International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

Live Acquisition & Analysis of Memory Dumps of WhatsApp Artifacts in Android Devices

Richu Ann Thomas

College Of Engineering Kallooppara, Cochin University of Science & Technology (CUSAT), Kerala, India

Abstract: ***An Instant messenger application can serve as a very useful yet very dangerous platform for the victim and the suspect to communicate WhatsApp is one of the world's most popular instant messengers. There by, the artifacts left by them have become very relevant evidences these days in crime investigation. In Android phones Information is stored in different formats at varied locations on the phone. In this paper we discuss about the Android live memory acquisition & an in-depth Android Memory Analysis to retrieve the chat logs, deleted and encrypted messages, VOIP & Cloud Backup features provided by WhatsApp as well as its SQLite databases & structure.***

Keywords: Android forensics, Instant Messenger Applications, WhatsApp, Memory Analysis, Live Memory Forensics, Cloud Backup

1. Introduction

The increased use of Instant messengers on Android phones has turned to be the goldmine for mobile and computer forensic experts. So the scope of Android forensics also increased. Traces and evidences left by messenger applications can be held on Android devices and right forensic technique is strongly required to retrieve those potential evidences.

By the advent of Smart phones, the cybercrimes have also increased rapidly. For investigators, data stored on Smart phones is likely to contain evidence crucial for resolving a criminal case. This evidence can either be stored in persistent memory or as live data in the system's main memory. The latter is typically lost when a device runs out of battery power or is shut down, making it harder to recover. Hence, a forensic investigator needs the abilities and proper means to recover such data from a mobile device. This field of forensic investigation is also called ***live memory forensics***.

1.1 WhatsApp

WhatsApp is one of the world's most popular instant messengers. With more than 1 billion active accounts, WhatsApp is the number one Instant messenger application used in around 180 countries of the world. Recognizing its popularity, Face book has paid $22 billion for the company. WhatsApp provides its users with various forms of communications, namely user-to-user communications, broadcast messages, and group chats. Like other messenger applications, WhatsApp artifacts can be valuable to examiners looking to recover evidence for a variety of investigation types.

WhatsApp communication history is not reflected in the mobile service bill, and WhatsApp messages are not stored on carrier's computers in case law enforcement officials need access to that information. WhatsApp messages communicate directly between end user devices, securely encrypted makes it impossible to intercept WhatsApp communications.
The most important things that has to be taken care in digital investigations involving WhatsApp applications are the users exchange files like plain text messages, multimedia files (containing images, audio, and video), contact cards, and geolocation information, as well as PDF documents etc. Another is the user profile information. Each user is associated with a profile, a set of information that includes his/her WhatsApp name, status line, and avatar (a graphic file, typically a picture). The profile of each user is stored on a central system, from which it is downloaded by other WhatsApp users that include that user in their contacts. The central systems also provide other services, like user registration, authentication, and message relay.

1.2 Why WhatsApp Artifacts??

Due to its enormous popularity, WhatsApp quickly became a target for all kind of Cyber criminals. The app has recently rolled out full end-to-end encryption feature is definitely a step forward in securing digital communication. WhatsApp partnered with or its uses apart of Security protocol of Open Whisper Systems for the cryptographic portions of messaging. The process involves a variation of Off the Record (OTR), Perfect Forward Secrecy (PFS), and the Double Ratchet Algorithm (DRA). The idea is simple. It has made the communication via WhatsApp private – sort of like a face-to-face conversation.ie. end-to-end encryption ensures that when you send a message, the only person who can read it is the person or group chat that you send that message to or the intended receiver. No one can see inside that message since it is locked. Not hackers. Not even WhatsApp since it implemented a secure lock mechanism recently where your messages are secured with a lock, and only the recipient and you have the special key needed to unlock and read the message. For this no need to turn on settings or set up special secret chats to secure your messages as it gets synched automatically with new version of WhatsApp.

The very fact makes it difficult for the police to investigate cases involving WhatsApp messaging, requesting WhatsApp history files from the mobile carrier or an Internet service provider is not possible since no logs are stored on carrier's side. The only way to acquire WhatsApp histories is imaging end-user devices or pulling data from local or cloud backups

2. Experimental Setup

To access this area, suitable commercial tools may be used (Cellebrite LTD., 2013; Micro Systemation, 2013; Oxygen Forensics, Inc., 2013a) but, unfortunately, I do not had access to them. In terms of hardware used for research I used SAMSUNG GT-S7582 mobile device ,a USB cable and a PC for analysis of data retrieved. In terms of software and versions used please refer Figure 1. Software Used for more details

OS & Softwares	Version
Android	4.2.2,Kernel version3.4.5-2788564
Kali Linux	Kali-Linux2.0.0-vm-i686
VMware Workstation	12.1.0 build-3272444
Autopsy	4.0.0
SQLite Browser	SQLite Version 3.9.2
Root Explorer	3.3.7
WhatsApp	2.16.57
Terminal Emulator	1.0.70
BusyBox	41
WhatsApp Viewer	v1.3
LiME	Version 2
Andriller	2.5.4.0

Figure 1: Experimental SetUp

3. Live Acquisition, Analysis Methodology & tools

In Android phones Information is stored in different formats at varied locations on the phone. As Android has a very stable security mechanism.It absolutely requires altering the device data so be careful as to avoid unnecessary changes to the device. Live imaging absolutely requires altering the device data. Imaging an Android device (whether dead or live) requires these things:

- A data connection between the device and the computer,using a standard USB cable.
- The environment should have basically installed with android-adb tools, adb fastboot ,java jdk,python
- An exploit ie. Rooting the device to obtain root permissions
- An imaging tool to acquire the live image of the device.

To do an in depth-analysis of Phone's memory to retrieve the WhatsApp Artifacts you are supposed to do the below mentioned processes

1) Data Acquisition & Analysis of Volatile RAM
2) Data Acquisition & Analysis of Internal Memory
3) Acquisition from Google backups

3.1 Data Acquisition & Analysis of Volatile RAM

Volatile RAM Acquisition is done with best available tool LiME developed by Sylve et-al which is the only tool that allows complete RAM dumping in Android phones. Further analysis was done using Volatility plugins.

RAM has to be considered as a critical piece of evidence. Because it was found that the synched account credentials are found simply plain text formats where as the same is found as encrypted another database called 'accounts.db'. Several URLs, process lists all we used recently can be found. I have done this with a tool named Bulk_Extractor

For Cross-Compiling LiME for Android the prerequisites are

- The mobile device should be rooted
- Mobile should be connected to the forensic workstation i.e. the system in USB debugging mode using a USB device. For that developer mode must be enabled
- Install the general android prerequisites
- Download and un(zip|tar) the android NDK
- Download and un(zip|tar) the android SDK
- Download and untar the kernel source for your device

Preparing The Module For Compilation

We need to create a Makefile to cross-compile our kernel module. A sample Makefile for cross-compiling is shipped with the LiME source. The contents of your Makefile should be similar to the following

```
obj-m := lime.o

lime-objs := main.o tcp.o disk.o

KDIR := /path/to/kernel-source
PWD := $(shell pwd)
CCPATH := /path/to/android-ndk/toolchains/arm-linux-androideabi-4.4.3/prebuilt/linux-x86/bin/
default: $(MAKE) ARCH=arm
CROSS_COMPILE=$(CCPATH)/arm-eabi- -C $(KDIR)
M=$(PWD) modules
```

Acquisition of Memory over TCP

```
$ adb push lime.ko /sdcard/lime.ko
$ adb forward tcp:4444 tcp:4444
$ adb shell
$ su
#
```

The following command loads the kernel module via *adb* on the target Android device:
insmod /sdcard/lime.ko "path=tcp:4444 format=lime"

On the host, the following command captures the memory dump via TCP port 444 to the file "ram.lime":
$ nc localhost 4444 > ram.lime

3.2 Data Acquisition & Analysis of Phone Internal Memory

Android uses several partitions (like boot, system, recovery, data etc) to organize files and folders on the device just like Windows OS. Each of these partitions has its own functionality. There are mainly 6 partitions that can be found in any Android devices/boot, /system,/recovery, /data, /cache, /misc, According to the Android Architecture the user application datas mainly resides in the /data partition

Phone Internal Acquisition done with dd tool. Acquired the images via Netcat and directly to the sd card & checked the integrity. I preferred acquisition over network since I took dd

images several times considering different scenarios.
dd commands via netcat
adb forward tcp:7000 tcp:7000
dd if=/dev/block/mmcblk0 | busybox nc -l -p 7000
nc 127.0.0.1 8888 > device_image.dd

For Analysis we used the best available tools like Autopsy, Andriller, Bulk Extractor etc.

Autopsy® is the premier open source digital forensics platform that has thousands of users worldwide. It has been developed by Basis Technology and an open source community. Autopsy has the core analysis features that are needed by law enforcement and corporate investigators to conduct an investigation of a hard drive or mobile device. It has both Linux & windows combatable versions and found that the latest Autopsy4.0.0 version in windows is more efficient for Android devices

Analyzed the whole image with autopsy & the located the WhatsApp data. They can be found inside Android device's userdata partition. It has a data sub partition which includes all the installed applications files where WhatsApp can be located /data/data/com.android.whatsapp.Contains subfolders cache, Databases, files including avatars, Log files,& shared preferences which contain several important xml files in encrypted form.

Here we extracted the whole folder including databases opened the db files with SQLite BrowserThere are several tools for viewing WhatsApp chat like WhatsApp viewer, WhatsApp Xtract etc. For that we just need to pull the database file & the WhatsApp key file from the device, decrypt the database & we can read the chat logs.

SQLite Database & Browser

WhatsApp stores all information on a SQLite database: the location and structure of the database are different from platform to platform

WhatsApp databases are msgstore.db, wa.db, axolotl.db, chattsettings.db, websessions.db For Android devices, there are two SQLite databases of value for investigators recovering WhatsApp artifacts: msgstore.db and wa.db. In order to gain access the msgstore.db and wa.db, an investigator must root or get a physical acquisition of the Android device otherwise, WhatsApp also stores a copy of the msgstore.db on the SD card, which is used for backups at the following location:/sdcard/WhatsApp/Databases/ msgstore.db.crypt

One caveat with this file is that it is encrypted and must be decrypted prior to analysis. WhatsApp uses several different types of encryption on this database depending on the version of WhatsApp being used.

The msgstore.db contains details on any chat conversations between a user and their contacts. Wa.db stores information on all the WhatsApp user's contacts. Both of these databases can be found under the databases folder at the following locations:
/data/data/com.whatsapp/databases/msgstore.db
/data/data/com.whatsapp/databases/wa.db

Both databases contains a list of tables .each tables are reserved for different data.

S.No	*SQLite databases*	*Associated tables*
1.	msgstore.db	chat_list group_participants group_participants_history media_refs messages messages_fts messages_fts_content messages_fts_segdir messages_fts_segments props
2	wa.db	android_metadata sqlite_sequence wa_contact_capabilities wa_contacts

The msgstore.db is a relatively simple SQLite database with two tables: chat_list and messages. The messages table contains a listing of all the messages that a user sends or receives from his/her contacts. WhatsApp uses the user's phone number as a unique identifier for both the user and their contacts. This table will include the contact's phone number, message contents, message status, timestamps, and any details around attachments included in the message. Attachments being sent through WhatsApp have their own table entry and the message contents will contain a null entry with a thumbnail and link to the photo/image being shared. This attachment is stored directly in the msgstore.db file. Some of the informative tables in msgstore.db are

- chat_list,
- group_participants,
- media_refs,
- messages

The chat_list table contains a listing of all the phone numbers that a user communicated with; however, this is not a complete listing of the user's contacts. For that we must look at the wa.db. Similar the case with group_participants also.

The wa.db contains a complete listing of a WhatsApp user's contacts including phone number, display name, timestamp, and any other information given upon registering with WhatsApp.

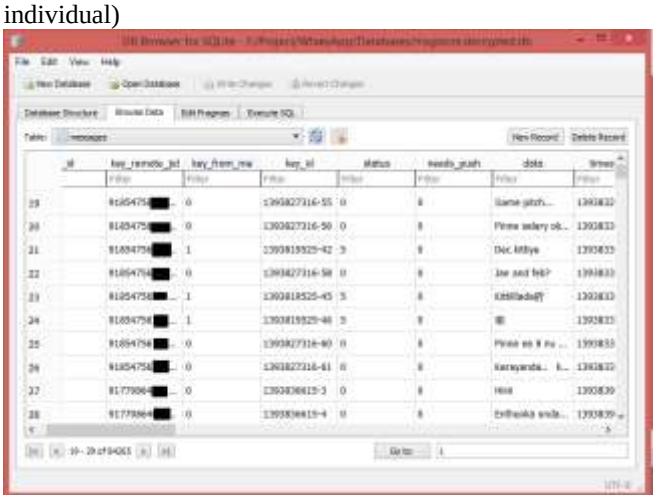
It contains four tables, namely wa_contacts, wa_contact_capabilities, that stores a record for each contact,& contact capabilities android_metadata, and sqlite_sequence, both storing housekeeping information having no evidentiary value.

DB FILE: AXOLOTL.DB

Table: **identities** – Contains the contacts (phone number) to which the chat is done

DB FILE: MSGSTORE.DB

Table: **messages** – Contains all messages (both group and individual)

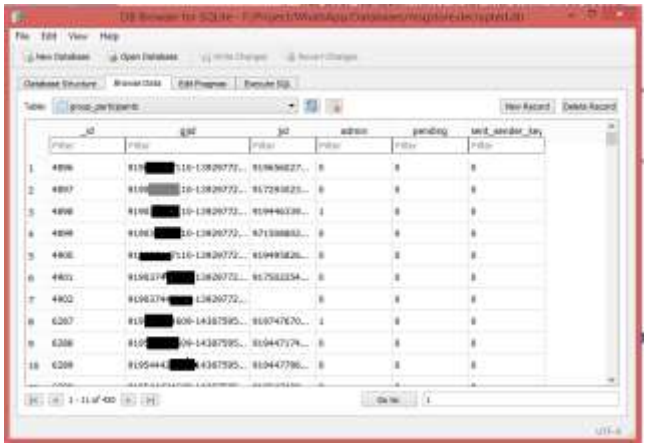


- The **status** column indicates message status values (applicable if there atleast single contact in a group satisfies these)
- 0 – received message
- 13 – sent message seen (blue tick)
- 5 – sent message unseen (double tick but not blue tick)
- 4 – sent message to WhatApp server, but not received in recipient WhatsApp app(single tick)
- 6 – sent/received calls
- **key_from_me** column specifies whether sent or received call.**key_from_me** column has values 0 or 1
- Value 0 – Not from me (contact's)
- Value 1- From me (WhatsApp current user)
- **recipient_count** column has 2 kind of values : value 0 for individual chats or shows groupmember count if it is message in a group
- **read_device_timestamp** is empty if message is not seen by everyone in a group
- **media_name** column contains name of the media storing in Whatsapp folder, which are sent
- **media_name** column contains caption of the media we provided during sending
- **media_url** gives location of the media in WhatsApp server
- **media_wa_type** :message type: '0' text, '1' image,'2' audio, '3' video, '4' contact_ card,'5' geo position)

2.Table: **chat_list** – Contains all contacts (subject column is empty) and groups (has value in subject column) where chat has been done (both sent & receive).
Group has column **creation** showing creation date
message_table_id column field values maps
to **docid** column values in the **messages_fts_content** table (contains all messages), which says last message (sent/received).

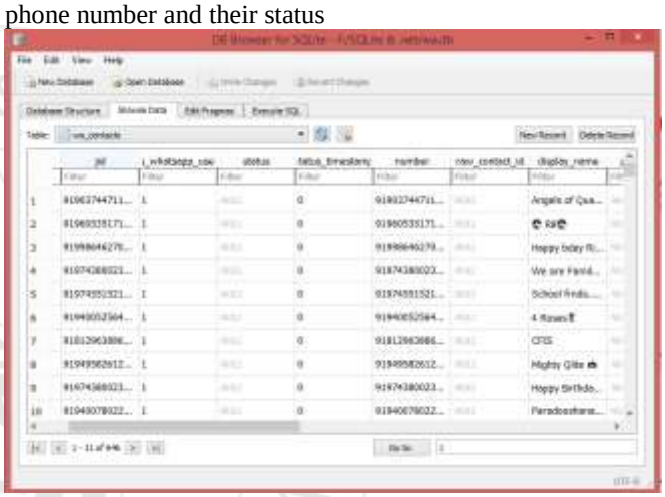
Table 3: group_participants – Contains participants in the group

- **gjid colum** contains group id
- **jid** column contains group participants id (creator does not has this field)
- **admin** colum has values 0 or 1:
- Value 0 – user
- Value 1 – admin



DB FILE: WA.DB

Table: **wa_contacts** – Contains contact's name display name, phone number and their status



3.3 Acquisition of Google backups

Google backup feature for WhatsApp recently developed was warmly accepted. These can be traced if we have the proper Synched account credentials.ie. Needs to get the phone synced email id & credentials, get mail access & download chats backed up in Google drive This is one of the scenarios where RAM Analysis prevails by which we happened to obtain the google backup. Even thou the application uninstalled from device if we have the valid credentials then the chat logs till previous backed up date can be retrieved while we try to re install.

But as per the forensic laws the device 's state should be maintained. Its not a forensically sound method to alter the state. So to obtain the datas I propose a better method to give a try by doing Ethical Hacking . Hack WhatsApp account by using MAC Spoofing Method.ie. The device's Wi-Fi-MAC address can be spoofed with another device. Once its done install WhatsApp in the hacker's phone with the victim's WhatsApp number. Also change to the linked email account also in the hacker's device. At the time of reinstalling we'l be able to fetch the chat logs backed up from victims account. For MAC Spoofing we need busy box & Terminal emulator to be installed in the hacker's device. **Terminal emulator** is used to access Linux command shell of phone

4. Results

When communicating users may exchange plain text messages, as well as multimedia files (containing images, audio, and video), contact cards, and geolocation information, and several kind of documents

Each user is associated with a profile, a set of information that includes his/her WhatsApp name, status line, and avatar (a graphic file, typically a picture). The profile of each user is stored on a central system, from which it is downloaded by other WhatsApp users that include that user in their contacts. The central systems provides other services also, like user registration, authentication, and message relay.

So it is concluded that the only way to acquire WhatsApp histories is imaging end-user devices or pulling data from local or cloud backups.

4.1 Retrieval of Data

Those data like personal & group chats, media files, documents are stored inside msgstore.db & contact data are stored inside wa.db.

Images, Videos, audios contacts, profile pictures, all documents exchanged like pdf that are allocated or those are allocated files can be easily traced from msgstore.db's media_refs & messages tables. Media types can be differentiated with the help of fields like media_wa_type. Whether it is send or received can be analyzed from media_ref table by providing appropriate queries.

WhatsApp contacts & their respective contact names, status associated all can be retrieved from wa.db's wa_contacts table. Groups & group participants can be retrieved by correlating wa.db's wa_contacts & msgstore.db's group_participants.Personal as well as Group Chat logs of each contacts can be found by correlating messages & wa_contacts & group_participants.

Profile pictures, status, group icons & subjects are actually stored in the central server.

WhatsApp stores profile picture in the location \data\com.whatsapp\files\Avatars with extension [.j]. These can be renamed with extension **[.jpg]**, the images will appear. The files are identified by name which are phone numbers

For retrieving datas I extracted the different databases & exported its informative tables into a fresh new SQLite database. Then we fetched datas directly & indirectly by correlating the several tables by writing appropriate queries.

WhatsApp log files are found in zip formats as well as plain text. They contain a huge amount of evidentiary data. All the activities done with the application is logged in the log files with the appropriate timestamps. So it requires a detailed evaluation.

4.2 Deleted files

They are unallocated files seen scattered over android partitions mainly the device's user data partition. Files are difficult to be traced since its format are misplaced. But almost all of the data can be retrieved unless the unallocated space is overwritten Information about the **deleted** contacts & **blocked** contacts can be found from the WhatsApp log files

5. Conclusion

In this paper I have discussed the methods of forensic acquisition & analysis of the artifacts left by WhatsApp Messenger on Android devices, and have shown how these artifacts can provide many information of evidentiary value. Here I focused on fetching the correct databases & to interpret the data in a proper method rather than depending on separate tools so that this method can be followed for almost other applications present in the device. Due Importance is given for RAM Analysis as it is proved to be a critical element in crime scene Investigation. WhatsApp always continues to include new features. Research needed in this field. Here I tried to make a detailed study on WhatsApp web & Cloud backups. Succeeded in retrieving the chat logs, deleted and encrypted messages as well as the VOIP features provided by WhatsApp.
I tried to design a forensically sound methodology to view the recovered artifacts through this research which enhance WhatsApp forensics & that will aid in forensic investigation. As Social messenger applications like WhatsApp are keeping on adding new features which really makes our world much closer more and more research of Android forensics is inevitable.

References

[1] Fan Zhou_, Yitao Yang_, Zhaokun Dingy, Guozi Sun_ 'Dump and Analysis of Android Volatile Memory on Wechat' IEEE ICC 2015
[2] Neha S. Thakur 'Forensic Analysis of WhatsApp on Android Smartphones' in 2013 **University of New Orleans**
[3] Cosimo Anglano 'Forensic analysis of WhatsApp Messenger on Android smartphones' Digital Investigation 11 (2014) 201–213
[4] LiME – Linux Memory Extractor Instructions v1.1 by Digital Forensic Solutions
[5] Nedaa B. Al Barghuthi1 and Huwida Said 'Social Networks IMForensics : EncryptionAnalysis, Journal of Communications Vol. 8, No. 11, November 2013
[6] https://www.quora.com/How-secure-is-WhatsApps-new-end-to-end-encryption
[7] WhatsApp Encryption Overview Technical white paper.pdf
[8] http://www.cyberciti.biz/programming/linux-memory-forensics-analysis-tools/--
[9] A.Hoog,"Android Forensics: Investigation, Analysis and Mobile Security for Google Android," Syngress Publishing, 1st edition, 2011.
[10] http://www.sqlite.org/.

[11] S.Leppert, "Android Memory Dump Analysis," Student Research Paper, Chair of Computer Science 1 (IT-Security), Friedrich-Alexander- University Erlangen-Nuremberg, Germany, 2012.
[12] AndroidMemoryForensicshttps://code.google.com/p/volatility/wiki/AndroidMemoryForensics
[13] https://www.whatsapp.com/faq/en/android/28000019
[14] http://tech.firstpost.com/news-analysis/whatsapp-now-lets-you-backup-conversations-photos-and-videos-to-google-drive-264060.html
[15] https://code.google.com/p/hotoloti/downloads/detail?name=Whatsapp_Xtract_V2.0_2012-05-02.zip

Author Profile

Richu Ann Thomas did her masters MTech in Cyber Forensics & Information Security from College of Engineering Kallooppara under Cochin University of Science & Technology[CUSAT] in 2016 & BTech in Computer Science & Engineering from Mahatma Gandhi University, Kottayam in 2012. During 2012 to 2014 she worked as an Associate Software Engineer in Quadra Software Solutions. She is now working as Information Security Analyst in Skillmine Technologies Bangalore India.