2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering

Framework for the Retrieval of Social Media and Instant Messaging Evidence from Volatile Memory

Ranul Deelaka Thantilage
School of Computer Science
University College Dublin
Belfied, Dublin 4, Ireland
ranulthantilage@gmail.com

Nhien-An Le-Khac
School of Computer Science
University College Dublin
Belfied, Dublin 4, Ireland
an.lekhac@ucd.ie

***Abstract*— The human society today has had to be confronted by the threat posed by criminals who are taking the advantages of social media and instant-messaging apps to conduct their criminal activities. In literature, there are many approaches for forensic acquisition and analysis of these apps. However, most of the approaches mainly focus on non-volatile storage media. Meanwhile, social media and instant-messaging platforms do not usually store information relating to social transactions in non-volatile bases. Hence, in this paper, we propose a framework for volatile memory forensics that is essential for the ready retrieval of evidence. Our approach is based on the retrieval of social media and instant-messaging evidence through the use of string search mechanism extended by the usage of regular expressions and the functionality of match groups. Our approach has been tested and proved to be effective with different social media apps on two popular operating systems: Windows and Mac.**

Keywords—Digital Forensics, Volatile Forensics, Forensic Framework, Social Media apps forensics, Instant Messaging forensics, Evidence Retrieval, Windows Forensics, Mac Forensics

I. INTRODUCTION

The increase of internet users globally has caused social media and instant messaging to be widely used at individual level. Statistics by Digital 2019 show over 3 billion social media users [1]. These users resort digital devices for the exchange of information. Instant messaging platforms are similar to social media but aim at creating individual or group conversations. These conversations might lead to important discoveries while solving criminal cases [2][12]. It may not be only plain conversations between two or more users. It might be the exchange of ideas or even planning of a criminal mastermind. For example, the chat history of a computer of a kidnapped child might be the only source of information an investigator can get hold of. The social accounts might be signed off, but if the computer is running the Random Access Memory (RAM) might hold the evidence needed. In an instance where a criminal mastermind plot its next attack and the law enforcement finds his/her safe house with a running computer. The history of the web browsers can be erased but the RAM will hold the evidence needed. Whether chat history or even location search on map providers such as Google, it will aid in identifying the next target location(s) [3]. The RAM can therefore be considered as 'home' to very critical evidence that might not be found elsewhere on a system. In addition, the RAM also holds information related to user credentials and log-in information to the local system or even social accounts. Hence, password hashes and other un-encrypted information can be located by volatile memory forensics.

Volatile forensics can therefore be considered a key sector in digital forensics when it comes to social media and instant-messaging (IM) based evidence retrieval. There are not many tools that specialize in in-depth social media analysis. Out of these most are very expensive commercial tools that are not accessible by most officers at law enforcement agencies. Even the commercial tools do not analyze all major social media and IM platforms.

Hence, in this paper we propose a framework for volatile memory forensics to acquire and analyze artefacts from social media and IM platforms. Our approach taken is divided into several key phases namely, exploring of RAM based in different operating systems, memory acquisition techniques, following on how applications write data to RAM, recognition of patterns in data written to RAM and building regular expressions based on these patterns.

We also evaluate our approach to extract evidence from a wide range of social media and IM apps installed on two popular operating systems: Windows OS and Mac OS. The experiments returned positive results of retrieving and analyzing evidences.

Law enforcement officers do not always get access to high-end expensive tools. This paper contributes to aiding them to self-analyze such social media and IM platforms to counter the efforts of unruly segments of society that disturbs social tranquility. Volatile memory forensics connecting with social media and IM is a branch of study not researched extensively due to its challenging background. Therefore, this research will also contribute to show scholars that there are patterns and similarities even in the physical memory of a system, which would open fresh research concepts to further study on. This research is therefore dedicated towards individuals and agencies involved in law enforcement and academics pursuing the discipline. The end-aim of all their cumulative effort is to achieve the hope of establishing peace, understanding, tranquility and goodwill in the troubled global human society. The rest of this paper is organized as follows: Sections II describes the research background and the review of related work in literature. We present the challenges of volatile forensics in Section III. We describe our proposed framework for volatile memory forensics in Section IV. We then evaluate our framework using different test cases in Section V. Finally, we present our conclusion and future work.

II. RELATED WORK

Volatile forensics also known as Live Data Forensics refer to the analysis of evidence from volatile sources of a computer system. Such data usually will not be available after the shutdown of a system. This data might also be consistently changing depending on the running processes. Live Data Forensics include the analysis of data in the RAM of a system as well as Page-file, Crash Dumps and Hibernation Files [4].

2324-9013/19/$31.00 ©2019 IEEE
DOI 10.1109/TrustCom/BigDataSE.2019.00070

Volatile data analysis can be considered difficult as this data is constantly changing. But such data is very important at an investigation and the RAM might be the only source where such data is stored. An investigator cannot exactly define that certain data will be found on a certain region in the RAM. Therefore, analysis of the RAM requires a greater level of expertise by the digital forensics examiner. At this point volatile memory forensics may aid the investigation. Such keys might be stored in volatile memory. Since the user would have entered the keys using the system there is a higher chance of the investigator finding the encryption key in the RAM.

Over the past decades, researchers have been conducted on the analysis of social media and instant messaging platforms for forensic evidence [12][19]. This is mainly due to the huge increase in the usage of such applications. Most research publications related to social media and instant messaging tend to focus on individual applications and mainly non-volatile means of forensics. Therefore, when compared with existing research, our contribution has a much wider scope since it is a common framework that goes beyond borders allowing the retrieval of evidence from major social media and instant messaging applications running on leading operating systems.

Memory acquisition and analysis techniques for Windows based operating systems where discussed by Stefan and Felix in 2011 [5]. The research focused mainly on analysis of files, cryptographic keys, network, process, system registry and applications.

Analysis of Android smartphones was discussed by Anglano in 2014 [6]. The research was focused only for WhatsApp Messenger application. It provided a clear description of the artifacts generated by WhatsApp Messenger.

Wouter in 2007 [7] focused on forensic artifacts left by Windows Live Messenger 8.0. This research was only focused for this application which was commonly known as MSN Messenger and was famous for its usage in Windows XP.

Facebook IM and Skype related forensics was researched in 2016 [8]. The research examined on both volatile and non-volatile means but only focused on Windows Store based installations of the two application on operating systems after Windows 8.1.

Additional research on Facebook IM was conducted [9] which shows the process of recovering and reconstructing the evidence left by the chat application. It only focused on evidence stored in the hard drive (non-volatile).

Yusoff, Ali and Ramlan [10] researched on forensic evidence retrieval of three social media and three instant messaging services in 2017. Their research was based only for mobile phones running Firefox OS.

Forensic analysis of Line Messenger was conducted by Ming and Chih [11]. It discussed the evidence gathering from the application running on Android OS in mobile devices.

Sgaras et al. [12] researched on forensic acquisition and analysis of instant messaging applications. This research focussed mainly on Tango chat application on both iOS and Android platforms. Discussions were also made on IM chat cloning and communication interception. Further the results were compared with two other IM applications, WhatsApp and Viber.

In 2008, Matthew, Shira and Marcus [13] researched on volatile based instant messaging platforms. Their research focussed on web browser-based chat applications only. They concentrated mainly on four IM applications, AIM Express, Google Talk, Meebo and E-Buddy for their tests.

Research on string search was taken forward during The Digital Forensic Research Conference held in Pittsburgh 2007 when Nicole Lang Beebe and Jan Clark proposed a post-retrieval solution. This solution allowed strings of text to be clustered by a self-organizing neural network [14].

The existing literature shows that each is based on a particular operating system, application platform or technique only. Also, most research is focussed on mobile device-based applications. The forensic evidence retrieval shown on most literature is based on non-volatile storage mediums. Therefore, it is much understood that there is a gap for a common framework tested and proven to retrieve evidence from mostly used social media and instant messaging applications on PCs through the analysis of volatile memory.

III. Challenges

Volatile forensics has its own set of challenges that an investigator should be cautious about [15]. While acquiring a memory dump the most practised mechanism would be software-based acquisition. At this stage the tool used to create the dump file will also run as a process in the system, which will in-turn add data to the RAM. This can even override crucial evidence at an investigation. Therefore, it is a must to choose a tool with very least memory footprint for the process of memory acquisition. Also, the digital forensic examiner should know what data is put into the RAM by the memory acquisition tool, so that he/she can differentiate that data with the actual evidence. The RAM is never the same, it changes constantly. Due to this reason an investigator should make sure to acquire the dump as soon as possible at the crime scene without making any alteration in the system. The examiner should also make sure not to reboot the system as data in the RAM once lost cannot be reproduced. A digital forensic examiner should make sure to use tools that are not pre-installed in the system. This is since the criminal would have already had access to these tools and therefore could have altered the tools in-order to reduce the RAM output or even to give a completely false output to the investigator.

Such challenges does not de-value the evidence retrieved [15]. It is just a matter of knowing for the investigator so that if a need arises an investigator can assure on the data integrity in front of a legal body.

IV. Proposed Framework

Data related to social media and instant messaging get stored in volatile memory in text format. Each application and operating system has its own way of writing data into the volatile memory. Finding the pattern in which the data is written to the memory, is the core approach. Explained below are the nine phases of the proposed framework (Figure 1).

A. Phase One – Running Social Media/IM Application

Social Media and IM applications run in different ways. All major platforms have more than one way of running it. It can be web-based, mobile-based, on Windows Store, installation as an app etc. Therefore, phase one of this framework would be to run the application on the target system.

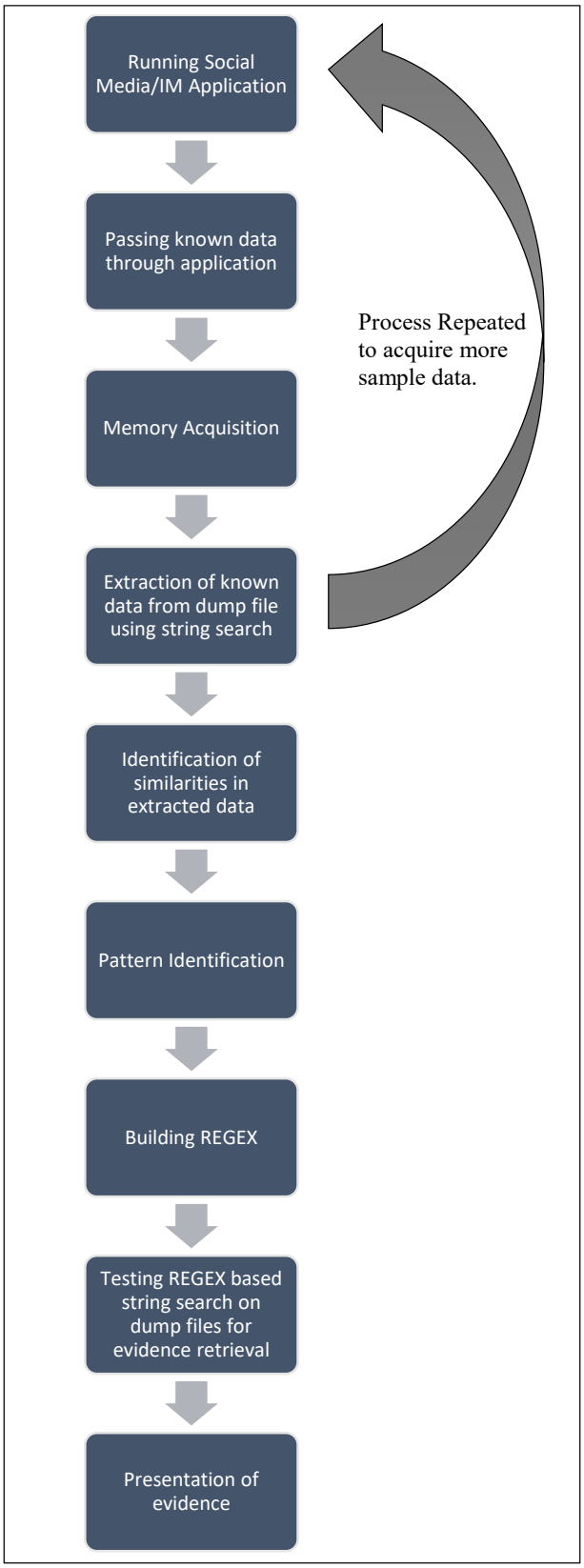

Fig. 1. Proposed framework.

B. Phase Two – Passing known data through application

Once the application is running in the system, known data of choice which would be unique should be exchanged using the Social Media/IM applications. This should be repeated several times under varying conditions and memory acquired in order to achieve best results in future phases. The text string being unique is essential. This would ensure better filtration of results while searching for the strings.

C. Phase Three – Memory Acquisition

Acquiring memory can be done using two mechanisms - either hardware-based or software-based acquisition [15][16]. When compared with hardware-based acquisition, software-based acquisition is cost effective and is available easily and therefore it is what is used by most digital forensic investigators.

Software-based acquisition is when an application tool is used to acquire a memory dump. Software-based memory acquisition creates a bit by bit copy of the RAM. As all other file types, memory dumps also have varying formats. These depend on a range of factors. It can be varying due to operating system or even varying due to the memory capture tool used. Some tools have their own proprietary dump formats as well. We will be creating a dump in format RAW as it is preferred by most digital forensic investigators [17][18].

This would require the use of a target operating system. At current testing purposes, the process is based on Windows 10 and Mac OS High Sierra PCs. These are newer operating system versions of Windows and Mac and therefore will ensure the research stays up-to-date with the current technology. The two operating systems are quite different from each other. Therefore, the memory acquisition process for each would differ.

For Windows OS, the tool DumpIT by MoonSols was used to acquire the memory dump. It is quite popular among investigators as the tool is portable and acquires the dump of the suspect system by running the tool from a removable drive. DumpIT also supports x86 and x64 based systems. As shown by Faiz and Prabowo [20], DumpIT leaves the least footprint in the memory. Therefore, it can be considered an ideal tool for Windows based memory acquisition.

Mac Memory Reader was a popular tool in the early days but now is not in issue. For Mac based workstations a tool that is used mostly today among investigators and academic professionals is OSXpmem. It is a tool developed by the team behind Google Rekall project which is an advanced forensic and incident response framework. The tool is part of the *pmem* suite of the project. This tool will be used for the memory acquisition of Mac based workstations.

D. Phase Four – Extraction of Known data from dump file using string search

This phase requires the analysis of the dump file. There are command-line tools such as 'strings' which is popular for such data extraction. Some hexadecimal readers also support the search of string through the find function. The known string of text passed was searched using the 'string search' technique, a traditional method used in volatile memory forensics [21]. The lines of the dump containing the matching strings of text were then extracted out.

Memory dump files are quite large and therefore this technique had to be facilitated using a tool which supports

large file reading. After a brief on several available tools, EmEditor, developed by Yutaka Emura of Emurasoft Inc., was chosen. This tool supports very large raw files, hexadecimal, ASCII, Unicode and regular expressions.

E. Phase Five – Identification of similarities in extracted data

This is a key phase in the framework. The earlier phases are repeated several times in order to acquire a larger amount of data, to better identify similarities. Each platform would have a similar way of writing data to the RAM. Slight variations can occur due to OS or application version but mostly keywords used to denote the data, headers and footers have similarities.

F. Phase Six – Pattern Identification

This phase can also be considered directly attached to phase five. The identified similarities would aid in recognizing a pattern of how data is written to the RAM by each application. For example, Facebook would have a tag for each data type which will be followed by one another and differentiated using a comma or equivalent symbol as shown in Figure 2.

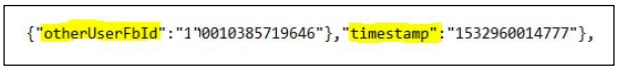

Fig. 2. RAM Dump - Facebook.

G. Phase Seven – Building REGEX

"One of the best traditional ways to do pattern matching is search for a sub string as a substring in a given string by "matches" is called as "Regular Expression" (RE)" [22]. When compared to different pattern-matching algorithms RE provides with a much broader support for complicated pattern-matching. In short it is also known as regex. Regular expressions are "wildcards on steroids" [23]. It describes a search pattern and a special string of text. Regular Expressions can also be used to capture groups of data. For example, by the use of parentheses one can group the regex in between them. These matched groups will be numbered and therefore can be back referenced either for the whole group or the sub groups denoted by parenthesis [24]. Therefore, a REGEX can easily be created for each pattern recognized during Phase Six. The Table 1 shows few of the regex created for testing and the retrievable artefacts.

H. Phase Eight – Testing REGEX based string search on dump files for evidence retrieval

The identified regular expression can then later be used for evidence retrieval directly from the memory dump. This would eliminate searching for known strings and allow the investigator to find forensics artefacts by the usage of the REGEX. This means that unknown data can be retrieved from the dump.

I. Phase Nine – Presentation of Evidence

This is the final phase of the framework and considered very important. This can also be considered similar to report generation. It is essential to keep in mind that the digital forensic investigator at some point will have to present the evidence retrieved to his/her superiors or a court of law (legal body). Therefore, it is a key concern to ensure data integrity at such investigations. The National Institute of Standards and Technology (NIST) has stated two important factors in order to make a digital forensic result acceptable at a court of law [25]. The result should be repeatable, meaning under same tools, lab, method and same person the same result should be generated. Also, the result should be reproducible, meaning under different person, lab and tools yet by using the same method the same result should be produced [26]. Therefore, standardized procedures should always be followed during investigations and evidence should be presented in an appropriate manner.

TABLE I. SAMPLE REGEX

Application	Privileged
Skype in Mac	imdisplayname\".(.*?)\"(?:.*?)messenger.live.com(?:.*?)content\".(.*?)\".composetime\".(.*?)\{(?:.*?)conversationId\".\d*\:(.*?)\" Match Group 1: Display Name Match Group 2: Message Text Match Group 3: Timestamp Match Group 4: Username
Google Maps in Mac	\/maps\/(?:.*?)\/(.*?)\/\@(.*?)\,(.*?)\, Match Group 1: Search Location Match Group 2: Latitude Match Group 3: Longitude
Facebook Messenger in Mac	,\"body\"\:\"(.*?)\"\,\"(?:.*?)\{\"actorFbId\"\:\"(\d*)(?:.*)\{\"otherUserFbId\"\:\"(\d*)\"\}\,\"timestamp\"\:\"(\d*)\"\} Match Group 1: Comment Text Match Group 2: Actor FB ID Match Group 3: Other User FB ID Match Group 4: Timestamp
Viber in Mac	…(.*?)0\{\"tech_info\"\:\{\"seq\"\:(\d*)\} Match Group 1: Message Text Match Group 2: Seq ID
WhatsApp in Mac	^(.*?)ŠÛ Extract Entire Line
iMessage in Mac	(?:.*?)\@(?:.*?)\son\s(\d{4}\-\d{2}\-\d{2})\sat\s(\d{2}\.\d{2}\.\d{2})\.ichat Extract Entire Line Match Group 1: Date Match Group 2: Time
Skype in Windows	\:(.*?)\{\"clientmessageid\"\:\"(\d*)\"\,\"composetime\"\:\"(.*?)\"\,\"content\"\:\"(.*?)\"\,\"messagetype\"\:\"(.*?)\"\} Match Group 1: Username Match Group 2: Client Message ID Match Group 3: Timestamp Match Group 4: Message Text Match Group 5: Message Type
Google Maps in Windows	\/maps\/place\/(.*?)\/\@(.*?)\,(.*?)\, Match Group 1: Search Location Match Group 2: Latitude Match Group 3: Longitude
Facebook Messenger in Windows	\:\{text\:\"(.*?)\"\,(?:.*?)\{id\:\"(\d*)\"\,(?:.*?)\,ttl\:\d\,timestamp_precise\:\"(\d*)\" Match Group 1: Status Text Match Group 2: User ID Match Group 3: Timestamp
Viber in Windows	…(.*?)0\{\"tech_info\"\:\{\"seq\"\:(\d*)\} Match Group 1: Message Text Match Group 2: Seq ID
WhatsApp Web on Windows	(.*?)(\d{11})@s.whatsapp.net Match Group 1: Message Match Group 2: Phone Number
Skype User Credentials on Windows	&login=(.*?)&(?:.*?)LoginOptions=\d&passwd=(.*?)&ps Match Group 1: username Match Group 2: password

V. TESTING AND EVALUATION

This section shows the test results after following the proposed framework on two operating systems.

A. Experimental platforms

The following are the specifications of the PCs used for testing on Windows and Mac workstations.

PC - 1
- Dell Mobile Precision M6600
- OS: Windows 10 Pro 64-bit
- Processor: Intel i7 - x64 based RAM: 20 GB

PC – 2
- Lenova Yoga 500
- OS: Windows 10 Home 64-bit
- Processor: Intel i5 - x64 based RAM: 4 GB

PC - 3
- MacBook Pro
- OS: macOS High Sierra - version 10.3.5 64-bit
- Processor: Intel i7 - x64 based RAM: 16 GB

PC - 4
- MacBook Pro
- OS: macOS High Sierra - version 10.3.6 64-bit
- Processor: Intel i7 - x64 based RAM: 16 GB

The research conducted was mainly focused on evidence related to social media and instant messaging. It can be evaluated, on the understanding, that, all major social media and instant messaging platforms were tested and evidence retrieved. In addition, Google Search engine and Google Maps platforms were tested.

The tested apps and services are Facebook, Twitter, Skype, Facebook Messenger, WhatsApp, Viber, Email Services: Gmail, Yahoo Mail, Hotmail, iMessage (Mac only), Google Text/Map Search, User Credentials.

Note that the concept brought forward with this framework can go beyond borders as its approach is common to other operating systems as well. In addition, Table II below will show a comparison of already available research work discussed earlier in Section II.

TABLE II. COMPARISON OF CURRENTLY AVAILABLE RESEARCH

Authors	Application	OS
Anglano 2014 [6]	WhatsApp	Android
Wouter 2007 [7]	Windows Live Messenger 8.0	Windows XP
Yang et al. 2016 [8]	Facebook IM, Skype	Windows 8.1 onwards
Awadhi et al., 2011 [9]	Facebook IM	Web Browser based
Yusoff et al. 2016 [10]	Facebook, Twitter, Google+, Telegram, OpenWapp, Line	Mobile based Firefox OS
Ming and Chih 2018 [11]	Line	Android
Matthew et al. 2008 [13]	AIM Express, Google Talk, Meebo, E-Buddy	Web Browser based
Christos et al. 2015 [12]	WhatsApp, Viber, Skype, Tango Chat	iOS and Android

B. Findings and Discussion

The framework allowed the retrieval of a great deal of forensic evidence from the social media and instant messaging applications it was tested with. The table III below shows a summary of the artefacts retrieved from both Windows and Mac based workstations. The artefacts and retrieval pattern would differ slightly from version to version in OS, application and platform, but the proposed framework is common which will allow it to retrieve evidence by identifying the accurate pattern for all major OS and application platforms.

TABLE III. ARTEFACTS RETRIEVED USING PROPOSED FRAMEWORK

Application	Evidence Retrieved
Facebook	Status Update text and ID Comment text and ID Post URL Privacy Status Timestamps FB Name
Twitter	Tweet text Hashtags User details Screen name Location
FB Messenger	Conversation text Timestamp Sender user ID Receiver user ID
Viber	Message text Sequence ID
WhatsApp	Message text Phone number
Gmail	Email body Session ID Language Subject Sender name Receiver email Timestamp Country
Hotmail	Email body Routing type Mailbox type Subject Sender email and name Receiver email and name
Yahoo Mail	Email body Subject Sender email Receiver name and email Timestamp
Skype	Message text Message ID Username Timestamp
Google Search	Search criteria Results URL
Yahoo Search	Search criteria Results URL
Google Maps	Search location name Location coordinates
iMessage	Sender account email Receiver account email Conversation text
User Credentials	Usernames Passwords

The tables II and III conclude the fact that most research prior was based on forensic retrieval of artefacts focused on few application(s) or operating system(s). A common framework, which can be used to aid in volatile forensics examinations was not present. The framework can therefore be used by digital forensic investigators in order to examine and extract a maximum number of artefacts relating to social media and instant messaging.

The remaining of the section will show evidence retrieved by the usage of regex created using the proposed framework.

Figure 3 shows the Skype username and password retrieved from the dump file using regex "*&login=(.*?)&(?:.*?)Login Options=\d&passwd=(.*?)&ps*". The evidence is grouped; group 1-username, group 2-password.

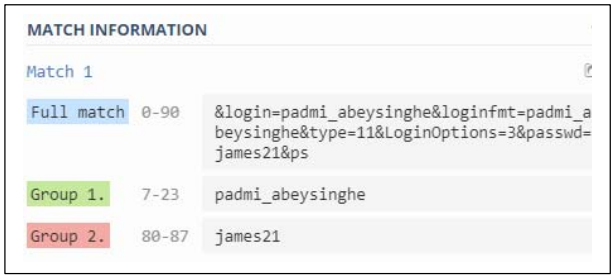

Fig. 3. Sample Evidence – Skype User Credentials.

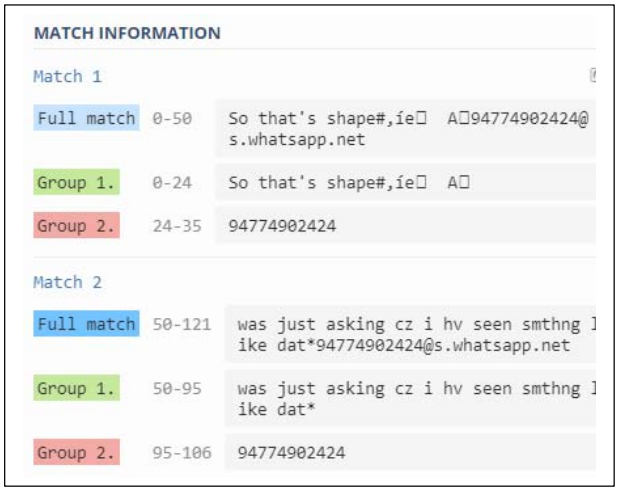

Fig. 4. Sample Evidence – WhatsApp.

Fig. 5. Sample Evidence – Facebook Chat.

Figure 4 shows WhatsApp evidence retrieved from a dump file using regex *"(.*?)(\d{11})@s.whatsapp.net"*. The evidence is grouped; group 1-message and group 2-phone number with country code.

Figure 5 shows a Facebook chat conversation retrieved. The grouped evidence are as follows; group 1- Target FB user ID, group 2-timestamp in Unix format, group 3-user account name, group 4-message. Regex used *"target_uid\":(\d*),\"time\":(\d*),\"type\":\"msg\",\"message\":\"(.*?):\s(.*?)\"|E(.*?),\s(.*?),\sYou:\s(.*?)"*.

Fig. 6. Sample Evidence – Skype Conversation.

Figure 6 shows a Skype conversation retrieved. Regex used *"<message(?:.*)<messageid>(.*)<\/messageid><originalarrivaltime>(.*).\d\d\dZ<\/originalarrivaltime><from>\d:(.*);epid=\{(?:.*)\}<\/from><messagetype>(.*)<\/messagetype><content>(.*)<\/content>"*. The grouped evidence are as follows; group 1-messaged id, group 2-timestamp, group 3-username, group 4-message type, group 5-message text.

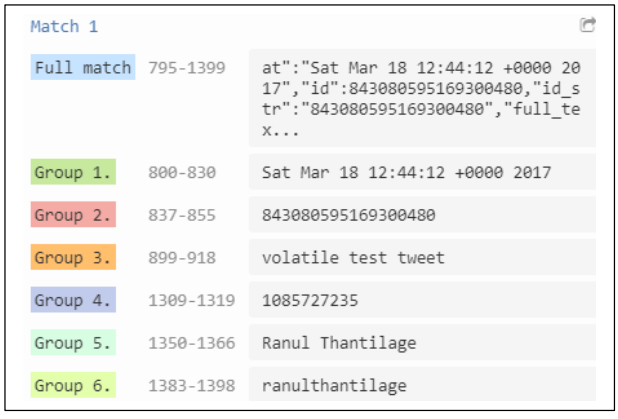

Fig. 7. Sample Evidence – Twitter tweet.

Figure 7 shows evidence retrieved related to Twitter by using regex: *"at\":\"(.*?)\",\"id\":(\d*),\"id_str\":(?:.*?)\"full_text\": \"(.*?)\",\"(?:.*?)\"entities\":\{\"hashtags\":\[(?:.*?)\](?:.*?)\"user\":\{\"id\":(\d*),(?:.*?)\"name\":\"(.*?)\",\"screen_name\":\"(.*?)\",\"location\":\"(.*?)\"\,"*.

The timestamp, tweet id, tweet text, user id, user display name, username and location of account are found.

Figure 8 shows the Gmail Account password retrieved using regex *"&Email=(.*?)&Passwd=(.*?)&rmShown=1"*. Group 1- username (email address) and group-2 password. Note that the '@' symbol is replaced by '%40' (also known as escape character) in the dump file.

Fig. 8. Sample Evidence – Gmail user credentials.

VI. Conclusion

In this paper, we present a framework for volatile memory forensics to acquire and analyze artefacts from social media and IM platforms. In order to demonstrate its feasibility and efficiency, we also evaluate our proposed framework with all popular social media and instant messaging platforms to retrieve relevant evidences. In addition, Google Search engine and Google Maps platforms were also tested. Most applications were tested for both Windows based and Mac based workstations. This shows that the concept brought forward with this framework can go beyond borders as its approach is common to other operating systems as well. Another advantage of our approach is to overcome the challenge of encryption. With a volatile approach we can retrieve the artefacts while the machine is running, in other words extracting and analyzing decrypted evidence.

Today, social media and IM apps are not limited on PCs but also widely used on mobile devices and becoming more popular in smart devices in a smart house and moreover integrated in smart vehicles. Hence, we are also looking at extending our approach to assist the current challenges in retrieving and analyzing artefacts for mobile forensics [27], smart home [28] and smart vehicle forensics [29][30].

References

[1] S. Kemp, "Digital 2019: Global Digital Overview — DataReportal – Global Digital Insights," 2019.

[2] RV. Voorst, T. Kechadi and N-A. Le-Khac, (2015) "Forensics Acquisition Of Imvu: A Case Study". Journal of Association of Digital Forensics, Security and Law, 10(4):69-78 DOI: https://doi.org/10.15394/jdfsl.2015.1212.

[3] A. Tillekens, N-A. Le-Khac and T-T. Pham-Thi, (2016) "A Bespoke Forensics GIS Tool", IEEE International Symposium On Mobile Computing, Wireless Networks, and Security, Nevada, USA, Dec 2016. DOI: https://doi.org/10.1109/CSCI.2016.0189.

[4] K. Hausknecht, D. Foit, and J. Burić, "RAM data significance in digital forensics," 2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc., no. May, pp. 1372–1375, 2015.

[5] S. Vömel and F. C. Freiling, "A survey of main memory acquisition and analysis techniques for the windows operating system," Digit. Investiation, vol. 8, no. 1, pp. 3–22, 2011.

[6] C. Anglano, "Forensic analysis of whats app messenger on Android smartphones," Digit. Investigation, vol. 11, no. 3, pp. 201–213, 2014.

[7] W. S. van Dongen, "Forensic artefacts left by Windows Live Messenger 8.0," Digit. Investig., vol. 4, no. 2, pp. 73–87, 2007.

[8] T. Y. Yang, A. Dehghantanha, K.-K. R. Choo, and Z. Muda, "Windows Instant Messaging App Forensics: Facebook and Skype as Case Studies," PLoS One, vol. 11, no. 3, p. e0150300, Mar. 2016.

[9] N. Al Mutawa, I. Al Awadhi, I. Baggili, and A. Marrington, "Forensic artifacts of Facebook's instant messaging service," 6th Int. Conf. Internet Technol. Secur. Trans., no. December, pp. 771–776, 2011.

[10] M. N. Yusoff, A. Dehghantanha, and R. Mahmod, "Forensic Investigation of Social Media and Instant Messaging Services in Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp, and Line as Case Studies," Contemp. Digit. Forensic Investig. Cloud Mob. Appl., pp. 41–62, 2016.

[11] M. S. Chang and C. Y. Chang, "Forensic Analysis of LINE Messenger on Android," J. Comput., vol. 29, no. 1, pp. 11–20, 2018.

[12] C. Sgaras, M.-T. Kechadi and N.-A. Le-Khac (2015) "Forensics Acquisition and Analysis of Instant Messaging and VoIP Applications" Lecture Notes in Computer Science, Vol.8915: 188-199 June 2015. DOI: https://doi.org/10.1007/978-3-319-20125-2_16.

[13] M. Kiley, S. Dankner, and M. Rogers, "Forensic analysis of volatile instant messaging," IFIP Fed. Inf. Process., vol.285, p. 129–138, 2008.

[14] N. L. Beebe and J. Clark, "Digital Forensic Text String Searching: Improving Information Retrieval Effectiveness by Thematically Clustering Search Results," in The Digital Forensic Research Conference, 2007.

[15] K. Amari, "Techniques and Tools for Recovering and Analyzing Data from Volatile Memory," 2009.

[16] W. Ahmed and B. Aslam, "A comparison of windows physical memory acquisition tools," in MILCOM 2015 - 2015 IEEE Military Communications Conference, 2015, pp. 1292–1297.

[17] G. Osborne, "Memory Forensics : Review of Acquisition and Analysis Techniques," Edinburgh, 2013.

[18] M. H. Ligh, A. Case, J. Levy, and Aa. Walters, The Art of Memory Forensics, vol. 53, no. 9. Indianapolis: John Wiley & Sons, Inc., 2014.

[19] M. Faheem, M-T. Kechadi, N-A. Le-Khac, (2015) "The State of the Art Forensic Techniques in Mobile Cloud Environment: A Survey, Challenges and Current Trend". International Journal of Digital Crime and Forensics, Vol.7 (2):1-19 DOI: http://dx.doi.org/10.4018/ijdcf.2015040101.

[20] M. N. Faiz and W. A. Prabowo, "Comparison of Acquisition Software for Digital Forensics Purposes," Inf. Syst. Comput. Network, Comput. Electron. Control, vol. 4, no. 1, p. 37, 2018.

[21] G. L. Garcia, "Forensic physical memory analysis: an overview of tools and techniques," Helsinki, 2007.

[22] L. Sharmila, U. Sakthi, A. Geethanjali, and S. Sagadevan, "Regular Expression Based Pattern Matching for Gene Expression Data to Identify the Abnormality Gnome," in 2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM), 2017, pp. 301–305.

[23] J. Goyvaerts, Regular Expressions: The Complete Tutorial. 2007.

[24] J. Goyvaerts, "Regular Expression Reference: Capturing Groups and Backreferences," 2017. [Online]. Available: https://www.regular-expressions.info/refcapture.html. [Accessed: 08-Aug-2018].

[25] M. S. Zareen, A. Waqar, and B. Aslam, "Digital forensics: Latest challenges and response," in 2013 2nd National Conference on Information Assurance (NCIA), 2013, pp. 21–29.

[26] V. L. L. Yinghua Guo, J. Slay, and J. Beckett, "Validation And Verification Of Computer Forensic Software Tools-Searching Function," in Digital Investigation Vol.6 , 2009, p. S12-S22

[27] M. Faheem, M-T. Kechadi, N-A. Le-Khac (2014) "Smartphone Forensics Analysis: A Case Study for Obtaining Root Access of an Android SamSung S3 Device and Analyse the Image without an Expensive Commercial Tool", Journal of Information Security, 5(3):83-90 (8 pages) DOI: http://dx.doi.org/10.4236/jis.2014.53009

[28] A. Goudbeek, K-K. R. Choo, N-A. Le-Khac (2018), "A Forensic Investigation Framework for Smart Home Environment", 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications, New York, USA, August, 2018, DOI: https://doi.org/10.1109/TrustCom/BigDataSE.2018.00201

[29] N-A. Le-Khac, D. Jacobs, J. Nijhoff, K. Bertens, K-K. R. Choo (2018), "Smart Vehicle Forensics: Challenges and Case Study", Future Generation of Computer Systems, Elsevier, July 2018, DOI: https://doi.org/10.1016/j.future.2018.05.081

[30] D. Steiner, L. Chen, D. Hayes, N-A. Le-Khac (2019), "Vehicle Communication within Networks – Investigation and Analysis approach: A Case Study", The 2019 ADFSL Conference on Digital Forensics, Security and Law, FL. USA, May 2019