WormHex: A Volatile Memory Analysis Tool for Retrieval of Social Media Evidence

Norah Almubairik, Wadha Almattar, Amani Alqarni

Abstract—Social media applications are increasingly being used in our everyday communications. These applications utilise end-to-end encryption mechanisms which make them suitable tools for criminals to exchange messages. These messages are preserved in the volatile memory until the device is restarted. Therefore, volatile forensics has become an important branch of digital forensics. In this study, the WormHex tool was developed to inspect the memory dump files for Windows and Mac based workstations. The tool supports digital investigators by enabling them to extract valuable data written in Arabic and English through web-based WhatsApp and Twitter applications. The results confirm that social media applications write their data into the memory, regardless of the operating system running the application, with there being no major differences between Windows and Mac.

Keywords—Volatile memory, REGEX, digital forensics, memory acquisition.

I. Introduction

TODAY, in the technological era, portable electronic devices have become part of our everyday life. Moreover, there has also been an increase in the use of social media applications that allow users to communicate and exchange messages [12]. Social media is undoubtedly a double-edged sword. On the one hand, it makes life easier; yet on the other hand, its use can result in disastrous consequences. Since communication through social media applications is encrypted using an end-to-end encryption mechanism, criminals frequently use these applications to exchange messages [12]. Given the fact that all users' activities are stored in the volatile memory until the device is rebooted, it makes it an excellent source for forensic investigators to explore [11]. Therefore, volatile memory forensics has become one of the most essential branches of digital forensics. In addition, investigators are able to extract valuable information from the data in the memory dump [12] [4].

In digital forensics, acquiring and analyzing data from workstations is considered a challenging task. Firstly, due to the fact that operating systems are continuously updated and the next generation of electronic devices are constantly being released, forensics acquisitions and analysis tools quickly become out of date. Secondly, collecting evidence written in Arabic is not easy, as there are far more tools available that acquire data in English compared to Arabic.

This study aims to support digital investigators analysing electronic crimes related to social media applications running on devices using both Windows and Mac operating systems. This study has two objectives. The first is to extract valuable information from WhatsApp, Twitter, and Telegram running on Windows and Mac OS, while the second is to explore the representation of Arabic text in the volatile memory. This study proposes a framework for the acquisition of data in the volatile memory of social media applications running on devices using Windows and Mac OS.

The rest of the paper is organised as follows: Section II provides the background related to data acquisition and regular expressions. Section III presents the work related to memory forensics and acquisition across different operating systems and devices. Furthermore, it presents the work that has been done to date on the subject of Arabic artifacts preserved in the memory. Section IV briefly describes the methodology followed in this study, while Section V describes the experiment conducted as part of this study. Section VI presents the final results, as well as an analysis of all of the findings. Finally, section VII concludes the paper, discusses the limitations, and provides recommendations for future studies on the subject.

II. Background

This section provides a brief introduction to data acquisition, including its types and methods. Furthermore, it focuses on regular expressions and their importance.

A. Data Acquisition

Data acquisition is commonly regarded as the process of copying data. However, in computer forensics, it refers to the task of collecting digital evidence from electronic media [9].

1) Types of Data Acquisition: There are two types of data acquisition: static acquisition and live acquisition. Static acquisition refers to the process of copying non-volatile data stored in internal or external hard drives, USB drives and flashcards. On the other hand, live acquisition refers to the acquisition of a machine that is still running and can retrieve both static and dynamic volatile data [6]. RAM, cache or any temporary storage medium are examples of live acquisition. When a digital device is powered on, it is essential to perform live acquisition, as it reveals important data that might not be shown in static acquisition (i.e., encrypted data). Furthermore, a large amount of data is written

N. Almubairik is with the Department of Networks and Communications, Imam Abdulrahman Bin Faisal University, Saudi Arabia (e-mail: naalmubairik@iau.edu.sa).

W. Almattar is with the Department of Computer Science, Imam Abdulrahman Bin Faisal University, Saudi Arabia (e-mail: wmalmattar@iau.edu.sa).

A. Alqarni is with the Department of Computer Science and Engineering, University of Hafr Al-batin, Saudi Arabia (e-mail: amaniqarni@uhb.edu.sa)

All authors have contributed to this study equally.

to memory, including encryption keys, hash passwords and different user activities. Social media applications running on workstations leave a footprint in the memory, including user profile information and messages that have been exchanged. Therefore, this research project focuses on live acquisition rather than on static acquisition.

2) Methods of Data Acquisition:

1) Disk-to-image file: This is the most common data acquisition method, in which one or many copies of a suspect's drive are made. These copies are bit-for-bit replications of the original drive [9]. One main advantage of this method is that many forensics tools (e.g., Encase, ProDiscover) can read and analyse these image files.
2) Disk-to-disk: The disk-to-image file method can be impossible if there are hardware or software errors or incompatibilities between the source and destination drives. Therefore, a disk-to-disk copy of the suspect's drive is needed. As part of this method, the investigator has to purchase a drive identical to the suspect's drive to use as the destination drive. Otherwise, the acquisition tool may adjust the destination drive's geometry, so that the copied data matches the original suspect drive. [9].
3) Logical data copy from a file or folder: The first two methods are time-consuming, as they involve the physical extraction of data from an entire drive. Thus, logical acquisition, which acquires only files of interest from the suspect's drive, is used. It should be noted that this technique, which involves physical extraction, needs root access, unlike logical extraction, which can be performed on rooted or unrooted devices [7]

B. Regular Expressions

Regular Expression, which is often abbreviated to REGEX, describes the patterns used to match characters. These patterns are used to find specific strings within a given text data set, as pattern-matching algorithms are based on it. Generally speaking, REGEX is used to different extents in tasks such Natural Language Processing (NLP); it is considered a cornerstone for building NLP based systems (e.g., chatbots). Most programming languages such as Java and Python offer REGEX as built-in options or are available through libraries. In addition, REGEX is mostly used in search engines, data extraction, string replacement and text processing programs. For example, REGEX can help extract emails from an unstructured data set.

III. Related Work

Due to the dramatic increase in the use of different social media applications that provide different services and entertainment, many studies have been conducted to investigate and analyse social media and instant messaging (IM) applications when seeking forensic evidence. Al Mutawa et al. [2] claimed that RAM analysis is considered a challenging task, due to the need for high-level expertise to acquire specific valuable data from the RAM. Moreover, RAM analysis may aid digital forensics examiners to acquire valuable evidence. Walnycky et al. [14] compared 20 applications in terms of their ability to reconstruct or intercept data, including passwords, pictures and videos. They found that most of these applications have some significant vulnerabilities in terms of storing data in the memory and transmitting it over the network. A number of studies have focused on acquisition in different operating systems. Stefan et al. [13] discussed memory acquisition for files, processes, system registry, applications and cryptographic keys running on the Windows operating system. In addition, a framework for volatile memory forensics was proposed by Thantilage et al. [12].The framework acquires social media and Instant Messages (IM) artifacts running on the memory of two popular operating systems: Windows OS and Mac OS. They explored the RAM data to identify the pattern needed to build regular expressions (REGEX) based on the extracted data. Then, they used these REGEX to extract valuable data from different memory dumps.

Another study was conducted by Al-Mutawa et al. [1] to determine if it is possible to recover activities performed using social media applications running on different smartphones (BlackBerry, iPhones, and Android). The results concluded that data cannot be recovered from BlackBerry devices, while a significant amount of data, which could be considered valuable information to a forensic investigator, can be recovered in the case of iPhone and Android phones. Another study on Android phones was conducted by Nisioti et al. [8]. They presented a methodology for retrieving instant messaging data from WhatsApp and Facebook from the RAM of an Android mobile device. Their results revealed that a large amount of data could be retrieved from the memory, even if the device's battery had been removed for a short time. Furthermore, they were able to retrieve data sent a few months before the data acquisition occurred. Furthermore, one study involved extracting Arabic artifacts left in social media memory. Al Mutawa et al. [2] examined the Arabic artifacts remaining after using Facebook chat. They found that Arabic conversation could not be directly found by searching for the keywords that were used in their controlled experiment, due to the fact there is a lack of Arabic representation in the memory. Therefore, they had to transform the Arabic keywords into Unicode escape character sequences, then use these to search for the keywords.

IV. Methodology

This section describes the design science research (DSR) methodology which was used to identify the objectives. This methodology was made up of four sequential phases as follows:

1) Problem awareness: Criminals tend to share information about their activities through social media applications running on different operating systems such as Windows and Mac OS. As long as the device is not shutdown or restarted, all of the user's activities are preserved on the volatile memory. This makes the volatile memory a rich source of evidence that can help digital forensics investigators. Consequently, this research aims to help investigators to extract valuable information from the volatile memory, whether written in Arabic or English.

2) Data Collection: Two steps were taken in this phase to investigate the problem and make suggestions regarding the stated problem:

- Literature review: As the basis of this research, a conventional literature review was conducted. It investigated the studies into forensics evidence gathered by social media applications to identify a possible gap in the knowledge.
- Online interviews: This study utilised semi-structured online interviews as another source of data. Arabic digital investigators were targeted in this study and a sample size of six was proposed. We received three responses; one from a digital investigator manager, one from a cybersecurity consultant and another from a certified Computer Hacking Forensics Investigator (CHFI). They were asked a number of questions such as:
  1) What are the most frequently used social media applications by criminals?
  2) What are the most commonly used social media applications that provide digital evidence to investigators?
  3) How frequently do you analyse the volatile memory?

Therefore, the aim was to develop a tool that is aligned with the investigative requirements and facilitates the process of analysing the volatile memory for digital investigators.

3) Development: : A WormHex tool was developed to inspects memory dump files of suspect devices and presents data that has forensic value.

4) Results: : All of the findings were reported in an organised and detailed manner. Moreover, the significance of the findings was discussed in depth in Section VI.

V. Experiment

This section presents the experiment conducted. It addresses the design of the experiment and its execution.

A. Experimental Design

The following are the specifications of the workstations, forensics acquisition and analysis tools, as well as testing accounts.

1) Workstations:

- PC -1
  - OS: Mac OS Catalina - version 10.15.5 64-bit
  - Processor: 2.4 GHz Dual-Core Intel Core i5
  - Memory: 8 GB
  - Browser: Google Chrome - version 86.0.4240.193
- PC -2
  - OS: Windows 10- 64 bit
  - Processor: Intel(R) Core(TM) i7-1065G7
  - Memory: 16 GB
  - Browser: Google Chrome - version 86.0.4240.198

2) Forensic Acquisition and Analysis Tools:

- OSXpmem tool - Version RC 1
- Belkasoft Live RAM Capture - Version 1.0
- Bless Hex-editor - Version 0.6.0
- HxD Hex-editor - Version 2.4.0.0
- Python - Version 3.9.0

3) Testing Accounts: To exchange data via social media applications in a controlled scenario, several testing accounts were created. In the case of WhatsApp and Telegram, one SIM card was used for registration. As for Twitter, a fake email account was created.

B. Experimental Operation

This section describes how the experiment was conducted. It discusses the running of social media applications, the passing of the unique data from the controlled scenario through the application, memory acquisition, the dump file extraction of known data coming from a controlled scenario using basic string search, pattern identification, building, the testing of REGEX and the presentation of evidence. Fig. 1 illustrates the overall operation of the experiment.

1) Running Social Media Applications: This phase aimed to run the social media applications on the target workstations (PC1 and PC2). WhatsApp, Twitter, and Telegram web-based applications ran on a Google Chrome web browser. Once the applications were running, several processes were created in the memory.

2) Passing Unique Data through the Applications: For each web-based application running on the workstation, specific actions were performed, as shown in Table I. These actions were repeated several times to help identify common patterns written into the memory. In addition, data chosen by the researchers were exchanged including, but not limited to, text, files and pictures. The texts were written in both Arabic and English. To help exclude irrelevant strings written into the memory comes from other application running in the memory, a controlled scenario was considered. Consequently, a list of unique strings were proposed by the researchers and passed through WhatsApp, Telegram and Twitter. For example, we sent a message to the WhatsApp account saying *"congratulationscongratulations"*. Another message saying *"Hi twitter"* was posted to the twitter account. This is helpful to distinguish between social media application patterns residing in the memory. This will be discussed in the following subsections in more detail.

3) Memory Acquisition: For the Mac-based workstation, the OSXpmem tool was used. This is an open-source OSX Memory Imager used to reliably extract the entire content of a computer's volatile memory on an Intel-based Mac. Before using OSXpmem, System Integrity Protection (SIP) was disabled. The SIP technology was designed to prevent malicious software from potentially modifying protected files or processes on the Mac, including the memory. Thus, reading the memory with the SIP enabled prevents memory acquisition, even with root-level privilege. As a result, it was essential to disable SIP, reboot the device, acquire the memory and then enable SIP for further processes.

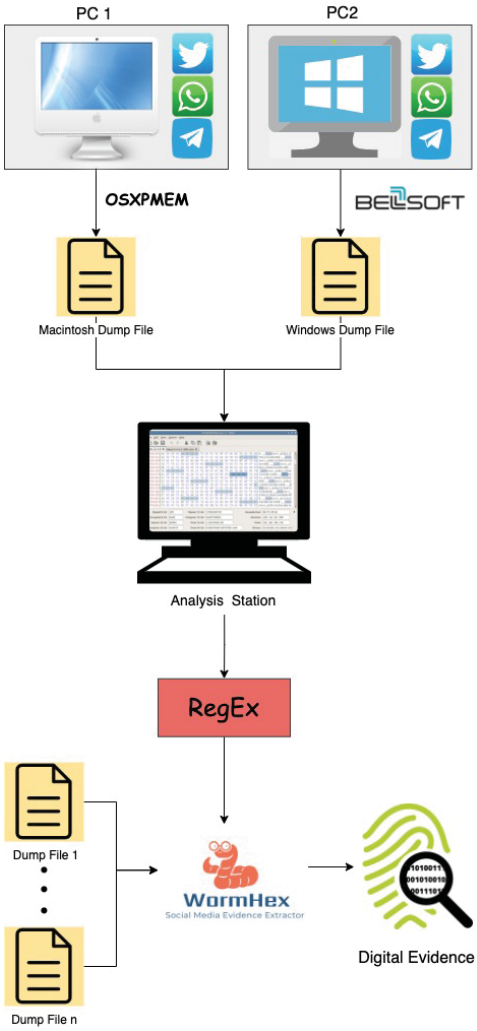

Fig. 1 Experiment Operation

As for the Windows OS, Belkasoft Live RAM Capturer was utilised. This is a lightweight free forensic tool that allows the physical acquisition of volatile memory. Unlike other computing tools (e.g. FTK by AccessData), Belkasoft operates on a system's kernel mode. Running the memory acquisition tool on the system's kernel mode makes it possible for the tool to overcome anti-dumping systems and, consequently, it can reach protected areas of memory [5]. The results of the Mac and Windows memory acquisition were stored as image files, called "dump files". There are different formats of memory dump such as the proprietary format and Advanced Forensics Format (AFF). In this experiment, we used the raw format, which writes bit-stream data to files. In the raw format, data are also transferred more quickly, minor data read errors in the memory are ignored and most computer forensics tools can read it. The dump file of the MAC was 8.5 GB and was 18.7 GB for the Windows computer.

4) Extraction of Selected Data from Dump File Using String Search: During this phase, a comprehensive analysis of memory dumps was performed. This enabled the retrieval of the unique data passed through the applications, as shown in Figs. 2 and 3. There are several hex editor tools that exist, including "Bless", which is known for its ability to read huge amounts from the raw memory and perform search operations. Using hex editors, we searched for the unique strings that passed through WhatsApp, Twitter and Telegram. If one search string showed several results, all of the results were extracted and stored in files for further analysis.

TABLE I
SOCIAL MEDIA APPLICATION ACTIONS

Application	Actions
WhatsApp	1. Login to WhatsApp 2. Communicate with a user/group - Send a text - Record an audio - Capture a picture & record a video - Send a photo & video from library - Send a document - Send a contact - Favorite a chat 3. Call 4. Video call 5. Send a broadcast 6. Add a status
Twitter	1. Login to Twitter 2. Post a textual tweet 3. Send a direct message to a user 4. Follow and unfollow users 5. Favorite, retweet, quote, and reply to a tweet 6. Mention someone 7. Search
Telegram	1. Login to Telegram 2. Communicate with a user - Send a text - Record an audio - Capture a picture & record a video - Send a photo & video from library - Send a document - Send a contact - Delete a chat 3. Call 4. Video Call 5. Send a broadcast 6. Search

Fig. 2 Extracting the unique data "congratulationscongratulations" from dump le

Fig. 3 Extracting the unique data "Hi twitter" from dump le

5) Pattern Identification: Each social media application has its way of writing data into the memory. Some applications leave structured data, while others leave unstructured data. This phase of the study aimed to identify the similarities a single action of a social media application leaves in the memory. For instance, writing tweets via Twitter leaves tweet information in the memory, including the tweet ID, creation timestamp, full text, retweet count, favorite count, reply count and quote count. Figs. 4 and 5 demonstrate how English and Arabic tweets are written into the memory, respectively.

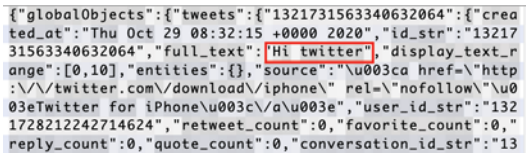

Fig. 4 Memory Dump of English tweet - Twitter

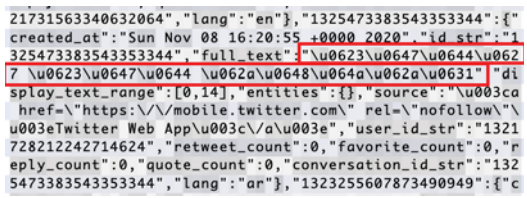

Fig. 5 Memory Dump of Arabic tweet - Twitter

6) Building REGEX: Based on each pattern recognised from the previous phase, regular expressions were built for each action in the social media applications. REGEX enables digital investigators to retrieve information directly from any physical memory dump file acquired during a crime scene, without going over the previous phases. The identified patterns in the memory dump contain information useful to digital investigators (e.g., message-ID, message text and timestamp) and may also contain useless data that may not contribute to digital crime analysis (e.g., constant values). The REGEX was built to extract useful data and filter out all of the irrelevant data. Fig. 7 shows a REGEX written in Python to extract tweets written in English and Arabic. In the case of WhatsApp, Fig. 6 presents a REGEX that can be used to extract mobile numbers appearing in the memory dump file. To make REGEX forensically acceptable, the results of the REGEX must be reproducible. This means that the same results are obtained if the investigation is repeated by another person using the same memory dump files.

```
# Whatsaap: Extract Mobile Numbers (Regex)
Mobiles = re.findall(r'\d{12}@s.whatsapp.net',mem)
```

Fig. 6 REGEX: Extraction Mobile Numbers - WhatsApp

```
tweets_regex = re.findall(r'"full_text":.*?,',mem)
tweets = [re.sub(r'"full_text":','',i)  for i in tweets_regex]
print (tweets)
```

Fig. 7 REGEX: Extraction Tweets (English, Arabic) - Twitter

7) Testing the Tool: The tool was run on several memory dump files to ensure it produced the results expected. It has been shown that the tool was able to present valuable information from any memory dumps.

VI. Results and Analysis

This section discusses the results of this experiment and interprets the significance of the findings. After conducting the empirical experiment, it was found that both Twitter and WhatsApp leave a large amount of data in the memory. Twitter data are fully structured, in the sense that the user profile and tweet information are written in a highly organised format. WhatsApp data, on the other hand, are semi-structured. This means that some data (e.g., mobile numbers) were immediately recognisable, while other data (e.g., sent files) were much harder to identify. As for Telegram, it discloses a limited amount of data and has no pattern that can be used to identify data. Table II presents the retrieved social media information from the memory dump files examined for this study.

It is worth mentioning that non-secret chats in Telegram are scattered throughout the memory, while secret chats are not shown at all. One factor that contributes to this observation is that the dumped memory files were acquired straight after the self-destruction of secret chats. The self-destruction timer varies from one second to one minute. It can be extended to one hour, one day, or even one week. As soon as the time runs out, the sent message disappears from both devices (i.e. sender and receiver devices) [10]. Even if the secret chat leaves a footprint in the memory, the digital investigator might not be able to extract it, as this depends on the suspect's self-destruction timer.

The analysis shows that social media web-based applications use different formats when writing Arabic data into the memory. In the case of Twitter, it was found that it writes Arabic texts as Unicode escape character sequences, which is not a human-readable format (see Fig. 5). WhatsApp, on the other hand, writes Arabic texts in hex decimal values. Although, hex decimal values were found in the dump files (i.e., by searching the keywords used in the experiment), there was no consistent pattern to build the REGEX adequately.

A popular belief regarding Windows and Mac OS devices is that there are many differences between these operating systems. However, in this experiment, it was found that they share one similarity, which is that the volatile memory of Windows and Mac OS preserves the same amount and type of data for all of the tested applications. Moreover, all of the WhatsApp and Twitter patterns found in the dump memory were identical for both systems. Therefore, regarding the volatile memory, there is no obvious differences between Windows and Mac OS devices. This proves the fact that Windows and Mac OS devices are identical in terms of multiple internal components such as the Intel processor.

VII. Conclusion

Volatile memory forensics is greatly important to digital investigators. This study aids digital investigators in general,

TABLE II
EVIDENCE EXTRACTION

Application	Data Extracted	RegEx
WhatsApp	Phone number	Mobiles = re.findall(r'12@s.whatsapp.net',mem)
	File names sent or received	Files = re.search(r'.*@s.whatsapp.net(.*)application/pdf(.*).pdf',mem)
Twitter	Tweet text	tweets_regex = re.findall(r'"full_text":.*?,',mem)
	Account's user name	name_regex = re.findall(r'"name":".*?,',mem)
	Account's screen name	screenName_regex = re.findall(r'"screen_name":".*?,',mem)
	Account creation	Account_regex = re.findall(r'"created_at.*?,',mem)
	Followers count	followers_regex = re.findall(r'normal_followers_count.*?,',mem)
Telegram	Nothing	Nothing

"mem" stands for memory dump files.

and investigators who deal with cases related to the Arabic community. The tool developed for this study, WormHex, can inspect memory dump files on a suspect's device and present the data that have forensics value [3]. This study focused on the analysis of the volatile memory of Windows and Mac OS devices, and shows that social media applications differ in the way they represent data in their memory. In short, Telegram data are scattered in the memory, making it unstructured and hard to understand. On the other hand, WhatsApp and Twitter were revealed to have more structured data in the memory, thus helping investigators to distinguish the pattern, and, in turn, extract evidence by building REGEX for each evidence pattern. Regarding Arabic text in the memory, Twitter writes it as Unicode escape character sequences, whereas WhatsApp presents it in the form of Hex decimal values. In addition, the results verified that there is no significant difference between these devices regarding their volatile memory. For future work, this experiment could be extended to extract more valuable information from WhatsApp and Twitter. Also, the extracted mobile numbers from WhatsApp could be linked with a telecom company (e.g., STC) to identify the names of the owners of specific numbers.

ACKNOWLEDGMENT

The authors would like to thank Mubarak Alshahrani for his support, which greatly added to the project's success.

REFERENCES

[1] Al Mutawa, Noora, Ibrahim Baggili and Andrew Marrington. 2012. "Forensic analysis of social networking applications on mobile devices." *Digital investigation* 9:S24–S33.

[2] Al Mutawa, Noora, Ibtesam Al Awadhi, Ibrahim Baggili and Andrew Marrington. 2011. Forensic artifacts of Facebook's instant messaging service. In *2011 International Conference for Internet Technology and Secured Transactions*. IEEE pp. 771–776.

[3] Alqarni, Amani, Wadha Almattar and Norah Almubairik. 2022. "WormHex.".
URL: *https://github.com/amaniaq/WormHex*

[4] Barradas, Diogo, Tiago Brito, David Duarte, Nuno Santos and Luís Rodrigues. 2017. Forensic Analysis of Communication Records of Web-based Messaging Applications from Physical Memory. pp. 43–54.

[5] Belkasoft. 2020. *Capture Live RAM Contents with Free Tool from Belkasoft*.
URL: *https://belkasoft.com/ramcapturer*

[6] Forte, Dario. 2008. "Volatile data vs. data at rest: the requirements of digital forensics." *Network Security* 2008:13–15.

[7] Hoog, Andrew. 2011. *Android forensics: investigation, analysis and mobile security for Google Android*. Elsevier.

[8] Nisioti, Antonia, Alexios Mylonas, Vasilios Katos, Paul D Yoo and Anargyros Chryssanthou. 2017. You can run but you cannot hide from memory: Extracting IM evidence of Android apps. In *2017 IEEE Symposium on Computers and Communications (ISCC)*. IEEE pp. 457–464.

[9] Sadeghi, Behrouz. 2015. *Guide to Computer forensics and investigations*.

[10] Telegram. N.d. "Telegram Privacy Policy." https://telegram.org/privacy.

[11] Thantilage, Ranul and Neera Jeyamohan. 2017. A volatile memory analysis tool for retrieval of social media evidence in windows 10 OS based workstations. pp. 86–88.

[12] Thantilage, Ranul and Nhien-An Le-Khac. 2019. Framework for the Retrieval of Social Media and Instant Messaging Evidence from Volatile Memory. pp. 476–482.

[13] Vömel, Stefan and Felix C Freiling. 2011. "A survey of main memory acquisition and analysis techniques for the windows operating system." *Digital Investigation* 8(1):3–22.

[14] Walnycky, Daniel, Ibrahim Baggili, Andrew Marrington, Jason Moore and Frank Breitinger. 2015. "Network and device forensic analysis of android social-messaging applications." *Digital Investigation* 14:S77–S84.