future internet

MDPI

Article

Browser Forensic Investigations of WhatsApp Web Utilizing IndexedDB Persistent Storage

Furkan Paligu and Cihan Varol *

Computer Science Department, Sam Houston State University, Huntsville, TX 77340, USA; fxp017@shsu.edu
* Correspondence: cxv007@shsu.edu; Tel.: +1-936-294-3930

Received: 10 October 2020; Accepted: 21 October 2020; Published: 28 October 2020

Abstract: Digital Evidence is becoming an indispensable factor in most legal cases. However, technological advancements that lead to artifact complexity, are forcing investigators to create sophisticated connections between the findings and the suspects for admissibility of evidence in court. This paper scrutinizes whether IndexedDB, an emerging browser technology, can be a source of digital evidence to provide additional and correlating support for traditional investigation methods. It particularly focuses on the artifacts of the worldwide popular application, WhatsApp. A single case pretest–posttest quasi experiment is applied with WhatsApp Messenger and Web Application to populate and investigate artifacts in IndexedDB storage of Google Chrome. The findings are characterized and presented with their potential to be utilized in forensic investigation verifications. The storage locations of the artifacts are laid out and operations of extraction, conversion and presentation are systematized. Additionally, a proof of concept tool is developed for demonstration. The results show that WhatsApp Web IndexedDB storage can be employed for time frame analysis, demonstrating its value in evidence verification.

Keywords: digital forensics; persistent storage; web browser forensics

1. Introduction

Digital evidence has been a major contributor to the decision making of many important cases over the past few decades [1]. Signature cases such as the murder trials of Casey Anthony [2] and Christian Aguilar [3], further indicate the strong dependency on digital evidence as the cases' final decisions were made based on digital evidence. As technology keeps developing, more cases will be dependent on digital evidence.

Unfortunately, the defense against digital evidence today is very sophisticated and its admissibility is becoming a concerning issue [4]. Today, the presence of hard evidence on a suspect's computer is not accepted as a sufficient determinant for a verdict. As the volume and complexity of digital technology increases, so does the responsibility of digital investigators to connect evidence to suspects.

A strong method to support the link of the evidence to the suspect is to back up the claim with matching information from independent sources. For instance, a suspect's access to a particular file, and extensive search records indicative of the same file's scrutinization on the internet can create correlating evidence chains. The timelines of creation, alteration, and last access of a sensitive file can indicate inconsistencies in the Windows Registry and browser's downloaded files information repository. Additionally, information from supplementary sources can be used to verify abnormalities in time frames indicating possible tampering of evidence.

Newly emerging technologies are good opportunities to support traditional evidence with additional information. As new technologies rely on different storage and processing techniques, tempering with their origins requires extra effort on the suspect's part. Therefore, a cross confirmation with additional sources serves as an extra layer of credibility for the findings. An obfuscation or change in the traditional sources is likely to create an inconsistency of information with alternative sources.

IndexedDB is a newly developed NoSQL (Not Only SQL) transactional database system that enables fast access to persistent data through JSON (JavaScript Object Notation) objects [5]. It can be operated through JavaScript code which makes it highly usable for web browsers. A profound advantage of IndexedDB lies in its storage capacity. IndexedDB offers a large storage area starting from 50 MB of data for each origin while its competitor Web Storage technology can store maximum 5 MB of data [6,7]. This advantage enabled IndexedDB to increase its utilization drastically in popular websites in the last few years. As many websites begin to use IndexedDB, increasingly it appears as a candidate to be a supporting source to traditional digital investigations.

This paper scrutinizes the prospect of IndexedDB storage for the digital forensic investigations of the WhatsApp Web Application. In this scope, it raises two research questions:

- Is it possible that the artifacts from the information stored by IndexedDB technology for the WhatsApp Web Application carry significant value for forensic investigations?
- Is the stored information suited to be utilized for effective time frame analysis during forensic investigations?

By answering these questions, this research investigates the hypothesis that IndexedDB storage carries forensically significant artifacts for the WhatsApp Web Application. These artifacts can be used to create time frame analyses in forensic investigations. The following sections share the previous work on IndexedDB forensics and security, provide a background of IndexedDB and LevelDB technologies, explain the methodology for this research, present the findings with a proof-of-concept tool BrowSwEx, and put forward a conclusion and possible future work.

2. Related Works

While several research studies have scrutinized the applicability of forensic techniques and tools on the WhatsApp mobile application [8–11] and its network investigations [12], fewer studies have focused on WhatsApp Web forensics [13,14]. Additionally, several of the research studies with a broader scope of instant messaging applications touched upon forensic analysis of WhatsApp [15,16]. However, their focus was not on the forensic investigations of persistent storage of IndexedDB, which is a relatively new subject area for digital forensics [6,17,18].

Anglano [8] investigated the artifacts left on Android devices by the WhatsApp Messenger application. The author considered all the artifacts left by WhatsApp Messenger and demonstrated a correlation of information merged from various artifacts. The acquired data reflected the activities of adding/deleting a contact, sending messages, and the feedback on the delivery of the messages. The scope of this research was limited to WhatsApp Messenger on Android devices, while WhatsApp Web and iOS systems were not addressed.

Thakur [9] provided an outline of a methodology for forensic investigations of WhatsApp on Android phones. The research covered volatile memory acquisition as an addition to static external storage acquisition. The research utilized the WhatsAppXtract tool on the SQLite database in the backed-up folder without rooting the android device for non-volatile memory acquisition. Whereas, a memory dump utility, Memfetch, was used for the extraction of information from the volatile memory. A tool called whatsappRamXtract was introduced for the analysis and the presentation of the extracted data. It was argued that the data analyzed was critical and forensically significant.

Shortall and Azhar [10] inspected the WhatsApp mobile application for forensic investigations on iOS 8.3, Android Lollipop 5.0.1, and Windows Phone 8.1 utilizing the forensic investigation tools of EnCase, UFED (Universal Forensic Extraction Device), and Oxygen Forensic Suite. The authors explicated that contact information was recovered from iOS and Android with Universal Forensic Extraction Device and Oxygen Forensic Suite. However, nothing was recovered from Windows Phone 8.1. It is discussed in the paper that the security mechanisms employed by Windows Phone 8.1 made it very difficult to perform a traditional forensic acquisition. As a result, the paper suggested live data forensic investigations to recover WhatsApp artifacts from Windows Phone 8.1 operating systems that were not accessible with traditional techniques.

Umar et al. [11] used NIST (National Institute of Standards and Technology) measurement parameters to evaluate the forensic investigation tools specializing in Android phones to compare their performances on WhatsApp forensics. Nine core assertions, seven optional assertions along with six core requirements, and optional feature requirements of NIST measurement parameters were utilized in the scope of the work. The research identified Belkasoft Evidence as the tool that scored the best among other tools, whereas WhatsApp Key/DB Extractor came out as the most cost-efficient, and Oxygen Forensic as the best at obtaining the artifacts.

Karpisek et al. [12] conducted research on the decryption of the information transferred during WhatsApp calls. The procedure included decryption of network traffic, obtaining artifacts of the call, and the development of a Python tool to convert obtained Wireshark dump files to HTML format. The research stated the artifacts can be obtained from the WhatsApp calls, such as phone numbers, duration of the call, IP addresses of the server, and the time of call termination.

Actoriano and Riadi [13] gave a forensic investigation framework for the WhatsApp mobile application and the WhatsApp Web browser application. A mobile information extraction technique used for the framework involves extracting the encrypted message storage file of WhatsApp conversations. Similarly, web application information extraction involves using FTK (Forensic Toolkit) Imager to obtain Google Chrome SQLite Database files for history, cache, and web session information. The persistent storage information including IndexedDB storage is not included in the research.

Vukadinovic [14] conducted a study to locate WhatsApp artifacts in both messenger and web client applications. For the WhatsApp Web application Windows and Mac operating systems with Chrome, Firefox and Safari browsers were covered. The study focused over the log file WhatsApp Web keeps for browser records and presented artifacts such as cached profile pictures and history logs for value to forensic investigations. Even though the artifacts from Web Storage were covered in detail, no detailed listing of IndexedDB artifacts were presented. The study stated the fact that web application was not keeping sent text messages in the client-side storage. The records of when the text messages were sent and when a voice/video call were active were not covered nor processed for verification purposes to investigations of traditional browser storage technologies.

Mahajan et al. [15] utilized internal memories of Android phones for the forensic investigations of WhatsApp and Viber. UFED (Universal Forensic Extraction Device) was used for file extraction and the Sqlite database browser was used for the investigation of the data. The research was conducted on five Android phones with three different operating systems installed: Froyo 2.2, GingerBread 2.3, and Ice-Cream Sandwich 4.0. In the paper, the authors discussed that based on the logs and history files investigated, forensically significant data were detected for WhatsApp and Viber in all different operating systems of Android phones.

Sgaras et al. [16] presented evidence collection methods for forensic investigations on WhatsApp, Viper, Skype, and Tango instant messaging applications. The research was performed on both iOS and Android operating systems. The definition of types of artifacts that can be found in the four instant messaging applications was listed along with the methodologies of extractions.

Additionally, a limited number of studies focused on IndexedDB forensics and security without targeting WhatsApp Web artifacts; Paligu et al. [6] introduced a tool called BrowStExPlus that analyzed IndexedDB artifacts of Mozilla Firefox, Google Chrome, and Internet Explorer. Kimak et al. [17] investigated database vulnerabilities that are introduced with web browser persistent storage technologies such as IndexedDB. Arefipour and Mozahhebi [18] presented a comparison of IndexedDB and Android SQLite with a security evaluation.

Overall, research on WhatsApp security and forensics remains mostly focused on mobile devices. In addition, only a handful of research has been conducted on IndexedDB, excluding WhatsApp which is still considered an emerging technology. Therefore, research on the WhatsApp Web and its forensic investigation based on web browser IndexedDB technology is a dire need.

3. Background for IndexedDB and LevelDB

IndexedDB is a relatively new object-oriented NoSQL transactional database. It makes use of JavaScript code to handle data on the client side. Its implementation details are shared by W3C (World Wide Web Consortium) which makes its structure constant through different platforms [19]. According to W3C specifications, IndexedDB utilizes B-tree structures for efficient database operations. B-trees are data structures that enable efficient manipulation of data in particularly large databases [20].

Google Chrome started its partial support for IndexedDB in 2012 with Chrome 11 and fully supported it since 2017 [21]. Similarly, major web browsers such as Firefox and Opera have provided full support for IndexedDB since 2017. In our previous study in 2019 [6], we discovered that top websites of the US listed by Alexa [22] have been highly utilizing IndexedDB technology since its full adoption.

SQLite is commonly used for most common browser storage technologies like history and bookmarks through all major web browsers. Similarly, Mozilla Firefox utilizes SQLite for supporting IndexedDB [23]. However, Google Chrome announced that it implemented IndexedDB over LevelDB, a technology Google has been developing since 2003 [24]. LevelDB implementation brings advantages to Chrome as it is implemented for fast operations of key-value pairs [25]. A benchmark study of 2011 looking at the advantages of LevelDB over SQLite v3 shows that LevelDB is better suited for batch updates of web browsers [26]. Since IndexedDB operation standards are established by W3C, Google Chrome LevelDB technology adopts similar use of IndexedDB through JavaScript commands. The LevelDB storage keeps its contents in an .ldb file. This file is kept locked, protecting its contents in accordance with the Same Origin Policy. Additionally .log and MANIFEST files are kept in the same location. When the database is opened, the information in the .log file is converted to datapoints and the MANIFEST file is updated with the information about the known datapoints. Datapoints are readable for the .ldb file whereas the .log file contains UTF (Unicode Transformation Format-16) encoded information as well as binary entries.

The highest level in the IndexedDB is a database. Therefore, the first step for creating IndexedDB storage is to establish a database schema. Databases are established with a given version. These databases in turn contain object stores that are similar to tables of traditional databases. If a new version of the database is defined with the indication that there are upgrades to the database, the onupgradeneeded function is called to recreate the object stores. Otherwise, the onsuccess function is called, where the object store is retrieved from the existing storage. Similarly, in the first attempt, the onupgradeneeded function creates the object store [27]. Code 1 shows an example of the initiation of a database in JavaScript code.

Code 1. Initiation of Database

```
    // Establish database and its objectstore
    var db;
    var request = window.indexedDB.open("newStudyDatabase", 1);

    // In case there is an error
    request.onerror = function(event) {
    // handling the error
}
    // When the we establish the database successfully
    request.onsuccess = function(event) {
    db = request.result; // if successful, get the database
}
    // If we need a new version of the database, or it is a first time call
    request.onupgradeneeded = function(event) {
    var db = event.target.result;
    var objectStore = db.createObjectStore("studyObjectStore", {keyPath: "id"});
```

In order to add information to the object store, we need to define a transaction on the database and its object store. This transaction is used call add function and can be handled with the onsuccess and onerror functions. A similar read function can be called on transactions to retrieve the data. Code 2 shows an example of data creation, its insertion and retrieval from the object store defined in Code 1.

Code 2. Data Insertion and Retrieval

```
// Insertions successful
request.onsuccess = function(event) {
// handle the successful insertion
}
// Insertions error
request.onerror = function(event) {
// handling the insertion error
}
var request = db.transaction([“studyObjectStore”], “readwrite”)
.objectStore(“studyObjectStore”)
.get({“01”})
// Retrieval successful
request.onsuccess = function(event) {
// do something with retrieved data such as ‘request.result.recordname’
}
// Retrieval error
request.onerror = function(event) {
// handling the retrieval error
}
```

Browsers secure the contents of an origin from other origins by enforcing the Same Origin Policy [28]. It means that the information obtained from one source is sandboxed and can only be reached by the same source. In this way, a website open in a browser tab cannot reach to the contents of another. IndexedDB is also subjected to Same Origin Policy. Therefore, you cannot establish a database and reach to the IndexedDB contents of another website. This makes forensic investigations relatively harder since the investigators cannot access IndexedDB contents of an origin through a web program developed in JavaScript. Luckily, LevelDB keeps log files that contain the same information recorded in the database. These files are named after the version of the database and a new file with different version name is created when there is updates to the database. However, with file carving it is possible to retrieve files even after new ones replace the old making them potentially available for investigation. More information on how to obtain the log files and its forensic investigations is presented in the next sections.

4. Materials and Methods

Two hypotheses are created to test the research question in this study.

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

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

4.1. Experimental Design

In this research, the Single Case Pretest–Posttest Quasi Experiment has been performed [29]. In a quasi-experiment, there is only one subject that is measured and put through treatment and then measured again for the change evaluations.

The subject of the research is the WhatsApp Web Application. The experimental environment is a Sony Vaio Laptop that operates a Windows 10 Single Language Operating System with Google Chrome v83.0.4103.97 (Official Build) (64-bit) browser installed. It is complemented with two Samsung S8+ phones that are installed with OS 10 - Q running the WhatsApp Messenger v2.20.172.

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

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

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

4.2. Pretest

The artifacts inherently found in the IndexedDB storage are tested with the following steps.

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

4.3. Treatment

The treatment of this research is the activities performed with WhatsApp Messenger and Web Application to create artifacts on the IndexedDB storage. The activities are created according to the observation of common user behavior with web browsers and messenger communication applications. When stored information of activity is inspected on three volunteer's WhatsApp Messenger and Web Application, the following activities are observed to constitute over ninety five percent of all activities in the last two weeks.

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

Additionally, in the preliminary and exploratory investigations performed to discover the potential of the research, some records of user presence were observed. Therefore, walking away from the computer with the phone is added to the given activity list.

Based on the observed behavior, the following steps are constructed as the treatment:

- Phone1 WhatsApp Messenger is connected to PC1 WhatsApp Web Application via QR code
- The message "This is message 1" is sent from PC1 (Phone1 connected) to Phone2
- The message "This is reply 1" is sent from Phone2 to PC1
- The link of the sample video [30] "Wildlife Windows 7 Sample Video" is sent from PC1 to Phone2
- The link of the sample video "Wildlife Windows 7 Sample Video" is sent from Phone2 to PC1

- The video "Wildlife Windows 7 Sample Video" is played in Phone1
- The video "Wildlife Windows 7 Sample Video" is played in PC1
- The video "Wildlife Windows 7 Sample Video" is played in Phone2
- A video call request is sent from Phone2 to Phone1. Call is not answered
- A video call request is sent from Phone2 to Phone1. Call is answered and lasted over 5 s
- A video call request is sent from Phone1 to Phone2. Call is answered and lasted over 10 s
- Phone1 is carried around twenty meters (estimated roughly with twenty steps) away from PC1
- Phone1 is carried back next to PC1
- Phone1 is disconnected from PC1 and reconnected after 5 s
- Phone2 account is blocked and unblocked from Phone1

4.4. Post-Hoc Test

As the procedure for the quasi-experiments, an independent test is conducted without the activities performed at the treatment. Employing the complete steps was taken after the set-up of the experimental environment. The observation of the artifacts created in the IndexedDB storage files of PC1 is presented in the Results section.

5. Results

The results obtained from the quasi-experiment are evaluated for the two hypotheses that are used in this research: the artifact value for forensic investigations and timeline relevance.

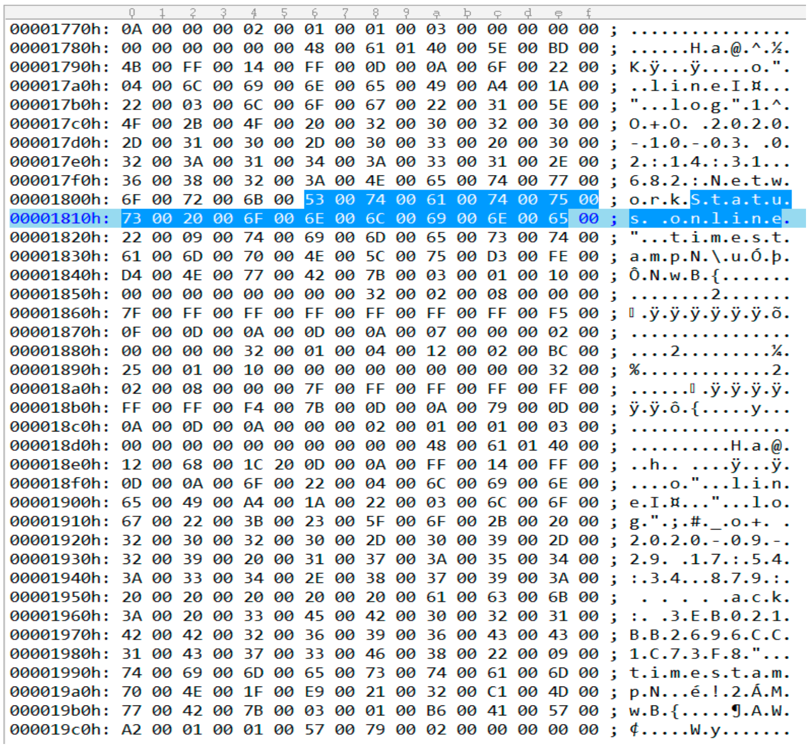
5.1. WhatsApp Web IndexedDB Artifacts Location and Storage File

The file storage location of IndexedDB in Chrome Browser used in a Windows 10 Single Language Operating System is "C:\Users\<username>\AppData\Local\Google\Chrome\User Data\Default\IndexedDB". The files are in .ldb format which is different than Firefox .sqlite and Internet Explorer .dat formats. There is an instance of .ldb file for IndexedDB storage for each website origin. Commonly, .ldb files are partially human-readable [6]. However, the files created for the WhatsApp Web appear to be almost completely understandable for a human reader. The clarity of the WhatsApp Web IndexedDB storage files make IndexedDB technology particularly suitable for WhatsApp forensic investigations. Additionally, it is observed that a .log file is created from an .ldb file during operation. The location of the .log files is the same as the .ldb files. These files include lines with characters that are not readable by humans as they contain binary data for entire object store collection of the origin within the database. However, the records that are observed valuable for forensic investigations are observed to be present in UTF-16 encoded human readable form in the same log files as well. Figure 1 shows an instance of .log files stored at IndexedDB file storage location.

The highlighted section shows a Network Status Online Record right after its time stamp in the human-readable format.

5.2. WhatsApp Web IndexedDB Artifacts

The treatment applied to the subject has resulted in the creation of artifacts at the IndexedDB storage files of the Chrome browser. It is observed that a great number of artifacts are created with the treatment. The application keeps records of technical operations, such as sync and async of the device, acknowledgement from the server about the contact availability, etc. We tried to narrow it down to records that can serve important evidence for the investigations. The artifacts, that we deemed significant, are listed in this section with their triggering treatments. The created following artifacts are indicating the user actions in the WhatsApp Web Application. A summary of the created artifacts is listed in Table 1 with their corresponding treatment action.



```
             0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00001770h: 0A 00 00 00 02 00 01 00 01 00 03 00 00 00 00 00 ; ................
00001780h: 00 00 00 00 00 00 48 00 61 01 40 00 5E 00 BD 00 ; ......H.a.@.^.½.
00001790h: 4B 00 FF 00 14 00 FF 00 0D 00 0A 00 6F 00 22 00 ; K.ÿ...ÿ.....o.".
000017a0h: 04 00 6C 00 69 00 6E 00 65 00 49 00 A4 00 1A 00 ; ..l.i.n.e.I.¤...
000017b0h: 22 00 03 00 6C 00 6F 00 67 00 22 00 31 00 5E 00 ; "...l.o.g.".1.^.
000017c0h: 4F 00 2B 00 4F 00 20 00 32 00 30 00 32 00 30 00 ; O.+.O. .2.0.2.0.
000017d0h: 2D 00 31 00 30 00 2D 00 30 00 33 00 20 00 30 00 ; -.1.0.-.0.3. .0.
000017e0h: 32 00 3A 00 31 00 34 00 3A 00 33 00 31 00 2E 00 ; 2.:.1.4.:.3.1...
000017f0h: 36 00 38 00 32 00 3A 00 4E 00 65 00 74 00 77 00 ; 6.8.2.:.N.e.t.w.
00001800h: 6F 00 72 00 6B 00 53 00 74 00 61 00 74 00 75 00 ; o.r.k.S.t.a.t.u.
00001810h: 73 00 20 00 6F 00 6E 00 6C 00 69 00 6E 00 65 00 ; s. .o.n.l.i.n.e.
00001820h: 22 00 09 00 74 00 69 00 6D 00 65 00 73 00 74 00 ; "...t.i.m.e.s.t.
00001830h: 61 00 6D 00 70 00 4E 00 5C 00 75 00 D3 00 FE 00 ; a.m.p.N.\.u.Ó.þ.
00001840h: D4 00 4E 00 77 00 42 00 7B 00 03 00 01 00 10 00 ; Ô.N.w.B.{.......
00001850h: 00 00 00 00 00 00 00 00 32 00 02 00 08 00 00 00 ; ........2.......
00001860h: 7F 00 FF 00 FF 00 FF 00 FF 00 FF 00 FF 00 F5 00 ; ▯.ÿ.ÿ.ÿ.ÿ.ÿ.ÿ.õ.
00001870h: 0F 00 0D 00 0A 00 0D 00 0A 00 07 00 00 00 02 00 ; ................
00001880h: 00 00 00 00 32 00 01 00 04 00 12 00 02 00 BC 00 ; ....2.........¼.
00001890h: 25 00 01 00 10 00 00 00 00 00 00 00 00 00 32 00 ; %.............2.
000018a0h: 02 00 08 00 00 00 7F 00 FF 00 FF 00 FF 00 FF 00 ; ......▯.ÿ.ÿ.ÿ.ÿ.
000018b0h: FF 00 FF 00 F4 00 7B 00 0D 00 0A 00 79 00 0D 00 ; ÿ.ÿ.ô.{.....y...
000018c0h: 0A 00 0D 00 0A 00 00 00 02 00 01 00 01 00 03 00 ; ................
000018d0h: 00 00 00 00 00 00 00 00 00 00 48 00 61 01 40 00 ; ..........H.a.@.
000018e0h: 12 00 68 00 1C 20 0D 00 0A 00 FF 00 14 00 FF 00 ; ..h.. ....ÿ...ÿ.
000018f0h: 0D 00 0A 00 6F 00 22 00 04 00 6C 00 69 00 6E 00 ; ....o."...l.i.n.
00001900h: 65 00 49 00 A4 00 1A 00 22 00 03 00 6C 00 6F 00 ; e.I.¤..."...l.o.
00001910h: 67 00 22 00 3B 00 23 00 5F 00 6F 00 2B 00 20 00 ; g.".;.#._.o.+. .
00001920h: 32 00 30 00 32 00 30 00 2D 00 30 00 39 00 2D 00 ; 2.0.2.0.-.0.9.-.
00001930h: 32 00 39 00 20 00 31 00 37 00 3A 00 35 00 34 00 ; 2.9. .1.7.:.5.4.
00001940h: 3A 00 33 00 34 00 2E 00 38 00 37 00 39 00 3A 00 ; :.3.4...8.7.9.:.
00001950h: 20 00 20 00 20 00 20 00 20 00 61 00 63 00 6B 00 ;  . . . . .a.c.k.
00001960h: 3A 00 20 00 33 00 45 00 42 00 30 00 32 00 31 00 ; :. .3.E.B.0.2.1.
00001970h: 42 00 42 00 32 00 36 00 39 00 36 00 43 00 43 00 ; B.B.2.6.9.6.C.C.
00001980h: 31 00 43 00 37 00 33 00 46 00 38 00 22 00 09 00 ; 1.C.7.3.F.8."...
00001990h: 74 00 69 00 6D 00 65 00 73 00 74 00 61 00 6D 00 ; t.i.m.e.s.t.a.m.
000019a0h: 70 00 4E 00 1F 00 E9 00 21 00 32 00 C1 00 4D 00 ; p.N...é.!.2.Á.M.
000019b0h: 77 00 42 00 7B 00 03 00 01 00 B6 00 41 00 57 00 ; w.B.{.....¶.A.W.
000019c0h: A2 00 01 00 01 00 57 00 79 00 02 00 00 00 00 00 ; ¢.....W.y.......
```

Figure 1. A .log file taken from Google Chrome IndexedDB file storage location.

Table 1. Treatments and Artifacts Created.

Treatment [1]	Artifact
• The message "This is message 1" is sent from PC1 to Phone2 • The message "This is reply 1" is sent from Phone2 to PC1	• Send Action Message Chat Record • Recv Action Message Relay Chat Record
• The video "Wildlife Windows 7 Sample Video" is played in Phone1 • The video "Wildlife Windows 7 Sample Video" is played in PC1 • The video "Wildlife Windows 7 Sample Video" is played in Phone2	• Media Load on Loaded Data Record
• A video call request is sent from Phone2 to Phone1. Call is not answered • A video call request is sent from Phone2 to Phone1. Call is answered and lasted over 5 s	• Recv: s<Number> [Call, ...]
• Phone1 is carried around twenty meters away from PC1 • Phone1 is carried back next to PC1	• Action Presence Unavailable Record • Action Presence Available Record
• Phone1 is disconnected from PC1 and reconnected after 5 s	• Stream:rememberMe: true Record
• Phone1 WhatsApp Messenger is connected to PC1 WhatsApp Web Application via QR code	• Network Status Online Record

[1] Treatments that have not created meaningful artifacts are not listed in this table.

5.2.1. Network Status Online

When the user's tab on the browser containing the WhatsApp Web Application is active, "Network Status Online" record is created in the IndexedDB storage. This record includes repeated information of record labels and the time stamp on its body. An instance of this record can be found in Record 1.

Record 1. Network Status Online

```
    {line: 2056, log: "*O=? 2020-08-29
    23:50:47.044:NetworkStatus online", timestamp:
    1598763047703.76}
    line: 2056
    log: "*O=? 2020-08-29 23:50:47.044:NetworkStatus online"
timestamp: 1598763047703.76
```

"NetworkStatus Online" carries forensically significant value as it is an indicator of the time the user is interacting with the application. In an investigation, the charges against a suspect often rely on the timestamps of the evidence [31]. Matching time settings between the evidence timestamps and user interaction with the computer can serve as a strong indicator of a suspect being responsible for the evidence.

5.2.2. Stream:rememberMe: True

"Stream:rememberMe" record is created when a user establishes the connection between the computer browser embodying the WhatsApp Web Application and the phone with WhatsApp Messenger. The record includes information on the record label and timestamp on its body. Record 2 shows an instance of this record.

Record 2. Stream:rememberMe

```
    {line: 2057, log: "*O=? 2020-08-29
    23:50:47.059:Stream:rememberMe: true", timestamp:
    1598763047704.23}
    line: 2057
    log: "*O=? 2020-08-29 23:50:47.059:Stream:rememberMe:
    true"
    timestamp: 1598763047704.23
```

"Stream:rememberMe" is used when the suspect wants WhatsApp Messenger to remember the computer. This preference may signal recurrent usage of the computer by the suspect. Digital forensic investigation reports include information on the behavior of the suspect [32]. This information is presented to the court to make better sense of the suspect's motives. A suspect's frequent interaction with the evidence computer is important as it gives an inside on the behavioral characteristics of the suspect.

5.2.3. Media Load on Loaded Data

A "MediaLoad:video.onloadeddata" record is created, when a media file such as a video file is opened on either messenger or web application. Timestamps indicating the opening time of the media is placed in the record with the record label. Record 3 shows an instance of media load records.

Many cases involving digital evidence are related to illegal images and videos of minors [33]. There are also cases of privacy issues between couples involving the distribution and exposition of private media [34]. In these cases, the information on when the media is accessed and how frequently it is accessed is crucial to the investigation [35].

Record 3. Media Load on Loaded Data

```
{line: 2330, log: "*O=? 2020-08-29
23:53:19.768:MediaLoad:video.onloadeddata #1",
timestamp: 1598763200175.39}
Line: 2330
log: "*O=? 2020-08-29
23:53:19.768:MediaLoad:video.onloadeddata #1"
timestamp: 1598763200175.39
```

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

"Recv: s<Number> [Call, ...]" record is created when a video or voice call is active using either messenger or the web application. Timestamp information of when the calls are active is included in the data with the record label. The application puts this log to the file multiple times during the call is active. It is also observed that a missed call can produce the "Recv: s<Number> [Call, ...]" record. An unanswered voice call that rings for 1 min is seen to generate over nine instances of the record. Record 4 demonstrates this record type with an instance.

Record 4. Recv: s<Number> [Call, ...]

```
{line: 2345, log: "*O=? 2020-08-29 23:53:51.325: recv: s29
[Call, ...]", timestamp: 1598763231721.61}
Line: 2345
log: "*O=? 2020-08-29 23:53:51.325: recv: s29 [Call, ...]"
timestamp: 1598763231721.61
```

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

5.2.5. Action Presence Available—Unavailable

An "action, presence, unavailable" record is created with the application client sending its presence information to the WhatsApp server. This record is indicative of a user being online or, in other words, actively using the application. However, we have observed that when the user walks away from the computer with the phone connected through WhatsApp Messenger, both devices become inactive in their use of the application and add the "action, presence, unavailable" record. This is the case even when the web application tab stays open.

The connection of the devices is not dependent on the distance. This is evident from the connection not being lost when a distance is introduced between the devices. Additionally, a connection can be established when the computer is connected through WIFI and the Android device is connected through a service provider. The preliminary evidence indicates a timeout mechanism for the record to be created. This is supported by the fact that it takes eight to eleven seconds for the record to be created. However, more information can be obtained through inspection of the network traffic generated by the devices during the connection. In this study our subject is the IndexedDB storage of suspect's computer and the artifacts created in it with the given treatment of the experiment.

The beginning of the time when the user is walking away from the computer is recorded with the timestamp of the record label. Although, its timing is not very precise as the records are created after eight to eleven seconds. It should be noted that when the connection between the phone and the computer is problematic, the presence unavailable record is occasionally added with a short duration of ten to twenty seconds before the "action, presence, available" record is added again. During the treatments, one such record has been observed. It is also notable that when the duration for the user's absence is long, multiple records are added. Similarly, the user available record is occasionally added without the user being offline or stepping away from the computer. Record 5 shows an instance of this record type.

Record 5. Action Presence Unavailable

```
{line: 2406, log: "*O=? 2020-08-29 23:55:10.199:sending:
1598763310.–44, action, presence, unavailable", timestamp:
1598763310592.405}
Line: 2406
log: "*O=? 2020-08-29 23:55:10.199:sending: 1598763310.-
-44, action, presence, unavailable"
 timestamp: 1598763310592.405
```

In cases involving digital evidence, one of the very common defenses is the claim that the suspect was not the person using the computer at the time of the offense [36]. Personal devices such as mobile phones are more likely to be used by a single individual. While computers can be used by more than one user, it is a more likely case in businesses with printing, database, and common purpose computers. A record displaying when the suspect walks away from the computer is potentially significant information to support or be against the defense of not being present during the time offense taking place.

5.2.6. Send Action Message Chat and Recv Action Message Relay Chat

The "send<code>, action, message, chat" and "recv<code> action, msg, relay, chat" records are created when a text message is sent from one account to another. The application keeps more detailed information on the handshake of the device to send the message and further details on its delivery. However, these messages vary in means of network delays and server states. A Plain Send—Receive record is observed to be sufficient to pinpoint a messaging activity for the time frame analysis. The Send—Receive records contain timestamps and record labels. Additionally, an identifier code such as 3EB0A2F3697 ... 6B3365 is present in the record. This code seems to contain information on the account the message is sent to, or the account the message is originated from. Therefore, it can be used for separating the conversations into different accounts. However, there is no identifiable way to determine what account the code belongs to.

Record 6. Send Action Message

```
{line: 2267, log: "*O=? 2020-08-29 23:51:41.554: send:
3EB00698747 ... 00B6AB,
action, message, chat,,3EB0069874767200B6AB", timestamp:
1598763101996.04}
line: 2267
log: "*O=? 2020-08-29 23:51:41.554: send:
3EB0069874767200B6AB,
action, message, chat,,3EB0069874767200B6AB"
 timestamp: 1598763101996.04
```

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

6. Discussion

The artifacts created in the IndexedDB storage by the WhatsApp Web Application appear to provide extensive information on the actions of a user. This information is recorded in a format including the time of the action in the UNIX epoch time format, additional to the human-readable format. UNIX epoch time is the number of seconds since 1 January 1970 (midnight UTC/GMT). Therefore, it is in a different time zone than the human-readable time in the records. For instance, 1598763047703.76 from Record 1 corresponds to “Sun, 30 Aug 2020 04:50:47 GMT”. In the same record, the time is seen as “2020-08-29 23:50:47.044” because the subject computer is in the (GMT-5) time zone. The records are divided with numbering which makes it easier for parsing. It can be observed in Records 1 to 6 that the information repeats in different formats. The first set of records represents the information in brackets whereas the subsequent record separates the time, label, and line number into different lines. This is indicating a design that intends to support different methods of information collection.

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

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

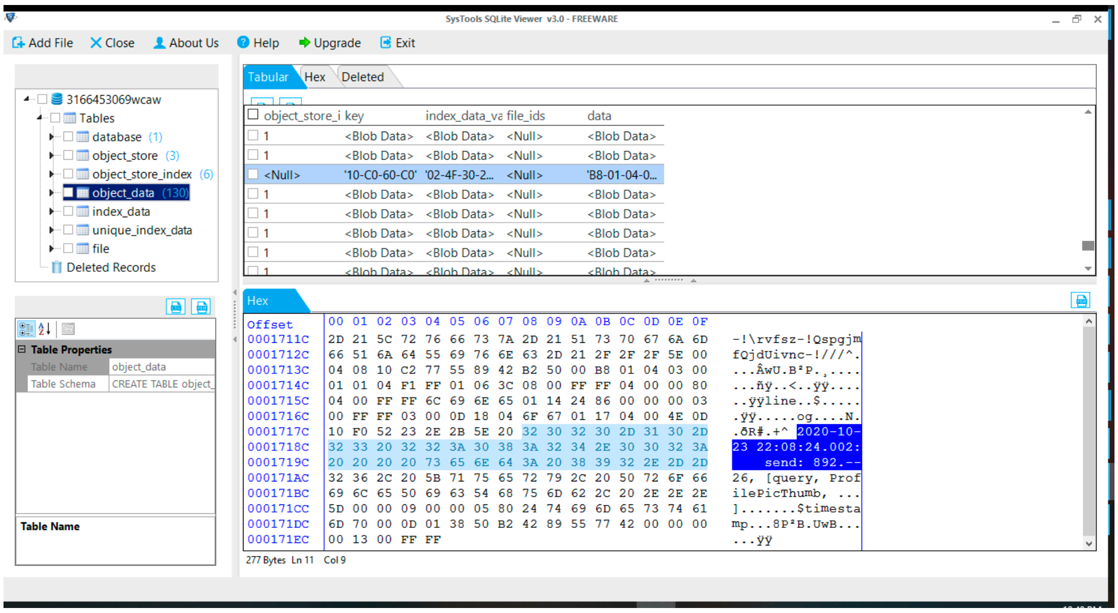
IndexedDB Implementation Across Different Browser Technologies

This study focuses on Google Chrome storage of WhatsApp Web artifacts on IndexedDB, which is implemented in LevelDB. However, various implementations exist across different browsers. Table 2 summarizes the technologies behind IndexedDB for major browsers. Microsoft Edge and Opera use LevelDB for IndexedDB like Google Chrome. Therefore, the concepts detailed in the study applies for these browsers as well.

Table 2. IndexedDB Technology Across Different Browsers.

Browser	Technology
Google Chrome	LevelDB
Microsoft Edge	LevelDB
Opera	LevelDB
Mozilla Firefox	SQLite
Internet Explorer	.dat files

Mozilla Firefox utilizes SQLite files for IndexedDB, which is easily accessible with SQLite browsers. Figure 2 shows a WhatsApp Web IndexedDB storage file displayed in the SQLite Browser [37]. It is important to highlight that there is no security mechanism blocking access to the IndexedDB SQLite file utilized for Mozilla Firefox. Internet Explorer utilizes .dat files for its entire browser storage. These files are specific to Internet Explorer and can be accessed by any text editor.



	object_store_i	key	index_data_va	file_ids	data
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	<Null>	'10-C0-60-C0'	'02-4F-30-2...	<Null>	'B8-01-04-0...
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>
☐	1	<Blob Data>	<Blob Data>	<Null>	<Blob Data>

```
Offset    00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
0001711C  2D 21 5C 72 76 66 73 7A 2D 21 51 73 70 67 6A 6D   -!\rvfsz-!Qspgjm
0001712C  66 51 6A 64 55 69 76 6E 63 2D 21 2F 2F 2F 5E 00   fQjdUivnc-!///^.
0001713C  04 08 10 C2 77 55 89 42 B2 50 00 B8 01 04 03 00   ...ÂwU.B²P.¸....
0001714C  01 01 04 F1 FF 01 06 3C 08 00 FF FF 04 00 00 80   ...ñÿ..<..ÿÿ....
0001715C  04 00 FF FF 6C 69 6E 65 01 14 24 86 00 00 00 03   ..ÿÿline..$.....
0001716C  00 FF FF 03 00 0D 18 04 6F 67 01 17 04 00 4E 0D   .ÿÿ.....og....N.
0001717C  10 F0 52 23 2E 2B 5E 20 32 30 32 30 2D 31 30 2D   .ðR#.+^ 2020-10-
0001718C  32 33 20 32 32 3A 30 38 3A 32 34 2E 30 30 32 3A   23 22:08:24.002:
0001719C  20 20 20 20 73 65 6E 64 3A 20 38 39 32 2E 2D 2D       send: 892.--
000171AC  32 36 2C 20 5B 71 75 65 72 79 2C 20 50 72 6F 66   26, [query, Prof
000171BC  69 6C 65 50 69 63 54 68 75 6D 62 2C 20 2E 2E 2E   ilePicThumb, ...
000171CC  5D 00 00 09 00 00 05 80 24 74 69 6D 65 73 74 61   ].......$timesta
000171DC  6D 70 00 0D 01 38 50 B2 42 89 55 77 42 00 00 00   mp...8P²B.UwB...
000171EC  00 13 00 FF FF                                    ...ÿÿ
```

Figure 2. SQLite Browser Displaying Mozilla Firefox IndexedDB SQLite File.

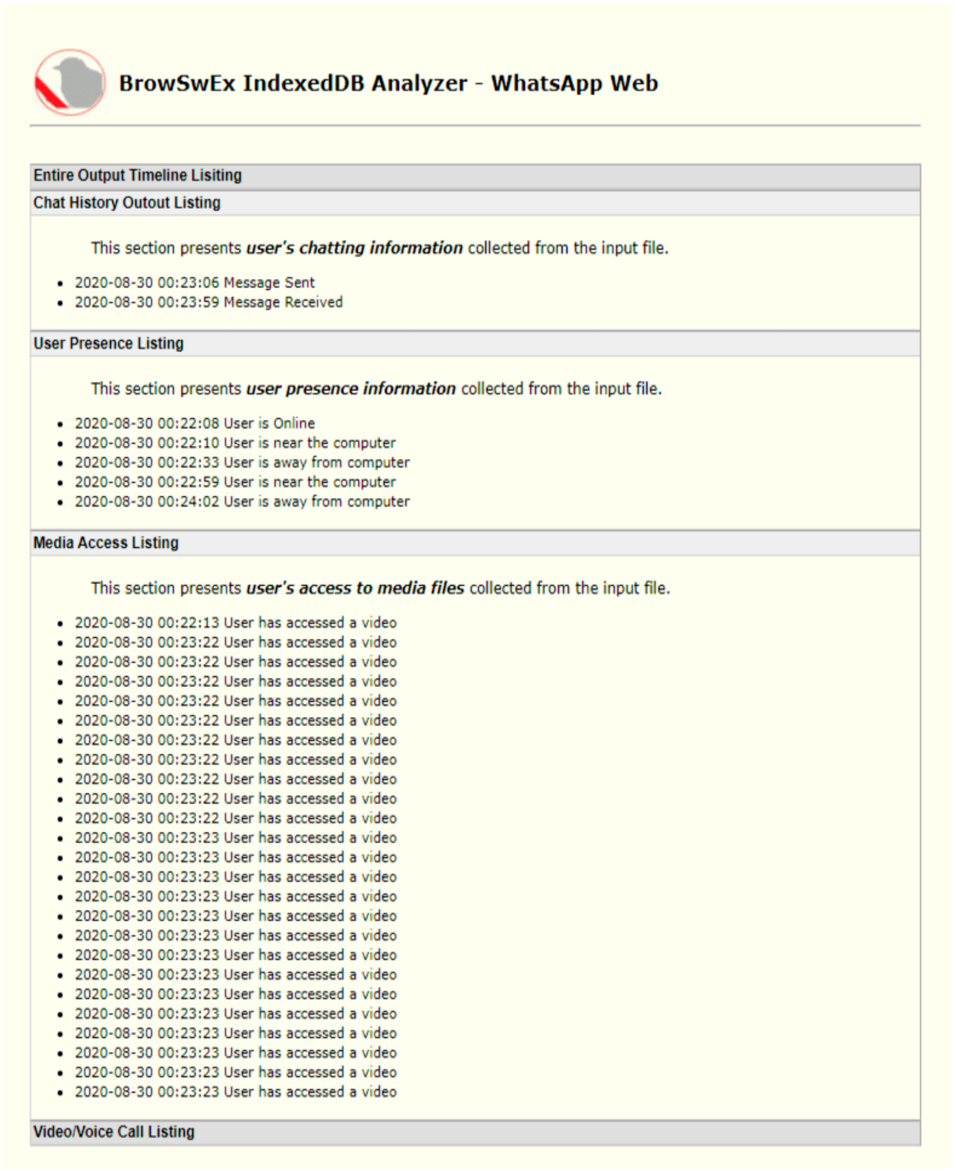
7. BrowSwEx

IndexedDB can be operated through JavaScript code. However, the Same Origin Policy prevents a database established by localhost or any other origin to access the WhatsApp Web content stored in the browser. Therefore, the only way to reach the records is through direct access to the files kept in the Google Chrome IndexedDB file location. When the WhatsApp Web is initiated in the Chrome browser, an .ldb file is created to contain all information about the present and previous interactions with the application. Google Chrome uses LevelDB for IndexedDB storage. LevelDB is developed with C/C++ technology. Therefore, direct access to the .ldb files can only be achieved using C/C++ programs [38,39]. However, as it is discussed in the background and results sections, .ldb files leave information in a .log file that is stored in the same file location as the .ldb file. Therefore, there are two ways to collect the WhatsApp Web IndexedDB records from Google Chrome. The first way is to utilize a C compiler with a LevelDB library [39]. The second way is to process the .log file by parsing the raw text log style information. For the proof of concept, BrowSwEx is developed using a PHP program that systematically parses .log file that is produced by .ldb file [40]. The pseudocode employed by BrowSwEx for processing the .log file can be seen in Pseudocode 1.

Pseudocode 1.

```
if input.log file is not present
Extract the <numberpattern>.log file from Chrome IndexedDB file location
    Rename the <numberpattern>.log file to input.log
While there are more lines to read in input.log
read a line
if the line contains time entry
    While there are more key record types to checked
        Bring a record type
            if the line matches with the record type
            Convert record type to description
            Add the time and record description to results
            else
            continue
    else
    continue
Sort results based on time.
```

After the .log file is processed, the obtained results are utilized in different functions for presentation of specified information. The tool utilizes the preg_match function and five specializing functions: EntireOutputList(), ChatOutputList(), PresenceOutputList(), MediaAccessOutputList() and VideoCallOutputList() to list the information and reach to the desired record formats listed in the previous section. An image of the tool presenting information in time frames can be seen in Figure 3.


BrowSwEx IndexedDB Analyzer - WhatsApp Web

Entire Output Timeline Lisiting

Chat History Outout Listing

This section presents ***user's chatting information*** collected from the input file.

- 2020-08-30 00:23:06 Message Sent
- 2020-08-30 00:23:59 Message Received

User Presence Listing

This section presents ***user presence information*** collected from the input file.

- 2020-08-30 00:22:08 User is Online
- 2020-08-30 00:22:10 User is near the computer
- 2020-08-30 00:22:33 User is away from computer
- 2020-08-30 00:22:59 User is near the computer
- 2020-08-30 00:24:02 User is away from computer

Media Access Listing

This section presents ***user's access to media files*** collected from the input file.

- 2020-08-30 00:22:13 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:22 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video
- 2020-08-30 00:23:23 User has accessed a video

Video/Voice Call Listing

Figure 3. BrowSwEx Listing WhatsApp Web Records.

7.1. BrowSwEx Verification

The treatments from the Single Case Pretest–Posttest Quasi Experiment were detected successfully by BrowSwEx. However, an extensive number of entries and variety of similar log types indicated potential for error. Therefore, further verification appears to be necessary. BrowSwEx was tested with the Google Chrome Developer Tools [41]. Google Chrome Developer Tools lists every record of persistent storage for the active website. The information is displayed in a raw format with excruciating details and no option to search or sort. For Google Chrome Developer tools to operate on the website, the computer needs to be running and the website needs to be authenticated by the user's phone.

BrowSwEx parses the recorded .log file taken from the Chrome IndexedDB file storage folder. In other words, the analysis can be performed offline without user credentials present for authentication. The approach to the verification of the information displayed by BrowSwEx is folded in two parts:

- The existence of the records displayed were cross checked by Google Chrome Developer Tools.
- The time of the treatment to create artifacts is checked against the time displayed in the tool user interface.

7.2. BrowSwEx Restrictions

In order to access the <numberpattern >.log file, the <userprofile> location (different for every username) in the "WhatsAppWebIDB.php" file has to be updated. Since every environment is different. In some cases, the user needs to manually obtain the <numberpattern >.log file from the IndexedDB file storage location, rename it to input.log and place it in the web server directory. Additionally, some records seem to be repetitive, e.g., when the user starts a video call, "Recv: s<Number> [Call, ...]" record is entered to IndexedDB file multiple times for the same time entry. After some consideration, these repetitive entries have not been eliminated. This is because the different logs contain different identifications which can be important in the recreation of the investigation results. Nonetheless, BrowSwEx is an open source tool and can be improved by the users according to their needs.

Users need to have a functional server that parses PHP, and transfer the tool files to their www directory to get the tool working. We have used a WAMP (Windows, Apache, MySQL, and PHP) server [42] with built-in PHP and Apache Server during the development of BrowSwEx.

8. Conclusions

In this research, we scrutinized the artifacts stored by IndexedDB technology for the WhatsApp Web Application. Two hypotheses are created for scrutinization: IndexedDB storage carries forensically significant artifacts for the WhatsApp Web Application and WhatsApp Web Application artifacts in IndexedDB can be used to create time frame analyses in forensic investigations.

A Single Case Pretest–Posttest Quasi Experiment was performed to evaluate the artifacts left in IndexedDB storage by the application. A proof-of-concept tool, BrowSwEx, was developed to demonstrate the value of the artifacts and introduce a technique of creating time frame analysis.

The results show that IndexedDB storage is a valuable source of information for forensic investigations when analyzed specifically for the WhatsApp Web Application. Information based on the actions of the suspect is detected to be suitable for utilization in a time frame indexed forensic investigation evidence presentation. Therefore, the two hypotheses created for the research are proved to be true.

Supplementary Materials: The following is available online at https://github.com/furkanpaligu/BrowSwEx, BrowSwEx tool software.

Author Contributions: Conceptualization, F.P. and C.V.; methodology, F.P. and C.V.; software F.P.; validation F.P. and C.V.; formal analysis F.P. and C.V.; resources, C.V; data curation, F.P.; writing—original draft preparation, F.P.; writing—review and editing, F.P. and C.V.; visualization, F.P.; supervision, C.V. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodison, S.E.; Davis, R.C.; Jackson, B.A. Digital Evidence and U.S. Criminal Justice System. Available online: https://www.ncjrs.gov/pdffiles1/nij/grants/248770.pdf (accessed on 29 August 2020).
2. Alvarez, L. Software designer reports error in Anthony trial. *New York Times*, 19 July 2011; A14.

3. Burch, A.D.S. Pedro Bravo Found Guilty of First-Degree Murder of Christian Aguilar. *Miami Herald*. 2014. Available online: https://www.miamiherald.com/news/local/community/miami-dade/article1980000.html (accessed on 29 August 2020).
4. Carroll, O. Challenges in Modern Digital Investigative Analysis. Crime Scene Investigator Network. 2019. Available online: https://www.crime-scene-investigator.net/challenges-in-modern-digital-investigative-analysis.html (accessed on 29 August 2020).
5. IndexedDB API. MDN Web Docs. Available online: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API (accessed on 29 September 2020).
6. Paligu, F.; Kumar, A.; Cho, H.; Varol, C. BrowStExPlus: A Tool to Aggregate Indexed DB Artifacts for Forensic Analysis. *J. Forensic Sci.* **2019**, *64*, 1370–1378. [CrossRef] [PubMed]
7. Storage for the Web. 2020. Available online: https://web.dev/storage-for-the-web (accessed on 29 September 2020).
8. Anglano, C. Forensic analysis of WhatsApp Messenger on Android smartphones. *Digit. Investig.* **2014**, *11*, 201–213. [CrossRef]
9. Thakur, N.S. Forensic Analysis of WhatsApp on Android Smartphones. Master's Thesis, University of New Orleans, New Orleans, LA, USA, 2013.
10. Shortall, A.; Azhar, M.H.B. Forensic acquisitions of WhatsApp data on popular mobile platforms. In Proceedings of the IEEE Sixth International Conference on Emerging Security Technologies, Braunschweig, Germany, 3–5 September 2015; pp. 13–17.
11. Umar, R.; Riadi, I.; Zamroni, G.M. A comparative study of forensic tools for WhatsApp analysis using NIST measurements. *Int. J. Adv. Comput. Sci. Appl.* **2017**, *8*, 69–75. [CrossRef]
12. Karpisek, F.; Baggili, I.; Breitinger, F. WhatsApp network forensics: Decrypting and understanding the WhatsApp call signaling messages. *Digit. Investig.* **2015**, *15*, 110–118. [CrossRef]
13. Actoriano, B.; Riadi, I. Forensic Investigation on WhatsApp Web Using Framework Integrated Digital Forensic Investigation Framework Version 2. *Int. J. Cyber Secur. Digit. Forensics IJCSDF* **2018**, *7*, 410–419.
14. Vukadinovic, N.V. WhatsApp Forensics: Locating Artifacts in Web and Desktop Clients. Master's Thesis, Purdue University Graduate School, West Lafayette, IN, USA, 2019.
15. Mahajan, A.; Dahiya, M.S.; Sanghvi, H.P. Forensic analysis of instant messenger applications on android devices. *arXiv* **2013**, arXiv:1304.4915. [CrossRef]
16. Sgaras, C.; Kechadi, M.T.; Le-Khac, N.A. *Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications*; Springer: Cham, Switzerland, 2012; pp. 188–199.
17. Kimak, S.; Ellman, J.; Laing, C. An investigation into possible attacks on HTML5 indexedDB and their prevention. In Proceedings of the 13th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting, Liverpool, UK, 25–26 June 2012.
18. Arefipour, S.; Mozahhebi, M. *Comparison of IndexedDB and SQLite Based on Developers' Concerns*; Student Essay; Gothenburg University: Gothenburg, Sweden, 2015.
19. W3C. Indexed Database Specification API 2.0. Available online: https://www.w3.org/TR/IndexedDB-2 (accessed on 30 September 2020).
20. Ferragina, P.; Grossi, R. The string B-tree: A new data structure for string search in external memory and its applications. *J. ACM* **1999**, *46*, 236–280. [CrossRef]
21. IndexedDB. Available online: https://caniuse.com/#search=indexedDB (accessed on 30 September 2020).
22. Alexa Top Sites in United States. Available online: https://www.alexa.com/topsites/countries/US (accessed on 30 September 2020).
23. Performance/Avoid SQLite in Your Next Firefox Feature. Available online: https://wiki.mozilla.org/Performance/Avoid_SQLite_In_Your_Next_Firefox_Feature (accessed on 30 September 2020).
24. Lin, J. Building a self-contained search engine in the browser. In Proceedings of the 2015 International Conference on the Theory of Information Retrieval, Northampton, MA, USA, 27 September 2015; pp. 309–312.
25. Dean, J.; Ghemawat, S. LevelDB: A Fast Persistent Key-Value Store. 27 July 2011. Available online: https://opensource.googleblog.com/2011/07/leveldb-fast-persistent-key-value-store.html (accessed on 30 September 2020).
26. LevelDB Benchmarks. Available online: http://www.lmdb.tech/bench/microbench/benchmark.html (accessed on 30 September 2020).
27. Using IndexedDB. MDN Web Docs. Available online: https://developer.mozilla.org/en-US/docs/Web/API/IndexedDB_API/Using_IndexedDB (accessed on 30 September 2020).

28. Same Origin Policy. W3. Available online: https://www.w3.org/Security/wiki/Same_Origin_Policy (accessed on 30 September 2020).
29. Cook, T.D.; Campbell, D.T. The design and conduct of true experiments and quasi-experiments in field settings. In *Reproduced in Part in Research in Organizations: Issues and Controversies*; Goodyear Publishing Company: Santa Monica, CA, USA, 1979.
30. Wildlife Windows 7 Sample Video. Available online: https://www.youtube.com/watch?v=a3ICNMQW7Ok (accessed on 29 August 2020).
31. Agarwal, A.; Gupta, M.; Gupta, S.; Gupta, S.C. Systematic digital forensic investigation model. *Int. J. Comput. Sci. Secur.* **2011**, *5*, 118–131.
32. Carrier, B.; Spafford, E. An event-based digital forensic investigation framework. Available online: http://www.digital-evidence.org/papers/dfrws_event.pdf (accessed on 10 October 2020).
33. Pollitt, M. A history of digital forensics. In *Springer IFIP International Conference on Digital Forensics*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 3–15.
34. What Is Nonconsensual Pornography? NCJFCJ. 2020. Available online: https://www.ncjfcj.org/news/what-is-nonconsensual-pornography/ (accessed on 29 August 2020).
35. Kao, D.Y. Cybercrime investigation countermeasure using created-accessed-modified model in cloud computing environments. *J. Supercomput.* **2016**, *72*, 141–160. [CrossRef]
36. Find Local Investigators. Computer Forensics/Cyber Crime. 2020. Available online: https://www.pinow.com/investigations/computer-forensics (accessed on 29 August 2020).
37. Sqlitebrowser. DB Browser for SQLite. Available online: https://sqlitebrowser.org (accessed on 24 October 2020).
38. Ideawu. PHP Working with LevelDB. 2013. Available online: http://www.ideawu.com/blog/post/40.html (accessed on 29 August 2020).
39. Dean, J.; Ghemawat, S. Leveldb. 2019. Available online: https://github.com/google/leveldb/ (accessed on 29 August 2020).
40. Paligu, F.; Varol, C. BrowSwEx. *Zenodo* **2020**. [CrossRef]
41. Developers Google. Chrome DevTools. 2019. Available online: https://developers.google.com/web/tools/chrome-devtools (accessed on 29 August 2020).
42. WAMP Server. Available online: https://www.wampserver.com/en/ (accessed on 29 August 2020).