FTI Proceedings: 4th international conference on computer applications and information security (iccais'2021) / March 19 / 2021/ Tunisia: https://fti-tn.net/iccais-2021-list-of-papers

Telegrip Forensic Tool

Norah Alkhathlan[1], Nourah Bin Fhaid[2], Deema Almassary[3], Sara Aldossary[4], Rowida Bajuifer[5], Rami Mohammad[6]

Department of Cybersecurity and Digital Forensics
Imam Abdulrahman bin Faisal University
Dammam, Saudi Arabia

E-mail: ([1]2170005844@iau.edu.sa, [2]2170002981@iau.edu.sa,[3]2170003481@iau.edu.sa, [4]2170006432@iau.edu.sa, [5]2170003168@iau.edu.sa, [6]rmmohammad@iau.edu.sa)

Abstract— The use of social media applications is growing rapidly worldwide which is driven by the growth of usage of mobile devices since it has changed the way we live our lives significantly. However, these applications are being used by criminals to help them in conducting cybercrimes. Which makes a significant need for forensics tools that provide features in which the digital evidence can be preserved and presented in a clear and factual manner. In this paper, we are presenting an overview of Telegram forensics along with the tools that can be used to perform data acquisition and analysis along with their features, methods, and limitations. Furthermore, we propose Telegrip, a Python-based forensic tool that aims to analyze image files, preserve evidence related to the Telegram application while maintaining the integrity of the evidence gathered and reports produced. Telegrip will provide several features that will overcome the limitations in the previous tools and assist digital investigators to extract and analyze artifacts generated on Android mobile phones by Telegram easily by using an interactive graphical user interface.

Keywords— Digital forensics; Telegram forensics; Social media forensics; Social networking; Physical acquistion; Android forensics.

I. INTRODUCTION

With the continuous advancement and new innovative designs, mobile devices are becoming compulsory in everyone's daily life since they provide numerous benefits to their users. Despite the fact that mobile devices have a significant positive impact and have made our lives much flexible and easier, cybercriminals are targeting mobile devices in committing their crimes efficiently leveraging the features of the messaging applications. WhatsApp, Telegram, and Facebook are examples of these applications which ease the communication of people's lives. However, cybercriminals utilize the security features of the applications such as anonymity, end-to-end encryption, and secret chats to conduct their crimes. Therefore, Telegram has become the new destination where all cybercriminals take advantage of its features in order to conduct illegal activities and exchange information related to cybercrimes. The security and anonymity feature of Telegram such as hosted chat groups known as "channels" are considered to be a reliable source to broadcast and to communicate privately in planning and conducting crimes. However, in Telegram forensic tools, there is a limitation on retrieving the deleted messages from Telegram application, therefore, this limitation has become a big challenge facing the investigator in terms of collecting evidence. Thus, there is a need for a Telegram forensic tool that will be capable to analyze information that was extracted from the suspect's device and help the digital investigators to draw reasonable conclusions by making the investigation process simple and efficient.

II. BACKGROUND

Instant messaging applications reduce the difficulties to communicate with people in many different regions. These applications support a Graphical User Interface (GUI) and solve the time conflicts since the user has to send the message and the recipients will reply any time they want. These applications were first introduced in the mid-1990s. Furthermore, in 2010 many instant messaging applications added additional features in the social networking services such as Myspace, Facebook, and Twitter and one of the most well-known IM applications is Telegram [1]. Telegram for Android was officially launched in October 2013 as an instant messaging and VoIP cloud-based application. Telegram application supported secret chats with self-destruct timer functionality in the first version of the application in 2013, where the user can set a timer, so the countdown begins at the moment the second party opens the picture or video sent by the first party, which will make the picture/video vanish permanently after time runs out. Moreover, Telegram has provided a feature of enabling its users to access their accounts from multiple devices which distinguished Telegram from other chatting applications at that time, as the Telegram vendors claim that it is more secure with its multi-data center architecture and encryption than other instant messaging applications. Moreover, it allows different types of secured individual communication which can be either textual nontextual messages, or voice calls [2]. These features which are supported by Telegram offer a great way to communicate for people even as cyber-criminals which helps them to perform their plans.

III. RELATED WORK

This section situates Telegrip tool with respect to previous studies and findings. It objectively analyses and assesses results, hypotheses, and procedures that are related to our scope and our focus areas.

A. Android Data Acquisition

The Android operating system is a very popular operating system for mobile phones. It provides a sophisticated service of communication over the internet providing the users with the ability to browse the internet, exchange text messages and emails, and much more. Linux kernel 2.6 developed in the Android operating system as the building block and the core. The Linux system default hard drive is /dev/hd0. While the interface operation between the Linux kernel and the flash devices is the memory technology device (MTD). In the device's memory is where the files of the system, libraries, and configurations are stored. For the application's data and databases, they are stored in the internal memory. The backup files are stored in the SD card as an external storage [3].

Collecting data and evidence from Android devices can be done using multiple methods. Manual Acquisition can be done where the examiner browses the devices capturing needed information without the need of any tool. Furthermore, a physical Acquisition can be done where the device is directly connected to the investigator device to conduct a bit-by-bit copy of the device, this technique clones deleted data as well as the unallocated spaces [4]. On the other hand, Special rooting scripts or tools can be used to gain the privileges of root user on the device. Dr.fone root is a software that can be used by installing it in the computer and connect the investigated device to it. Moreover, the Logical data acquisition technique is the device logical storage copy, the extraction of data exists in the device done by gain access to the file system. The logical acquisition can acquire SMS, media, call history, system logs, and application data. Adb command can be used to extract the data needed or analyzing the backed-up data by performing the adb backup command. Also, AFLogical OSE tool extracts much information from a USB debugging enabled Android devices such as contacts, callLog, and SMS messages [5]. In [6] a logical data acquisition framework for Android devices was proposed named LASM. The framework is based on the data migration concept which means the transmission of the backed-up data from one mobile device to another. LASM uses an intermediate rooted device to gain access to the information of the unrooted targeted device following it by the data acquisition process in a forensic platform that uses the ADB tool.

B. Android File System

[7] examines a process which uses internal (NAND) memory of Sony Ericson Xperia x10i to physically and logically extract the data and analyze the Android file system YAFFS (Yet Another Flash File System) using various methods for data extraction. Various tools were used for data extraction. Then the extracted data are analyzed to determine the best and most effective method for recreating the internal (NAND) memory. YAFFS is used in NAND flash devices as a journaling file system. NAND memory is separated into chunks and blocks. Chunks are further comprised of Object Header, which contains information relating to any file. Thus, in order to rebuild and recreate the file structure, Object Header information is required. Prior to collecting the data from Android devices, superuser access is required which is gained by rooting the device. Android Debug Bridge (adb) tool was used for obtaining a logical image of the Android device. The pull command in adb copied the required data files from the device. The logical image thus obtained extracted only about 85% of data files, neglecting mainly system files. A physical image, on the other hand, is a bit-by-bit copy that pertains to the internal flash memory. Flash memory is divided into partitions that are managed by MTD (Memory Technology Device). The following adb command was executed, in order to view the MTD partitions:

adb shell cat /proc/mtd.

According to the research in [8], there are six portions of file systems found in Android devices: user data, boot, system, cache, and recovery. Their research showed that YAFFS2 is utilized by many Android-based devices as a file system and newer devices were found to employ the EXT4 file system. Furthermore, the research has found that no overwriting can change content on the device since the recovery partition of the file system does not store any user data when normal operations are performed.

C. Telegram Digital Forensics

Telegram messenger has three essential files; Telegram.apk, directory storage files stored in the phone's local memory that stores documents, images, audios, and videos, and databases of all the activities to be conducted for further testing and review [9].

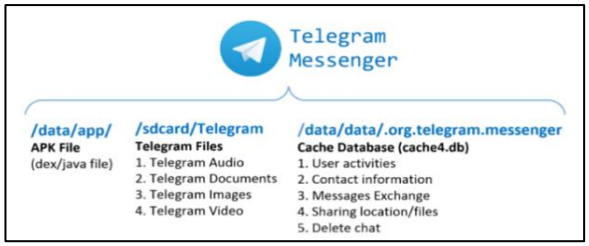

Figure 1. Telegram application structure [9]

Telegram application has many features that were provided for its users such; end-to-end encryption that is done in the secret communication in the chats. Furthermore, Telegram has offered the users an important feature where they can permanently delete their messages not only from their devices but from their remote parties' devices as well even after viewing the messages. These features were provided by Telegram in order to enhance the security and to make it one of the most secure chatting applications, but these features have raised many security concerns since this application can be "a paradise for criminals" [10] since they do not need to use their actual personal information and their phone number are not required. Therefore, criminals can misuse the Telegram application to do illegal activities and take advantage of the anonymity feature.

In order to help the law enforcement and provide the ability to extract Telegram information, Oxygen Forensic® Cloud Extractor has provided the ability to use a phone number or token derived from Android devices to retrieve data from the Telegram cloud which will include contacts, calls, private and group chats, and channels data. Although Oxygen Forensic® Cloud Extractor provides several features that will help digital investigators to extract information from Telegram application easily, digital investigators do not have the ability to extract secret chats that are end-to-end encrypted since this tool uses the cloud for information extraction, so this feature will be missed in this tool which might obstruct the process of digital investigation since these secret chats may contain important evidence [11].

In paper [9] analysis and investigation of Telegram digital forensics were done in order to help digital investigators to interpret the artifacts extracted from the Telegram application. The sequence of activities conducted in this paper was first the installation process for the Telegram application in the smartphone, then the sign-up process by using two online and offline acquisition methods. In the online acquisition, information about the chat text and multimedia including the timestamp and date of the incident was retrieved from adb logcat, which can be used as admissible digital evidence in court. Offline forensics is needed for the investigator in order to collect user records about who signed in associated with the phone number used for the registration process. The "adb backup -all" command was used to acquire offline forensics. As for offline forensics, the files are stored in the same cache4.db database but are shown in multiple tables. Telegram records contact-information in the "user" table, where the values in the "delete" column can be either 1 or 0, where 1 indicates that the contact has been removed, whereas 0 indicates that the user is already stored in the contacts. Furthermore, if a block has been made against the user that is stored in the contacts, the table "blocked users" on cache4.db is populated with the user ID, otherwise, the table would be empty. For the audio files, although the audio files are stored in the Telegram "Audio" folder with the .ogg extension, it can be opened directly if the investigator has an audio player that supports ogg files, otherwise, the investigator can use the converter to convert ".ogg" to ".wav" extension in order to open it with any audio player. Whereas the documents will be stored in the shared "Telegram Documents folder" table.

IV. FINDINGS AND DISCUSSION

The vast majority of our daily activities are carried out through mobile applications, which usually produce and store large data sets on the smartphone which makes the forensic analysis of these data play a vital role during an investigation. However, these applications have also been used by criminals to help them conducting and planning for cyber-crimes. Which makes a huge need for forensics analysis tools that can preserve the evidence and present it in a meaningful matter. While considering the fact that there is no standard path for the program files and these programs often get new updates in short periods. In this paper, we are presenting an overview of different forensic tools and researches about Telegram forensic tools that can be used to perform the data acquisition process along with their features, methods, and limitations. Since the Android operating system has a large volume of data that can be collected and analyzed to conduct and complete the digital investigation process, it was the chosen operating system for the scope of this paper. Although the proposed Telegram forensic tools and methods previously have various features that help digital investigators efficiently to extract information from Telegram application, digital investigators will encounter issues with the limitations of previous methods and tools such as retrieving secret chats which could obstruct the process of digital investigation. Therefore, Telegrip forensic tool has overcome the limitations in the other tools in order to provide the digital investigators with features in which the digital evidence can be preserved and presented in a clear and factual manner.

V. CONTRIBUTION

In this paper, we propose Telegrip GUI-based digital forensic tool that will be capable of extracting detailed information of the image obtained from the investigated Android mobile devices. The tool is developed by Python programming language using Python integrated

development environment (PyCharm) and MySQL database management system. The tool will extract helpful artifacts that will help digital investigators in the investigation process. These artifacts include (user information, messages, photos, videos, and documents. Telegrip aims to accomplish security goals, where each goal addresses different aspects of protection. For Integrity, the investigator will be able to summarize the findings of Telegram-related cases by generating a report. For each generated report a hash value will be calculated in order to strengthen the validation of the digital evidence in court and maintaining the integrity. The hash value will be calculated using the SHA-256 algorithm which considered one of the most secure hashing algorithms.

Furthermore, authentication and confidentiality will be covered by providing password-protected cases which can only be accessed by its creator. Where the case's password will be securely stored by using a salted hash which will create a unique password in order to prevent unauthorized access.

VI. EXPERIMENT

In this paper, an experiment was conducted on the Telegram messenger application. The used smartphone was the Samsung Galaxy S6 edge with version 7.0 (Nougat) of the Android operating system. The acquisition and analysis were done using the Windows operating system to perform physical acquisition. The extracted image represents artifacts of the Telegram version (v7.4.1).

A. Physical acquistion:

Telegram artifacts are located at:

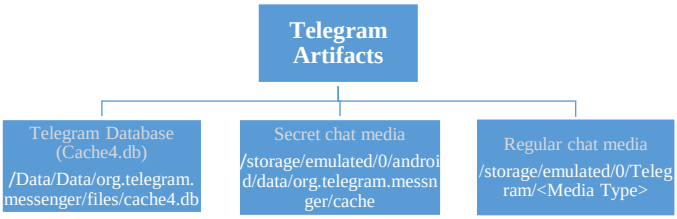

Figure 2. Telegram artifacts

The following commands were used for data acquisition:

```
D:\platform-tools>adb -d shell
zerolte:/ $ su
zerolte:/ # cp '/data/data/org.telegram.messenger/files/cache4.db' '/sdcard/cache4.db'
```

Figure 3. Copying cache4.db from the protected memory to the unprotected

```
D:\platform-tools>adb pull /sdcard/cache4.db
/sdcard/cache4.db: 1 file pulled, 0 skipped. 27.6 MB/s (1294336 bytes in 0.045s)
```

Figure 4. Cache4.db acquision

```
D:\platform-tools>adb pull /sdcard/Telegram app3/
/sdcard/Telegram/: 7 files pulled, 0 skipped. 3.2 MB/s (1418445 bytes in 0.426s)
```

Figure 5. Regular chat media acquistion

```
D:\platform-tools>adb pull /storage/emulated/0/android/data/org.telegram.messenger/cache app4/
/storage/emulated/0/android/data/org.telegram.messenger/ca...files pulled, 0 skipped. 1.8 MB/s
```

Figure 6. Secret chat media acquisition

B. Telegram's database analysis:

After the acquisition of cache4.db, we have carefully chosen the most significant tables to be analyzed in order to extract and parse the artifacts. In this section, the users, messages, enc_chats, and media_v2 tables were analyzed using SQLite DB browser. While analyzing the structure of Telegram database, the following attributes were found in most of the tables:

- Uid: unique user ID
- Mid: unique message ID
- Date: Unix epoch timestamp format
- Data: Content and other information stored in a binary large object BLOB

a) Messages table

Telegram exchanged messages are stored in the Messages table. One of the distinguished columns in this table is (out) which contains the value '0' for a received chat and '1' for a sent chat. Furthermore, values of (mid) and (uid) can be used to determine if the message was in a group, channel, or secret chat. For example, if mid has a big positive digit value and uid has a negative value we can state it is a channel. While if we have a negative mid-value and a negative or positive uid-value, it is a secret chat.

b) Enc_chats table

Telegram messenger stores information of secret chat in table enc_chats. Some of the important information related to enc_chat are: (user) the secret chat partner, (name) the name of the other party along with the username, and (admin_id) is the id of the user who initiated the secret chat.

c) Media_v2 table

It stores Mid, Uid, type, and the data field. The type field specify the media type which has different values: type 0: Image (jpg)/ Video (mp4), type 1: Document, type 2: Audio (.ogg), type 3: URL, type 4: music, and type 5: GIF (mp4).

d) Users table

Users' information is stored by Telegram in the "users" table where it contains (name) username and nickname, (status) last seen status with Unix epoch timestamp format, and (data) which contains the full name (name and username) of the user along with the phone number if exists.

The signature of each user's data record is 'c1 58 84 93' which is located at the first significant four bytes at offset 00x0.

C. Analysis:

In cache4.db, Telegram stores user ID in the secret messages in a different form than the real user ID. In order to find the identity of the user who conducted the secret chat, a conversion needs to be done from decimal to hexadecimal. From this result, the uid in the enc_chats table is found as the calculated number. Then, from the

enc_chats table, the "user" number from the same record where the uid was found, and therefore this number will be linked to the uid in the users' table in order to find the identity of the user who conducted the secret chat.To find the exchanged secret chat we used a signature-based method to look for the signature of secret chat that is 'fa 55 55 55' as shown in fig. 7 which can be located at the initial four bytes at offset 0x00 of the data field in the messages table. The secret messages can be seen as plaintext from the hex editor. After verifying the signature of the secret chat, parsing the data blob takes place. Subsequently, the parsed data can be used to extracts the secret chat messages by joining the recodes of enc_chats with the messages table.

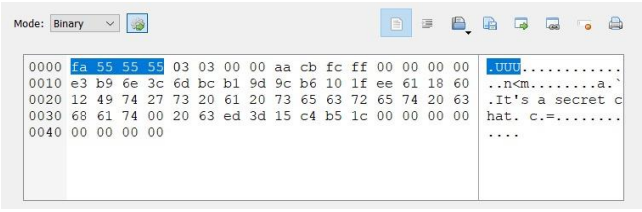

Figure 7. Secret chat signature

VII. CONCLUSION

The security and anonymity features of Telegram have not only attracted peers and businesses, but cyber criminals also utilized this application as a go-to-market for their criminal activities. Telegrip forensic tool aims to benefit the investigator in the digital investigation process of Telegram-related cases. It is intended to be used in extracting information from the Telegram application by providing the victim's or suspect's device image. Numerous functionalities Telegrip has provided to aid the investigator in analyzing Telegram crime cases in an effective way, such as examining the acquired image to extract needed information, searching for evidence of interest, and generating a report for the case. Several experiments will be conducted to assess the capacity of the proposed tool and the results will thoroughly be analyzed to come up with generalized conclusions and findings that could be inspiring to the researchers in the field to continue the road that this paper will present.

ACKNOWLEDGMENT

We would like to express our sincere gratitude to Mr. Maksym Boiko, and Mr. Ali Alwashali for their valuable guidance and support through sharing their knowledge and efforts which were a valuable motive to us to complete this project.

REFERENCES:

[1] Hintea, D., Sangings, A., & Bird, R. (2018). Forensic Analysis of the Telegram Instant Messenger Application on Android Devices. ECCWS 2018 17th European Conference on Cyber Warfare and Security (p. 217). Academic Conferences and publishing limited.

[2] The Evolution of Telegram. (2020). Retrieved from Telegram.org: https://telegram.org/evolution

[3] Rao, V., & Chakravarthy, A. (2016). Forensic Analysis of Android Mobile Devices. IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2016). Jaipur, India: IEEE.

[4] Roy, N. R., Khanna, A. K., & Aneja, L. (2016). Android Phone Forensic: Tools and Techniques. International Conference on Computing, Communication and Automation (ICCCA2016) (pp. 605-610). IEEE.

[5] Boueiz, M.-R. (2020). Importance of rooting in an Android data acquisition. 2020 8th International Symposium on Digital Forensics and Security (ISDFS). IEEE.

[6] Feng, P., Li, Q., Zhang, P., & Chen, Z. (2018). Logical acquisition method based on data migration for Android. Elsevier, 55-62.

[7] Quick, D., & Alzaabi, M. (2011). Forensic analysis of the android fifile system YAFFS2. Australian Digital Forensics Conference. secau Security Research Centre, Edith Cowan University, Perth, Western Australia.

[8] Vidas, T., Zhang, C., & Christin, N. (2011). Toward a general collection methodology for Android devices. Digital Investigation, 14-24.

[9] Satrya, G. B., Daely, P. T., & Arief, M. (2016). Digital Forensic Analysis of Telegram Messenger on Android Devices. International Conference on Information, Communication Technology and System (ICTS), (p. 7).

[10] Kochi. (2019, November 25). Telegram app is paradise for criminals, police in HC. Mathrubhumi.

[11] Oxygen.forensics. (2020, May 19). Telegram Forensics. Retrieved from blog.oxygen-forensic: https://blog.oxygen-forensic.com/telegram-forensics/