Journal of Computers Vol. 30 No. 1, 2019, pp. 114-125
doi:10.3966/199115992019023001012

LINE Messenger Forensics on Windows 10

Ming Sang Chang[1*], Chih Yen Chang[2]

[1] Department of Information Management, Central Police University, Taiwan, ROC
mschang@mail.cpu.edu.tw

[2] Graduate Institute of Communication Engineering, National Taiwan University, Taiwan, ROC
gsmmcc@gmail.com

Received 19 February 2017; Revised 30 January 2018; Accepted 1 March 2018

Abstract. There are many cybercrime and malicious activities caused by taking advantage of the rapid growth of social media and instant messaging applications. The misuse of social media and instant messaging in user end devices may lead to cybercriminals with malicious purposes. In order to identify crimes, it is essentially required to retrieve these traces and evidences by using appropriate forensic technique. This paper studies the artifacts left by LINE application on Windows 10 and presents evidence gathering of LINE messenger application. It proves beneficial for forensic analysts and practitioners as it assists them in course of mapping and locating digital evidences.

Keywords: digital forensics, instant messaging, investigation, LINE

1 Introduction

There are many cybercrime and malicious activities caused by taking advantage of the rapid growth of social media and instant messaging applications. Many social media and instant messaging providers have extended their services to smart phones, tablet computers, and personal computers. The misuse of social media and instant messaging in user end devices may lead to cybercriminals with malicious purposes.

LINE is a proprietary instant messenger (IM) application on smartphones, tablet computers and personal computers. LINE users can exchange texts, images, video and audio. They also can conduct free VoIP conversations and video conferences. LINE first launched in Japan in 2011 [1]. It reaches 100 million users within eighteen months and 200 million users only six months later [2]. LINE became Japan's largest social network in 2013. In October 2014 LINE announced that it had attracted 560 million users worldwide with 170 million active user accounts [3-4]. In the fourth quarter of 2016 LINE announced they have more than 217 million monthly active users [5].

LINE is a cross platform application available for Windows, MAC, iOS, Android, etc. As the use of LINE increasing rapidly, cybercrimes, such as slander spreading, copyright infringement, cyber stalking and cyber bullying, becomes more and more severe. To solve IM based cybercrimes, investigators need to perform forensic analysis of suspicious devices to find digital evidences.

Depending on the IM application in use, there are several methods that can be performed to recover IM artifacts from client devices. These evidences can be used to profile the behavior of its user and may even allow the investigator to predict the users' actions [6-8]. Each device and application has its own acquisition requirements and potential sets of evidence.

It has a few paper about LINE forensics. The reference paper about LINE forensics are on [14-15]. [14] provided a forensic analysis of the artifacts left by the LINE instant messaging application on an Android device. [15] provided the forensic analysis of LINE services on Firefox OS. To our knowledge, no detailed analysis of LINE artifacts on Windows 10 has been undertaken, hence this research aims to fill

* Corresponding Author

the gap and provides a road map of LINE forensic artifacts. In this paper, we seek to identify potential terrestrial artifacts that may remain after the use of the LINE application on a Windows 10 client device. We attempt to answer the following questions in this research:

(1) What data does it remain on a Windows 10 device and the locations on a hard drive after a user has used LINE?

(2) What data does it remain in Random Access Memory after a user has used the LINE services on a Windows 10 device?

Findings from this research will contribute to the forensic community's understanding of the types of terrestrial artifacts that are likely to remain after the use of LINE services on devices running Windows 10.

This paper has organized as follows. In section 2 introduces the related works. In section 3, we outline the research methodology. In section 4, results and analysis are discussed. In section 5, we discuss our research findings. Finally, section 6 is conclusions.

2 Related Works

The evidences were stored on three principle areas by using IM. They are hard drive, memory, and network. Some IM services have the ability to log information on the user's hard drive [9]. To use IM service, an account must be established to create a screen name providing with user information. Some instant messenger providers might assist the investigation with information of the account owner.

Evidence can be found in various internet file caches used by Internet Explorer for volatile IM and each cache holds different pieces of data. Apart from the normal files, files left by instant messenger on a hard drive can be in temp file format and will generally be deleted and could be very difficult to retrieve once the machine is power down. An operating system generally stores information of all the installed and uninstalled applications in the system. The uninstalled application also leaves evidence. If a user has deleted an instant messenger application, there is a chance that a record can be found in the registry to prove that the instant messenger has once installed. Information is also stored within the memory. Since every application requires memory to execute, evidence could be left behind in the system's memory. The analysis on live memory has allows us to extend the possibility in providing additional contextual information for any cases. For any Windows based operating system, important evidences can usually be found beneath the physical memory, hibernation file and pagefile [10].

Artifacts of instant messaging have been of interest in many different digital forensic studies. Cosimo Anglano et al. [11] presented the forensic analysis of the artifacts generated on Android smartphones by ChatSecure. Songyang Wu et al. [12] described how to acquire the data of WeChat and how to decode the encrypted database. Ovens et al. [13] located and documented artifacts by Kik messenger on iOS. Asif et al. [17] provided a forensic analysis of the artifacts left by the LINE instant messaging application on an Android device. Yusoff et al. [15] provided the forensic analysis of instant messaging services in Firefox OS. Chu et al. [16] focused on live data acquisition from personal computer and was able to identify distinct strings that will assist forensic practitioners with reconstruction of the previous Facebook sessions. Iqbal et al. [14] studied the artifacts left by the ChatON instant messaging application. The analysis was conducted on an iPhone running iOS6 and a Samsung Galaxy Note running Android 4.1. Walnycky et al. [18] added that artifacts of the Facebook Messenger could vary depending on user settings, OS version, and manufacturer. Azfar et al. [19] adapt a widely used adversary model from the cryptographic literature to formally capture a forensic investigator's capabilities during the collection and analysis of evidentiary materials from mobile devices. In 2013 Mahajan et al. [20] performed forensic analysis of Whatsapp and Viber on five android phones using UFED and manual analysis. Anglano [21] carried out Whatsapp forensics on Android in 2014 using YouWave virtualization platform. Levendoski et al. [22] concluded that artifacts of the Yahoo Messenger client produced a different directory structure on Windows Vista and 7. Wong et al. [23] and Mutawa et al. [24] demonstrated that artifacts of the Facebook web-application could be recovered from memory dumps and web browsing cache.

To our knowledge, no detailed analysis of LINE artifacts on Windows 10 has been undertaken, hence this research aims to fill the gap and provides a road map of LINE forensic artifacts.

3 Methodology

In our research, we use virtual machines (VMs) with a standard installation of Windows 10. The LINE application was installed on Windows 10. We set up 18 different configurations. This allowed us to examine a variety of test in several configurations and to facilitate forensic analysis of LINE Messenger. The study was focused on identifying data remnants of the activities of LINE. This is undertaken to determine the remnants an examiner should search for when Instant Messenger is suspected. Our research also includes the circumstances of using anti-forensic methodology to hide evidence, and whether remnants remain to identify the use of LINE Messenger.

There are 18 virtual machines which replicate different circumstance of usage to gather remnants in relation to the use of LINE on Windows 10. The virtual machines were created for each different circumstance of LINE activities. This represents different physical computer systems available for analysis, with different circumstances and data remnants available for analysis on each VM. According to the activities of LINE, we create a base VM and 17 different VMs. The virtual machines reduce the costs of the study, since neither many real personal computers are necessary to carry out the experiments.

The base VM is to compare the subsequent VMs to determine the changes made. It is possible to observe the changes of file systems. Our experimental test-bed consists of a set of virtual machines. That is VMware Workstation V12.0.0. For each experiment, Windows 10 Enterprise was installed on every virtual machine. The LINE Messenger V4.2.0.654 for windows was installed on all virtual machines. In each experiment, we assign only a role to each virtual device. We use it to carry out the corresponding activities. At the end of the experiment, we suspend the virtual device. We parse the file implementing the corresponding internal memory and hard drive by means of WinHex 17.9.0.0 and EnCase V7.04. Then we extract the files where LINE Messenger stores the data it generates.

According to the activities of LINE, we create seven sub-experiment systems. They are Base-VM, Login-VM, Snd-VM, Rcv-VM, Keep-VM, Delete-VM, and Delete_Keep-VM. In all experiments, there are 18 virtual machines to gather the data in relation to the activities of LINE as shown in Fig. 1.

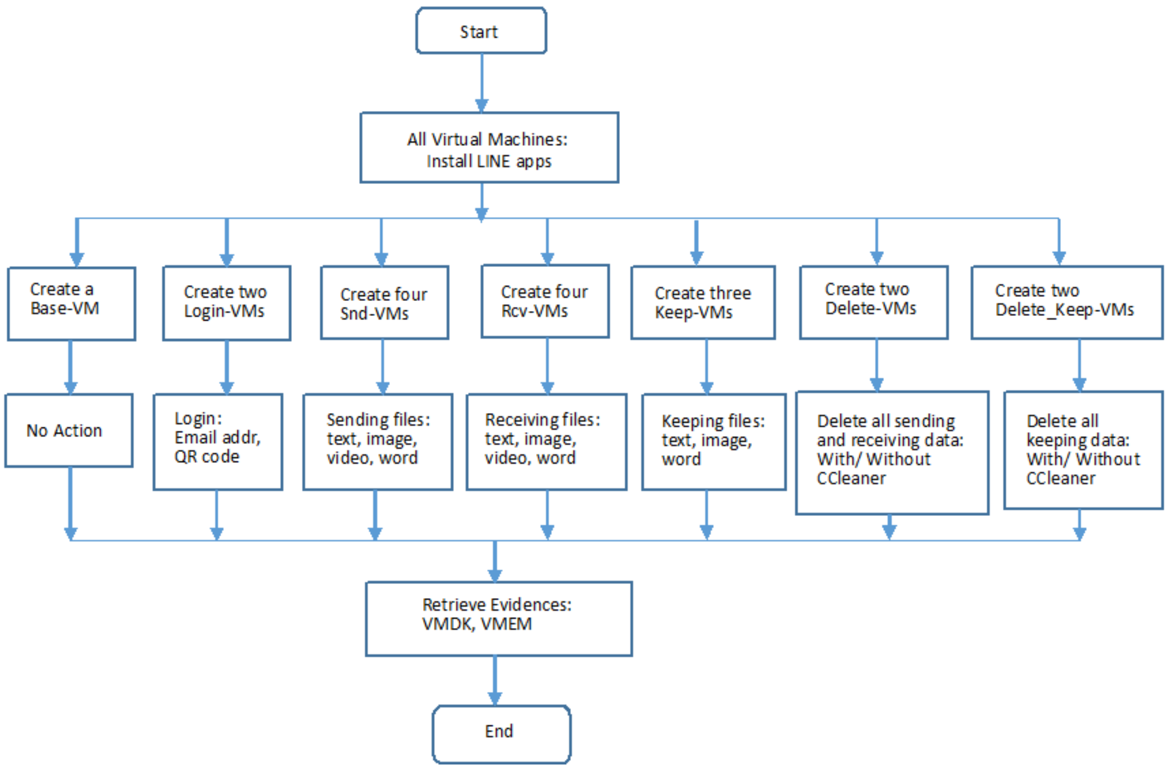

Fig. 1. All virtual machines and the examination procedure for LINE forensics

The different actions undertaken are as follows. They will be divided in seven cases.

(1) The first case was to install the LINE messenger into base virtual machine.

(2) The second case was to make two copies of the base machine for each scenario. An account of

LINE was created for these experiments. We use email address and QR code to sign in LINE on two different virtual machines. We do nothing and sign out. Then we use SQLite Database Browser V2.0 to analyze LINE database files and use WinHex and EnCase to analyze memory and hard drive to find the remnants of account and password.

(3) The third case was to make four copies of the base virtual machine for each scenario. There are four scenarios as the activities of Snd-VM on Fig. 1. After sending action, sender signs out and finds the data remnants.

(4) The forth case was to make four copies of the base virtual machine for each scenario. There are four scenarios as the activities of Rcv-VM on Fig. 1. After receiving action, receiver signs out and finds the data remnants.

(5) The fifth case was to make three copies of the base virtual machine for each scenario. There are three scenarios as the activities of Keep-VM on Fig. 1. After keeping action, we sign out and finds the data remnants.

(6) The sixth case was to make two copies of the base virtual machine with LINE for each scenario. We do the same actions as case 3 and 4. Then we delete all the sending and receiving data. We log out and find the data remnants. In the other scenario, after we delete all the sending and receiving data, we use CCleaner to remove LINE application and delete temporary, history, cookies, recycle bin, memory dumps, log files, etc. Then we retrieve the data remnants.

(7) The seventh case was to make two copies of the base virtual machine for each scenario. We do the same actions as case 5. Then we delete all the keeping messages. We log out and find the data remnants. In the other scenario, after we delete all the keeping messages, we use CCleaner to remove LINE and delete temporary, history, cookies, recycle bin, memory dumps, log files, etc. Then we retrieve the data remnants.

4 Result and Analysis

4.1 LINE Database Files

There is a main location for the recovered LINE artifacts. All chat database and text data are on C\Users\[UserName]\AppData\Local\LINE. There are two subdirectories name *Cache* and *Data* in the LINE directory. All of image files are in the Cache directory. Many database files are in the Data directory. The contents of database files are encrypted. All contents can't be read except timestamp. The chat table is shown as Fig. 2.

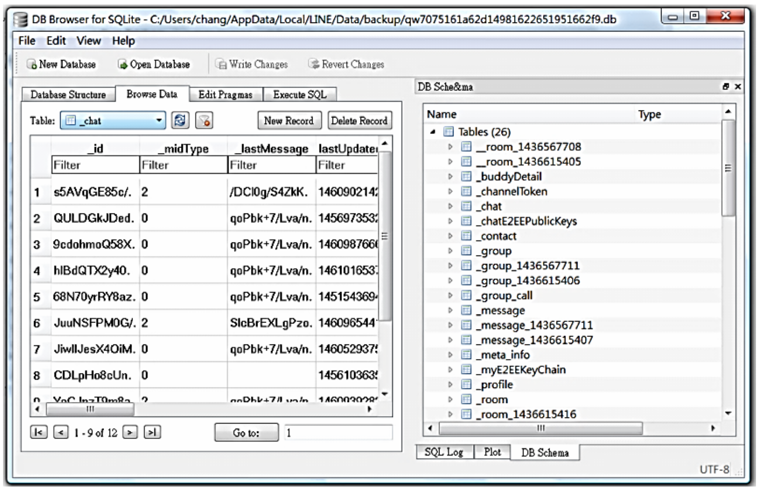

Fig. 2. The content of the chat table

The database file contains many tables relate to regular chat activities such as contact information and chat messages as shown in Table 1. While opening the database files we find the schema but can't read the contents except we can decrypt it.

Table 1. The information of chat activities

Table	Column	Information
chat	id	Each chat session is stored with a unique identifier.
	midType	Media type
	lastMessage	The last message
	lastaupdateTime	Timestamps of last update
contact	mid	Unique identifier of each contact
	createdTime	Timestamps of creation
	displayName	the name that appears in chats
message	from	sender
	to	receiver
	createdTime	Timestamps of creation
	text	a text message is stored in this field
group	id	Unique identifier of chat group
	createdTime	Timestamps of creation
	name	Name of the group as specified during creation of the group.

4.2 Login-VM

Login with email address. We can't find any remnants about email address and password in Virtual Machine Disk (VMDK) file. In the Virtual Machine Memory (VMEM) file, the email address (testabc2016@gmail.com) can be found as Fig. 3. We can't find the password. We believe the password with a secure encryption so that we can't find it.


```
2D307680  oMyhcme":true,"privacyReceiveMessagesFrcmNotFriend":true,"privac
2D3076C0  yAgreeUseLineCoinToPaidCall":false,"privacyAgreeUsePaidCall":fal
2D307700  se,"privacyAllowFriendRequest":true,"contactMyTicket":"41AExhb81
2D307740  F","identityProvider":1,"identityIdentifier":"testabc2016@gmail.
2D307780  ccm","snsAccounts":{},"phoneRegistration":true,"emailConfirmatio
2D3077C0  nStatus":3,"accountMigrationPincodeType":2,"enforcedInputAccount
2D307800  MigrationPincode":false,"securityCenterSettingsType":0,"preferen
2D307840  ceLocale":"zh_TW","custcmModes":{},"e2eeEnable":false},"e2eePubK
```


Fig. 3. The remnants of login with email address

Login with QR code. We use QuickMark application to decode QR code and get a character string (http://line.me/R/au/q/Jlebl0zwDGWSZEMBlqeyRl4AR1PR2lb7). This character string is a key to find login information and password. We can't find any remnants about QR code and password. But we find our LINE account identification (ID, u67f209fef81e3693683aa1df98432e33) is shown as Fig. 4. A search for the QR code and password produced no matches in the forensic image and memory dump in the experiment.


```
009EC400  ":"INTERNAL"}}]},"postInfo":{"allowCcmment":true,"allowEdit":fal
009EC440  se,"allowFriendRequest":true,"allowLike":true,"allowLikeShare":f
009EC480  alse,"allowPhotoCcmment":true,"allowRecall":true,"allowShare":fa
009EC4C0  lse,"appSn":1341209850,"ccmmentCount":0,"ccmmentLinkPermission":
009EC500  "ALL","createdTime":1447124569000,"enableCcmmentApproval":false,
009EC540  "hasSharedToPost":false,"hcmeId":"u67f209fef81e3693683aa1df98432
009EC580  e33","likeCount":0,"likeLinkPermission":"ALL","liked":false,"pos
009EC5C0  tId":"1144712456906014091","readPermission":{"type":"FRIEND"},"s
009EC600  tatus":"NORMAL","updatedTime":1447124569000,"url":{"targetUrl":"
009EC640  #HOME_END","type":"INTERNAL"}},"updatedPost":false,"userInfo":{"
```


Fig. 4. The remnants of login with QR code

In these two experiments the data remnants can be found in volatile memory. There are no remnants on VMDK file.

4.3 Snd-VM

Sending text message. The remnants can't be found in VMDK files. In volatile memory the sending user name (Yaaichu), chat message (Send you a message.), time stamp (1446205306500), and the receiver ID (u300bc27d6c3e5fe3390861c4ae9f240d) are shown as Fig. 5.

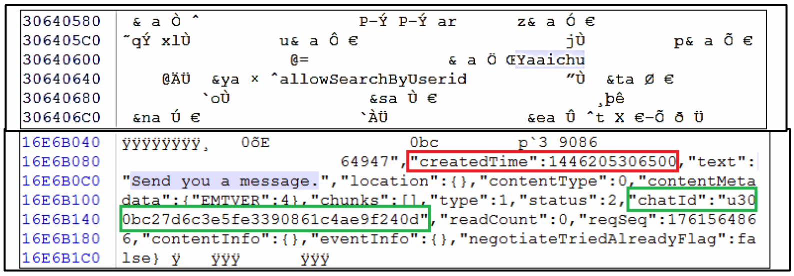

Fig. 5. The remnants of sending text message

Sending image. The locations of remnants of image file are shown in Table 2. We also find file name, file size (208916), time stamp (1446204865123), sending user ID (u67f209fef81e3693683aa1df9 8432e33), receiving user ID (u300bc27d6c3e5fe3390861c4ae9f240d), and the locations of sending file as shown in Fig. 6.

Table 2. The locations of remnants of sending image file

C\Users\[UserName]\ntuser.dat.LOG1
C\Users\[UserName]\NTUSER.DAT
C\$MFT
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\IMG_1028.lnk
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V01.log
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\5f7b5f1e01b83767.automaticDestinations-ms
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat

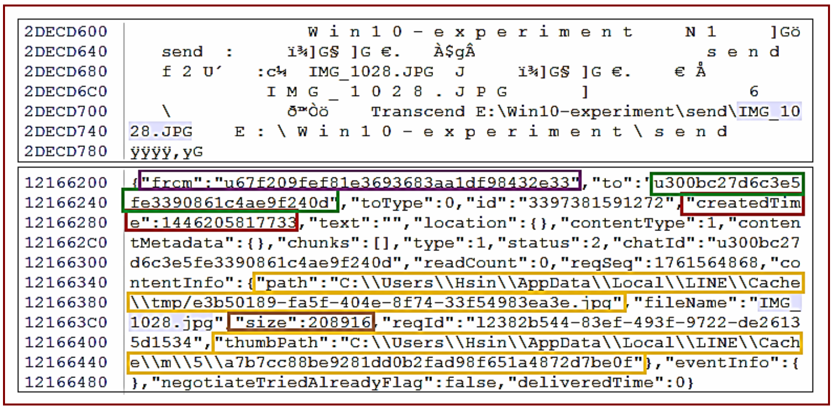

Fig. 6. The remnants of sending image file

Sending video & sending word files. We can find file name, file size, time stamp, sending user ID, receiving user ID, and the directory of sending file as the same as **Sending image.**

In these four experiments the remnants can be found in volatile memory and hard disk drive. When a user sends a file using the LINE app, there will be records remaining in Windows system files such as $MFT, ntuser.dat.LOG1, NTUSER.DAT, and WebCacheV01.dat to indicate the filenames, and directory paths for the sending files. The remnants are also recorded in memory as Fig. 5 and Fig. 6.

4.4 Rcv-VM

Receiving text message. The remnants can’t be found in VMDK files. In volatile memory the sending user’s name (Yaaichu), sender ID, receiver ID, text message (I know,) and time stamp are shown as Fig. 7.

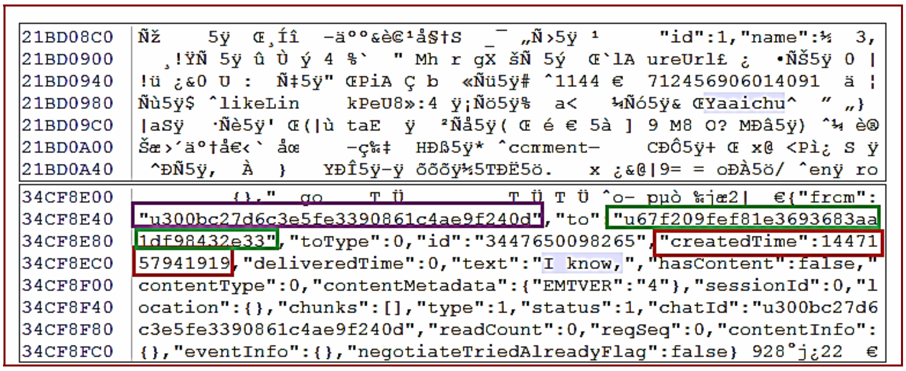

Fig. 7. The remnants of receiving text message

Receiving image. The locations of remnants of image file are shown in Table 3. We also find filename, file size, time stamp, sending user ID, receiving user ID, and the directory of receiving file in the volatile memory.

Table 3. The locations of remnants of receiving image file

C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V01.log
C\Windows\Prefetch\ReadyBoot\ReadyBoot.etl
C\Users\[UserName]\NTUSER.DAT
C\$Extend\$UsnJrnl·$J
C\ProgramData\Microsoft\Search\Data\Applications\Windows\Windows.edb
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\IMG_0101.lnk
C\Users\[UserName]\ntuser.dat.LOG2
C\Users\[UserName]\ntuser.dat.LOG1
C\$MFT
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat
C\$LogFile

Receiving video & receiving word file. The file name, file size, time stamp, sending user ID, receiving user ID, and the directory of sending file can be found as **Receiving image.**

In these four experiments the remnants can be found in volatile memory and hard disk drive. When a user receives a file using the LINE app, there will be records remaining in Windows system files such as $LogFile, $MFT, and $UsnJrnl to indicate the filenames, and directory paths for the downloaded files.

4.5 Keep-VM

Keep text message. The remnants can't be found in VMDK files. In volatile memory a keeping message (test_test_550) is shown as Fig. 8.

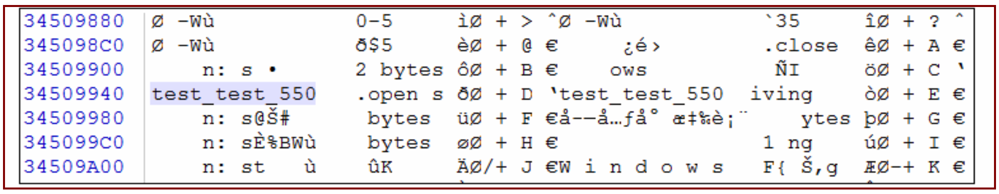

Fig. 8. The remnants of keeping text message

Keep image. The locations of remnants of image file are shown in Table 4. In volatile memory the remnants of image file are shown as Fig. 9.

Table 4. The locations of remnants of keeping image file

C\Lost Files\API-MS-Win-EventLog-Legacy-L1-1-0.dll
C\Users\[UserName]\ntuser.dat.LOG2
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\5f7b5f1e01b83767.automaticDestinations-ms
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V01.log
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\IMG_1225.lnk
C\Lost Files\api-ms-win-security-sddl-l1-1-0.dll
C\$MFT

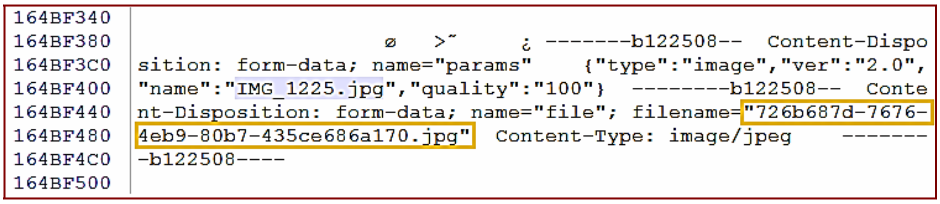

Fig. 9. The remnants of keeping image

Keep word file. The remnants are the same as Keeping image.

4.6 Delete-VM

Delete all data. The locations of remnants are shown in Table 5.

Table 5. The locations of remnants of deleting all data

C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V0100001.log
C\Users\[UserName]\NTUSER.DAT
C\Windows\Prefetch\ReadyBoot\ReadyBoot.etl
C\Users\[UserName]\ntuser.dat.LOG2
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat
C\$MFT
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\ball23.lnk
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\5f7b5f1e01b83767.automaticDestinations-ms
C\Users\[UserName]\ntuser.dat.LOG1

Delete all data with Ccleaner. We do the same actions as **Delete all data** and use CCleaner to remove LINE apps and delete temporary, history, cookies, recycle bin, memory dumps, log files, etc. The locations of remnants are as Table 5 except for C\$MFT and C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V0100001.log.

4.7 Delete_Keep-VM

Delete all keeping data. We do all the keep actions as **Keep-VM**. Then we delete all of the keeping data. The locations of remnants are shown in Table 6.

Delete all keeping data with CCleaner. We do the same actions as **Delete all keeping data** and use CCleaner to remove LINE apps and delete temporary, history, cookies, recycle bin, memory dumps, log files, etc. The locations of remnants are as Table 6 except for C\$MFT, C\$LogFile, and C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V01.log.

Table 6. The locations of remnants of deleting all keeping data

C\Users\[UserName]\NTUSER.DAT
C\ProgramData\Microsoft\Search\Data\Applications\Windows\edb.log
C\Users\[UserName]\ntuser.dat.LOG2
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\WebCacheV01.dat
C\$MFT
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\AutomaticDestinations\5f7b5f1e01b83767.automaticDestinations-ms
C\Users\[UserName]\AppData\Local\Microsoft\Windows\WebCache\V01.log
C\$Extend\$UsnJrnl·$J
C\Windows\Prefetch\ReadyBoot\ReadyBoot.etl
C\Users\[UserName]\AppData\Roaming\Microsoft\Windows\Recent\flower3ac.lnk
C\$LogFile

Table 7 shows the meaning of terminology about Section 4.

Table 7. The meaning of terminology

Term	Description
$LogFile	Contains transaction log of file system metadata changes.
$MFT	Master File Table. Describes all files on the volume, including file names, timestamps, stream names, and lists of cluster numbers where data streams reside, indexes, security identifiers, and file attributes like “read only”, “compressed”, “encrypted”, etc.
ntuser.dat.LOG1	The ntuser.dat.log1 file contains a log of changes made to the registry files for your user account.
NTUSER.DAT	A user profile contains personal files and preference settings
automaticDestinations-ms	The automaticdestinations-ms file extension is associated with Microsoft Windows and used for Jump List files, that contain jump data about certain application.
ReadyBoot.etl	the logging process for ready boost
Unallocated Clusters	free space on a hard drive
$UsnJrnl:$J	actual journal entries are stored in the $UsnJrnl:$J
WebCacheV01.dat	users web history: the database structure for IE 11
edbtmp.log	All current transaction logs are stored in the edb.log file. Once this log file has reached the 5 MB limit, a new file is created and named edbttmp.log.
Lnk	a file extension for a shortcut file used by Microsoft Windows to point to an executable file

5 Discussions

In this research, we identified artifacts for LINE application. We focus on both the volatile memory and hard drive. Our experiments showed that critical application data is present in the RAM and it can be extracted for further analysis. Our hard drive analysis has shown that LINE application activities remain some artifacts in different locations. This indicated that once a user used LINE apps, records will remain in the application folder. We also find the contents of database files are encrypted.

The analysis was performed manually on each file to identify the file types, search for related LINE activities data, and determine the location of stored artifacts. There are several keywords used to detect the remnants, including email address, usernames, user ID, LINE, and etc. Further explanation of analysis and findings are provided as follows.

Login information. Logging in to the LINE would leave the account name in internal memory which can be detected by searching email address. Utilizing this account name to further search the internal memory can identify the assigned LINE user ID and the timestamp of the last login as shown in Fig. 3 and Fig. 4. However, there is no trace of the user password could be detected in the internal memory. We believe the password is protected with a secure encryption so that we could not find it. All of image files are found in the LINE *Cache* directory. There are many pictures can be found in the directory such as the user profile picture and cover picture.

Sending message. The examination of the LINE messages sent by the user are as Fig. 5 and Fig. 6.

These artifacts remained in the internal memory as plain text can be retrieved using user ID, or username as search keywords. The time stamp, chat ID, file name, file size, and file location also remain in the internal memory and can be retrieved using username as search keyword. From the experiments, text message sent can only be retrieved from the internal memory and the remnants of file is also in hard drive as Table 2.

Receiving message. The examination of the LINE received messages is as Fig. 7. These artifacts remain in the internal memory in plain text format, and can be retrieved by using user ID as search keyword. The time stamp, chat ID, filename, and file location also remain in the internal memory and can be retrieved by using filename extension as search keyword. From the experiments, the received text message can only be retrieved from the internal memory and the remnants of file are also in hard drive as Table 3.

Keeping message. The examination of the LINE kept messages are as Fig. 8 and Fig. 9. These artifacts remain in the internal memory in plain text format. The filename, time stamp, and user ID also remain in the internal memory and can be retrieved using filename extension or *Keep* as search keyword. In the memory the original filename is converted to a unique name in LINE, and both of them can be retrieved using filename extension or *Keep* as search keyword. From the experiments, the kept text message can only be retrieved from the internal memory and the remnants of file are also in hard drive as Table 4.

Deleting message. The examination of the LINE revealed deleting messages and files as Table 5 & Table 6. These artifacts of deleting message still remain in the internal memory in plain text but not all of the sending messages, receiving messages, and keeping messages can be found. The filename, file location, time stamp, and user name also can be found in the internal memory. After deleting action, the remnants in memory are less than before. The artifacts can still be retrieved using filename extension or *Keep* as search keyword.

The examination of the LINE deleted messages and files are as Table 5 & Table 6. These artifacts of deleted message still remain in the internal memory as plain text, but not all of the sent messages, received messages, and kept messages can be found. The filename, file location, time stamp, and user name also can be found in the internal memory. After deleting action, the remnants in memory may be less than before. However, the artifacts can still be retrieved by using filename extension or *Keep* as search keyword.

The significance and location of artifacts are worth to be noted. In our research, they were determined by: (1) Directories maintained by LINE app in the application folders. (2) Database schema held by LINE app in the application caches. (3) The cache copies of the transferred and downloaded files in the application folder.

Data stored in various forms and locations by the developer can become treasure for an investigator. The artifacts findings are summarized in Table 8.

Table 8. Summary of findings

Virtual Machine		Volatile Memory (RAM)	Hard Drive
Login-VM	Email addr	Account name found	Not found
	QR code	Not found	Not found
Snd-VM	Text	Content found	Not found
	Image file	Found	Found
	Video file	Found	Found
	Word file	Found	Found
Rcv-VM	Text	Content found	Not found
	Image file	Found	Found
	Video file	Found	Found
	Word file	Found	Found
Keep-VM	Text	Content found	Not found
	Image file	Found	Found
	Word file	Found	Found
Delete-VM	Delete	Found	Found
	Delete with CCleaner	Found	Found
Delete_Keep-VM	Delete	Found	Found
	Delete with CCleaner	Found	Found

6 Conclusions

Instant messaging becomes popular among individuals and business organizations. Applications such as LINE, WhatsApp, WeChat, Skype, and Facebook Messenger are some of the commonly used applications that may also be leveraged to commit crimes. It is important to identify the forensic artifacts left by these application to trace the criminal behaviors. In this paper we have presented the findings from our forensic examination of LINE application on Windows 10. The study consists of installation, uninstallation, logins, conversations, transferred files, and other activities in LINE. The results indicated that it leaves useful evidential material on the hard drive and memory dumps from use of LINE.

We would like to note that a limitation of our work is that it was tested on a Windows 10 device. We leave the analysis of LINE Messenger on different Operating System and to analyze the network traffic of LINE messaging applications as future work.

References

[1] LINE Corporation, About LINE Corporation. <http://linecorp.com/en/company/info>, 2017 (accessed 13.01.17).

[2] Enricko Lukman, LINE Hits 200 Million Users, Adding 100 Million in Just 6 Months. <https://www.techinasia.com/line-hits-200-million-users-adding-100-million-users-6-months>, 2017 (accessed 10.01.17).

[3] Horwitz Josh, Line finally reveals its monthly active user count. <https://www.techinasia.com/line-japanese-messaging-app-has-170-million-monthly-active-users>, 2018 (accessed 06.01.18).

[4] Akky Akimoto, Looking at 2013's Japanese social-media scene, The Japan Times 2013. <http://www.japantimes.co.jp/life/2013/12/17/digital/looking-at-2013s-japanese-social-media-scene-3/#.V6rgbtR97RY>, 2017 (accessed 15.01.17).

[5] LINE: number of monthly active users 2014-2016. <https://www.statista.com/statistics/327292/number-of-monthly-active-line-app-users/>, 2017 (accessed 03.01.17).

[6] A. Orebaugh, J. Allnutt, Data Mining Instant Messaging Communications to Perform Author Identification for Cybercrime Investigations, Digital Forensics and Cyber Crime, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, 2010.

[7] A. Iqbal, H. Obaidli, A. Marrington, A. Jones, Windows surface RT tablet forensics, Digital Investigation 11(S1)(2014) S87-S93.

[8] The United Nations Office on Drugs and Crime, Comprehensive study on Cybercrime Technical Report. <https://www.sbs.ox.ac.uk/cybersecurity-capacity/content/unodc-comprehensive-study-cybercrime>, 2018 (accessed 03.07.18).

[9] R. Gonzales, B. Schofield, W. Hagy, Investigations Involving the Internet and Computer Networks, National Institute of Justice, Washington, 2007.

[10] Y. Gao, T. Cao, Memory forensics for QQ from a live system, Journal of Computers 5(4)(2010) 541-548.

[11] C. Anglanoa, M. Canonicoa, M. Guazzonea, Forensic analysis of the ChatSecure instant messaging application on android smartphones, Digital Investigation 19(2016) 44-59.

[12] S. Wu, Y. Zhang, X. Wang, X. Xiong, L. Du, Forensic analysis of WeChat on Android smartphones, Digital Investigation 21(2017) 3-10.

[13] K.M. Ovens, G. Morison, Forensic analysis of Kik messenger on iOS devices, Digital Investigation 17(2106) 40-52.

[14] A. Iqbal, H. Alobaidli, A. Almarzooqi, A. Jones, LINE IM app Forensic Analysis, in: Proc. 2015 International Conference on High-capacity Optical Networks and Enabling/Emerging Technologies, 2015.

[15] M. Yusoff, A. Dehghantanha, R. Mahmod, Forensic Investigation of Social Media and Instant Messaging Services in

Firefox OS: Facebook, Twitter, Google+, Telegram, OpenWapp, and Line as Case Studies, Contemporary Digital Forensic Investigations of Cloud and Mobile Applications, 2017 (Chapter 4).

[16] H.-C. Chu, D.-J. Deng, J.-H. Park, Live Data Mining Concerning Social Networking Forensics Based on a Facebook Session Through Aggregation of Social Data, IEEE Journal on Selected Areas in Communications 29(7)(2011) 1368–1376.

[17] I. Asif, A. Marrington, I. Baggili, Forensic artifacts of the ChatON Instant Messaging application, in: Proc. 2013. International Workshop on Systematic Approaches to Digital Forensic Engineering (SADFE), 2013.

[18] D. Walnycky, I. Baggili, A. Marrington, J. Moore, F. Breitinger, Network and device forensic analysis of Android social-messaging applications, Digital Investigation 14(S1)(2015) S77–84.

[19] A. Azfar, K.-KR. Choo, L. Liu, An Android Social App Forensics Adversary Model, in: Proc. 2016. International Conference on System Sciences, 2016.

[20] A. Mahajan, M.-S. Dahiya, H.-P. Sanghvi, Forensic Analysis of Instant Messenger Applications on Android Devices, International Journal of Computer Applications 68(8)(2013) 38-44.

[21] C. Anglano, Forensic analysis of WhatsApp Messenger on Android smartphones, Digital Investigation 11(3)(2014) 201-213.

[22] M. Levendoski, T. Datar, M. Rogers, Yahoo! Messenger Forensics on Windows Vista and Windows 7, in: Proc. 2011. International Conference on Digital Forensics and Cyber Crime, 2011.

[23] K. Wong, C.-T. Lai, C.-K. Yeung, W.-L. Lee, P.-H. Chan, Facebook Forensics, Valkyrie-X Security Research Group. <https://www.fbiic.gov/public/2011/jul/facebook_forensics-finalized.pdf>, 2016 (accessed 10.09.16).

[24] N. Mutawa, I. Awadhi, I. Baggili, A. Marrington, Forensic artifacts of Facebook's instant messaging service, in: Proc. 2011. International Conference for Internet Technology and Secured Transactions, 2011.