

Analiza systemu Android pod kątem informatyki śledczej

Krzysztof Uszko

SPIS TREŚCI

Wstęp	2
Interesujące zagadnienia.....	2
Dostępność do plików	3
Rodzaje danych i oprogramowanie	3
Przygotowanie telefonu	5
1. Wiadomości tekstowe	
1.1. SMSy i MMSy	6
1.2. Messenger	8
1.3. WhatsApp	9
2. Kontakty i rejestr połączeń	12
3. Skrzynka pocztowa	13
4. Domyślna wyszukiwarka	14
5. Konto Google	15
5.1 Google Chrome	16
5.2 YouTube	17
5.3 Google Maps	17
6. Notatnik	18
Logi Systemowe	19
Pliki użytkownika	21
Własne narzędzie do analizy	22
Bibliografia	24

Wstęp

System operacyjny dla urządzeń mobilnych (telefony komórkowe, tablety, ale również telewizory czy netbooki). Oparty na jądrze Linuxa. Pierwszą wersję Android SDK opublikowano w 2007 roku, natomiast rok później Google zaprezentowało pierwsze prototypy telefonów opartych o ten właśnie system.

Aktualna wersja to Android 11 (wydana 8 września 2020 r)

Stanowi niespełna 73% rynku mobilnych systemów operacyjnych



(źródło: <https://gs.statcounter.com/os-market-share/mobile/worldwide>)

Procentowy rozkład udziału poszczególnych wersji systemu:



(źródło: <https://gs.statcounter.com/android-version-market-share/mobile-tablet/worldwide>)

Wszystkie przykłady w raporcie będą oparte o wersje 5.1 Lollipop i nowsze (SDK API level 22+)

Interesujące informacje dla informatyki śledczej:

1. Wiadomości tekstowe - nie tylko SMSy, ale również konwersacje przeprowadzone za pomocą popularnych aplikacji tj. Messenger czy WhatsApp
2. Rejestry połączeń
3. Zdjęcia, dokumenty, pliki dźwiękowe
4. Połączenia bluetooth, wi-fi - połączone sieci wi-fi, zapamiętane urządzenia bluetooth
5. Rejestry GPS - odczytywanie logów lokalizacji w określonym czasie
6. Historia wyszukiwania domyślnej przeglądarki
7. Konto google i aplikacje z niego korzystające
8. Logi systemowe Androida

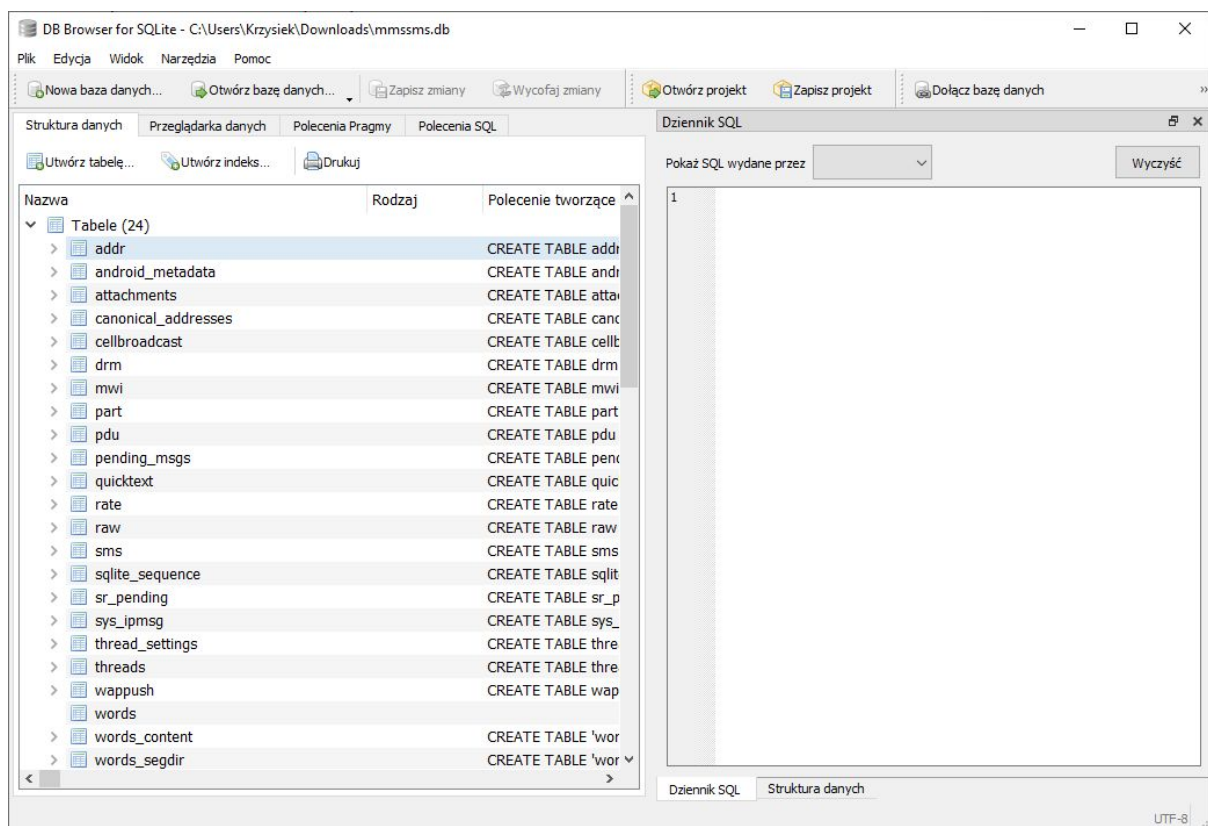
Dostępność do plików:

W systemie Android istnieją dwie opcje dostęp do plików:

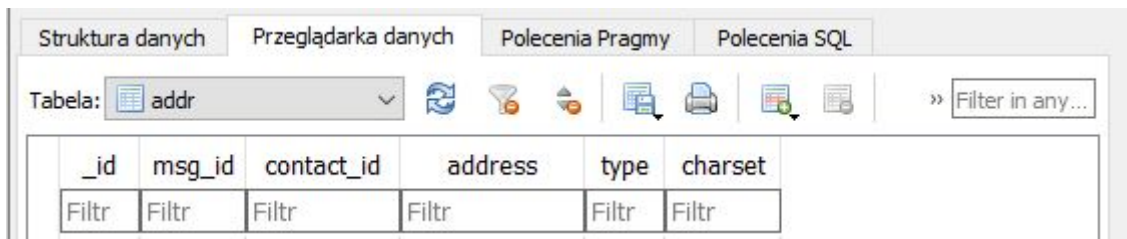
1. Dostęp bez uprzywilejowania: dostęp do danych możliwy jest jedynie za pośrednictwem aplikacji systemowych -> np. za pomocą aplikacji "Galeria" mamy dostęp do zdjęć, a "Wiadomości" wyświetlają nam rozmowy SMS/MMS
2. Dostęp root -> mamy pełen dostęp do plików -> jak w systemach unixowych używając superuser'a. Taka funkcjonalność pozwala nam na bezpośrednie działanie na plikach z bazami danych np.
/data/data/com.android.providers.telephony/databases , czy też wyświetlanie ukrytych plików logów.

Rodzaje danych i użyte oprogramowanie:

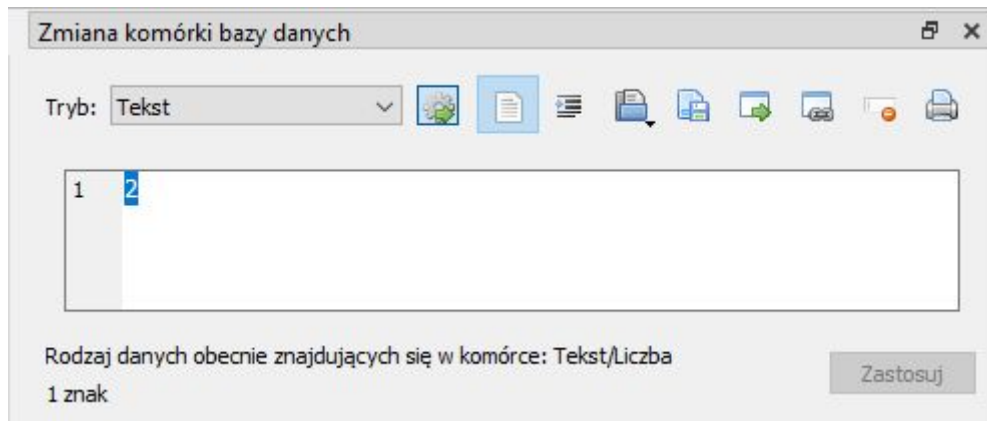
Większość danych aplikacji w systemie Android zapisywana jest w formie baz danych SQLite. SQLite to biblioteka obsługująca język zapytań SQL. System jest szybki oraz praktyczny, a do tego posiada API w wielu językach programowania. Do przeglądania tego typu baz danych będę używać programu [DB Browser](#).



Oprogramowanie to jest bardzo intuicyjne. Składa się z: paska narzędzi, paska projektu oraz obszaru roboczego - w którym po zaimportowaniu bazy znajdziemy m.in jej strukturę czy też widok danych w formie tabeli. Każdą analizę warto zacząć od przejrzania "Struktury danych", w celu zaznajomienia się z formatem zapisu bazy, oraz polami w poszczególnych tabelach.



W polu “Przeglądarka danych” mamy dostępną listę tabel -> po wyborze ukaze nam się tabela wraz z danymi. Mamy tutaj również dostępną opcję filtrowania (np. w celu wyszukania odpowiednich fraz w danej kolumnie).



W prawej części sekcji roboczej możemy wybrać inne interesujące funkcje tj: Dziennik SQL, wykresy, Zmiana komórki bazy danych lub zdalne DB. W moim przypadku będę używać głównie funkcji ‘Zmiany komórki bazy danych’.

Przygotowanie telefonu:

W celu wykonania pełnej analizy systemu zrootowałem testowany telefon. Do tego celu wykorzystałem oprogramowanie [KingoRoot](#), które jest jedną z bezpieczniejszych metod na uzyskanie dostępu root. Innym sposobem na uzyskanie większych przywilejów jest odblokowanie bootloadera i "zflashowanie" odpowiedniego oprogramowania. Więcej informacji można np. znaleźć [tutaj](#).

Sam KingoRoot instaluje pakiet, za pośrednictwem którego gwarantowany jest dostęp root. Do przeglądania i kopiowania plików użyję [ADB](#) - Android Debug Bridge. Jest to narzędzie do debugowania w systemie Android, które umożliwia komunikację z poziomem konsoli systemowej CMD.

```
C:\Windows\system32>adb devices
List of devices attached
DML7P7OR999999999          device

C:\Windows\system32>adb root

C:\Windows\system32>adb shell
shell@PIXI3-5:/ $ su
root@PIXI3-5:/ #
```

1. Wiadomości tekstowe

1.1 SMSy i MMSy

Większość danych w systemie Android znajdziemy w folderze /data/data. Tutaj każdy pakiet ma swój osobny katalog. Domyślne dane dla SMSów i MMSów znajdziemy w folderze com.android.providers.telephony lub od 7 wersji Androida odpowiednio w com.android.providers.telephony.

```
root@PIXI3-5:/ # cd data/data
root@PIXI3-5:/data/data # cd com.android.providers.telephony
root@PIXI3-5:/data/data/com.android.providers.telephony # ls
app_parts
databases
lib
shared_prefs
root@PIXI3-5:/data/data/com.android.providers.telephony # cd databases
root@PIXI3-5:/data/data/com.android.providers.telephony/databases # ls
HbpcdLookup.db
HbpcdLookup.db-journal
cb.db
cb.db-journal
mmssms.db
mmssms.db-journal
telephony.db
telephony.db-journal
```

W celu dalszego zarządzania bazą danych smsów i mmsów kopiujemy plik na kartę SD.

```
p mmssms.db /sdcard
```

Interesujący nas plik to baza danych SQLite. Do przeglądania jej treści wykorzystam program [DB Browser](#). Po przeglądnięciu struktury bazy mmssms.db ciekawe wydają się być tabele 'sms' oraz 'threads'.

Tabela 'sms' zawiera wszystkie wiadomości SMS/MMS w formie pojedynczego wiersza.

Warte uwagi pola to: address - numer telefonu nadawcy lub odbiorcy, date lub date_sent (w zależności czy dany telefon był odbiorcą czy nadawcą), read (status odczytania wiadomości), body (treść wiadomości) oraz creator (jaki pakiet wywołał operację tworzenia lub odbierania wiadomości - np. niektóre aplikacje z przydzielonym dostępem mogą je wysyłać)

Tabela: sms																
_id	thread_id	address	m_size	person	date	date_sent	protocol	read	status	type	reply_path_present	subject	body	service_center	locked	
Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	

Przykładowe wiadomości:

195	204	121 T-Mobile	225	0	1588239264317	1588233751000	0	0	-1	1	0	NULL	5 GB Internetu na Majówkę od T-Mobile! Zainstal...	+48602006031	0	
196	205	126 InPost	126	0	1588319017191	1588319016000	0	1	-1	1	0	NULL	Paczka czeka w Paczkomacie NSA10N Tarnowsk...	+48602006031	0	

'Threads' to tabela zawierająca wątki rozmów - nic innego jak pogrupowane wiadomości z tabeli sms, według takiej samej relacji nadawca-odbiorca. Ta tabela pozwala w łatwy sposób określić ile wiadomości zostało wysłanych między tymi numerami (kolumny: message_count i read_count). Pole snippet - to treść ostatniej wiadomości danego wątku.

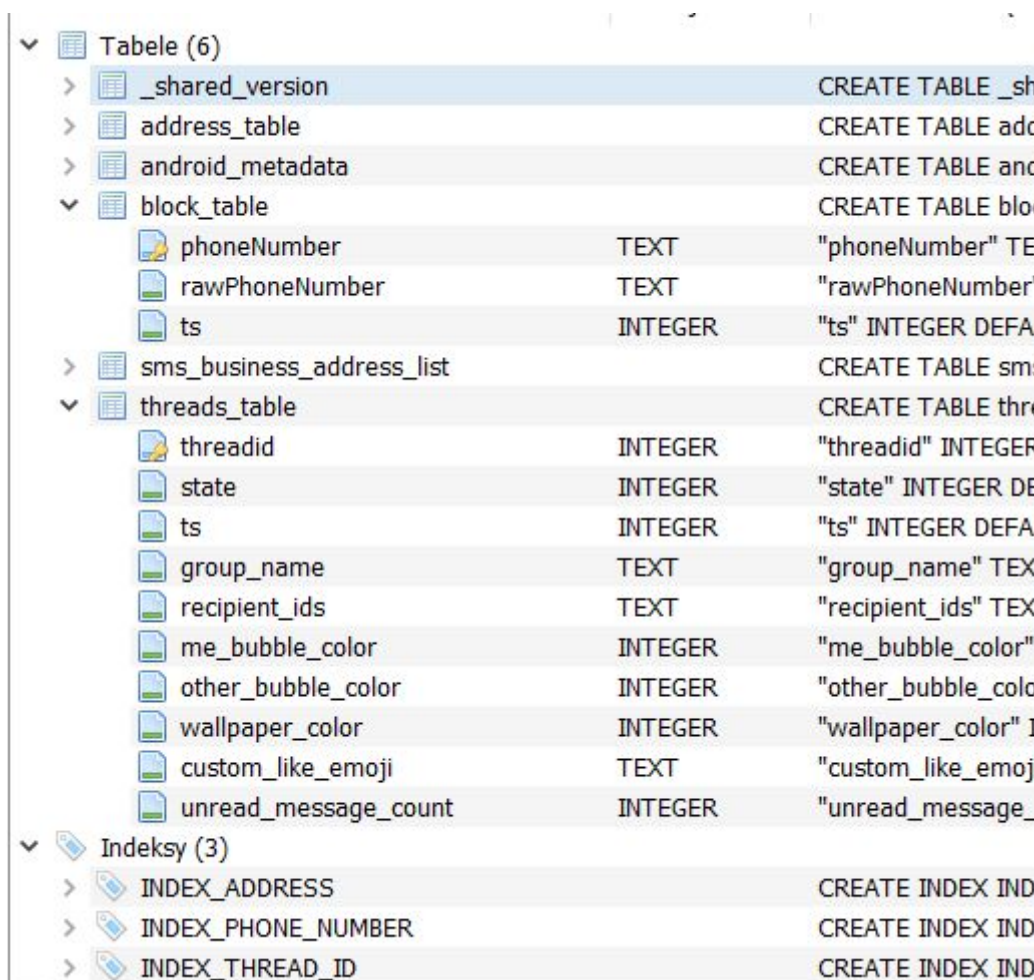
Odpowiednie pliki dotyczące wiadomości SMS możemy znaleźć również w plikach innych aplikacji. Jedną z popularniejszych obsługujących wysyłanie i odbieranie wiadomości jest Facebook Messenger. Nie należy jednak mylić tej funkcjonalności z czatem internetowym oferowanym przez tą aplikację (o artefaktach z czatu w dalszej części projektu). Pliki baz danych aplikacji znajdziemy w pakiecie [com.facebook.orca/databases](https://www.facebook.com/orca/databases)

```
root@PIXI3-5:/data/data/com.facebook.orca/databases #
```

```
p smstakeover_db /sdcard
```

<- plik wiadomości sms

Jako że na testowanym telefonie aplikacja Messenger nie była domyślną aplikacją do obsługi wiadomości SMS, to baza danych będzie pusta. Warto jednak przyjrzeć się jej strukturze:



Tabele (6)			
>	_shared_version		CREATE TABLE _sh
>	address_table		CREATE TABLE add
>	android_metadata		CREATE TABLE and
▼	block_table		CREATE TABLE bloc
	phoneNumber	TEXT	"phoneNumber" TE
	rawPhoneNumber	TEXT	"rawPhoneNumber"
	ts	INTEGER	"ts" INTEGER DEFAI
>	sms_business_address_list		CREATE TABLE sms
▼	threads_table		CREATE TABLE thre
	threadid	INTEGER	"threadid" INTEGER
	state	INTEGER	"state" INTEGER DE
	ts	INTEGER	"ts" INTEGER DEFAI
	group_name	TEXT	"group_name" TEX
	recipient_ids	TEXT	"recipient_ids" TEX
	me_bubble_color	INTEGER	"me_bubble_color"
	other_bubble_color	INTEGER	"other_bubble_colo
	wallpaper_color	INTEGER	"wallpaper_color" I
	custom_like_emoji	TEXT	"custom_like_emoji
	unread_message_count	INTEGER	"unread_message_
Indeksy (3)			
>	INDEX_ADDRESS		CREATE INDEX IND
>	INDEX_PHONE_NUMBER		CREATE INDEX IND
>	INDEX_THREAD_ID		CREATE INDEX IND

1.2 Messenger

Messenger to darmowy komunikator - oprócz wysyłania wiadomości tekstowych, umożliwia dzielenie się zdjęciami, nagraniami wideo, czy nagraniami głosowymi. W 2017 roku było 1.82 mld użytkowników internetowych komunikatorów internetowych, gdzie Messenger jest obecnie najpopularniejszym z nich.

```
130|root@PIX13-5:/data/data/com.facebook.orca/databases #  
p threads_db2 /sdcard
```

Plik składa się z 25 tabel i 28 indeksów. Interesujące tabele to: messages (zawierają każdą wiadomość jako 1 wpis), group_conversations - informacje o konwersacjach grupowych, threads - pogrupowane rozmowy wg. uczestników, thread_users - dane uczestników czatu,

▼ Tabele (25)

> _shared_version	> montage_message_reactions
> android_metadata	> properties
> fb_event_members	> ranked_threads
> fb_events	> sqlite_sequence
> folder_counts	> sqliteproc_metadata
> folders	> sqliteproc_schema
> group_conversations	> thread_participants
> message_reactions	> thread_themes
> messages	> thread_users
> montage_directs	> threads
> montage_message_interactive_overlays	> threads_metadata
> montage_message_poll	> virtual_folders
> montage_message_poll_options	

Przykładowe wiadomości odczytane z tabeli messages:

542	1717	mid.\$cAABa9BscbzZ8ZAJOr12MsOBcAOY-	ONE_TO_ONE:...	informatyka śledcza	{"user_key":"FACEBOOK:...
543	1718	mid.\$cAABa9BscbzZ8ZAJnk12MsOaUp34r	ONE_TO_ONE:...	test	{"user_key":"FACEBOOK:...
544	1720	mid.\$cAABa9BscbzZ8ZAJ4b12MsOrLq9Lf	ONE_TO_ONE:...	test1	{"user_key":"FACEBOOK:...
545	1721	mid.\$cAABa9BscbzZ8ZAJumV2MsOhWUgDT	ONE_TO_ONE:...	NULL	{"user_key":"FACEBOOK:...

Wartym uwagi jest to, że cofnięcie wysyłania wiadomości zmienia treść wiadomości na NULL, natomiast ukrycie wiadomości u danego użytkownika już nie kasuje treści.

W przypadku wysyłania innej treści niż tekstowa, w kolumnie messages.attachments dołączone są obiekty JSON reprezentujące dane załączniki.

```
[{"id":"461435295251525","fbid":"461435295251525",
"mime_type":"image/jpeg","filename":
"image-461435295251525","file_size":92139,
"image_data_width":395,"image_data_height":280,
"urls":["\\\"MEDIUM_PREVIEW\\\"":\\"\\\"width\\\"":395,
\\\"height\\\"":280,\\\"src\\\"":\\"\\\"
https://scontent.xx.fbcdn.net/v/w1/t1.15752-9/1297222
58_3103135469792655_5936287423566681367_n.png.webp?_n
c_cat=110&ccb=2&nc_sid=ae9488&nc_ohc=lafhydokvugAX9
5m2k5&nc_ad=z-m&nc_cid=0&nc_ht=scontent.xx&nc_tp=
30&nc_rmd=260&oh=5253c29f68916db57a8232ee75ffa2cd&oe
```

Dane z kolumny threads.senders dla wybranego wątku:

```
[{"user_key":"FACEBOOK:100004473368374","name":
"Krzysztof Uszko","email":null,"phone":null,
"smsParticipantFbid":null,"is_commerce":false,
"messagingActorType":"FACEBOOK",
"graphQLWorkForeignEntityDetail":null}]
```

W tym przypadku byłem zarówno nadawcą jak i odbiorcą wiadomości, więc istnieje tylko jeden wpis użytkownika -> w normalnej lub grupowej konwersacji umieszczane są dane wszystkich uczestników.

1.3 WhatsApp

W założeniu identyczna do omówionego wcześniej Facebook Messengera, również jest częścią Facebook Inc. Początek sukcesu zawdzięcza dużej ilości użytkowników w Chinach. Prócz czatu główną funkcjonalnością są rozmowy audio - video.

Interesujące pliki:

lokalizacja /data/data/com.whatsapp/databases/:

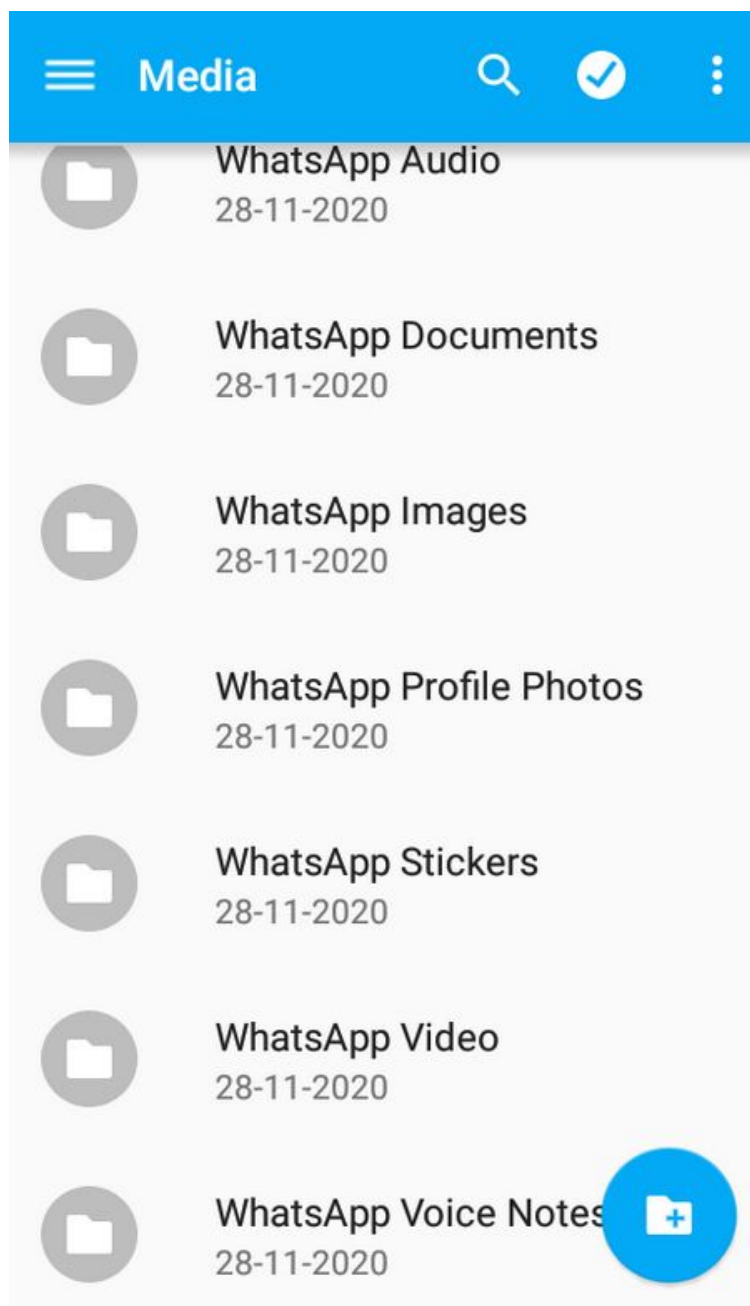
1. wa.db - baza danych zapisanych kontaktów
2. msgstore.db - baza danych wysyłanych wiadomości

lokalizacja WhatsApp/Databases (domyślnie folder jest tworzony na karcie SD jeżeli przy instalacji telefon taką posiada)

1. msgstore.db.crypt12 - aktualny backup wiadomości
2. msgstore-{data}.db.crypt12 - backup wiadomości z danej daty

Pliki te są szyfrowane, klucz można znaleźć w lokalizacji: /data/data/com.whatsapp/files/key

lokalizacja WhatsApp/Media (również na karcie SD jeśli telefon taką posiada, w przeciwnym przypadku /data/media/WhatsApp):



Z pliku wa.db najbardziej interesująca jest tabela wa_contacts - zawierająca informacje o kontaktach - numery / nazwa własna i wyświetlania / status

	_id	jid	is_whatsapp_user	status	status_timestamp	number	raw_contact_id
	ltr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	48[REDACTED]84@s.whatsapp.net	0	NULL	0	[REDACTED]84	624
2	2	48[REDACTED]66@s.whatsapp.net	0	NULL	0	[REDACTED]66	657
3	3	48[REDACTED]721@s.whatsapp.net	1	Hey there! I am using WhatsApp.	1603386728000	[REDACTED]721	658

W pliku msgstore.db podobnie jak we wcześniejszych przypadkach interesować nas będzie tabela 'messages' - gdzie każda wiadomość to osobny wpis w tabeli. W bazie WhatsApp'a odpowiednikiem tabeli 'threads' jest tabela 'chat'

4	4	48[REDACTED]770@s.whatsapp.net	0	4C063245409D84A855EF4A2E8773BB1F	0	0	.	1607168368000
---	---	--------------------------------	---	----------------------------------	---	---	---	---------------

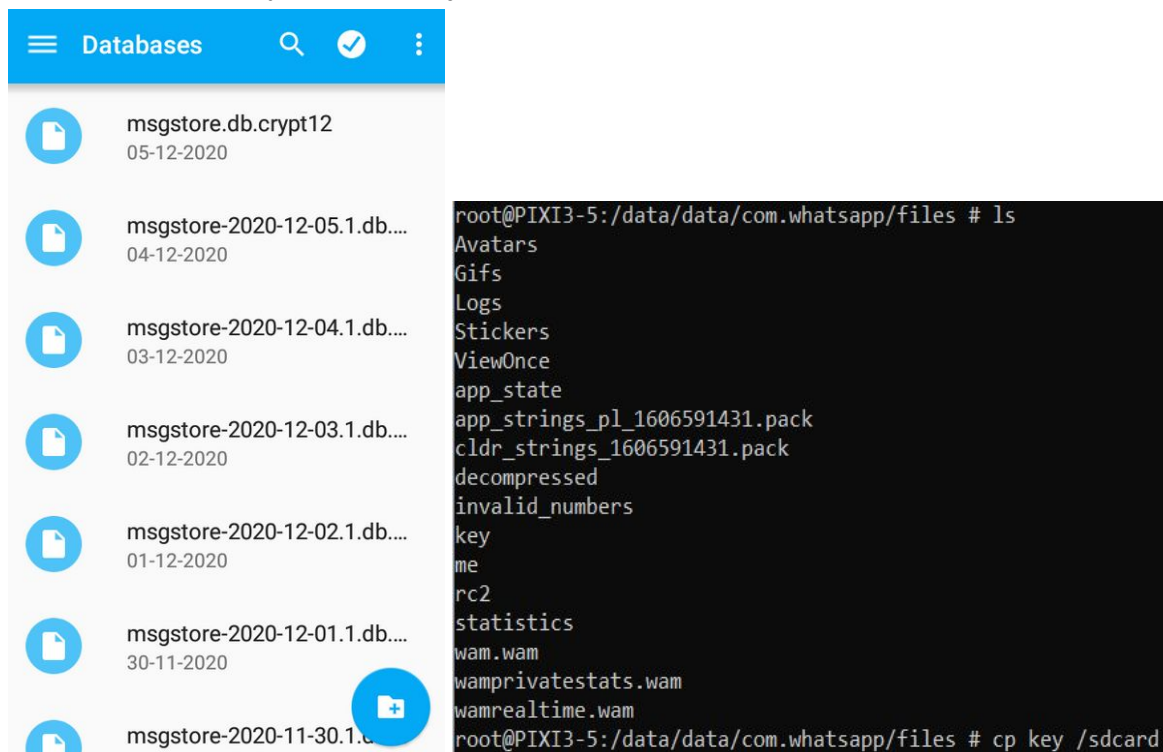
Przykładowy wpis zawiera adres nadawcy/odbiorcy, id wiadomości, jej treść, timestamp. Co ciekawe wiadomości mogą mieć również dołączoną lokalizację.

latitude	longitude
Filtr	Filtr
0.0	0.0

Jednak w przypadku moich danych ani nadawca ani odbiorca nie korzystali z takiej funkcji.

Jak odzyskać skasowane wiadomości?

W wyżej opisanym folderze znajdującym się domyślnie na karcie SD znajdujemy plik back-up'a z dnia który nas interesuje.



The image shows a mobile application interface with a list of database files under the heading 'Databases'. The files are listed with their names and dates:

- msgstore.db.crypt12 (05-12-2020)
- msgstore-2020-12-05.1.db.... (04-12-2020)
- msgstore-2020-12-04.1.db.... (03-12-2020)
- msgstore-2020-12-03.1.db.... (02-12-2020)
- msgstore-2020-12-02.1.db.... (01-12-2020)
- msgstore-2020-12-01.1.db.... (30-11-2020)
- msgstore-2020-11-30.1.db.... (29-11-2020)

To the right, a terminal window shows the output of a command to list files in a directory:

```
root@PIXI3-5:/data/data/com.whatsapp/files # ls
Avatars
Gifs
Logs
Stickers
ViewOnce
app_state
app_strings_pl_1606591431.pack
cldr_strings_1606591431.pack
decompressed
invalid_numbers
key
me
rc2
statistics
wam.wam
wamprivatstats.wam
wamrealtime.wam
root@PIXI3-5:/data/data/com.whatsapp/files # cp key /sdcard
```

Następnie znajdujemy plik key, który jak nazwa wskazuje jest kluczem do odszyfrowania pliku db.crypt12. W celu odszyfrowania danych można skorzystać z [crypt12 Decryptora](#).

```
C:\Users\Krzysiek\Downloads>java -jar decrypt12.jar key msgstore.db.crypt12 decrypted_msgstore.db
```

Zdeszyfrowany plik to nic innego jak plik msgstore.db omawiany wcześniej - jednak jego zapis dotyczy danego dnia.

2. Kontakty i rejestr połączeń

Informacje znajdują się w tej samej bazie danych, umieszczonej w lokalizacji:

```
root@PIXI3-5:/data/data/com.android.providers.contacts/databases #
```

Głównym plikiem jest contacts2.db jednak, w tej samej lokalizacji znajdziemy inne powiązane nazwy - są to zaszyfrowane stany bazy danych z różnych dni.

```
contacts2.db
contacts2.db-journal
contacts2.db-mj0416079B6
contacts2.db-mj05410D94B
```

Dane dotyczące kontaktów znajdziemy w tabeli 'data' pliku contacts2.db. Każdemu kontaktowi odpowiadają 2 wpisy -> pierwszy z mimetype_id = 5 to wpis z numerem telefonu - potrzebny dla pakietów działających na kontaktach (wysyłanie SMS-ów, czy rozmowa przychodząca), natomiast wpis z mimetype_id = 7 odpowiada nazwom zdefiniowanym przez użytkownika - tak wyświetla się kontakt np. w aplikacji 'Kontakty'.

Tabela: data

	_id	mimetype_id	raw_contact_id	data1	data2
	Filtr	Filtr	Filtr	Filtr	Filtr
1	65	5	33	[REDACTED]61	2
2	66	7	33	[REDACTED]	[REDACTED]

Za dane rejestru połączeń odpowiada tabela 'calls'.

Tabela: calls

	_id	number	presentation	date	duration	data_usage	type	features	subscription_component_name
	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
1	860	[REDACTED]14	1	1611582156403	15	NULL	2	0	com.android.phone/...
2	861	+48 [REDACTED]14	1	1611582196565	5	NULL	1	0	com.android.phone/...

Interesująca jest również tabela 'groups' gdzie znajdziemy zapamiętane przez Androida grupy kontaktów.

Tabela: groups

	_id	account_id	sourceid	title	notes	system_id
	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	2	6	My Contacts	System Group: My Contacts	Contacts
2	2	2	7bf687e2892d1116	Starred in Android	Starred in Android	NULL
3	3	2	d	Friends	System Group: Friends	Friends
4	4	2	e	Family	System Group: Family	Family
5	5	2	f	Coworkers	System Group: Coworkers	Coworkers

Warto również zapoznać się ze zdefiniowanymi kontami. Na tej podstawie można np. wywnioskować, że niektóre wiadomości były wysyłane z innej karty SIM.

Tabela: accounts

	_id	account_name	account_type
	Filtr	Filtr	Filtr
1	1	USIM1	USIM Account
2	2@gmail.com	com.google
3	3	Phone	Local Phone Account
4	5	USIM2	USIM Account
5	7	USIM3	USIM Account
6	9	USIM4	USIM Account
7	11	WhatsApp	com.whatsapp

3.Skrzynka pocztowa

W każdej wersji Androida można skonfigurować własnego agenta E-mail - jest to aplikacja dzięki której możemy odbierać i wysyłać maile. Jest to bardzo praktyczne rozwiązanie, ponieważ pozwala na połączenie wielu kont e-mail, przy czym możemy dostawać powiadomienia o przychodzących wiadomościach dla odpowiedniej skrzynki. Domyślnym pakietem Androida dla tego rozwiązania jest com.tct.email.

```
root@PIXI3-5:/data/data/com.tct.email/databases # ls
EmailProvider.db
EmailProvider.db-journal
EmailProviderBody.db
```

Plikiem odpowiadającym za całego agenta e-mail jest EmailProvider.db

Uwaga ! -> Gmail używa identycznego pliku jednak znajdującego się w innej lokalizacji:

```
root@PIXI3-5:/data/data/com.google.android.gm/databases # ls
EmailProvider.db
```

W tabeli 'account' znajdziemy powiązane konta.

Tabela: Account

	_id	displayName	emailAddress	syncInterval	flags	isDefault	senderName	ringtoneUri
	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	krzysztofuszko@interia.pl	krzysztofuszko@interia.pl	15	8	0	krzysztofuszko	content://settings/system/notification_sound

W tabeli 'Message' znajdziemy wszystkie wiadomości e-mail -> każda wiadomość to oddzielny wpis.

Tabela: Message

	_id	syncServerId	displayName	timeStamp	subject
	Filtr	Filtr	Filtr	Filtr	Filtr
1	1	7444	Plush ABO (Polkomtel) - dostarczone przez Interię	1607191160000	Plush Abonament - Królem Internetu - sprawdź, i...
2	2	7443	Interia	1607189720000	Głosujesz i wygrywasz 20 000 złotych!>

W celu wyświetlenia wiadomości wysłanych możemy zastosować proste polecenie SQL:

SQL 1							
1 SELECT * FROM Message WHERE fromList LIKE 'krzysztofuszeko <krzysztofuszeko@interia.pl>';							
	_id	syncServerId	syncServerTimeStamp	displayName	timeStamp	subject	flagRead
1	21		0	<[REDACTED]2@o2.pl>	1607191245847	Informatyka śledcza	1

4. Domyślna wyszukiwarka

W nowszych wersjach systemu Android domyślną wyszukiwarką jest Google Chrome (o niej w punkcie nr.5). Jednak przy starszych wersjach systemu warto sprawdzić następującą lokalizację:

```
root@PIXI3-5:/data/data/com.android.browser/databases #
```

Znajdziemy tutaj 2 pliki: websites.db -> plik z zakładkami,
browser2.db -> ogólna baza danych przeglądarki

W tabeli 'websites' pliku websites.db znajdziemy wpisy dotyczące zapisanych zakładek dla przeglądarki.

Tabela: websites		
_id	url	title
Filtr	Filtr	Filtr
1	9	
2	8 http://www.facebook.com	Facebook
3	7 http://www.nasza-klasa.pl	Nasza Klasa
4	6 http://www.allegro.pl	Allegro
5	5 http://www.onet.pl	Onet

W bazie browser2.db interesującymi tabelami są 'history' oraz 'searches'

Tabela: history						
_id	title	url	created	date	visits	
Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	
1	52 m.t-mobile.pl	http://m.t-mobile.pl/	0	1507389814394	2	

/cache - dodatkowo w tym katalogu znajdziemy pliki pamięci cache związane z wyszukiwaniem. Używając komendy ls -l możemy wybrać jeden z nich według daty

powstania

```
-rw----- u0_a132 u0_a132      97011 2021-01-25 15:34 6f3f49cd288da9d6_0
-rw----- u0_a132 u0_a132       148 2021-01-25 15:34 6f3f49cd288da9d6_1
-rw----- u0_a132 u0_a132     27303 2021-01-25 15:34 6f76c74b6eb12240_0
-rw----- u0_a132 u0_a132     8483 2021-01-25 15:34 6fdcaeb12aac1fce_0
-rw----- u0_a132 u0_a132       129 2021-01-25 15:34 6fdcaeb12aac1fce_1
```

Przykładowa treść pliku cache:

```
0\r0000 < 000 https://www.kingoapp.com/static/images/onionbrowser-icon.png@PNG
IHDR 1 0 0fw0 0tEXtSoftware Adobe ImageReady0e< 0&iTtXML:com.adobe.xmp <?xpacket begin=" " id="W5M0MpCehi
HzreSzNTczkc9d"?> <x:xmpmeta xmlns:x="adobe:ns:meta/" x:xmptk="Adobe XMP Core 5.6-c138 79.159824, 2016/09/14-01:09:01
"> <rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"> <rdf:Description rdf:about="" xmlns:xmp="http://
```

5.Konto Google i powiązane aplikacje

Informacje o kontach znajdują się odpowiednio w:

- /data/system/users/0/accounts.db - dla wersji Marshmallow (6.0 i wcześniejszej)
- /data/system_ce/0/accounts_ce.db - dla wersji Nougat (7.1 i późniejszej)

Tabela: accounts				
	_id	name	type	password
	Filtr	Filtr	Filtr	Filtr
1	1	[REDACTED]@gmail.com	com.google	oauth2rt_1/...
2	11	WhatsApp	com.whatsapp	NULL
3	12	Messenger	com.facebook.messenger	NULL
4	13	krzysztofuszko@interia.pl	com.tct.email	[REDACTED]

Ciekawostka - hasło do poczty jest przechowywane w plain-text'cie

Prócz samych danych do kont interesujące są również tokeny uwierzytelniające dla aplikacji - działają one podobnie jak klucze sesji - za ich pomocą 'przedstawia' się aplikacji w celu uzyskania dostępu do danych zasobów.

Tabela: authtokens				
	_id	accounts_id	type	authtoken
	Filtr	Filtr	Filtr	Filtr
1	1	1	SID	DQAAAAkBAADZd4am-6sf2ge_ZQMF5t0MfXgN-...
2	2	1	LSID	DQAAAAsBAADK3ps8LQ_pSG_kO5GfA_watUZuDU...
3	3	1	com.google.android.gms:...	DQAAAAkBAADZd4am-6sf2ge_ZQMF5t0MfXgN-...
4	4	1	com.google.android.gms:...	ya29.agKKFcMOPyI1h7eLnHZVYVhsn2EEFNiO0h4...
5	7	1	com.google.android.music:...	DQAAAAwBAAClJsXCDou3t2WOxJfI8vz7Y6eE-...
6	9	1	com.google.android.gms:...	DQAAAA0BAAClJsXCDou3t2WOxJfI8vz7Y6eE-...
7	32	1	com.google.android.gsf:...	DQAAAAsBAADPLZO4GqLWcpTLBH2dqH2cQGfVK...
8	154	1	com.android.chrome:...	ya29.cAJCKwln6wK9XzFH6Bt_c8ozJ0VmxBIJLqvo...

5.1 Google Chrome

Interesujące ścieżki:

/data/data/com.android.chrome/:

- /cache - podobnie jak z wcześniej opisaną domyślną aplikacją - katalog z plikami pamięci cache
- /databases - w zależności od wersji API Androida mogą zostać tu umieszczone bazy danych, jednak nie jest to konieczne - w innym przypadku dane są przechowywane w bazach zlokalizowanych w /app_chrome/Default
- /app_chrome/Default - katalog, gdzie każdy plik to osobna baza danych odpowiedzialna za coś innego (łatwo domyślić się po nazwie czego dotyczą).

Zawierają identyczne informacje co plik browsers2.db w starszych Androidach, jednak dane rozdzielone są na bazy, a nie jak w tamtym przypadku - tabele.

```
generic_x86:/data/data/com.android.chrome/app_chrome/Default # ls
000009.log                LOG.old                  Top\ Sites
AutofillStrikeDatabase    Local\ Storage           Top\ Sites-journal
BudgetDatabase            Login\ Data              Translate\ Ranker\ Model
CURRENT                   Login\ Data-journal      TransportSecurity
Cookies                   MANIFEST-000008         UsageReportsBuffer
Cookies-journal           NTPSnippets             UsageStats
DeltaFileLevelDb          Network\ Action\ Predictor Visited\ Links
Download\ Service         Network\ Action\ Predictor-journal Web\ Data
Favicons                  Network\ Persistent\ State Web\ Data-journal
Favicons-journal          Offline\ Pages           blob_storage
Feature\ Engagement\ Tracker Preferences              data_reduction_proxy_leveldb
GPUCache                  README                  page_load_capping_opt_out.db
History                   Search\ Logos            page_load_capping_opt_out.db-journal
History\ Provider\ Cache  Session\ Storage        previews_hint_cache_store
History-journal           Shortcuts                previews_opt_out.db
LOCK                      Shortcuts-journal        previews_opt_out.db-journal
LOG                       Sync\ Data                shared_proto_db
```

Szczegółowa analiza każdego z plików nie ma sensu - jak widać jest ich sporo, jednak każdy z nich jest zapisany w formacie SQLite, i przyjmuje podobną strukturę do wcześniej opisywanych plików.

5.2 YouTube

```
root@PIXI3-5:/data/data/com.google.android.youtube #
```

/databases/identity.db - baza danych zawierająca informacje o użytkownikach korzystających z tej bazy danych - zawiera trzy tabele: android_metadata, profile, identity - jednak tylko ta ostatnia jest interesująca ze względu na informatykę śledczą.

Tabela: identity				
id	account	page_id	is_persona	datasync_id
Filtr	Filtr	Filtr	Filtr	Filtr
1	0Gt5FN9v5sGsz3AbO4GqgA [REDACTED]@gmail.com		0	104595321849683237406

/files/ondevicesuggest - folder zawierający pliki .bin zawierający "tagi" - sugestie polecenia tematyki filmików



fragment pliku z tagami - kodowanie jest bliżej nieokreślone - UTF-8 udaje się odczytać kawałek danych, jednak spora część dalej pozostaje nieczytelna - da się jednak domyślić z kontekstu czego mogą one dotyczyć

5.3 Google Maps

```
root@PIXI3-5:/data/data/com.google.android.apps.maps #
```

/databases/gmm_myplaces.db /databases/gmm_sync.db

Tabela: sync_item_data							
corpus	client_id	server_id	timestamp	feature_fprint	latitude_e6	longitude_e6	r
Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Filtr	Fi
1	6 User Parameters	User Parameters	0	NULL	0	0	
2	3 list:104595321849683237406:1000	7gtgnoRtJoWw_orreUxrsRI3ufa0fg	0	NULL	0	0	
3	4 AUTO_GEN_KEY_FOR_SYNC_0	list:6d796d617073676d6d	0	NULL	0	0	
4	5 AUTO_GEN_KEY_FOR_SYNC_0	253E3C8427FD3ECC	0	-7952100205609268100	50074877	19997738	
5	3 list:104595321849683237406:1001	NULL	0	NULL	0	0	
6	4 AUTO_GEN_KEY_FOR_SYNC_1	list:6d796d617073676d6d	0	NULL	0	0	
7	5 AUTO_GEN_KEY_FOR_SYNC_1	253E3D09F98C5688	0	-5780170588995463161	50060962	19934107	

bazy danych z miejscami zdefiniowanymi przez użytkownika

/cache/odelay_cache.cs

```
Flisaków
Nowy Sącz*
Flisaków, Nowy Sącz y-G+=""
R U* FlisakówbBChIJeY-0lq76PUcRK-KsGK76GBA=""E
Jana Kilińskiego
Nowy Sącz*
Jana Kilińskiego, Nowy Sącz G-G2(=""
```

pamięć cache historii wyszukiwania, rozszerzenie .cs wskazuje na kod źródłowy kodu C#, jednak formatowanie jest znacznie inne.

Jedyne co udało mi się ustalić odnośnie historii wyświetlanych / polubionych filmów, odwiedzonych miejsc i tras (np. z google maps timeline), to że są one przechowywane na serwerze w przydzielonej pamięci dla danego konta google. Nie istnieją więc bazy danych z nimi związane (być może część z tych akcji jest logowana w plikach typu .cache, jednak do tej pory nie znalazłem takiej informacji w dokumentacjach oraz ciężko znaleźć informacje nt. sposobu formatowania takich plików)

6. Notatnik

```
note.db journal
root@PIXI3-5:/data/data/com.tct.note/databases # cp note.db /sdcard
```

W bazie danych domyślnego notatnika interesującym jest głównie tabela 'note_text'. To w niej znajdziemy notatki zapisane w formie tekstu jawnego.

Tabela: note_text					Filter in any column	
	_id	note_id	content		text	
	Filtr	Filtr	Filtr		Filtr	
1	2	2	Poniedziałek 18:00 jazdy		Poniedziałek 18:00 jazdy	
2	3	3	20 20 20 20 10 10 10 ...		20 20 20 20 10 10 10 ...	
3	4	4	K510a 703		K510a 703	

Logi systemowe

logcat - narzędzie do wyświetlania logów, zawiera bardzo dużo informacji,

```
root@PIXI3-5:/ # logcat -t '01-25 16:10:0.0' -b events
```

przykładowe polecenie wyświetlające logi typu I (Info) najnowsze od wskazanej daty

```
I/notification_visibility_changed( 696): [0|kingoroot.supersu|16|null|10092;0|com.android.vending|478177066|null|10046,0|com.facebook.orca|10000|ONE_TO_ONE:1780254269:100004473368374|10084;0|com.whatsapp|20|null|10085]
I/notification_cancel( 696): [10054,12207,com.tct.email,-931835202,NULL,0,0,64,8,NULL]
I/notification_cancel( 696): [10054,12207,com.tct.email,-931835202,NULL,0,0,64,8,NULL]
I/c2dm ( 1464): [8,com.facebook.orca:0:1611591326793630,31,19]
I/notification_enqueue( 696): [10084,14118,com.facebook.orca,10000,ONE_TO_ONE:1780254269:100004473368374,0,Notification(pri=1 contentView=null vibrate=[0] sound=null defaults=0x0 flags=0x1 color=0xff0a7cff category=msg actions=2 vis=PRIVATE),1]
I/notification_visibility_changed( 696): [0|com.facebook.orca|10000|ONE_TO_ONE:1780254269:100004473368374|10084,0|com.android.vending|478177066|null|10046]
```

cześć wykonanego wyżej polecenia, zawierająca informacje o oczekujących powiadomieniach z poczty,facebooka, i aplikacji kingoroot

/data/tombstones - lokalizacja zawiera pliki 'tombstone' - pliki z informacjami dotyczącymi procesów, które zakończyły się 'przez przypadek' - odpowiadają Linuxowym Core Dump'om (zrzutom pamięci procesów)

```
-----
pid: 9495, tid: 10057, name: CameraBackgroun >>> com.example.krzysiek.carmas <<<
cannot get registers: No such process

backtrace:
#00 pc 0003b554 /system/lib/libc.so (__epoll_pwait+20)
#01 pc 000152a3 /system/lib/libc.so (epoll_pwait+26)
#02 pc 000152b1 /system/lib/libc.so (epoll_wait+6)
#03 pc 00012937 /system/lib/libutils.so (android::Looper::pollInner(int)+98)
#04 pc 00012b61 /system/lib/libutils.so (android::Looper::pollOnce(int, int*, void**)+92)
#05 pc 00085549 /system/lib/libandroid_runtime.so (android::NativeMessageQueue::pollOnce(_JNIEnv*, int)+22)
#06 pc 000b56cb /data/dalvik-cache/arm/system@framework@boot.oat

stack:
```

/data/anr - lokalizacja z logami związana z problemem "Application not responding"

```
"GCDaemon" daemon prio=5 tid=8 Waiting
| group="system" sCount=1 dsCount=0 obj=0x12c071c0 self=0xb4630400
| sysTid=29502 nice=0 cgrp=bg_non_interactive sched=0/0 handle=0xb3911000
| state=S schedstat=( 684539 11580538 4 ) utm=0 stm=0 core=0 HZ=100
| stack=0xafcdc000-0xafcde000 stackSize=1036KB
| held mutexes=
at java.lang.Object.wait!(Native method)
- waiting on <0x1cb243a3> (a java.lang.Daemons$GCDaemon)
at java.lang.Daemons$GCDaemon.run(Daemons.java:344)
- locked <0x1cb243a3> (a java.lang.Daemons$GCDaemon)
at java.lang.Thread.run(Thread.java:818)
```

dumpsys - polecenie pozwalające na zrzut całej pamięci systemu Android. Zawiera bardzo dużo informacji, m.in: stany aplikacji, działające service'y, konfiguracje środowisk, aktywne połączenia i wiele innych. Ogromne źródło informacji, łatwiej przefiltrować interesujących informacji niż analizować cały log. Poniżej zamieszczam parę ciekawszych zrzutów:

ostatnia znana lokalizacja

pamięć cache google quicksearch box'a

lokalizacja wszystkich baz danych dla danego pakietu - w tym przypadku facebook

bugreport - jak nazwa wskazuje - polecenie służy do raportowania bugów - również bardzo potężne narzędzie (w moim przypadku potokowanie go do pliku tekstowego zajęło parę minut, i dalej trwało)

powyższy zrzut dotyczy próby połączenia z siecią wi-fi

Pliki użytkownika

W projekcie nie skupiłem się nad plikami użytkownika typu: galerie zdjęć, pobrane pliki, dokumenty, nagrania głosowe itd. Wynika to z faktu że wersji Androida jest dosyć sporo i często niosą one ze sobą zmiany. Dodatkowo lokalizacje takich danych mogą wynikać z używanych przez właściciela telefonu aplikacji (np. pobrana ze sklepu Play aplikacja do robienia zdjęć może zapisywać je gdzie indziej niż ta domyślna).

Analiza tego typu plików na podstawie jednego urządzenia nie miałaby większego sensu. Biorąc pod uwagę analizę śledczą musimy spodziewać się ukrywania danych w przeróżnych lokalizacjach, dlatego najlepszym rozwiązaniem jest przeszukiwanie plików według ich rozszerzeń czy też nagłówków.

Przykładowe przydatne narzędzie, które można wykorzystać: [MobileFileSearch](#)

Własne narzędzie

Dodatkowo, w ramach projektu udało mi się napisać prosty program w pythonie, znacznie usprawniający prace nad analizą Androida.

Narzędzie składa się z dwóch programów - pierwszy kopiuje interesujące pod względem informatyki śledczej bazy danych na kartę sd:

```
databases = [
    '/data/data/com.android.providers.telephony/databases/mmssms.db',
    '/data/data/com.facebook.orca/databases/sms takeover_db',
    '/data/data/com.facebook.orca/databases/threads_db2',
    '/data/data/com.whatsapp/databases/wa.db',
    '/data/data/com.whatsapp/databases/msgstore.db',
    '/data/data/com.android.providers.contacts/databases/contacts2.db',
    '/data/data/com.tct.email/databases/EmailProvider.db',
    '/data/data/com.google.android.gm/databases/EmailProvider.db',
    '/data/data/com.android.browser/databases/browser2.db',
    '/data/data/com.android.browser/databases/websites.db',
    '/data/system/users/0/accounts.db',
    '/data/system_ce/0/accounts_ce.db',
    '/data/data/com.tct.note/databases/note.db',
    '/data/data/com.google.android.youtube/databases/identity.db',
    '/data/data/com.google.android.apps.maps/databases/gmm_sync.db',
    '/data/data/com.google.android.apps.maps/databases/gmm_myplaces.db',
    '/data/data/com.android.chrome/app_chrome/Default'
```

(tabela ze ścieżkami do poszczególnych baz danych)

Kolejny pobiera je do lokalnej pamięci komputera a następnie wykonując odpowiednie zapytania SQL uzyskuje tylko te najbardziej interesujące informacje:

```
dbs = [
    ['threads_db2', ['messages', '_id, text, sender, timestamp_ms, attachments, shares']], #facebook messenger
    ['msgstore.db', ['messages', '_id, key_remote_jid, data, timestamp, media size'], ['chat list', '_id, key_remote_jid, message table id, last_read_message table id, sort_timestamp']], #facebook messenger
    ['contacts2.db', ['data', '_id, raw_contact_id, data1, data2, data3, data4, data5'], ['calls', '_id, number, duration, date'], ['accounts', '_id, account_name, account_type']], #ko
    ['mmssms.db', ['sms', '_id, thread_id, address, date, date2, body, creator'], ['threads', '_id, date, date_sent, message_count, readcount, snippet']], #smsy domyslnie
    ['sms takeover_db', ['threads_table', 'threadid, state, ts, group_name, recipient_ids'], ['smsy za posrednictwem facebook messenger']], #smsy za posrednictwem facebook messenger
    ['wa.db', ['wa_contacts', '_id, jid, status, number, raw_contact_id']], #kontakty whatsapp
    ['EmailProvider.db', ['account', '_id, displayName, emailAddress, senderName'], ['Message', '_id, displayName, timeStamp, subject']], #e-mail
    ['browser2.db', ['history', '_id, title, url, date, visits']], #przeglądarka
    ['websites.db', ['websites', '_id, url, title']], #przeglądarka - zakładki
    ['accounts.db', ['accounts', '_id, name, type, password'], ['authtokens', '_id, type, authToken']], #konta google
    ['note.db', ['note text', '_id, note_id, content']], #notatki
    ['identity.db', ['identity', '_id, account, datasync_id']], #youtube
    ['gmm_sync.db', ['sync_item_data', 'corpus, client_id, server_id, latitude_e6, longitude_e6']] #maps
```

(tabela z bazami danych, oraz wybranymi dla danej bazy tabelami i kolumnami)

Program włącza serwer na porcie 8080, przez który możemy dostać się do interesującej nas bazy danych. Każdy request HTML odpowiada zapytaniu SQL do pliku bazy danych, następnie formatowanie wyniku w tabelę HTML, kolejno następuje wyświetlenie i ostylowanie jej w przeglądarce.

Poniżej zamieszczam parę screenów

id	title	url	date	visits
52	m t mobile pl	http://m.t-mobile.pl/	1507389814394	2
53	Rozkład jazdy	https://www.google.pl/search?hl=pl&oeuf=8&q=mpk+nowy+sacz&quib=1506957508117&source=browser-type&dev=loc=0	1506957523009	4
54	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=week&id=2700	1506697530055	1
55	Rozkład jazdy	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=week&id=2724	1507389877319	1
56	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=week&id=2724	1506957549591	1
57	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=week&id=2622	15073898710895	2
58	Rozkład jazdy	https://www.google.pl/search?hl=pl&oeuf=8&q=mpk+nowy+sacz&quib=1507389828173&source=browser-type&dev=loc=0	1507389834555	2
59	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=saturday&id=2700	150738981896	1
60	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=saturday&id=2724	1507389869213	1
61	ROZKLAD JAZDY	http://www.mpk.nowysacz.pl/policzenia-rozklad-jazdy/?view=show&type=saturday&id=2622	1508213667384	2

[illegible]

/contacts					
localhost:8080/contacts					
2351	1165	11 2	2	None	None
2352	1165	Centrum Powiadamiana Ratunkowego	Centrum	Ratunkowego	None
2353	1166	997	2	None	None
2354	1166	Policja	Policja	None	None
2355	1167	998	2	None	None
2356	1167	Straz Pozarna	Straz	Pozarna	None
2357	1168	999	2	None	None
2358	1168	Pogotowie Ratunkowe	Pogotowie	Ratunkowe	None
2359	1169	+48 790 200 200	2	None	+48790200200
2360	1169	Poczta glosowa	Poczta	glosowa	None

_id	number	duration	date
860	14	15	1611582156403
861	+48 14	5	1611582196565

_id	account_name	account_type
1	USIM1	USIM Account
2	2@gmail.com	com.google
3	Phone	Local Phone Account
5	USIM2	USIM Account
7	USIM3	USIM Account
9	USIM4	USIM Account
11	WhatsApp	com.whatsapp
12	USIM5	USIM Account

(Do projektu dołączam bazę z historią przeglądarki oraz notatnika - dla sprawdzenia poprawności działania wystarczy odpalić program poleceniem: `python android.py`, włączyć w przeglądarce adres: <http://localhost:8080> i wybrać jedną z udostępnionych baz, drugi program: `get_databases.py`, wymaga dostępu root do analizowanego telefonu oraz dołączonej do niego karty SD - ścieżka do karty jest wpisana na sztywno `/sdcard`, niektóre Android mogą "mountować" kartę pod inną ścieżką)

Bibliografia

Oprócz zamieszczanych w trakcie sprawozdania linków skorzystałem również z:

<https://android.stackexchange.com/questions/14430/how-can-i-view-and-examine-the-android-log> - dosyć obszerny wątek nt. logów w androidzie
https://www.group-ib.com/blog/whatsapp_forensic_artifacts - artefakty aplikacji WhatsApp
<https://android.stackexchange.com/questions/16915/where-on-the-file-system-are-sms-messages-stored> - wątek o lokalizacji bazy danych sms/mms
<https://android.stackexchange.com/questions/41455/where-is-the-data-for-contacts-storage-located> - wątek o lokalizacji rejestru połączeń
<https://android.stackexchange.com/questions/28296/how-to-fully-backup-non-rooted-devices> - trochę o backup'ie androida bez dostępu root
<https://stackoverflow.com/questions/38495426/how-to-open-adb-shell-and-execute-commands-inside-shell-using-python> - sposób na połączenie pythona i adb
<https://www.dataforensics.org/android-phone-forensics-analysis/> - artykuł o analizie śledczej Androida