Mobile Forensic Tools: An Insight into WhatsApp Key DB Extractor

George ROTARU
School of Computer Science – University College Dublin
mrgeorge.ro@gmail.com

Gabriela TATU
National Institute for Research & Development in Informatics - ICI Bucharest
gabriela.tatu@ici.ro

Abstract: Due to the popularity of WhatsApp mobile application, many digital investigators are required to extract and analyze conversations stored in its database. In cases where live data forensics is performed, they must act under time pressure, retrieve useful artifacts and leave the smallest possible footprint on the suspect device. From non-rooted Android devices, WhatsApp decrypted datasets and private key can be obtained with the use of WhatsApp Key DB Extractor. The SQLite files can then be parsed and analyzed with an application chosen by the investigator.

Keywords: WhatsApp, Database, Decrypt, Extract, Live Data Forensics

INTRODUCTION

In many cases, digital investigators find themselves in the position of identifying reliable information in the shortest possible amount of time.

Especially when conducting live data forensics, there are situations in which they have to acquire or to exploit critical evidence so that the leading officer can make an informed decision.

As a practical example, consider that an abduction has occurred and the victim's mobile phone has been found at the crime scene. In this case, the digital investigator must perform live data forensics in order to quickly find any kind of information related to that person's last conversations, whereabouts or relevant messages and to narrow down the list of suspects, by aiding the investigation in any way.

Because time is of the essence, the investigator has to target specific mobile applications and to search through their databases in order to find significant data.

One of the most used instant messaging applications both in Romania and worldwide is WhatsApp, so this paper will explain how to retrieve and decrypt its database from an Android mobile phone, without any root access required.

We will accomplish this with the use of WhatsApp Key DB Extractor, rather a batch file than a computer application. We will explain how it works by doing a step-by-step guided practical example.

We will also briefly present WhatsApp Viewer, a tool used to view and parse the extracted database, but this is outside the scope of this paper as an investigator can use any SQLite viewer he or she is accustomed to.

WhatsApp databases can be decrypted with many commercial tools but, in certain cases, it would need more processing time, resources and preparation.

Last but not least, it is always good to have an open source free tool on which you can rely and which you can control and this is exactly what you need: its functionality is transparent and customizable.

LITERATURE REVIEW

WHATSAPP

One of the pioneers in the mobile application category that offer free online chatting (messaging) is WhatsApp. WhatsApp's key appeal is that it allows users to make and receive calls and messages based on a valid telephone number and on an Internet connection. Its capacity to streamline communications, regardless of device or location, has made it the favored choice of over 2 billion users worldwide.

Although WhatsApp provides similar functionality to those of other messaging apps like iMessasge, it provides four distinct advantages, the most important of which is cross-platform compatibility.

A phone number is required to create a WhatsApp account. WhatsApp, unlike other services, does not employ a unique username system based on the user's choice. Instead, it uses his/her phone number to identify him/her. This implies that anyone using the app is added by default to the contact list.

WhatsApp has implemented an end-to-end encryption (SourceForge, 2021), which guarantees that only the recipients of the messages can see them. Third parties are unable to decrypt WhatsApp conversations.

SQLite databases are used to store data. From a digital forensic standpoint, the 2 most relevant files are wa.db and msgstore.db. The first carries contact information, while the second contains message information.

In many cases, a digital forensics investigator can discover msgstore files with crypt* extension. These are msgstore backup files that have been encrypted.

In cases where the examined devices are rooted, the most recent msgstore database in unencrypted format and the cipher key can be exacted with ease (Developer Android, n.d.).

WHATSAPP KEY DB EXTRACTOR

The WhatsApp Key/DB Extractor application was first created by Abinash Bishoyi. It came to assist digital forensic investigators in extracting the cipher key from devices powered by Android OS in a non-rooted state. Additionally, the tool was able to extract the most recent unencrypted files like wa.db (contacts database) and msgstore.db (message database). It's worth mentioning that Android versions 4.0+ are required so that WhatsApp Key/DB Extractor can work (Developer Android, n.d.).

The script uses the Android Debug Bridge drivers and can be deployed on any operating system like Linux, Windows or Mac OS X.

Only logical acquisitions can be performed by WhatsApp Key/DB Extractor. In the process, no supplementary device data or metadata can be obtained (e.g. international mobile equipment identity). Because the script is limited to data acquisition, viewing or interpreting the results must be accomplished with the use of other tools.

The artifacts expected to be extracted by the application are the text message and the image. It is impossible to obtain a document or video artifacts. The picture artifact contains metadata and a low-resolution image of the original photo, based on WhatsApp's thumbnail size. The message artifact gathers information like message content, timestamp, sender and receiver and file attachments (Curl, 2021)

Databases and private key are obtained by applying the apk downgrade technique (Software Informer, 2021).

WHATSAPP VIEWER

WhatsApp Viewer is used to view WhatsApp smartphones chat extracted on a user PC device. WhatsApp Viewer has the ability to display chats from an extracted Android msgstore.db file. This viewer supports crypt5, crypt7, crypt8, and crypt12 versions of the database to be displayed. Extracted messages can be copied to user PC and old message information are easier to read, without pressing "show old messages". The user can export the message information data as HTML, TXT, and JSON files (Digital Forensics, 2016).

PREREQUISITE

For testing, the following hardware and software were used:

- A 64-bit computer running Windows 10 Enterprise (version 20H2, OS build 19042.630);
- Java SE Runtime Environment (build 1.8.0_241-b07) (ADB ClockworkMod, n.d.);
- WhatsApp Key DB Extractor ver.4.7-E1.0 (GnuWin32 – SourceForge, 2009);
- Universal ADB Drivers (Conway, 2017);
- WhatsApp apk ver. 2.11.431 (Oracle, 2020);
- Samsung mobile phone model SM-G965F running Android version 10 and WhatsApp ver. 2.20.205.16;
- Samsung mobile phone model SM-G960F running Android version 10 and WhatsApp ver. 2.20.205.16 (the mobile phones are PIN protected and the password is known; also, they don't have any rootkits installed);
- Samsung USB-C data charging cable;
- WhatsApp Viewer ver. 1.9 (Kunang & Khristian, 2016).

METHODOLOGY

The objective of this paper is to present how to successfully extract and decrypt WhatsApp databases from the two mobile phones previously presented using the script entitled *WhatsApp Key DB Extractor*.

The research is divided into three stages: forensic analysis, results overview and conclusion.

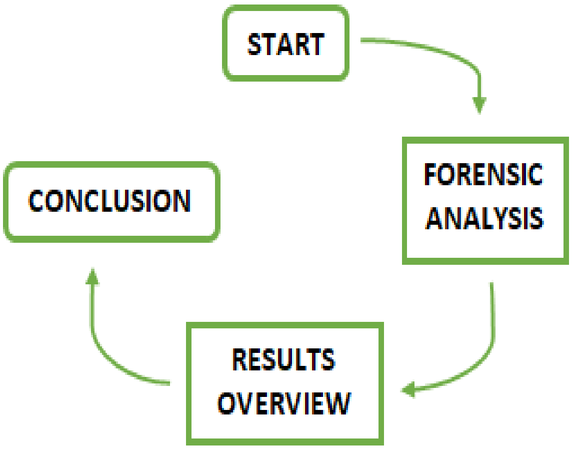


Fig. 1:*Research methodology*

Forensic analysis: We will perform a forensic analysis on the smartphone devices using *WhatsApp Key DB Extractor* in order to obtain a decrypted WhatsApp database from each smartphone. This is not performed under laboratory conditions and there is no control over the datasets as the devices are personal items.

Results overview: To view decrypted data from each extracted database, we will use WhatsApp Viewer. This application is preferred because it somehow simulates WhatsApp interface and it is user friendly.

Conclusion: The success or failure to extract and/or decrypt the databases is presented.

forensic analysis, results overview and conclusion.

LIMITATIONS

Unfortunately, we were unable to conduct the tests in a closed environment and on devices on which we personally generated experimental data and logged every activity. This could lead to truncated results as we cannot verify if all data were extracted or if the database maintains its structure.

Also, we did not have any access to other mobile phones, from other vendors, that run on Android. The comparison of results between different makes and models would have made the findings of this paper more reliable.

None of the devices had a SD card inserted and none could be provided.

Although the tool can run under MacOS and Linux, we did not perform any tests and we cannot present its behavior on these platforms.

The script uses Android's USB backup capability, starting with 4.0 version. It cannot operate on phones or tablets running earlier versions nor on devices that have the backup feature turned off.

Cipher keys can update in WhatsApp on a regular basis. If needed, repeat the entire procedure to get the most recent cipher key.

RESULTS AND DISSCUTION

After downloading the .zip file that contains the application, this was locally extracted, resulting in a parent directory with other three subfolders and multiple files:

[WhatsApp-Key-DB-Extractor-4.7-E1.0]
LICENSE
README.md
WhatsAppKeyDBExtract.bat
WhatsAppKeyDBExtract.ps1
WhatsAppKeyDBExtract.sh
[WhatsApp-Key-DB-Extractor-4.7-E1.0\bin]
abe.jar
adb.exe
AdbWinApi.dll
AdbWinUsbApi.dll
curl.exe
grep.exe
libiconv2.dll
libintl3.dll
pcre3.dll
regex2.dll
tar.exe
[WhatsApp-Key-DB-Extractor-4.7-E1.0\extracted]
.placeholder
[WhatsApp-Key-DB-Extractor-4.7-E1.0\tmp]
.placeholder

From the main folder, we will only refer to WhatsAppKeyDBExtract.bat, as this runs under Windows.

Download WhatsApp apk ver. 2.11.431 and save it in the tmp folder with the name *LegacywhatsApp.apk*. This will skip the process of downloading the *.apk* used for downgrading, after you run the script.

Parse WhatsAppKeyDBExtract.bat and understand its design and functionality, line by line, command by command. We will not describe every single line of the script as it also contains commands for the command-line interpreter and the comments in the script are pretty much self-explanatory.

MOBILE DEVICE PREPARATION

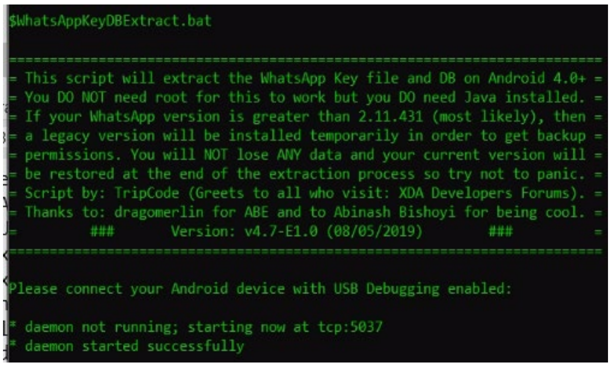
- Access the *Settings* menu in WhatsApp and under *Chats* select *Chat backup* and create a fresh backup of all the conversations;
- Enable *USB debugging* by accessing *Settings – About phone from the menu*;
- Under *Software information* tap *Build number* seven times;
- Enter phone lock code to unlock debugging menu (which will appear last in the main settings menu);
- Access *Developer options* and enable *USB debugging* (personally, I always enable the *Stay awake* option too, so my phone's screen will not go to sleep while charging);
- Connect the mobile device to the computer using the USB-C cable (be sure this is the only Android device connected at that time).

DEPLOY THE SCRIPT

Open a command line, go to the main directory of the script and run WhatsAppKeyDBExtract.bat.

In the following, we will explain what you will encounter in different stages of the running process:

bin\adb.exe kill-server
bin\adb.exe start-server
bin\adb.exe wait-for-device



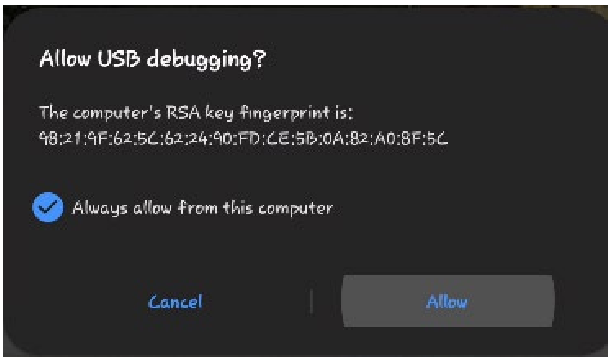
```
$WhatsAppKeyDBExtract.bat


=========================================================================
= This script will extract the WhatsApp Key file and DB on Android 4.0+ =
= You DO NOT need root for this to work but you DO need Java installed. =
= If your WhatsApp version is greater than 2.11.431 (most likely), then =
= a legacy version will be installed temporarily in order to get backup =
= permissions. You will NOT lose ANY data and your current version will =
= be restored at the end of the extraction process so try not to panic. =
= Script by: TripCode (Greets to all who visit: XDA Developers Forums). =
= Thanks to: dragomerlin for ABE and to Abinash Bishoyi for being cool. =
=        ###       Version: v4.7-E1.0 (08/05/2019)        ###           =
=========================================================================


Please connect your Android device with USB Debugging enabled:

* daemon not running; starting now at tcp:5037
* daemon started successfully
```

Fig. 2:*Script deployment message*

Android Debug Bridge (GnuWin32 – SourceForge, 2003) is used to communicate with the mobile phone. In case a server is already running, the service will be stopped and restarted from the bin directory, waiting for the user to attach a device.

At this step, we have connected the device which runs WhatsApp.

A message will appear on the mobile screen, asking you if you allow USB debugging. Select "allow" and be sure to check the *Always allow from this computer* option so that you won't be prompted every time you connect.



Allow USB debugging?

The computer's RSA key fingerprint is:
98:21:9F:62:5C:62:24:90:FD:CE:5B:0A:82:A0:8F:5C

☑ Always allow from this computer

Fig. 3:*Android message for computer connectivity*

If the script returns an error at this stage and stops from executing, just run it again. We believe it does not wait for the user to set up the mobile device and adb fails to connect.

bin\adb.exe shell getprop ro.build.version.sdk

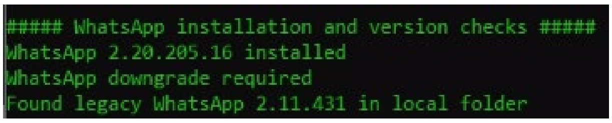
Get Android sdk version and test, in the following lines, if it is Android 4 or higher, for compatibility.

bin\adb.exe shell pm path com.whatsapp | bin\grep.exe package

Get the path of the WhatsApp apk currently running on the system. The output of the first command is piped to grep (Umar, Riadi & Zamroni, 2017), which searches for the string package and outputs the lines containing it.

bin\adb.exe shell dumpsys package com.whatsapp | bin\grep.exe versionName

Get version number for the currently installed WhatsApp.



```
##### WhatsApp installation and version checks #####
WhatsApp 2.20.205.16 installed
WhatsApp downgrade required
Found legacy WhatsApp 2.11.431 in local folder
```

Fig. 4:Displaying WhatsApp installed version

bin\curl.exe -sI http://www.cdn.whatsapp.net/android/2.11.431/WhatsApp.apk | bin\grep.exe Content-Length

curl (WhatCrypt, n.d.) is a command line tool for transferring data with URLs. The command gets the size of this specific apk (*-I* option is for showing document info only and -s puts curl into silent mode thus no output is returned) and the following tests it against a predefined length.

bin\adb.exe shell am force-stop com.whatsapp

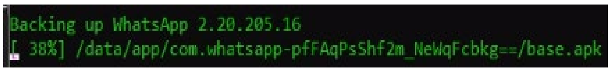
bin\adb.exe shell am kill com.whatsapp

According to the Android skd version installed, WhatsApp service is killed by one of the previous commands.

bin\adb.exe pull %apkpath% tmp

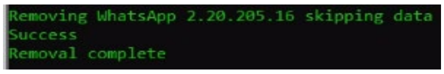
bin\adb.exe shell pm uninstall -k com.whatsapp

Current WhatsApp apk backup and uninstalling are performed.



```
Backing up WhatsApp 2.20.205.16
[ 38%] /data/app/com.whatsapp-pfFAqPsShf2m_NeWqFcbkg==/base.apk
```

Fig. 5:Running current WhatsApp version backup

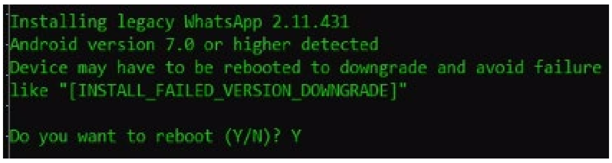


```
Removing WhatsApp 2.20.205.16 skipping data
Success
Removal complete
```

Fig. 6:Uninstalling WhatsApp current version

bin\adb.exe reboot

When prompted, select to reboot the mobile device in order to avoid new WhatsApp installation failure. After reboot, unlock the device, if necessary, and you will see WhatsApp is no more installed.

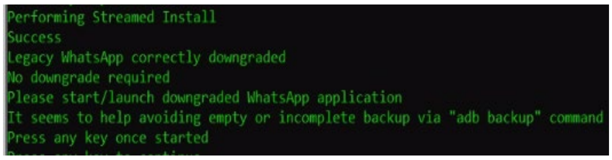


```
Installing legacy WhatsApp 2.11.431
Android version 7.0 or higher detected
Device may have to be rebooted to downgrade and avoid failure
like "[INSTALL_FAILED_VERSION_DOWNGRADE]"

Do you want to reboot (Y/N)? Y
```

Fig. 7:*Installing legacy WhatsApp*

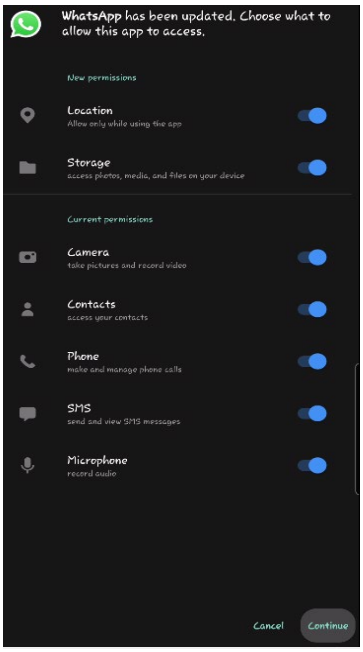
LegacyWhatsApp.apk is installed according to user preference and to Android sdk version, (-*r* option reinstalls the app keeping its data and -*d* allows to downgrade the version code).

When WhatsApp update prompt appears on device, select *Continue* then search for and open the application.



```
Performing Streamed Install
Success
Legacy WhatsApp correctly downgraded
No downgrade required
Please start/launch downgraded WhatsApp application
It seems to help avoiding empty or incomplete backup via "adb backup" command
Press any key once started
```

Fig. 8:*Legacy WhatsApp successful installation*



WhatsApp has been updated. Choose what to allow this app to access.

New permissions

Location
Allow only while using the app

Storage
access photos, media, and files on your device

Current permissions

Camera
take pictures and record video

Contacts
access your contacts

Phone
make and manage phone calls

SMS
send and view SMS messages

Microphone
record audio

Fig. 9:*Display of Android message after installing legacy WhatsApp*

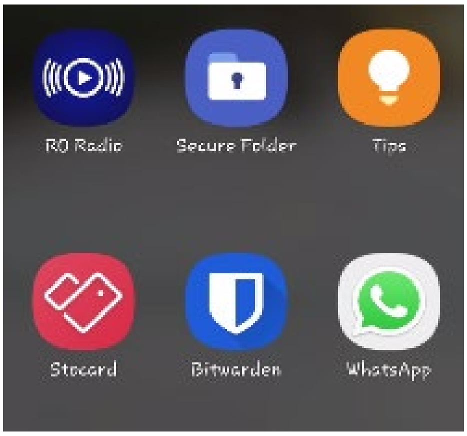
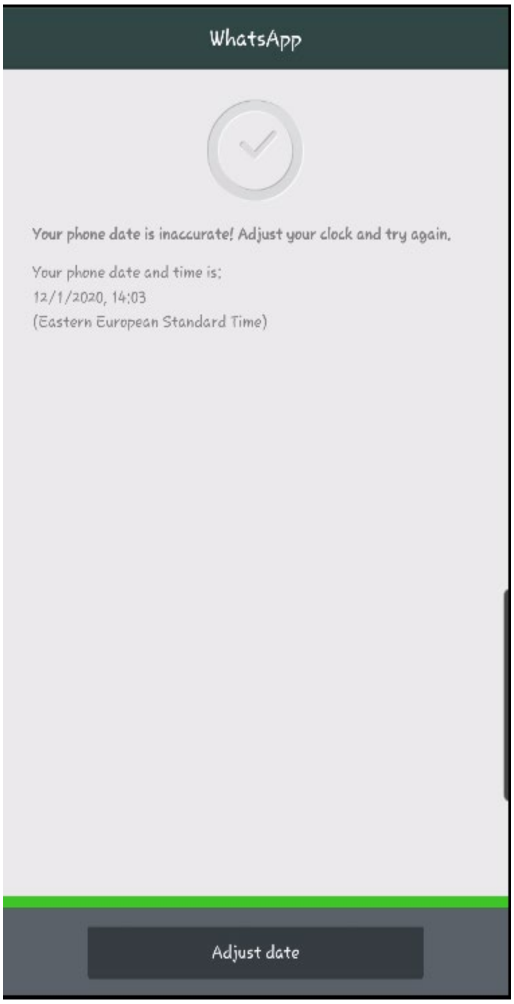


Fig. 10:*Display of Android legacy WhatsApp menu icon*

On opening, you will receive again a prompt showing inaccurate phone date and time. Select *Adjust date* and keep this window open. Then allow the script to continue executing in the command line.

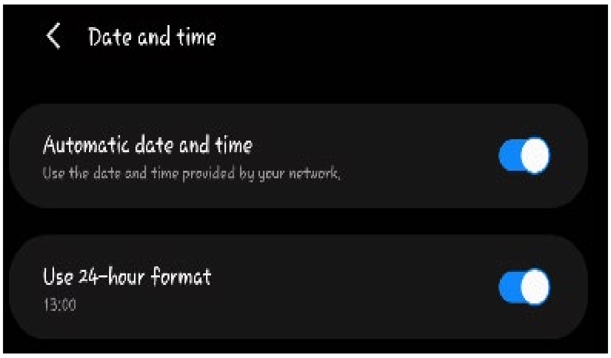


WhatsApp

Your phone date is inaccurate! Adjust your clock and try again.

Your phone date and time is:
12/1/2020, 14:03
(Eastern European Standard Time)

Fig. 11:*Display of WhatsApp date error*



Date and time

Automatic date and time
Use the date and time provided by your network.

Use 24-hour format
13:00

Fig. 12:Display of Android date and time options

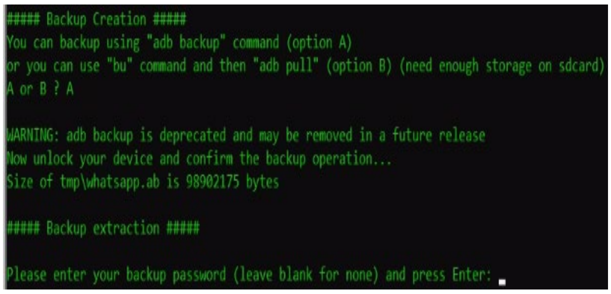
```
bin\adb.exe shell dumpsys package com.whatsapp | bin\grep.exe versionName
```

Get version number of newly installed WhatsApp to see if apk downgrade was successful.

By considering the backup options provided by the script and the hardware limitations, we run only the scenario where the backup was saved on the local machine, in the tmp folder.

```
bin\adb.exe backup -f tmp\whatsapp.ab com.whatsapp
bin\adb.exe backup -f tmp\whatsapp.ab -noapk com.whatsapp
```

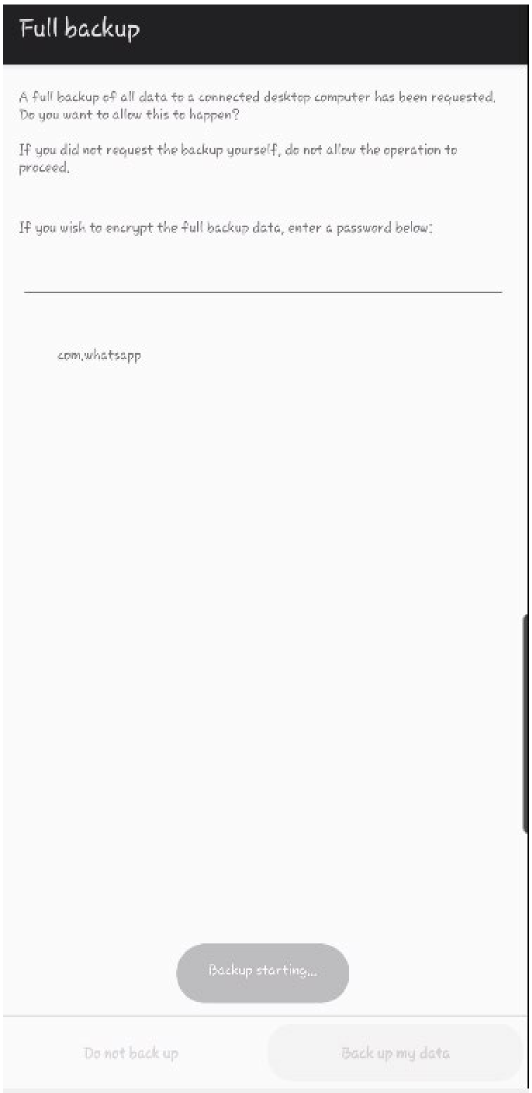
Considering Android sdk version, a non-password protected backup of for WhatsApp is deployed in tmp folder.



```
##### Backup Creation #####
You can backup using "adb backup" command (option A)
or you can use "bu" command and then "adb pull" (option B) (need enough storage on sdcard)
A or B ? A

WARNING: adb backup is deprecated and may be removed in a future release
Now unlock your device and confirm the backup operation...
Size of tmp\whatsapp.ab is 98902175 bytes

##### Backup extraction #####

Please enter your backup password (leave blank for none) and press Enter:
```

Fig. 13:Script options for WhatsApp backup

Select *Back up my data* from the new prompt, without entering a password. After it finishes, the screen goes back to the *Date and time* selection. Leave it be until the script reinstalls a previous WhatsApp version.



Full backup

A full backup of all data to a connected desktop computer has been requested. Do you want to allow this to happen?

If you did not request the backup yourself, do not allow the operation to proceed.

If you wish to encrypt the full backup data, enter a password below:

com.whatsapp

Fig. 14:Display of Android WhatsApp backup warning

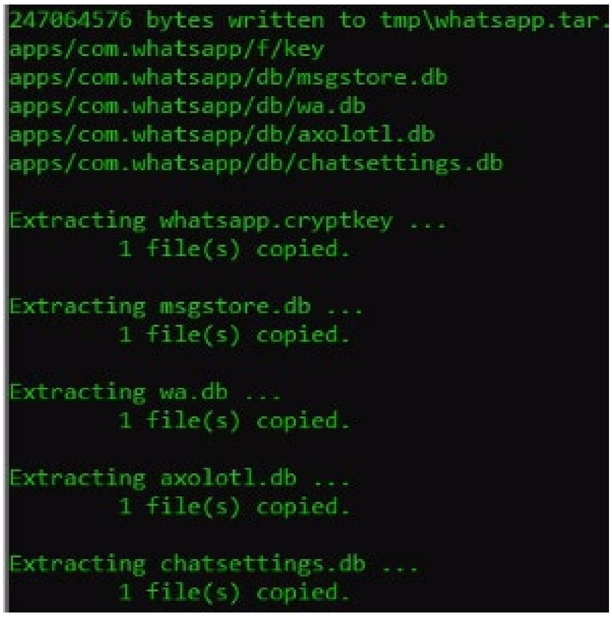
```
java -jar bin\abe.jar unpack tmp\whatsapp.ab tmp\whatsapp.tar
```

Using Android Backup Extractor java library (GitHub, 2019), the backup is unpacked in an archive named whatsapp.tar.

```
bin\tar.exe xvf tmp\whatsapp.tar -C tmp\apps/com.whatsapp/*
```

Using tar (WhatsApp, 2021), specific files from the archive are extracted to tmp folder (using the options *x* for extraction, *v* for verbose output, *f* to specify archive file and -C to change directory).

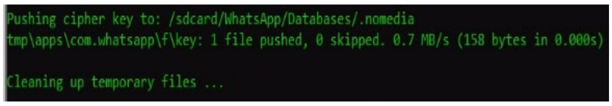
At this stage, encryption keys, msgstore.db, wa.db, axolotl1.db and chatsettings.db, respectively, are extracted.



```
247064576 bytes written to tmp\whatsapp.tar.
apps/com.whatsapp/f/key
apps/com.whatsapp/db/msgstore.db
apps/com.whatsapp/db/wa.db
apps/com.whatsapp/db/axolotl.db
apps/com.whatsapp/db/chatsettings.db

Extracting whatsapp.cryptkey ...
        1 file(s) copied.

Extracting msgstore.db ...
        1 file(s) copied.

Extracting wa.db ...
        1 file(s) copied.

Extracting axolotl.db ...
        1 file(s) copied.

Extracting chatsettings.db ...
        1 file(s) copied.
```

Fig. 15:*File extraction in progress*

Next lines test if extraction succeeded, copy files in extracted folder and deletes the temporary ones. A backup of the cipher key is also copied in the WhatsApp database directory of the system, where it is stored in a file named .nomedia. The goal is to provide Android developers with a consistent approach for offering WhatsApp decryption for app customers who want to run the script.

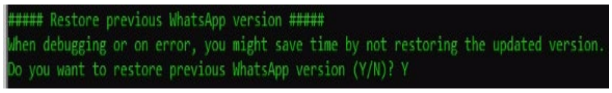


```
Pushing cipher key to: /sdcard/WhatsApp/Databases/.nomedia
tmp\apps\com.whatsapp\f\key: 1 file pushed, 0 skipped. 0.7 MB/s (158 bytes in 0.000s)

Cleaning up temporary files ...
```

Fig. 16:*Finishing file extraction*

bin\curl.exe -o tmp\%apkname% http://www.cdn.whatsapp.net/android/%versionName%/WhatsApp.apk

If user chooses to restore the previous WhatsApp version on the device, the script checks for the backup apk and if does not exist, it will be downloaded.

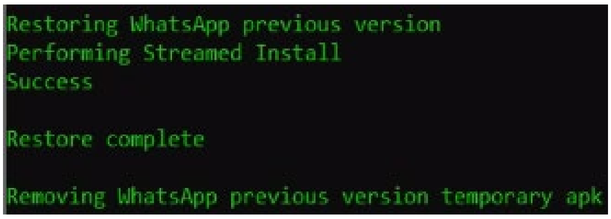


```
##### Restore previous WhatsApp version #####
When debugging or on error, you might save time by not restoring the updated version.
Do you want to restore previous WhatsApp version (Y/N)? Y
```

Fig. 17:*Options for reinstalling previous WhatsApp version*

bin\adb.exe install -r -d tmp\%apkname%
bin\adb.exe install -r tmp\%apkname%

Command lines for WhatsApp reinstallation, consistent with the running Android sdk version.



```
Restoring WhatsApp previous version
Performing Streamed Install
Success

Restore complete

Removing WhatsApp previous version temporary apk
```

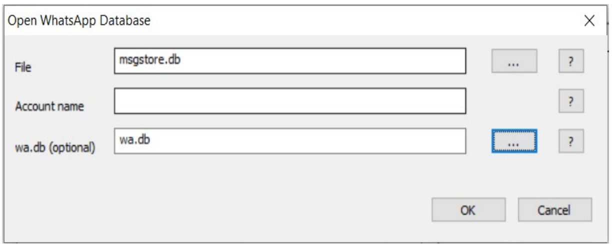
Fig. 18:*Successful reinstallation*

bin\adb.exe kill-server

The final step is to stop the adb server thus disconnecting from the device.

VIEW DECRYPTED DATASETS

Using WhatsApp Viewer, we opened msgstore.db (main database file) and wa.db (used for resolving contact names) files.



Open WhatsApp Database

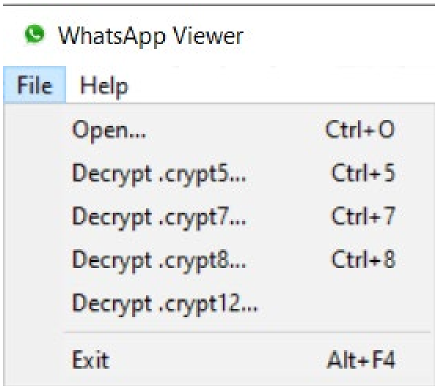
File: msgstore.db

Account name:

wa.db (optional): wa.db

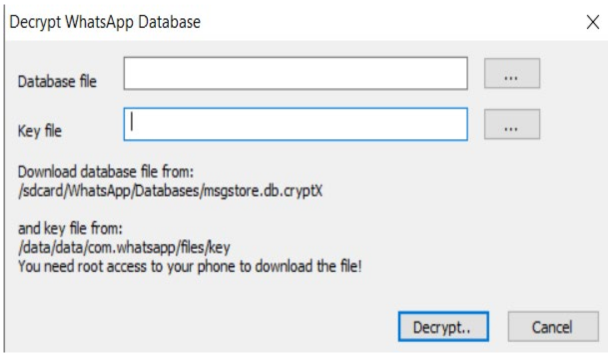
Fig. 19:*WhatsApp Viewer open file options*

There is also an option to decrypt a .crypt* file by providing a database file and a key file.



WhatsApp Viewer
File
Help
Open... Ctrl+O
Decrypt .crypt5... Ctrl+5
Decrypt .crypt7... Ctrl+7
Decrypt .crypt8... Ctrl+8
Decrypt .crypt12...
Exit Alt+F4

Fig. 20:*WhatsApp Viewer file options*



Decrypt WhatsApp Database

Database file

Key file

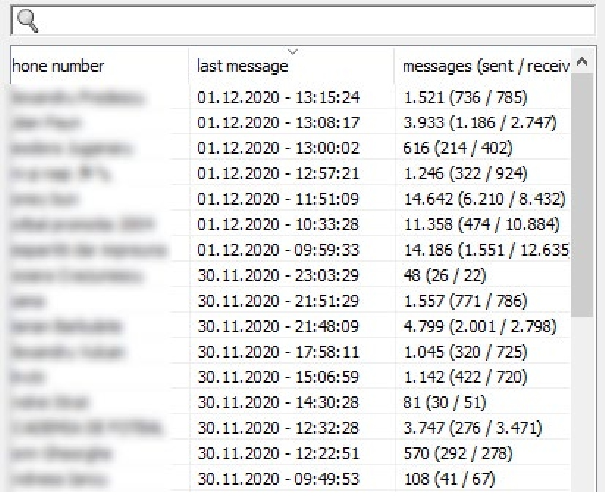
Download database file from:
/sdcard/WhatsApp/Databases/msgstore.db.cryptX

and key file from:
/data/data/com.whatsapp/files/key
You need root access to your phone to download the file!

Fig. 21:*WhatsApp Viewer file decryption options*

After importing the data, we get a panel on the left side of the window with a search box and three columns: phone number (which, in my case, displays the contact name if the number is saved, the phone number and the group name), last message (date and time) and messages (the total number of messages divided into sent and received).

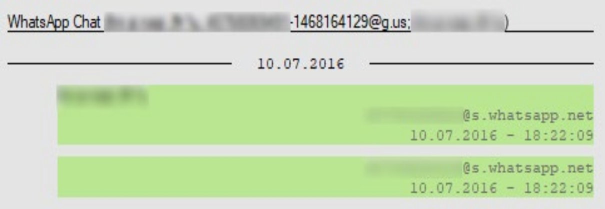
Unfortunately, the search box performs searches only in the phone number column (by name and number). It does not have any option to either filter by date and time or to search in all the conversations by key words.



hone number	last message	messages (sent / receiv
[redacted]	01.12.2020 - 13:15:24	1.521 (736 / 785)
[redacted]	01.12.2020 - 13:08:17	3.933 (1.186 / 2.747)
[redacted]	01.12.2020 - 13:00:02	616 (214 / 402)
[redacted]	01.12.2020 - 12:57:21	1.246 (322 / 924)
[redacted]	01.12.2020 - 11:51:09	14.642 (6.210 / 8.432)
[redacted]	01.12.2020 - 10:33:28	11.358 (474 / 10.884)
[redacted]	01.12.2020 - 09:59:33	14.186 (1.551 / 12.635
[redacted]	30.11.2020 - 23:03:29	48 (26 / 22)
[redacted]	30.11.2020 - 21:51:29	1.557 (771 / 786)
[redacted]	30.11.2020 - 21:48:09	4.799 (2.001 / 2.798)
[redacted]	30.11.2020 - 17:58:11	1.045 (320 / 725)
[redacted]	30.11.2020 - 15:06:59	1.142 (422 / 720)
[redacted]	30.11.2020 - 14:30:28	81 (30 / 51)
[redacted]	30.11.2020 - 12:32:28	3.747 (276 / 3.471)
[redacted]	30.11.2020 - 12:22:51	570 (292 / 278)
[redacted]	30.11.2020 - 09:49:53	108 (41 / 67)

Fig. 22:*WhatsApp Viewer conversation list*

Once a conversation is selected, it is displayed in the main window. Contact details and messages are displayed, with the oldest entry first. In case of groups, you can see when it was created, by whom and when the user joined the group.



WhatsApp Chat [redacted]-1468164129@g.us; [redacted])

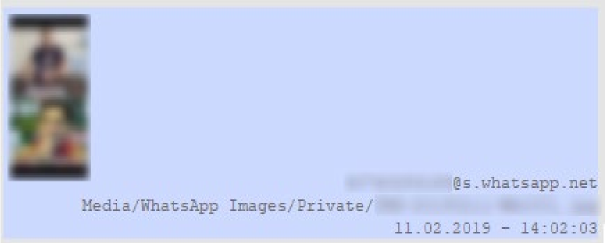
10.07.2016

[redacted]
[redacted]@s.whatsapp.net
10.07.2016 - 18:22:09

[redacted]@s.whatsapp.net
10.07.2016 - 18:22:09

Fig. 23:*WhatsApp Viewer conversation view*

In case of media or other types of files, metadata and the path to store location are displayed. In case of an image or video files, a thumbnail is presented.



[redacted]@s.whatsapp.net
Media/WhatsApp Images/Private/[redacted]
11.02.2019 - 14:02:03

Fig. 24:*WhatsApp Viewer conversation view*

There is a clear separator between messages from different dates.

An entire conversation can be exported into *.txt*, *.html* or *.json* format.

DISCUSSION

By reviewing the data resulted, we can spot a few shortcomings regarding the acquisition process, from a digital forensics perspective: the first would be that no hashing algorithm is performed on the original files processed on the device and, furthermore, the original extracted files are not kept for later use, for comparison or for certifying the origin of the datasets (e.g. Android built-in *md5sum*); secondly, no audit trail that could emphasize user actions taken on the suspect device is generated.

Furthermore, the script does not extract artifacts like photos, documents or any kind of media files. As it is well known, these could hold a key evidence for an investigation and we would have to rely on other tools or to pull them manually using adb. Thus, media or document files were not examined. The main focus of the script is to collect SQLite tables and this leaves out other accessible folders which contain sent or received files.

By viewing the datasets with WhatsApp Viewer, thumbnails of pictures and videos will appear, with their corresponding metadata, but the original files must be extracted in another way.

Although present, deleted SQLite artifacts were not parsed by the viewer.

The script was not designed to extract all the data from WhatsApp folders. It aids an investigation by decrypting an enormous amount of data that can be analyzed in any SQLite viewer thus allowing to quickly search and retrieve useful artifacts (even deleted ones). A decrypted dataset from one of our test phones was larger than 200Mb and contained hundreds of conversations, not to mention the multitude of messages or words. This was obtained in under 10 minutes.

Another strong point of the script is that, with a basic knowledge of the Bash script, anyone can understand how it works and can modify and improve it according to his/her needs. Also, is comes with embedded executable files and libraries (except for JRE) so you would not rely on the applications installed (or not) on the operating system it runs.

Overall, we are pleased with its functionality and we think it accomplishes the mission for which it was created.

The script is easy to run, to debug, to configure and to customize.

CONCLUSIONS

On not rooted Android devices, with no or known pass code, the script works like a charm. Not only it is able to retrieve the most recent backup conversations from WhatsApp, but it also gets the encryption key, in a matter of minutes (depending on the size of the backup and on the hardware configuration of the devices).

It is reliable, stable and can decrypt databases even on devices running the most recent updates.

Datasets can be parsed with any SQLite viewer and important artifacts can be retrieved by an investigator.

REFERENCE LIST

ADB ClockworkMod (n.d.). *Universal ADB Drivers*. https://adb.clockworkmod.com/

Conway, A. (2017, August 25). *How to Downgrade an App on Android – No Root Needed* XDA Developers. https://www.xda-developers.com/downgrade-an-app-android-no-root/

Curl (2021). *curl 7.79.1 for Windows*. https://curl.se/windows/

Developer Android (n.d.). *Android Debug Bridge (adb)*. https://developer.android.com/studio/command-line/adb

Digital Forensics (2016). *Extracting WhatsApp database and the cipher key from a non-rooted Android device*. https://www.digitalforensics.com/blog/extracting-whatsapp-database-and-the-cipher-key-from-a-non-rooted-android-device/

GitHub (2019). *p4r4d0x86 WhatsApp Key/DB Extractor*. https://github.com/p4r4d0x86/WhatsApp-Key-DB-Extractor

GnuWin32 – SourceForge (2003). *Tar (archiver tools) for Windows*. http://gnuwin32.sourceforge.net/packages/gtar.htm

GnuWin32 – SourceForge (2009). *Grep for Windows*. http://gnuwin32.sourceforge.net/packages/grep.htm

Kunang, Y. N. & Khristian, A. (2016). Implementation of Forensic. Procedures for WhatsApp Applications on Android Phones. *Annual Research Seminar*, *2*(1), 102-105.

Oracle (2020). JDK *8u241 Update Release Notes – Java™ SE Development* Kit 8, *Update 241* (JDK *8u241*). https://www.oracle.com/java/technologies/javase/8u241-relnotes.html

SourceForge (2021). *Android backup processor*. https://sourceforge.net/projects/adbextractor/

Software Informer (2021). *WhatsApp Viewer 1.9.* https://whatsapp-database-viewer.software.informer.com/1.9/

Umar, R., Riadi, I. & Zamroni G. M. (2017). A Comparative Study of Forensic Tools for WhatsApp Analysis using NIST Measurements. *International Journal of Advanced Computer Science and Applications*, *8*(12), 69-75.

WhatCrypt (n.d.). *WhatCrypt Tools*. http://whatcrypt.com/WhatsApp-2.11.431.apk

WhatsApp (2021). *WhatsApp Security.* https://www.whatsapp.com/security/