two times in the IndexedDB file before the file ends.

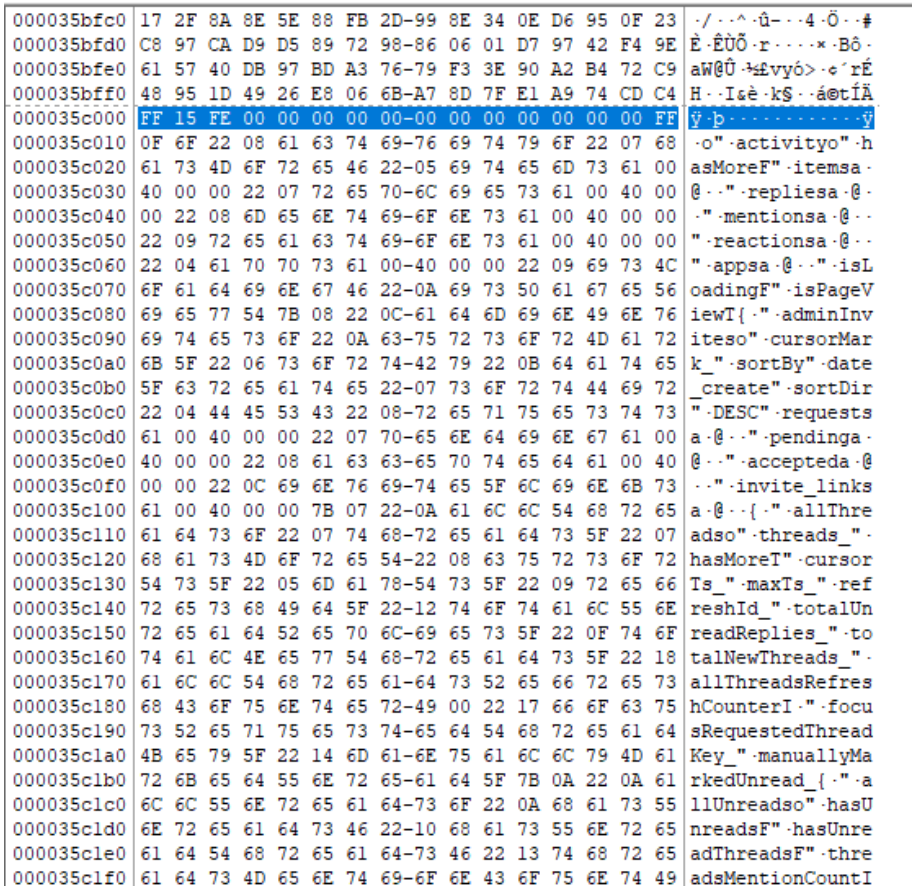


Figure 20. Chrome IndexedDB - recovered after logout

The deletion of the IndexedDB can help establish the exact moment when the user logged out of the Slack workspace by inspecting the timestamps in the filesystem journal. However, it was only possible to recover the external object, and the .log and .ldb files were seemingly overridden during the experiment.

5.5.2 Recovery of IndexedDB files in Firefox

Firefox was analysed using the same process as that of the image of Google Chrome. As shown in Figure 21, the IndexedDB entry will be deleted after the logout.

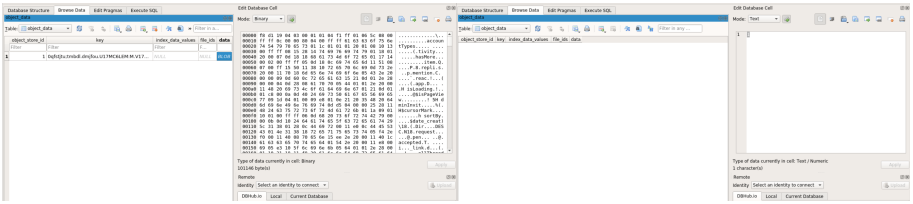


Figure 21. Firefox IndexedDB - before and after logout

However, Pawlaszczyk and Hummert proposed a potential recovery method for SQLite databases [53]. According to them, it is possible to recover deleted entries from a SQLite database as long as the data is not yet overwritten [53].

A comparison of the SQLite database before and after the log-out was able to validate that the content of the SQLite databases only changed slightly. To recover the entry, the FQLite tool [53] was used. As shown in Figure 22, the tool was able to recover the entry.

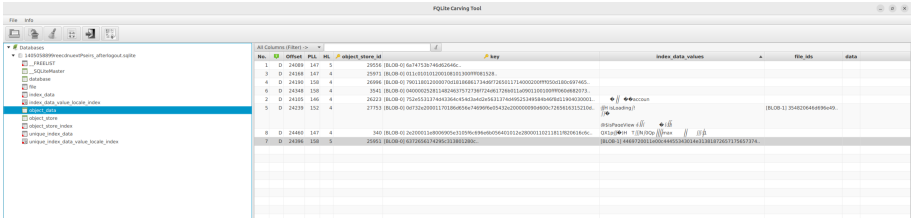


Figure 22. FQLite - Firefox IndexedDB - recovered entry

5.5.3 Conclusion - Logout

While logging out of Slack prevents access to the data stored in the IndexedDB, it is still possible to recover the data in Google Chrome and Mozilla Firefox. Due to the different technologies used, the recovery process depends on the context. The ability of an investigator to recover data depends on the amount of data written between the time of

logout and the time of image collection.

The behaviour could potentially be considered a vulnerability, therefore the result of the work was shared with Salesforce on HackerOne[1]. Because the vulnerability can only be exploited by a physical attacker or a compromised machine, the report was classified as informative.

5.6 Private Browsing

The ability to browse in private mode (or incognito mode in Google Chrome) is a standard feature of common web browsers. This feature can prevent the evidence from being stored on the hard drive [28]. According to [28], privacy mode is used as an anti-forensic method. The literature has already covered a wide range of the different aspects of private browsing. Privacy leakage was analysed by [28], [54], [55]. Horsman, Findlay, Edwick, *et al.* analysed the privacy leakage of 30 different browsers [28]. The research [28] traced privacy leakage in Avant, Comodo Dragon, Edge, Epic, and Internet Explorer [28]. Fernandez-Fuentes, F Pena, and Cabaleiro focused their analysis on Mozilla Firefox and Google Chrome in a Linux environment [54]. Their work also revealed that no data is stored on the hard drive when privacy mode is enabled. However, they were able to retrieve artefacts from the RAM.
Kathiravan, Mohamad Amran, Mat Razali, *et al.* presented an overview of six studies that analysed various privacy features on common browsers. Based on the studies, the authors concluded that the privacy mode could cause some leakage [55]. This suspicion is based on the results of [56] and [57] which indicated that the browsers store data, at least temporarily, on the hard drive. However, the number of artefacts available was limited.

A limitation of the previously mentioned works is that neither study included client-side storage and notifications in their study, despite both potentially containing evidence in the case of Slack. To determine whether privacy browsing in Firefox and Chrome also obfuscates this evidence, another experiment, similar to previous experiments, was conducted. A list of the actions performed can be found in Table 5. Unlike previous experiments, the Slack website was visited in the privacy mode of both browsers.

5.6.1 Notifications

The experiment revealed that Firefox does not allow notifications during privacy mode. Therefore, it is not possible to collect a previous notification, as discussed in Chapter 5.4.

[1] https://hackerone.com/

Notifications can be enabled on Google Chrome; nevertheless, similar to the experiment with the notifications, there was no persistent data left.

5.6.2 Client-Side Storage

Furthermore, it was also not possible to gather evidence from client-side storage, when Slack was used in private browsing mode. Both browsers, Chrome and Firefox, utilised techniques to avoid privacy leakage. Based on [58], Firefox encrypts the data stored in IndexedDB before writing it to the hard drive. This technique seems to mask the content so that it is not possible to recover potential evidence.

5.6.3 Other artefacts

To validate previous research, potential artefacts like web history, cookies, and caches were also analysed. Similarly to the notification and client-side storage, no artefacts could be recovered. This agrees with previous research, such as [28].

5.6.4 Conclusion - Private Browsing

The result of the experiment was corroborated by the results of [28], [54], [55]. Neither Mozilla Firefox nor Google Chrome exhibited privacy leakage when private browsing modes were used. The potential technologies, such as IndexedDB and notifications, were not considered in previous research, as do not present an exception to the findings. However, the experiment was only conducted on the most recent browser versions with the default settings. Some mentions in [58] indicate that Firefox versions below 115 (released on 04.07.2023) do not support websites that use IndexedDB for private browsing. As a workaround, it was possible to enable IndexedDB for private browsing "by setting dom.indexedDB.privateBrowsing.enabled to true"[58]. In this scenario, the IndexedDB would not be encrypted and could be analysed.

5.7 Anti-forensic

After analysing what artefacts are available and how the data can be recovered, the follow-up question, namely how someone could prevent it, was posed. While the use of the private browsing mode seemingly does not leak any information, other methods should also be considered. Anti-forensic methods are used to destroy or mask potential evidence as well as delay or mislead future investigations [59]. In previous experiments, the option to log out of the Slack application and the usage of the private browsing mode, were analysed.

This actions could be counted as anti-forensic methodes. Although the private browsing mode could hide evidence, simply logging out of the website was not sufficient to do so if the investigator attempts to recover the IndexedDB files.

Because of the potentially large number of different techniques, only the most likely ones were considered in this chapter. Therefore, only actions performed in the browser GUI were included. This mainly applies to the ability to delete several stored pieces of information, such as cookies, cache, and history. The experiment included the actions specified in Table 5. During the treatment phase, the “Clear Browserdata” button was pressed.

5.7.1 Google Chrome

After deleting all browser data, all artefacts identified in Chapter 5.2 were removed. However, based on the techniques used in Chapter 5.5, the blob file of the Indexed DB in Chrome can be recovered again, as shown in Figure 23. The figure shows the file before (left) and after (right) clearing all the browser data. While the data was initially deleted, it could be recovered as long as the previous files were not overridden.

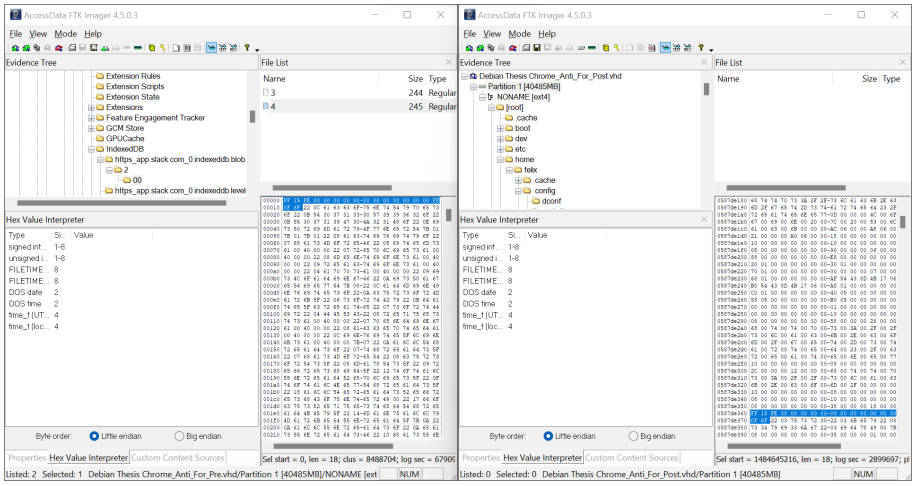


Figure 23. Antiforensic - Chrome IndexedDB - recovered entry

5.7.2 Mozilla Firefox

Similarly to Google Chrome, the artefacts identified in Chapter 5.2 are not accessible after the deletion of browser data. As shown in Figure 24, which shows the SQLite file before (left) and the deleted SQLite file after (right), the SQLite file was deleted with the deletion of browser data. This differs from the IndexedDB recovery after the logout from Slack. While recovering a certain database entry after logout, the deletion of browser data requires

the recovery of the whole SQLite file. As shown on the right side of Figure 24, the content of the file is still available until the data is overridden.

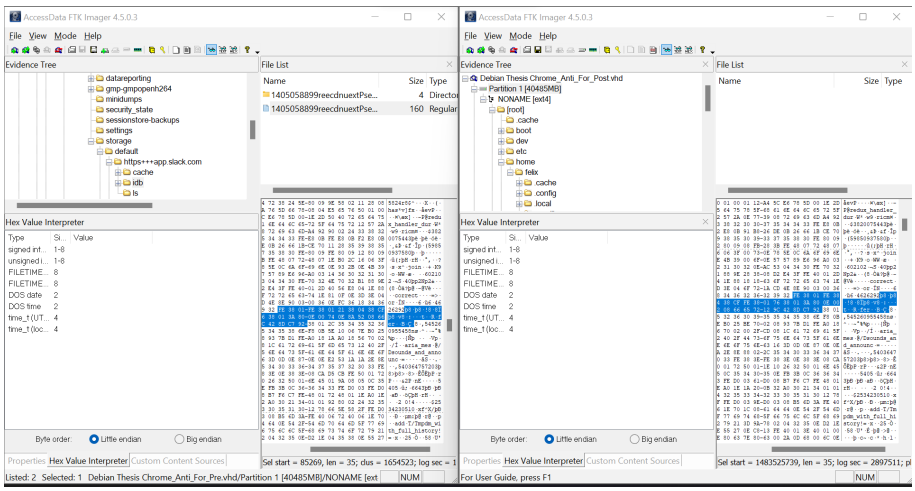


Figure 24. Antiforensic - Firefox IndexedDB - recovered entry

5.7.3 Conclusion - Antiforensic

Various methods to mask user behaviour in web browsers exist. Basic techniques, including deleting browser data or using private mode, were discussed in the context of the Slack web application. The previous chapter proved that the deletion of browser data does not fully remove artefacts. Both web browsers, Google Chrome and Mozilla Firefox, do not fully delete the data in the IndexedDB, allowing an investigator to recover it.

5.8 Deleted and Modified Messages

Slack allows the user to modify or delete messages after sending them. An experiment was conducted to determine the possibilities of analysing these messages. For the experiment, a workspace with five users was created. Four of the users used a different combination of operating systems and web browsers. Each user composed 50 unique messages and sent one image, and 10 messages were later deleted and another 10 modified. The sent image was also deleted. Similar to the experiment to determine the amount of messages stored, the messages were created with the same format. Each message, therefore, is consistent and represents a line from Sherlock Holmes, and each has a unique identifier. To modify the messages, the identifier was changed to another unique value. After the treatment, the four different IndexedDB files and the Firefox notification file were collected.

Analysing the IndexedDB revealed that the deleted messages were not stored in the IndexedDB. Furthermore, no indication revealed that the original message of the modified

messages was still stored in the IndexedDB. However, modified messages contained the following attribute:

Listing 5.3. Attributes modified messages

```
[...]"edited":{"user":"U071UQQS4RG","ts":"1712967767.000000"},[...]
```

Therefore, that a message was edited previously could be confirmed, which allows to gather information about the last user who edited the message and when this was done. The current IndexedDB file does not contain the original content or deleted messages. Therefore, similar to the results in Chapter 5.5, it was assumed that a previous IndexedDB could still contain the information. However, none of the hard drives seemingly contained a previous IndexedDB file.

Notifications collected from Firefox usage were examined in Chapter 5.4. Inside notificationstorage.json file, the last messages created in each channel are stored. When the last message on a channel is deleted or modified, the content can be recovered. While the web notification theoretically allows one to alter a message after sending it, this does not seem to happen in the case of Slack.

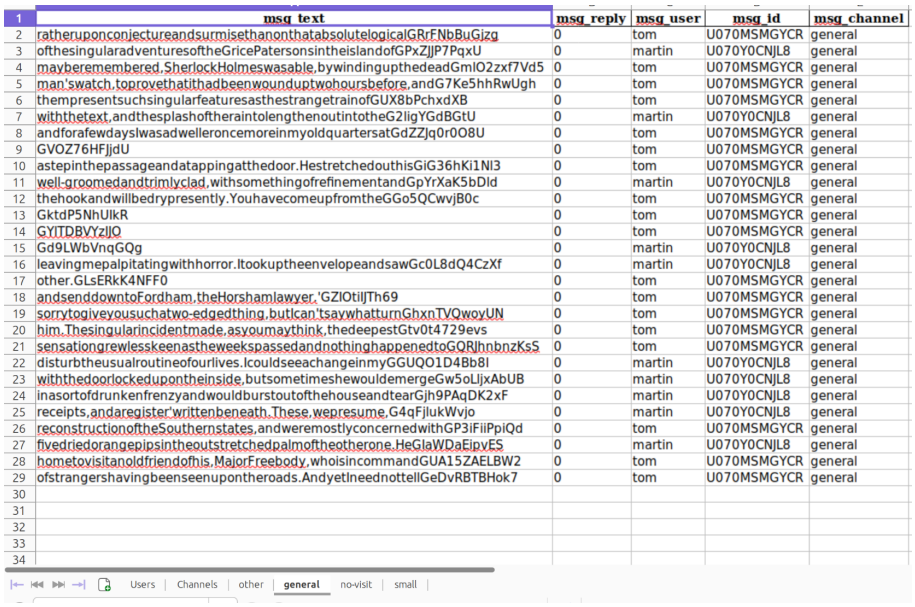
5.8.1 Conclusion - Deleted and Modified Messages

The ability to recover the original content of modified or deleted messages is fairly limited if the investigator needs to rely on the hard drive of a user. Based on the experiment described previously in Chapter 5.4, the only way to recover the content of those messages is if a notification is created. Additionally this message has to be the last message received in a channel and the user has notifications enabled. Due to this limitation, deleted or modified messages created by the user cannot be recovered. Insights into this content can also be obtained by recovering the IndexedDB from another user if the user last visited Slack shortly before the message was deleted or modified. Furthermore, it is possible to recover sent files that were later deleted. Those files can still be in the browser cache even after the file is deleted.

5.9 Automation of the process

The results have shown that common forensic tools are currently incapable of completely extracting the artefacts discussed in the thesis. Therefore, the relevance of IndexedDB for an analysis of Slack is crucial. To help in the investigation of the IndexedDB, a Python script was created to parse the relevant information and output the insights in a structured way. Currently, the created tool creates a spreadsheet with different sheets, separated

for each recovered channel, with messages and an overview for each user, as shown in Figure 25. The code can be found on GitLab[1]. The tool uses regular expressions to extract the data from the JSON file, which are appended in Appendix 3. This choice was made because the tools presented in Chapter 2.5 were not always able to recover a completely valid JSON file. The tool is used to validate the findings if an IndexedDB is extracted. This helps inlinking message, channel, and user information, among others, without creating excessive overhead.



	msg_text	msg_reply	msg_user	msg_id	msg_channel
2	ratheruponconjectureandsurmisethanonthatabsolutelogicalGRrFNbBuGjzg	0	tom	U070MSMGYCR	general
3	ofthesingularadventuresoftheGricePatersonsintheislandofGPxZJJP7PqxU	0	martin	U070Y0CNJL8	general
4	mayberemembered.SherlockHolmeswasable,bywindinguptheadeadGmIO2zxf7Vd5	0	tom	U070MSMGYCR	general
5	man'swatch,toprovethatithadbeenwoundduptwohoursbefore,andG7Ke5hhRwUgh	0	tom	U070MSMGYCR	general
6	thempresentsuchsingularfeaturesasthestrangetrainofGUX8bPchxdXB	0	tom	U070MSMGYCR	general
7	withthetext,andthesplashoftheraintolengthenoutintotheG2ligYGdBGtU	0	martin	U070Y0CNJL8	general
8	andforafewdayslwasadwelleroncemoreinmyoldquartersatGdZZJq0r0O8U	0	tom	U070MSMGYCR	general
9	GVOZ76HFJjdU	0	tom	U070MSMGYCR	general
10	astepinthepassageandatappingatthedoor.HestretchedouthisGiG36hKi1NI3	0	tom	U070MSMGYCR	general
11	well-groomedandtrimlyclad,withsomethingofrefinementandGpYrXaK5bDId	0	martin	U070Y0CNJL8	general
12	thehookandwillbedrypresently.YouhavecomeupfromtheGGo5QCwvjB0c	0	tom	U070MSMGYCR	general
13	GktdP5NhUlkR	0	tom	U070MSMGYCR	general
14	GYITDBVYzIJO	0	tom	U070MSMGYCR	general
15	Gd9LWbVnqGQg	0	martin	U070Y0CNJL8	general
16	leavingmepalpitatingwithhorror.ItookuptheenvelopeandsawGc0L8dQ4CzXf	0	martin	U070Y0CNJL8	general
17	other.GLsERkK4NFF0	0	tom	U070MSMGYCR	general
18	andsenddowntoFordham,theHorshamlawyer.'GZIOtilJTh69	0	tom	U070MSMGYCR	general
19	sorrytogiveyousuchatwo-edgedthing,butIcan'tsaywhatturnGhxnTVQwoyUN	0	tom	U070MSMGYCR	general
20	him.Thesingularincidentmade,asyoumaythink,thedeepestGtv0t4729evs	0	tom	U070MSMGYCR	general
21	sensationgrewlesskeenastheweekspassedandnothinghappenedtoGQRJhnbnzKsS	0	tom	U070MSMGYCR	general
22	disturbtheusualroutineofourlives.IcouldseeachangeinmyGGUQO1D4Bb8I	0	martin	U070Y0CNJL8	general
23	withthedoorlockeduponthesinside,butsometimeshewouldemergeGw5oLjxAbUB	0	martin	U070Y0CNJL8	general
24	inasortofdrunkenfrenzyandwouldburstoutofthehouseandtearGjh9PAqDK2xF	0	martin	U070Y0CNJL8	general
25	receipts,andaregister'writtenbeneath.These,wepresume,G4qFjlukWvjo	0	martin	U070Y0CNJL8	general
26	reconstructionoftheSouthernstates,andweremostlyconcernedwithGP3iFiiPpiQd	0	tom	U070MSMGYCR	general
27	fivedriedorangepipsintheoutstretchedpalmoftheotherone.HeGlaWDaEipvES	0	martin	U070Y0CNJL8	general
28	hometovisitanoldfriendofhis,MajorFreebody,whoisincommandGUA15ZAELBW2	0	tom	U070MSMGYCR	general
29	ofstrangershavingbeenseenuponthe roads.AndyetIneednottellGeDvRBTBHok7	0	tom	U070MSMGYCR	general
30					
31					
32					
33					
34					

Users | Channels | other | general | no-visit | small

Figure 25. Tool - output

The tool can be executed via the command line. The tools currently supports the following arguments:

- *–help* - describes how to use the tool
- *–input <file-path>* - allows to specify an inputfile. This argument is required.
- *–output <file-path>* - specifies an output file. The standard is output.xls
- *–filter_msg <search string>* - only returns messages, that include the search string
- *–filter_channel <channel name>* - returns only the messages from one channel
- *–filter_user <username>* - returns only information, regarding the specified user
- *–print_user* - prints all user information
- *–print_channel* - prints all channel information

[1]https://gitlab.cs.ttu.ee/fewasc/slackwebappforensic/-/tree/main/Analysis

6. Discussion

Based on the results gathered in the previous chapter, this chapter presents the interpretation of these results. It is intended to provide a deeper understanding of the severity of different artefacts and identify potential limitations.

6.1 Interpretation of the experimental results

Chapter 5 provided an overview of the different experiments that were conducted as well as the results. The following chapter discusses this result further. Furthermore, this chapter aims to contextualise the results of this work and its relevance.

This work aimed to identify the artefacts that could occur during the use of the Slack web application. During the experimentation phase, the first experiment detailed in Chapter 5.1 examined the possibilities for forensic analysts with current software tools. While the tools are not specific to web browser forensics, they allow for a variety of analyses suitable for most scenarios. Both tested tools focus on similar web artefacts, which mainly consist of history, cookies, caches, and bookmarks. In the case of the cache, the files recovered by Autopsy and BEC differ slightly compared to those recovered by the dedicated tool ChromeCacheView. In the case of the latter, different image and pdf files used on Slack that were not available in Autopsy and BEC could be recovered. Table 4 reveals that the amount of useful information from both tools is fairly limited. While BEC can recover some of the messages sent on Slack, this function is limited. Combined with the results of later experiments, neither tool seemed to utilise the IndexedDB or notifications as a source of information. BEC can partially parse the IndexedDB from Google Chrome, presumably because of the similarity to the installed Slack application. In general, this shows that IndexedDB and notifications are not yet seen as standard data sources by vendors. In the case of the IndexedDB, this behaviour seems to slowly change, as some vendors are currently implementing options to analyse the IndexedDB [60]. Support for tools analysing IndexedDB and notifications may be limited due to the lack of research on these sources.

With manual analysis of the images, the output of the forensic software could be confirmed. As summarised in Table 11, extensive information about user behaviour could be obtained based on the different sources. The most helpful source during an investigation would probably be IndexdDB. Inside the IndexedDB, a forensic investigator can find a wealth of information, such as the Slack workspace, the channel in the workspace, user information,

and messages. This result is comparable to the research on similar applications, such as Discord [23], [29], Cisco Webex [25], WhatsApp Web [24], [37], Instagram [38] and Microsoft Teams [41], [42]. Unlike similar works, Slack heavily uses external objects in the IndexedDB. Although Slack does not utilise the database feature of the IndexedDB, it stores the complete information as one blob in one entry of the database. This behaviour is not considered best practice when handling an IndexedDB [61]. The uniqueness in the implementation of Slack could therefore make it harder for forensic software to obtain the data.

Due to the scarce research available in the field of web browser notification and its high forensic value, further research is merited. To gain more insight, an experiment was conducted that covered different circumstances in both web browsers. Although Google Chrome does not seem to store messages that are not received via push API, some notifications are stored in Firefox. The results suggest that, in the case of Slack, the last message received from a channel or directly from a user can always be recovered. This may be because only the last message with the same title, for example, "New message in #random", is stored in the file. As demonstrated in Chapter 5.4, the notification could contain a variety of interesting artefacts and potentially even offer more insight into the active window during a certain time. However, more research is needed to determine what other value notifications can have on different websites.

Considering the limited amount of research on IndexedDB, especially in special circumstances, this was emphasised. Therefore, multiple experiments focused on the data stored in the IndexedDB were conducted. The experiments covered the number of messages stored in the IndexedDB as well as the potential limitations associated with a message written to the IndexedDB. The result of the series of experiments demonstrated that the messages were not always directly stored in the IndexedDB. In both web browsers, extracting the IndexedDB shortly after the message was received could not result in the retrieval of the latest messages. The results revealed that waiting for 10 minutes or refreshing the site could solve this. Furthermore, Chapter 5.3 experimentally demonstrated that the assumption that messages are only stored when a certain amount of message is written in a channel is wrong. The assumption was based on the idea that a small number of messages could be quickly retrieved from the server. The results showed that this is not the case. Furthermore, the experiment indicates that, depending on the size, more than 20,000 messages could potentially be stored in the IndexedDB. This number emphasises that the IndexedDB can contain a highly relevant amount of data. Being able to store such a high amount of data further increases the potential value of the IndexedDB during an investigation.

Other special circumstances considered were site logout, deleting browser data, and using private browsing mode. Those scenarios were not previously researched considering the IndexedDB. In each scenario, the IndexedDB was handled slightly differently by the web browser. When a user logged out of Slack, the indexed database was deleted via the JavaScript function. This explains the behaviour observed where the IndexedDB content is no longer openly available. As presented in Chapter 5.5, it was possible in both web browsers to recover the content of the IndexedDB anyway. This possibility is especially helpful because it allows an investigator to get insights into Slack, even if it is not possible to simply visit the website. Similar to logging out of Slack, the user could delete the browser data, which also does not allow the investigator to simply log into the website. However, as explained in Chapter 5.7, not only are the entries in the IndexdDB deleted, but this function also deletes all files that contain IndexedDB information. It is again possible to recover the most relevant files, so that the investigators can still recover the artefacts from the IndexedDB. Private browsing mode is another special circumstance when considering a forensic analysis of the web browser activity. The literature has suggested that artefacts cannot be recovered after the session is ended [28], [54], [55]. However, because these works did not include IndexedDB, the experiment was repeated to check the claim, and the results in the literature were confirmed. Similarly to the other artefacts, it was not possible to find artefacts stored in the IndexedDB after the private browsing session. The Mozilla bug notes [58] reveal that the IndexedDB is encrypted during the session, resulting in no recoverable artefacts.

The conducted research provided value by mainly exploring the different artefacts available specifically in the Slack web application. This provided further insights into the behaviour of the IndexedDB. The IndexedDB is currently an under-represented information source; it has only slowly received more attention in recent years. To aid the investigation of the Slack IndexedDB, a tool for easier parsing of the JSON file was developed and introduced in Chapter 5.9.

6.2 Answer to the research questions

In Chapter 1.1, the main research questions of the thesis were stated. Based on the previous chapter, they can be answered as follows.

RQ1: What artefacts remain on the device after using the Slack web application?

During the various experiments, a variety of different artefacts were identified. Based on the analysis conducted, the most interesting source of artefacts was the IndexedDB. Inside this file, a great percentage of messages and users can be recovered. Other interesting sources are the browser cache, browser history, and notifications. Table 11, presents an overview of the different actions in Slack and the sources from which they can be recovered.

RQ2: Are there differences in artefacts between different browsers and operating systems?

By exploring the differences between the different web browsers and operating systems, some differences were identified. Comparing the differences between the operating systems revealed no significant differences in the stored artefacts. The main difference was the different filepaths of the sources. Specifically, the difference between the browsers, Chrome and Firefox was more relevant. Here, especially, the different ways to store client-side data and notifications differ.

RQ3: Can artefacts be extracted in an automated way?

It was demonstrated that tools such as Autopsy and BEC can display some of the identified artefacts, which is useful for determining previous user activities on Slack. Sources, cache, client-side storage, and notification are only partially covered by these tools. However, tools like MZCacheView and ChromeCacheView proved that the cache can also be automatically extracted. The software tools to extract the content of the IndexedDB are, however, limited. For both browsers, there are Python implementations; for both, these are currently under development and slightly unstable. To further aid the investigation, a tool was developed that can extract valuable information from the extracted IndexedDB. Currently, no known software can extract information from notifications.

H0: It is possible to recover useful artefacts from the web browser after using the Slack web application. Considering the answers to research questions RQ1–RQ3, there are several artefacts left after using the Slack web application. In Chapter 5.2 the following

artefacts were identified:

- User information
- Messages
- Shared files
- User profile pictures
- Channel information

The main source of evidence is IndexedDB, which contains messages, user information, and channels. Depending on the web browser usage, using private modes or clearing browser data, for example, can affect the available artefacts.

6.3 Limitations of the approach

This work has demonstrated that numerous artefacts can offer insights into user behaviour on the Slack application. However, depending on the purpose and circumstances of an investigation, this approach may not be the best. The main limitation of the explored approach is private browsing. As established in the thesis, the use of private mode can prevent any data from being found on the hard drive. However, because the architecture protects private browsing, the user is not fully protected against data leakage. Therefore, it could be of interest to an investigator to search for login credentials. Another alternative can be to request the data directly from Slack. Both would allow the investigator to still be able to obtain the content.

Furthermore, the results of the research for IndexedDB may not be completely applicable to other websites using the same technology. Slack mainly uses the IndexedDB to store JSON as an external object. This results in only a few entries in the database. Other websites, for example, WhatsApp Web, create one entry for every message [24], [37]. Therefore, recovering data after the deletion of the browser data or the deletion of the IndexedDB could be limited.

6.4 Recommendation for the investigation of Slack

Based on the previous results, the following process is recommended for forensic investigations of user behaviour on Slack. Although the process depends heavily on the goal of the investigation, The proposed recommendations mainly apply to the investigation of one user. Investigations as part of larger e-discovery processes in organisations should be primarily based on the Slack Discovery API. The main reason is due to the large amount of resources

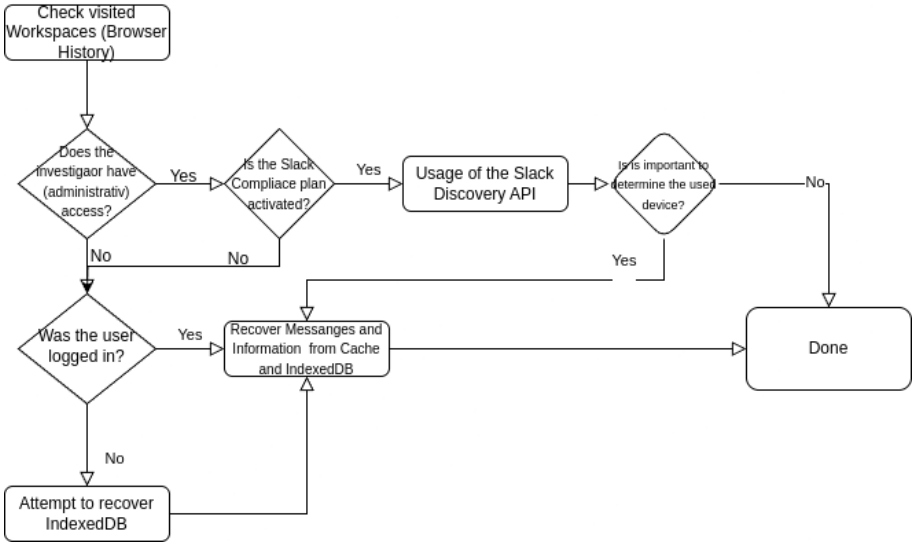


Figure 26. Flowchart - Recommend investigation path

needed for other approaches. An exception to this rule is if the e-discovery process is targeting a shadow IT infrastructure. As explained in the introduction, a nonorganisational hosted workspace could not easily utilise the Discovery API. A similar approach could be applied in law enforcement investigations: analysing the artefacts from Slack on a larger device could help secure a warrant for the entire workspace.

The recommendation based on the research conducted, in the case of one device, is shown in Figure 26. In general, the idea is to first identify the workspaces the user is/was using. Based on all the findings, the best possible way to analyse Slack content is through the Discovery API. Therefore, an analysis of the hard drive is often the second-best option during an investigation. When the Discovery API is not an option, the analysis of the hard drive can be performed to get an almost complete view of the activities. The Slack applications installed are expected to store more content. The analysis of the IndexedDB file, where the messages are stored, is similar to the web version.

If the user has been logged in during the image collection, the findings can be validated by visiting the website with their browser. However, a user could delete messages that could not be validated in this case.

7. Conclusion

In this thesis, the artefacts of the Slack web application were studied and analysed in different scenarios. During the research, around 20 different Slack workspaces were created, and approximately 60 different images were analysed. These images were the result of eight different experiments to elucidate the influence of different actions. Based on the research conducted, the following key findings can be summarised.

1. The use of the Slack web application allows an investigator to recover a wide variety of artefacts that could aid in an investigation. In the thesis, cookies, cache, browser history, notifications, and persistent storage were analysed. Although cookies did not provide deep insight into user behaviour, the other technologies did. The experiment in Chapter 5.2 revealed that the browser cache stored multiple different files, such as shared documents or profile pictures. The browser history not only provided timestamps but also more metadata. Some examples were when the user had a certain number of unread messages, with whom the user (potentially) talked, or which channel was visited. Notifications, when enabled, could provide more information about received messages, depending on the browser used. The most valuable source is potentially the IndexedDB, which stores all channel and user information, as well as all (public) messages in a workspace.
2. A second key finding is the difference between various operating systems and web browsers. While there are seemingly no great differences, except the file paths, between the operating systems, there are smaller differences between the web browsers. There was a known difference in the technology used to implement the indexed database. Another important difference is the handling of notifications. Although both browsers seem to store some information about notifications sent via PushAPI, no artefacts were found after using NotifcationAPI in Google Chrome.
3. A third key finding of the thesis is that Slack uses the IndexedDB uniquely. This behaviour differentiated the analysis of Slack from other applications, such as Microsoft Teams or WhatsApp Web. While those applications mainly store their data in unique entries in the IndexedDB, the Slack application only uses one entry to store all the data. Therefore, some behaviour of the IndexedDB could slightly differ from other applications, which needs to be considered during an investigation.
4. Although the focus is on the Slack application, the thesis also contributes two more general findings. First, it provides the thesis with greater insight into the possibilities of using the IndexedDB in forensic investigations. As Furkan Paligu demonstrated,

the IndexedDB provides forensic value. However, research on this is fairly limited. In this thesis, it was shown that the content of the IndexedDB can still be recovered. This applies after the deletion of the IndexedDB through the website or by clearing browser data. Those insights provide further valuable insights on the use of the IndexedDB in the field of web browser forensics.

5. Lastly, the thesis focused on the notifications emitted by the web browser. Although the research on notifications was limited, the notification exhibited great potential to gain more insight into user behaviour.

In conclusion, the thesis elucidates the artefacts left by the Slack web application. However, the value of the work depends on the circumstances of the investigation. As shown in Chapter 6.4, other approaches can be used to obtain the information.

7.1 Further Research

In this thesis, different artefacts that were created due to the usage of the Slack web application were discovered. During the research, multiple ideas that needed further research were identified. In the first step, research could be performed on additional web applications, which could be of potential value for investigation. Examples of those applications were, for example, Google Suite or Outlook. Especially interesting is the analysis of the IndexedDB usage of different applications. This thesis confirmed that the assessment of Furkan Paligu on IndexedDB is a valuable source of information [14], which is also true for Slack. To further evaluate the value of IndexedDB, further tools could be developed. During the experiments, it may be valuable to have a tool that can detect when the StorageAPI is called. This should be protocoled by noting what kind of information is added or deleted to the client-side storage. This could support further research to show more clearly which action results in which form of artefacts. In addition, in the area of the IndexedDB, further research in the field of encryption could provide significant value. The evaluation of other web applications, like Protonmail[1], seemed to only store encrypted data in the client-side storage. An overview of different methods to encrypt the data before storing it and an analysis of performance differences could be of further value for future web development. Although encryption could limit effectiveness in forensic investigations, this method could help protect user data.

Another interesting research area in the field of browser forensics is the analysis of notifications. Research in the area of notification is surprisingly limited. Although some research on notifications was done in this thesis, this research was limited to only one website. Here, it was established that the analysis of notifications can be especially helpful when other sources, such as cache or client-side storage, are not reliable.

Lastly, a comprehensive overview in the field of browser forensics for further operating systems like macOS, iOS, and Android could provide additional value. Specifically, research on the forensic value of client-side storage and notifications of those operating systems is currently lacking.

[1] https://mail.proton.me/

References

[1] Dominic Kent, *The State Of Workplace Messaging 2023: Microsoft Teams, Webex, & More*. [Online]. Available: https://dispatch.m.io/state-of-workplace-messaging/ (visited on 11/04/2023).

[2] David Storm, *How Uber was hacked — again*, Sep. 2022. [Online]. Available: https://blog.avast.com/uber-hack (visited on 11/16/2023).

[3] J. Starke, *The shadow it dilemma*, Mar. 2016. [Online]. Available: https://blogs.cisco.com/cloud/the-shadow-it-dilemma (visited on 02/17/2024).

[4] N. Vaidya, *Cloud forensics: Trends and challenges*, Sep. 2020. [Online]. Available: https://www.ijert.org/cloud-forensics-trends-and-challenges (visited on 02/22/2024).

[5] J. Dokko and M. Shin, "A digital forensic investigation and verification model for industrial espionage," in *Springer eBooks*. Dec. 2018, pp. 128–146. DOI: 10.1007/978-3-030-05487-8_7.

[6] G. S. Smith, "Computer forensics: Helping to achieve the auditor's fraud mission," *Journal of Forensic Accounting*, vol. 6, no. 1, pp. 119–134, 2005.

[7] K. Punwar, "Framework for analysis and forecasting on browser forensics," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 5, pp. 1994–1999, May 2019. DOI: 10.22214/ijraset.2019.5332.

[8] S. Mahaju and T. Atkison, "Evaluation of firefox browser forensics tools," ser. ACM SE '17, Kennesaw, GA, USA: Association for Computing Machinery, 2017, pp. 5–12, ISBN: 9781450350242. DOI: 10.1145/3077286.3077310.

[9] P. Anuradha, T. R. Kumar, and N. V. Sobhana, "Recovering deleted browsing artifacts from web browser log files in linux environment," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, 2016, pp. 1–4. DOI: 10.1109/CDAN.2016.7570957.

[10] A. Nalawade, S. Bharne, and V. Mane, "Forensic analysis and evidence collection for web browser activity," *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Sep. 2016. DOI: 10.1109/icacdot.2016.7877639.

[11] S. Mahaju and T. Atkison, "Evaluation of firefox browser forensics tools," *Proceedings of the SouthEast Conference*, Apr. 2017. DOI: 10.1145/3077286.3077310.

[12] G. Horsman, "I didn't see that! an examination of internet browser cache behaviour following website visits," *Digital Investigation*, vol. 25, pp. 105–113, Jun. 2018. DOI: 10.1016/j.diin.2018.02.006.

[13] T. Gros, R. Dirauf, and F. Freiling, "Systematic analysis of browser history evidence," *2020 13th International Conference on Systematic Approaches to Digital Forensic Engineering (SADFE)*, May 2020. DOI: 10.1109/sadfe51007.2020.00010.

[14] Furkan Paligu, *Investigation of IndexedDB persistent storage for digital forensics*, Aug. 2022. [Online]. Available: https://shsu-ir.tdl.org/server/api/core/bitstreams/af90f219-26e7-4ec8-97b2-dbcb46b8c23b/content.

[15] Vincent Lo, "Google Chrome Notification Analysis in Depth," *GIAC*, Jul. 2021. [Online]. Available: https://www.giac.org/paper/gcih/20579/google-chrome-notification-analysis-in-depth/128522.

[16] H. Luo, H. Jiang, Zhichao Yan, and Yaodong Yang, "Fast transaction logging for smartphones," in *2016 32nd Symposium on Mass Storage Systems and Technologies (MSST)*, Santa Clara, CA: IEEE, 2016, pp. 1–5, ISBN: 978-1-4673-9055-2. DOI: 10.1109/MSST.2016.7897094.

[17] Alex Caithness, "Hang on! That's not SQLite! Chrome, Electron and LevelDB," Sep. 2020. [Online]. Available: https://www.cclsolutionsgroup.com/post/hang-on-thats-not-sqlite-chrome-electron-and-leveldb (visited on 04/03/2024).

[18] C. Group, *Cclgroupltd/ccl_chrome_indexeddb: Python re-implementations of chrome-esque applications.* [Online]. Available: https://github.com/cclgroupltd/ccl_chrome_indexeddb (visited on 04/21/2024).

[19] NtNinja. [Online]. Available: https://gitlab.com/ntninja/moz-idb-edit (visited on 04/21/2024).

[20] WHATWG, Apr. 2024. [Online]. Available: https://notifications.spec.whatwg.org/ (visited on 03/25/2024).

[21] MozDevNet, *Push api - web apis: Mdn*, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Push_API?retiredLocale=de (visited on 04/02/2024).

[22] Slack, *Slack Features*, englisch. [Online]. Available: https://slack.com/features/ (visited on 03/09/2024).

[23] K. Gupta, C. Varol, and B. Zhou, "Digital forensic analysis of discord on google chrome," en, *Forensic Science International: Digital Investigation*, vol. 44, p. 301 479, Mar. 2023, ISSN: 26662817. DOI: 10.1016/j.fsidi.2022.301479.

[24] F. Paligu and C. Varol, "Browser Forensic Investigations of WhatsApp Web Utilizing IndexedDB Persistent Storage," en, *Future Internet*, vol. 12, no. 11, p. 184, Oct. 2020, ISSN: 1999-5903. DOI: 10.3390/fi12110184.

[25] Z. Khalid, F. Iqbal, F. Kamoun, L. A. Khan, and B. Shah, "Forensic investigation of Cisco WebEx desktop client, web, and Android smartphone applications," en, *Annals of Telecommunications*, vol. 78, no. 3-4, pp. 183–208, Apr. 2023, ISSN: 0003-4347, 1958-9395. DOI: 10.1007/s12243-022-00919-6.

[26] J. Oh, S. Lee, and S. Lee, "Advanced evidence collection and analysis of web browser activity," en, *Digital Investigation*, vol. 8, S62–S70, Aug. 2011, ISSN: 17422876. DOI: 10.1016/j.diin.2011.05.008.

[27] A. Nalawade, S. Bharne, and V. Mane, "Forensic analysis and evidence collection for web browser activity," in *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*, Pune, India: IEEE, Sep. 2016, pp. 518–522, ISBN: 978-1-5090-2080-5. DOI: 10.1109/ICACDOT.2016.7877639.

[28] G. Horsman, B. Findlay, J. Edwick, *et al.*, "A forensic examination of web browser privacy-modes," en, *Forensic Science International: Reports*, vol. 1, p. 100 036, Nov. 2019, ISSN: 26659107. DOI: 10.1016/j.fsir.2019.100036.

[29] M. Davis, B. McInnes, and I. Ahmed, "Forensic investigation of instant messaging services on linux OS: Discord and Slack as case studies," en, *Forensic Science International: Digital Investigation*, vol. 42, p. 301 401, Jul. 2022, ISSN: 26662817. DOI: 10.1016/j.fsidi.2022.301401.

[30] Joe Pochron, *Need to Collect Data from Slack? Read this First.* English, Mar. 2018. [Online]. Available: https://www.transperfect.com/blog/need-to-collect-data-from-slack--read-this-first (visited on 11/06/2023).

[31] R. Hager, *That slack email you just got asking to reset your password is legit, not a scam*, Feb. 2021. [Online]. Available: https://www.androidpolice.com/2021/02/05/that-slack-email-you-just-got-asking-to-reset-your-password-is-legit-not-a-scam/ (visited on 05/05/2024).

[32] C. Page, *Slack urges users to reset passwords after android bug potentially exposed credentials*, Feb. 2021. [Online]. Available: https://www.forbes.com/sites/carlypage/2021/02/11/slack-urges-users-to-reset-passwords-after-android-bug-potentially-exposed-credentials/ (visited on 05/05/2024).

[33] N. Statt, *Slack quickly removes message invites in its new dm feature over harassment concerns*, Mar. 2021. [Online]. Available: https://www.theverge.com/2021/3/24/22348743/slack-connect-dm-abuse-harassment-disable-message-invite-response (visited on 05/05/2024).

[34] R. Carter, *Is slack encrypted? slack encryption in 2023*, Sep. 2023. [Online]. Available: https://www.uctoday.com/collaboration/is-slack-encrypted-slack-encryption-in-2023/ (visited on 05/05/2024).

[35] Slack, *Security at slack*, 2019. [Online]. Available: https://a.slack-edge.com/80588/marketing/downloads/security/Security_White_Paper_2019.pdf (visited on 03/17/2024).

[36] Y. Chen, Y. Gao, N. Ceccio, R. Chatterjee, K. Fawaz, and E. Fernandes, "Experimental security analysis of the app model in business collaboration platforms," in *31st USENIX Security Symposium (USENIX Security 22)*, Boston, MA: USENIX Association, Aug. 2022, pp. 2011–2028, ISBN: 978-1-939133-31-1. [Online]. Available: https://www.usenix.org/conference/usenixsecurity22/presentation/chen-yunang-experimental.

[37] Y. Salem, M. Owda, and A. Y. Owda, "An experimental approach for locating WhatsApp digital forensics artefacts on Windows 10 and the cloud," en, *International Journal of Electronic Security and Digital Forensics*, vol. 15, no. 3, p. 281, 2023, ISSN: 1751-911X, 1751-9128. DOI: 10.1504/IJESDF.2023.130662.

[38] F. Paligu and C. Varol, "Browser Forensic Investigations of Instagram Utilizing IndexedDB Persistent Storage," en, *Future Internet*, vol. 14, no. 6, p. 188, Jun. 2022, ISSN: 1999-5903. DOI: 10.3390/fi14060188.

[39] T. Pandela and I. Riadi, "Browser Forensics on Web-based Tiktok Applications," *International Journal of Computer Applications*, vol. 175, no. 34, pp. 47–52, Dec. 2020, ISSN: 09758887. DOI: 10.5120/ijca2020920897.

[40] M. A. H. B. Azhar, J. Timms, and B. Tilley, "Forensic Investigations of Google Meet and Microsoft Teams – Two Popular Conferencing Tools in the Pandemic," en, in *Digital Forensics and Cyber Crime*, P. Gladyshev, S. Goel, J. James, G. Markowsky, and D. Johnson, Eds., vol. 441, Cham: Springer International Publishing, 2022, pp. 20–34, ISBN: 978-3-031-06364-0 978-3-031-06365-7. DOI: 10.1007/978-3-031-06365-7_2.

[41] H. R. Bowling, "A Forensic Analysis of Microsoft Teams," 12454516 Bytes, 2021, Artwork Size: 12454516 Bytes Publisher: Purdue University Graduate School. DOI: 10.25394/PGS.15091329.V1.

[42] F. Paligu and C. Varol, "Microsoft Teams desktop application forensic investigations utilizing IndexedDB storage," en, *Journal of Forensic Sciences*, vol. 67, no. 4, pp. 1513–1533, Jul. 2022, ISSN: 0022-1198, 1556-4029. DOI: 10.1111/1556-4029.15014.

[43] F. Paligu, A. Kumar, H. Cho, and C. Varol, "BrowStExPlus: A Tool to Aggregate Indexed Artifacts for Forensic Analysis," en, *Journal of Forensic Sciences*, vol. 64, no. 5, pp. 1370–1378, Sep. 2019, ISSN: 0022-1198, 1556-4029. DOI: 10.1111/1556-4029.14043.

[44] N. D. W. Cahyani, D. S. Pratama, and N. H. A. Rahman, "Proactive Acquisition using Bot on Discord," en, *International Journal of Advanced Computer Science and Applications*, vol. 14, no. 5, 2023, ISSN: 21565570, 2158107X. DOI: 10.14569/IJACSA.2023.0140533.

[45] A. Mendoza, A. Kumar, D. Midcap, H. Cho, and C. Varol, "BrowStEx: A tool to aggregate browser storage artifacts for forensic analysis," en, *Digital Investigation*, vol. 14, pp. 63–75, Sep. 2015, ISSN: 17422876. DOI: 10.1016/j.diin.2015.08.001.

[46] Hevner, March, H.-S. Park, and Ram, "Design science in information systems research," *MIS Quarterly*, vol. 28, no. 1, pp. 75–75, Jan. 2004. DOI: https://doi.org/10.2307/25148625.

[47] R. Montasari, V. Carpenter, and R. Hill, "A road map for digital forensics research: A novel approach for establishing the design science research process in digital forensics," *International Journal of Electronic Security and Digital Forensics*, vol. 11, no. 2, pp. 194–224, 2019. DOI: 10.1504/IJESDF.2019.10018777.

[48] Cook, Thomas D. and Campbell, D. T., "The design and conduct of true experiments and quasi-experiments in field settings," English, in *Reproduced in part in Research in Organizations: Issues and Controversies*, Mowday, R. T. and Steers, R. M., Eds., Goodyear Publishing Company, 1979.

[49] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, pp. 1–22, Jan. 2013. DOI: 10.1155/2013/496701.

[50] Google, *Leveldb/doc/impl.md at main google/leveldb*. [Online]. Available: https://github.com/google/leveldb/blob/main/doc/impl.md (visited on 04/21/2024).

[51] MozDevNet, *Notification.timestamp property - web apis: Mdn*, 2024. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/API/Notification/timestamp (visited on 04/02/2024).

[52] P. Anuradha, T. R. Kumar, and N. V. Sobhana, "Recovering deleted browsing artifacts from web browser log files in Linux environment," in *2016 Symposium on Colossal Data Analysis and Networking (CDAN)*, Indore, Madhya Pradesh, India: IEEE, Mar. 2016, pp. 1–4, ISBN: 978-1-5090-0669-4. DOI: 10.1109/CDAN.2016.7570957.

[53] D. Pawlaszczyk and C. Hummert, "Making the Invisible Visible – Techniques for Recovering Deleted SQLite Data Records," en, *International Journal of Cyber Forensics and Advanced Threat Investigations*, vol. 1, no. 1-3, pp. 27–41, Feb. 2021, ISSN: 27539997. DOI: 10.46386/ijcfati.v1i1-3.17.

[54] X. Fernandez-Fuentes, T. F Pena, and J. C. Cabaleiro, "Digital forensic analysis methodology for private browsing: Firefox and Chrome on Linux as a case study," en, *Computers & Security*, vol. 115, p. 102 626, Apr. 2022, ISSN: 01674048. DOI: 10.1016/j.cose.2022.102626.

[55] Y. Kathiravan, M. F. Mohamad Amran, N. A. Mat Razali, *et al.*, "A study on private browsing in windows environment," *Journal of Defence Science, Engineering & Technology*, vol. 3, no. 1, Jun. 2020, ISSN: 27735281. DOI: 10.58247/jdset-2020-0301-03.

[56] K. Satvat, M. Forshaw, F. Hao, and E. Toreini, "On the privacy of private browsing – A forensic approach," en, *Journal of Information Security and Applications*, vol. 19, no. 1, pp. 88–100, Feb. 2014, ISSN: 22142126. DOI: 10.1016/j.jisa.2014.02.002.

[57] N. Tsalis, A. Mylonas, A. Nisioti, D. Gritzalis, and V. Katos, "Exploring the protection of private browsing in desktop browsers," en, *Computers & Security*, vol. 67, pp. 181–197, Jun. 2017, ISSN: 01674048. DOI: 10.1016/j.cose.2017.03.006.

[58] *Bugzilla - bug 1639542*, May 2020. [Online]. Available: https://bugzilla.mozilla.org/show_bug.cgi?id=1639542 (visited on 04/21/2024).

[59] G. Horsman and D. Errickson, "When finding nothing may be evidence of something: Anti-forensics and digital tool marks," en, *Science & Justice*, vol. 59, no. 5, pp. 565–572, Sep. 2019, ISSN: 13550306. DOI: 10.1016/j.scijus.2019.06.004.

[60] Foxton, Jan. 2024. [Online]. Available: https://www.foxtonforensics.com/blog/post/investigating-microsoft-teams-indexeddb-data (visited on 04/21/2024).

[61] P. Walton, *Best practices for using indexeddb*, Jun. 2017. [Online]. Available: https://web.dev/articles/indexeddb-best-practices (visited on 04/21/2024).

Appendix 1 – Non-Exclusive License for Reproduction and Publication of a Graduation Thesis[1]

I Felix Waschke,

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Forensic analysis of the Slack web application", supervised by Pavel Tšikul
   1.1. to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;
   1.2. to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.
2. I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.
3. I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

12.05.2024

[1]The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.

Appendix 2 - JSON structure IndexedDB

User


```
"U06M6GQFT6U":{
       "files":[
       ],
       "activity":[
       ],
       "stars":[
       ],
       "mentions":[
       ],
       "id":"U06M6GQFT6U",
       "team_id":"T06M6GNN33J",
       "name":"Laura",
       "deleted":false,
       "color":"e7392d",
       "real_name":"Laura",
       "tz":"Europe/Athens",
       "tz_label":"Eastern␣European␣Time",
       "tz_offset":7200,
       "profile":{
         "title":"Sales",
         "phone":"",
         "skype":"",
         "real_name":"Laura",
         "real_name_normalized":"Laura",
         "display_name":"Laura",
         "display_name_normalized":"Laura",
         "fields":"None",
         "status_text":"",
         "status_emoji":"",
         "status_expiration":0,
         "avatar_hash":"5a2cad10b204",
         "image_original":"https://s3-us-west-2.amazonaws.com/slack-
            files2/avatars/2024-02-25/6683224580599
            _5a2cad10b204caf64ad7_original.jpg",
         "is_custom_image":true,
         "email":"Laura@*****.*****.me",
         "first_name":"Laura",
         "last_name":"",
         "status_text_canonical":"",
```

```
      "team":"T06M6GNN33J",
      "statusEmojiDisplayInfo":{
      },
   "image_24":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-24",
   "image_32":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-32",
   "image_48":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-48",
   "image_72":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-72",
   "image_192":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-192",
   "image_512":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-512",
   "image_1024":"https://ca.slack-edge.com/T06M6GNN33J-U06M6GQFT6U-5
      a2cad10b204-1024",
   "always_active":false
    },
    "is_admin":false,
    "is_owner":false,
    "is_primary_owner":false,
    "is_restricted":false,
    "is_ultra_restricted":false,
    "is_bot":false,
    "is_app_user":false,
    "updated":1708882530.0,
    "is_email_confirmed":true,
    "who_can_share_contact_card":"EVERYONE",
    "is_stranger":false,
    "member_color":"e7392d",
    "is_invited_user":false,
    "isExternal":false,
    "isUnknown":false,
    "isNonExistent":false,
    "_name_lc":"Laura",
    "_first_name_lc":"Laura",
    "_last_name_lc":"",
    "_real_name_lc":"Laura",
    "_real_name_normalized_lc":"Laura",
    "_display_name_lc":"Laura",
    "_display_name_normalized_lc":"Laura",
    "_email_normalized_lc":"Laura@*****.*****.me"
}
```

Message


```
"1708883657.137899":{
         "thread_ts":"<Undefined>",
         "slackbot_feels":"None",
         "_hidden_reply":false,
         "reply_count":0,
         "replies":"<Undefined>",
         "latest_reply":"<Undefined>",
         "reply_users":"<Undefined>",
         "reply_users_count":"<Undefined>",
         "files":"<Undefined>",
         "attachments":"<Undefined>",
         "blocks":[
           {
             "type":"rich_text",
             "block_id":"xNSkC",
             "elements":[
               {
                 "type":"rich_text_section",
                 "elements":[
                   {
                     "type":"text",
                     "text":"Hello,␣thrilled␣to␣join␣the␣team!"
                   }
                 ]
               }
             ]
           }
         ],
         "blocksProcessed":[
           {
             "type":"rich_text",
             "blockId":"xNSkC",
             "elements":[
               {
                 "type":"rich_text_section",
                 "elements":[
                   {
                     "type":"text",
                     "text":"Hello,␣thrilled␣to␣join␣the␣team!"
                   }
                 ]
               }
             ]
           }
```

```
  ],
  "client_msg_id":"1b316906-f17b-4abb-bb0a-200093f14c46",
  "edited":{
    "user":"U06L372MCG7",
    "ts":"1708883836.000000"
  },
  "type":"message",
  "ts":"1708883657.137899",
  "channel":"C06LL70GJ20",
  "no_display":false,
  "user":"U06L372MCG7",
  "_rxn_key":"message-1708883657.137899-C06LL70GJ20",
  "subtype":"<Undefined>",
  "text":"Hello, thrilled to join the team!",
  "__meta__":{
    "lastUpdatedTs":"6474.800000000003"
  }
}
```

Channel


```
C06LHNY162E":{
␣␣␣␣␣␣␣␣␣"id":"C06LHNY162E",
␣␣␣␣␣␣␣␣␣"name":"meme",
␣␣␣␣␣␣␣␣␣"is_channel":true,
␣␣␣␣␣␣␣␣␣"is_group":false,
␣␣␣␣␣␣␣␣␣"is_im":false,
␣␣␣␣␣␣␣␣␣"is_mpim":false,
␣␣␣␣␣␣␣␣␣"is_private":false,
␣␣␣␣␣␣␣␣␣"created":1708882326.0,
␣␣␣␣␣␣␣␣␣"is_archived":false,
␣␣␣␣␣␣␣␣␣"is_general":false,
␣␣␣␣␣␣␣␣␣"unlinked":0,
␣␣␣␣␣␣␣␣␣"name_normalized":"meme",
␣␣␣␣␣␣␣␣␣"is_frozen":false,
␣␣␣␣␣␣␣␣␣"is_org_shared":false,
␣␣␣␣␣␣␣␣␣"is_pending_ext_shared":false,
␣␣␣␣␣␣␣␣␣"context_team_id":"T06M6GNN33J",
␣␣␣␣␣␣␣␣␣"updated":1708883900189.0,
␣␣␣␣␣␣␣␣␣"parent_conversation":"None",
␣␣␣␣␣␣␣␣␣"creator":"U06LERBF129",
␣␣␣␣␣␣␣␣␣"use_case":"project",
␣␣␣␣␣␣␣␣␣"is_ext_shared":false,
␣␣␣␣␣␣␣␣␣"topic":{
␣␣␣␣␣␣␣␣␣␣␣␣"value":"Have fun here!",
␣␣␣␣␣␣␣␣␣␣␣␣"creator":"U06M6GQFT6U",
␣␣␣␣␣␣␣␣␣␣␣␣"last_set":1708883900.0
␣␣␣␣␣␣␣␣␣},
␣␣␣␣␣␣␣␣␣"purpose":{
␣␣␣␣␣␣␣␣␣␣␣␣"value":"This channel is for everything #meme. Hold
    meetings, share docs,
          and make decisions together with your team.",
␣␣␣␣␣␣␣␣␣␣␣␣"creator":"U06LERBF129",
␣␣␣␣␣␣␣␣␣␣␣␣"last_set":1708882326.0
␣␣␣␣␣␣␣␣␣},
␣␣␣␣␣␣␣␣␣"previous_names":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"is_member":true,
␣␣␣␣␣␣␣␣␣"is_org_mandatory":false,
␣␣␣␣␣␣␣␣␣"is_org_default":false,
␣␣␣␣␣␣␣␣␣"_name_lc":"meme",
␣␣␣␣␣␣␣␣␣"_show_in_list_even_though_no_unreads":false,
␣␣␣␣␣␣␣␣␣"scroll_top":0,
␣␣␣␣␣␣␣␣␣"history_is_being_fetched":false,
```

```
␣␣␣␣␣␣␣␣␣"unread_highlight_cnt":0,
␣␣␣␣␣␣␣␣␣"unread_highlights":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"unread_cnt":0,
␣␣␣␣␣␣␣␣␣"unreads":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"has_fetched_history_after_scrollback":false,
␣␣␣␣␣␣␣␣␣"oldest_msg_ts":"None",
␣␣␣␣␣␣␣␣␣"internalTeamIds":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"connectedLimitedTeamIds":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"connectedTeamIds":[

␣␣␣␣␣␣␣␣␣],
␣␣␣␣␣␣␣␣␣"isNonExistent":false,
␣␣␣␣␣␣␣␣␣"isUnknown":false,
␣␣␣␣␣␣␣␣␣"fromAnotherTeam":false
␣␣␣␣␣␣}
```

Appendix 3 - Regex parsing IndexedDB

Regex - Message

```
"\d{10}\.\d{6}":\{("thread_ts":"[\d\.]+",)?"slackbot_feels":[^,]+,"
    _hidden_reply":[^,]+,"reply_count":(?P<msg_reply>\d+),("
    latest_reply":"[\d\.]+",)?("reply_users":\["\w+"\],)?("
    reply_users_count":\d+,)?"blocks":\[\{"type":"rich_text","block_id":
    "[^,"]+","elements":\[\{"type":"rich_text_section","elements":\[\{"
    type":"text","text":"(?P<msg_text>[^"]+)"\}\]\}\]\}\]\}\],"
    blocksProcessed":\[\{"type":"rich_text","blockId":"[^,"]+","
    elements":\[\{"type":"rich_text_section","elements":\[\{"type":"
    text","text":"[^"]+"\}\]\}\]\}\],"client_msg_id":"[^"]+",("edited"
    :{"user":"(\w{11})","ts":"[\d\.]+"},)?("source_team_id":"[^"]+",)?"
    type":"message","ts":"[^"]+","channel":"(?P<msg_channel>[^,"]+)","
    no_display":[^,]+,"user":"(?P<msg_user>[^,"]+)",("subtype":"\w+",)?
    "_rxn_key":"[^"]+","text":"[^_]+__meta__":\{"lastUpdatedTs":"[\d\.]+
    "\}(,"is_locked":\w+)?\}
```

Regex - Channel

```
"(?P<channel_id>\w{11})":\{"id":"[^"]+","name":"(?P<channel_name>[^"]+)
    ","is_channel":[^,]+,"is_group":[^,]+,"is_im":[^,]+,"is_mpim":[^,]+,
    "is_private":[^,]+,"created":(?P<channel_time_created>\d+),"
    is_archived":[^,]+,"is_general":[^,]+,"unlinked":\d+,"
    name_normalized":"[^"]+","is_frozen":[^,]+,"is_org_shared":[^,]+,"
    is_pending_ext_shared":[^,]+,"context_team_id":"[^"]+","updated"
    :[^,]+,"parent_conversation":[^,]+,"creator":"(?P<channel_creator
    >[^"]*)",("use_case":"[^"]*",)?"is_ext_shared":[^,]*,"topic":\{"
    value":"(?P<channel_topic_value>[^"]*)","creator":"[^"]*","last_set
    ":\d+\},"purpose":\{"value":"(?P<channel_purpose_value>[^"]*)","
    creator":"[^"]*","last_set":\d+\},("properties":\{"use_case":"[^"]*
    "\},)?"previous_names":\[[^\]]*\],"is_member":[^,]+,("is_read_only"
    :[^,]+,)?("is_thread_only":[^,]+,)?("is_non_threadable":[^,]+,)?("
    unread_count_display":\d+,)?"is_org_mandatory":[^,]+,"
    is_org_default":[^,]+,"_name_lc":"[^"]+","
    _show_in_list_even_though_no_unreads":[^,]+,"scroll_top":[^,]+,"
    history_is_being_fetched":[^,]+,"unread_highlight_cnt":[^,]+,"
    unread_highlights":\[[^]]*\],"unread_cnt":[^,]+,"unreads":\[[^]]*\],
    "has_fetched_history_after_scrollback":[^,]+,"oldest_msg_ts":[^,]+,
    "internalTeamIds":\[[^]]*\],"connectedLimitedTeamIds":\[[^]]*\],"
    connectedTeamIds":\[[^]]*\],"isNonExistent":[^,]+,"isUnknown":[^,]+,
    "fromAnotherTeam":[^}]+\}
```

Regex - User

```
"(?P<user_id>\w{11})":\{"files":\[[^\]]*\],"activity":\[[^\]]*\],"stars
    ":\[[^\]]*\],"mentions":\[[^\]]*\],"id":"\w{11}","team_id":"\w{11}",
    "name":"(?P<user_name>[^"]+)","deleted":[^,]+,"color":"[^"]+","
    real_name":"[^"]*","tz":"[^"]+","tz_label":"[^"]+","tz_offset":\d*,
    "profile":\{"title":"[^"]*","phone":"[^"]*","skype":"[^"]*","
    real_name":"[^"]*","real_name_normalized":"[^"]*","display_name":"
    [^"]*","display_name_normalized":"[^"]*","fields":\{[.]*\},"
    status_text":"[^"]*","status_emoji":"[^"]*","status_expiration":\d+,
    "avatar_hash":"[^"]*","email":"(?P<User_email>[^"]+)","first_name":
    "[^"]*","last_name":"[^"]*","status_text_canonical":"[^"]*","team":
    "[^"]*","statusEmojiDisplayInfo":\{[^}]*\},[^}]*,"is_custom_image"
    :[^,]*,"always_active":[^}]*\},"is_admin":(?P<is_admin>[^,]*),"
    is_owner":(?P<is_owner>[^,]*),"is_primary_owner":[^,]*,"
    is_restricted":[^,]*,"is_ultra_restricted":[^,]*,"is_bot":[^,]*,"
    is_app_user":[^,]*,"updated":\d+,"is_email_confirmed":[^,]*,"
    who_can_share_contact_card":"[^"]*","is_stranger":[^,]*,"
    member_color":"[^"]*","is_invited_user":[^,]*,"isExternal":[^,]*,"
    isUnknown":[^,]*,"isNonExistent":[^,]*,"_name_lc":"[^"]*","
    _first_name_lc":"[^"]*","_last_name_lc":"[^"]*","_real_name_lc":"[^
    "]*","_real_name_normalized_lc":"[^"]*","_display_name_lc":"[^"]*",
    "_display_name_normalized_lc":"[^"]*","_email_normalized_lc":"[^"]*
    "(,"is_self":[^,]*)?(,"manual_presence":"[^"]*")?(,"first_login":\d
    *)?\}
```