

EXCELLIUM

L'Investigateur, le Smartphone et l'Application WhatsApp

Guenaëlle De Julis



Bio

- ~8 ans dans la sécurité informatique
- Membre de CERT-XLM depuis 1 an

- Consultante en entreprise
 - Analyse et corrélation de journaux
 - Sécurité des applications
 - Formation aux développeurs
 - Projets
 - Authentification à divulgation nulle de connaissance
 - Détection de comportement frauduleux
 - Protection de contenu

- Cryptographie
 - Analyses de source d'aléa physique
 - Méthodologies d'évaluation
 - Correction d'anomalies

Au menu

- Acquisition
 - **Spécificités** de l'application WhatsApp
 - **Stockage** des conversations et contacts
 - Autre **artefacts**
 - En pratique, Android **versus** iOS
- Structure et contenu des données
 - Android **versus** iOS
 - Identification des **personnes**
 - Attribution et contenu des **conversations** (directes et groupes)
 - **Traces** des opérations
- Analyse
 - Extraction, agrégation et **automatisation**
 - Python et la bibliothèque **pandas**

Acquisition

- L'application WhatsApp
 - Stockage : **SQLite**
 - **Confidentialité**: Chiffrement de bout-en-bout

« WhatsApp end-to-end encryption ensures only you and the person you're communicating with can read what's sent, and nobody in between, **not even WhatsApp** »

faq.whatsapp.com/general/28030015

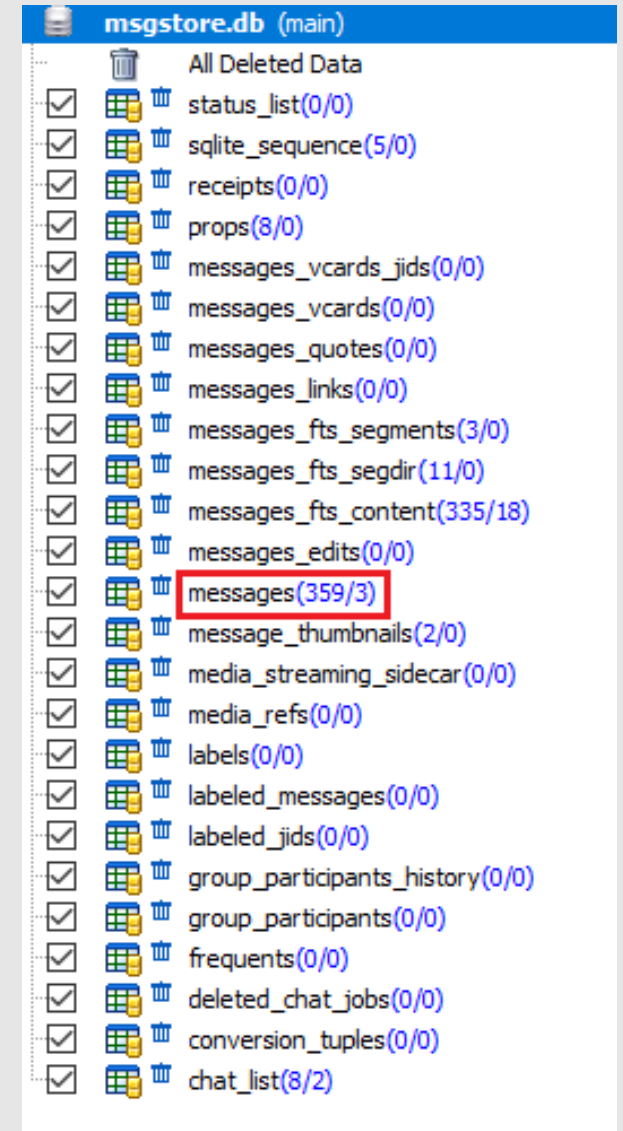
- iOS
 - Sauvegarde par **iTunes**
 - Export par **iExplorer** (ou équivalent)
 - Conversations et contacts **en clair**:
 - group.net.whatsapp.WhatsApp.shared/ChatStorage.sqlite*
 - group.net.whatsapp.WhatsApp.shared/ContactsV2.sqlite*
 - Accès au système de fichiers requis pour d'**autres artefacts** (journaux, mot de passe, BDD temporaire)
 - net.whatsapp.WhatsApp/Documents/*

Acquisition

- Android
 - **Sauvegardes** des conversations sans accès root mais **chiffrées** (variante de AES256):
WhatsApp/Databases/msgstore.db.crypt{0-12}
 - Clef de déchiffrement et BDDs en clair: **accès root requis**
data/com.whatsapp/files/key
data/com.whatsapp/databases/wa.db
data/com.whatsapp/databases/msgstore.db
data/com.whatsapp/XXX : autres artefacts (configurations, journaux)
 - Hack **sans root** (résultat non garanti)
 - pour Android ≥ 4.4 et < 7 : **downgrade** de l'application à 2.11.431 + *adb*:
<https://forum.xda-developers.com/showthread.php?t=2770982>
 - pour Android > 7 : « réinstallation » de l'application (**downgrade impossible** commit du 24/02/2016 sur l'OS)
<https://plainsec.org/extracting-cipher-key-from-whatsapp-on-android-7-and-greater-without-root/>
 - depuis une sauvegarde **Google Drive**:
<https://forum.xda-developers.com/android/apps-games/tool-whatsapp-google-drive-extractor-t3483633>
 - application de sauvegarde locale fournie par certains constructeurs

Structure et contenu des données - Android

- Conversations (*msgstore.db*)
 - 25 tables
 - L'essentiel dans *messages*
- Contacts (*wa.db*)
 - 11 tables
 - Liée au carnet d'adresse
 - L'essentiel dans *wa_contacts*
- Messages supprimés
 - Déchiffrer une **sauvegarde**
 - **Journaux**
- Sources:
 - https://www.group-ib.com/blog/whatsapp_forensic_artifacts
 - <https://arxiv.org/pdf/1507.07739>



msgstore.db (main)	
<input type="checkbox"/>	All Deleted Data
<input checked="" type="checkbox"/>	status_list(0/0)
<input checked="" type="checkbox"/>	sqlite_sequence(5/0)
<input checked="" type="checkbox"/>	receipts(0/0)
<input checked="" type="checkbox"/>	props(8/0)
<input checked="" type="checkbox"/>	messages_vcards_jids(0/0)
<input checked="" type="checkbox"/>	messages_vcards(0/0)
<input checked="" type="checkbox"/>	messages_quotes(0/0)
<input checked="" type="checkbox"/>	messages_links(0/0)
<input checked="" type="checkbox"/>	messages_fts_segments(3/0)
<input checked="" type="checkbox"/>	messages_fts_segdir(11/0)
<input checked="" type="checkbox"/>	messages_fts_content(335/18)
<input checked="" type="checkbox"/>	messages_edits(0/0)
<input checked="" type="checkbox"/>	messages(359/3)
<input checked="" type="checkbox"/>	message_thumbnails(2/0)
<input checked="" type="checkbox"/>	media_streaming_sidecar(0/0)
<input checked="" type="checkbox"/>	media_refs(0/0)
<input checked="" type="checkbox"/>	labels(0/0)
<input checked="" type="checkbox"/>	labeled_messages(0/0)
<input checked="" type="checkbox"/>	labeled_jids(0/0)
<input checked="" type="checkbox"/>	group_participants_history(0/0)
<input checked="" type="checkbox"/>	group_participants(0/0)
<input checked="" type="checkbox"/>	frequents(0/0)
<input checked="" type="checkbox"/>	deleted_chat_jobs(0/0)
<input checked="" type="checkbox"/>	conversion_tuples(0/0)
<input checked="" type="checkbox"/>	chat_list(8/2)

Structure et contenu des données - Android

- **Groupe** de discussion :
 - une entrée par destinataire
- Sons/Vidéos/Images :
 - stockés dans *WhatsApp/Media*
 - en base : URL **serveur central**
- Champs pertinents dans *messages* :

key_remote_jid	Jabber ID de l'auteur du message (serveur central pour un groupe de discussion)
key_from_me	'0' pour message reçu '1' pour message envoyé
status	'0' reçu '4' attente du serveur central '5' reçu par le destinataire '13' lu par le destinataire

timestamp	Date d' envoi pour un message envoyé (insertion dans la base pour un reçu)
received_timestamp	Date de réception d'un message
receipt_server_timestamp	Date de réception par le serveur pour un envoi
receipt_device_timestamp	Date de réception du destinataire pour un envoi
read_device_timestamp	Date de lecture par le destinataire
raw_data	Miniature si image ou vidéo
remote_resource	Jabber ID de l'émetteur pour une discussion de groupe
media_wa_type	'0' texte, '1' image, '2' audio, '3' vidéo, '4' contact, '5' position géographique, ...
media_url	URL sur le serveur central pour un multimédia
media_mimetype	Pour un message multimédia
media_size	Pour un message multimédia
media_hash	Pour un message multimédia
data	Texte

Structure et contenu des données - iOS

- WhatsApp version: **2.19.92**
- Conversations (*ChatStorage.sqlite*)
 - 18 tables
 - L'essentiel dans *ZWMESSAGE*
 - Mais besoin de
 - ZWMESSAGEINFO*
 - ZWAMEDIAITEM*
 - ZWAGROUPINFO*
 - ZWAGROUPMEMBER*
- Contacts (*ContactsV2.sqlite*)
 - 4 tables
 - L'essentiel dans *ZWAADDRESSBOOKCONTACT*

ChatStorage.sqlite (main)	
	All Deleted Data
<input checked="" type="checkbox"/>	ZWAZ1PAYMENTTRANSACTION(0/0)
<input checked="" type="checkbox"/>	ZWAVCARDMENTION(0/0)
<input checked="" type="checkbox"/>	ZWAPROFILEPUSHNAME(222/0)
<input checked="" type="checkbox"/>	ZWAPROFILEPICTUREITEM(275/0)
<input checked="" type="checkbox"/>	ZWMESSAGEINFO(698/1)
<input checked="" type="checkbox"/>	ZWMESSAGEDATAITEM(461/2)
<input checked="" type="checkbox"/>	ZWMESSAGE(4447/32)
<input checked="" type="checkbox"/>	ZWAMEDIAITEM(1560/41)
<input checked="" type="checkbox"/>	ZWAGROUPMEMBERSCHANGE(14/0)
<input checked="" type="checkbox"/>	ZWAGROUPMEMBER(606/4)
<input checked="" type="checkbox"/>	ZWAGROUPINFO(5/0)
<input checked="" type="checkbox"/>	ZWACHATSESSION(12/0)
<input checked="" type="checkbox"/>	ZWACHATPUSHCONFIG(3/0)
<input checked="" type="checkbox"/>	ZWACHATPROPERTIES(0/0)
<input checked="" type="checkbox"/>	ZWABLACKLISTITEM(7/0)
<input checked="" type="checkbox"/>	Z_PRIMARYKEY(15/0)
<input checked="" type="checkbox"/>	Z_MODELCACHE(1/0)
<input checked="" type="checkbox"/>	Z_METADATA(1/0)

Structure et contenu des données - iOS

- Champs pertinents dans *ZWAMESSAGE*

Z_PK	Clef primaire, = ZMESSAGE dans les autres tables
ZMESSAGESTATUS	'0' pour opération de gestion '1' reçu par tous les destinataires '8' lu par tous les destinataires (autres codes '3', '5', '6' mais signification non identifiée)
ZMESSAGETYPE	'0' texte, '1' image, '2' vidéo, '3' audio, '4' contact, '5' géolocalisation, '6' gestion d'un groupe, '7' URL, '8' fichier, ...
ZGROUPMEMBER	Identifiant de l'émetteur dans un groupe, = Z_PK dans ZWAGROUPMEMBER
ZMESSAGEINFO	Identifiant pour suivre le statut d'un envoi à un groupe , = Z_PK dans ZWAMESSAGEINFO
ZMESSAGEDATE	Date d'envoi ou réception
ZFROMJID	Jabber ID de l'émetteur pour une réception, = serveur central pour une discussion de groupe
ZTOJID	Jabber ID du destinataire pour un envoi, = serveur central pour une discussion de groupe
ZTEXT	Texte, émojis
ZMEDIAITEM	Identifiant du multimédia, = Z_PK dans ZWAMEDIAITEM
ZCHATSESSION	Identifiant de la discussion, réutilisé dans ZWAGROUPINFO

Structure et contenu des données - iOS

- Champs pertinents dans *ZWAGROUPINFO*

ZCHATSESSION	Identifiant de la discussion (même valeur que dans ZWAMESSAGE)
ZCREATIONDATE	Date de création
ZCREATORJID	Jabber ID du créateur

- Champs pertinents dans *ZWAGROUPMEMBER*

Z_PK	Clef primaire, = ZGROUPMEMBER dans ZWAMESSAGE
ZCHATSESSION	Identifiant de la discussion de groupe (même clef que dans ZWAMESSAGE)
ZMEMBERJID	Jabber ID du participant (Z_PK, ZMEMBERJID) non unique

- Champs pertinents dans *ZMESSAGEINFO*

Z_PK	Clef primaire = ZMESSAGEINFO dans ZWAMESSAGE
ZRECEIPTINFO	Blob statut réception/lecture pour chaque membre d'un groupe

- Champs pertinents dans *ZMEDIAITEM*

Z_PK	Clef primaire
ZFILESIZE	Pour un multimédia
ZMEDIALOCALPATH	Chemin local vers le fichier <i>group.net.whatsapp.WhatsApp.shared/Media</i>
ZMEDIAURL	URL serveur central
ZXMPPTHUMBPATH	Chemin vers la miniature <i>group.net.whatsapp.WhatsApp.shared/Media</i>
ZVCARDSTRING	Mime type ... parfois

Structure et contenu des données - iOS

- **Groupe** de discussion
 - **Identification** des émetteurs avec *ZWAGROUPMEMBER* et *ZWAGROUPINFO*
 - **Statuts** des messages avec *ZMESSAGEINFO*
 - Opérations de **gestion** dans *ZWAMESSAGE* (ajout/suppression d'un participant, rôle admin)
 - *ZTOJID* : participant impacté
 - *ZTEXT* : Jabber ID de l'admin
- Messages supprimés
 - Opération **délectable** : *Z_PK* manquant
 - Contenu probable dans *group.net.whatsapp.WhatsApp.shared/ChatStorage.sqlite-wal*
 - Accès système de fichiers requis
 - Seulement si l'application est en fonctionnement

Structure et contenu des données

- Android versus iOS : différences
 - Implémentation
 - tables et nommages
 - **valeurs** (statut/type d'un message)
 - un peu plus d'**informations** pour Android (dates de réception/lecture, médias hash)
 - Comportement **groupes de discussion**
 - corrélation de 4 tables pour iOS
 - 1 ligne par participant pour Android
 - Comportement **messages supprimés**
 - BDD temporaire pour iOS ?
 - sauvegardes pour Android
- Android versus iOS : similarités
 - **Contenu** des contacts
 - Jabber ID, unique, à l'installation, `<num_tel>@s.whatsapp.net`
 - Numéro de téléphone (peut changer)
 - Pseudonyme choisi par le contact (peut changer)
 - Alias choisi par l'utilisateur (pas fiable)
 - **Création/modification** des contacts journaux
 - **Redondance** dans les tables
 - Documentation
 - Rien d'**officiel**
 - De nombreux articles ... \pm **exacts/complets** (variabilité avec les versions de l'application ?)

Analyse

- Visionneur SQL
 - Pour un aperçu
 - Extraction **fastidieuse** (corrélation des tables, lisibilité pour un rapport, ...)
 - Automatisation ?
- Python et la bibliothèque *pandas*
 - Structure de données : **DataFrame**

Two-dimensional size-mutable, potentially heterogeneous tabular data structure with labeled axes (rows and columns). Arithmetic operations align on both row and column labels. Can be thought of as a **dict-like container for Series objects**. The primary pandas data structure.
<https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html>

- Idée principale : requêtes simples sur la BDD, puis filtres/agrégation/boucles en python

Analyse

- Extraire les messages sur une période donnée

```
connection = sqlite3.connect('./iphone_acquisition/ChatStorage.sqlite')

ios_start = int(datetime(2001, 1, 1).timestamp())
timeframe_start = datetime(2019, 5, 1).timestamp()
timeframe_end = datetime(2019, 5, 2).timestamp()
start = int(timeframe_start) - ios_start
end = int(timeframe_end) - ios_start

where = "ZMESSAGEDATE >= {} AND ZMESSAGEDATE <= {}".format(start, end)
query = "SELECT * FROM ZWMESSAGE WHERE {}".format(where)
df_messages = pandas.read_sql_query(query, connection)
```

Analyse

- Contenu d'une dataframe

```
print(df_messages.head())
```

	Z_PK ...	ZFROMJID	ZTEXT	ZTOJID
0	41883 ...	XXX@s.whatsapp.net	<blabla1>	None
1	41884 ...	None	<blabla2>	XXX@s.whatsapp.net
2	41885 ...	None	<blabla3>	XXX@s.whatsapp.net
3	41886 ...	XXX@s.whatsapp.net	<blabla4>	None
4	41887 ...	None	<blabla5>	XXX@s.whatsapp.net

[5 rows x 34 columns]

Analyse

- Sélectionner seulement certaines colonnes

```
columns = [  
    'Z_PK',  
    'ZMESSAGEDATE',  
    'ZCHATSESSION',  
    'ZGROUPMEMBER',  
    'ZFROMJID',  
    'ZTOJID',  
    'ZMEDIAITEM',  
    'ZTEXT',  
]  
print(df_messages[columns])
```


Analyse

- Identifier les **messages manquants** ...
 - Total
 - Liste des index

```
last_zpk = df_messages['Z_PK'].iloc[-1]
nb_rows = len(df_messages.index)
print('Missing {} messages'.format(last_zpk - nb_rows))

complete_zpk = list(range(1, last_zpk+1))
missing_zpk = set(complete_zpk).symmetric_difference(set(df_messages['Z_PK']))
print('Missing Z_PK are : {}'.format(','.join(missing_zpk)))
```

Analyse

- ... et les périodes associées

```
previous_zpk = missing_zpk[0] - 1
nb_missing = 1
for i in range(0, len(missing_zpk)):
    next_zpk = missing_zpk[i] + 1
    if next_zpk in missing_zpk:
        nb_missing += 1
    else:
        filter_start = df_messages['Z_PK'] == previous_zpk
        filter_end = df_messages['Z_PK'] == next_zpk
        start = df_messages.loc[filter_start, 'ZMESSAGEDATE']
        end = df_messages.loc[filter_end, 'ZMESSAGEDATE']
        print('missing {} message(s) from {} to {}'.format(nb_missing, start, end))

    nb_missing = 1
    if i < len(missing_zpk):
        previous_zpk = missing_zpk[i+1] - 1
```

Analyse

- **Identifier** les participants d'une discussion de groupe et **associer** leur message

```
df_messages_group123 = df_messages_timeframe[df_messages['ZCHATSESSION'] == 123]

for _, message in df_messages_group123.iterrows():
    if message['ZFROMJID'] == '-1': direction = 'sent by'
    if message['ZTOJID'] == '-1': direction = 'received from'

    query = "SELECT * FROM ZWAGROUPMEMBER WHERE Z_PK={}".format(message['ZGROUPMEMBER'])
    df_member = pandas.read_sql_query(query, connection)
    jabber_id = df_member.at[0, 'ZMEMBERJID']

    print('On {}, message {} {}: {}'.format(
        message['ZMESSAGEDATE'] + ios_start,
        direction,
        jabber_id,
        message['ZTEXT']
    ))
```

Conclusion

- Acquisition **sans modification du smartphone**
 - iOS : conversations et contacts **accessibles**
 - Android : **peut-être**
- Analyse
 - **Identifier** les participants (Jabber ID)
 - Identifier les opérations de **gestion des groupes** de conversation (ajout, suppression, admin rôle)
 - Récupérer les **messages** (textes, images, sons, vidéos) non supprimés
 - Déterminer le **statut** des messages (envoyé, reçu, lu)
 - **Détecter** les messages supprimés (mais contenus probablement perdus)
- Autres artefacts : **journaux** de l'application
 - Avec **root/jailbreak**
 - Succès non garanti (OS, version, modèle, constructeur, chiffrement, ...)
- Outil d'analyse
 - Python + pandas
 - Agrégation et automatisation

Des questions ?

